

Low Latency Image Processing of Transportation System Using Parallel Processing coincident Multi-threading (PPcM)

Supervisor: Dr. Amitabha Chakrabarty

Co-Supervisor: Md. Saiful Islam

Submitted by:

Sanjana Khan Shammi-13301042

Samia Sultana- 13301108

Mashrura Tasbih-12301009



Department of Computer Science & Engineering, BRAC University

Submitted on: 18th April, 2017

Declaration

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree

Signature of Supervisor

Dr. Amitabha Chakrabarty

Signature of Author

Sanjana Khan Shammi

Signature of Author

Samia Sultana

Signature of Author

Mashrura Tasbih

Acknowledgment

All thanks to Almighty ALLAH, the creator and the owner of this universe, the most merciful, beneficent and the most gracious, who provided us guidance, strength and abilities to complete this research. We are especially thankful to Dr. Amitabha Chakrabarty, our thesis supervisor, for his help, guidance and support in completion of my project. We are also thankful to Md.Saiful Islam, our co-supervisor , for his assist, help and encouragement. We are grateful to the BRAC University, Faculty Staffs of the Computer Science and Engineering, who have been a light of guidance for us in the whole study period at BRAC University, particularly in building our base in education and enhancing our knowledge. Finally, we would like to express our sincere gratefulness to our beloved parents, brothers and sisters for their love and care. We are grateful to all of our friends who helped us directly or indirectly to complete our thesis.

Table of Contents

Declaration.....	2
Acknowledgement.....	3
Abstract.....	9
Table of Contents.....	4
List of Figures.....	7
Chapter 1: Introduction	
1.1 Introduction.....	10
1.2 Motivation.....	11
1.3 Thesis Outline.....	12
1.3.1 Contribution summary.....	12
1.3.2 Thesis Orientation	13
Chapter 2: Background Information	
2.1 Literature Review.....	14
2.2 Image Processing.....	17
2.2.1 Image Pre-processing.....	18
2.2.2 Feature or Information Extraction	25
2.3Multithreading.....	26
2.4 Hardware Overview.....	26
2.4.1 Raspberry Pi.....	27
2.4.2 Camera Module.....	28

2.4.3	Motor Driver Module	28
2.4.4	Buck Converter.....	28

Chapter 3: System Implementation

3.1	Block Diagram	31
3.2	Traffic Sign	32
3.3	Owners Security.....	34
3.3.1	Face Recognition.....	35
3.3.2	Notify the owner.....	36
3.4	Obstacle Detection.....	37
3.5	Traffic signal.....	38

Chapter 4: Experimental Result

4.1	Comparison between without Threading, Multithreading & PPcM.....	40
4.1.1	without Threading.....	40
4.2.2	Multithreading.....	40
4.2.3	PPcM(Parallel Processing coincident Multithreading)	41
4.2	Experimental Result of Individual Feature	41
4.2.1	Traffic Sign.....	42
4.2.2	Owner's Security.....	43
4.2.3	Obstacle detection.....	44
4.2.4	Traffic Signal.....	45

4.3 Overall comparison and perspective.....	46
4.3.1 FPS(Frame Per Second)	46
4.3.2 Processing Time.....	47
Chapter 5: Limitation	
5.1 Limitations.....	48
Chapter 6: Future Work	
6.1 Future Work.....	49
Chapter 7: Conclusion	
7.1 Conclusion.....	50
References.....	51

List of Figure

Fig- 2.2.1(a) Threshold image.....	20
Fig- 2.2.1(b) Noise Filtering image.....	21
Fig- 2.2.1(c) Contouring image.....	21
Fig- 2.2.1(d) RGB 2 GRAY image.....	22
Fig- 2.2.1(e) Blurring image.....	23
Fig- 2.2.1(f) HSV image.....	25
Fig- 2.4 List of Hardware equipments	27
Fig- 3.1: Diagram of the Proposed System.....	30
Fig- 3.1: Block diagram of the System.....	31
Fig- 3.3 Flowchart of owner's security.....	34
Fig- 4.2.1(a) Terminal result of Traffic sign using PPcM.....	42
Fig- 4.2.1(b) Identification of Traffic sign.....	42
Fig- 4.2.2(a) Terminal result of Owner's security using PPcM.....	43
Fig- 4.2.2(b) Identification of Owner's security.....	43
Fig- 4.2.3 (a)Terminal result of Obstacle Detection Using PPcM	44
Fig- 4.2.2(b) Identification of Owner's security.....	44
Fig- 4.2.3 (a)Terminal result of Obstacle Detection Using PPcM	45

Fig- 4.2.3 (b) Identification of Obstacle Detection.....	45
Fig- 4.3.1 Graph of the comparison of frames per second.....	46
Fig- 4.3.2 Graph of the comparison of processing time.....	47

Abstract

Image processing is one of the most widely used and dominant techniques in different sector of multimedia services. It has been considered as the most prominent and successful method evaluating its development in the area of interactive mass communication. While multi-threading is a very popular and prevalent process where multiple activities can proceed concurrently in the same program. As a result, in our research we try to bring a new approach of combining the image processing in the midst of multi-threading and parallel processing. Since the advance transportation system is one of the most significant and prominent representation of developing technology in this modern era. Therefore we implemented our proposed procedure on various feature of autonomous and advanced transportation technology for justifying our methodology. Our proposed algorithm PPcM (**parallel processing coincident multi-threading**) gives more efficient and proficient result in account of time, control, communication and computing techniques and ensure low latency.

Chapter 1

Introduction

1.1 Introduction

Image processing is one of the most prominent medium of processing an image and use it on various technology for utilizing it in the best possible way. Since a long time, image processing has been a very influential topic and it is developing drastically day by day. In different genre, this image processing technology is used for more advancement and effectiveness. That's why we try to make this process more efficient with the help of multi-threading which is a famous process of multi-tasking and make the image processing as fast as possible on the factor of time. We have also included the occurrence of parallel processing which help the multi-threading work more efficiently. Because in image processing time is a very important factor as delay of time and delay in arrival of frame may cause a lot of problem in the final result of the process. So, considering the importance of time, in our proposed algorithm(PPcM) we tried to implement the concept of multithreading and parallel processing so that we can make it as time efficient as possible.

On the other hand, smart car and advance transportation system are the symbol of this modern era which is rapidly developing with each passing day [1]. Each and every feature of the car or transportation is made in order to make people's life easier, comfortable and more secure. That's why in our research we try to build an autonomous transportation system with the help of our proposed algorithm which is consist of image processing and multithreading. Each feature of the system is build up by using our algorithm of image processing and multithreading for some important feature of a transportation system like detecting traffic signal, detecting the traffic sign, obstacle detection and it will be controlled by detecting the white straight line of the road. For security concern there is a feature of face detection which will be done before entering the car. All these features are established to demonstrate our proposed algorithm which gives a desired and efficient result with the consideration of time, control and computation.

1.2 Motivation

Image processing is one of the most advanced and widely used procedure for all terms of modern technology. The function of image processing has two sovereign operation. First of all, in the progression of pictorial information in human apprehension and secondly, the processing of a representation data for an autonomous machine perception. In this modern era, we can see the usage of image processing in many important sector and the digital image processing has a broad range of application like remote sensing, image and data storage for transmission in business application, medical imaging, forensic science, defence and military sector and industrial imaging. In addition to the mentioned application image processing is also widely used to solve many kind of problems through recognition and analyzation. For having so much importance in so many fields, we chose image processing as our research field.

On the contrary, Multithreaded applications have multiple threads executing in a shared address space. The widespread programming and execution model of multithreaded functions increase the efficiency of any program. Threading allows application remain responsive while doing lengthy and complex operations. So, to improve the overall performance, multithreading is a very important concept to use and increase the overall adaptivity and productiveness. So, when we use multithreading it definitely gives a better result. But in PPcM, parallel processing is also used along with multithreading which gives the best result and fastest fps.

Moreover, transportation is one of the basic requirement for the regular life. Since the last decade we have seen enormous growth in the field the transportation with the mean of technology and preference. With the accumulating technology and autonomy day by day in the field of transportation, are essentially concern with the safety issues and traffic congestion which is increasing day by day with the increment of both population and transportation. The advancement in this area plays a very significant role to deal with society and people's punctuality and comfort. In this digital generation, the fast, advance, autonomous, secure and comfortable transportation has also become a basic need of the people which will also save

their time. In addition with all of these, a safe, smooth, pleasant environment is also needed which will decrease the number accident and improve air quality. So, intelligent transportation system is a very broad category to work with for the betterment of the people.

As a result, for our research we chose these three popular areas to work with so that we can provide something effective, time efficient, advance which will be up-to-date technology and will be betterment for the people and society. In our intelligent transportation system, there will be feature like detecting obstacle, traffic sign and signal and control the system according to the scenario which will make the system autonomous and reduce people's anxiety about the car's movement. There is also a feature about the car's security by face recognition of the owner and also send an sms if someone tries to steal the car. All these features are done by our proposed algorithm which is consist of image processing, multithreading and parallel processing.

1.3 Thesis Outline

Since image processing is used in many field for the development, in our research firstly we tried to assemble it in a better and proficient way so that everyone can get the benefit and use it in a more organized way. To prove our proposed algorithm, we implemented it in a hardware based advanced transportation system where each attribute is manufactured by the algorithm which makes it autonomous and increase the mobility [2]. The hardware system is also developed by us.

1.3.1 Contribution Summary

The summary of the overall research and contribution are mentioned below:

- First of all, research on image processing and its implementation procedure on various types of medium like Matlab, OpenCV (python) etc has been done. We tried both way, but then we choose OpenCV for our final implementation
- Secondly, the hardware setup has been built to make the advance transportation system and the main microcontroller of the system is Raspberry Pi.

- After that, all the image processing algorithm in our hardware system has been implemented to run the system with the help of image processing and control the system according to the need.
- Lastly, as image processing algorithms are already very popular, we wanted to make ours better and more efficient. That's why we started to do more research to make it better and came up with the multithreading idea which makes it more proficient on the basis of the factor of time as it can pursue more FPS (frame per second) and makes it much faster than before. It also decreases the overall elapsed time which was more time-consuming earlier.

1.3.2 Thesis Orientation

The rest of the thesis report is organized as follows-

- Chapter 2 has all the background information about image processing, multithreading, hardware material's description and the literature review about the related work that have been done.
- Chapter 3 includes all the implementation and elaboration of each feature along with its algorithm.
- Chapter 4 demonstrates the experimental result.
- Chapter 5 has the limitation of the system.
- Chapter 6 has the future goal.
- Chapter 7 concludes the report.

Chapter 2

Background Information

In this chapter, the overall background description has been demonstrated. First of all, there is the description of literature survey which basically explains the work has been done in this topic so far or the papers that has been researched for our implementation. And also the basic information of image processing, multi-threading etc. The hardware equipments that have been used to build the system is also described in this section.

2.1 Literature Review

This paper reviewed smart vehicle in the contrary of implementation of image processing in a more efficient manner. To avoid road accident and make driving safer, comfortable and easier, using sensors and image processing algorithms we built an efficient automobile. Increasing number of accidents at busy junctions is a major threat faced by today's world. Nowadays people are driving very fast; we lost many valuable lives by making small mistake while driving. In order to avoid such accidents we attempted a project and for that we had to review many works done on this regards. Many works have already been done to make cars smarter than ever. While doing this project work we had to study a lot of them. While comparing a normal vehicle with a smart vehicle, the smart one is highly computerized and automated. Omnipresent sensors, devices, networks, and information are opening doors to a smart world in which smart devices have extended computational intelligence throughout the physical environment to provide reliable, relevant services to users [3]. These devices are getting smarter, multi-functional, and customizable for users to access and store wide-ranging information via many applications.

This is the era of growing demand of image and video processing. Almost everywhere computer vision is used in life. In order to cope with the growing plea of image and video processing open source computer vision library which is also known as openCV is used [4]. It is very important to use the most efficient computer vision to get the best result of the analysis of images and videos. OpenCV is a library containing over 500 optimized image

processing algorithms, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision and robotics.

With the headway of current advances regions identified with apply autonomy and PC vision; real time image processing has turned into an important innovation to contemplate. By utilizing open source PC vision library, a picture can be caught on the bases of its hue, saturation and colour value (HSV) range. The essential library capacities for picture handling and processing are used. Essential library capacities are utilized for loading a picture, making windows to hold picture at run time, saving pictures, and to separate images using their colour values. While processing, the pictures are changed over from their fundamental plan Red, Green, and Blue (RGB) to a more appropriate one that is HSV. The purpose of computer vision is to simulate the way of human eyes directly by using computer. Computer vision is a kind of research field which tries to percept and represent the 3D information for real objects. Its core is to reconstruct the visual aspects of 3D object by analysing the 2D information extracted accordingly. Real life 3D objects are represented by 2D images. The process of object detection analyses the input image and to determine the number, location, size, position of the objects [5]. Object detection is the base for object tracking and object recognition, whose results directly affect the process and accuracy of object recognition. The common object detection method is: colour-based approach, detecting objects based on their colour values. The method is strong adaptability and robustness.

For this project we implemented image processing algorithms on raspberry pi. Raspberry pi is single board computer and an embedded system to decrease complexity of systems in real time application. This is low budget and mainly based on python. Raspberry pi consist of Camera slot Interface (CSI) to interface the pi camera. Here, the Dark and Low contrast images captured by using the Raspberry Pi camera module are enhanced in order to identify the particular region of image or to detect an object and many more applications. This concept is used in the real time application. Because of its credit card sized and less weight in the design it is very convenient to use. However, the image captured by pi camera will consist of unwanted things due to atmospheric conditions; hence it is necessary to remove noise present in images. Image processing is basically manipulating and handling image or video frame as input and the output can be either another image or a video frame or a set of

characteristics or some action [6]. There are many image processing algorithms exist. At first the pi camera takes a video frame and does a lot of operation on it. It can contour the image, resize it, convert it to grey scale or in to HSV form, display it, detect particular objects in an image etc. any of these methods can be used as they all are known as image processing. Raspberry pi works pretty great with image processing algorithms and python.

To detect features or to detect a specific object, shape matching is required. For that Hough transform is used. The Hough transform is a technique which can be used to isolate features of a particular shape within an image. The Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses etc. since it requires that the desired features be specified in some parametric form. When a simple analytic description of a feature is not possible, the Hough transform is used there. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

Authentication is a substantial issue in system control in computer based communication. A person's identity is his/her face. The strategies to endeavour this physical feature have seen a great change since the approach of image processing techniques [7]. Automatic face recognition can be used in many aspects. It can be done by using image processing. Since technology's advancement has no bound now it should be implemented for our safety and security. If human face recognition can be implemented in a system where the door will not open unless it can match the face of its owner, it will ensure the security. The Raspberry pi module is used for face detection and recognition. The camera will be connected to the Raspberry pi module. The system will create a dataset holding the pictures of the person or persons who it needs to be recognized and then it will match the real time faces with the pictures in its dataset. It most certainly will preserve the owners security and safety.

Most of the CPUs are made today are using multiprocessing to make things faster than ever. This multiprocessing has impacted the computational world the most. CPU's core can execute different statements in parallel which is way more time efficient than single core processing. Traditional sequential programs will not run faster on multicore CPUs since such programs have a single thread of execution. Without a doubt, such programs are run on CPUs

with more cores; sequential programs use the available hardware less efficiently. To take advantage of multicore hardware, programs must be designed and written as parallel programs, with multiple threads of execution [8]. Multithread programming not only will use the underlying hardware efficiently, it will also run significantly faster using those hardware. For seamless implementation of hardware and software co-designed components, in hybrid chips both CPU and FPGA components provide the probable of providing a unified platform. Considering the potential of these hybrid chips which require new high level programming model capabilities will achieve more efficiency than the current methods. General multithreaded model can lead to programming efficiency improvement in a lot better way.

While getting efficient and reliable output from real-time digital image processing using multithreading, there are two types of classes are dealt with [9]. The first one is class A which one deals with the single pixel operations and the other one is class B where neighbourhood pixel values generate the output which is dedicated to convolution operations. To create a sub-image by dividing a real image is very complicated. The biggest problem is information loss which happens especially at the boundary pixels while partitioning. Finding the ideal size of the sub-image is also an issue. However, the main goal is to write a code with which it will get the maximum performance via multithreading is also very burdensome. So the aim is to reduce information loss while improving performance.

2.2 Image Processing

Today, there is no area of technology left where image processing has not somehow impacted. Image processing is a method of performing various operations on images to extract information to use the result for further processing or to enhance the images. It converts the image into digital form and executes mathematical operations[10]. It is a type of signal dispensation which inputs image or video frame and generated output as an image or some features from that image. Usually it is a two dimensional signal. It is now a rapidly growing technology in every sector of modern life and computer science.

The many things can be done using image processing includes:

- Image display and printing
- Image editing and manipulation
- Image enhancement
- Feature detection
- Image compression
- Information extraction
- Template matching etc.

The processing of image includes the three steps. They are:

- Importing the image with from a folder or using a device.
- Evaluating and manipulating the image which includes data compression and image enhancement and spotting patterns.
- In last stage which is output, the result can be altered image or report that is based on image analysis.

In this paper only few of them are considered and described. The broad range of image processing has helped this project get to achieve its goals in a lot of way. In this project of autonomous vehicle there are many functions of image processing are involved. In this age of technology we are keen to implement it to make life simpler. This automated vehicle is one of the attempts of making life easier and safer. This autonomous, intelligent automobile will extract information from its surroundings most of them are in form of images and video frames and will process them using many methods of image processing and will take applicable actions based on the processed data.

2.2.1 Image Pre-processing

Sometimes images captured by sensors can have errors related to geometry or brightness value of the pixel. These errors can be corrected using appropriate mathematical models. This is a sort of an image enhancement which is the modification of image by changing its pixel brightness values to better its visual impact for further use of that image. Image enhancement many techniques to improve the visual form of an image which is better suited for human or machine interpretation. Images may have different types of noise. In image enhancement, the aim is to accentuate certain image features for later analysis. As in contrast and edge

enhancement, pseudo-colouring, noise filtering, sharpening, and magnifying. Image enhancement is useful in feature extraction, image analysis and an image display. The enhancement process does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application dependent. Some of the enhancement techniques are:

- a. Threshold
- b. Noise filtering
- c. Contour
- d. RGB to grey
- e. Blur
- f. HSV

a. Threshold

Thresholding is a method of image segmentation. It can produce a binary image from a grayscale image. For many applications, it is useful to be able to separate out the regions of the image corresponding to objects in which we are concerned, from the regions of the image that correspond to background. Thresholding provides an easy and appropriate way to perform this segmentation on the basis of the different intensities or colours in the foreground and background of an image. It is also useful to be able to see what areas of an image consist of pixels whose values lie within a specified range. Thresholding creates binary images from grey-level ones by turning all pixels below some threshold to zero and all pixels above that threshold to one [11].

If $g(x,y)$ is threshold version of $f(x,y)$ at some global threshold T [12],

$$g(x,y) = 1 \text{ if } f(x,y) \geq T$$
$$0 \text{ otherwise}$$

The process of thresholding along with its input and output:



Fig- 2.2.1(a) Threshold image

b. Noise Filtering

To remove unnecessary information from an image, noise filtering is used. It is also used to remove various types of noises from the images. Mostly this feature is interactive. Several filters like low pass, high pass, mean, median etc., are available. A typical noise model is the Gaussian distribution parameterized by μ and σ [13]

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



Fig 2.2.1(b) Noise Filtering image

c. Contour

When dealing with a region of object, several compact representations are available that can facilitate manipulation of and measurements on the object. Contour is also known as border following or boundary following; contour tracing is a method that is applied to digital images in order to extract their boundary[14].

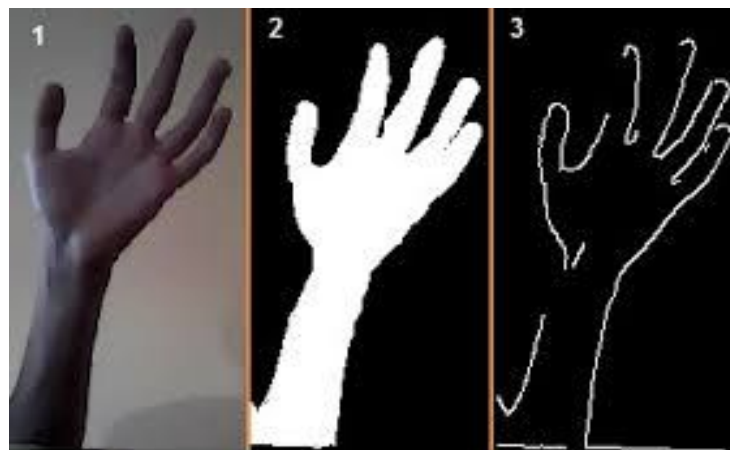


Fig-2.2.1(c) Contouring image

d. RGB to Gray

In images and computing, a grayscale digital image is an image in which the value of each

pixel is a single sample, as in, it conveys only intensity information. These sorts of images are also known as black-and-white, and they are composed exclusively of shades of grey, varying from black at the weakest intensity to white at the strongest. On different screens in example monitor, cell phone screen, digital camera screen different colours are displayed by varying the amount of red, green and blue light that shines through the pixels. This is known as the RGB colour model. The RGB encoding pure red is (255,0,0), pure green is (0,255,0), pure blue is (0,0,255). Grayscale images are extracted in black, white, and all the shades of grey in between. The RGB encoded grey scale value is a set of equal numbers. Conversion of a RGB image into a grayscale image is done to do a lot of other application on that grayscale image which is more convenient than the original one [15]. The basic equation used in the conversion process in

$$\text{Grayscale} = (R + G + B)/3$$

This is known as the average method of grayscale conversion. It may need some ration adjustment of red, green or blue colors. If red is decreased and green is increased for the perfect grayscale image, the equation will be,

$$\text{grayscale image} = ((0.3 * R) + (0.59 * G) + (0.11 * B))$$



Fig-2.2.1(d) RGB 2 GRAY image

e. Blurring

An image looks more sharp and detailed, by blurring we simply reduce the edges. By doing

this it makes the transition from one colour to another look smoother in an image. In image processing Gaussian blur or Gaussian smoothing is the result of blurring an image using Gaussian function. It is a very widely used function typically to reduce image noise and reduce detail. Gaussian blur is used in image pre-processing stage in order to enhance image structures in different scales [16]. This is a type of image blurring filter that uses a Gaussian function to calculate the transformation to apply to each pixel in the image The equation of a Gaussian function in one dimension is,

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

In two dimensions, it is product of two such Gaussians,

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Here x is the distance from the origin in the horizontal axis, y is the distance from the origin in vertical axis and σ is the standard deviation of Gaussian distribution.



Fig-2.2.1(e) Blurring image

f. HSV

HSV stands for hue, saturation and value. This is the most common cylindrical-coordinate representation of points in an RGB colour model. Value range for 'hue', 'saturation' and 'value' are respectively 0-179, 0-255 and 0-255. 'Hue' represents the colour. The hue is the proportion of the distance around the edge of the hexagon which passes through the projected point, originally measured on the range [0, 1) but now typically measured in degrees[0°, 360°). For points which project onto the origin in the chromaticity plane, hue is undefined [17].

$H' = \text{undefined}, \text{ if } c = 0$

$$\frac{G-B}{C} \text{ mod } 6, \text{ if } M = R$$

$$\frac{B-R}{C} + 2, \text{ if } M = G$$

$$\frac{R-G}{C} + 4, \text{ if } M = B$$

$$H = 60^\circ \times H'$$

'Saturation' represents the amount to which that respective colour is mixed with white. The equation of saturation is,

$$S = 0, \text{ if } V = 0$$

$$\frac{C}{V}, \text{ otherwise}$$

The 'value' represents the amount to which that respective colour is mixed with black. This is a very important pre-processing of image which is further used for many different image processing methods.

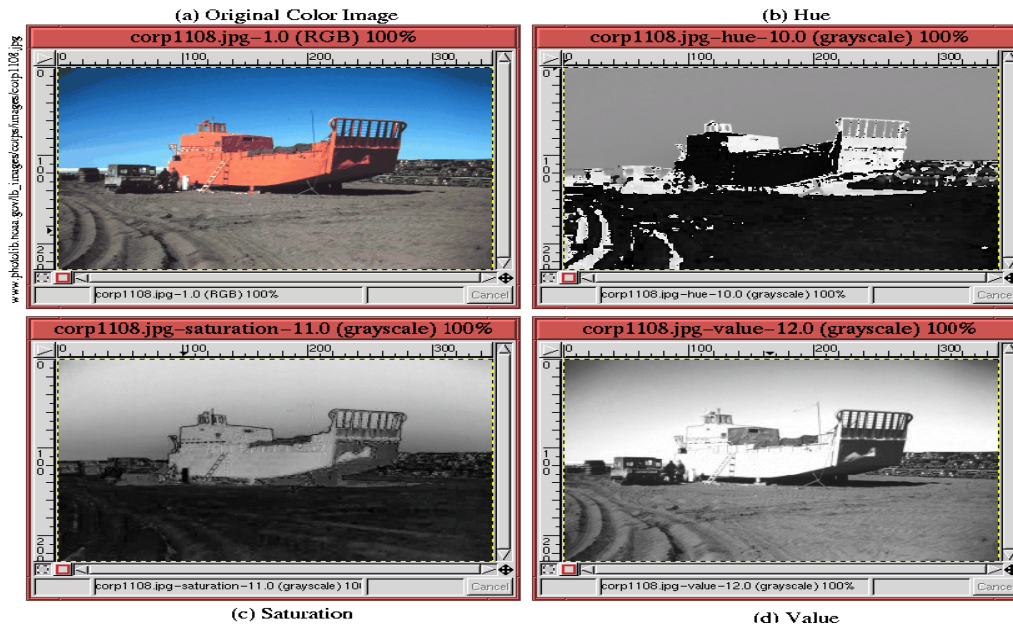


Fig-2.2.1(f) HSV image

2.2.2 Feature or Information Extraction

The feature extraction methods are established to extract features in synthetic aperture radar images. This method extracts high-level features needed in order to perform classification of targets. Features are unique items such as size, shape, composition, location and so on. Segmentation techniques are used to isolate the desired object from the scene so that measurements can be made on it afterwards. Quantitative measurements of object structures allow classification and description of the image.

At the point when the pre-processing and the desired level of segmentation has been accomplished, some feature extraction technique is applied, which is followed by application of classification and post processing methods. The efficiency of feature or object recognition depends on this feature extraction technique. One of the methods of feature extraction is feature selection which is the single most important factor in attaining high recognition performance. In our project we used Hough transformation, Hough circle detection, rectangle detection and many more of the methods.

2.3 Multithreading

Multithreading is a process of executing multiple threads simultaneously. This is the capability of a program or an operating system process to manage its use by more than one user at the same time and to even manage multiple requests by the same user without having to have multiple copies of the programming running in the computer. Each user request for a program or system service is kept track of as a thread with a separate identity.

Multiprocessing and multithreading, both are used to achieve multitasking. But we use multithreading than multiprocessing because threads share a common memory area[18]. They don't allocate separate memory area so it saves memory, and context-switching between the threads takes less time than process.

The major advantage of multithreading is if a thread gets a lot of cache misses, the other threads can continue taking advantage of the unused resources, which may lead to faster overall execution as these resources would have been idle if only a single thread were executed. Also, if a thread cannot use all the computing resources, running another thread may prevent those resources from becoming idle. If several threads work on the same set of data, they can actually share their cache, leading to better cache usage or synchronization on its values.

This approach is used to attain concurrency and parallelism. Using multithread in our program made it work way faster. It makes it time efficient and can process faster which of course is an achievement.

2.4 Hardware Overview

To build the prototype we had to combine with some hardware where we integrated the programs. The components used are basically a microprocessor, some sensors, motors, motor drivers, battery etc.

Main components used are given in the following table:

Product	Quantity
Raspberry pi 3 B	1
Raspberry pi 3 B case	1
Heatsink kit for raspberry pi(3 piece)	1
Raspberry camera module 5MP	1
L298N motor driver	1
6V DC motor	2
Buck converter	2
M-M Jumper(Single)	5
M-F Jumper(Single)	5
F-F Jumper(Single)	5

Fig- 2.3 List of Hardware equipments

2.4.1 Raspberry Pi

To implement the whole system we will be using raspberry pi 3 B as the microprocessor. This is the third generation raspberry pi. For the raspberry pi we also needed a case, which is specially made for this one and 3 pieces of heatsink kits to keep the raspberry pi cool [19].

This small computer has many features and some of them are-

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GB RAM

- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot (now push-pull rather than push-push)
- VideoCore IV 3D graphics core

2.4.2 Camera Module

For video recording and still images we will use raspberry pi camera module of 5MP. The features of it are:

- It can take high definition videos and still photographs.
- It's easy to use and implement.
- It has plenty of advanced options such as, slow-motion, time-lapse etc.
- It is a 5 megapixel fixed focus camera that supports 1080p30, 720p60 and VGA90 video modes.

The module attaches using a 15cm ribbon cable to the CSI port in the raspberry pi.

2.4.3 Motor Driver Module

We will be using a L298N motor driver module, which is also known as H bridge motor controller module. This is typically used to control motors speed and direction. It can be used for other projects such as driving the brightness of certain lighting projects such as high powered LED arrays. An H-Bridge is a circuit that can drive a current in either polarity and be controlled by Pulse Width Modulation (PWM).

2.4.4 Buck Converter

The buck converter is a type of DC-to-DC or AC-to-DC converter that has an output voltage magnitude that is either greater than or less than the input voltage magnitude. It is useful

where electrical isolation is not needed between the switching circuit and the output. The switching transistor between the input and output of the buck converter continually switches on and off at high frequency. Some buck converter can directly take the input from a DC source, such as battery. [20]

Chapter 3

System Implementation

Advance transportation System is a very applicable as well as beneficial field. As mentioned earlier image processing has a broad range of usage in various sector like military, defence, medical, industry etc. But in our research it has been implemented in a vehicular network as vehicles are really important for people of the world for their movement from one place to another. People needs to spend a lot of time for travelling and also a lot of our energy absorbed by concentrating on the traffic issues. So, we wanted to make people's life easier and comfortable so that they can store their energy and attention for other important work and can be more relaxed for the driving and travelling issues. Also the medium of transportation is rapidly developing with the growth of population. And considering the security of the system, face recognition of the owner has been implemented which is also done by image processing. With a view to maintain the security of the system, only owners will be allowed to get inside the car.

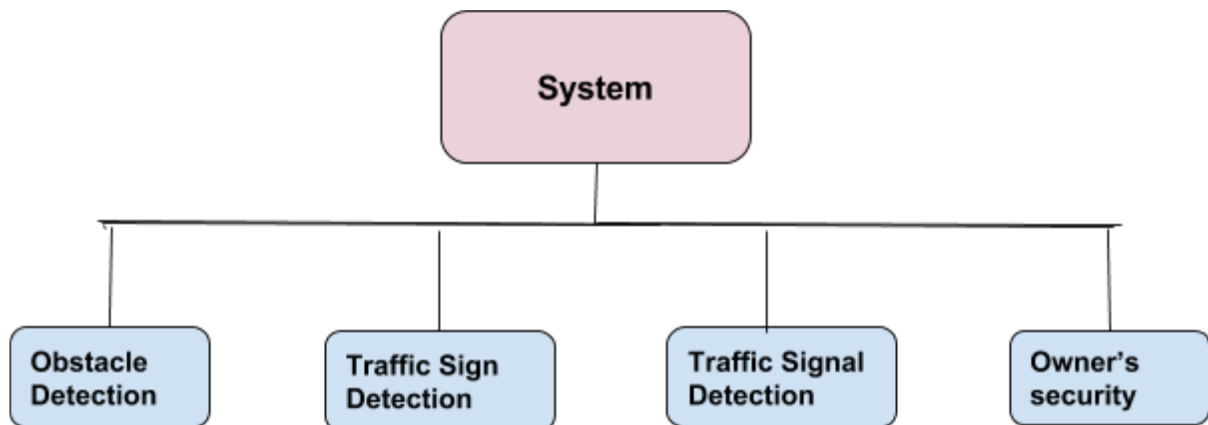


Fig- 3 Diagram of the Proposed System

3.1 Block Diagram

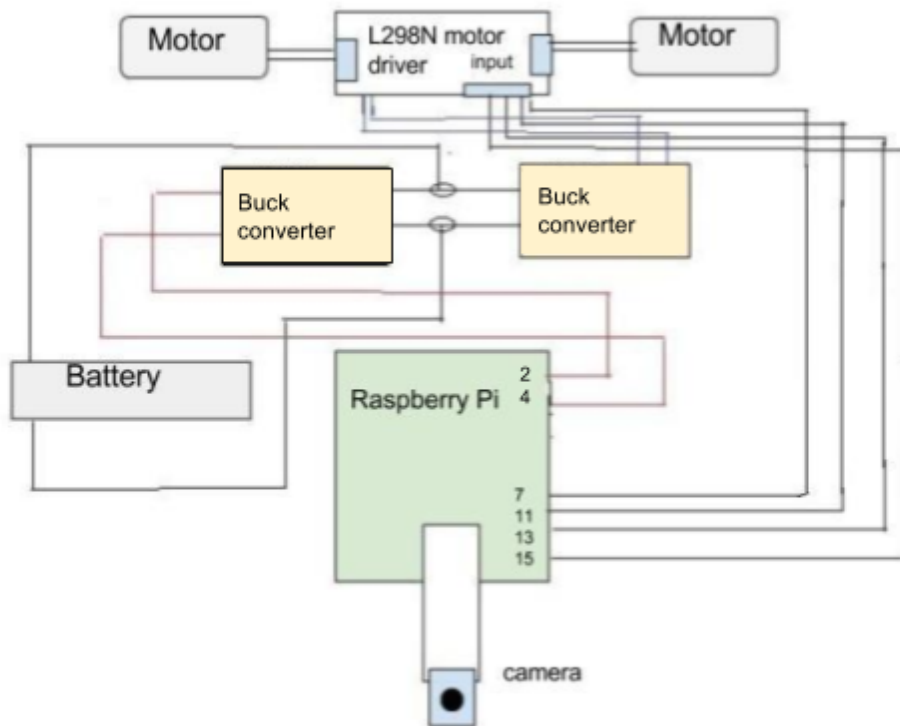


Fig-3.1 Block diagram of the System

In the basic block diagram of the system, four input pin from L298N motor driver come to the raspberry pi pin 7,11, 13 and 15 respectively. Two wires are connected from motor driver to motor, so that it can control the motor. Two buck converters have been used, two wires from the driver are connected to one converter. Both the converters are also connected with each other. From the other converter two wires are connected with pin2 and pin4 of raspberry pi. Battery is connected by two wires with both the controller. And there is a camera module connected to raspberry pi.

3.2 Traffic Sign

While travelling on the road, we can see many kind of traffic sign which help us to instruct the road or to reach our destination. Such as- stop, U-turn, turn left, turn right, no cycle etc. In our system, we have implemented this feature and this will be also done with image processing. Firstly, all the possible image of traffic sign will be stored in a folder. As soon as the system, started to process all the images that it observe in its expedition, it will try to match the images from its corresponding folder images, and if any of the image matches, it will do the action that is affiliated under the condition of that image. Like- if the system observe the stop sign through image processing and then if it finds the match image that is stored in the database, it will stop the motor of the car. So the car will be stopped as it has seen the stop sign board. Thus by processing each sign , it will make the car autonomous and reduce the pressure of the driver of monitoring all the time around and act according to it. Algorithm of this traffic sign detection is given below-

Algorithm:

```
vs ← PiVideoStream().start()
```

```
fps ← FPS.start()
```

```
while fps < num_frame
```

```
    frame ← vs.read()
```

```
    blur ← vs.processing()
```

```
    edge ← Auto-cany (blur)
```

```
    img ← vs.identify_cons (edge)
```

```
    for each i in RefImages
```

```
        if i matches img
```

```
            give decision
```


break

else

continue

In traditional image processing method. there are some other steps, like canny, threshold, contour. But in PPcM, some of these methods have been removed from the main class, and shifted these to the threading file which is imported in the beginning. So, these methods parallelly occurring with the main class.

3.3 Owner's Security

A car's or system's security is very important nowadays as the number of stealing is increasing day by day. So, everyone is very much concerned about their car's security from the very beginning. That's why this owner's security system has been implemented to make the system more secure. It is basically a two stage implementation where in the first stage, it will try to recognize the face and in the second stage it will notify the owner by sending an sms if the face doesn't match. A flow chart of the overall procedure of this security system is given below-

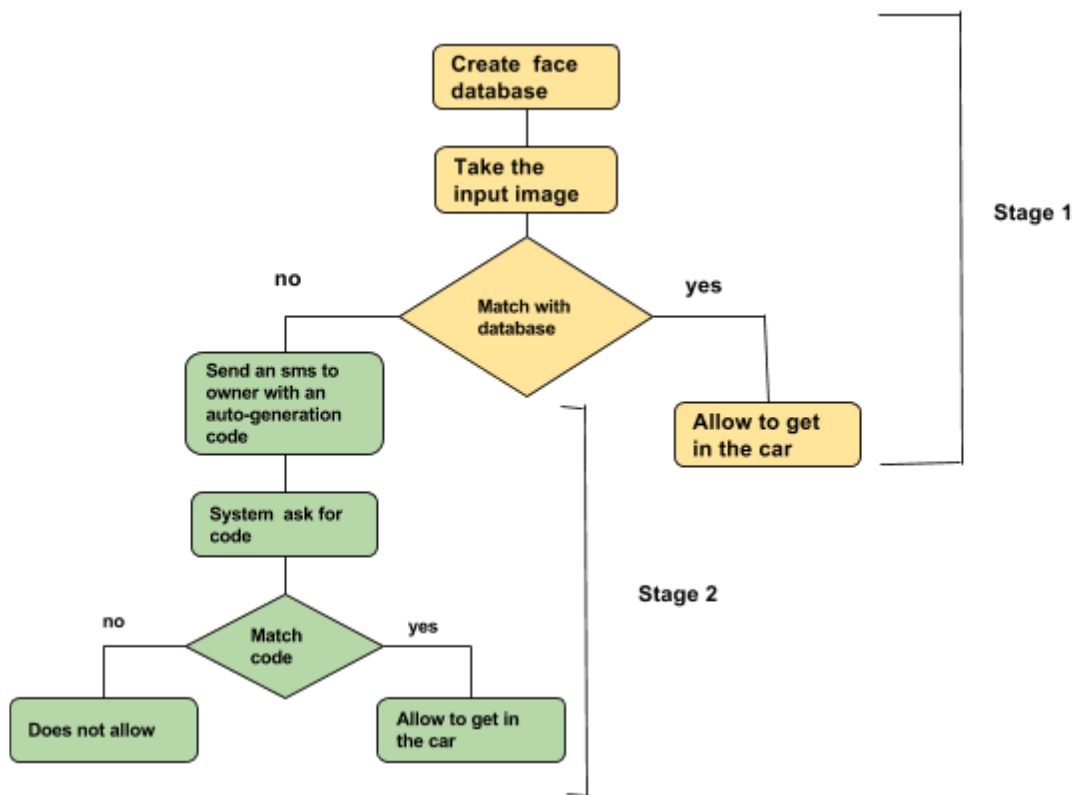


Fig- 3.2 Flowchart of owner's security

3.3.1 Face recognition (Stage one)

First of all, the user will be able to create a face database where there will be storage of all the people's image who have the probability to get into the car. After that, whenever someone tries to get inside the car, his/her face will match with the database images. If the face matches with stored images, the door of the car will open and the person will be able to get into the car. But if the face does not match, it will send an sms through Raspberry Pi, which will be described elaborately in the second stage. As of now, the algorithm for face recognition is given below:

Algorithm:

```
vs ← PiVideoStream().start()
```

```
fps ← FPS.start()
```

```
while fps < num_frame
```

```
    frame ← vs.read()
```

```
    vs.compareface()
```

```
    num ← vs.readnum()
```

```
    if num = 30
```

```
        send sms
```

```
        break
```

In the face recognition, there are also some functions which need to be done to recognize the faces. Like `cv2.rectangle()` which is done in a loop, then there were also crop, resize. But for PPcM, these methods are removed from the main class, and shifted to the the thread file which was imported in the beginning so that these process can be done concurrently.

3.3.2 Notify the Owner (stage two)

whenever some unknown person whose face is not stored in the database try to enter in the car, the the system will not unlock the door and it will send sms to the owner of the car whose number will also be stored in the database. The sms will be send through Raspberry Pi and each sms will cost almost BDT 3.00 and we think it is not very much if we consider the safety of the car. Along with the sms there will also be auto generation code number. Because in some cases it would be possible that, the owner himself have given some allowance to use his car but his face image is not stored the database. So it is not the case of stealing. In that case the owner can provide that code number to the person. If the person gives the right code as input , he will be allowed to open the door. But if it is a case of stealing then, that person will not be able to put the right code. That time the door will not open and he will not be able to steal. On the other hand, the owner will also get to know through that sms that someone is trying to steal his car and take some necessary steps to catch that person. In short all the process of this part is given below:

- Notify the owner by sms.
- Send an auto generation code along with the sms.
- The owner share the code if the person is known.
- By entering right code, the person will be allowed.
- If the person is trying to steal, the owner will get to know it by that sms.
- If the person enter wrong code or no code, he will not be able to get into the car.

3.4 Obstacle Detection

While passing on the road, there can be traffic where the car need to stop down its motor. Again, there can be any stone or obstacle or any person in front of where the car have to stop to control the situation otherwise any severe accident may occur which can be dangerous. To avoid such situation and to make our system autonomous, this obstacle detection feature has been implemented. And it will be also done with image processing. And PPcM algorithm will give a faster result compare to without using any algorithm or multithreading. The basic steps of the implementation of obstacle detection with our PPcM algorithm is given below-

Algorithm:

```
motor ← true

vs ← PiVideoStream().start()

fps ← FPS.start()

while fps < num_frame
    frame ← vs.read()
    vs.detect_car()
    measure ← vs.measure_read()

    if measure < 4
        motor ← false
    else
        motor ← true
```

For obstacle detection, cv2.rectangle and resize methods are also needed. But these methods have shifted to the thread file to avoid the extra loop and minimize the processing time.

3.5 Traffic signal

During passing through the road, the most important thing is to follow the traffic signal, where the green light indicate vehicles can go forward and red light indicate that now vehicles need to stop. And this feature is also implemented in our system, definitely using image processing as our whole system is based on image processing. It is a bit tricky to identify the signal, because it can be assumed that anything that is round and red or round and green can be the signal. But in a real time scenario, there can be many other things apart from traffic signal that is round shaped and green or red coloured. So, to confirm that yes, it is a light , we also measure the intensity of the object. the steps to identification of the light is-

- Identify the shape (round is expected).
- Identify the colour (red, green and yellow is expected).
- Measure the intensity for the confirmation of the light .

Now, along with the recognition the algorithm is also need to be mentioned. The basic steps of the algorithm are given below:

Algorithm:

motor ← true

vs ← PiVideoStream().start()

fps ← FPS.start()

while *fps* < num_frame

frame ← vs.read()

 vs.findRange()

 vs.decision_r ()

result_a = vs.read_result_a()

if *result_a* > 10

motor ← false

vs.decision_g()

vs.decision_y()

motor ← true

Here, `decision_r`, `decision_g` and `decision_y` basically give the decision for red, green and yellow colour. Two important methods are, `r_circles()` which determine the circles and range `[-r,r]` determines the intensity . These shifted from the main class to the thread file to run concurrently along with the main file.

Chapter 4

Experimental Result

In this chapter, the difference between the three schemes that has been implemented and the comparison between their experimental result has been explained. It clearly demonstrates the better result among these three.

4.1 Comparison between without threading, Multithreading and PPcM

In our research, all the features that are mentioned in chapter 03, are implemented in three different way. And after implementing it, in a three different way, the comparison of the processing time and frame number per second output will be shown in this chapter through the terminal value which is also implement in bar charts to analyze the comparison in a better way. But at first we will discuss here about the basic difference of without threading, multithreading and PPcM.

4.1.1 Without threading

In this processing the traditional image processing procedure has been implemented in the normal way to fetch each frame. So, after fetching a frame it will go through all the processing stuff of image processing like RGB2GRAY, blur, threshold, contour, measurement, HSV etc. Throughout this time, all the other frames that are coming will not be accepted. After completing all the process, the next frame that just arrived that will go again for the operation. so, in between there are loss of lots of frames which slow downs the main video as it take much time to capture. So, there will be much delay which is not good for image processing at all as time is the most important factor in image processing.

4.1.2 Multithreading

In this process at first imutils has been imported from which the pi video stream file has been operated to use the threading. So, after importing this, now it fetches all the frames with the help of threading. In this way, there is no loss of frame. Each frame is hold until each one is

coming back after the processing. So, it is much faster and better than the previous one where we are losing many frames. It also gives us a good speed compared to the previous one. But as the other frames need to wait for the previous one's result to return, there is also a little delay. So, the latency is smaller compare to the original one but greater compare to our process PPcM.

4.1.3 PPcM (Parallel Processing coincident Multithreading)

In this process, a separate threading file has been created where all the processing substance of image processing like RGB2GRAY, blur, threshold, contour, measurement, HSV etc have been done. And pi video stream has been imported from this file instead of the previous built in one. So, in this process no frames need to wait. As all the processing are done in the newly created threading file that we are importing. So, there is no frame loss. There is also very less latency and the frame fetch is also super fast. In the chapter 03, all the mentioned algorithm have been carried out this way.

4.2 Experimental Result of Individual Feature

There are three different types of output on the basis of time elapsed and fps for a particular frame number when the above mentioned three different types of procedure in each of the feature of our advance transportation system have been implemented. And by observing the result , it is very easy to compare.

4.2.1 Traffic sign

In traffic sign, again the fps is greater which is 43 and elapsed time is smaller which is 7.44 sec in PPcM which is more efficient compared to the other procedure. In the second image, there are two output of traffic sign identification, stop and turn around.

```
[INFO] approx. FPS: 7.99
(cv) pi@raspberrypi:~ $ scrot
(cv) pi@raspberrypi:~ $ python traffic_sign_without_threading.py
[INFO] Frames from `picamera` module for Traffic_Sign_Detection WITHOUT_Threading
[INFO] Total Frame: 320.00
[INFO] elapsed time: 32.07
[INFO] approx. FPS: 10.01
(cv) pi@raspberrypi:~ $ python traffic_sign_with_threading.py
[INFO] Frames from `picamera` module for Traffic_Sign_Detection WITH_Threading
[INFO] Total Frame: 320.00
[INFO] elapsed time: 11.66
[INFO] approx. FPS: 27.45
Exception in thread Thread-1 (most likely raised during interpreter shutdown):(c
v) pi@raspberrypi:~ $ python traffic_sign_with_PPcM.py
[INFO] Frames from `picamera` module for Traffic_sign_Detection WITH_PPcM
[INFO] Total Frame: 320.00
[INFO] elapsed time: 7.44
[INFO] approx. FPS: 43.00
(cv) pi@raspberrypi:~ $ scrot
```

Fig- 4.2.1(a) Terminal result of Traffic sign using PPcm

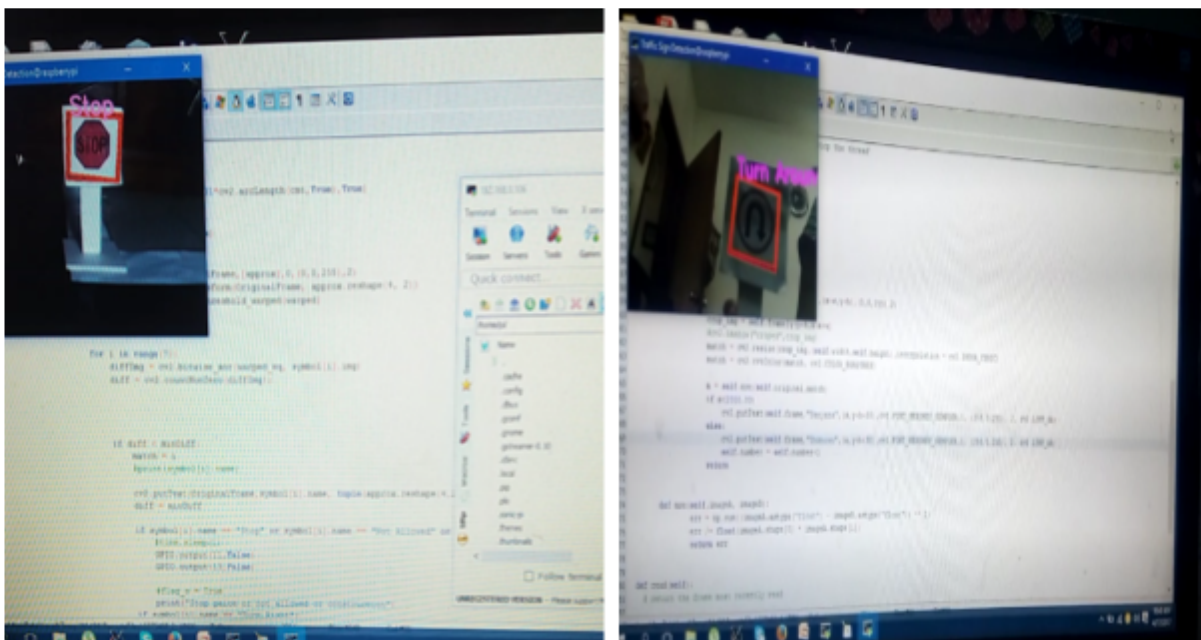
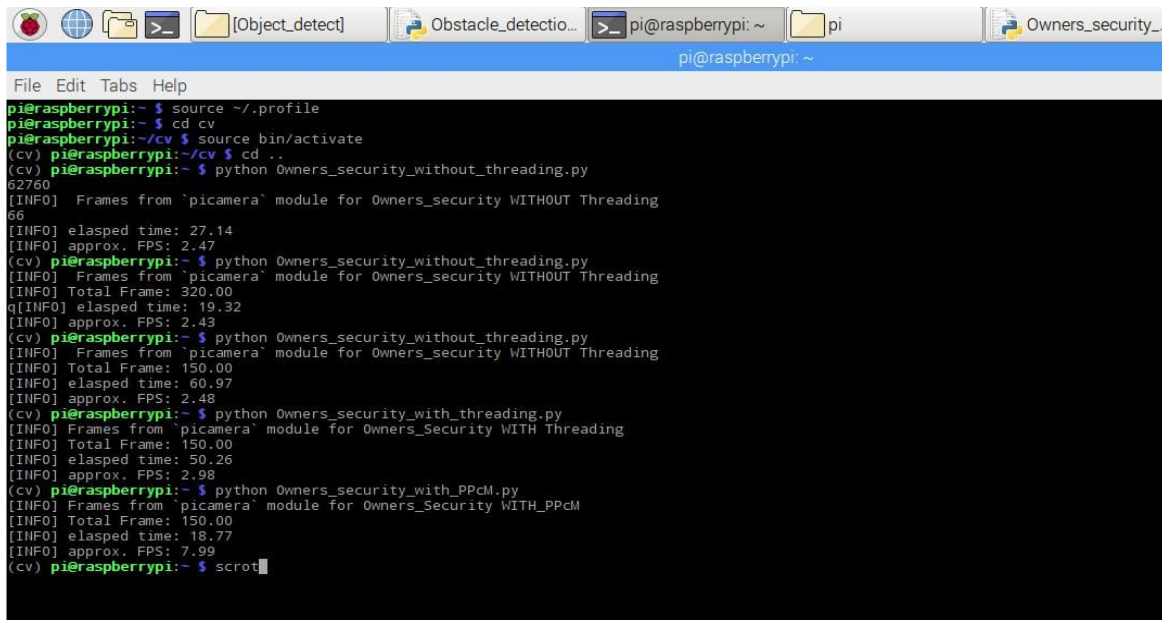


Fig- 4.2.1(b) Identification of Traffic sign

4.2.2 Owner's Security

In this feature, the efficient and accelerated result of PPcM in account of time and fame per second has been shown and also the identification of a known and unknown face.



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ source ~/.profile
pi@raspberrypi:~$ cd cv
pi@raspberrypi:~/cv$ source bin/activate
(cv) pi@raspberrypi:~/cv$ cd ..
(cv) pi@raspberrypi:~$ python Owners_security_without_threading.py
62760
[INFO] Frames from 'picamera' module for Owners_security WITHOUT Threading
66
[INFO] elapsed time: 27.14
[INFO] approx. FPS: 2.47
(cv) pi@raspberrypi:~$ python Owners_security_without_threading.py
[INFO] Frames from 'picamera' module for Owners_security WITHOUT Threading
[INFO] Total Frame: 320.00
q[INFO] elapsed time: 19.32
[INFO] approx. FPS: 2.43
(cv) pi@raspberrypi:~$ python Owners_security_without_threading.py
[INFO] Frames from 'picamera' module for Owners_security WITHOUT Threading
[INFO] Total Frame: 150.00
[INFO] elapsed time: 60.97
[INFO] approx. FPS: 2.48
(cv) pi@raspberrypi:~$ python Owners_security_with_threading.py
[INFO] Frames from 'picamera' module for Owners_Security WITH Threading
[INFO] Total Frame: 150.00
[INFO] elapsed time: 50.26
[INFO] approx. FPS: 2.98
(cv) pi@raspberrypi:~$ python Owners_security_with_PPcM.py
[INFO] Frames from 'picamera' module for Owners_Security WITH_PPcM
[INFO] Total Frame: 150.00
[INFO] elapsed time: 18.77
[INFO] approx. FPS: 7.99
(cv) pi@raspberrypi:~$ scrot
```

Fig- 4.2.2(a) Terminal result of Owner's security using PPcM

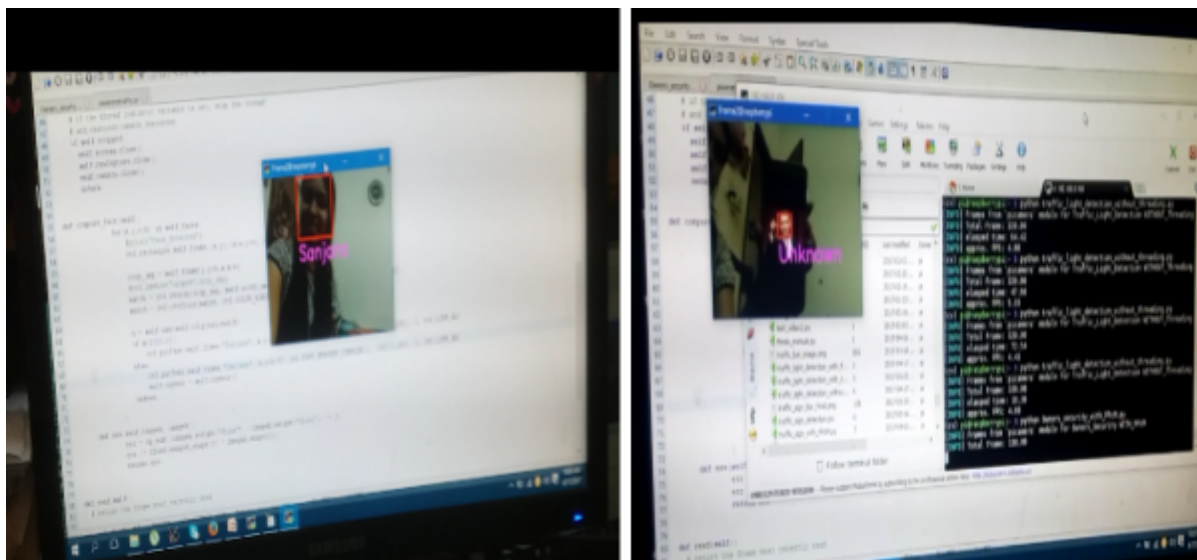
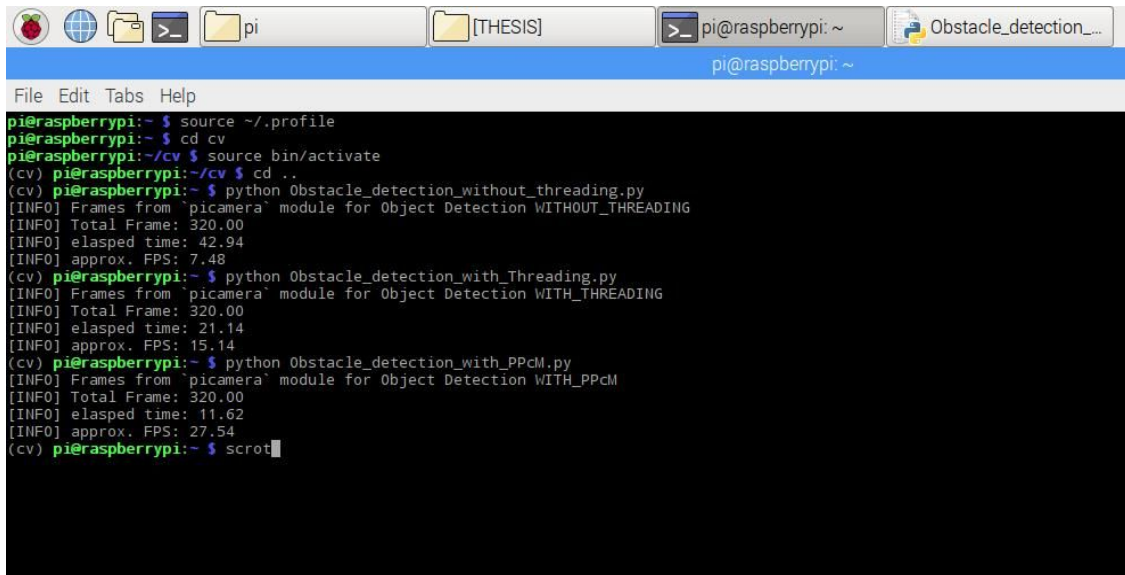


Fig- 4.2.2(b) Identification of Owner's security

4.2.3 Obstacle detection

In the following image we can see that for the particular frame number 320 we are getting the maximum number of frames and lowest elapsed by using PPcM. In the second image, there is also an identification of an obstacle along with the measure of the distance.



```
pi@raspberrypi:~$ source ~/.profile
pi@raspberrypi:~$ cd cv
pi@raspberrypi:~/cv$ source bin/activate
(cv) pi@raspberrypi:~/cv$ cd ..
(cv) pi@raspberrypi:~$ python Obstacle_detection_without_threading.py
[INFO] Frames from 'picamera' module for Object Detection WITHOUT_THREADING
[INFO] Total Frame: 320.00
[INFO] elapsed time: 42.94
[INFO] approx. FPS: 7.48
(cv) pi@raspberrypi:~$ python Obstacle_detection_with_Threading.py
[INFO] Frames from 'picamera' module for Object Detection WITH_THREADING
[INFO] Total Frame: 320.00
[INFO] elapsed time: 21.14
[INFO] approx. FPS: 15.14
(cv) pi@raspberrypi:~$ python Obstacle_detection_with_PPcM.py
[INFO] Frames from 'picamera' module for Object Detection WITH_PPcM
[INFO] Total Frame: 320.00
[INFO] elapsed time: 11.62
[INFO] approx. FPS: 27.54
(cv) pi@raspberrypi:~$ scrot
```

Fig- 4.2.3 (a) Terminal result of Obstacle Detection Using PPcM

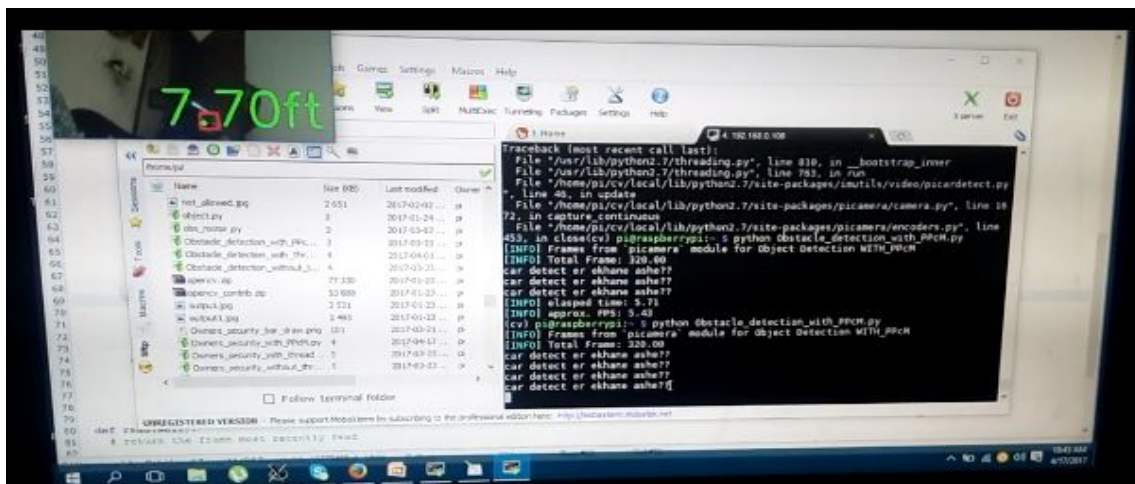


Fig- 4.2.3 (b) Identification of Obstacle Detection

4.2.4 Traffic Signal

Here, also the most effectual and low latency result is given by PPcM which can be determined by the terminal result. From the second output image, the red light identification can be assured.

```
[INFO] elapsed time: 14.77
[INFO] approx. FPS: 21.67
(cv) pi@raspberrypi:~$ python traffic_light_detection_without_threading.py
[INFO] Frames from 'picamera' module for Traffic_Light_Detection WITHOUT_Threading
[INFO] Total Frame: 320.00
[INFO] elapsed time: 37.08
[INFO] approx. FPS: 8.66
(cv) pi@raspberrypi:~$ python traffic_light_detection_with_threading.py
[INFO] Frames from 'picamera' module for Traffic_Light_Detection WITH_Threading
[INFO] Total Frame: 320.00
[INFO] elapsed time: 24.20
[INFO] approx. FPS: 13.22
(cv) pi@raspberrypi:~$ python traffic_light_detection_with_PPcM.py
[INFO] Frames from 'picamera' module for Traffic_Light_Detection WITH_PPcM
[INFO] Total Frame: 320.00
[INFO] elapsed time: 15.03
[INFO] approx. FPS: 21.29
(cv) pi@raspberrypi:~$ scrot
```

Fig- 4.2.4(a) Terminal result of Traffic signal

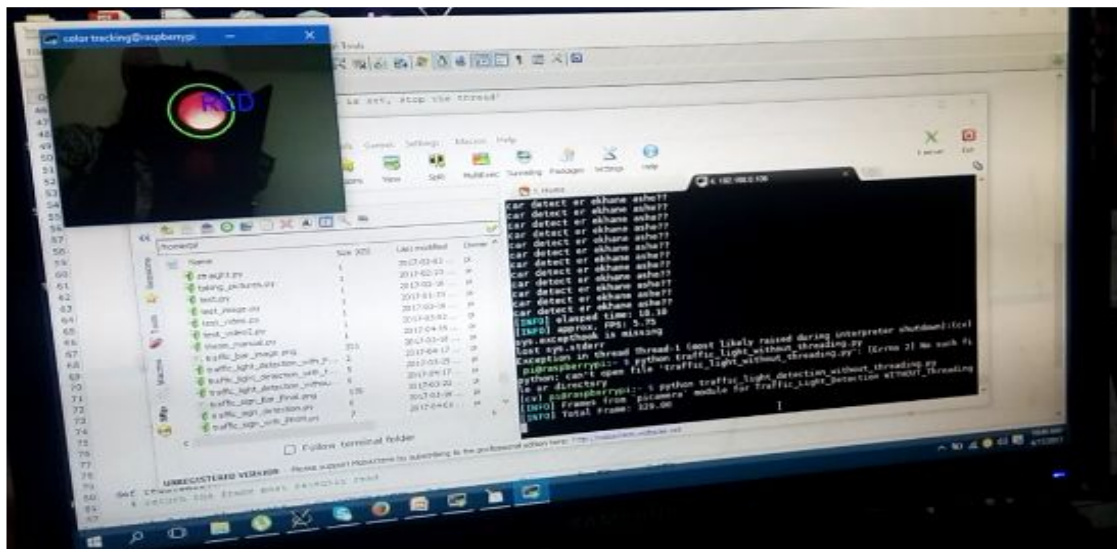


Fig- 4.2.4(b) Identification of Traffic signal

4.3 Overall Comparison and Perspective

To understand and illustrate the result more accurately , we have invoke all our values in the following bar chart and the results will clearly show us the difference between each.

4.3.1 FPS (Frame per second)

For image processing the more number of frames one can execute , it will give a more accurate result. From the following graph we can see PPcM can fetch maximum number of frames for each of the feature compare to multithreading which gives a medium result and without threading which has the lowest fps.

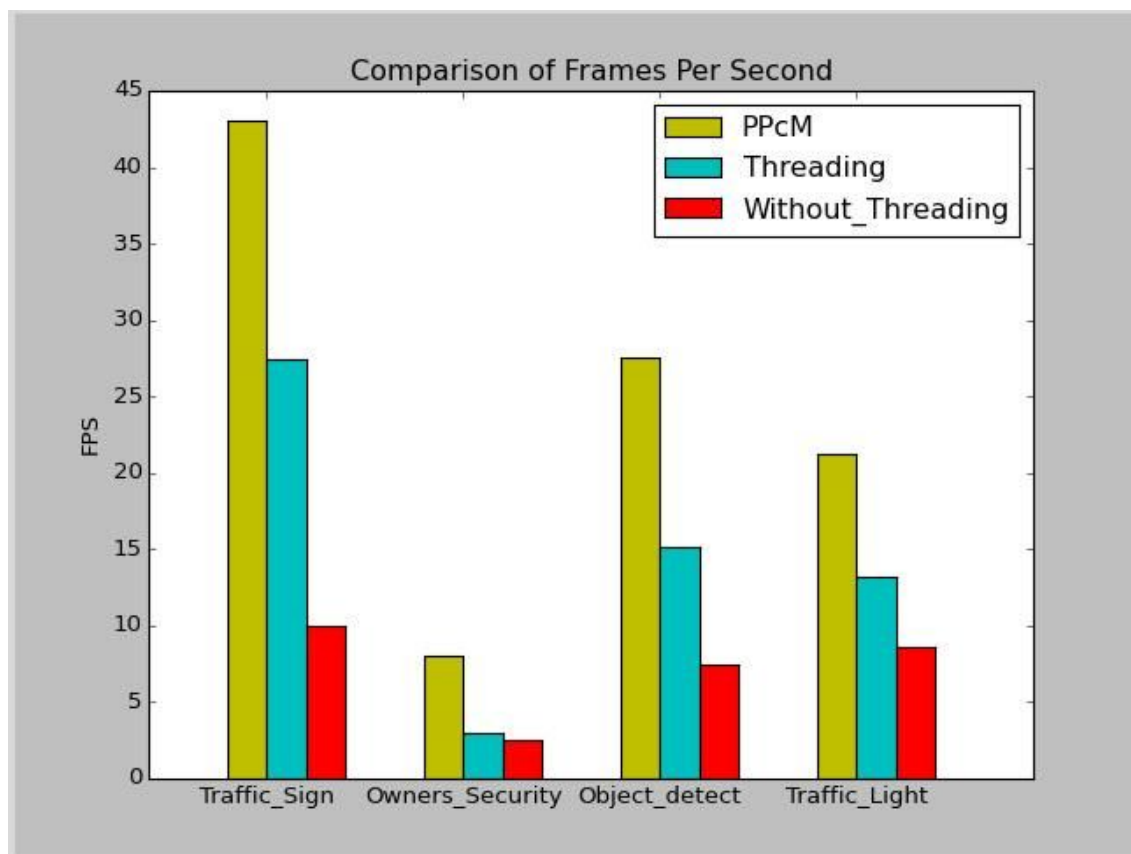


Fig- 4.3.1 Graph of the comparison of frames per second

4.3.2 Processing Time

For processing time, the less amount of time a process take, that process will be more time efficient. Time is the most important factor in image processing as the whole system is depend on time accuracy. And high latency can cause failure of the overall system. In the following bar chart we clearly can differentiate the processing time and also come to the conclusion that PPcM takes the lowest time for processing.

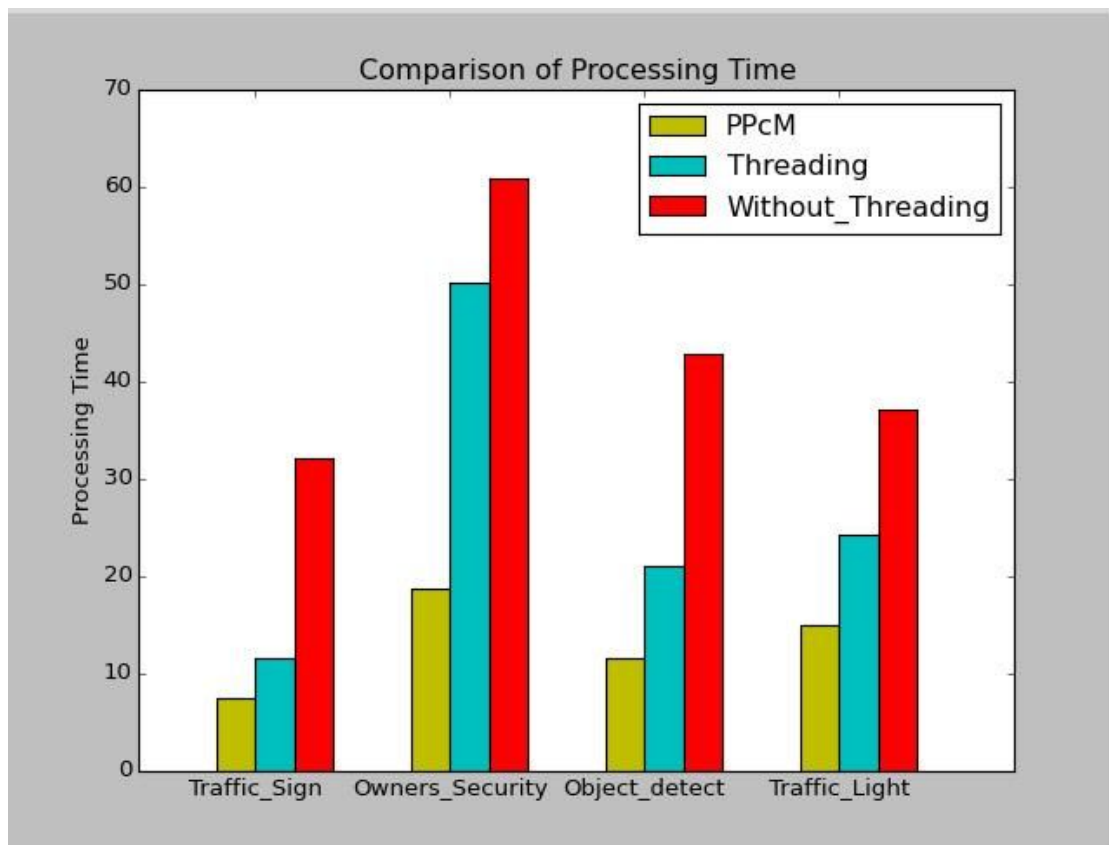


Fig- 4.3.2 Graph of the comparison of processing

Chapter 5

Limitation

5.1 Limitation

There are some problems we faced for which we have some limitations in our system. As we need to implement our algorithm in hardware setting, so any problem or specification in hardware caused the limitation basically, and some also due to time limitations that we have for the research. The limitations that we have in our system is given below-

- 1. Resolution of the Pi camera:** The resolution and camera features of Pi camera and a webcam is not same. So, there are some features of our system that works perfectly for webcam. But since we are making an advance autonomous car using PPCM, we need a proper microcontroller for that. So, we used Raspberry Pi as mentioned earlier. Pi camera was giving a little bit different result compared to webcam. So, again we need to change the code according to pi camera. So, Pi camera's resolution, intensity is a limitation.
- 2. Memory of Raspberry pi:** The short memory of raspberry pi sometimes create an issue while storing many data or images as we need to store many data and images for features like, traffic sign and owner's security. . And also while running, sometimes it slow down its overall process because of having long execution.
- 3. Space complexity:** As we have mentioned in the whole paper, we basically minimize the latency and focused on processing time. Due to lack of time, we could not consider the space .

Chapter 6

Future Work

6.1 Future Work

As we specified in limitation that we could not consider the space due to lack of time. Actually we did a lot of research on space efficiency of image processing but could not implement it because of time. So, we want to implement it in our further and future work of this research. When we consider the space of image processing image compression is the most used and popular element to discuss. The aim of compression is to reduce the size of the image which will save the space[21]. There are some famous algorithm which gives good result. Like huffman coding which basically encode and decode the image according to need. Fractal algorithm [22] which encode the image and gives a better compression ratio. SPIHT algorithm [23] which also gives a good compression ratio and maintain a good image compression quality. Whereas, DWT plays an important role to compress the given image without the loss of any information in that particular image [22]. So, there are many image compression algorithm. Some work for compression ratio, some for better compressed image quality or some give better peak signal to noise ratio. Our future goal is to constitute our own compression algorithm which will work gracefully for most of the parameters. And all in all our algorithm will give a very efficient result on account of both time and space in image processing.

Chapter 7

Conclusion

7.1 Conclusion

In our research we have shown that our procedure PPcM gives a better result on the basis of fps and processing time than multithreading and obviously without threading. PPcM increases the number of fps which fine tune and smoothen the steaming of the video and ensure low latency as there is no frame loss. As well as, it decreases the processing time and the algorithm gives a faster result. We also build a prototype of advanced, autonomous transportation system using image processing to effectively prove our result which has been shown in the images of terminal output and bar charts in the previous chapter. With the help of future work on space, we hope that we can conduct our work on image processing with more precision.

References

- [1] Anna Maria Vegni, Mauro Biagi and Roberto Cusani, "Smart Vehicles, Technologies and Main Applications in Vehicular Ad hoc Networks ", Vegni et al Intech, 2013. [Online]
Available: <https://cdn.intechopen.com/pdfs-wm/42787.pdf>
- [2] Ashna Jain, Harshitha Reddy and Sarthak Dubey, "Automated Driving Vehicle Using Image Processing ", International Journal of Computer Science and Engineering, Volume-2, Issue-4, pp.138-140, 2014.
- [3] Shih-Chia Huang, Bo-Hao Chen, Sheng-kai Chou, "Smart Car", IEEE Computational Intelligence Magazine, Volume: 11, Issue: 4, Nov. 2016
- [4] Slavomir Matuska *, Robert Hudec _ and Miroslav Benco, "The Comparison of CPU Time Consumption for image Processing Algorithm in Matlab and OpenCV"
- [5] Nidhi, "Image Processing and Object Detection", International Journal of Applied Research 2015; 1(9): 396-399
- [6] K.S.Shilpashree, Loksha, Hadimani Shivkumar," Implementation of Image Processing on Raspberry Pi", International Journal of Advanced Research in Computer and Communication Engineering, *Vol. 4*, Issue 5, May 2015
- [7] Ajinkya Patil, Mrudang Shukla Mtech, " Implementation of classroom attendance system based on face recognition in class ", International Journal of Advances in Engineering & Technology, July, 2014
- [8] Joel C. Adams, Patrick A. Crain, Christopher P. Dilley, "Seeing Multithreaded Behavior Using TSGL", 2016 IEEE International Parallel and Distributed Processing Symposium Workshops
- [9] Samyan Q., Sahar, Talha, Aslam M., "Real Time Digital Image Processing Using Point Operations in Multithreaded Systems", 2015 Fourteenth Mexican International Conference on Artificial Intelligence .

[10] Introduction to Image Processing, Engineers Garage. [Online]

Available: <https://www.engineersgarage.com/articles/image-processing-tutorial-applications>.

[11] H. K. Anusuya Devi, "Thresholding: A Pixel-Level Image Processing Methodology Preprocessing Technique for an OCR System for the Brahmi Script ", Ancient Asia, Journal of the society of south Asian Archaeology.

[12] Bryan S. Morse, Lecture 4: Thresholding, Brigham Young University, 1998–2000. [Online]

Available:http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf

[13] Jason Corso, "Linear Filtering and Image Processing". [Online]

Available: web.eecs.umich.edu/~jjcorso/t/598F14

[14] Contour Tracing , Image Processing Place. [Online]

Available:

http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/index.html

[15] RGB to Grayscale Conversion, Tutorials Point. [Online]

Available: https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm

[16] Gaussian Blur Tutorial. [Online]

Available: <https://docs.gimp.org/en/plugin-gauss.html>

[17] Shamik Sural, Gang Qian and Sakti Pramanik, "Segmentation and Histogram Generation Using the Hsv Color Space for Image Retrieval", IEEE ICIP, vol 2, pp. 589-592, 2002.

[18] Understanding Basic Multithreading Concepts, Oracle, 2010. [Online]

Available: <https://docs.oracle.com/cd/E19455-01/806-5257/6je9h032e/index.html>

[19] Raspberry Pi 3 is the first with built-in Wi-Fi, WIRED. [Online]

Available:

<http://www.wired.co.uk/article/raspberry-pi-three-wifi-bluetooth-release-price-cost>

[20] Buck Converters, Learn about Electronics and Power supplies, February 2017[Online]

Available: <http://www.learnabout-electronics.org/PSU/psu31.php>

[21] G. Epshi, V. Saravanan and B.Venkatalakshmi, " Design of Effective Low Power Image Compression Algorithm", IEEE WiSPNET, volume 16, pp. 2197-2200, 2016

[22] R.Praisline Jasmi, Mr.B.Perumal and Dr.M.Pallikonda Rajasekaran," Comparison of Image Compression Techniques using Huffman Coding, Dwt and Fractal Algorithm ", 2015 International Conference on Computer Communication and Informatics (ICCCI -2015), Jan. 08 – 10, 2015, Coimbatore, India

[23] K.Chithra, Mr.B.Perumal, Dr.M.Pallikonda Rajasekaran and Mr. T.Arun Prasath, " A Quantitative Assessment of Image Compression Parameters and its Algorithm ", Proceedings of 2015 Global Conference on Communication Technologies(GCCT 2015)