"Automatic Number Plate Recognition for use in Different Countries using Raspberry pi" (Real Time Approach)

Radhika S. Dangare, Prof. G. D. Dalvi

Abstract: ***The paper aims at designing a system which captures the image of the number plate automatically of a vehicle and these details were verified using Raspberry Pi processor for authentication. The system also alerts the authorities when any unauthorized image of number plate is detected using buzzer alarm system. Automation is the most frequently spelled term in the field of electronics. The hunger for automation brought many revolutions in the existing technologies. This paper makes use of an onboard computer, which is commonly termed as Raspberry Pi processor. It acts as heart of the project. This onboard computer can efficiently communicate with the output and input modules which are being used. The Raspberry Pi is a creditcard-sized single-board computer developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZFS 700 MHz processor, Video Core IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage. The device which is able to perform the task is a Raspberry Pi processor. When any vehicle passes by the system, the image of the number plate of every vehicle is captured using camera. The image of the number plate details are fed as input to the Raspberry Pi processor. The Processor takes responsibility to check the authentication details of every vehicle. Once the vehicle details are recognized then the processor operates the gate using stepper motor. The system also alerts the user through buzzer alarm whenever it detects an unauthorized image of number plate was detected. To perform this task, Raspberry Pi processor is programmed using embedded 'Linux'.***

Keywords: soc, Raspberry Pi,Linux

1. Introduction

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 Raspberry Pi is manufactured through licensed manufacturing deals with Newark element14 (Premier Farnell), RS Components and Ego man. All of these companies sell the Raspberry Pi online. Ego man produces a version for distribution solely in China and Taiwan, which can be distinguished from other Pi's by their red coloring and lack of FCC/CE marks. The hardware is the same across all manufacturers. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor (The firmware includes a number of "Turbo" modes so that the user can attempt over clocking, up to 1 GHz, without affecting the warranty), VideoCore IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage. The Raspberry Pi does not come with a realtime clock, so an OS must use a network time server, or ask the user for time information at boot time to get access to time and date for file time and date stamping. However, a real-time clock (such as the DS1307) with battery backup can be added via the I²C interface.

2. Project Objectives

Usage of image authentication technology.

- Capturing of Vehicle number plate details
- Using camera. Unauthorized authentication and alerting
- Through buzzer alarm. Number plate recognition indication even
- Through LED indicators.

3. Problem Definition

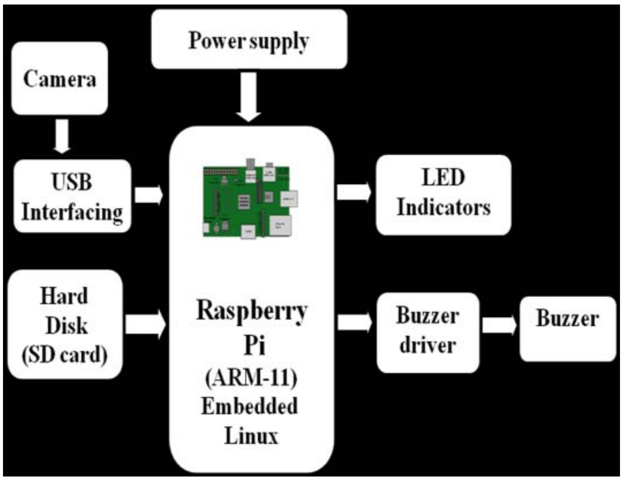
In this paper [1] there is no mention of usage of microcontroller for detection of images. As security plays a major role, authenticating of number plate is required. This project includes authentication using android mobile with special application developed in it. Edge based algorithm cannot reduce noise to overcome this problem Neural Network Technique is used. Different fonts in number plate is not mentioned. Experiments are made to verify different font sizes.

4. Specifications of the Project

4.1 Project Description

The paper aims at designing a system which automatically captures the image of the number plate of a vehicle and these details were verified using Raspberry Pi processor for authentication. The system also alerts the authorities when any unauthorized image of number plate was detected using buzzer alarm system. Automation is the most frequently spelled term in the field of electronics. This paper makes use of an onboard computer, which is commonly termed as Raspberry Pi processor. The onboard computer can efficiently communicate with the output and input modules which are being used. The Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, Video Core IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage. The device which is able to perform the task is a Raspberry Pi processor. When any vehicle passes by the system, the image of the number plate of every vehicle is captured using camera. The image of the number plate details are fed as input to the Raspberry Pi processor. The Processor takes responsibility to check the authentication details of every vehicle. Once the vehicle details are recognized then the processor operates the gate using stepper motor. The system also alerts the user through buzzer alarm whenever it detects an unauthorized image of number plate was detected. To perform this task, Raspberry Pi processor is programmed using embedded 'Linux'

4.2 Block diagram Overview



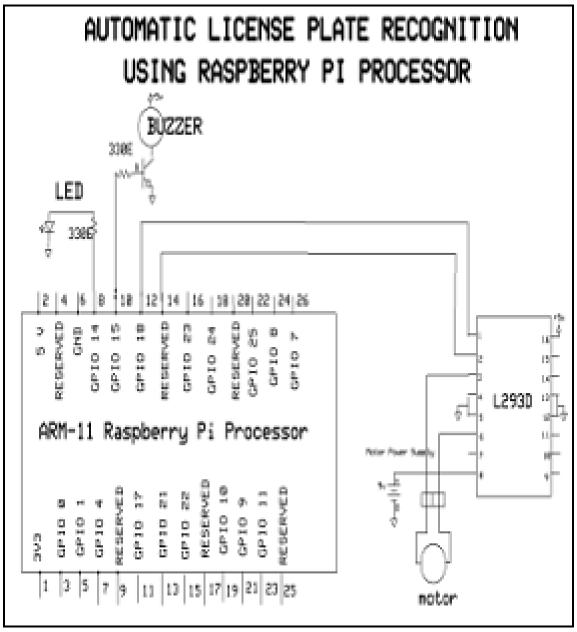
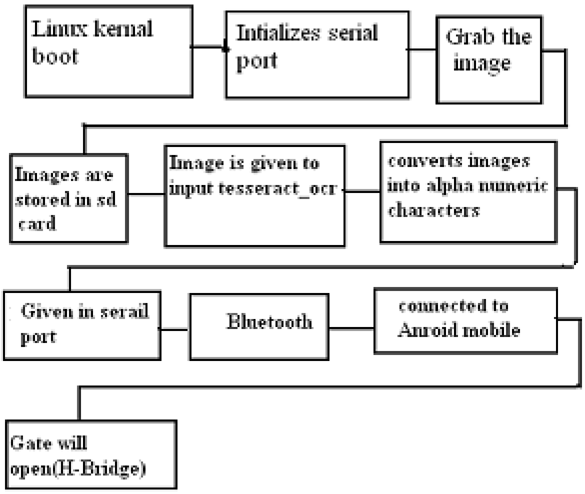


Figure 1: Architectural description of Raspberry pi 6.1

Description: Arm-11 processor consists of 26 GPIO pins. Among which we are using 3 pins are used for led, buzzer and motor. For running motor the driver used is L293D.It has 16 pins.3rd and 6th pin are used to rotate motor in clockwise and anticlockwise.4th and 5th pins are grounded. Process Flow/ Flowchart explaining how gate operates.


Linux kernal boot
Intializes serial port
Grab the image
Images are stored in sd card
Image is given to input tesseract_ocr
converts images into alpha numeric characters
Given in serail port
Bluetooth
connected to Anroid mobile
Gate will open(H-Bridge)

5. Design and Implementation

5.1 Hardware Tools

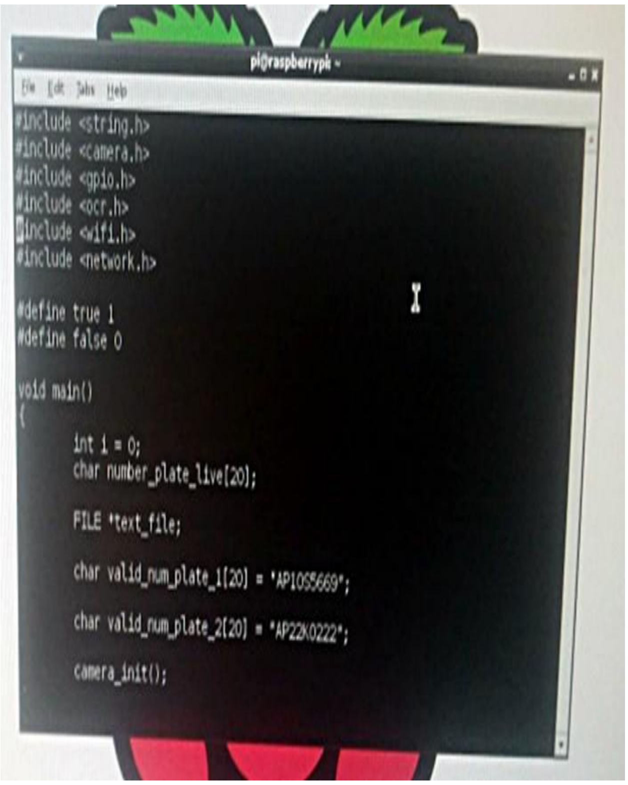
The ARM1176JZF-S processor incorporates an integer core that implements the ARM11 ARM architecture v6. It supports the ARM and Thumb™ instruction sets, Jazelle technology to enable direct execution of Java bytecodes, and a range of SIMD DSP instructions that operate on 16-bit or 8-bit data values in 32-bit registers. The ARM1176™ applications processors deployed broadly in devices ranging from smart phones to digital TV's to eReaders, delivering media and browser performance, a secure computing environment, and performance up to 1GHz in low cost designs. The ARM1176JZ-S processor features ARM TrustZone technology for secure applications and ARM Jazelle technology for efficient embedded Java execution. Optional tightly coupled memories simplify ARM9™ processor migration and real-time design, while AMBA 3 AXITM interfaces improve memory bus performance. DVFS support enables power optimization below the best-in-class nominal static and dynamic power of the ARM11TM processor architecture

Figure 2: Overview of Raspberry pi

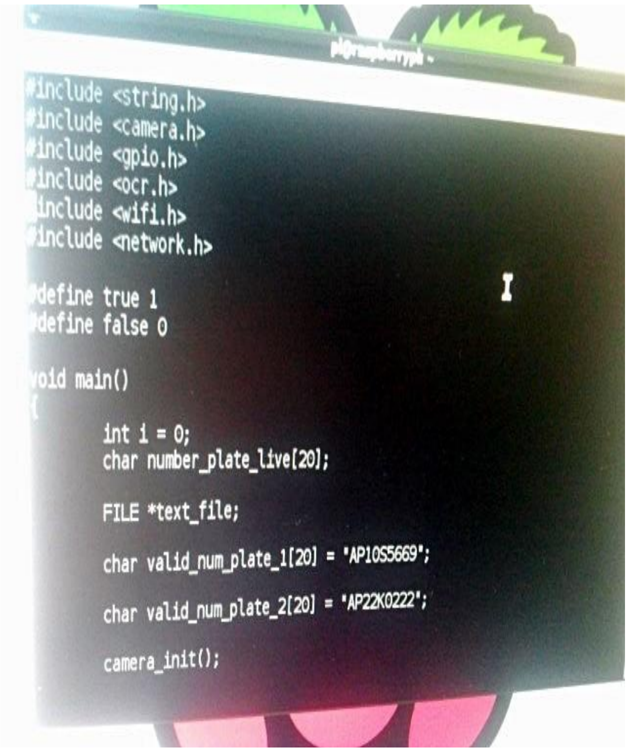
6. Overall Features

Processor:700MHZARM11 Ram:512MB SRam:400MHZ Operatingvoltage : 5V Video Connections : 1HDMI Audio : Stereo over HDMI OS : Raspbian, Anroid Power : 150-350 GPIO Capability : 8 pins Memory : 32 kb for level 1 cache 128kb for level2cache

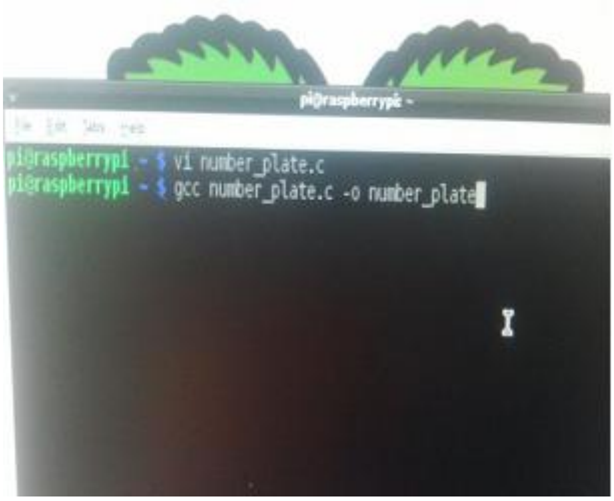
7. Results



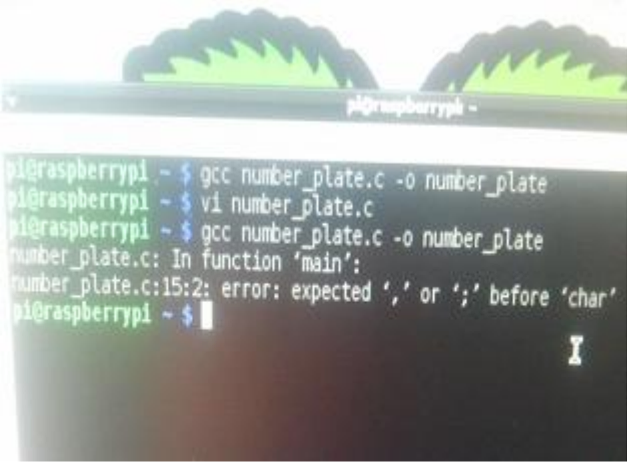
```
#include <string.h>
#include <camera.h>
#include <gpio.h>
#include <ocr.h>
#include <wifi.h>
#include <network.h>

#define true 1
#define false 0

void main()
{
        int i = 0;
        char number_plate_live[20];

        FILE *text_file;

        char valid_num_plate_1[20] = "AP10S5669";

        char valid_num_plate_2[20] = "AP22K0222";

        camera_init();
```



```
#include <string.h>
#include <camera.h>
#include <gpio.h>
#include <ocr.h>
#include <wifi.h>
#include <network.h>

#define true 1
#define false 0

void main()
{
        int i = 0;
        char number_plate_live[20];

        FILE *text_file;

        char valid_num_plate_1[20] = "AP10S5669";

        char valid_num_plate_2[20] = "AP22K0222";

        camera_init();
```

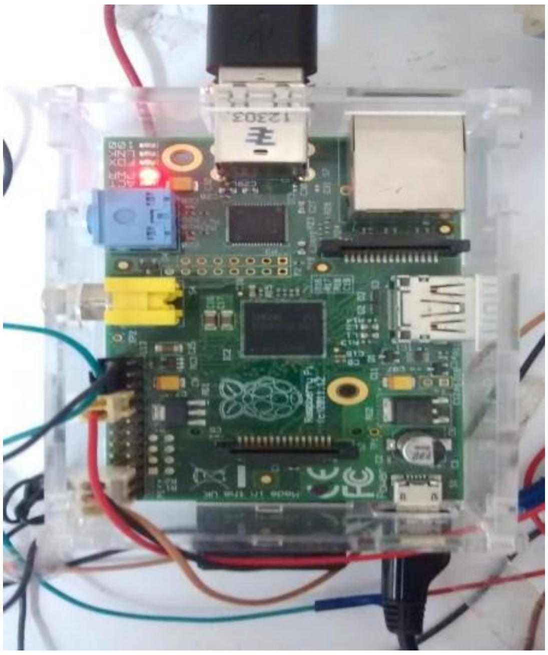


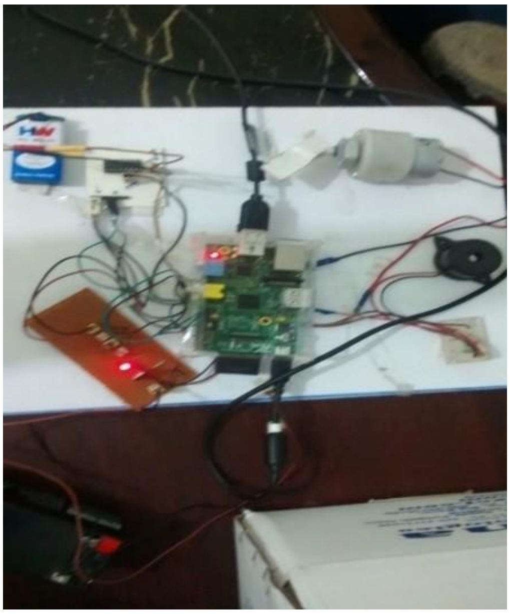
```
pi@raspberrypi ~ $ vi number_plate.c
pi@raspberrypi ~ $ gcc number_plate.c -o number_plate
```

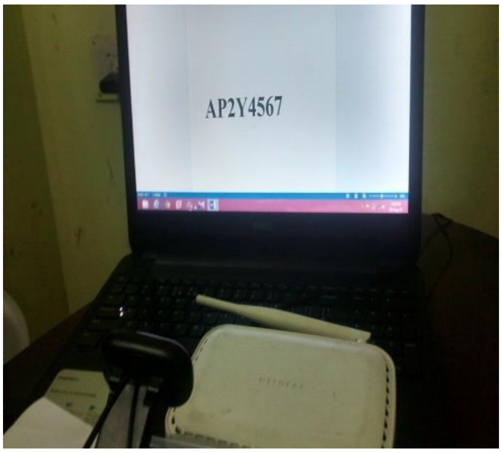


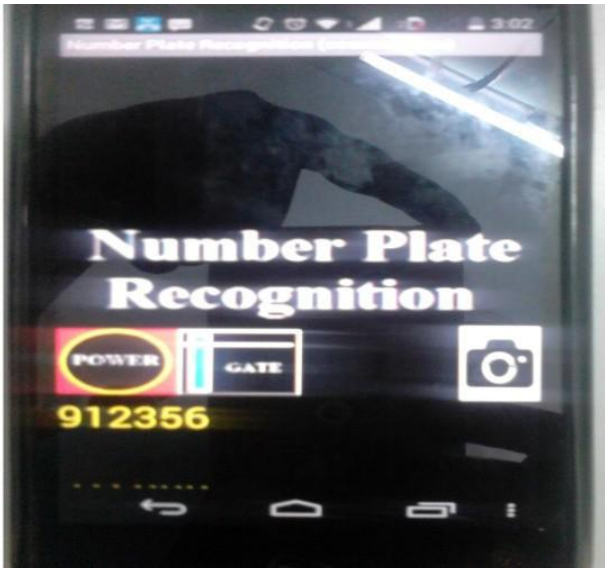
```
pi@raspberrypi ~ $ gcc number_plate.c -o number_plate
pi@raspberrypi ~ $ vi number_plate.c
pi@raspberrypi ~ $ gcc number_plate.c -o number_plate
number_plate.c: In function 'main':
number_plate.c:15:2: error: expected ',' or ';' before 'char'
pi@raspberrypi ~ $
```

8. Experimental Setup



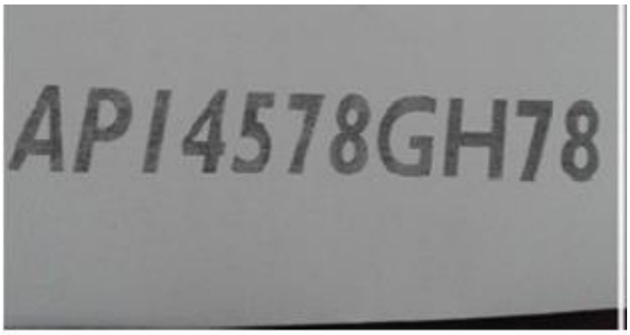






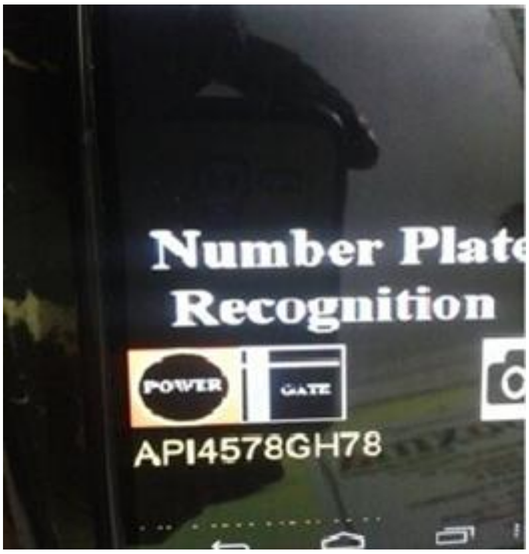
9. Results

Case 1: Font type: calibri Font size: 45



API4578GH78

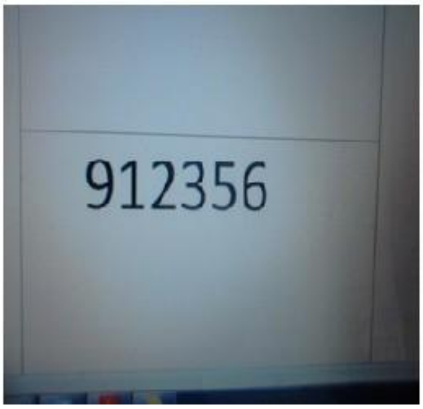
a) Input image



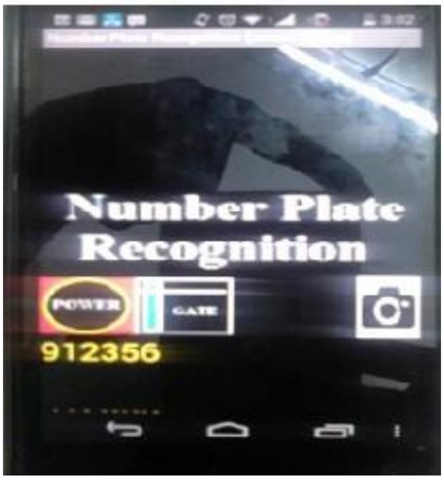
API4578GH78

b) Received image on android application

Case 2: Font type: Times new roman Font size: 30

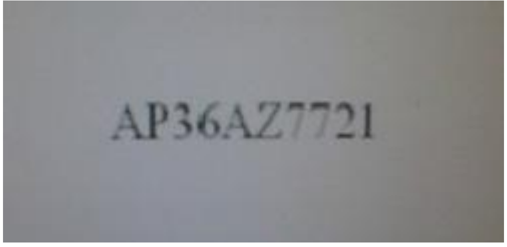


a) Input Image

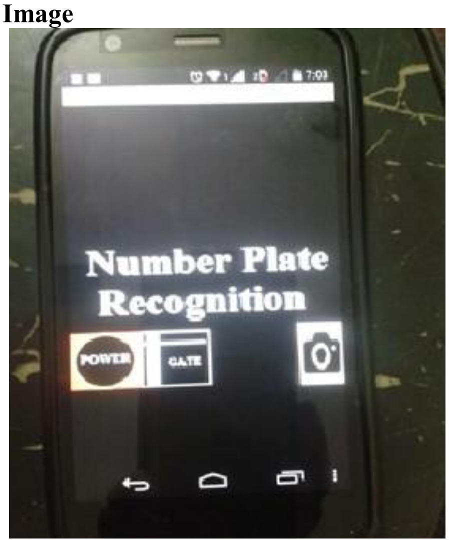


b) Received image on users android application
Output received refers to the input image

Case 3: Font type: Arial Font size: 30



a) Input Image



b) Failure in reception
Output image shows a failure of reception due to fading in the input image.

10. Conclusions and Future Scope

10.1 Conclusion

The paper was designed a system which captures the image of the number plate of a vehicle and these details of number plate were verified with the predefined details using Raspberry Pi processor for authentication. The system alerts the authorities when any unauthorized image of number plate was detected through camera using buzzer alarm system. License plate extraction method is designed for real-time Indian license plate extraction. Input to the system is an image which contains the license plate, acquired by a digital camera of the front or rear of the vehicle; and its output is the license plate region. The method comprises the following major stages, which are: RGB to gray-scale conversion, vertical edge detection and image linearization, analysis and dilation, vertical projection and thresholding, extracting the accurate location of the license plate, filtration and image enhancement, linearization and smoothing process, and Character segmentation for horizontal and vertical. Software tool we are using as Linux operating system. Linux . Linux Operating System has primarily three components. Kernel, System Library System Utility . Linux is an open-source Operating System. People can change codes and add programs to Linux OS which will help use your computer better. Linux is customizable but Windows is not. Linux has high security.

Results has been verified by using Raspberry pi ,using an application on Android mobile. After camera recognizes the number plate the information is sent to user through wifi module. User checks and authenticates the system. If there is any unauthorized user buzzer will on automatically. Experimental results shows the proposed system using Raspberry pi can authenticate the system successfully.

10.2 Future Work

This paper can be extended using high efficiency GPS receiver and a GPRS module. The GPRS module gives the intimation of the vehicle racking directly on to the predefined web link for tracking the vehicle on Google maps. The project can be extended using USB camera for vehicle monitoring from longer distances. The project can be extended using memory card using which the traveled path can be stored which helps in storing the tracked path along with speed and time.

References

[1] T. L. Chien, H. Guo, K.L. Su and S.V. Shiau, "Develop a Multiple Interface Based Fire Fighting Robot," IEEE International Conference on Mechatronics, May 2007.

[2] K. L. Su, "Automatic Fire Detection System Using Adaptive Fusion Algorithm for Fire Fighting Robot," IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, Oct. 2006.

[3] T. L. Chien, H. Guo, K.L. Su and S.V. Shiau, "Develop a Multiple Interface Based Fire Fighting Robot," IEEE International Conference on Mechatronics, May 2007.

[4] J.H. Park, B.W. Kim, D.J. Park and M.J. Kim, "A system architecture of wireless communication for fire-fighting robots," Proceedings of the 17th World Congress The International Federation of Automatic Control, July 2008.

[5] Benjamin C. Kuo, Step Motors and Control Systems, SRL Publishing Company, Champagne, IL, 1979.