Personal Driver Assistance System (PDAS) based on Raspberry Pi

Saagar Minocha, Ankur Roychowdhury, Parvez Dittia, Pulkit Leekha

***Abstract*—This research document provides an analysis and evaluation of the current advanced driver assistance systems (ADAS) and the traditional rear-view systems that are in place in vehicles. Advanced driver assistance systems (ADAS) are systems developed to automate/adapt/enhance vehicle systems for safety and better driving. Safety features are designed to avoid collisions and accidents by offering technologies that alert the driver to potential problems, or to avoid collisions by implementing safeguards and taking over control of the vehicle. The traditional cost of such systems is relatively high depending upon the features they incorporate. The product design described in the report aims to overcome the problems of the traditional rear view system along with incorporating cost effective and efficient ADAS features utilizing a camera array (of 2-3 cameras) ,a USB Wi-Fi adaptor, an LCD display and a microcomputer (Raspberry Pi). The microcomputer utilizes image processing and computer vision algorithms to perform several tasks including lane detection, face recognition and rear view assistance (blind spot removal). The product is also capable of displaying current weather report along with weather forecasts for the next 5 days. The product also includes an Emergency SMS service that can be used to send an emergency message including details of current whereabouts to the registered emergency contact. The cost of the product is relatively less when compared to the products in the current ADAS market and boasts of many more features for its price.**

***Index Terms*— Computer Vision, Car Assistance, Image Processing, OpenCV, Raspberry Pi.**

I. Introduction

A long-felt need in the art of vehicle rear-view vision systems is to reduce the amount of time spent gathering information of the condition around the vehicle in order to safely carry out a vehicle manoeuvre such as a turn or a lane change. It has been determined that up to about 50 percent of manoeuvre time is spent gathering information with conventional rear-view mirrors. This typically requires viewing one or more mirrors, turning the head and body to check blind spots, and double-checking the mirrors prior to executing the manoeuvre. Some improvement has been obtained by adjusting mirror optics, for example, to increase the observed field of view rearward of the vehicle. However, this is achieved with an increase in image distortion which makes driver awareness of conditions to the side and rear of the vehicle even more difficult. To overcome the problem of constantly checking the mirrors, a single field of view (providing complete rear view of the vehicle) at a convenient eye-level for the driver has long been desired. This product aims to provide a vehicle vision system that includes a plurality of cameras having respective fields of view exterior of the vehicle. A processor is operable to process image data captured by the cameras and to generate images of the environment surrounding the vehicle. The processor is operable to generate a rear view feed of the vehicle along with several additional functions that use cameras in the front as well as on the back of the vehicle.

The additional functions that the product is capable of performing include lane departure system, face recognition, current weather report with a 5 day weather forecast and an Emergency SMS service. A display screen (TFT) is operable to display the generated feeds of the different cameras and a button based system is provided to aid in switching between the functions performed by the device .A rear-view vision system for a vehicle includes at least one image capture device directed rearward with respect to the direction of travel of the vehicle. A display system displays an image synthesized from output of the image capture device. The display system is preferably contiguous with the forward field of view of the vehicle driver at a focal length that is forward of the vehicle passenger compartment. A plurality of image capture devices may be provided and the display system displays a unitary image synthesized from outputs of the image capture devices which approximates a rearward-facing view from a single location, such as forward of the vehicle.

With this innovation we would be able to provide the driver with a clear and unobstructed view of the scenario around him/her thereby eliminating the possibility of blind spots, removing the need to view the different mirrors constantly and improving the overall experience and interaction of the driver with the vehicle. The additional functions provide an enhanced driving experience and mirrors an advanced driver assistance system. The product is made portable enough to support "non-smart cars" and is cost effective. The current product can further be boosted with a better processor which in turn can be used to incorporate a number of other unique features including drowsiness detection, car diagnostic information, travel environment conditions etc.

Manuscript received April 22, 2015.

Saagar Minocha, Computer Science and Engineering, Bharati Vidyapeeth's College Of Engineering, New Delhi, India, 9650137374,).

Ankur Roychowdhury, Computer Science and Engineering, Bharati Vidyapeeth's College Of Engineering, New Delhi, India, 9999980842,

Parvez Dittia, Computer Science and Engineering, Bharati Vidyapeeth's College Of Engineering, New Delhi, India, 9654074170,

Pulkit Leekha, Computer Science and Engineering, Bharati Vidyapeeth's College Of Engineering, New Delhi, India, 9910992477.

II. Background

A. *Computer Vision*

Computer vision would be our main category of work as it includes methods for acquiring, processing, analysing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models to the construction of computer vision systems. Sub-domains of computer vision include scene reconstruction, event detection, video tracking, object recognition, learning, indexing, motion estimation, and image restoration.

B. *Human Computer Interaction*

Human-computer interaction (HCI) is the study of how people design, implement, and use interactive computer systems and how computers affect individuals, organizations, and society. This encompasses not only ease of use but also new interaction techniques for supporting user tasks, providing better access to information, and creating more powerful forms of communication. It involves input and output devices and the interaction techniques that use them; how information is presented and requested; how the computer's actions are controlled and monitored; all forms of help, documentation, and training; the tools used to design, build, test, and evaluate user interfaces; and the processes that developers follow when creating interfaces. HCI in the large is an interdisciplinary area. It is emerging as a speciality concern within several disciplines, each with different emphases: computer science (application design and engineering of human interfaces), psychology (the application of theories of cognitive processes and the empirical analysis of user behaviour), sociology and anthropology (interactions between technology, work, and organization), and industrial design (interactive products).

III. Structure And Principle

The core system consists of 3 USB webcams connected to the USB ports of a Raspberry Pi development board (B+ Model).[1] A 7inch TFT display screen is connected to the 4 pole 3.5 mm jack on the Raspberry Pi board via a 4 pole audio/video cable. A push button system based (on a breadboard) designed specifically for controlling the system is connected to the GPIO pins via jumper cables. The button system consists of 5 push buttons (4 pin) performing the functions of switching to rear view, switching to lane detection program ,switching to weather display , Emergency SMS service and system shutdown. A USB Wi-Fi adaptor responsible for providing internet connectivity for the above mentioned features. When the system is in action, the TFT screen is connected to the power source (usually cigarette lighter car charger for cars) and the Raspberry Pi is powered using a micro USB power supply

(usually with the help of a power bank).

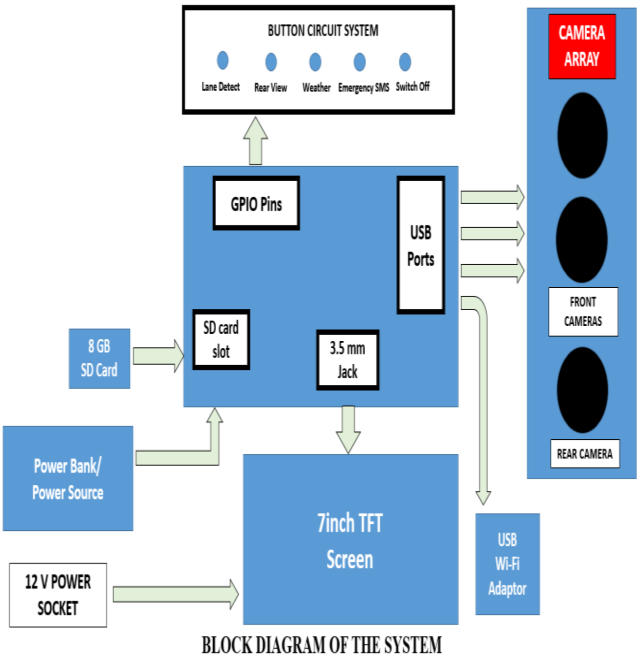


Fig 1: Block Diagram of the System

IV. Hardware Design

A. *Raspberry Pi*

The Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. The introductory price for the B+ model is around US$35.The following are the specifications of the model [1]:

- Chip : Broadcom BCM2835 SoC (System On Chip)
- Core Arch. : ARM11
- CPU : 700 MHz (can be overclocked)
- Memory : 512MB SDRAM
- Power : Micro USB socket 5V, 2A
- Video O/P : HDMI , Composite RCA (PAL and NTSC)
- USB : 4 x USB 2.0 Connector
- GPIO : 40-pin expansion header
- Camera : 15-pin MIPI Camera Serial Interface

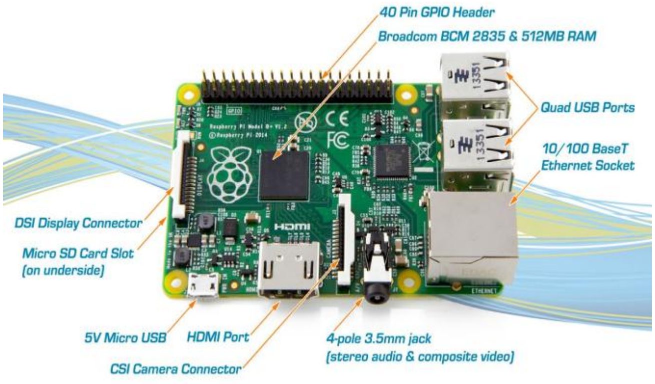

40 Pin GPIO Header
Broadcom BCM 2835 & 512MB RAM
Quad USB Ports
10/100 BaseT Ethernet Socket
DSI Display Connector
Micro SD Card Slot (on underside)
5V Micro USB
HDMI Port
CSI Camera Connector
4-pole 3.5mm jack (stereo audio & composite video)

Fig 2: Raspberry Pi B+ Model

B. *Camera and Display Module*

The camera module consists of 3 USB webcams. 2 cameras are placed in front of the car and perform the driver face recognition and lane detection functions. The camera at the back of the car is responsible for rear view assistance system. The processed image feed from the cameras is displayed on a

7inch TFT screen for ease of view of the driver. The image feeds from different cameras can be switched with the help of the button system.

C. *Button System via GPIO pins*

General Purpose Input/Output (a.k.a. GPIO) is a generic pin on the chip whose behaviour (including whether it is an input or output pin) can be controlled (programmed) through software. The Raspberry Pi allows peripherals and expansion boards to access the CPU by exposing the inputs and outputs. The button system consists of 5 4-pin push buttons connected to 5 unique GPIO pins and the RPi. GPIO library is used in the Python script to detect the button press. The buttons are connected via a breadboard using jumper wires. [2]

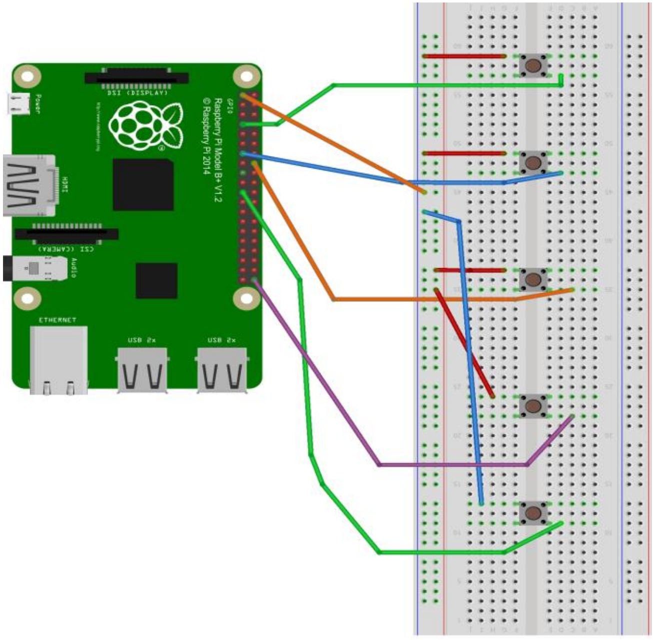


Fig 3: The Button System circuit

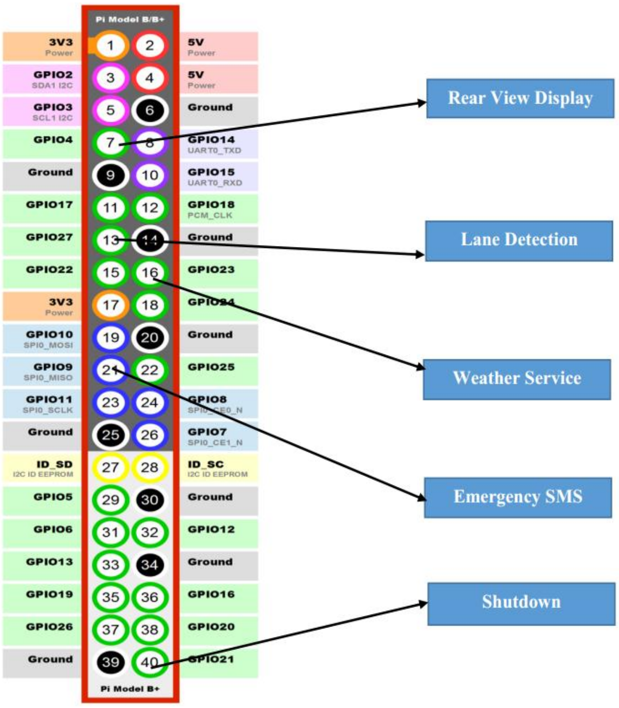


Fig 4: GPIO pin layout

V. Software Design

A. *Raspbian OS*

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. It comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation. Features of Raspbian include:

- Created specially for RPi hardware
- Supports hard float ABI
- Desktop LXDE
- Easy to use, install and customize

B. *OpenCV*

OpenCV is the most popular and advanced code library for Computer Vision related applications today, spanning from many very basic tasks (capture and pre-processing of image data) to high-level algorithms (feature extraction, motion tracking, machine learning). It is free software and provides a rich API in C, C++, Java and Python. The library itself is platform-independent and often used for real-time image processing and computer vision. OpenCV is released under a BSD license and hence it’s free for both academic and commercial use.

C. *Software Modules of the system*

Face Recognition Module

The demo model boots up to the Face Recognition Process and after face recognition of the driver responds with a welcome message. The device then automatically switches to the rear- view process. The algorithm used for face recognition is Fisherfaces. The demo model has been configured to detect the faces of the team members through training with 50-60 sample images of the members, respectively. Thus, each member’s face is detected and correspondingly his name is displayed along with the detection rectangle. A welcome message specific to the member is generated via vocal response.

In this product, we use the Fisherfaces approach to face recognition as it appears to be the best at extrapolating and interpolating over variation in lighting. It also appears to be the best at simultaneously handling variation in lighting and expression. Taking a pattern classification approach in this method, each pixel in an image is considered as a coordinate in a high-dimensional space. It takes advantage of the observation that the images of a particular face, under varying illumination but fixed pose, lie in a 3D linear subspace of the high dimensional image space if the face is a Lambertian surface without shadowing. However, since faces are not truly Lambertian surfaces and do indeed produce self-shadowing, images will deviate from this linear subspace. Rather than explicitly modelling this deviation, it linearly projects the image into a subspace in a manner which discounts those regions of the face with large deviation. The projection method is based on Fisher’s Linear Discriminant and produces well separated classes in a low-dimensional subspace, even under severe variation in lighting and facial expressions. The Eigenface technique, another method based on linearly projecting the image space to a low dimensional subspace, has similar computational requirements. Yet, extensive experimental results demonstrate that the “Fisherface” method has error rates that are lower than those of the Eigenface technique for tests on the Harvard and Yale Face Databases.

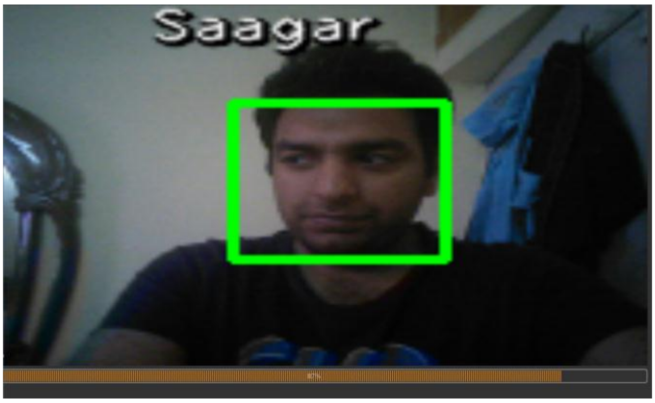


Fig 5: Face Recognition module in action

Rear-view Module

The Rear-view process is switched to automatically after the face recognition process is completed and can be switched to from the lane detection process using the specified push button. The VideoCapture class in OpenCV is used to capture the video from the camera placed at the rear end of the vehicle.

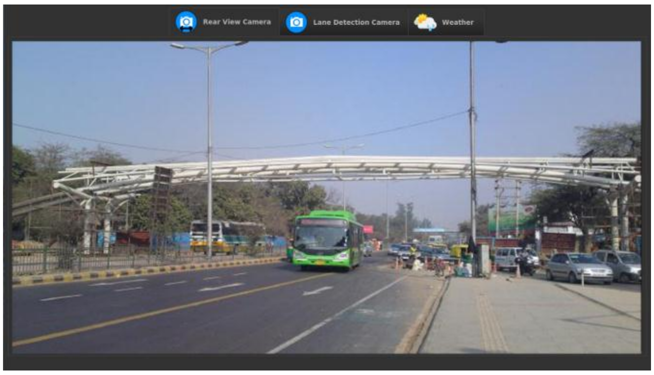


Fig 6: Rear View module in action

Lane Detection Module

The algorithm used for the lane detection process is the widely used Hough Line Transform. The Hough Line Transform is a transform used to detect straight lines. To apply the Transform, first an edge detection pre-processing is desirable. Using Hough Transform a line can be detected by finding the number of intersections between curves. The more curves intersecting means that the line represented by that intersection have more points. In general, we can define a threshold of the minimum number of intersections needed to detect a line. Hough Line Transform keeps track of the intersection between curves of every point in the image. If the number of intersections is above some threshold, then it declares it as a line with the parameters of the intersection point.

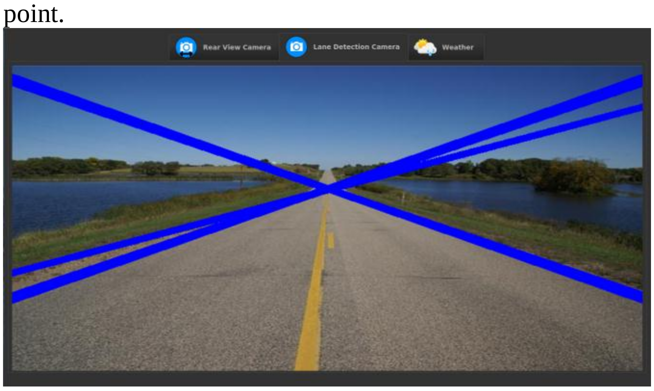
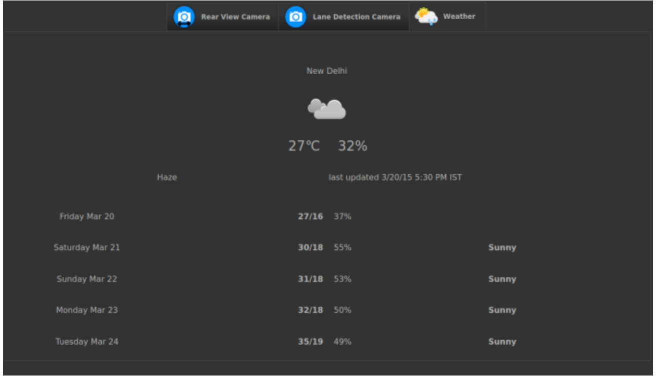


Fig 7: Lane Detection module in action

Weather Service Module

The weather service is provided via pywapi module which provides a python wrapper around the Yahoo! Weather, Weather.com and NOAA APIs. The pywapi version used is 0.3.8. A location_id variable is used to provide a 5 digit US zip code or location ID. To find the location ID, a function get_location_ids (search_string) is used. The location ID INXX096 for New Delhi is passed onto the server via the API function get_weather_from_weather_com .the results include a dictionary of weather data for the current day along with forecasts of the next 5 days. The fields included are current conditions, temperature (min/max), humidity (in %) and last updated.



```
                          New Delhi

                        27°C   32%

        Haze                  last updated 3/20/15 5:30 PM IST

Friday Mar 20           27/16  37%
Saturday Mar 21         30/18  55%                Sunny
Sunday Mar 22           31/18  53%                Sunny
Monday Mar 23           32/18  50%                Sunny
Tuesday Mar 24          35/19  49%                Sunny
```

Fig 8: Weather Service module in action

Emergency SMS Module

The emergency SMS service sends the location of the car to the registered emergency contact and thus, helps in quick action in times of emergency. The BlueNMEA application fetches the location information using the mobile phone's GPS sensor. The location information is then sent to Raspberry Pi in NMEA (National Marine Electronics Association) format. The Raspberry Pi acts as a GPS receiver in the process with gpsd server daemon listening for particular IP address and port and sending the JSON object stream to the gpsd client program (gpspipe). This information is then reverse geocoded using the Google Maps API and the address is generated in readable form. This address is sent in the form of an SMS to the emergency contact using the Twilio API.

The important components of the Emergency SMS process are as follows:

1. BlueNMEA: An Android application which sends location data over Bluetooth (RFCOMM) or TCP in the NMEA format.
2. gpsd: A service daemon that monitors one or more GPSes or AIS receivers attached to a host computer through serial or USB ports. It manages a collection of sensors and makes reports from all of them available as a JSON object stream on a well-known TCP/IP port.
3. gpspipe: A tool to connect to gpsd and output the received sentences to stdout. This makes the program useful as a pipe from gpsd to another program or file.
4. Reverse Geocoding: The process of back (reverse) coding of a point location (latitude, longitude) to a readable address or place name. This permits the identification of nearby street addresses, places, and/or areal subdivisions such as neighbourhoods, county, state, or country.
5. Twilio SMS Service: SMS API service which allows developers to programmatically make and receive phone calls and send and receive text messages using its web service APIs.

VI. Flow Chart: Design flow of the system

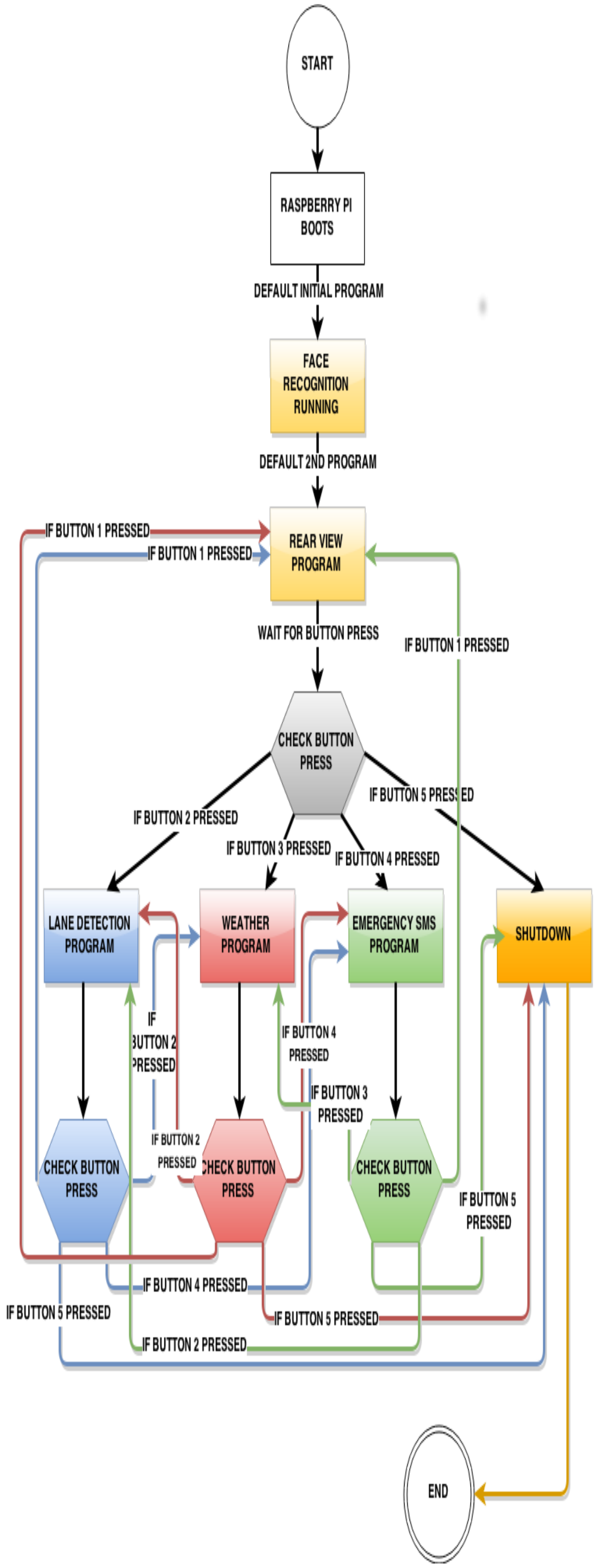


Fig 9: Design Flow Diagram of the system

VII. Cost and Feature Analysis

The following table compares the cost and features of already existing driver assistance systems with proposed Personal Driver Assistance System.

I. Cost/Feature Analysis

SYSTEM NAME	FEATURES (General)
Traditional Rear View Mirror COST: Rs. 500	• Consists of side view mirrors and central rear view mirror • Regular rear view vision • Basic features of a mirror
Rear View Display System COST: Rs. 10000	• Single camera installed to provide the feed of the car's rear. No need to view 3 different mirrors • Camera was set up behind the rear bumper which was the ideal location to keep safe. • The screen was put upfront over the glove compartment to provide a clear view to the person.
Rear mounted radar with view inbuilt. COST: Rs. 5000	• Use of screen was refrained but a small screen was installed behind the rear view mirror itself. • Only activated in reverse gear. • Helpful for parking purposes
Night vision with pedestrian crossing: COST: Rs. 15000	• It pinpoints pedestrians, highlighting them on a dashboard display. The system is incorporated with a pedestrian identifier that also shows the direction the pedestrian is moving. • Expensive • Not commercially popular
Personal Driver Assistant System COST: Rs. 9000	• Conventional rear view display system. • Driver's face recognition system for security. • Front lane detection for safety. • Many more features to be introduced.

VIII. Conclusion

In this paper, a cost effective and efficient prototype for a personal driver assistance system is created and proposed as an alternative for the traditional rear view systems as well as existing carputers owing to its superior features against its cost. The prototype was successfully constructed and tested on a standard car.

Acknowledgment

We take this opportunity to extend our sincere thanks to Dr. Dharmender Saini for offering us the opportunity to work under his guidance on this unique and innovative project. We are deeply indebted to him for guiding us throughout the project and providing valuable inputs and insights. His knowledge and experience served as a continuous source of support for us. We would also like to take this occasion to whole heartedly thank Dr. Bindu Garg (Head of Department (CSE)). Without her invaluable support, guidance and constructive criticism, this project would not have been possible. We would like to express our sincere gratitude towards the college administration for financially helping us by bearing a percentage of the cost of the project. Finally, we wish to thank the Computer Science and Engineering faculty at Bharati Vidyapeeth's College OF Engineering, New Delhi for the valuable education they have imparted us over the course duration.

References

[1] Raspberry Pi Documentation, *http://www.raspberrypi.org/documentation/*

[2] Physical computing with Raspberry Pi, *https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/buttons_and_switches/*

[3] Raspberry Pi Forums, *http://www.raspberrypi.org/forums/*

[4] Face recognition, *https://github.com/bytefish/facerec*, September 20 2014

[5] Lane detection, *https://github.com/funningboy/carCV,* Feb 6 2014

[6] Getting started with Videos, *http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_gui/py_video_display/py_video_display.html*

[7] Face detection using Haar Cascades, *http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html*

[8] George C. Schedivy, US Patent 7564479 B2, 2009, *Rear view camera display mounted on a vehicle overhead console.*

[9] Evan Solida, US Patent 7564479 B2, 2014, *rear-view display system for a bicycle.*

[10] Jan-Carsten Becker, US 36562A1,2010, *Driver Assistance System*

[11] Georg, Pflug, US Patent 0333729A1, 2014, *Vehicle Vision Display System*

[12] Micheal J. Baur, US Patent 0333770A1,2014, *Rear View Vision System for Vehicle*

[13] Eugene Kaznacheev, Joshua Tasker, *"Python Weather API"*, *https://code.google.com/p/python-weather-api/*

[14] Max Kellermann, *"BlueNMEA"*, *http://max.kellermann.name/projects/blue-nmea/*

[15] gpsd documentation, *http://www.catb.org/gpsd/index.html#documentation*

[16] gpspipe, *http://thomasloughlin.com/gpspipe-gps-client/*

[17] Reverse Geocoding, *https://developers.google.com/maps/documentation/javascript/examples/geocoding-reverse*

[18] Twilio API documentation, *https://www.twilio.com/docs*

[19] Prashant Roy, Ankit Khetrapal, Sandeep Rathore , *"Driver Assistance System"*, International Journal Of Engineering Development And Research , 2014

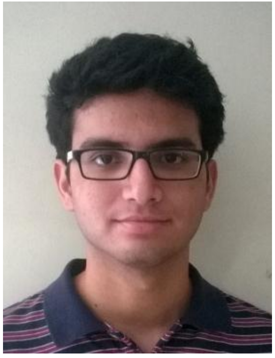
[20] Narayan Pandharinath Pawar, Minakshee M.Patil, *"Intelligent Driver Assistance System For Automotive"*,International Journal of Research in Engineering and Technology , May 2014

[21] Narayan Pandharinath Pawar, Minakshee M.Patil, "*Driver Assistance System based on Raspberry Pi*", International Journal of Computer Applications ,June 2014

[22] Thomas Hummel, Matthias Kühn, Jenö Bende, Antje Lang, "*Advanced Driver Assistance Systems*", August 2011

Saagar Minocha, Computer Science and Engineering, Bharati Vidyapeeth's College Of Engineering, New Delhi, India.

Ankur Roychowdhury, Computer Science and Engineering, Bharati Vidyapeeth's College Of Engineering, New Delhi, India.



Parvez Dittia, Computer Science and Engineering, Bharati Vidyapeeth's College Of Engineering, New Delhi, India.

Pulkit Leekha, Computer Science and Engineering, Bharati Vidyapeeth's College Of Engineering, New Delhi, India.