CopVision

An automatic e-challan generation system

CPG No:- 34

Under the mentorship of:
DR ROHIT AHUJA

NIPUNN MALHOTRA	101703374
NISHANT GOEL	101703376
PARAS ARORA	101703382

Thapar Institute
OF ENGINEERING & TECHNOLOGY
(Deemed to be university)

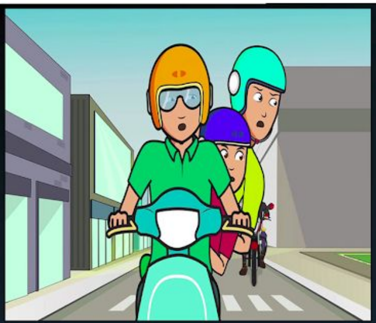
CPG 34

CopVision

CopVision

Under the mentorship of
DR. ROHIT AHUJA

Nipunn Malhotra (101703374)
Nishant Goel (101703376)
Paras Arora (101703382)



1

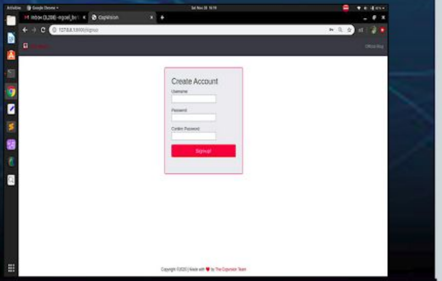
WHY THIS PROJECT?

The project is an entirely new concept. There is no software out in the market which can check tripling as well as if a person is not wearing helmet and generate challans.

SOFTWARE DESIGN

The web server gives the admin permissions to upload the video. Any Employee of CopVision can approve or reject a challan. Also, a general user can view his/her challan by just going to the interface.As its aim, the website also hasdifferent webpages motivating people to follow rules.

2



Create Account

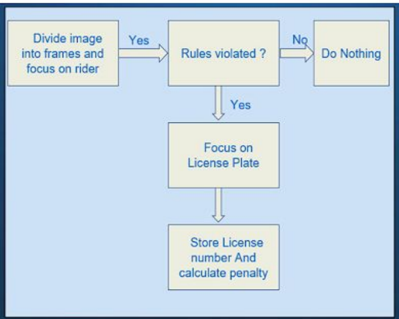
Username

Password

Confirm Password

Signup!

Copyright ©2020 | Made with ♥ by The Copvision Team



Divide image into frames and focus on rider

Yes

Rules violated ?

No

Do Nothing

Yes

Focus on License Plate

Store License number And calculate penalty

3

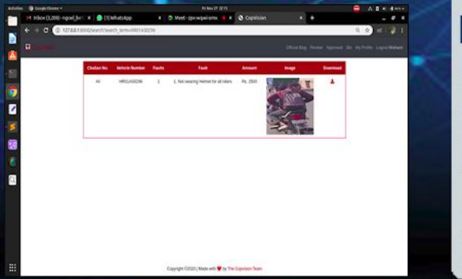
RULE VIOLATION DETECTION

The rule violation of the uploaded video is done easily by applying ML algorithms such as CNN,OCR and using models like darknet and algorithms for improving image quality.

USER INTERFACE

It is an easy to use, user friendly interface. All the utilities in the interface are self explanatory. With just few clicks, any process can be done with high speed. At the same time, user gets a reliable experience with a cool look of the interface.

4



Challan No.	Vehicle Number	Faults	Fault	Amount	Image	Download
44	HR01A30296	1	1. Not wearing Helmet for all riders	Rs. 2500		

Copyright ©2020 | Made with ♥ by The Copvision Team

5

FUTURE SCOPE

In future we aim to increase the efficiency of the results to a new level. At the same time, we also plan to add new utilities such as overspeed detection while also improving the present utilities.

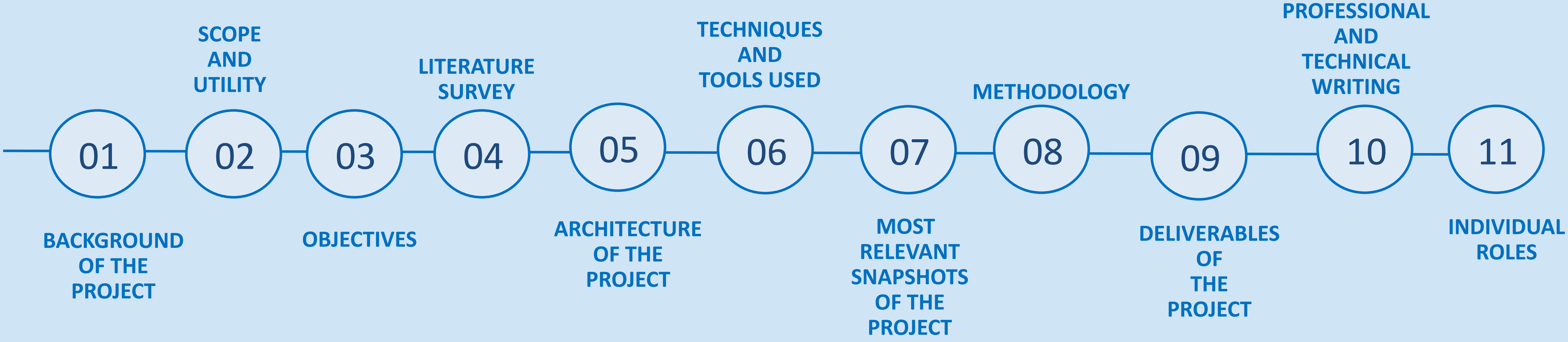
TECHNOLOGY USED:

OpenCV

django

COE FOURTH YEAR 2020
Computer Engineering Dept.

TABLE OF CONTENTS



01 BACKGROUND OF THE PROJECT

02 SCOPE AND UTILITY

03 OBJECTIVES

04 LITERATURE SURVEY

05 ARCHITECTURE OF THE PROJECT

06 TECHNIQUES AND TOOLS USED

07 MOST RELEVANT SNAPSHOTS OF THE PROJECT

08 METHODOLOGY

09 DELIVERABLES OF THE PROJECT

10 PROFESSIONAL AND TECHNICAL WRITING

11 INDIVIDUAL ROLES

01

BACKGROUND OF THE PROJECT

Project Overview

- Safety is the utmost concern in today’s world of growing technologies and commutation.
- Two-wheeler is a popular mode of transportation in India.
- There is a high risk involved due to **lesser protection.**
- To reduce the risk involved, it is mandatory for bike-riders to :-
  - **Use a helmet while driving,**
  - **Follow speed limits,**
  - **Avoid tripling**
- People often ride two-wheelers without wearing helmets and go beyond the speed limits because of their **carefree attitude.**
- It is a tedious task for cops to catch and fine 2-wheeler drivers violating the traffic rules due to :
  - High traffic
  - High speed of riders
  - casual behaviour of cops

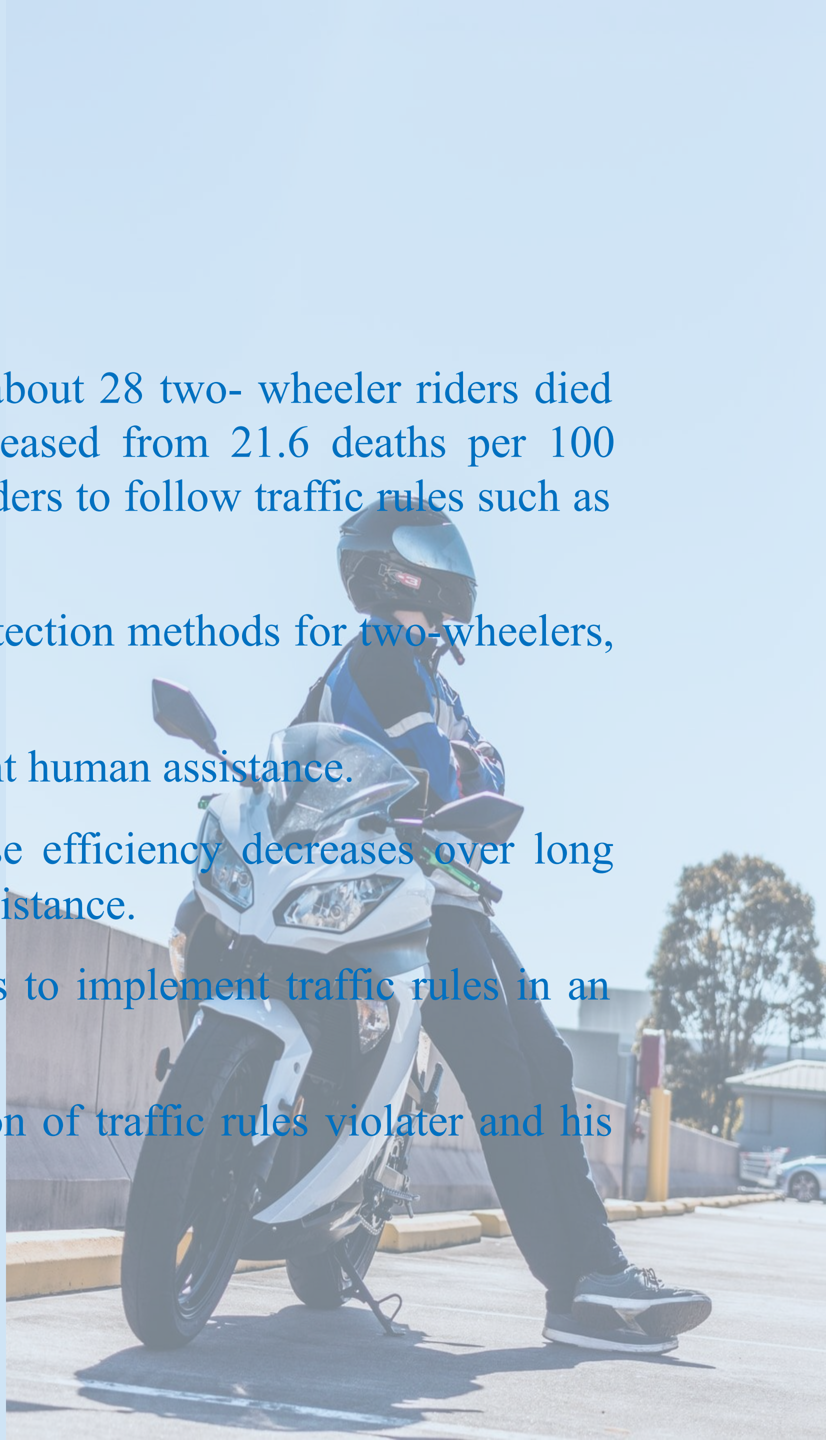
- The Software returns the penalty of a rider if he will not commit a mistake again during the same year so that it encourages the rider to follow the rules in future.
- The software increases the penalty amount if the rider violates the rules again and again thus ensuring strictness.
- If even on increasing penalty person continues to violate traffic rules , the license of the rider will be confiscated and will be returned after giving driving test again.
- The software does not discriminate between and VIP or politician and a normal person and gives the same penalty for all of them.

02

SCOPE AND UTILITY

Scope And Utility

- According to an analysis of data shared by states with the transport ministry, about 28 two- wheeler riders died daily on Indian roads in 2016 for not wearing helmets. It has steadily increased from 21.6 deaths per 100 accidents in 2005 to 29.1 in 2015. To reduce this rate, it is important for bike riders to follow traffic rules such as wearing a helmet, following speed limits and avoiding tripling.
- Not much work is done on the video surveillance based traffic rule violation detection methods for two-wheelers, existing projects focus only on four wheelers.
- The existing video surveillance based methods are passive and require significant human assistance.
- In general, such systems are infeasible due to involvement of humans, whose efficiency decreases over long duration. It is also very expensive and difficult to provide 24 hours of human assistance.
- "CopVision" automates the whole process, decreasing human effort and helps to implement traffic rules in an effective way.
- This project also helps the traffic police by making their job easier of detection of traffic rules violater and his number plate or RC details.



03

OBJECTIVES

PROBLEM STATEMENT :-

To build an automated system to generate E-challans based on image recognition by detection of any violation of traffic rules by two wheeler rider.

OBJECTIVE :-

- Increase awareness and strictness among people.
- Deployment of an automatic e-challan generation for:-
  - Not wearing Helmet for both rider and pillion rider
  - Over Speeding
  - Triple riding
- To avoid the casual behaviour of cops
- To reduce workload of traffic police

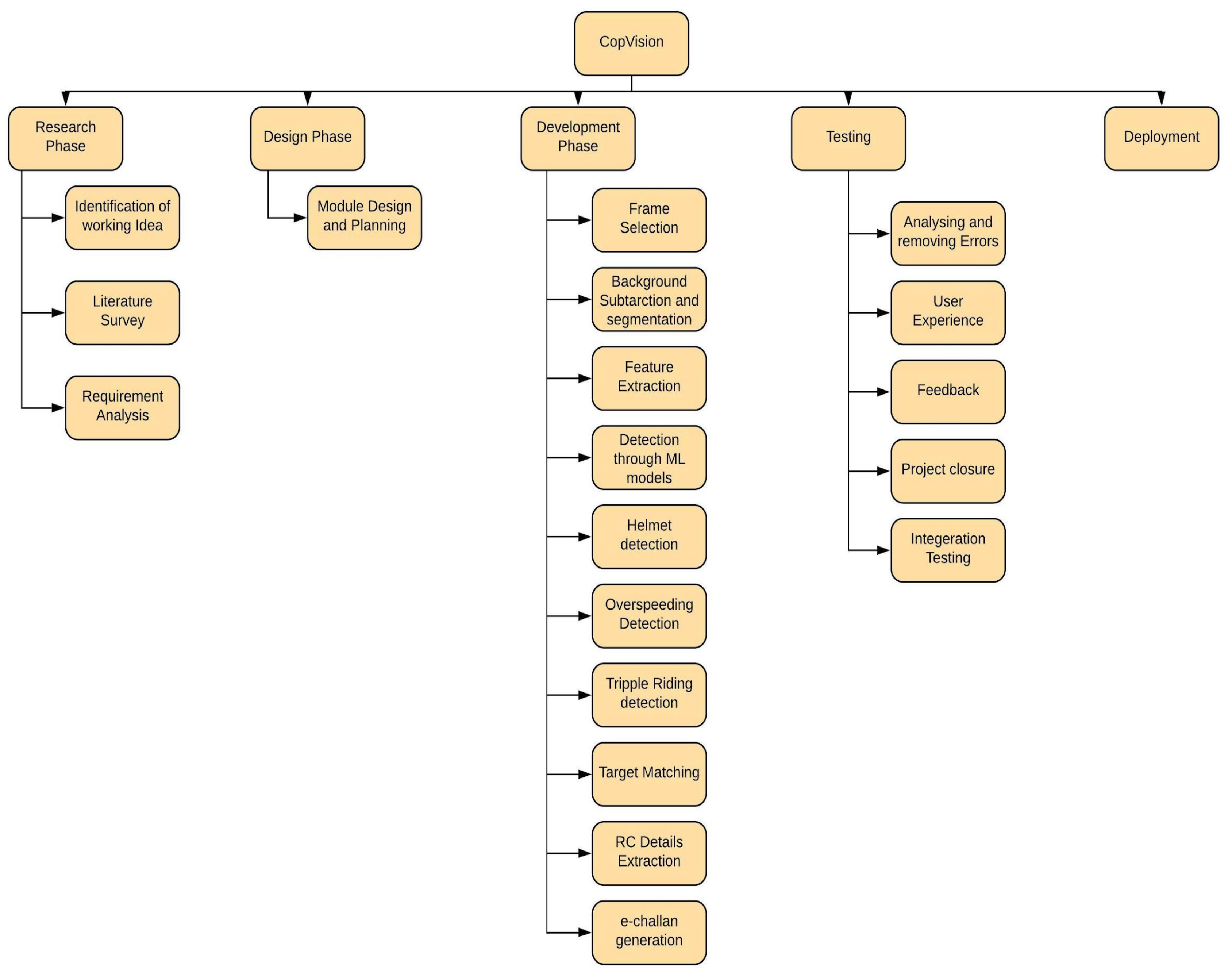
04

LITERATURE SURVEY

S.NO	PAPER TITLE	FINDINGS	CITATIONS
1	Detection of Motorcyclists without Helmet in Videos using Convolutional Neural Network	Adaptive Background modelling	C. Vishnu [1]
2	A Study of Low-resolution Helmet Image Recognition Combining Statistical Features with ANN	Fog removal image processing technique	H. Xue [2]
3	License plate recognition from still images and video sequences: A survey	Public image database for researchers	Anagnost opoulos et a l. [3]
4	Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time	Use of HOG, SIFT and LBP algorithm for extraction of features from upper part of body for helmet detection	Kunal Dahiya, Dinesh Singh, C. Krishna Mohan [4]
5	Real-Time Vehicle Speed Detection Algorithm using Motion Vector Technique	Use of Video Processing and Vector -Valued function for vehicle motion velocity estimation in MATLAB	Ranjit, S.S.S., Anas, S.A., Subramaniam, S.K., Lim, K.C [5]
6	Vehicle speed detection in video image sequences using CVS method	Captured videos were collected by stationary camera which was calibrated based on geometrical equations and speed calculated using position of each frame.	Karim, M.R. and Dehghani, A. [6]

05

ARCHITECTURE OF THE PROJECT

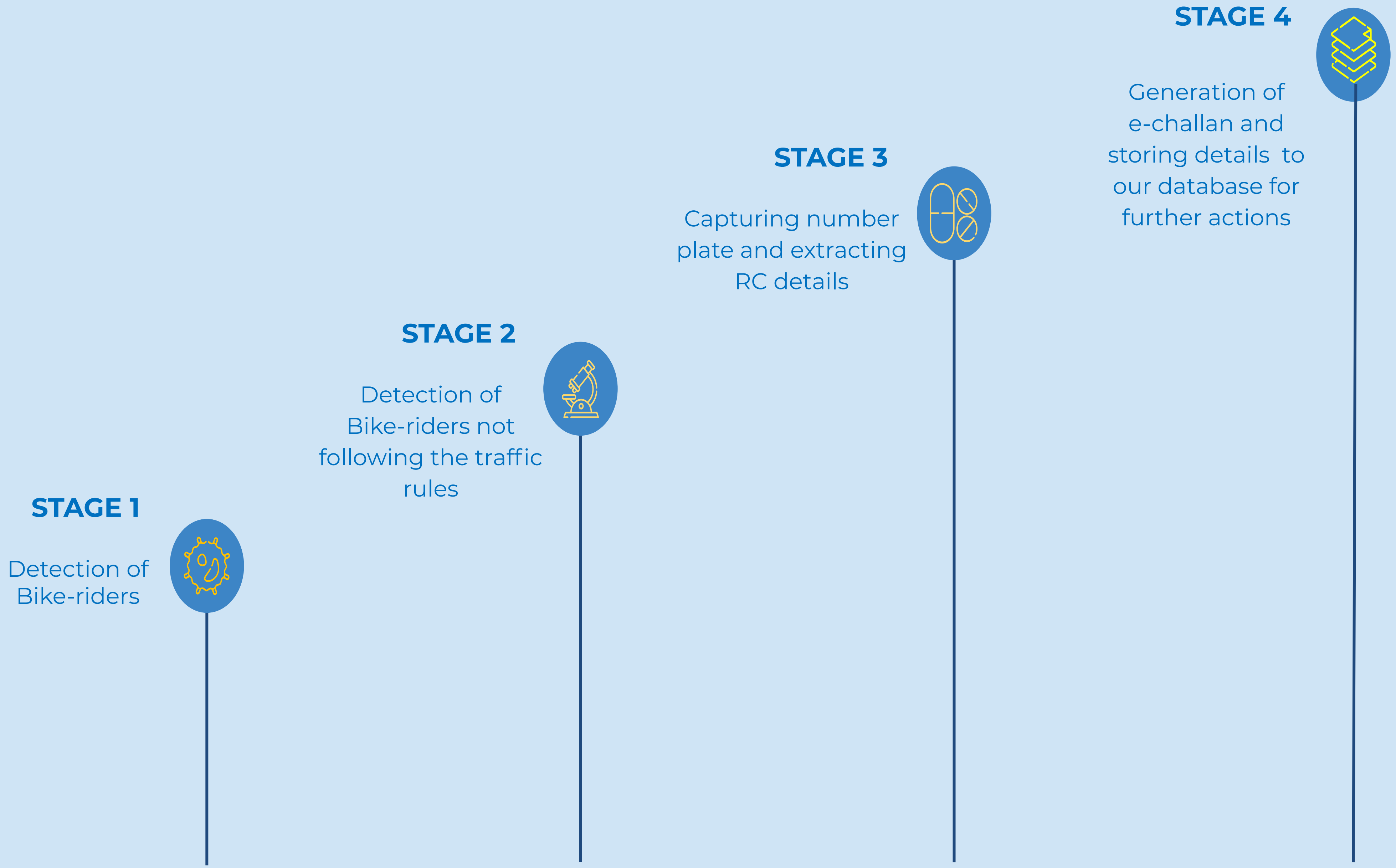


CopVision

- Research Phase
  - Identification of working Idea
  - Literature Survey
  - Requirement Analysis
- Design Phase
  - Module Design and Planning
- Development Phase
  - Frame Selection
  - Background Subtarction and segmentation
  - Feature Extraction
  - Detection through ML models
  - Helmet detection
  - Overspeeding Detection
  - Tripple Riding detection
  - Target Matching
  - RC Details Extraction
  - e-challan generation
- Testing
  - Analysing and removing Errors
  - User Experience
  - Feedback
  - Project closure
  - Integeration Testing
- Deployment

06

TECHNIQUES AND TOOLS USED



STAGE 1

Detection of Bike-riders

STAGE 2

Detection of Bike-riders not following the traffic rules

STAGE 3

Capturing number plate and extracting RC details

STAGE 4

Generation of e-challan and storing details to our database for further actions

TensorFlow

Django

OpenCV
for Recognition

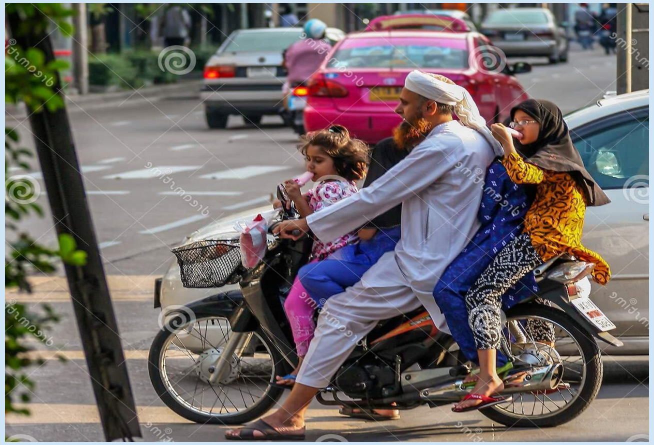
Convolutional Neural
Networks/SVM
for classification

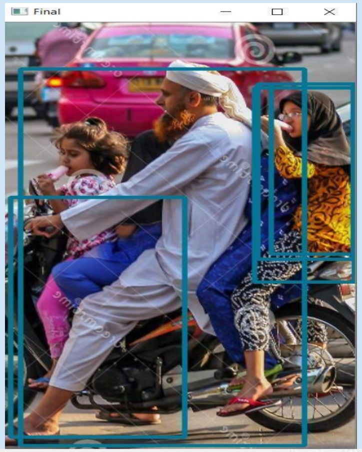
Python

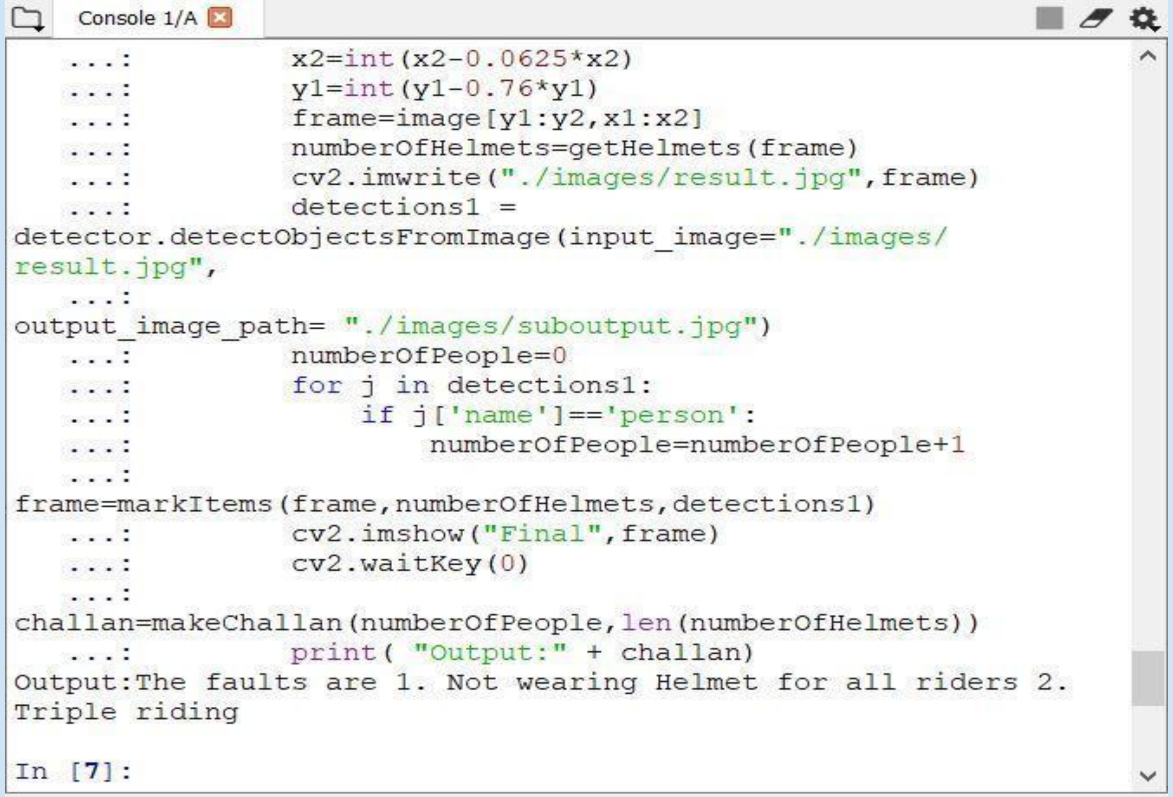
07

MOST RELEVANT SNAPSHOTS OF THE PROJECT

RULE VIOLATION

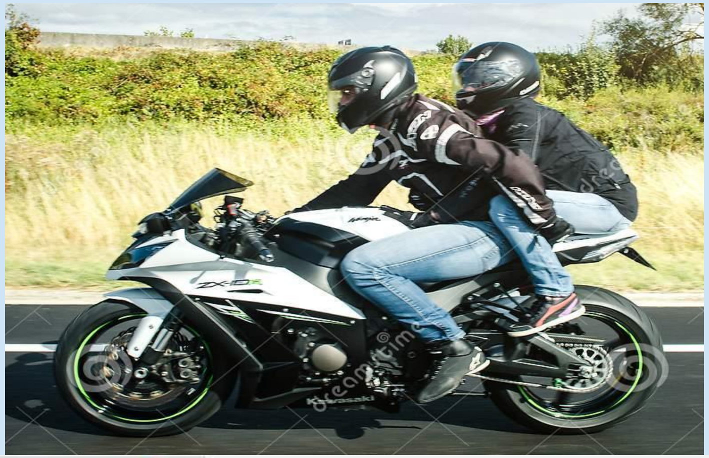


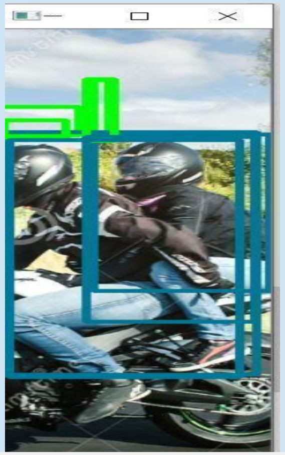


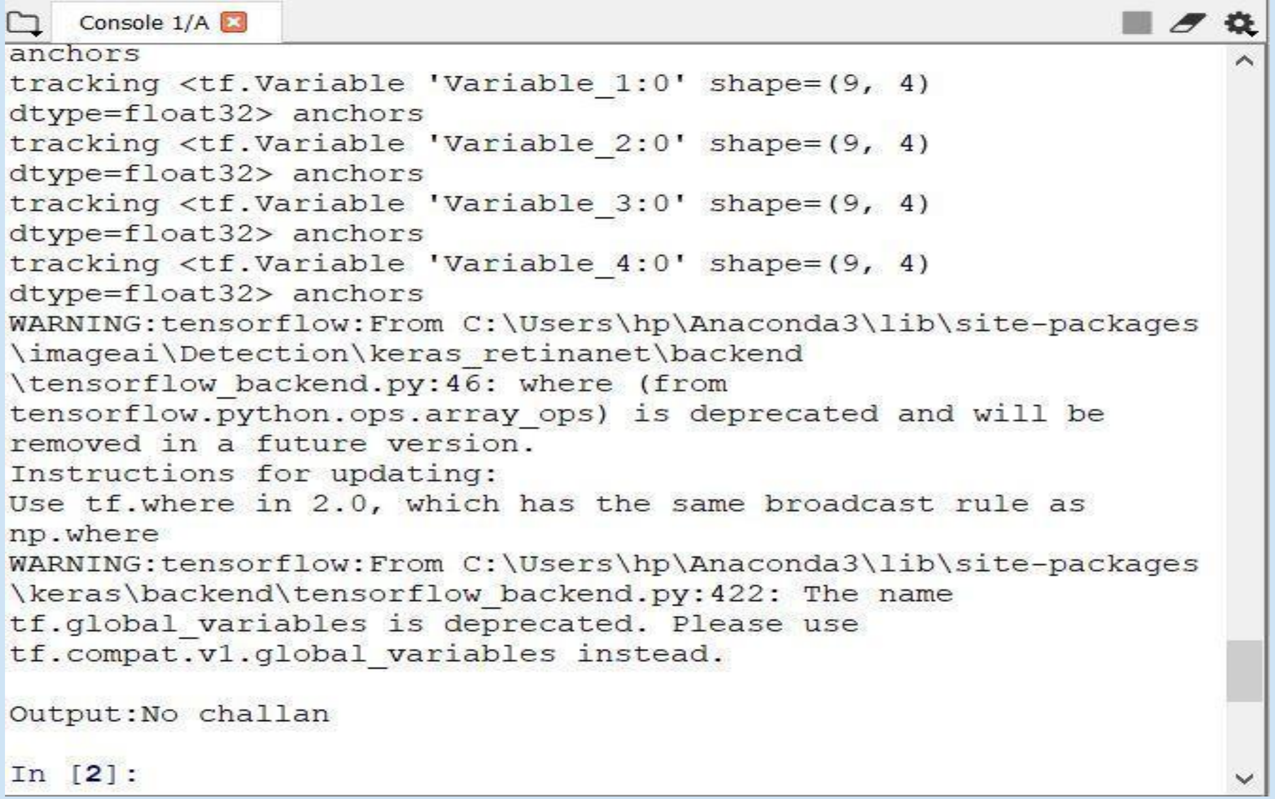


```
   ...:           x2=int(x2-0.0625*x2)
   ...:           y1=int(y1-0.76*y1)
   ...:           frame=image[y1:y2,x1:x2]
   ...:           numberOfHelmets=getHelmets(frame)
   ...:           cv2.imwrite("./images/result.jpg",frame)
   ...:           detections1 =
detector.detectObjectsFromImage(input_image="./images/
result.jpg",
   ...:
output_image_path= "./images/suboutput.jpg")
   ...:           numberOfPeople=0
   ...:           for j in detections1:
   ...:               if j['name']=='person':
   ...:                   numberOfPeople=numberOfPeople+1
   ...:
frame=markItems(frame,numberOfHelmets,detections1)
   ...:           cv2.imshow("Final",frame)
   ...:           cv2.waitKey(0)
   ...:
challan=makeChallan(numberOfPeople,len(numberOfHelmets))
   ...:           print( "Output:" + challan)
Output:The faults are 1. Not wearing Helmet for all riders 2.
Triple riding

In [7]:
```

NO RULE VIOLATED





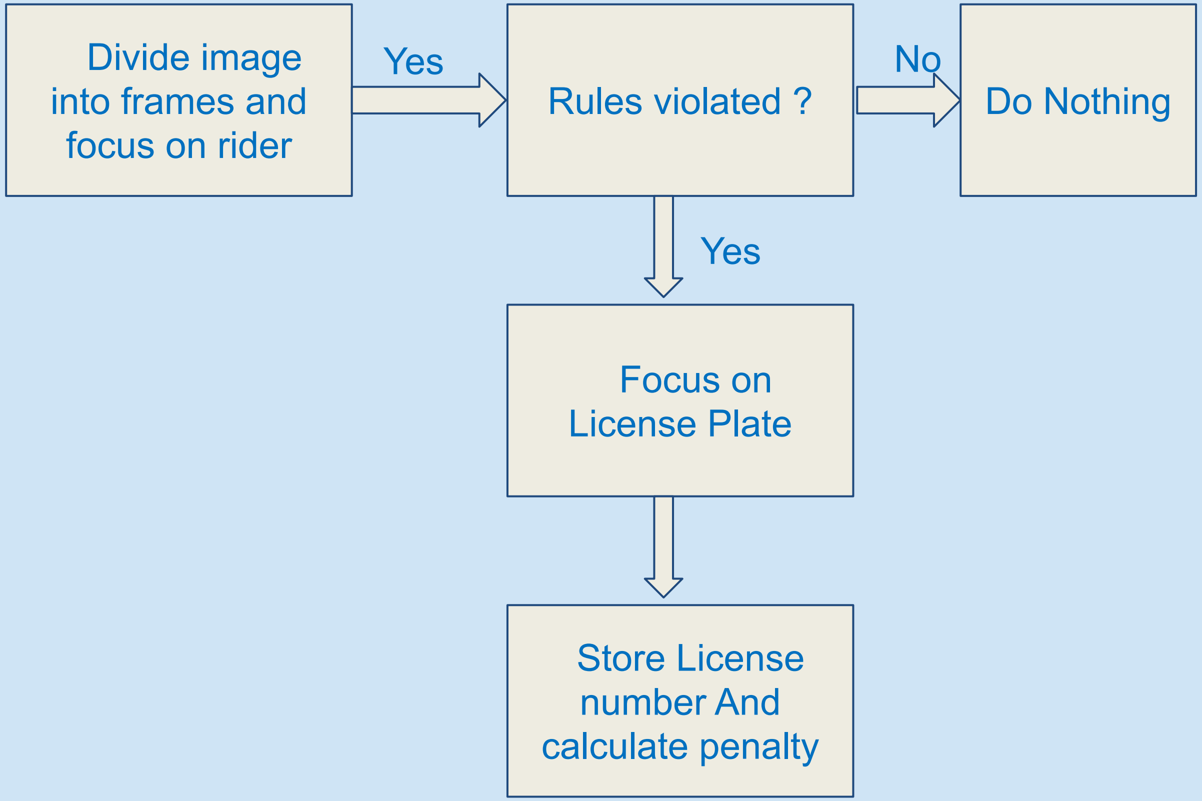


```
anchors
tracking <tf.Variable 'Variable_1:0' shape=(9, 4)
dtype=float32> anchors
tracking <tf.Variable 'Variable_2:0' shape=(9, 4)
dtype=float32> anchors
tracking <tf.Variable 'Variable_3:0' shape=(9, 4)
dtype=float32> anchors
tracking <tf.Variable 'Variable_4:0' shape=(9, 4)
dtype=float32> anchors
WARNING:tensorflow:From C:\Users\hp\Anaconda3\lib\site-packages
\imageai\Detection\keras_retinanet\backend
\tensorflow_backend.py:46: where (from
tensorflow.python.ops.array_ops) is deprecated and will be
removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as
np.where
WARNING:tensorflow:From C:\Users\hp\Anaconda3\lib\site-packages
\keras\backend\tensorflow_backend.py:422: The name
tf.global_variables is deprecated. Please use
tf.compat.v1.global_variables instead.

Output:No challan

In [2]:
```

08

METHODOLOGY

Flowchart for Rules Violation Detection



Divide image into frames and focus on rider

Yes

Rules violated ?

No

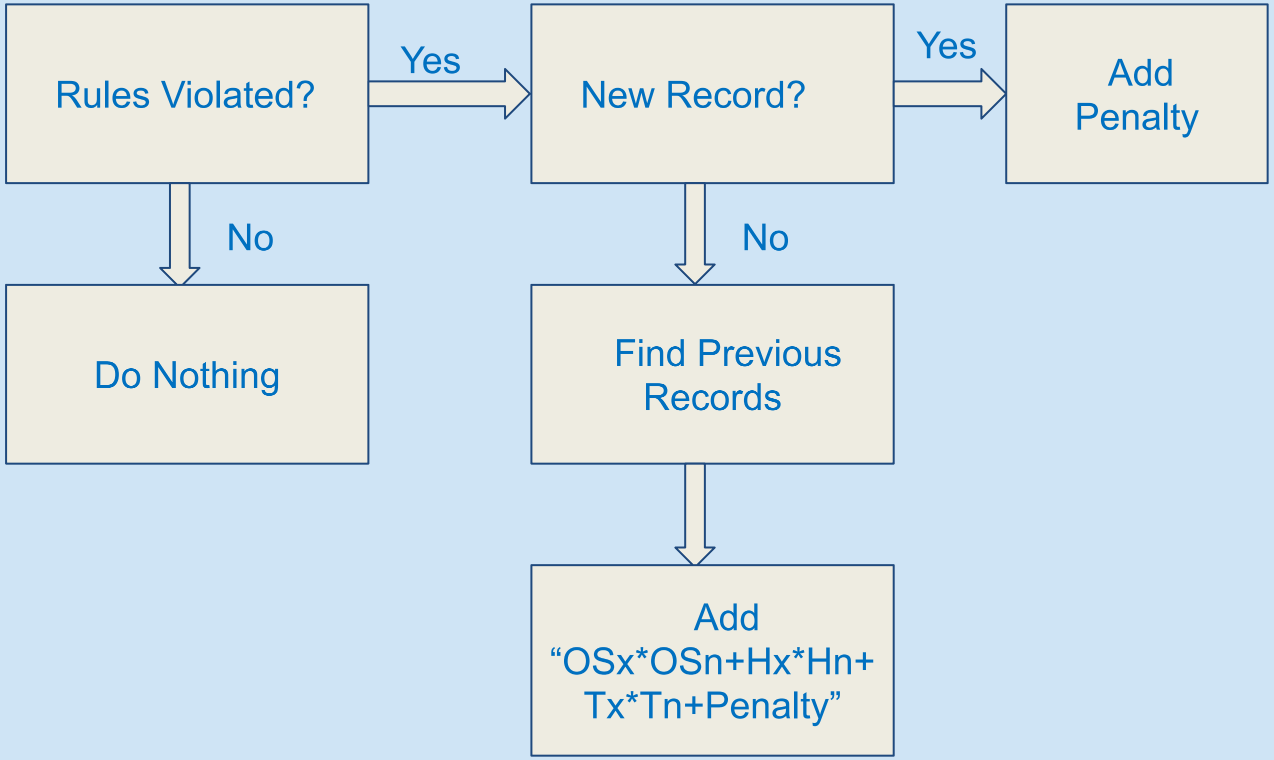
Do Nothing

Yes

Focus on License Plate

Store License number And calculate penalty

Flowchart for penalty calculation



Rules Violated?

Yes

New Record?

Yes

Add Penalty

No

Do Nothing

No

Find Previous Records

Add "OSx*OSn+Hx*Hn+ Tx*Tn+Penalty"

METHODOLOGY

To complete our objective, we started with literature review to know the requirements of our project and also analyze the feasibility of our project in every manner possible that makes it a good option for real time implementation in government traffic police surveillance systems.

Prototypes will be discussed along with different modules and their respective coding. Afterwards we will move forward on the practical implementation.

In the first phase, we detect a bike-rider in the video frame.

In the second phase, we locate the head of the bike-rider and detect whether the rider is using a helmet or not. Moreover we will also extract details of triple bike riders and store their RC details from vehicle Number plate. In order to reduce false predictions, we consolidate the results from consecutive frames for final prediction.

In third phase, we will make a web interface using web development which uses the methodology to achieve our objectives as follows:

Background Modeling: Background subtraction is used to separate the objects in motion such as bikes, humans, cars from static objects such as trees, roads and buildings in frames extracted from the video. Gaussian filter (smoothing) operation is applied on it to improve quality. Further, thresholding accompanied with Morphological operations like closing operations are carried out, through which contours are drawn.

Phase I:

Detection of Bike-riders:-

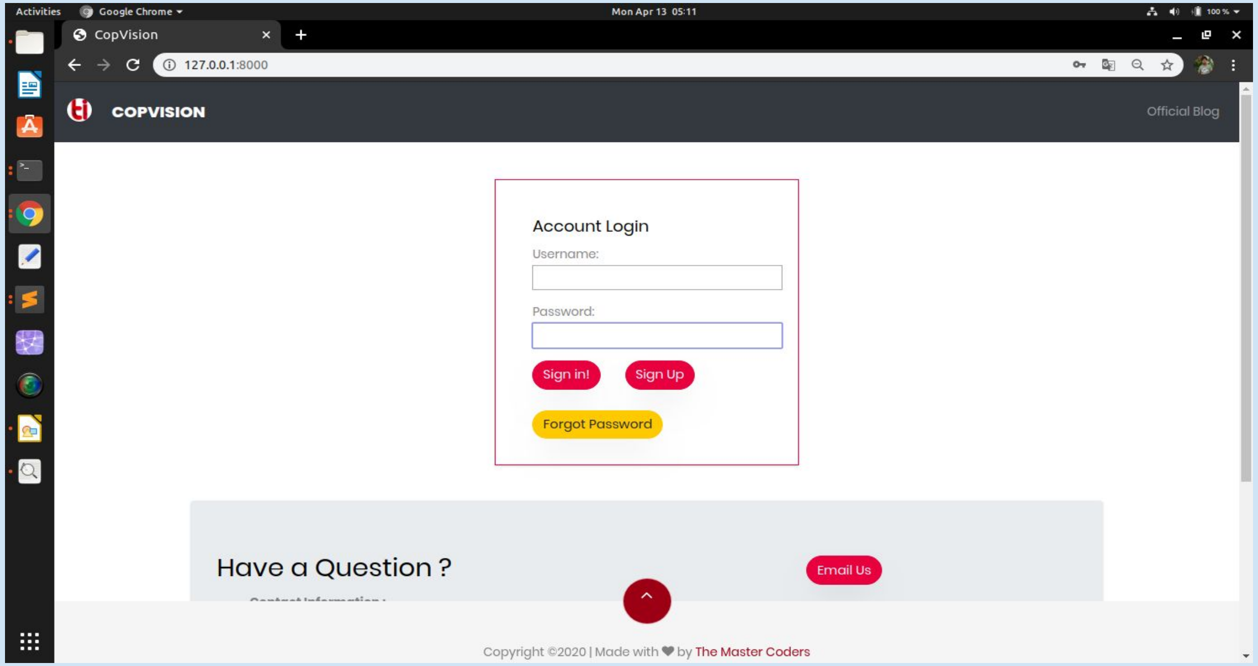
- Thresholding: Image segmentation is done by performing thresholding operation on the frames obtained in order to get contours, which further are attached by performing another morphological operation- closing.
- Aspect ratio: Centre of mass is tracked and we try to distinguish between a car and a bike, a horizontal center line(CL) on the frame enables us to do the same by using the concept of aspect ratio;

Phase II:

Detection of Bike-riders not following traffic rules:-

- Feature Extraction: Identified region (25%) around the head of the bike-rider is used to determine if the bike rider is using the helmet or not.
- Classification: (through SVM- support vector machine; Machine Learning algorithm) The method needs to determine if the biker is violating the law (i.e. not using the helmet and tripling). For this we consider two classes, Bike-rider not following traffic rules (Positive result) and Bike-riders following traffic rules (Negative result).

After building all the modules we will integrate them. Testing will be our next phase to make it more feasible, reliable and compatible in every aspect. Also, we will try to make further improvements.



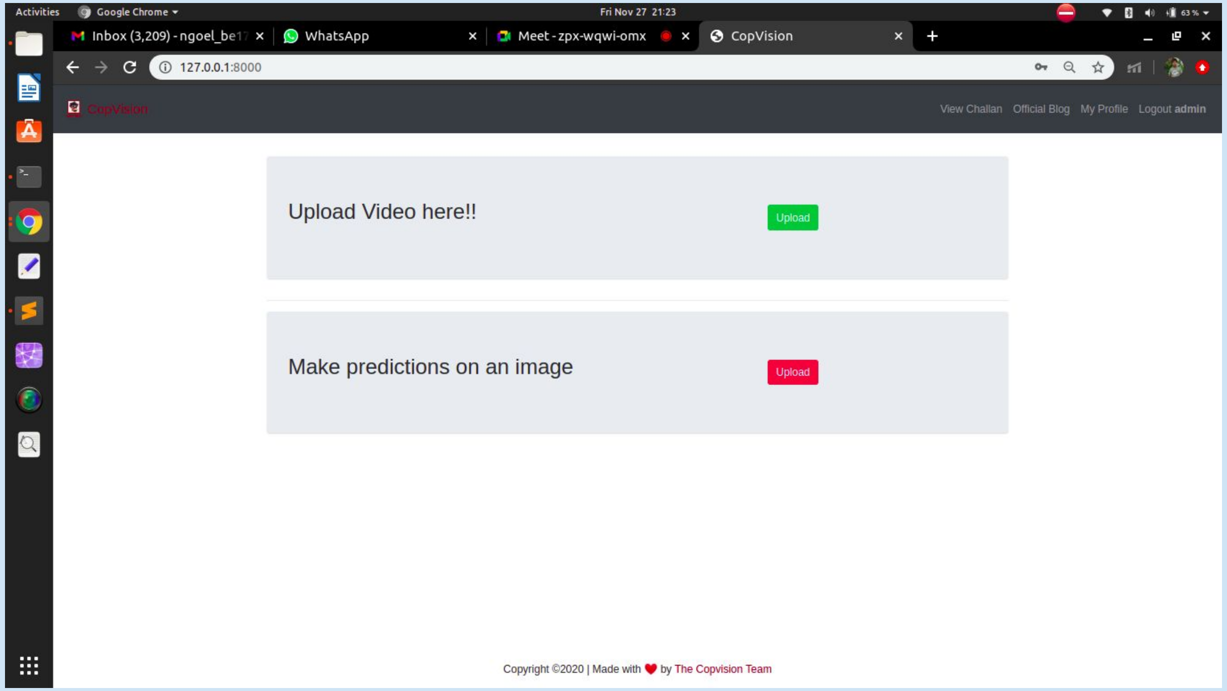
Account Login

Username:

Password:

Have a Question ?

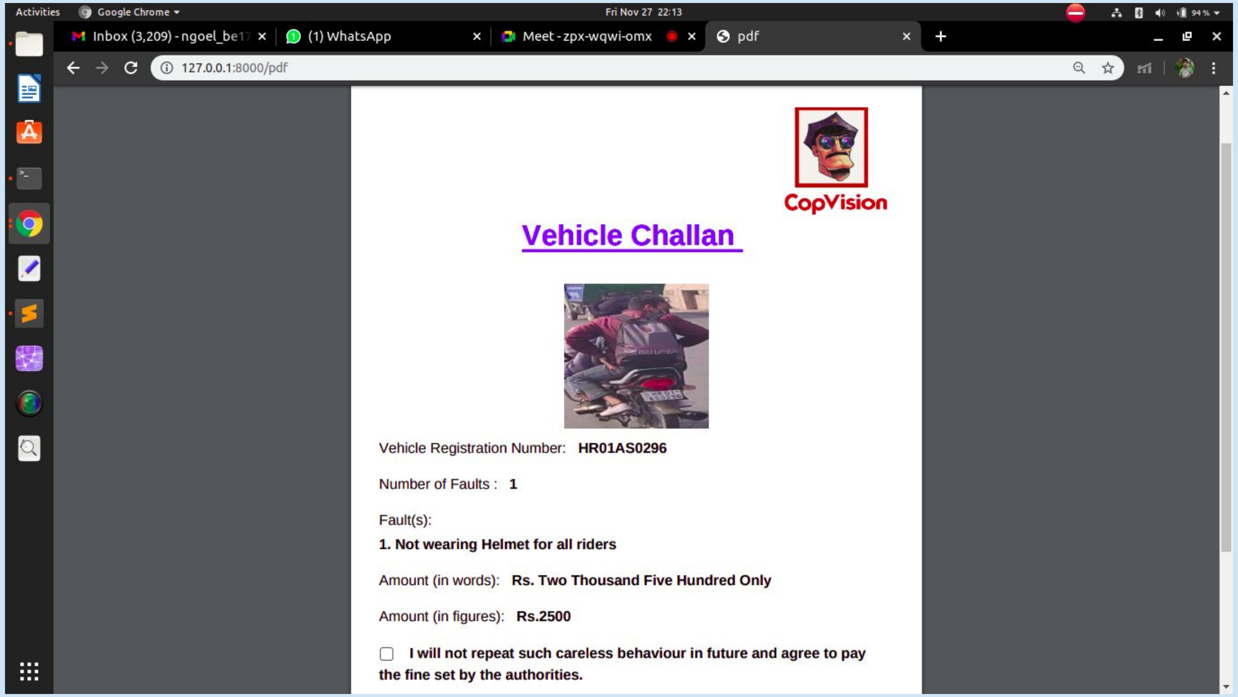
Copyright ©2020 | Made with ♥ by The Master Coders



Upload Video here!!

Make predictions on an image

Copyright ©2020 | Made with ♥ by The Copvision Team



Vehicle Challan

Vehicle Registration Number: **HR01AS0296**

Number of Faults : **1**

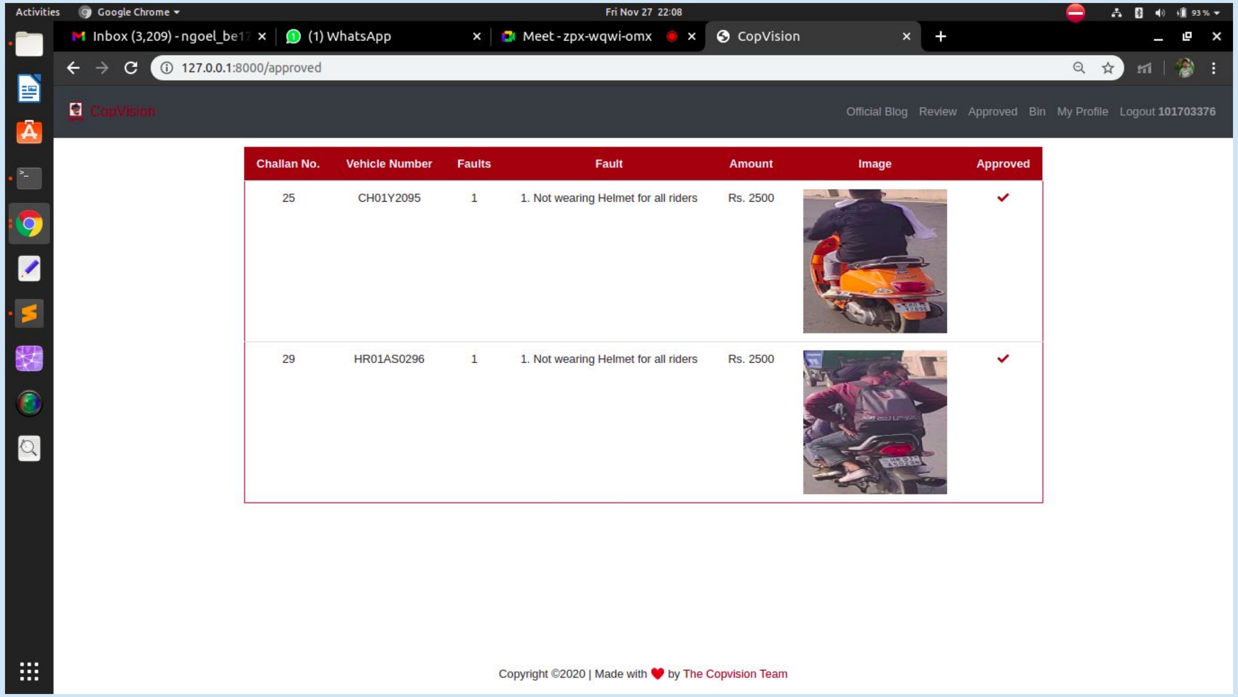
Fault(s):

1. Not wearing Helmet for all riders

Amount (in words): **Rs. Two Thousand Five Hundred Only**

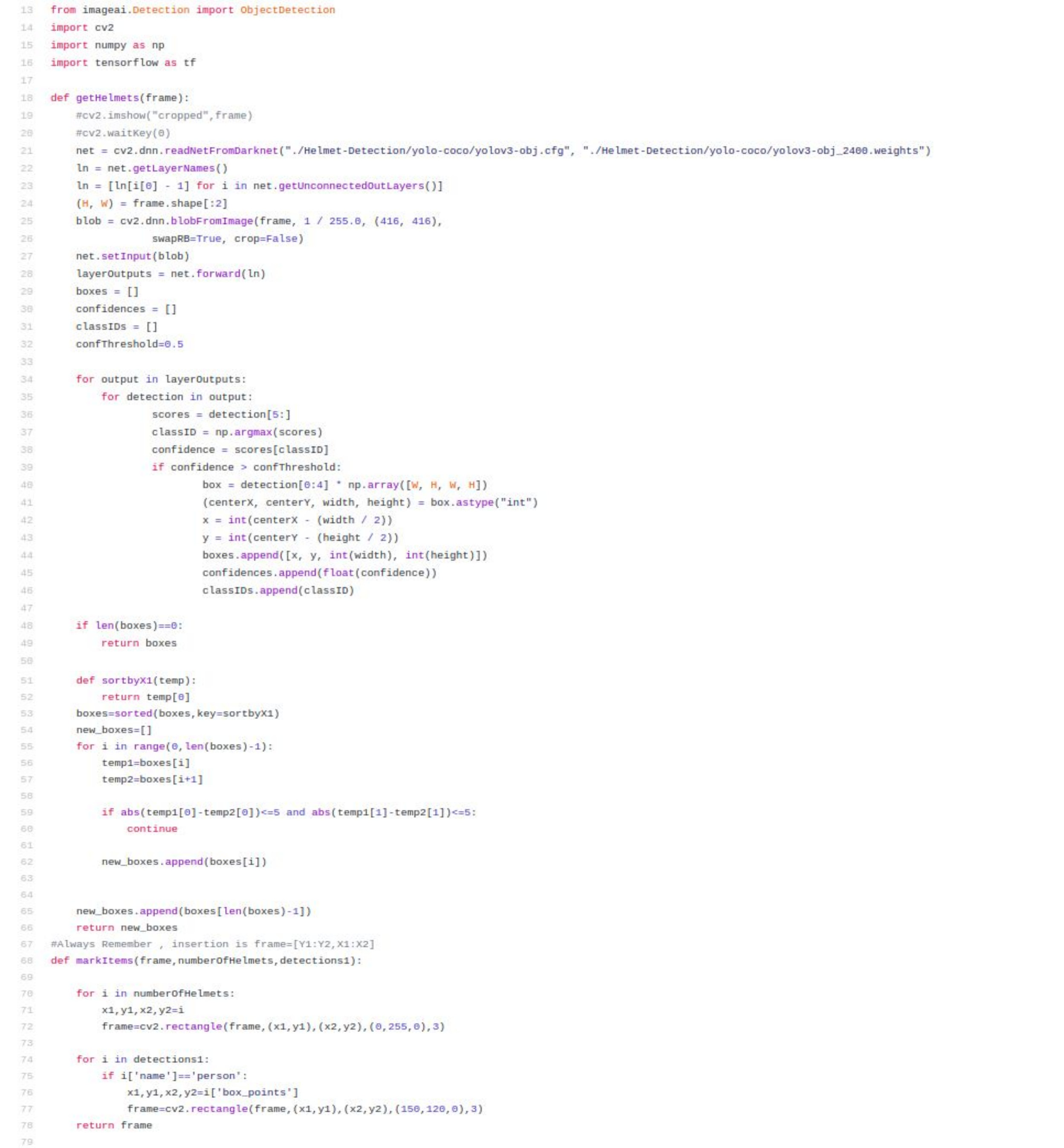
Amount (in figures): **Rs.2500**

☐ **I will not repeat such careless behaviour in future and agree to pay the fine set by the authorities.**

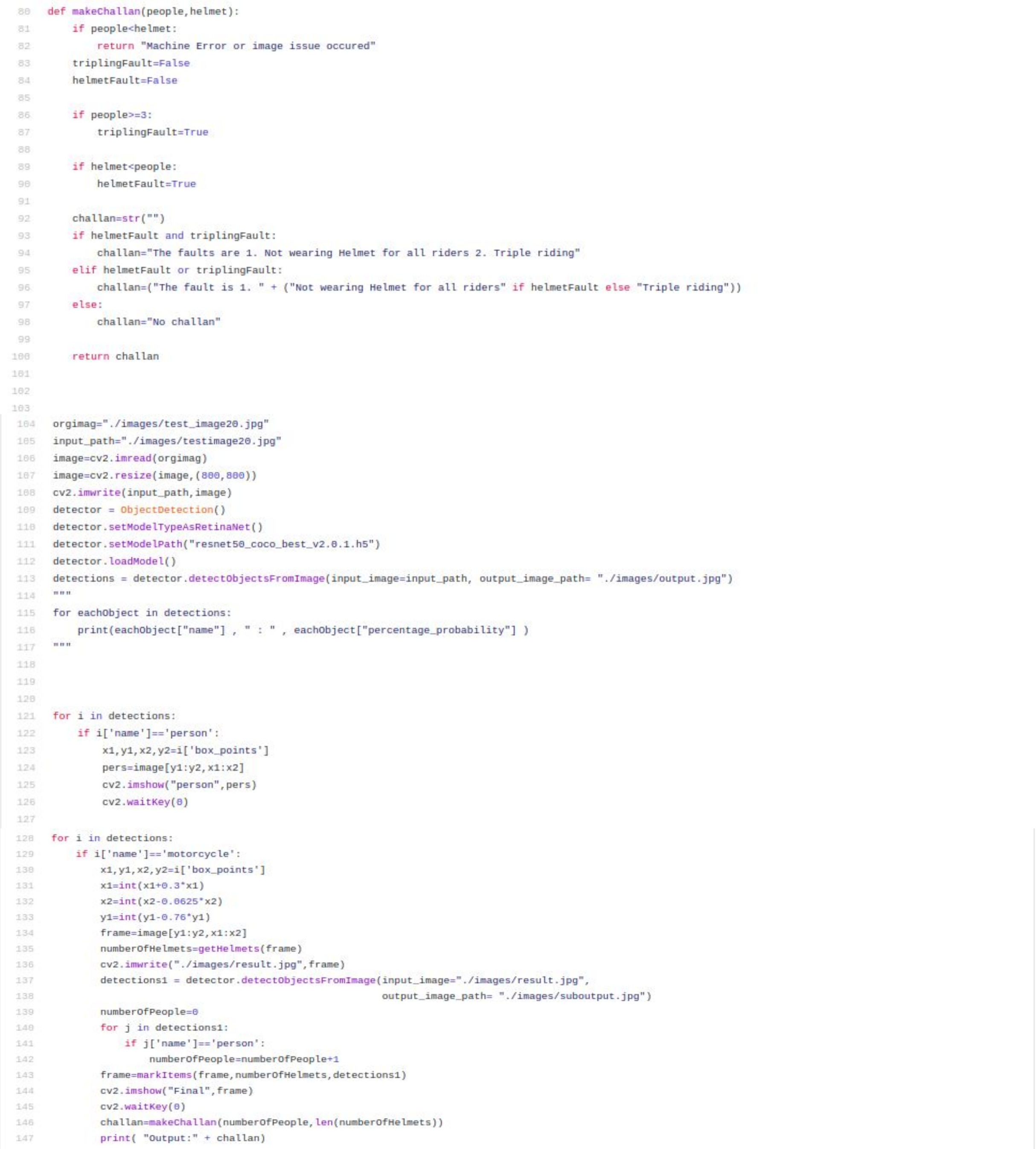


Challan No.	Vehicle Number	Faults	Fault	Amount	Image	Approved
25	CH01Y2095	1	1. Not wearing Helmet for all riders	Rs. 2500		✓
29	HR01AS0296	1	1. Not wearing Helmet for all riders	Rs. 2500		✓

Copyright ©2020 | Made with ♥ by The Copvision Team



```python
from imageai.Detection import ObjectDetection
import cv2
import numpy as np
import tensorflow as tf

def getHelmets(frame):
    #cv2.imshow("cropped",frame)
    #cv2.waitKey(0)
    net = cv2.dnn.readNetFromDarknet("./Helmet-Detection/yolo-coco/yolov3-obj.cfg", "./Helmet-Detection/yolo-coco/yolov3-obj_2400.weights")
    ln = net.getLayerNames()
    ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]
    (H, W) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),
                swapRB=True, crop=False)
    net.setInput(blob)
    layerOutputs = net.forward(ln)
    boxes = []
    confidences = []
    classIDs = []
    confThreshold=0.5

    for output in layerOutputs:
        for detection in output:
                scores = detection[5:]
                classID = np.argmax(scores)
                confidence = scores[classID]
                if confidence > confThreshold:
                        box = detection[0:4] * np.array([W, H, W, H])
                        (centerX, centerY, width, height) = box.astype("int")
                        x = int(centerX - (width / 2))
                        y = int(centerY - (height / 2))
                        boxes.append([x, y, int(width), int(height)])
                        confidences.append(float(confidence))
                        classIDs.append(classID)

    if len(boxes)==0:
        return boxes

    def sortbyX1(temp):
        return temp[0]
    boxes=sorted(boxes,key=sortbyX1)
    new_boxes=[]
    for i in range(0,len(boxes)-1):
        temp1=boxes[i]
        temp2=boxes[i+1]

        if abs(temp1[0]-temp2[0])<=5 and abs(temp1[1]-temp2[1])<=5:
            continue

        new_boxes.append(boxes[i])


    new_boxes.append(boxes[len(boxes)-1])
    return new_boxes
#Always Remember , insertion is frame=[Y1:Y2,X1:X2]
def markItems(frame,numberOfHelmets,detections1):

    for i in numberOfHelmets:
        x1,y1,x2,y2=i
        frame=cv2.rectangle(frame,(x1,y1),(x2,y2),(0,255,0),3)

    for i in detections1:
        if i['name']=='person':
            x1,y1,x2,y2=i['box_points']
            frame=cv2.rectangle(frame,(x1,y1),(x2,y2),(150,120,0),3)
    return frame

```



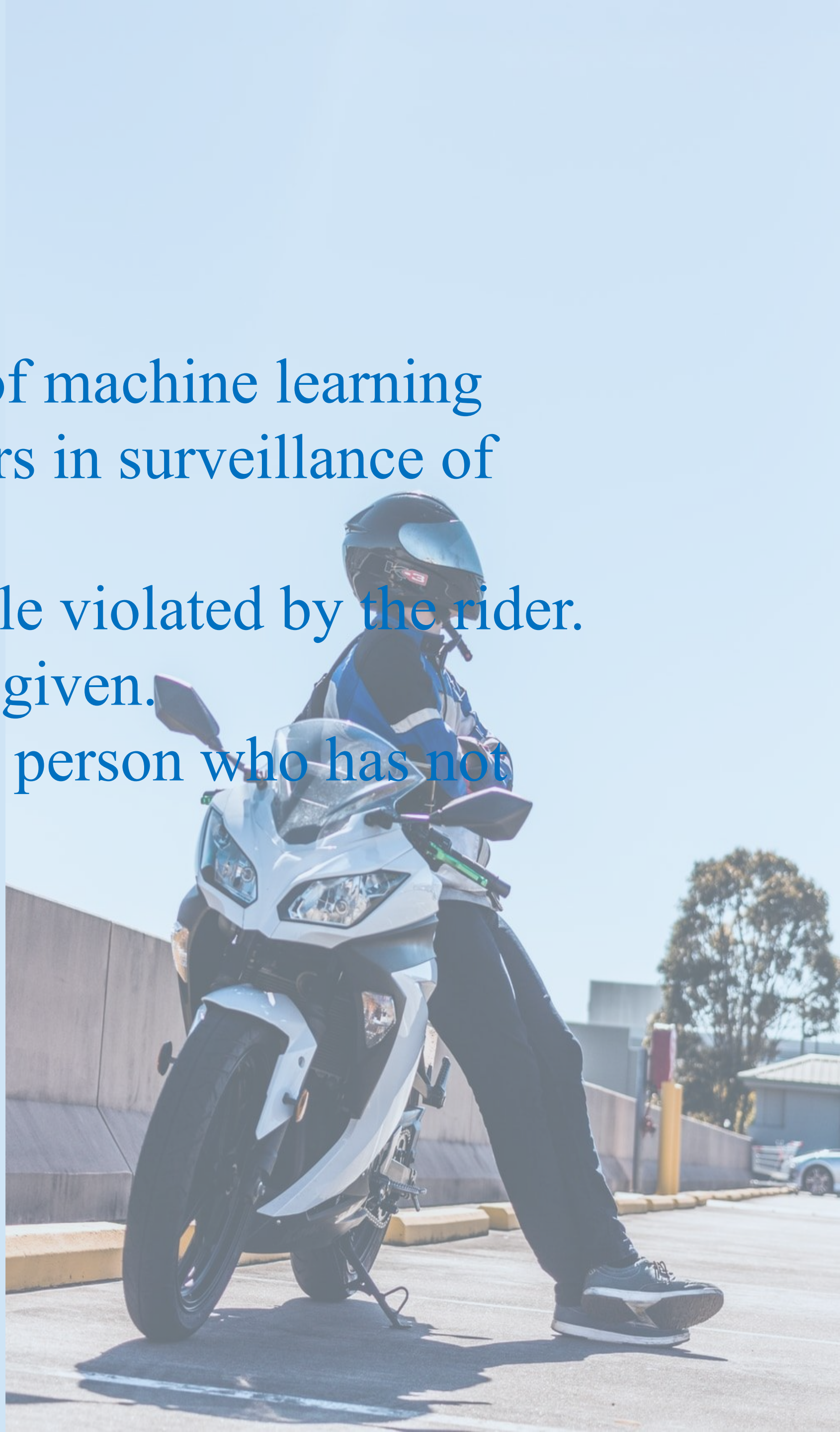
```python
def makeChallan(people,helmet):
    if people<helmet:
        return "Machine Error or image issue occured"
    triplingFault=False
    helmetFault=False

    if people>=3:
        triplingFault=True

    if helmet<people:
        helmetFault=True

    challan=str("")
    if helmetFault and triplingFault:
        challan="The faults are 1. Not wearing Helmet for all riders 2. Triple riding"
    elif helmetFault or triplingFault:
        challan=("The fault is 1. " + ("Not wearing Helmet for all riders" if helmetFault else "Triple riding"))
    else:
        challan="No challan"

    return challan



orgimag="./images/test_image20.jpg"
input_path="./images/testimage20.jpg"
image=cv2.imread(orgimag)
image=cv2.resize(image,(800,800))
cv2.imwrite(input_path,image)
detector = ObjectDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath("resnet50_coco_best_v2.0.1.h5")
detector.loadModel()
detections = detector.detectObjectsFromImage(input_image=input_path, output_image_path= "./images/output.jpg")
"""
for eachObject in detections:
    print(eachObject["name"] , " : " , eachObject["percentage_probability"] )
"""



for i in detections:
    if i['name']=='person':
        x1,y1,x2,y2=i['box_points']
        pers=image[y1:y2,x1:x2]
        cv2.imshow("person",pers)
        cv2.waitKey(0)

for i in detections:
    if i['name']=='motorcycle':
        x1,y1,x2,y2=i['box_points']
        x1=int(x1+0.3*x1)
        x2=int(x2-0.0625*x2)
        y1=int(y1-0.76*y1)
        frame=image[y1:y2,x1:x2]
        numberOfHelmets=getHelmets(frame)
        cv2.imwrite("./images/result.jpg",frame)
        detections1 = detector.detectObjectsFromImage(input_image="./images/result.jpg",
                                                      output_image_path= "./images/suboutput.jpg")
        numberOfPeople=0
        for j in detections1:
            if j['name']=='person':
                numberOfPeople=numberOfPeople+1
        frame=markItems(frame,numberOfHelmets,detections1)
        cv2.imshow("Final",frame)
        cv2.waitKey(0)
        challan=makeChallan(numberOfPeople,len(numberOfHelmets))
        print( "Output:" + challan)
```

09

DELIVERABLES OF THE PROJECT

DELIVERABLES OF THE PROJECT

- Detects bike riders without helmets and tripling by use of machine learning models with a high accuracy by minimizing human errors in surveillance of safety rules.
- A software to generate E-challans based on the traffic rule violated by the rider.
- A user can also view the history of challans he has been given.
- Also calculate the “The most cautious citizen” to get the person who has not broken a traffic rule again after getting a challan.
- Store data of all the challan and their details in database.



10

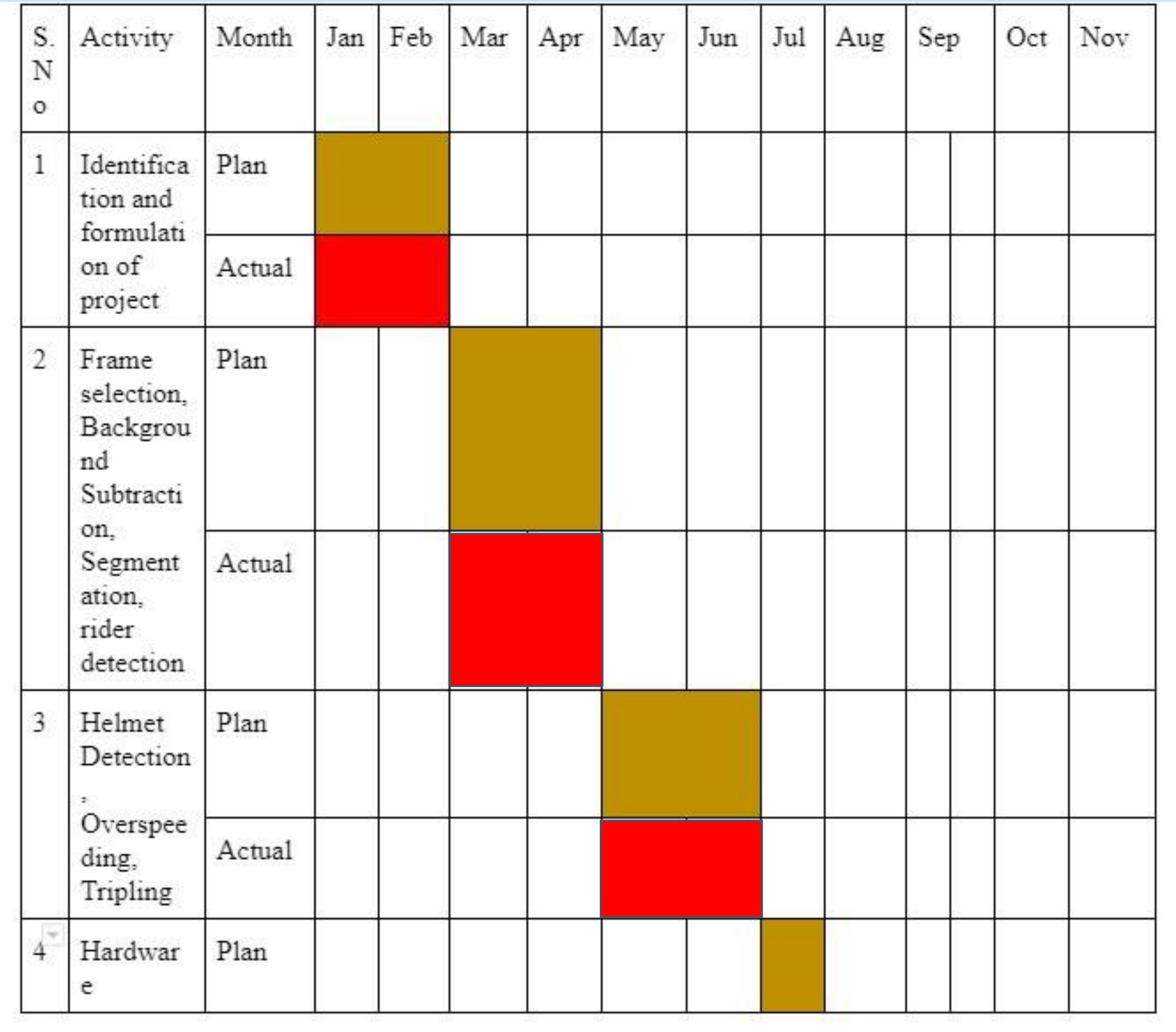
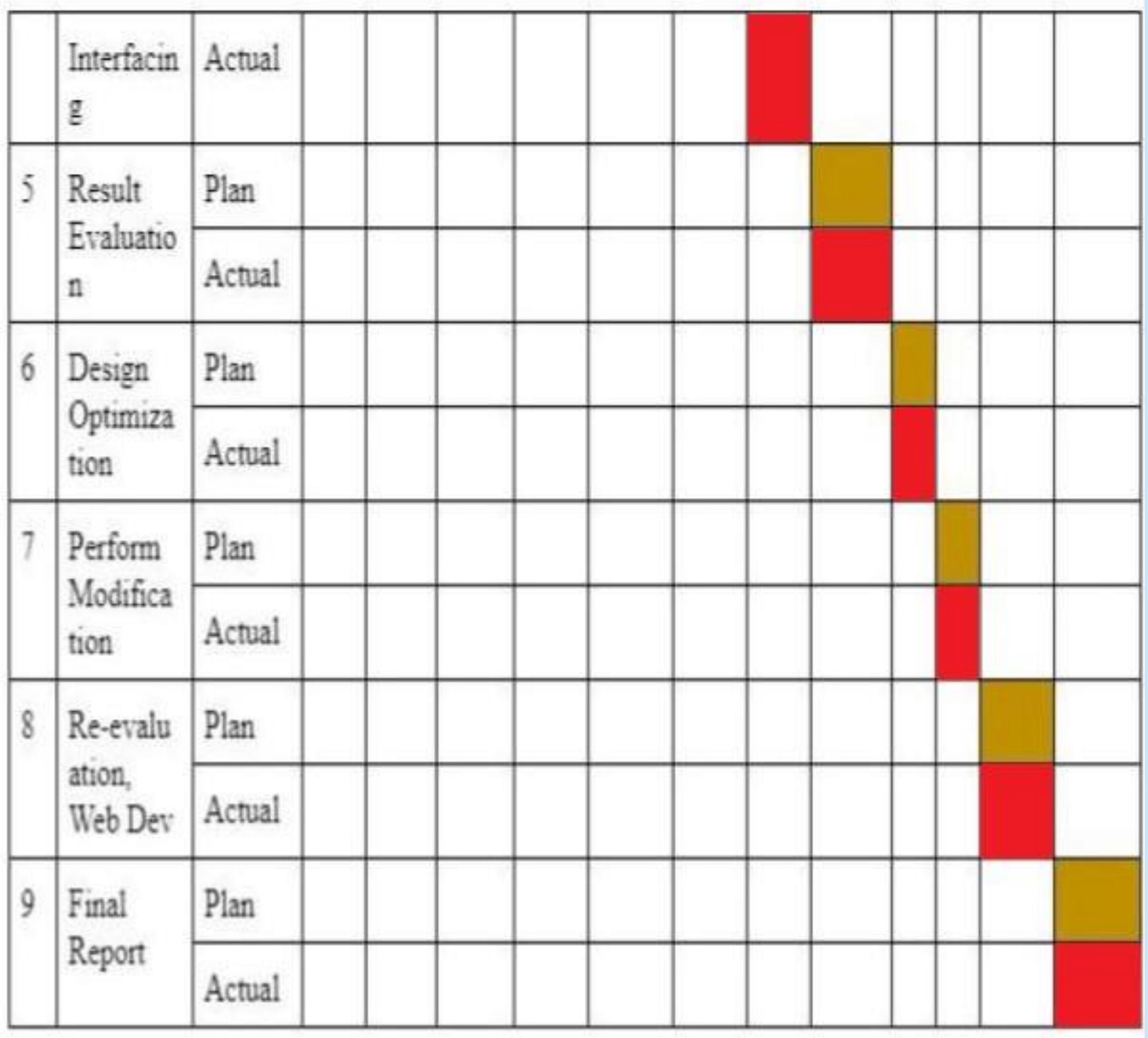
PROFESSIONAL AND TECHNICAL LEARNING

Subject Code	Subject Name	Description
UCS615	Image Processing	The idea of the project is chosen through this subject's knowledge. In this project, MATLAB was used to extract Histogram-oriented features from the dataset, and to map them with the target (wearing helmet or not,tripling).
UML501	Machine Learning	The importance of Machine learning in our project lies in the object Learning detection and extraction steps; along with model building and predicted values on new data. OpenCV library has been used to do video capture, extract frames, to make contours and to create bounding boxes around the two-wheeler. R platform is used for model building and predicting newvalues.
COURSERA	Technical Report Writing	Completed an online course authorized by Rice University and offered through Coursera for technical writing which helped us build skills for research paper writings.

11

INDIVIDUAL ROLES

WORK PLAN

S. No	Activity	Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep		Oct	Nov
1	Identification and formulation of project	Plan	🟫	🟫										
		Actual	🟥	🟥										
2	Frame selection, Background Subtraction, Segmentation, rider detection	Plan			🟫	🟫								
		Actual			🟥	🟥								
3	Helmet Detection , Overspeeding, Tripling	Plan					🟫	🟫						
		Actual					🟥	🟥						
4	Hardware	Plan							🟫					
	Interfacing	Actual							🟥					
5	Result Evaluation	Plan								🟫				
		Actual								🟥				
6	Design Optimization	Plan									🟫			
		Actual									🟥			
7	Perform Modification	Plan										🟫		
		Actual										🟥		
8	Re-evaluation, Web Dev	Plan											🟫	
		Actual											🟥	
9	Final Report	Plan												🟫
		Actual												🟥

INDIVIDUAL ROLE

Name	Roll Number	Roles
Nipunn Malhotra	101703374	● Code for 2-Wheeler detection ● Design Specification ● Helmet detection ● Number plate detection ● Challan front-end ● Testing and debugging
Nishant Goel	101703376	● Tripling detection ● Front-end development ● Web app Development in Django ● Documentation ● Database design and development ● Testing and debugging
Paras Arora	101703382	● Motorcycle detection code ● UML diagrams ● Data Collection ● Documentation ● Challan Generation ● Testing and debugging

REFERENCES

BIBLIOGRAPHY

1. Vishnu, C., Singh, D., Mohan, C.K. and Babu, S., 2017, May. Detection of motorcyclists without helmet in videos using convolutional neural network. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 3036-3041). IEEE.
2. JIANG, X., XUE, H., ZHANG, L. and ZHOU, Y., 2016. A Study of Low-resolution Safety Helmet Image Recognition Combining Statistical Features with Artificial Neural Network. International Journal of Simulation--Systems, Science & Technology, 17(37).
3. Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Psoroulas, I.D., Loumos, V. and Kayafas, E., 2008. License plate recognition from still images and video sequences: A survey. IEEE Transactions on intelligent transportation systems, 9(3), pp.377-391
4. Dahiya, K., Singh, D. and Mohan, C.K., 2016, July. Automatic detection of bike-riders without helmet using surveillance videos in real-time. In 2016 International Joint Conference on Neural Networks (IJCNN) (pp. 3046-3051). IEEE.
5. Ranjit, S.S.S., Anas, S.A., Subramaniam, S.K., Lim, K.C., Fayeez, A.F.I. and Amirah, A.R., 2012. Real-Time Vehicle Speed Detection Algorithm using Motion Vector Technique.
6. Karim, M.R. and Dehghani, A., 2010. Vehicle speed detection in video image sequences using CVS method. International journal of the Physical Sciences, 5(17), pp.2555-2563.
7. https://www.indiatoday.in/diu/story/two-wheeler-death-road-accidents-helmets-states-india-1602794-2019-09-24