FINAL YEAR PROJECT [CSC-499]

TRAFFIC VIOLATION DETECTION SYSTEM

SUBMITTED BY

MUHAMMAD KASHIF 16SBSCSM081

MUHAMMAD YASIR HASAN 16SBSCSM017

SUPERVISED BY

DR. SYED ASIF ALI

COURSE CO-ORDINATOR

MR. BASIT HASSAN

MS WAJIHA NAIM

DEPARTMENT OF COMPUTER SCIENCE

SINDH MADRESSATUL ISLAM UNIVERSITY

KARACHI

ACKNOWLEDGMENT

The success and outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Dr. Syed Asif Ali, for providing me an opportunity to do the project work and giving us all support and guidance, which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

I owe my deep gratitude to our project coordinators Mr. Basit Hassan and Mrs. Syeda Wajiha Naim, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of computer science, who helped us in successfully completing our project work. Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

Muhammad Kashif & Yasir Hassan,
Department of Computer Science,
Sindh Madressatul Islam University

TABLE OF CONTENTS

CHAPTER 1 6

TRAFFIC VIOLATION DETECTION SYSTEM (TVDS) 6

1. INTRODUCTION 6

2. BACKGROUND 6

3. PROBLEM STATEMENT 7

4. CONTRIBUTION 7

5. OBJECTIVE 8

6. CONSTRAINTS 8

7. SCOPE 8

8. VIOLATION DETECTION 9

9. SUMMARY 9

CHAPTER 2 10

LITERATURE REVIEW 10

1. REAL-TIME VEHICULAR TRAFFIC VIOLATION DETECTION 11

2. METHOD OF MONITORING VEHICULAR TRAFFIC 11

3. TRAFFIC VIOLATION DETECTION (VIDEO DEMO) 11

4. LOGIPIX TRAFFIC VIOLATION MANAGEMENT SOLUTION 12

4.1. FEATURES AND FUNCTIONS 12

5. PARKING MANAGEMENT (CityView) 13

3.1. POWER FEATURES 13

6. METRIC PARKING SYSTEMS 13

7. AIMS PARKING TICKET MANAGEMENT 14

7.1. SERVICES 14

7.2. AIMS SUCCESS STORIES 14

8. SECURE PARK 15

9. SUMMARY 16

CHAPTER 3 17

PROPOSED SYSTEM 17

1. BACKGROUND 17

2. SMART OBJECTIVES 17

3. IMPORTANCE OF SEATBELT 17

4. REQUIREMENT SPECIFICATIONS 17

5. METHODOLOGY 18

6. REQUIREMENT ENGINEERING PROCESS 19

7. TEAM STRUCTURE 19

8. PROCESS LIFECYCLE 20

9. WORKFLOW 21

10. SYSTEM DESIGN 21

CHAPTER 4 22

DESIGN AND ARCHITECTURE 22

1. GOAL **22**

2. CONCEPTUAL DIAGRAM **22**

2.1. LOW LEVEL DESIGN 24

2.2. HIGH LEVEL DESIGN 25

3. DATA FLOW DIAGRAM **26**

4. CLASS DIAGRAM **27**

5. USE CASES **28**

6. ACTIVITY DIAGRAM **29**

7. SUMMARY **30**

CHAPTER 5 **31**

SYSTEM IMPLEMENTATION **31**

1. SOFTWARE ARCHITECTURE **31**

2. DEVELOPMENT ENVIRONMENT **31**

3. APP ARCHITECTURE **31**

3.1. APP OVERVIEW 32

4. PROJECT CONFIGURATION **33**

5. FRONT END **33**

5.1. MAIN ACTIVITY 33

5.2. MANIFEST/PERMISSIONS 35

5.3. GOOGLE SERVICES 36

6. BACK-END **40**

6.1. FINAL TRAINED MODEL SUMMARY 40

CHAPTER 6 **43**

EVALUATION AND FUTURE WORK **43**

1. OVERVIEW **43**

2. OBJECT DETECTION **43**

2.1. OBJECT DETECTION ON MOBILE 44

3. THE COMPUTER VISION INDUSTRY **44**

4. DISTRACTED DRIVING **44**

5. KEY CAPABILITIES OF DETECTION MODEL **44**

5.1. FAST OBJECT DETECTION 44

5.2. OPTIMIZATION 45

5.3. PROMINENT OBJECT DETECTION 45

5.4. COARSE CLASSIFICATION 45

6. PRE-REQUISITE FOR IMAGE TRAINING **45**

6.1. DEEP LEARNING 45

6.2. TENSORFLOW 46

6.1. KERAS 46

7. IMPLEMENTATION PATH **46**

7.1. ASSEMBLE TRAINING DATA 46

7.2. TRAIN A NEW MODEL 46

7.3. USE THE MODEL IN YOUR APP 46

8. EVALUATION DETAILS **46**

9. CONFUSION MATRIX **48**

10. MODEL PRODUCTION **48**

11. OBJECT DETECTION AND FUTURE POSSIBILITIES **48**

REFERENCES **49**

CHAPTER 1

Traffic Violation Detection System (TVDS)

1. INTRODUCTION

In present world, road accidents are very common with the main reason being the careless and rash driving. The requirement to check this has been important and several approaches have been used. With the innovation in technology, different bodies are looking for computerized alternative to control this problem of reckless driving.

The seat belt is the single most effective feature in a vehicle to reduce the severity of injury to the vehicle occupants that results from road traffic crashes. The wearing of safety belts is compulsory for drivers and passengers of motor vehicles, occupying seats equipped with such belts. Seat-belts, of course, do not prevent a crash taking place. They do, however, play a crucial role in reducing the severity of injury to vehicle occupants involved in a crash.

We are offering a comprehensive solution to uncover traffic violations such as not wearing seat belts, not carrying driver's license, broken windshields putting fancy number plates etc. Quantity of motor vehicles has been increased rapidly in recent years. It is an obligation now to be aware of traffic signs and rules.

Our project is divided in three categories. First there is Image acquisition followed by its processing and finally detecting the violation. The situation there is compared with the preset threshold and camera is triggered if the violation is detected. Acquisition and transfer of image is done to the server via internet. The server runs an Image processing program which generates the penalty challan.

Another feature is number plate detection. Using image processing to classify number plate by isolating the number plate and optical character recognition to identify the characters of the number plate.

2. BACKGROUND

At present Traffic congestion problem occurs almost in all the big cities of Pakistan. Among the major urban cities like Karachi, Islamabad, Lahore, Faisalabad, Hyderabad etc. especially Karachi, the business hub of Pakistan, is now facing significant traffic congestion problem.

Though it is a common problem in our country, and it hampers our everyday life we need to think about this matter and come to a solution to reduce traffic congestion in the state. To come up with a solution we need to know why traffic congestion happens. Pakistan is sixth most populated country in the world. It is one of reason for traffic congestion. As population is increased no of vehicle increased. As a result, number of road accidents and traffic rule violation increased. In this project, we are developing an application to solve traffic congestion by detecting traffic rule violation and generate fine for the specific traffic violation rule.

3. PROBLEM STATEMENT

The existing system has loopholes and less flexibility. One must pay the fine at the moment. We are developing a system that will facilitate the user to pay the generated penalty slip within period of time (some days) which will be doubled otherwise. We are to design a system which identifies every license plate and be resistant to image noise and disruptions that may occur.

4. CONTRIBUTION

We are making an application to facilitate penalty and fine system. These fines and penalties will be applicable on violation of traffic rules and regulations. The violation will be detected by the smart system. This will be tracked by unique challan number. We have set these constraints as pre-requisite for proposed system. They include but not limited to:

1. High definition imaging
2. Nominal skewing or rotation
3. Better lighting

5. OBJECTIVE

The goal of this work is aimed at developing an application that depicts smart traffic offence analysis tool. Another appealing feature of this application is to develop a system that is faster, efficient and manageable. We are focused in preparing complete, integrated solution for traffic offence management.

6. CONSTRAINTS

Some factors may affect the efficiency of the system. These factors can be categorized into weather, plate placement, vehicle movement, lighting conditions, mechanical damages, image noises etc. Most of them can be avoided by using proper lighting, specialist toolkits for recording, creating proper workplace environment and appropriate image processing techniques.

7. SCOPE

With the advent of twenty first century, there is an inclination in the need for proper monitoring of road traffic. Old methods for traffic managements have loopholes and deficiencies. These systems are slow to response, bulky in installation and unable to detect vehicles properly.

We are designing an autonomous system based on an application that is easy to install and use prevailing infrastructure. This system can be upgraded easily and have more flexibility in redesign and maintenance by making minor changes in the algorithms. The proposed system allows proper monitoring and tracking of traffic violation detection by image classification techniques. Then comes some deep learning techniques to further breakdown the images and detect the type of violations.

8. VIOLATION DETECTION

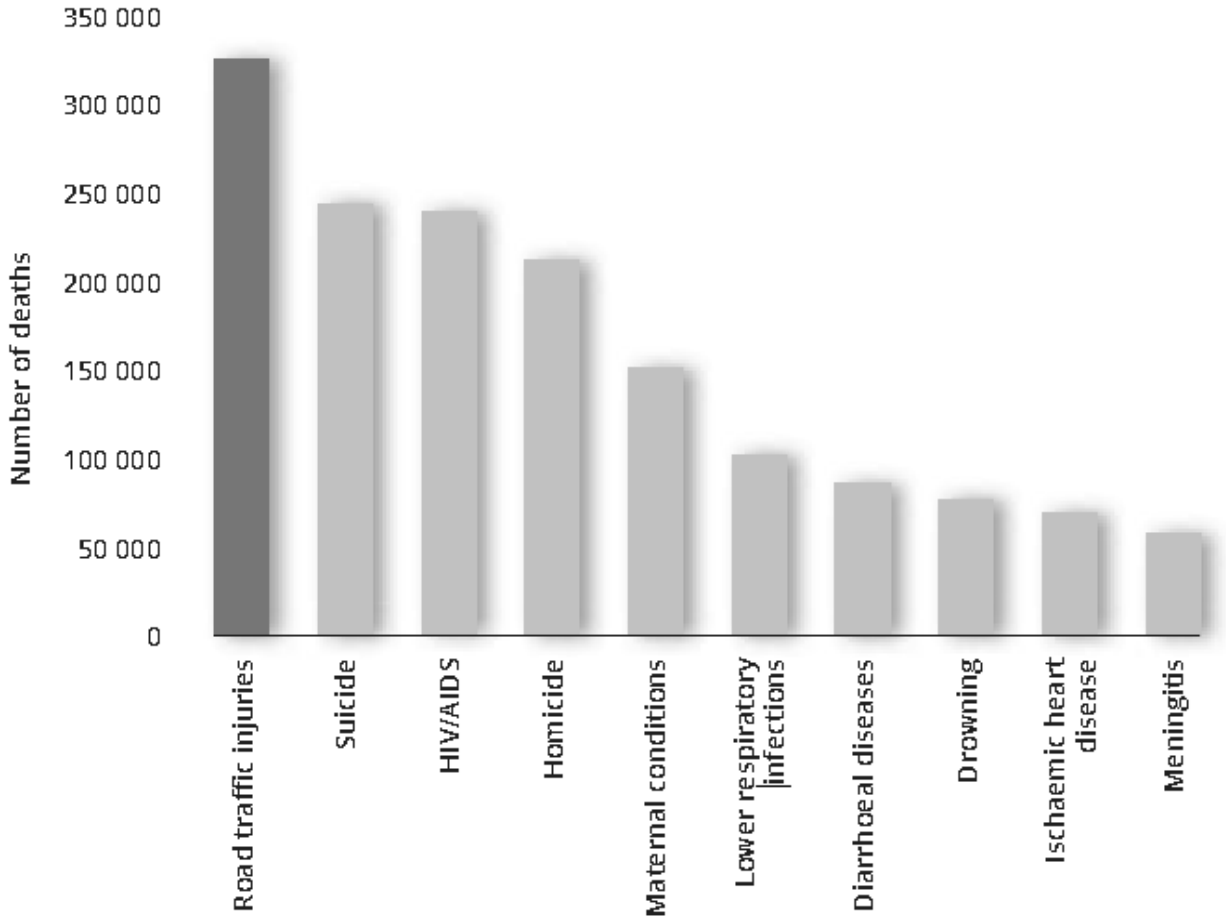
It is the most important part of this system. In Pakistan many people break the traffic rule and they are not punished. Our system can detect traffic rule violation and can fine the offender instantly. The use of traffic inspection images to implement automatic analysis of whether drivers and passengers wear seat belt can make up for the deficiencies in traditional traffic monitoring that mainly studies vehicle targets while ignoring the staff goals and their corresponding behavior analysis. The overall traffic regulation mode of ‘‘from car to person’’ has been realized and the dominant idea of ‘‘people-oriented’’ modern intelligent transportation system is embodied.

9. SUMMARY

In this chapter, we have discussed the introduction of our project, that is Traffic Violation Detection System. We need to develop an efficient system to minimize the possibility of violations of traffic laws and these violations in turn are cause of road accidents. Also, we have discussed different problems that require definite solutions, that is why we are developing this system. The existing system for challan and ticketing is manual and time consuming. It has many loopholes and blockheads. We are contributing in an efficient system that is manageable in different weather conditions, lighting, license plate placement and background noise.

CHAPTER 2
LITERATURE REVIEW

Road traffic injuries claim more than 1.2 million lives each year and have a huge impact on health and development. They are the leading cause of death among young people aged between 15 and 29 years, and cost governments approximately 3% of GDP. Despite this massive – and largely preventable – human and economic toll, action to combat this global challenge has been insufficient. Below is the graph that shows the major causes of death among people of ages 15-29.



Graph 1 Global Report on Road Safety by WHO, 2015

To design an efficient and smart system to detect traffic laws violations, we have reviewed some existing systems worldwide, those who depicts similar behaviors. Some of them are discussed in this chapter of literature review.

1. REAL-TIME VEHICULAR TRAFFIC VIOLATION DETECTION

Violations and offences of traffic rules are getting serious as the volume of traffic increasing. The existing systems are lack in ability to examine high amount of traffic monitoring and distinguish several types of violations in quick time. We are proposing an algorithm for that can determine many types of traffic violations happening on the roads. To obtain a real-time analysis, analogous computing techniques are being used. We are applying both, real time as well as synthetic data in our experiments. It has been observed from the results of those experiments that out proposed system can determine all types of violations, Ou et al., 2012.

2. METHOD OF MONITORING VEHICULAR TRAFFIC

Pietzsch (1997) patented this solution as a procedure for tracking traffic violations and offering warnings to drivers of traffic commotions, erroneous driving, unsafe road conditions, and severe weather. A mainframe computer network and a signal network are merged through a combined network to adjust the luminous elements by managing, if required, under time-bounded space and conditions.

3. TRAFFIC VIOLATION DETECTION (VIDEO DEMO)

A video analysis software to detect some kinds of violations by analyzing video streams from traffic cameras. The software is able to detect and store traffic violations that are related to intersections of vehicles with solid lines and traffic light violations.

This software was created for demonstration purpose and in some cases, it is not able to detect corresponding violations. In case of further development, it will be able to automatically detect most categories

of traffic violations and not only the ones which are related to solid line intersections and red light. To analyze a video and to detect the violations which are present in the video, the following approach was taken. First, models of the road and of vehicle movements are obtained by processing the video stream. Second, these models are combined and analyzed to detect traffic violations; Gevorgyan, 2017.

4. LOGIPIX TRAFFIC VIOLATION MANAGEMENT SOLUTION

Traffic violations in urban areas has become a crucial problem. They have an update to date solution for detection of violations and upholding safety of traffic in everyday life. Preventive measures for traffic violations are key to ensure the security of automobiles and pedestrians. End-to-end Traffic Violation solution by Logipix was developed with precisely above-mentioned issues in mind.

4.1.**FEATURES AND FUNCTIONS**

According to Logipix (2017), features and functions in their Traffic Violation Management Solution are:

- Autonomous operation.
- Built-in panel for input/output to connect all the traffic lights.
- Interface for ANPR system.
- Peripheral database and its interface with customer-managed lists.
- Compiled packages for violation.
- Interface for the authorities

5. PARKING MANAGEMENT (CityView)

Parking Management System by City View powers the process for tickets from top to bottom. Our system appreciates ease of data entry, capabilities for advanced search and autonomous penalty calculations. Scheduled permits for parking and tracking, inventory for lot, parking for suburbs, and towing management; all are improved by City View's system.

3.1. POWER FEATURES

CityView (2017) has highlighted the following features in their solution:

- Ease of Data Entry.
- Location Management.
- Exception Notifications
- Determining Fines.
- Reporting Capabilities

6. METRIC PARKING SYSTEMS

Metric Parking Systems has a moto to provide parking solutions, not just ticket machines. Their objective is to provide advanced system for parking and believe that the number of terminals creating a parking system that is right for any organization. They have 40 years of excellence in parking industry and their parking ticket machines; more than 60,000, are implemented in more than 45 countries across the world.

The METRIC Elite LS is a multi-service payment terminal and can be used for the following applications:

- Paying and Displaying Parking
- Paying by Space
- Barricade-controlled parking
- Payment Terminals
- EV Charging

In 2015, METRIC was presented with the Parking Technology Award at the British Parking Awards for the Vivo Park system, a barrier-free ANPR system that manages free parking in retail. The Bell Punch Company. The Ultimate Metric Parking Systems (2018) has Enlighted the following features in their current Elite (2012) solution with cutting edge technology:

- Parking Terminals
- Pay on foot parking system

7. AIMS PARKING TICKET MANAGEMENT

Parking System by AIMS records ticket for parking from the issuance to payment, and every phase of transaction. All the ticket issued are recorded and displayed on the AIMS portal where user can enter the records by hand, make fiscal transaction, post various comments, view transactions history, attach multiple files, make financial adjustments, view different categories of tickets, access appeals, and edit transaction One comprehensive window for all the needs.

7.1. SERVICES

AIMS being global organization for parking ticket management system, services the following aspects of world as:

- Municipalities
- Hospitals
- Higher Education
- Restaurant
- Airports

7.2. AIMS SUCCESS STORIES

- Las Vegas upgraded from their Car Tracking software and Duncan AutoCite in July 2014 (City of Las Vegas).
- Georgia Southern University substituted their existing software with AIMS (Georgia Southern University).

- The City of Syracuse Parking Violations Bureau transitioned from manual System to the AIMS Parking Management Software in 2002. They manage all aspects of parking enforcement and violation collections operation, from enforcement through DMV owner-retrieval, billing and collections. (City of Syracuse, New York)
- The Township of Lower Merion, PA implemented the AIMS system in 2008. They use AIMS system for in-house management of parking enforcement, collections and multiple parking permit programs. The AIMS system provides all the features and functionality we need for efficient ticket issuance and collections, real-time enforcement, field integration with our meter kiosk, online ticket payments and more. (Lower Merion, Pennsylvania)
- At University of Calgary, AIMS replaced aging in-house built system.
- At the University of Waterloo, they found AIMS fitting to their needs (University of Waterloo).
- The University of found their solution in AIMS system that overwhelms their prevailing system (University of Toronto at Scarborough, 2005).
- AIMS is simple to use (Brock University).

8. SECURE PARK

Secure Park provides an efficient system for parking that makes the resolves the issues. It has multiple features from simplicity, easiness and affordability. It has been proposed by numerous parking practitioners and can be availed by be used by anyone. Their products suite gives you tools to implement several systems from a thousand stand garage to single vehicle stand, without the complication or investment capital. With help of their Android application and Bluetooth printers, traffic patrollers can swiftly issue violation tickets.

Efficient database stores license plate and other information, across multiple sites. Not only that, but the reports can be generated pretty easily.

9. SUMMARY

Although road traffic injuries have been a leading cause of mortality for many years, most traffic crashes are both predictable and preventable. There is considerable evidence on interventions that are effective at making roads safer: countries that have successfully implemented these interventions have seen corresponding reductions in road traffic deaths. Rolling out these interventions globally offers huge potential to mitigate future damage and save lives at a global level.

The proposed application helps traffic police to penalize traffic offences. And helps them in analyzing traffic details with centralized data repository stored in the department server. There by generates graphs for the easier analysis by which the traffic police can enforce the traffic discipline. By this application, we can reduce the traffic offences drastically.

CHAPTER 3
PROPOSED SYSTEM

The overall system is distributed in three stages: acquisition of image and its transfer, preprocessing and Violation detection. The situation there is compared with the preset threshold and camera is triggered if the violation is detected. Acquisition and transfer of image is done to the server via internet. The server runs an Image processing program which generates the penalty challan.

1. BACKGROUND

The system is developed to detect violations of traffic rules and how their report against drivers. On the other hand, the system being developed must be affordable, and provide alternatives in selection the right equipment.

2. SMART OBJECTIVES

For defining the objectives of the system, it should be ensured that each one of them should be SMART (specific, measurable, achievable, realistic and time bounded). It is a good practice to at least consider the amount of these conditions might met. All in all, the process goes on and on and new information becomes available as the project grows. Changes will be needed to accommodate all the features of objectives and categorizations of tasks in achievement of the completed project.

3. IMPORTANCE OF SEATBELT

Wearing seat belt allows you sit in same posture with little jerk. If you are not wearing you might get back pain even with small accident. Drivers and other passengers sitting in a vehicle who are not wearing the seat belts can become a projectile when car strikes. They can hit wind screen and get serious injuries. There is a quite high chance of moving out of the vehicle through windows or doors in case of accident. Getting out from vehicle is the most dangerous situation one can face.

4. REQUIREMENT SPECIFICATIONS

The process of writing down the user and system requirement in a document is termed as requirement specifications. These requirements should be clear, concise, comprehensive and on the other hand they should be easy to comprehend. Practically, it is tough to attain to stakeholders and interpret them the requirements of system and there can arise conflicts and inconsistencies in the requirements.

5. METHODOLOGY

Here we are using mixed team organization. As the name suggests, it extracts the ideas from both the democratic organization and the chief-programmer organization. This mixed team structure is very popular and used in many software developments companies. See fig below for reference:

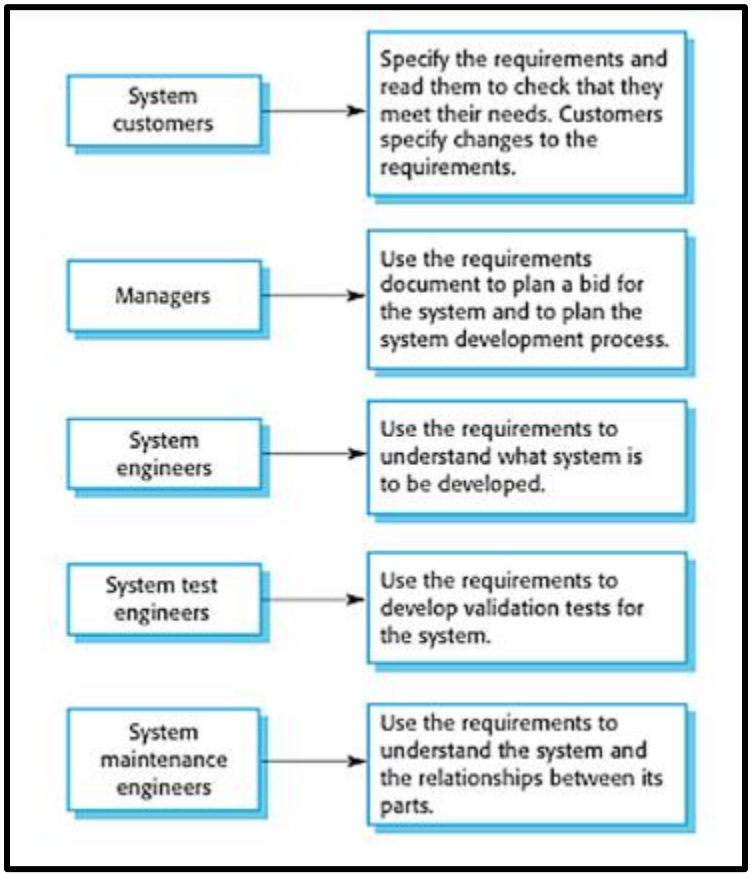


Figure 1: Methodology

6. REQUIREMENT ENGINEERING PROCESS

Requirement engineering is a process having four steps, which are mentioned below:

- Case Study
- Gathering Requirement
- Specification for Software Requirement
- Validating Software Requirement

7. TEAM STRUCTURE

Team structure highlights the issue of any institution or group of the individual project teams. There are a few viable examples in which the individual teams can be structured. There are generally three kinds of team structures, that is chief programmer, democratic, and the mixed team organizations. Although there are numerous variants of these structures. Various problems require different structures for team.

At this phase, expertise and skill defines the roles required to complete main tasks. This can be the team recruitment, criteria selection. If the team have been identified, or the team leaders have been chosen, there comes the opportunity to involve them in deciding the team structure. Key responsibilities might be assigned as well.

8. PROCESS LIFECYCLE

- There is a Product Owner at the top o SRUM. A person representing the end user's greatest interest who has the power to develop the final product.
- That person is responsible for creating the Backlog, which is a catalog of including tasks and needs of the final product. That backlog should be prioritized.
- For example, if the Scrum is used to design a car, steps like "Must have an engine" would be the top of priority as the car would not work without it. "Painted red" would be of lower priority however, it might be valuable to me, but it is not the requirement for the car to operate.
- Then there is Sprint. It is a preset timeframe that is required by the team to complete the tasks from the Backlog. This timeframe depends on the efficiency of the team, typically it is two weeks.
- Team meeting is an everyday task to give periodic updates in the Daily Scrum. This process is also known as "Daily Stand-Ups."
- Every Sprint close with a review, or Presentation, that enables the team to review the work completed and finding different ways to improve it in next sprint.

9. WORKFLOW

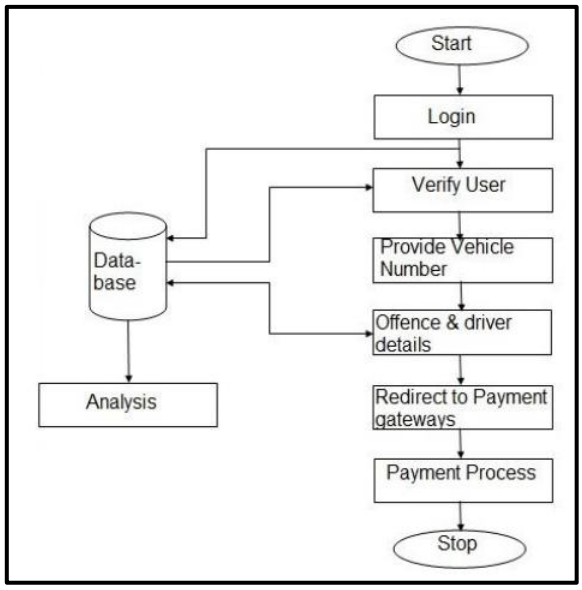


Figure 2: Workflow

10. SYSTEM DESIGN

The whole system of Traffic Violation Detection System has following components:

- **PROVIDE VEHICLE NUMBER:** After the successful login, the user has to provide the vehicle number of the vehicle rider who has committed the traffic offence.
- **OFFENCE AND DRIVER DETAILS:** Once the vehicle driver is caught, if his details are not available in the database then all the information regarding the offence committed are stored in the database for future reference and the notification will be sent to his mobile number provided by him.
- **DETECTION:** Once the vehicle driver is caught if his/her details are already available in the database and if the offences committed by him has crossed the fixed limit then the police can either seize the vehicle or can refer to cancel the license of the vehicle rider.
- **ANALYSIS:** Using the information available in the database, the higher traffic authorities can analyze the data and they can take necessary measures to create awareness among the public regarding traffic discipline

CHAPTER 4
DESIGN AND ARCHITECTURE

Here in the design phase of traffic violation detection system, there is an emphasis to develop a details specification for achieving a viable solution for user's information for technological requirements. That can be system or functional requirement including logical description of entities, attributes, relations, schema and all the data that is documented during the requirement analysis of the system. After designing system architecture, analysis phase is further extended to allocate the system and database design in a way that suits the implementation using all the physical and logical constraints of the environment.

1. GOAL

Design phase is used to convert all the documented details into system design specifications. After the approval of the design from all the stakeholders, the team begins the development and moves to the system implementation and development phase.

In design phase, all the features from analysis and definition phase of the system are extracted to complete a list of requirements to implement the system. More than one designs are created during this phase of process until it meets the whole requirement. Products and milestone of design phase may vary from project to project and they include diagrams, sketches, flowcharts, site maps, screen mockups, wireframes, prototypes, photo impression, schemas etc.

2. CONCEPTUAL DIAGRAM

The idea behind traffic violation system is to provide a panel to automatically track traffic violations. When the traffic constable takes the image, it will detect the type of violation and generates the challan accordingly. Below is the conceptual diagram of the system.

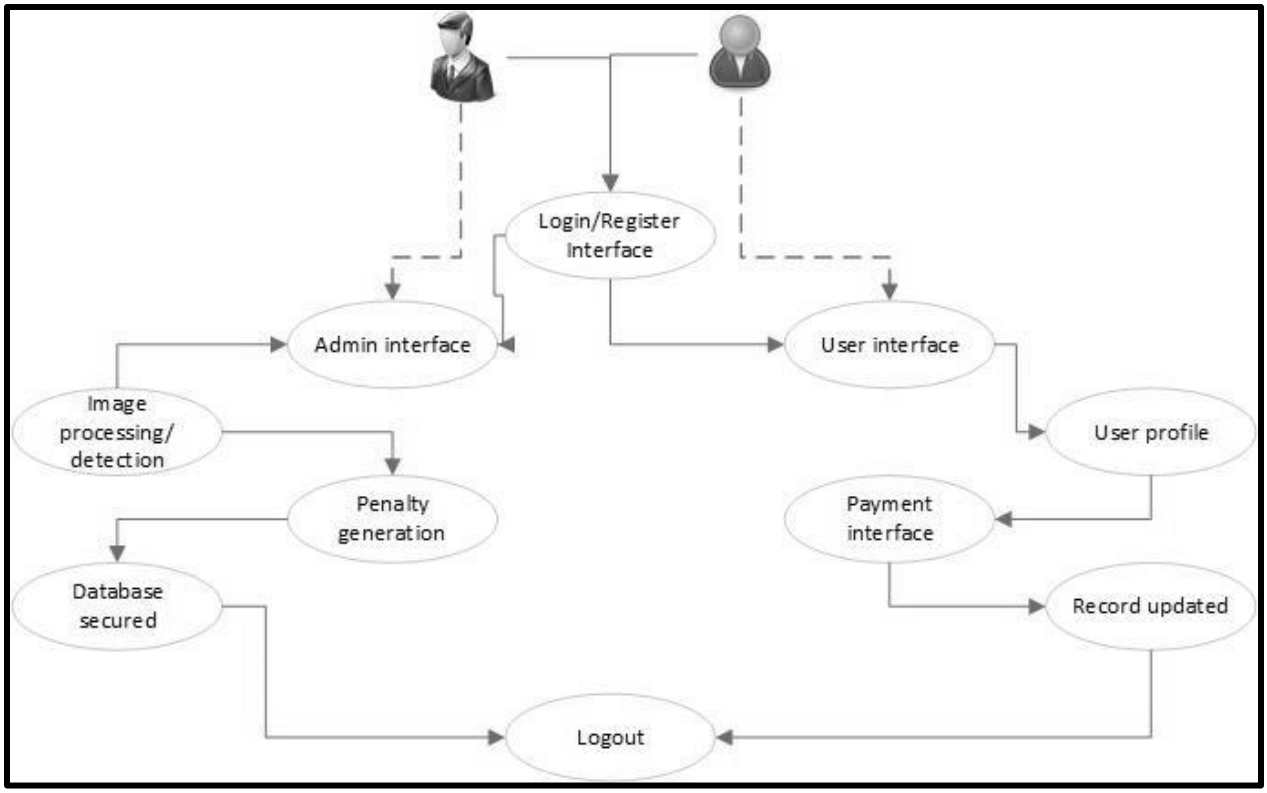


Figure 1: Conceptual Diagram

2.1. LOW LEVEL DESIGN

Picture below depicts the simple working of the system which is the level 0 or low-level context diagram.

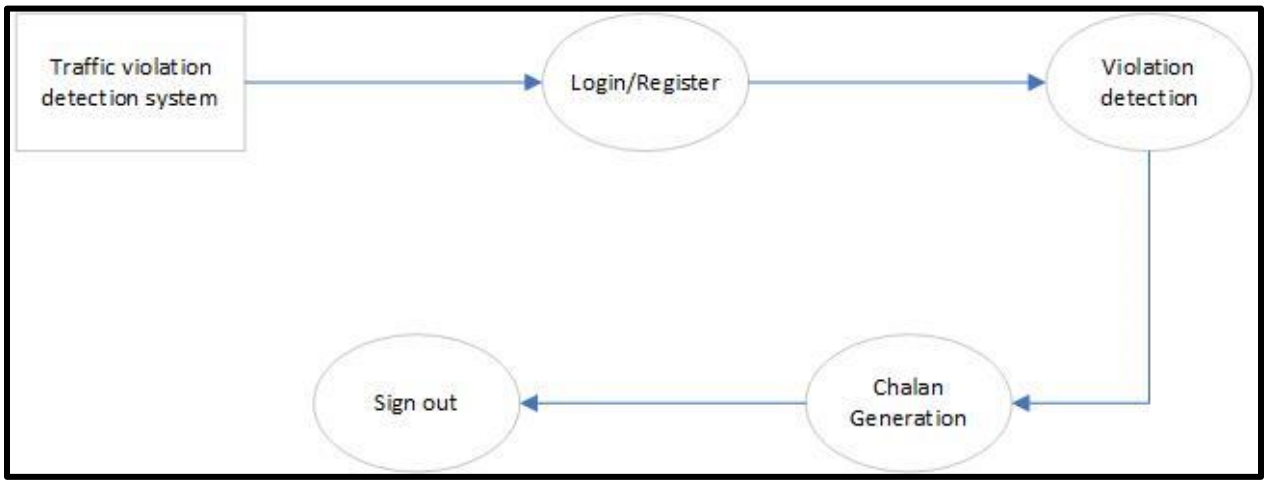


Figure 2 Low Level Design

User enters the system by logging in or registration process. Then system prompts to detect any violation if occur. When the user opts for violation detection, the system does so on the image just been captured. According to the type of violation happened, the system generates the challan.

2.2. HIGH LEVEL DESIGN

Below is the context level 1 or high-level design of traffic violation detection system.

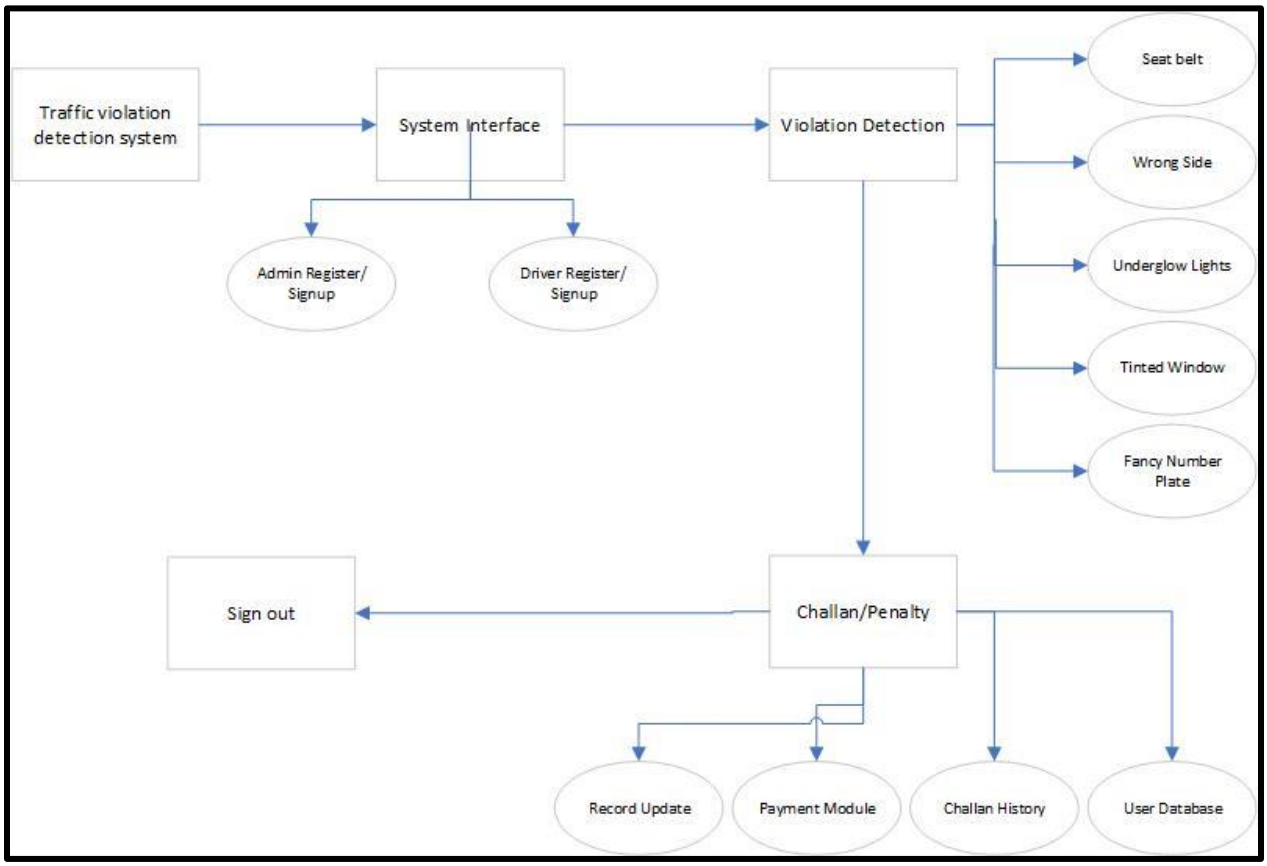


Figure 3: High Level Design

TVDS starts with the user interface that enables the login or registration features. User has a choice to either enter the dashboard or starts detecting any violation. Moving on to detection phase, there can be several types of traffic violations, i.e. seat belt, wrong side, under glow headlights, broken windshields, fancy number plates etc. According to the type of violation incurred, the challan is generated and then it redirects the user to payment.

3. DATA FLOW DIAGRAM

The picture below depicts the flow of data throughout the system.

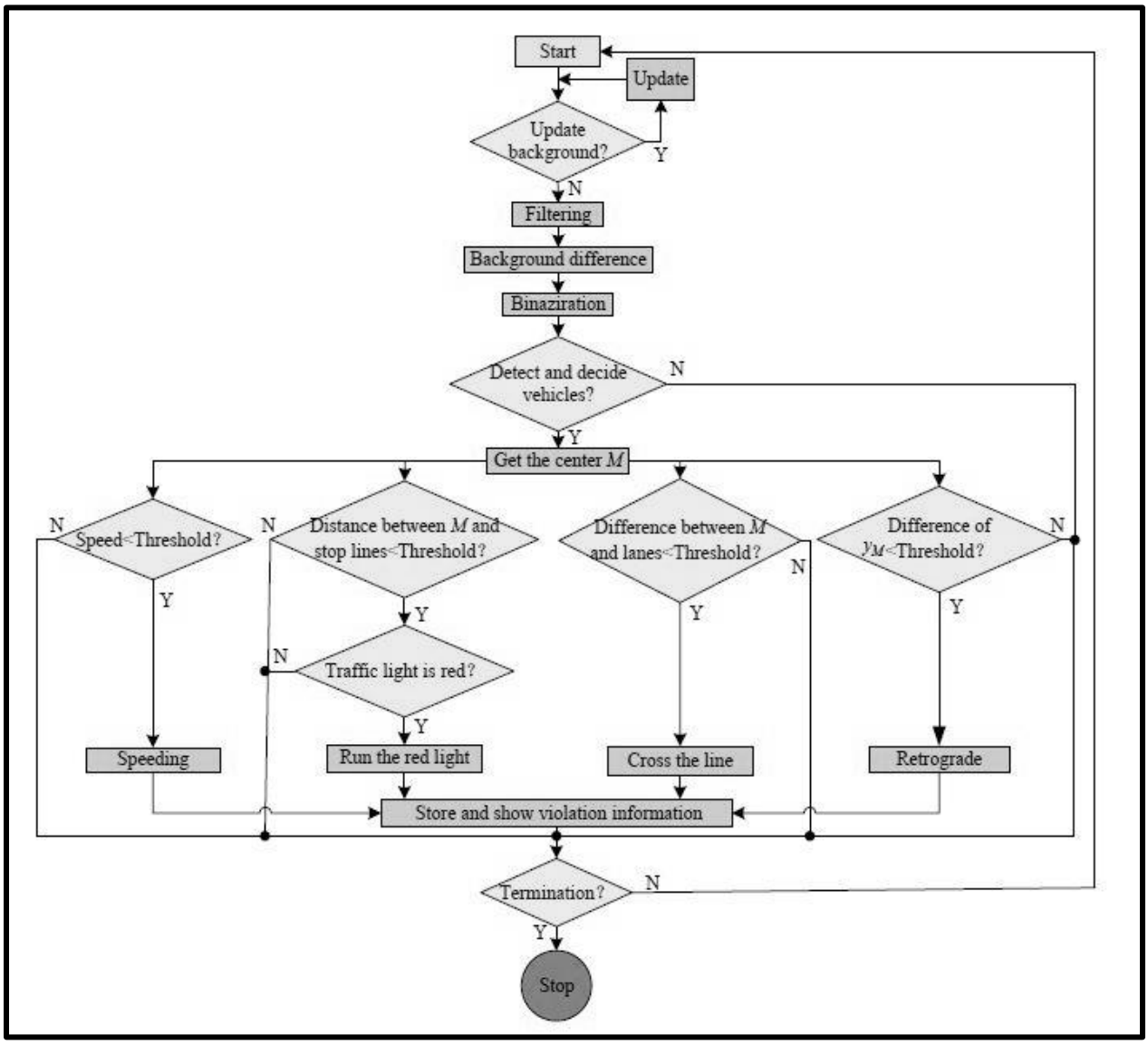


Figure 4: Data Flow Diagram

4. CLASS DIAGRAM

The image below illustrates the UML class diagram of traffic violation detection system:

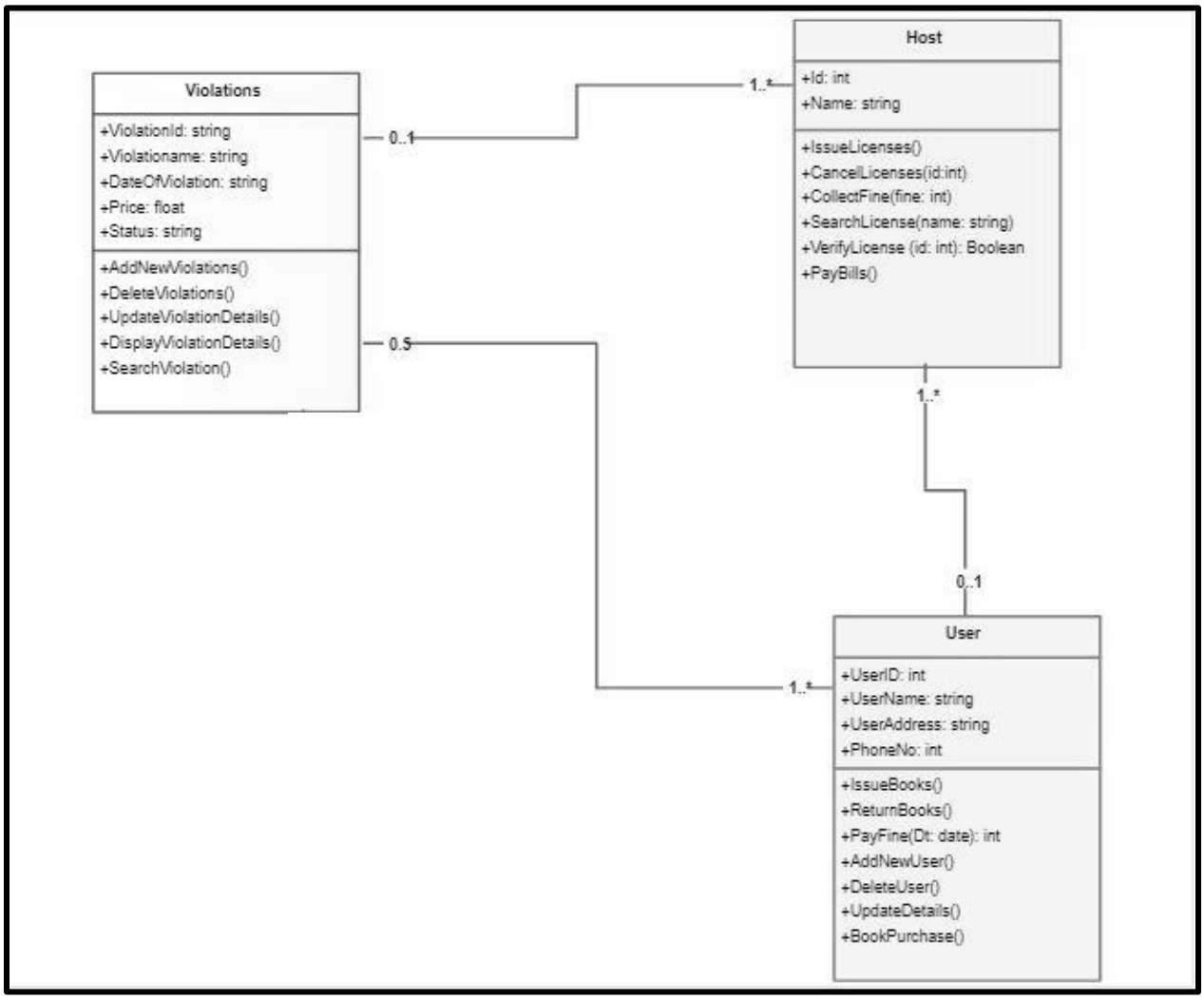


Figure 5: UML Class Diagram

UML diagram above shows the tables and schema for the system. It has mainly three tables for violations, system host and end user. As diagram already depicts, all the tables above have there attributes with their types and calling methods as well.

5. USE CASES

The picture below depicts the overall use caes of traffic violation detection system.

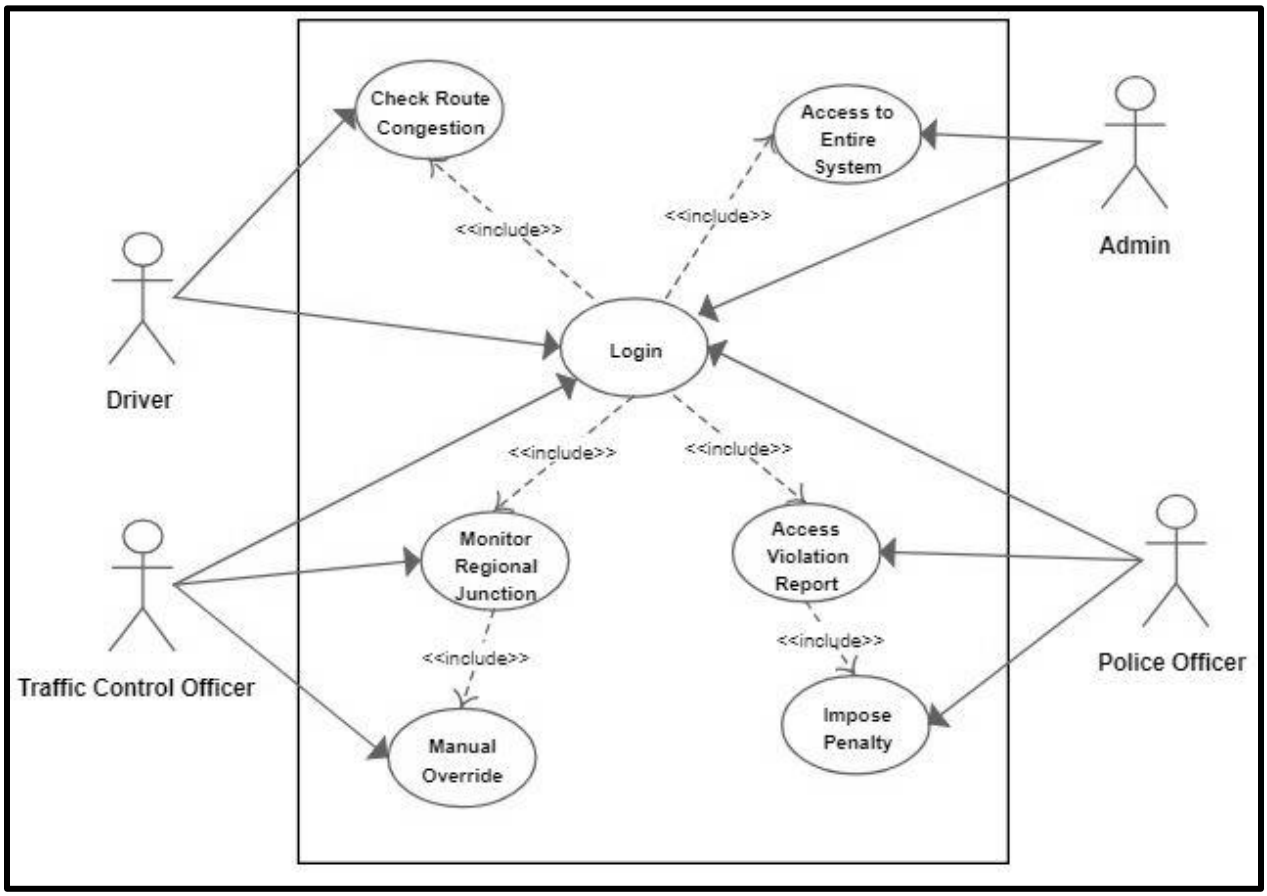


Figure 6: Use Cases

The above diagram is the combine use case of traffic violation detection system for user (driver), traffic constable and system admin.

6. ACTIVITY DIAGRAM

Following picture shows the activity diagram of traffic violation detection system.

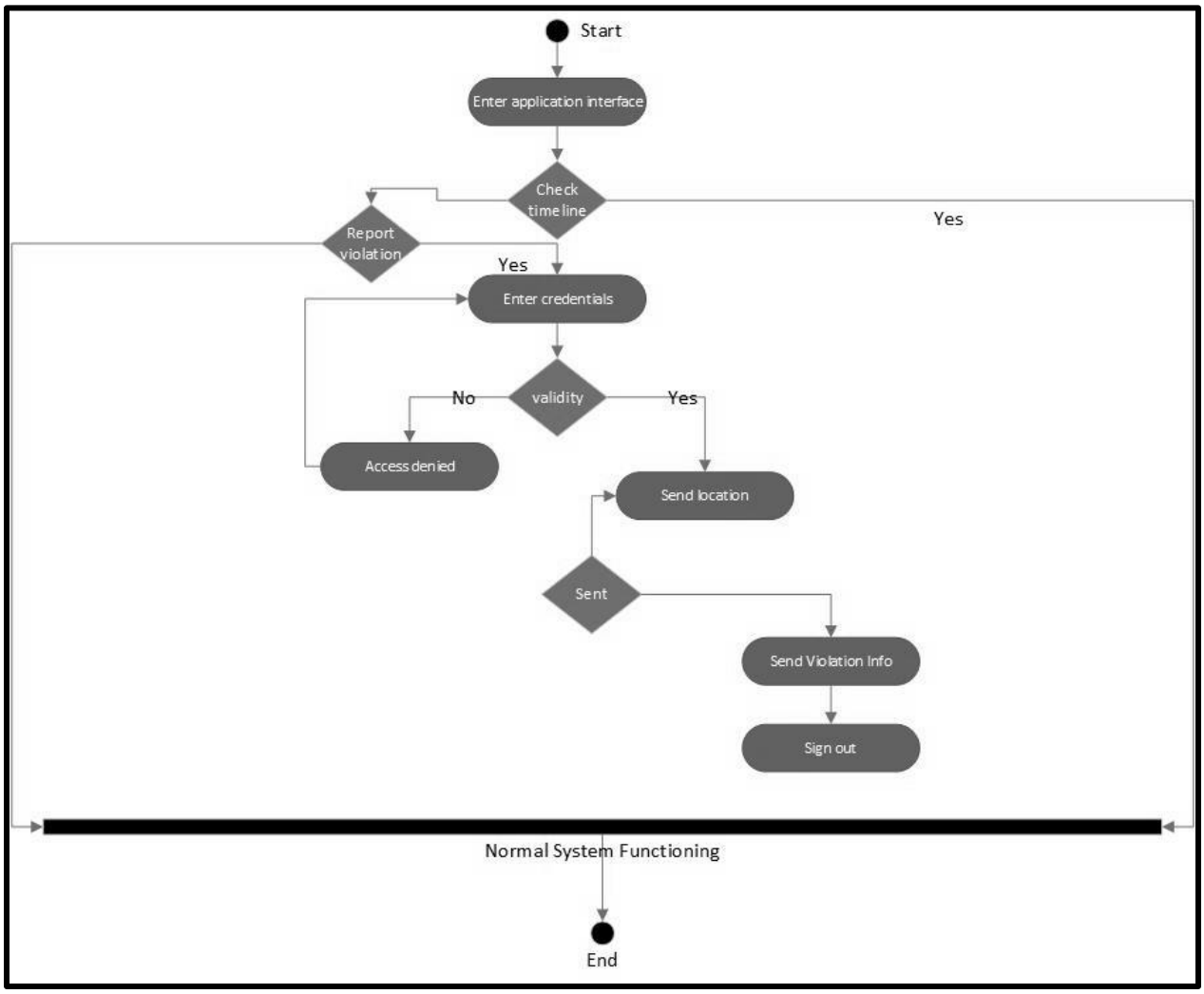


Figure 7: Activity Diagram

7. SUMMARY

Not every model or diagram designed in this phase meets the requirement on the whole. There can be several models, and, in each improvement, they reach to perfection. Design is for human perceive. It is not the actual requirement of the system. During the process, poor designs are must. First attempted perfection is very rare (almost to negligible).

CHAPTER 5
SYSTEM IMPLEMENTATION

1. SOFTWARE ARCHITECTURE

Structural design for Traffic violation detection system (TVDS) provides introductory services for system design. It is an abstract model to define the architecture, behavior, and more views of the system.

2. DEVELOPMENT ENVIRONMENT

Traffic Violation Detection is a server-based application. The main architecture is based on python and integrated with android. The Interface is designed using android interface().

3. APP ARCHITECTURE

App architecture of Traffic violation detection system (TVDS) is made simple and consistent. For example, consider that happens when a picture is taken with the system and how it triggers:

- Violation detection triggers the camera intent. It launches an app to handle device camera on the request. Current activity is paused, and the camera activity is created and started.
- Activity that layers the camera application might triggers other intents as well, that is the launch the device explorer, which may launch yet another app or the device's internet browser to view the penalty records of current user .
- After the picture is taken, it triggers the violation and alerts the system.

3.1. **APP OVERVIEW**

We are designing an application for the system, traffic violation detection that can be extended for web or android application. This system consists of different views and fragments that are shown in picture below:

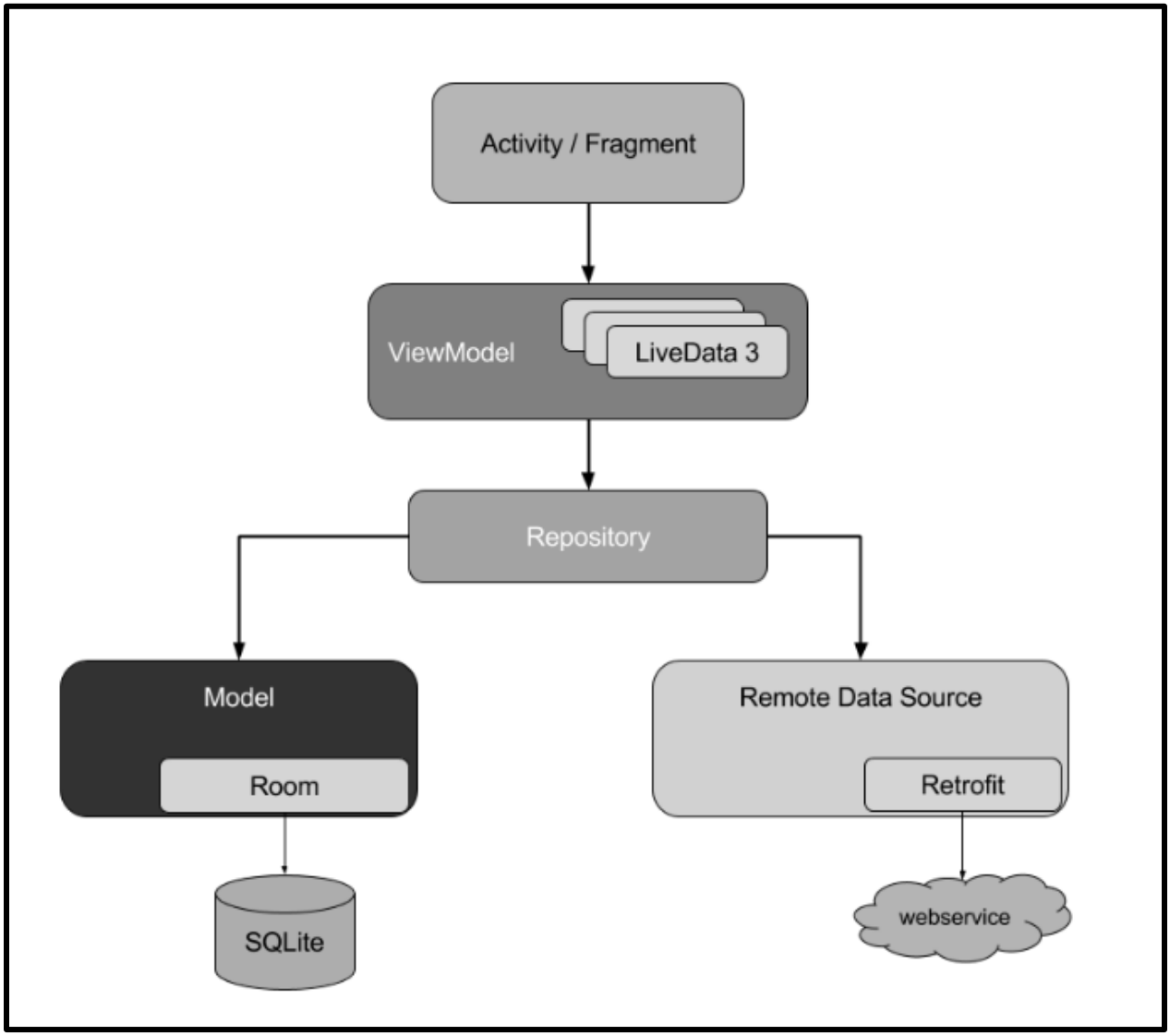


Figure 1: View Model

4. PROJECT CONFIGURATION

Name: Traffic Violation Detection System,

Front-Language: Android (Kotlin & XML)

Backend-Language: Python

Model base: Model is based on Python 3.8.

Model Evaluation: Firebase

5. FRONT END

5.1. MAIN ACTIVITY

```
package com.google.firebase.codelab.mlkit.automl
import android.content.pm.PackageManager
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat

/** Base activity that requests all needed permission at launch */
abstract class BaseActivity : AppCompatActivity(),
  ActivityCompat.OnRequestPermissionsResultCallback {
  private val requiredPermissions: Array<String> by lazy {
    try {
      this.packageManager.getPackageInfo(
        this.packageName,
        PackageManager.GET_PERMISSIONS
      ).requestedPermissions ?: arrayOf()
    } catch (e: PackageManager.NameNotFoundException) {
      arrayOf<String>()
```

```
  }
 }
 private fun allPermissionsGranted() = requiredPermissions.none { !isPermissionGranted(it) }
 private fun requestRuntimePermissions() {
  val allNeededPermissions = requiredPermissions.filter { !isPermissionGranted(it) }
  if (allNeededPermissions.isNotEmpty()) {
   ActivityCompat.requestPermissions(
    this,
    allNeededPermissions.toTypedArray(),
    PERMISSION_REQUESTS
   )
  }
 }
 private fun isPermissionGranted(permission: String): Boolean {
  when (ContextCompat.checkSelfPermission(this, permission)) {
   PackageManager.PERMISSION_GRANTED -> {
    Log.i(TAG, "Permission granted: $permission")
    return true
   }
   else -> {
    Log.i(TAG, "Permission NOT granted: $permission")
    return false
   }
  }
 }

 override fun onCreate(savedInstanceState: Bundle?) {
  super.onCreate(savedInstanceState)

  if (!allPermissionsGranted()) {
   requestRuntimePermissions()
```

```
    }
  }
  companion object {
    /** Tag for the [Log].  */
    private const val TAG = "BaseActivity"
    private const val PERMISSION_REQUESTS = 1
  }
}
```

5.2. MANIFEST/PERMISSIONS

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        package="com.google.firebase.codelab.mlkit.automl">

    <uses-permission android:name="android.permission.CAMERA"/>

    <uses-feature android:name="android.hardware.camera"/>
    <uses-feature android:name="android.hardware.camera.autofocus"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/MaterialTheme"
        tools:ignore="GoogleAppIndexingWarning">

        <activity
            android:name=".StillImageActivity"
            android:screenOrientation="portrait">

            <intent-filter>
```

```
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>

    </activity>

    <provider
        android:name="androidx.core.content.FileProvider"
        android:authorities="com.google.firebase.codelab.mlkit.automl.fileprovider"
        android:exported="false"
        android:grantUriPermissions="true">
        <meta-data
            android:name="android.support.FILE_PROVIDER_PATHS"
            android:resource="@xml/file_paths"/>
    </provider>
  </application>

</manifest>
```

5.3. GOOGLE SERVICES

```
{
  "project_info": {
    "project_number": "291047932720",
    "firebase_url": "https://tvds-e75ae.firebaseio.com",
    "project_id": "tvds-e75ae",
    "storage_bucket": "tvds-e75ae.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:291047932720:android:abd2389cfa79a53c4cfef5",
```

```
      "android_client_info": {
        "package_name": "com.google.firebase.codelab.mlkit.automl"
      }
    },
    "oauth_client": [
      {
        "client_id": "291047932720-
c3kfdr6u0h75jf7m3ldfbkgikcmvd9hq.apps.googleusercontent.com",
        "client_type": 3
      }
    ],
    "api_key": [
      {
        "current_key": "AIzaSyA8AmLpiTREvzIyZpUyl6AYivOKQZ-VgFs"
      }
    ],
    "services": {
      "appinvite_service": {
        "other_platform_oauth_client": [
          {
            "client_id": "291047932720-
c3kfdr6u0h75jf7m3ldfbkgikcmvd9hq.apps.googleusercontent.com",
            "client_type": 3
          }
        ]
      }
    }
  },
  {
    "client_info": {
      "mobilesdk_app_id": "1:291047932720:android:05e62d7f1be7624b4cfef5",
```

```
    "android_client_info": {
     "package_name": "org.tensorflow.lite.examples.classification"
    }
   },
   "oauth_client": [
    {
     "client_id": "291047932720-
1bdkbhahce94o4ummqt10sfifenu0tmb.apps.googleusercontent.com",
     "client_type": 1,
     "android_info": {
      "package_name": "org.tensorflow.lite.examples.classification",
      "certificate_hash": "a55f00115f4ee8efa203c2eb3c750745aab2c0e5"
     }
    },
    {
     "client_id": "291047932720-
c3kfdr6u0h75jf7m3ldfbkgikcmvd9hq.apps.googleusercontent.com",
     "client_type": 3
    }
   ],
   "api_key": [
    {
     "current_key": "AIzaSyA8AmLpiTREvzIyZpUyl6AYivOKQZ-VgFs"
    }
   ],
   "services": {
    "appinvite_service": {
     "other_platform_oauth_client": [
      {
       "client_id": "291047932720-
c3kfdr6u0h75jf7m3ldfbkgikcmvd9hq.apps.googleusercontent.com",
```

```
            "client_type": 3
          }
        ]
      }
    }
  },
  {
    "client_info": {
      "mobilesdk_app_id": "1:291047932720:android:4833e549af11ba294cfef5",
      "android_client_info": {
        "package_name": "org.tensorflow.lite.examples.detection"
      }
    },
    "oauth_client": [
      {
        "client_id": "291047932720-
0puuj5kiin3rnajutsv6tatv5nfr6iql.apps.googleusercontent.com",
        "client_type": 1,
        "android_info": {
          "package_name": "org.tensorflow.lite.examples.detection",
          "certificate_hash": "a55f00115f4ee8efa203c2eb3c750745aab2c0e5"
        }
      },
      {
        "client_id": "291047932720-
c3kfdr6u0h75jf7m3ldfbkgikcmvd9hq.apps.googleusercontent.com",
        "client_type": 3
      }
    ],
    "api_key": [
      {
```

```
        "current_key": "AIzaSyA8AmLpiTREvzIyZpUyl6AYivOKQZ-VgFs"
       }
      ],
      "services": {
       "appinvite_service": {
        "other_platform_oauth_client": [
         {
          "client_id": "291047932720-
c3kfdr6u0h75jf7m3ldfbkgikcmvd9hq.apps.googleusercontent.com",
          "client_type": 3
         }
        ]
       }
      }
     }
    ],
    "configuration_version": "1"
}
```


6. BACK-END

6.1. FINAL TRAINED MODEL SUMMARY

Model: "sequential_1"

_____

Layer (type) Output Shape Param #

==========

conv2d_1 (Conv2D) (None, 148, 148, 32) 896

_____

activation_1 (Activation) (None, 148, 148, 32) 0

_____

max_pooling2d_1 (MaxPooling2 (None, 74, 74, 32) 0

_____

conv2d_2 (Conv2D) (None, 72, 72, 32) 9248

_____

activation_2 (Activation) (None, 72, 72, 32) 0

_____

max_pooling2d_2 (MaxPooling2 (None, 36, 36, 32) 0

_____

conv2d_3 (Conv2D) (None, 34, 34, 64) 18496

_____

activation_3 (Activation) (None, 34, 34, 64) 0

_____

max_pooling2d_3 (MaxPooling2 (None, 17, 17, 64) 0

_____

flatten_1 (Flatten) (None, 18496) 0

_____

dense_1 (Dense) (None, 64) 1183808

_____

activation_4 (Activation) (None, 64) 0

_____

dropout_1 (Dropout) (None, 64) 0

_____

dense_2 (Dense) (None, 1) 65

_____

activation_5 (Activation) (None, 1) 0

==========

Total params: 1,212,513

Trainable params: 1,212,513

Non-trainable params: 0

CHAPTER 6
EVALUATION AND FUTURE WORK

1. OVERVIEW

With object detection and tracking API, it is possible to localize and track in real time the most prominent objects in an image or live camera feed. One can also optionally classify detected objects into one of several general categories.

Object detection and tracking with coarse classification is useful for building live visual search experiences. Because object detection and tracking happen quickly and completely on the device, it works well as the front end of a longer visual search pipeline. After the detection and filtration of objects, segmented data of object can be moved to a cloud backend, or to a custom ML model. A custom model is designed here in traffic violation detection system, that can be extended for any web or android application.

Nowadays vehicles are increasing exponentially. We need to be extra careful about the road incident. As there are a couple of things around you, which can distract the driving. We can use this model and implement a smart system, which will monitor driver activities at the time of driving and can alert him about the activities. It will reduce the number of roads accidents and will increase road safety.

2. OBJECT DETECTION

With recent advancements in deep learning-based computer vision models , object detection applications are easier to develop than ever before. Besides significant performance improvements, these techniques have also been leveraging massive image datasets to reduce the need for large datasets. In addition, with current approaches focusing on full end-to-end pipelines, performance has also improved significantly, enabling real-time use cases.

2.1. OBJECT DETECTION ON MOBILE

The benefits of using object detection aren't limited to applications that run on servers or in the cloud. In fact, object detection models can be made small and fast enough to run directly on mobile and edge devices, opening up a range of possibilities, including applications for real-time video surveillance, crowd counting, anomaly detection, and more. From brand loyalty, to user engagement and retention, and beyond, implementing object detection on-device has the potential to delight users in new and lasting ways, all while reducing cloud costs and keeping user data private.

3. THE COMPUTER VISION INDUSTRY

The computer vision industry is projected to hit $48.6 billion by 2022, with applications in industries ranging from video collaboration and medicine to education and security. Computer vision provides machines with the ability to see and visually sense the world around them, similar to how humans use their own eyes. It pertains to the automatic extraction, analysis and understanding of useful information from one or several images. Object detection is one form of computer vision that's gaining momentum in both the enterprise and consumer-facing tech communities.

4. DISTRACTED DRIVING

We've all been there: a light turns green and the car in front of you doesn't budge. Or, a previously unremarkable vehicle suddenly slows and starts swerving from side-to-side. When you pass the offending driver, what do you expect to see? You certainly aren't surprised when you spot a driver who is texting, seemingly enraptured by social media, or in a lively hand-held conversation on their phone. According to the CDC motor vehicle safety division, one in five car accidents is caused by a distracted driver. Sadly, this translates to 425,000 people injured and 3,000 people killed by distracted driving every year.

5. KEY CAPABILITIES OF DETECTION MODEL

5.1. FAST OBJECT DETECTION

Detect objects and get their location in the image. Track objects across images.

5.2. OPTIMIZATION

The object detection and tracking model is optimized for mobile devices and intended for use in real-time applications, even on lower-end devices.

5.3. PROMINENT OBJECT DETECTION

Automatically determine the most prominent object in an image.

5.4. COARSE CLASSIFICATION

Classify objects into broad categories, which you can use to filter out objects you're not interested in. The following categories are supported: home goods, fashion goods, food, plants, places, and unknown.

6. PRE-REQUISITE FOR IMAGE TRAINING

Following are the mandatory requirement (skill) one should possess to train a deep learning image classification model:

- Python Programming
- Basics of Machine Learning
- Basics of neural networks

6.1. DEEP LEARNING

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

6.2. TENSORFLOW

TensorFlow is an open-source deep learning framework created by Google Brain. Tensorflow's Object Detection API is a powerful tool which enables everyone to create their own powerful Image Classifiers. No coding or programming knowledge is needed to use Tensorflow's Object Detection API. But to understand it's working, knowing python programming and basics of machine learning helps.

6.1. KERAS

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research. Allows for easy and fast prototyping. Supports both convolutional networks and recurrent networks, as well as combinations of the two.

7. IMPLEMENTATION PATH

7.1. ASSEMBLE TRAINING DATA

Put together a dataset of examples of each label you want your model to recognize.

7.2. TRAIN A NEW MODEL

In the Firebase console, import your training data and use it to train a new model.

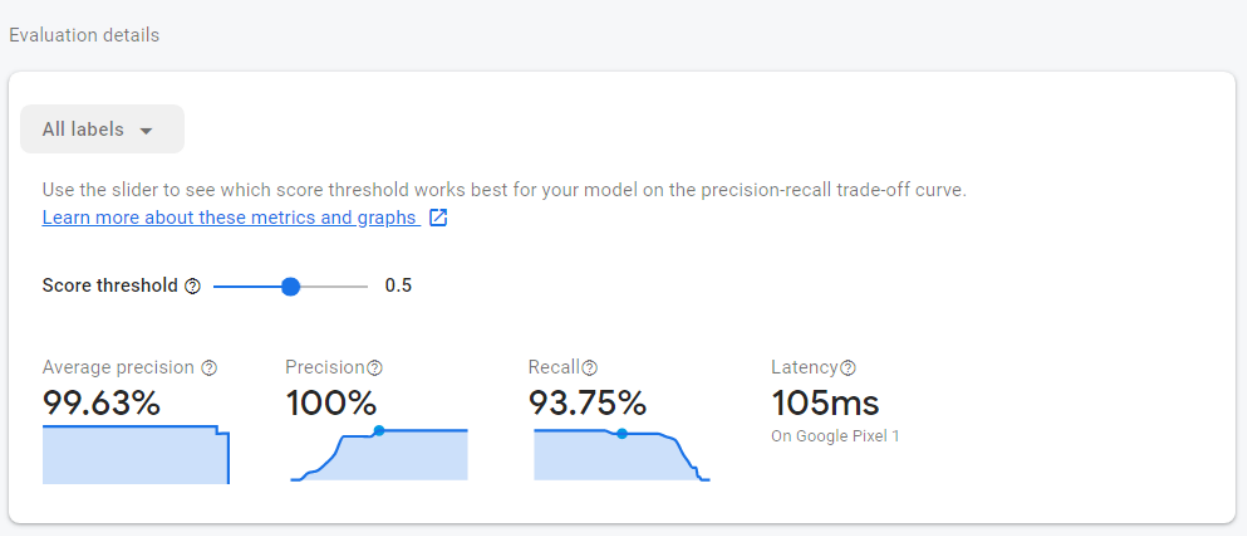
7.3. USE THE MODEL IN YOUR APP

Bundle the model with your app or let the ML Kit SDK download it from Firebase. Then, use the model to label images on the device.

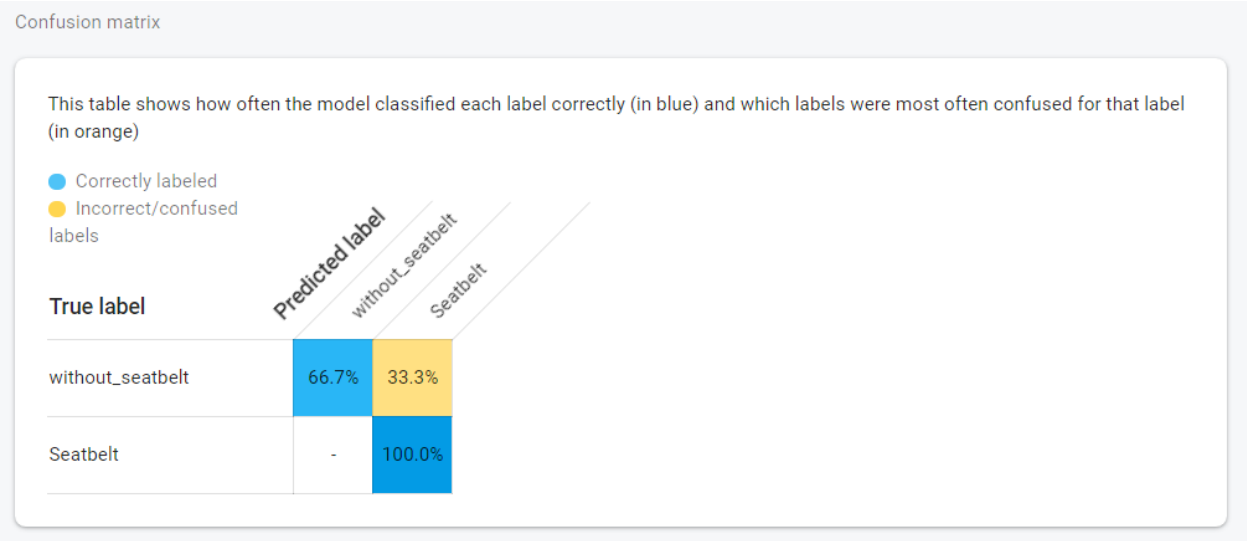
8. EVALUATION DETAILS

Evaluating the model for fairness requires thinking deeply about use case and what impact model could have on end users when it gets things wrong. This means understanding the impact of different types of errors for different user groups. This is where it's important to think about potential fairness issues. For example, do model errors affect all users equally, or are they more harmful to certain user groups?

Seatbelt detection model for traffic violation detection system was hosted on firebase. The model was trained and tested at Firebase server and here are the evaluation details for the model.



9. CONFUSION MATRIX



10. MODEL PRODUCTION

Implement simple fixes. If the model is not working perfect, retraining it with new data isn't the only answer. Sometimes a simple pre- or post-processing step to remove certain words or types of images can be an effective solution. Adjust the score thresholds of model to find an acceptably 'fair' balance between precision and recall, understanding of how different error types impact users.

Once the model is built and serving predictions, data distribution may change subtly over time, and it may no longer reflect the relevant contexts of application. Be sure to monitor model's performance over time to ensure that it's doing as well as you expect and collect feedback from end users to identify potential issues that may require new data and retraining.

11. OBJECT DETECTION AND FUTURE POSSIBILITIES

Object detection is breaking into a wide range of industries, with use cases ranging from personal security to productivity in the workplace. It is applied in many areas of computer vision, including image retrieval, security, surveillance, automated vehicle systems and machine inspection. Significant challenges stay on the field of object recognition. The possibilities are endless when it comes to future use cases for object detection.

REFERENCES

Ou, G., Gao, Y., Liu, Y. (2012). Real-Time Vehicular Traffic Violation Detection in Traffic Monitoring Stream. *Web Intelligence and Intelligent Agent Technology, 3, 15-19.*

Pietzsch, H.W., Opitz, R., Edelmann, R., Jaki, J. (Sep 1997). U.S. Patent No. 5673039. Ettlingen: U.Germany. Patent and Trademark Office.

Gevorgyan, M. [Menua Gevorgyan]. (2017, Dec 18). Traffic Violation Detection Demo [Video file]. Retrieved from https://www.youtube.com/watch?v=ovw_5ea0jmQ

Logipix. (n.d.). Retrieved from http://www.logipix.com/index.php/traffic-video-surveillance-system

Metrix Parking Systems. (n.d.). Retrieved from: https://www.metricgroup.co.uk/metric-parking-systems/

AIM Parking Management Software. (n.d.). Retrieved from: https://aimsparking.com/features-benefits/ticket-management

City of Las Vegas. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/city-las-vegas

Georgia Southern University. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/georgia-southern-university

City of Syracuse, NY. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/city-syracuse-ny-0

Washington State University. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/washington-state-university

Lower Merion, Pennsylvania. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/university-pennsylvania

University of Calgary. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/university-calgary

Town of Provincetown, Massachusetts. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/town-provincetown-massachusetts

Hanover Police Department, New Hampshire. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/hanover-police-department-new-hampshire

University of Waterloo. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/university-waterloo-0

University of Toronto – Scarborough. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/university-toronto-scarborough

WSU Vancouver. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/wsu-vancouver

University of Missouri, Kansas City. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/university-missouri-kansas-city

Brock University. *AIM Parking Management Software*. Retrieved from: https://aimsparking.com/success-stories/brock-university

Klubsuwan, K., Koodtalang, W., Mungsing, S. (December 2015,). Traffic Violation Detection Using Multiple Trajectories of Vehicles. Shruti Jawanjal Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, *5(12), 81-83*.

Xiaoling, W., Li-Min, M., Biaobiao, Z., Junjie, L., K.-L, D. (2013). A video-based traffic violation detection system. Proceedings - 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer, MEC 2013. 1191-1194. 10.1109/MEC.2013.6885246.