Traffic Rules Surveillance System using Facial Recognition

Rohan Ram Vaswani[1], Ankita Jayprakash Vartak[1], Manik Ramanna Chandey[1]
Mr. Amit Hatekar[2], Mr. K. K. Mathew[2]
[1]Graduate Students, [2]Assistant Professor, [2]Associate Professor
Thadomal Shahani Engineering College, Mumbai

***Abstract*— This paper presents a prototype of a system that improvises the e-Challan system implemented in Mumbai since 2016. The prototype presented makes use of facial recognition along with number plate extraction to identify the drivers violating traffic rules, extract their credentials from the Government database, and penalize them digitally. Raspberry Pi microcontroller is interfaced with Ultrasonic sensor, to detect vehicle motion if the traffic signal is red, and a USB camera, to capture a photograph of the driver and the vehicle. The captured photograph is processed using Haar Cascades facial recognition and Tesseract OCR tool. Programming is implemented by means of Python and its OpenCV library.**

Keywords*— Raspberry Pi, Python, OpenCV, Tesseract, Haar Cascades, VNC, Image Processing, OCR, Edge Detection, Character Segmentation, Number Plate Extraction, Facial Recognition, Camera, e-Challan, RTO, Traffic Rules, Traffic Fines

I. INTRODUCTION

According to a survey conducted by Ministry of Statistics and Programme Implementation (MOSPI), the vehicular traffic in India, as on 31st March 2016, has crossed 23 crores, with the numbers being on a rapid upsurge every passing day [1]. In order to keep a check on this catastrophic traffic, the Government of India formulates protocols to ensure smooth functioning of vehicles in tandem with maintaining safety and security of the citizens. Disobeying those guidelines may lead to unfortunate accidents, loss of property, serious injuries, or even death. Consequently, in addition to road safety measures, the Government also introduces stipulated fines and penalties for those not following the measures at any point of time.

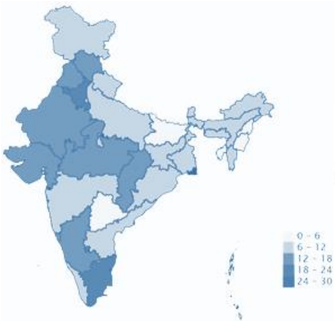


Fig. 1 (a) Distribution of number of road accidents across India

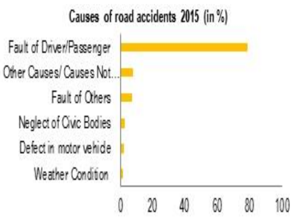


Fig. 1 (b) Graphical analysis of causes of road accidents

Courtesy: PRS Legislative Research (prsindia.org)

In order to ensure that citizens abide by the traffic rules, the Regional Transport Office (RTO) appoints Traffic Police whose duty is to penalize culprits. Yet, it was a more common practice than not to observe people ruthlessly violating traffic rules and eventually getting away with a truncated or no penalty imposed. The reason being that they were successful in bribing the Traffic Cops, who disregarded their ethics and had no second thoughts before falling into the clutches of corruption.

This had been the sorry state of the society since many decades, until August 2016. On the 69th Independence Day, Mumbai's traffic system got inducted with a network of 4500+ CCTV cameras interfaced with traffic signals across the city [2]. The fine collection system went digital and the menace of bribing and corruption was expected to abate substantially by this move. Cameras were juxtaposed with traffic signals with an objective of capturing a photograph of a vehicle that violated traffic signal or a vehicle that went beyond the white line, drawn posterior to zebra crossings, while the signal was red. Then, using image processing and Optical Character Recognition (OCR) the numbers and characters on the number plate of the vehicle were analyzed, using which the details of owner were obtained from the RTO database. Furthermore, corresponding amount of fine was debited from the account of the owner and an electronic challan (e-Challan) was issued to the owner.

Notwithstanding, a major conceptual shortcoming of the e-Challan system was the framework being designed from a vehicle-owner's perspective rather than that of a driver. Hypothetically, if Person A is driving person B's car and is violating the red signal, it is B (owner) who is getting fined despite the fact that A (driver) is the culprit. Having said so, the prototype presented in this paper exploits facial recognition to capture face of the driver of the vehicle; compare and match biometrics of her face with the entries in the driver's license database of RTO; extract her credentials; and levy fine into her account, keeping the owner of the vehicle unaffected.

II. BUILDING BLOCKS

HARDWARE

Raspberry Pi

Fig. 2 Raspberry Pi

Raspberry Pi is a set of small single-board multipurpose computers developed by the Raspberry Pi Foundation. Launched in February 2016, the Raspberry Pi 3 B is the earliest model of the 3rd generation [3]. Its specifications are:

Quad Core 1.2 GHz 64-bit CPU
1 GB RAM
Wireless LAN and Bluetooth connectivity
100 Base Ethernet
40-pin extended GPIO
4 USB 2 ports
Micro SD Port
Full size HDMI port
CSI camera port for Raspberry Pi camera
DSI display port for Raspberry Pi touchscreen display

Ultrasonic Sensor

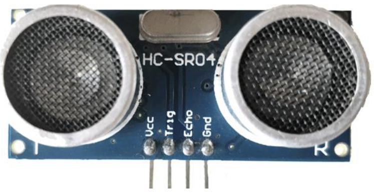


Fig. 3 Ultrasonic Sensor

An ultrasonic sensor is an instrument that works on the principle of measuring distance to an object using ultrasonic sound waves [4]. Ultrasonic waves are sound waves with frequency above the audible range of human beings (>20 kHz). They generate a high-frequency pulse of sound and evaluate the properties of the echo pulse that is reflected by the target object. They evaluate the following three fundamental parameters: Time of flight (for distance), Doppler shift (for velocity), and Amplitude attenuation (for direction).

Universal Serial Bus (USB) Camera

Fig. 4 USB Camera

USB Cameras are imaging cameras employing USB 2.0 or USB 3.0 technology for transferring image data. USB Cameras are designed to easily interface with computers or other devices. The ubiquitousness of USB technology in computer systems as well as the 480 Mb/s transfer rate of USB 2.0 and that of 5 Gb/s of USB 3.0, makes USB Cameras suitable for many imaging applications [5].

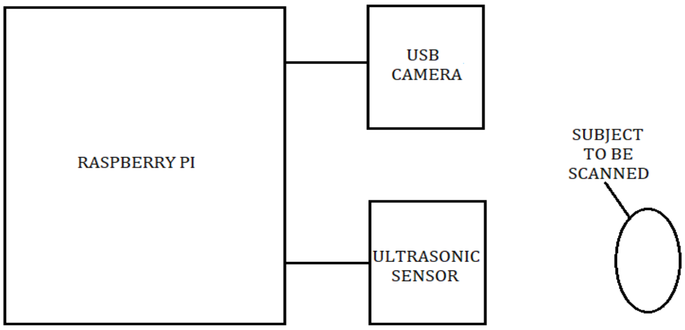


Fig. 5 Block Diagram

SOFTWARE

Raspbian OS

Raspbian is a free operating system optimized for the Raspberry Pi hardware. It consists of set of basic programs and utilities that make a Raspberry Pi run. It comes with over 35,000 packages, libraries, and pre-compiled softwares bundled in a decent format for easy installation and usage [6].

Python

Developed in late 1980's, Python is a multipurpose and high-level programming language. It offers a diverse platform for programmers with its plethora of libraries. Python is also one of the preferred choices of environments when it comes to working with image processing [7].

Open source Computer Vision (OpenCV)

OpenCV is a library of programming functions mainly aimed at real-time computer vision, developed by Intel. OpenCV is written in C++ and is built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products [8]. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms are used to detect and recognize faces, identify objects, classify human actions in videos, track moving objects, extract 3D models of objects, etc.

Tesseract

Optical Character Recognition **(**OCR) is the automatic process of converting typed, handwritten, or printed text to machine-encoded text that we can access and manipulate via a string variable. Tesseract is an OCR engine used for various operating systems. Tesseract engine was originally developed at Hewlett-Packard (HP) but was never commercially exploited [9]. In 2005, HP transferred Tesseract to the Information Science Research Institute (ISRI) and it was released as open source tool.

Haar Cascades

Haar Cascade is a machine learning object detection algorithm in OpenCV used to detect objects or faces in an image or video. A cascade function is trained from a set of positive and negative images. It is then used to identify similar objects in other images [10].

Virtual Network Computing (VNC)

VNC is a popular technology that enables remote desktop sharing over multiple networks. VNC is used to view visual desktop display of one computer on another computer and control that computer over a network connection. The machine that shares its screen is termed as a VNC Server. The program that displays the screen data is termed as VNC client (or VNC Viewer) [11].

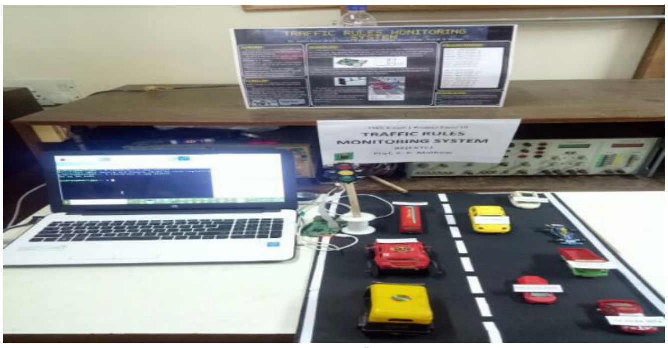

TRAFFIC RULES MONITORING SYSTEM
TRAFFIC RULES MONITORING SYSTEM

Fig. 6 Prototype Implementation

III. THE TECHNOLOGY

Tesseract

When an image is fed into Tesseract, it initially performs edge detection to extract only outlines of prominent objects in the image. For character detection, we would be needing only that portion of the image that contains the characters, where as the rest of the image is considered as noise. Region of Interest (ROI) is the region that contains the objects we want to read by comparing the objects to the character set. Tesseract identifies the ROI with its algorithms and crops only that part of image for further processing.

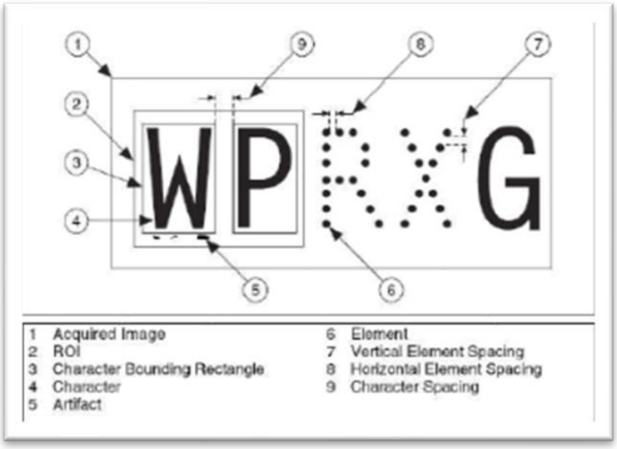

1
2
3
4
5
6
7
8
9
1 Acquired Image
2 ROI
3 Character Bounding Rectangle
4 Character
5 Artifact
6 Element
7 Vertical Element Spacing
8 Horizontal Element Spacing
9 Character Spacing

Fig. 7 Tesseract Steps

The smallest rectangle that fully encloses any character is called Character bounding rectangle [12]. Within the ROI, each character is isolated by one character bounding rectangle each.

Post ROI fixation and character binding, Tesseract adjusts element spacing among the characters. Horizontal element spacing is the gap between two elements that are horizontally afjacent. Tesseract sets this value to 4 or 5 for dot-matrix characters where as 1 or 2 for stroke characters. Dot-matrix characters are comprised of small discrete elements. Stroke characters are continuous characters wherein gaps are only due to imperfections in the image. Vertical element spacing is the gap between two vertically adjacent elements; and this value is by default set to 0.

Now Tesseract performs removal of small particles. The process of removing small particles, such as salt and pepper noise, involves applying a predefined number of 3x3 erosions to the thresholded image [13]. Tesseract fully restores any objects that remain after applying the erosions.

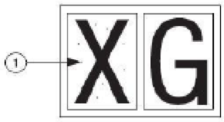


Fig. 8 Eliminating Noise

During the reading procedure, the machine vision application created with Tesseract function segments each object in the image and compares the object with characters in the character set created in the training procedure. This is known as character segmentation. Tesseract extracts unique features from each segmented object in image and compares each object to each character stored in the character set. Tesseract then returns the object that best matches the characters in the character set as recognized characters [14].

Haar Cascades

To train the Haar Cascade function, initially the algorithm is fed with a certain number (at least 25) of positive and negative images of the same face. Further, facial features (*e.g.:* - eyes, nose, lips, etc.) are extracted from each positive and negative image. Haar image masks performing edge detection and line detection are applied to each image for extraction. The masks are similar to convolutional kernels. Haar features are single values obtained from each Haar image mask by subtracting sum of pixels under white areas from sum of pixels under black areas. A threshold value (*e.g.:* - 36) is calculated by trial and error; every edge with magnitude less than 36 is considered a false edge and is set to 0.

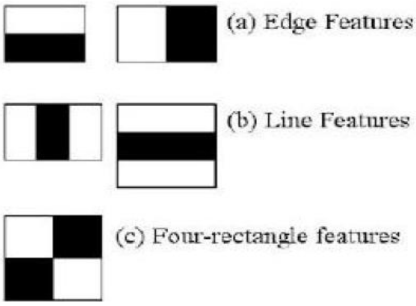


Fig. 9 Haar Image Masks

Now, all possible locations of each kernel of a fixed size, say 4x4, are calculated. All locations where facial features are detected, are marked. Since we are applying all kernels for all facial features, miscalculations are bound to arise. Thus, features with the least error rate are shortlisted, which means they are the features that best classify the face and non-face images. Each image is given an equal weight during the start. Successively, weights of misclassified images are decreased. Again, same process is conducted. New error rates and weights are found. This process is continued until expected accuracy or error rate is achieved or required number of features are obtained [15].

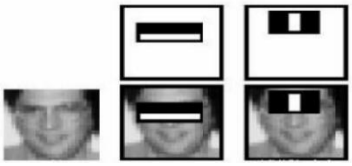


Fig. 10 Extraction of Facial Features

In an image, most of the region is non-face. It is more feasible to have a simpler method to check whether a window is not a face region. If not, it is discarded. Focus is shifted to regions where there is more possibility of a face. For this 'Cascade Classifiers' were introduced. Instead of applying all the 1000 features on a window, group the features into different stages of classifiers and apply one-by-one. If a window fails at first stage, discard it, and don't apply remaining features on it, and continue with the remaining windows.

Fig 11 (a) Before Haar Cascades

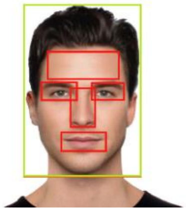
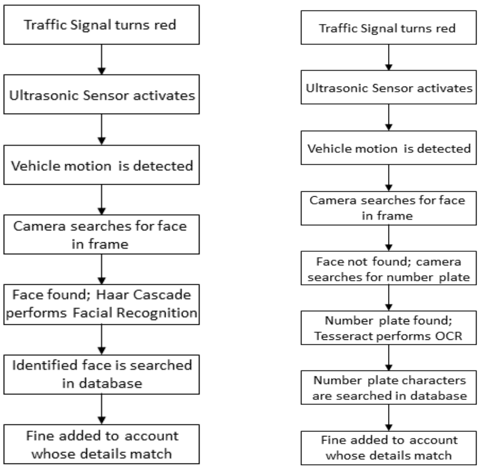


Fig. 11 (b) After Haar Cascades

IV. THE APPROACH


Traffic Signal turns red
Ultrasonic Sensor activates
Vehicle motion is detected
Camera searches for face in frame
Face found; Haar Cascade performs Facial Recognition
Identified face is searched in database
Fine added to account whose details match
Traffic Signal turns red
Ultrasonic Sensor activates
Vehicle motion is detected
Camera searches for face in frame
Face not found; camera searches for number plate
Number plate found; Tesseract performs OCR
Number plate characters are searched in database
Fine added to account whose details match

To begin with, we will have to consider a hypothetical condition in which the traffic signal is always red, or create a dedicated traffic signal algorithm with stipulated durations. Once red, the ultrasonic sensor, positioned at the foot of the traffic signal, gets activated and detects motion of any vehicle beyond the traffic signal pole. Suppose a vehicle is attempting to cross the red signal and its motion is detected, the sensor immediately transmits an alert to the Raspberry Pi, which in turn activates the USB Camera. The camera readily focuses on the vehicle and captures its image.

Raspberry Pi now performs image processing on the captured image look for face. Here, if face is detected then that portion is cropped, and Facial Recognition is performed to compare details related to the obtained face with samples present in the database. 'It is to be noted that if multiple faces are detected in the same image then the algorithm takes into account only the face whose co-ordinates are to the bottom-left most portion of the image.' Then, corresponding amount of fine is debited from account of the person whose details match with that of the obtained face. The final state of the database that stores the details of the drivers along with a separate column showing amounts of fines is displayed.

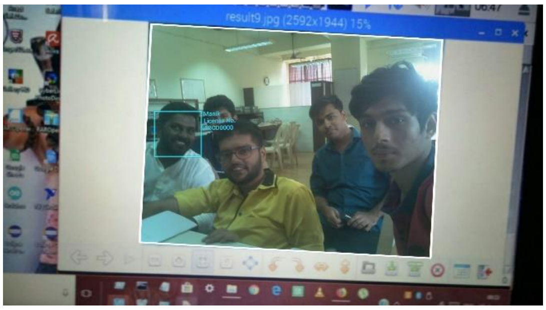


Fig. 13 (a) Driver Detection when Multiple Faces

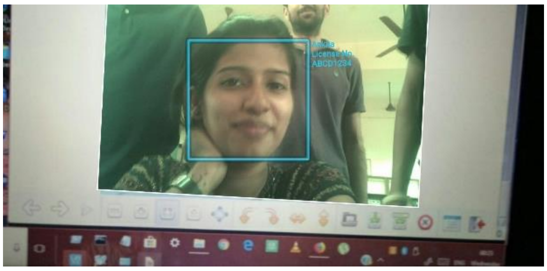


Fig. 13 (b) Driver Detection when Single Face

Nevertheless, if face is not detected in the image due to any of the reasons then Raspberry Pi moves ahead with OCR and performs number plate extraction. Firstly, the image is negated and undergoes Sobel edge detection such that only contours remain. Then, the ROI is obtained, *i.e.*, the number plate is cropped, and character segmentation is performed. As a result, the vehicle number will be procured, which will be searched in the database and profile with exact match will be fined.

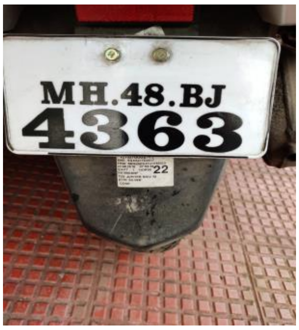


Fig. 14 (a) Before Tesseract

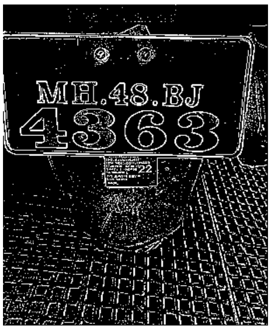


Fig. 14 (b) Negation and Sobel Edge Detection

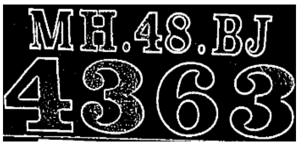
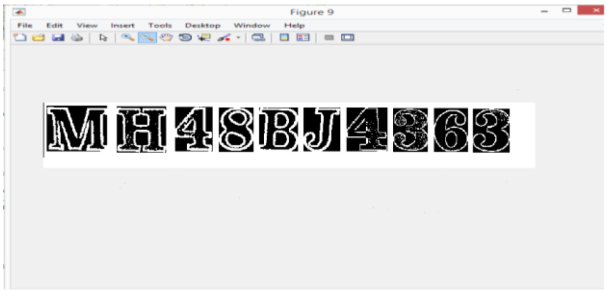


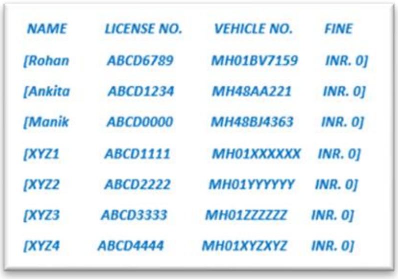
Fig. 14 (c) ROI Extraction



MH48BJ4363

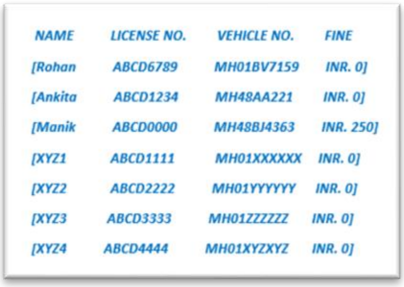
Fig. 14 (d) After Tesseract

The fine column in the database will be updated reflecting the corresponding amount of fine added to the profile whose number plate details match, and final state of the database will be displayed.



NAME	LICENSE NO.	VEHICLE NO.	FINE
[Rohan	ABCD6789	MH01BV7159	INR. 0]
[Ankita	ABCD1234	MH48AA221	INR. 0]
[Manik	ABCD0000	MH48BJ4363	INR. 0]
[XYZ1	ABCD1111	MH01XXXXXX	INR. 0]
[XYZ2	ABCD2222	MH01YYYYYY	INR. 0]
[XYZ3	ABCD3333	MH01ZZZZZZ	INR. 0]
[XYZ4	ABCD4444	MH01XYZXYZ	INR. 0]

Fig. 15 (a) Before Crediting Fine



NAME	LICENSE NO.	VEHICLE NO.	FINE
[Rohan	ABCD6789	MH01BV7159	INR. 0]
[Ankita	ABCD1234	MH48AA221	INR. 0]
[Manik	ABCD0000	MH48BJ4363	INR. 250]
[XYZ1	ABCD1111	MH01XXXXXX	INR. 0]
[XYZ2	ABCD2222	MH01YYYYYY	INR. 0]
[XYZ3	ABCD3333	MH01ZZZZZZ	INR. 0]
[XYZ4	ABCD4444	MH01XYZXYZ	INR. 0]

Fig. 15 (b) After Crediting Fine

V. LIMITATIONS AND REMEDIES

Camera Altitude

Cameras used in the e-Challan system are mounted at the same height as that of a standard P Cantilever traffic signal pole (30 ft. approx. [16]). Zooming on to the face of the driver and capturing its crisp picture would be daunting from that altitude. The proposed system adjusts height of the cameras to 15 feet – comfortable enough to capture face of the driver driving both single-decked, as well as double-decked vehicles, along with not losing line of sight of vehicles on the farther side of the pole.

Face Occluded

There may be several scenarios where camera is unable to capture face of the driver due to: i) face being covered with sunglasses, religious headgear, bandana, stole, or any other form of garment; ii) glare reflecting from glass of the vehicle due to sunlight or any other source of light; iii) moisture or water droplets accumulating on glass of the vehicle; iv) other unforeseen situations (*e.g.:* - a flying food wrapper or newspaper appearing on glass of the vehicle obstructing line of sight of face, etc.). In every such scenario, facial recognition fails as the data of face of the driver is not captured. The system then will be left with no choice but to rely on number plate extraction, and fine the owner instead of the driver. Number plate extraction will serve as a back-up to facial recognition.

Face Data Unavailable

RTO database comprises details of driver’s license of every citizen along with photograph on her license. Since a driver’s license comes with a validity of 20 years [17], it is likely that over that span of time, deformities arise in the facial structure which may render the system impuissant to match the face of the driver correctly with the corresponding face in photograph on her license. To resolve this, citizen’s will be asked to periodically update a ‘profile picture’ on e-Challan portal, and every updated profile picture will be appended to the ‘training set’ – associated with the profile of that citizen – on which the facial recognition algorithm will train itself for more efficient outcomes.

Driver Profile Unavailable

If there is no profile procured for a driver – in other words, the driver doesn’t hold a driving license – then the details of the owner will be fetched by number plate extraction and the owner will be penalized not only for the traffic rule that the driver has violated but also for the offence of driving without license.

Helmets on Two-wheelers

Since facial recognition won’t work with riders wearing helmets on motorbikes and scooters, as of now the only solution for two-wheelers would be to follow the conventional e-Challan procedure of number plate extraction and fining the owner.

VI. FUTURE SCOPE

SUBJECT TO FACIAL RECOGNITION

Automatic Toll Plazas

Facial Recognition can be employed at toll collection checkpoints where cameras would be installed to scan face of the driver crossing a checkpoint. Her details would be fetched from a database, corresponding amount of ticket fare would be deducted from her e-wallet, and an e-receipt for the same would be e-mailed to her along with a notification via SMS.

Smart Petrol Pumps

Fuel dispenser machines at petrol pumps would be interfaced with cameras capable of performing Facial Recognition. The dispensers would calculate charges to be paid by driver subject to the quantity of petrol/diesel/gas consumed, would scan face of the driver to fetch her details, would deduct the charges from her e-wallet, and would dispatch an e-mail and an SMS to her.

Advanced Parking Lots

Traditional paid parking lots can get rid of manual labor with adoption of Facial Recognition technology. During entry, face of the driver would be scanned to fetch her identity and the corresponding timestamp would be recorded. During exit again, face of the driver would be scanned accompanied with corresponding timestamp; parking charges would be calculated based on the difference of the two timestamps, and the amount calculated would be deducted from the driver’s e-wallet with an e-receipt e-mailed and a notification sent via SMS.

SUBJECT TO IMAGE PROCESSING

Smart Signal based on Traffic Congestion

Traffic Signals can be made to alter the durations for which they remain red, green, or yellow, based on density of traffic and crowd of pedestrians. To achieve this, cameras on every traffic signal would be initially fed with an image of an empty road devoid of any vehicles or pedestrians. Following which, the traffic signals would record real-time footage of the vehicles plying and people traversing the same road and compare frames with the initially fed frame of the empty road. Ingenious image processing algorithms would use contour detection and edge detection to calculate density of traffic and number of people and: i) would increase the duration for green signal if road is densely congested with vehicles, but sparsely with pedestrians; ii) would increase the duration for red signal if road is sparsely congested with vehicles, but densely congested with pedestrians; iii) would increase the duration of yellow signal if road is densely congested with both vehicles and pedestrians; and iv) would not revise the durations if the road is sparsely congested with both vehicles and pedestrians.

Variable Intensity Street Lights

Employing the same strategy, cameras mounted on streetlights would be initially fed with image of an empty road. With contour and edge detection techniques, the cameras would identify a subject (vehicle or pedestrian) in the vicinity and increase intensity of the light emitted. Once the camera confirms that subject has crossed the vicinity, light would be dimmed. Image Processing would also aid in varying the intensity of these intelligent streetlights based on the amount of light already present (emanating from other streetlights; focus lights; lights coming from houses, shops, restaurants; sunlight; or moonlight) by observing brightness and saturation of the frames captured. Such intelligent streetlights would automatically dim themselves down at sunrise and light up at sunset. Moreover, network of such intelligent streetlights in proximity would be made to function in co-ordination over specific areas.

Speed Limit Control

Background subtraction is an Image Processing algorithm that would be helpful in identifying vehicles over speeding. Frame Per Second (FPS) rate of cameras installed at checkpoints would be synchronized with the speed limit of that particular checkpoint. A continuous video footage would be recorded and frames with subjects (vehicles) would be undergoing background subtraction. If resulting image would be found to be granulated or over-pixelated, the subject was moving faster than the presided speed limit and would be subject to penalty.

Helmet Monitoring

Image Processing would examine whether a rider on a two-wheeler has donned helmet or not by either contour detection or face detection and would fine the ones not donning helmet by number plate extraction.

VII. APPENDIX

THE ALGORITHM USED

```
import numpy as np
import cv2
face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade=cv2.CascadeClassifier('haarcascade_eye.xml')
Hello = cv2.imread('Rohan.jpg')
Hi = cv2.cvtColor(Hello, cv2.COLOR_BGR2GRAY)
Rect(x, y, w, h).
chehra = face_cascade.detectMultiScale(Hi, 1.3, 5)
for (x, y, w, h) in chehra:
cv2.rectangle(Hello,(x,y),(x+w,y+h),(255,0,0),2)
roi_gray = Hi[y:y+h, x:x+w]
roi_color = Hello[y:y+h, x:x+w]
eyes_detected = eye_cascade.detectMultiScale(roi_gray)
for (ex,ey,ew,eh) in eyes_detected:
cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
cv2.imshow('img',Hello)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

VIII. CONCLUSION

In order to ensure that citizens follow traffic rules in harmony, the Government of India introduced the e-Challan system. This system captures image of number plate of vehicle that violates traffic law, and penalize its owner by fetching her data with the help of the number plate. The system presented rectifies the e-Challan system by employing facial recognition and penalizing the person driving the vehicle, instead of the owner. Although not fool proof, the presented system holds immense potential to revolutionize road transport management and operations that go along with it, in the coming times.

IX. REFERENCES

[1] Government of India Ministry of Statistics and Programme Implementation (MOPSI), Number of Motor Vehicles Registered in India (Taxed and Tax-exempted), Motor Vehicles – Statistical Year Book India 2018, http://mospi.nic.in/statistical-year-book-india/2018/189, 2018.

[2] The Hindustan Times, Mumbai joins list of cities using CCTV to issue e-challan for traffic violations, https://www.hindustantimes.com/mumbai-news/broke-a-traffic-rule-get-ready-to-receive-challan-via-sms/story-VWp4tuGW3RzIL1dRVIogEI.html, October 05, 2016.

[3] Raspberry Pi Org, What is Raspberry Pi, Getting Started, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[4] MaxBotix®, Understanding How Ultrasonic Sensors Work, https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm.

[5] Edmund Optics Worldwide, USB Cameras, https://www.edmundoptics.com/c/usb-cameras/1013/.

[6] Raspbian Org, Bytemark Hosting, Welcome to Raspbian, https://www.raspbian.org/.

[7] Tutorials Point, Image Processing in Python, https://www.tutorialspoint.com/image-processing-in-python, March 27, 2019.

[8] OpenCV Org, About OpenCV, https://opencv.org/about/.

[9] Google Code, The Official Google Code Blog, Announcing Tesseract OCR, http://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html, August 30, 2006.

[10] Wikipedia, The Free Encyclopaedia, Article, Category: Feature Detection (Computer Vision), Haar-like Feature, https://en.wikipedia.org/wiki/Haar-like_feature.

[11] Wikipedia, The Free Encyclopaedia, Article, Category: Virtual Network Computing, Virtual Network Computing, https://en.wikipedia.org/wiki/Virtual_Network_Computing.

[12] S. Wang and H. Lee, "Detection and recognition of license plate characters with different appearances," in Proc. Conf. Intell. Transp. Syst., 2003, vol. 2, pp. 979–984.

[13] X. Shi, W. Zhao, and Y. Shen, Automatic License Plate Recognition Sys- tem Based on Color Image Processing, vol. 3483, O. Gervasi et al., Ed. New York: Springer-Verlag, 2005, pp. 1159–1168.

[14] O. Martinsky, Algorithmic And Mathematical Principles Of Automatic Number Plate Recognition Systems, B.Sc. thesis, Department of Intelligent Systems, Faculty of Information Technology, Brno University of Technology (2007).

[15] Andrew W. Senior and Ruud M. Bolle, IBM T.J.Watson Research Center. Face Recognition and Its Applications.

[16] Traffic Lights India, i Traffico™ R. R. Electronics, Traffic Signal Poles, https://trafficlightsindia.com/product/traffic-signal-poles/ .

[17] Bank Bazaar, Driving Licence, FAQs on Driving Licence, What is the validity period for a Driving Licence in India?, https://www.bankbazaar.com/driving-licence.html.