VEHICLE SEARCHING WITH DEEP LEARNING

A Main Project Report

submitted by

SURFA FARSEENA (MES20MCA-2054)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

July 2022

DECLARATION

I undersigned hereby declare that the project report **VEHICLE SEARCHING WITH DEEP LEARNING**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Ms.Febin Aziz, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: KUTTIPPURAM

Date: 08-07-2022

SURFA FARSEENA (MES20MCA-2054)

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **VEHICLE SEARCHING WITH DEEP LEARNING** is a bona fide record of the Main Project work carried out by **SURFA FARSEENA (MES20MCA-2054)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years.I wish to express my sincere thanks to my project coordinator Ms.Priya J.D Assistant Professor, Dept. of Computer Applications who guided me for the successful completion of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work.I would express my sincere thanks to my guide Ms.Febin Aziz ,Assistant Professor for her immense guidance to complete the project successfully.I would like to express my sincere thanks to all the staff members of Master of Computer Applications department for their support and valuable suggestion for doing the project work.Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

SURFA FARSEENA (MES20MCA-2054)

Abstract

Vehicle Searching System deals with detecting vehicle images from traffic surveillance videos.Nowadays traffic surveillance systems are installed in almost every city to record events and traffic.This surveillance videos can be used for vehicles searching with the help of Machine Learning. The system can be used by police officials to identify outlaw's vehicle in crime.Typically, police officials manually identify the vehicles in recorded video according to vehicle's appearances.This process is time-consuming and inclined to faults due to human fatigue for long duration videos.Moreover,hiring employees is costly.But with the help of Deep Learning it is possible to make this process more simpler.In this project, with the help of image processing and computer vision the vehicles are classified mainly based on two vehicle characteristics, i.e.types and colors.This project proposes a convolutional neural network framework to overcome the previously mentioned problems in vehicle searching in surveillance videos. CNN is a type of Deep Learning which is very well-known in image recognition field.This work mainly focuses on the performance of vehicle classification modules which are vehicle type classification and vehicle color classification.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Object detection and classification can be used for various purposes nowadays. Object detection is used for vehicle and pedestrian detection, traffic sign and lane detection, vehicle make detection etc.Ability to detect and classify traffic related objects makes it possible to improve the state of traffic flow, prevent serious traffic accidents and even register crimes, such as stolen vehicles or speeding.This is very important since the number of vehicle is constantly rising.For humans it is easy to recognize vehicles in images or videos or distinguish between different vehicle types. The goal of this project is to develop a system that searches for vehicle with the help of convolutional neural network (CNN) to perform vehicle detection and classification on traffic surveillance videos.In the case of criminal investigation processes it is extremely important to automatically search for suspicious vehicles.Surveillance cameras are installed in most of the places in cities. The main purpose of installing surveillance cameras are real-time monitoring and events searching.This project focuses only on event searching. For searching objective, the surveillance systems are used by police officials eg.to search for specific vehicle. Generally the police officials require the information of vehicle's characteristics, e.g.vehicle's color, vehicle's type as a clue for identifying the vehicle.

They often spend a lot of time monitoring traffic camera footage by themselves. Usually,searching time can be more than the video duration and they have to repeat the searching again several times.In addition, they might make mistakes with the weariness after a long period of searching.This project proposes a convolutional neural network framework to solve these problems in vehicle searching in surveillance videos.Efficacy of CNNs in image recognition is the main reasons why the world recognizes the power of deep learning. A dataset is created that contains images to enable recognition and classification of vehicles.The purpose of this proposed system is to reduce human effort in vehicle searching using deep learning.

1.1.1 Motivation

Vehicle searching with deep learning deals with searching vehicle from traffic surveillance videos.It consists of a CNN structures which are used as classifiers in both vehicle type classification and vehicle color classification. This method requires only one input which is a vehicle image fed into the system. The system extract the features of this vehicle image and perform classification on the extracted features. Finally the system detects vehicle from the video based on user input.The goal of this project is to develop a system that searches for vehicle with the help of convolutional neural network (CNN) to perform vehicle detection and classification on traffic surveillance videos. In the case of criminal investigation processes it is extremely important to automatically search for suspicious vehicles. Surveillance cameras are installed in most of the places in cities. The main purpose of installing surveillance cameras are real-time monitoring and events searching. This project focuses only on event searching. For searching objective, the surveillance systems are used by police officials.

1.2 Objective

To develop a system that searches for vehicle with the help of convolutional neural network (CNN) to perform vehicle detection and classification on traffic surveillance videos.To improve the accuracy of vehicle type and vehicle color classification which are previously mentioned. The classifier in this work is selected to be convolutional neural network with two convolution layers. The CNN is chosen because of its performance in image recognition.

1.3 Report Organization

The project report is divided into four sections.Section1, Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

Zhou et al. proposed deep neural network or deep learning approaches for vehicle detection and vehicle classification. In detection, they used YOLO as a detection model. Alexnet was used as classification approaches. In classification modules, there are four kinds of classification, i.e. passenger vs other, cars vs vans, sedans vs taxis, and sedans vs vans vs taxis. After applying both structures, they were fine-tuned to be suitable with the public dataset which provided . The experimental results showed the accuracy of more than 90 percentage.

Chapter 3

Methodology

3.1 Introduction

Object detection and classification can be used for various purposes nowadays.Object detection is used for vehicle and pedestrian detection, traffic sign and lane detection, vehicle make detection etc.Ability to detect and classify traffic related objects makes it possible to improve the state of traffic flow, prevent serious traffic accidents and even register crimes, such as stolen vehicles or speeding.This is very important since the number of vehicle is constantly rising.For humans it is easy to recognize vehicles in images or videos or distinguish between different vehicle types. The goal of this project is to develop a system that searches for vehicle with the help of convolutional neural network (CNN) to perform vehicle detection and classification on traffic surveillance videos.In the case of criminal investigation processes it is extremely important to automatically search for suspicious vehicles.Surveillance cameras are installed in most of the places in cities. The main purpose of installing surveillance cameras are real-time monitoring and events searching.This project focuses only on event searching. For searching objective, the surveillance systems are used by police officials eg.to search for specific vehicle.

Generally the police officials require the information of vehicle's characteristics, e.g.vehicle's color, vehicle's type as a clue for identifying the vehicle. They often spend a lot of time monitoring traffic camera footage by themselves.Usually,searching time can be more than the video duration and they have to repeat the searching again several times.In addition, they might make mistakes with the weariness after a long period of searching.This project proposes a convolutional neural network framework to solve these problems in vehicle searching in surveillance videos.Efficacy of CNNs in image recognition is the main reasons why the world recognizes the power of deep learning. A dataset is created that contains images to enable recognition and classification of vehicles.The purpose of this proposed system is to reduce human effort in vehicle searching using deep learning.

3.2 Modules

The project is divided into 2 functional modules.They are,

1. Admin

• Login

• Add or manage police

• Add or manage Camera

• Assign work

• Video upload

2.Traffic Police

• Login

• Works view

• Works status update

• View camera

• View video

• Searching

3.3 Developing Environment

• OPERATING SYSTEM: WINDOWS 10

• FRONT END: HTML, CSS, JAVASCRIPT

• BACK END: Mysql

• IDE USED: Jetbrains Pycharm, Android studio

• TECHNOLOGY USED: PYTHON JAVA

• FRAME WORK USED: Flask

3.4 Work Flow

Convolutional neural networks are the most popular neural network in deep learning. The main characteristic of these networks is convolution, which is designed to learn higher features in the data. The networks are well suited to object recognition with images and consistently

top classifier in image classification competitions.The classifier in this work is selected to be convolutional neural network with two convolution layers. The CNN is chosen because of its performance in image recognition.This work consists of a CNN structures which are used as classifiers in both vehicle type classification and vehicle color classification.In type classification, three classes are categorized, i.e. small, medium, large. There are six classes in color classification, i.e. black, white, blue, green, yellow, red. The proposed method requires only one input which is a vehicle image fed into the system.

3.5 Feature Extraction

In the proposed system traffic surveillance video is given as the input. In the feature extraction module, it crops the vehicle images from the video systematically. After cropping the video into images the module extracts the features of these images.

3.6 Classification

The output of the feature extraction module is given to the classification module. In this module the classification is done based on two classifications. They are vehicle type and color. In the type classification, four classes are categorized. They are small, medium, large and unknown. In the case of color classification there are seven classes. They are black, white, blue, green, yellow, red, and unknown. Both classifications consist of unknown class. This class contains the vehicle with ambiguous characteristics and irrelevant colors. Overlapped vehicles and brown color are came under this category. The classifier in this work is convolutional neural network with two convolution layers. The vehicle images of dataset created is fed into CNN structure. Firstly, the original vehicle image is resized. The resized image is fed to the first convolutional layer. The output of the first convolution layer is modeled by the activation function ReLU. Then the output is passed to the second convolutional layer and the same operations are repeated. After that, the output of second convolution layer is converted into vector form and then fed into fully connected layer. The final layer is a predictor.

3.7 Search Manager

After the classification process the vehicle images are stored into a model file. This model file is given to the search manager module to predict the output. The search manager module will stores and filters the results according to the given query commands. When a police official performs a search the search manager module loads the model file and input data together and compare them. The search manager returns the result to the police officials according to this comaparison through a user interface.

3.8 Agile Methodology

This project was developed using Agile Development Model. The entire project was divided into four sprints. In the first sprint, the characters for the password was developed. The designing of front-end and development of back-end was done in the second , third and fourth sprint respectively.

3.8.1 User story

User StoryID	As a <type of user>	I want to	So that I can
1	Admin	Login	Login successful with correct username and password
2	Admin	Add and manage police	Add and manage police information
3	Admin	Manage camera	Add and manage camera information on different location
4	Admin	Assign work	Assign to work police
5	Admin	Video upload	Upload video based on camera
6	Traffic police	Login	Login successful with correct username and password
7	Traffic police	Works view and status update	View allocated works from admin and updated works status
8	Traffic police	View camera	View camera information
9	Traffic police	View Videos	view videos
10	Traffic police	Searching	Searching vehicle from videos

Table 3.1: User Story

3.8.2 Product Backlog

User Story ID	Priority <High/ Medium/ Low>	Size (Hours)	Sprint <#>	Status <Planned/ In progress/ Completed	Release Date	Release Goal
1	Medium	8	1	Completed	01/05/2022	Table design
2	High	10		Completed	12/05/2022	Form design
3	Medium	6		Completed	20/05/2022	Basic coding
4	High	5	2	Completed	25/05/2022	Login successful with correct username and password
5	High	3		Completed	29/05/2022	Add and manage police information
6	High	2	3	Completed	02/05/2022	Dataset training
7	High	11		Completed	05/06/2022	Camera application and frame capture
8	Medium	4	4	Completed	20/06/2022	View camera information
9	Medium	4		Completed	24/06/2022	View videos
10	High	7		Completed	29/06/2022	Search vehicle from Videos
11	High	3	5	Completed	01/07/2022	Testing & Output generation

Table 3.2: Product Backlog

3.8.3 Project plan

User StoryID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	20/04/2022	01/05/2022	3	Completed
2		04/05/2022	12/05/2022	7	Completed
3		15/05/2022	20/05/2022	2	Completed
4	Sprint 2	21/05/2022	25/05/2022	3	Completed
5		26/05/2022	29/05/2022	3	Completed
6	Sprint 3	30/05/2022	02/06/2022	7	Completed
7		03/05/2022	05/06/2022	7	Completed
8	Sprint 4	06/06/2022	20/06/2022	2	Completed
9		21/06/2022	24/06/2022	3	Completed
10		25/06/2022	29/06/2022	7	Completed
11	Sprint 5	30/06/2022	01/07/2022	2	Completed

Table 3.3: Project Plan

3.8.4 Sprint plan

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	1	1	0	1	0	0	0	0	0	0
Form design	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Basic coding	20/05/2022	6	1	0	0	0	0	0	0	0	0	1	1	1	1	0
UserStory #4,#5																
Login	25/05/2022	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Add and manage police information	29/05/2022	5	1	1	1	0	0	1	0	0	0	0	0	0	0	1
UserStory #6,#7																
Dataset training	02/06/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Camera application and frame capture	05/06/2022	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0
UserStory #8,#9,#10																
View camera information	20/06/2022	4	0	0	0	0	0	0	0	0	0	0	1	1	1	1
View Videos	29/06/2022	6	0	0	0	0	0	0	0	1	0	0	1	1	1	1
Search vehicle from Videos	1/07/2022	4	0	0	0	0	1	1	1	1	0	0	0	0	0	0
UserStory #11																
Testing & Output generation	4/07/2022	3	0	0	0	0	0	0	0	0	0	0	1	1	1	1
Total		51	7	5	2	3	4	3	2	3	1	2	5	4	4	4

Table 3.4: Sprint Plan

3.8.5 Sprint Actuals

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	0	1	1	0	0	0	0	0	0	0
Form design	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Basic coding	20/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Login																
Add and manage police information																
UserStory #6,#7																
Dataset training																
Camera application and frame capture																
UserStory #8,#9,#10																
View camera information																
View Videos																
Search vehicle from videos																
UserStory #11																
Testing & Output generation																
Total		22	3	2	1	3	2	1	2	1	1	1	2	1	1	0

Table 3.5: Sprint Actua11

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	1	1	0	1	0	0	0	0	0	0
Form design	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Basic coding	20/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Login	25/05/20	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Add and manage police information	29/05/2022	5	1	1	1	0	0	1	0	0	0	0	0	0	0	1
UserStory #6,#7																
Dataset training																
Camera application and frame capture																
UserStory #8,#9,#10																
View camera information																
View Videos																
Search vehicle from videos																
UserStory #11																
Testing & Output generation																
Total		29	5	4	2	3	3	2	1	2	1	1	2	1	1	1

Table 3.6: Sprint Actual2

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	1	1	0	1	0	0	0	0	0	0
Form design	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Basic coding	20/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Login	25/05/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Add and manage police information	29/05/2022	5	1	1	1	0	0	1	0	0	0	0	0	0	0	1
UserStory #6,#7																
Dataset training	02/06/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Camera application and frame capture	05/06/2022	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0
UserStory #8,#9,#10																
View camera information																
View Videos																
Search vehicle from videos																
UserStory #11																
Testing & Output generation																
Total		34	7	6	2	3	3	2	1	2	1	1	2	1	1	1

Table 3.7: Sprint Actual 3

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	1	1	0	1	0	0	0	0	0	0
Form design	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Basic coding	20/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Login	25/05/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Add and manage police information	29/05/2022	5	1	1	1	0	0	1	0	0	0	0	0	0	0	1
UserStory #6,#7																
Dataset training	02/06/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Camera application and frame capture	05/06/2022	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0
UserStory #8,#9,#10																
View camera information	20/06/2022	4	0	0	0	0	0	0	0	0	0	0	1	1	1	1
View Videos	24/06/2022	6	0	0	0	0	0	0	0	0	0	0	1	1	1	1
Search vehicle from videos	29/06/2022	4	0	0	0	0	1	1	1	1	0	0	0	0	0	0
UserStory #11																
Testing & Output generation																
Total		48	7	6	2	3	4	3	3	3	1	1	4	3	3	3

Table 3.8: Sprint Actual4

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2,#3																
Table design	01/05/2022	8	1	1	1	2	1	1	0	1	0	0	0	0	0	0
Form design	12/05/2022	8	1	1	0	1	2	0	1	1	0	0	1	0	0	0
Basic coding	20/05/2022	6	1	0	0	0	0	0	0	0	1	1	1	1	1	0
UserStory #4,#5																
Login	25/05/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Add and manage police information	29/05/2022	5	1	1	1	0	0	1	0	0	0	0	0	0	0	1
UserStory #6,#7																
Dataset training	02/06/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Camera application and frame capture	05/06/2022	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0
UserStory #8,#9,#10																
View camera information	20/06/2022	4	0	0	0	0	0	0	0	0	0	0	1	1	1	1
View Videos	24/06/2022	6	0	0	0	0	0	0	0	0	0	0	1	1	1	1
Search vehicle from videos	29/06/2022	4	0	0	0	0	1	1	1	1	0	0	0	0	0	0
UserStory #11																
Testing & Output generation	01/07/2022	3	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Total		51	7	5	2	3	4	3	2	3	1	2	5	4	4	4

Table 3.9: Sprint Actual 5

Chapter 4

Results and Discussions

4.1 Results

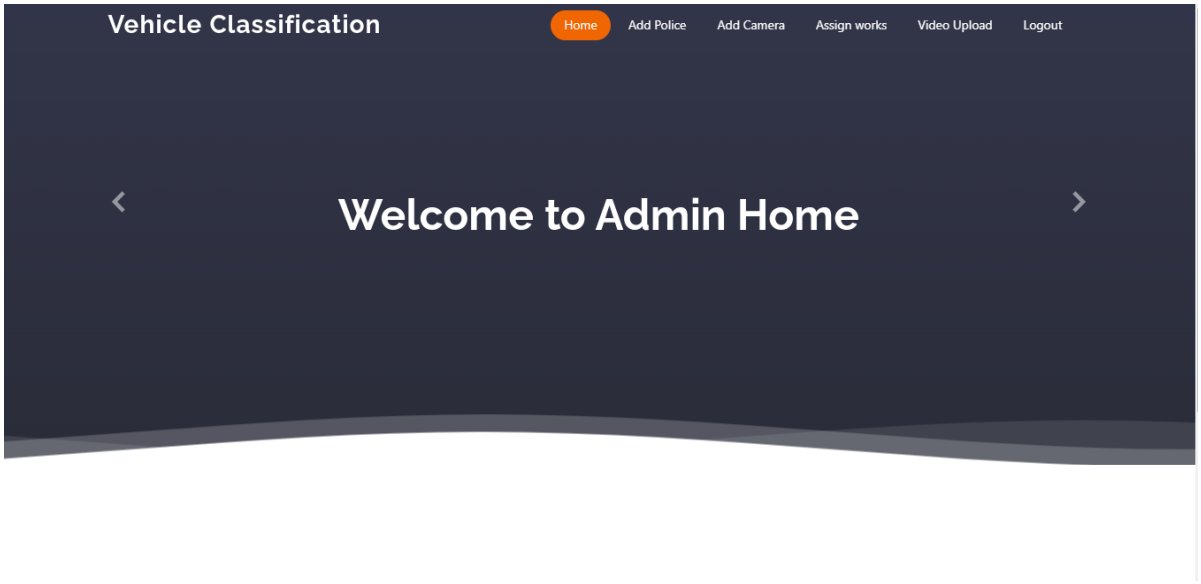

Figure 4.1: User Interface 1

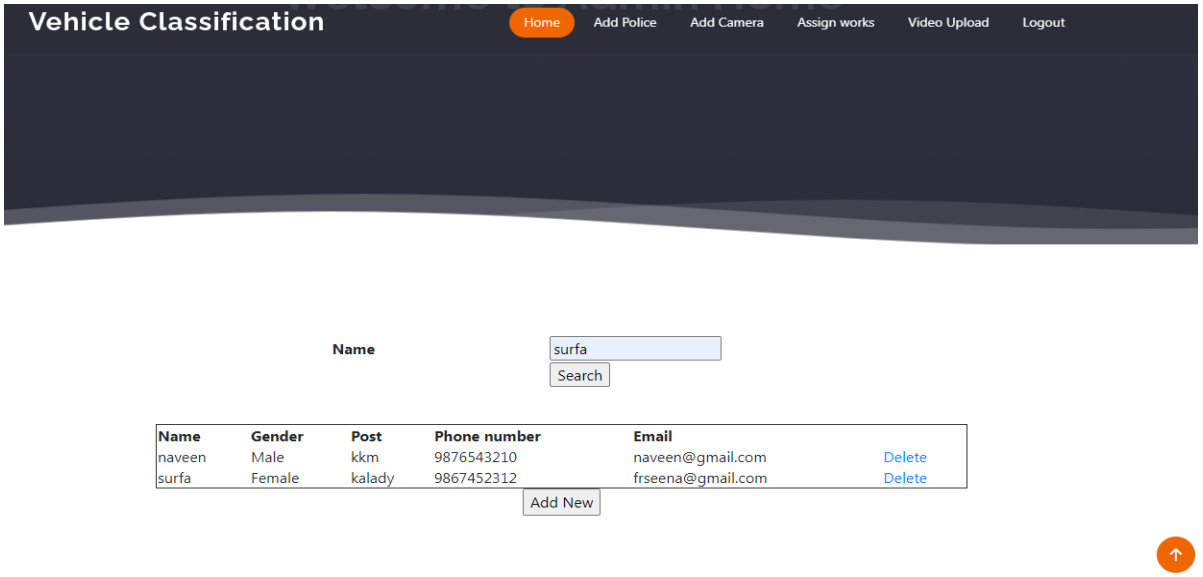

Figure 4.2: User Interface 2

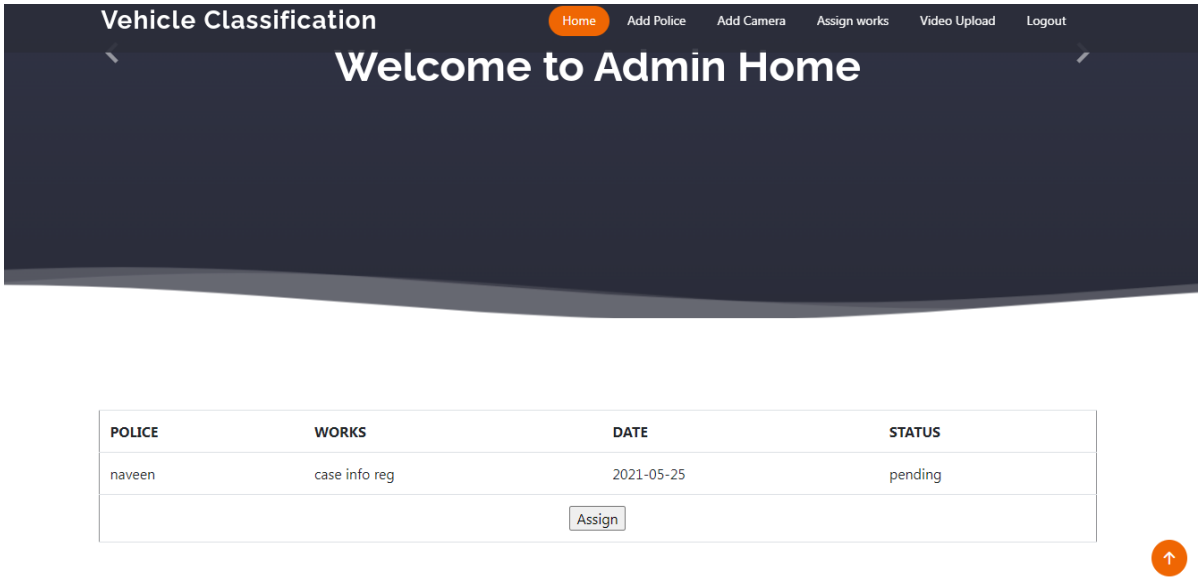

Figure 4.3: User Interface 3

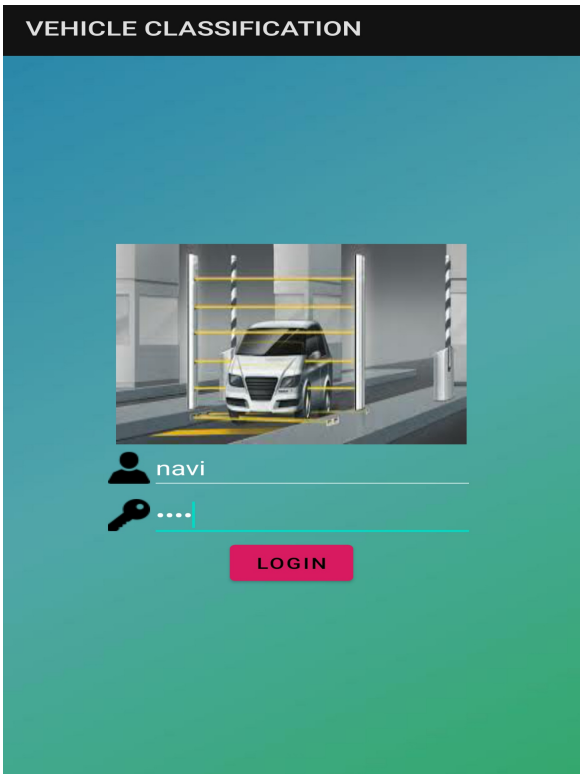

Figure 4.4: User Interface 4

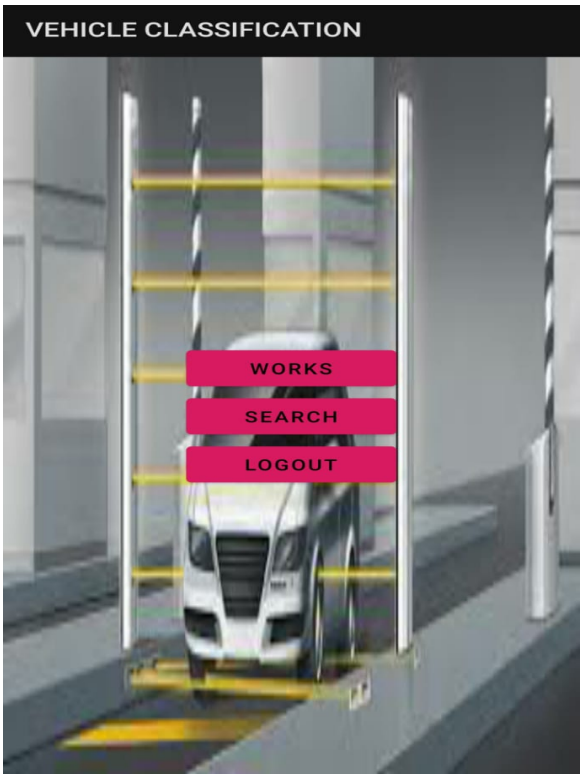

Figure 4.5: User Interface 5

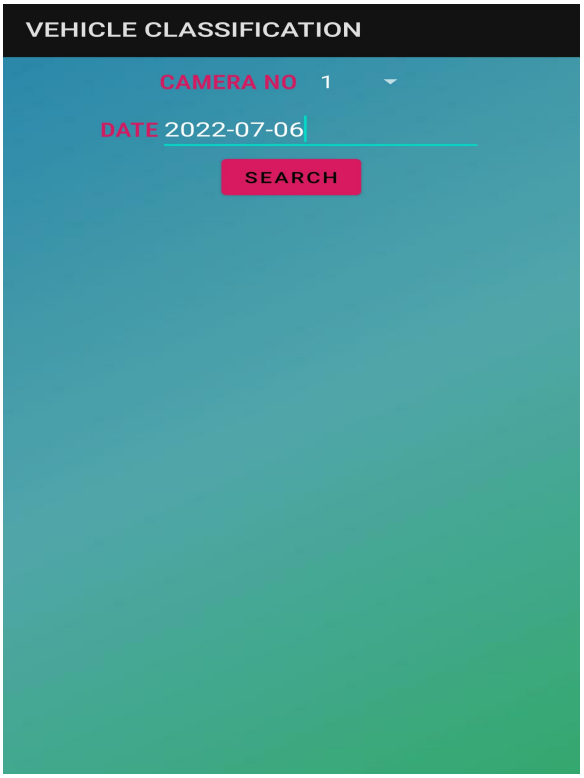

Figure 4.6: User Interface 6

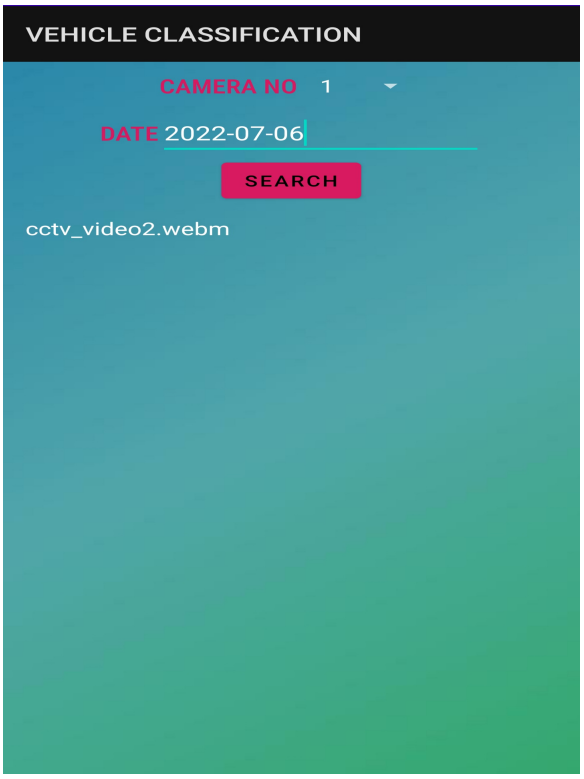

Figure 4.7: User Interface 7

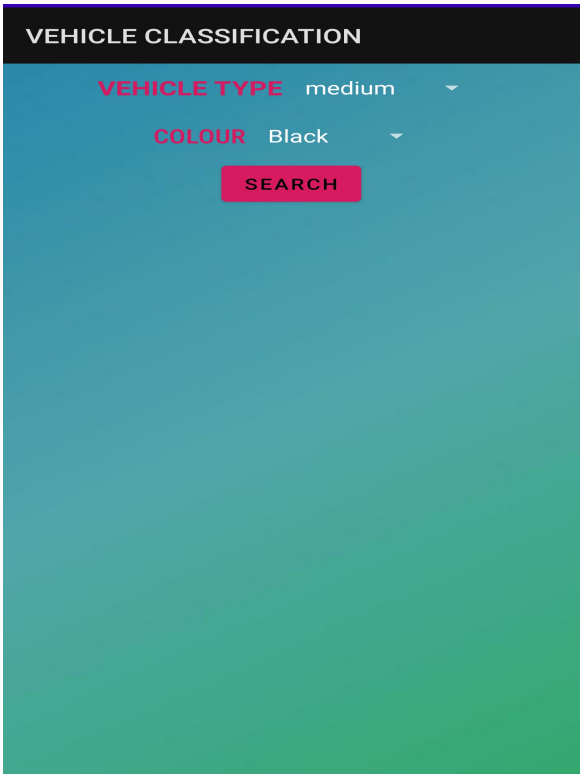

Figure 4.8: User Interface 8

Figure 4.9: User Interface 9

4.2 Discussions

CNN is proposed as a type and color classifiers to classify vehicle characteristics from vehicle image which systematically cropped by machine. The experiment's results show that CNN outperforms other methods in type classification. However, CNN improves a little accuracy in color classification. For the future work, improving the accuracy of color classification will be the major goal. Furthermore, many aspects should be explored and experimented, e.g. different input image size also, deeper CNN structure such as Alexnet structure which many researches applied and fine-tuned the structure and different hyper parameters

4.3 Datasets

In this project the dataset is created by using various vehicle images. The dataset contain a total of 2587 vehicle images. These images are categorized into three subsets. The first set is the images of the heavy vehicles. The second set is the images of vehicles that came under medium category. The third set contains the images of small vehicle.

Chapter 5

Conclusions

Vehicle searching system proposes a model to detect vehicle images from traffic surveillance videos. Vehicle classification is one of the important domains on traffic systems. In a variety of applications, it has become important. Typically police officers search for vehicle in traffic camera footage by checking the entire video multiple times. This is a tiresome and time consuming work. The proposed system is implemented to help the police officials to search for a vehicle much faster. In this the police don't need to watch the entire video to check a vehicle is in the video. This system is useful in the case of crimes to detect the suspect's vehicle. In this project CNN is proposed as a type and color classifiers to classify vehicle characteristics from vehicle image which systematically cropped by machine. The experiment's results show that CNN outperforms other methods in type classification.

References

[1] **P.O. Gislason, J.A. Benediktsson, and J.R. Sveinsson, "Random forests for land cover classification, "Pattern Recognition Letters, vol. 27, no. 4, pp. 294300, Mar. 2006**

[2] **K. Saripan, C. Deachakul, and W. Moungmai, "Smart vehicle search in surveillanc e video," 2015**

[3] **Watcharin Maungmai, Chaiwat Nuthong, "Vehicle Classification with Deep Learning", IEEE 4th International Conference on Computer and Communication Systems (ICCCS) (2019).**

[4] **P.M. Daigavane, Preeti R. Bajaj, M.B. Daigavane, "Vehicle Detection and Neural Network Application for Vehicle Classification", International Conference on Computational Intelligence and Communication Net- works(2011).**

Appendix

Source Code

```
   # This piece of software is bound by The MIT License (MIT)
   # Copyright (c) 2013 Siddharth Agrawal
   # Code written by : Siddharth Agrawal
   # Email ID : siddharth.950@gmail.com
from flask import *
import os
from werkzeug.utils import secure_filename
from src.dbop import iud, selectall, selectone,selectalls,iudd
path="static//videos"
app = Flask(__name__)
import cv2

@app.route('/')
def main():

   return render_template('login.html')

@app.route('/login', methods=['post'])
def login():
   un = request.form['username']
   passwd = request.form['passwd']
   qry = "SELECT * FROM `login` WHERE `username`='" + un + "' AND `password`='" + passwd + "' and type='admin'"
   res = selectone(qry)
   if res is None:
      return ''' <script>alert('Invalid username or password');window.location='/'</script> '''
   else:
      return ''' <script>window.location='/adminhome'</script> '''


@app.route('/adminhome')
def adminhome():
   return render_template('Admin home.html')

@app.route('/addpolice')
def addpolice():
   qry = "SELECT * FROM `police`"
   res = selectall(qry)
   return render_template('Add police.html', val=res)


@app.route('/addcam')
def addcam():
   qry = "SELECT * FROM `camera`"
   value = selectall(qry)
```

```
    return render_template('Cam details.html', val=value)

@app.route('/deletecam')
def deletecam():
    id=request.args.get('id')
    qry = "DELETE FROM `camera` WHERE cid="+str(id)
    iudd(qry)

    return ''' <script>window.location='/addcam'</script> '''



@app.route('/deletepolice')
def deletepolice():
    id=request.args.get('id')
    qry = "DELETE FROM `login` WHERE id="+str(id)
    iudd(qry)
    qry="DELETE FROM `police` WHERE l_id="+str(id)
    iudd(qry)

    return ''' <script>window.location='/addpolice'</script> '''


@app.route('/policeregistration',methods=['post'])
def policeregistration():
    return render_template('Police Registration.html')

@app.route('/camregistration',methods=['get','post'])
def camregistration():
    return render_template('Cam Registration.html')



@app.route('/camreg', methods=['post'])
def camreg():
    Camerno=request.form['textfield']
    Landmark = request.form['textfield2']
    Place = request.form['textfield3']
    qry="insert into camera values(null,%s,%s,%s)"
    val=Camerno,Landmark,Place
    iud(qry,val)
    return '''<script>alert("registered");window.location="/addcam"</script>'''



@app.route('/policereg', methods=['post'])
def policereg():
    name=request.form['textfield']
    dob = request.form['textfield2']
    gender = request.form['radiobutton']
    place = request.form['textfield4']
    pin = request.form['textfield5']
    post = request.form['textfield6']
    email = request.form['textfield7']
    phno = request.form['textfield8']
    opost = request.form['textfield9']
    uname = request.form['textfield10']
    password = request.form['textfield11']


    qry="INSERT INTO `login` VALUES(NULL,%s,%s,'police')"
    val=uname,password
    id=iud(qry,val)
```

```
    qry="INSERT INTO `police` VALUES(NULL,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
    val=str(id),name,dob,gender,place,pin,post,email,phno,opost
    print (val)
    iud(qry,val)
    return '''<script>alert("registered");window.location="/addpolice"</script>'''



@app.route('/videoup')
def videoup():
    qry="SELECT * FROM camera"
    res=selectall(qry)
    return render_template('videoup.html',res=res)

@app.route('/videoup1', methods=['post'])
def videoup1():
    cid=request.form['cam']
    file=request.files['file']
    var=secure_filename(file.filename)
    file.save(os.path.join(path,var))
    qry="insert into videoup values(null,%s,%s,curdate(),curtime())"
    val=cid,var
    vid=iud(qry,val)

    cam = cv2.VideoCapture(os.path.join(path, var))

    currentframe = 0

    while (True):

        ret, frame = cam.read()

        if ret:
            if currentframe % 15 == 0:
                n = 'frame' + str(vid) + str(currentframe) + '.jpg';
                name = 'static/frame/frame' + str(vid) + str(currentframe) + '.jpg'

                print('creating...' + name)
                qry="INSERT INTO frame VALUES(NULL,%s,%s,'pending')"
                val = vid, n
                iud(qry,val)

                cv2.imwrite(name, frame)
            currentframe += 1

        else:
            break

    cam.release()
    cv2.destroyAllWindows()
    return '''<script>alert("uploaded successfully");window.location="/adminhome"</script>'''
@app.route('/searchpolice', methods=['post'])
def searchpolice():
    name=request.form['textfield']
    qry = "SELECT * FROM `police` where name=%s"
    res =selectalls(qry,name)
    return render_template('Add police.html', val=res)

@app.route('/viewassign')
def viewassign():
    qry="SELECT `police`.`Name`,`work`.`work`,`work`.`date`,`work`.`status` FROM `work` JOIN `police` ON
        `police`.`l_id`=`work`.uid"
    s=selectall(qry)
```

```
    return render_template('assignedwrks.html', val=s)
@app.route('/assign', methods=['post'])
def assign():
    qry="SELECT * FROM `police`"
    s=selectall(qry)
    return render_template('assign.html',v=s)
@app.route('/assign1', methods=['post'])
def assign1():
    pid=request.form['slt']
    wrk=request.form['textarea']
    qry="insert into work values(null,%s,%s,curdate(),'pending')"
    value=(wrk,str(pid))
    iud(qry,value)
    return '''<script>alert(" success");window.location="/viewassign"</script>'''

app.run(debug=True)
```

Database Design

Attribute Name	Datatype	Length	Description
id	Integer	11	Primary Key
username	Varchar	50	
password	Varchar	50	
type	Varchar	50	

Table A.1: Login

Attribute Name	Datatype	Length	Description
p-id	Integer	11	Primary Key
l-id	integer	11	
name	varchar	50	
dob	date		
gender	varchar	50	
place	varchar	100	
post	varchar	50	
pin	Integer	10	
email	varchar	50	
phone-no	bigint	11	
officerpost	varchar	50	

Table A.2: Police

Attribute Name	Datatype	Length	Description
c-id	Integer	11	Primary Key
cam-no	Integer	11	
landmark	Varchar	50	
place	varchar	100	

Table A.3: Camera

Attribute Name	Datatype	Length	Description
id	Integer	11	Primary Key
v-id	Integer	11	
frame	Varchar	50	
status	text		

Table A.4: Frame

Attribute Name	Datatype	Length	Description
t-id	Integer	11	Primary Key
f-id	Integer	11	
tag	Varchar	20	
color	Varchar	20	

Table A.5: Frame-tag

Attribute Name	Datatype	Length	Description
v-id	Integer	50	Primary Key
cam-id	Integer	50	
video	Varchar	100	
date	Varchar	50	
time	varchar	50	

Table A.6: Video-upload

Attribute Name	Datatype	Length	Description
w-id	Integer	11	Primary Key
work	text		
u-id	Integer	11	
date	Varchar	90	
status	Varchar	100	

Table A.7: Work

Dataflow Diagram

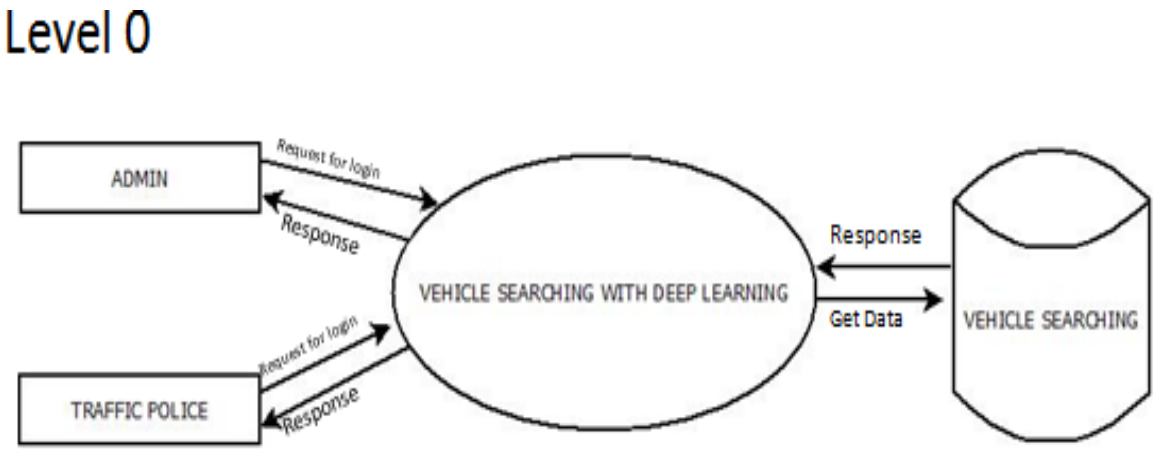

Figure A.1: Dataflow Diagram

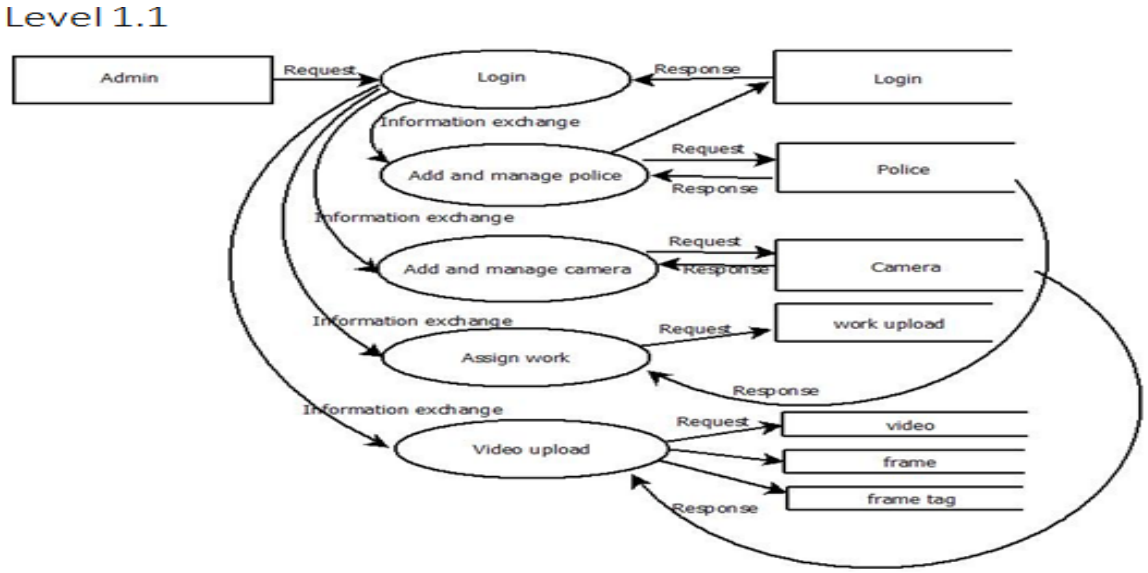

Figure A.2: Dataflow Diagram

Level 1.2

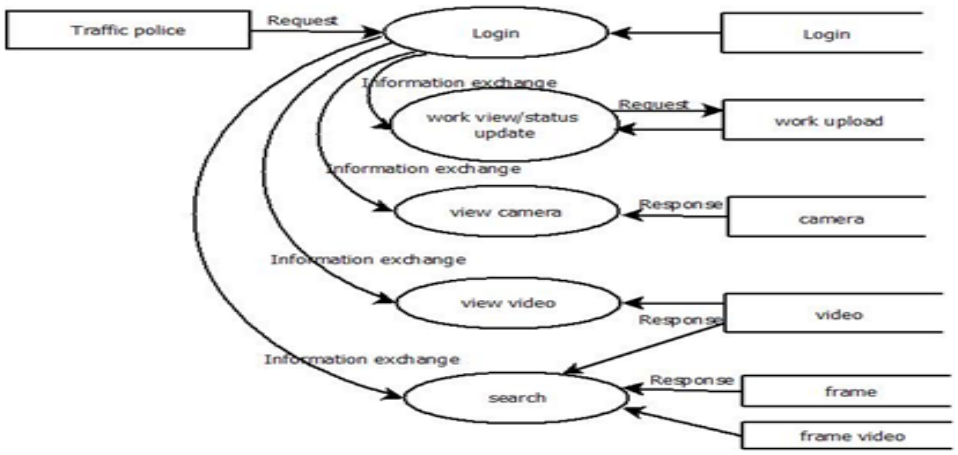

Figure A.3: Dataflow Diagram