MATLAB
Automatic Car Parking Indicator System

Yash Rana
19BCE220
Institute of Technology Nirma University
19bce220@nirmauni.ac.in

***Abstract*—Nowadays, we are already surrounded by the Gadgets, technology, new inventions and so on. There are so many tasks that have been done and functioned by the online applications as well as smart tools known as computerized systems. Now you might have seen many cars in shopping malls, theatres and at any other gathering area. So to reduce the human efforts, what can we do by the means of computerized detection. So, through the detection of the car parking slots, we can easily identify the empty slots faster. Hence, the main objective of this project is to detect the cars and find out the empty slots with fewer observations. Basically, we will be fixing the cameras facing towards the parking area, taking a number of images, storing it, and detecting it in real time. But here we have used only a number of images to detect the empty slots. So, here we go...**

I. INTRODUCTION

This paper is about the car parking system using the MATLAB software. This software detects the total available slots in the parking area. Thus, it aids the driver to know if he will be able to park the car in that parking area or not. Nowadays, cars have become an integral part of our lives. As more and more families are moving to urban areas, the need for a car arises for them. Expanding cities leading to more automobiles and thus exponentially increasing demand for parking areas. Parking is a tedious as well as time-consuming task. In the busy life of an urban employee, he/she can't spend very long time just waiting to park the car and sometimes getting disappointed by the fact that parking is full. So, what if we place one screen that tells us where the empty slot is available to park or even the security guard can tell us the same by looking at the screen easily. By this, we come to know how many total parking slots are available. So, we have used MATLAB's Image Processing Toolbox to analyze, detect and build a code for the Automatic Car Parking Indicator System.

II. BASIC-FLOW DIAGRAM

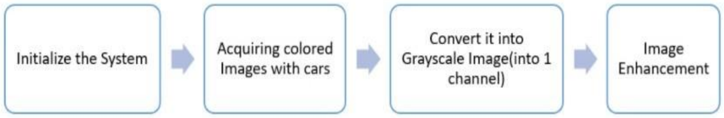
Initialize the System
Acquiring colored Images with cars
Convert it into Grayscale Image(into 1 channel)
Image Enhancement

Fig. 1. Basic-Flow-Diagram

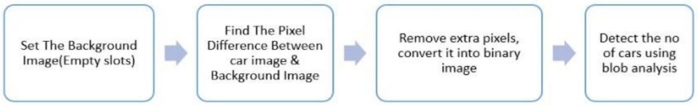
Set The Background Image(Empty slots)
Find The Pixel Difference Between car image & Background Image
Remove extra pixels, convert it into binary image
Detect the no of cars using blob analysis

Fig. 2. Basic-Flow-Diagram

And at the end, display the output of how many total cars have been detected as well as how many empty slots are available for the next person to park his car.

III. IN DEPTH METHODOLOGY

A. Program Initialization

In this part, we initialize the code by providing an image of an empty parking area. This acts as a reference when we try to detect empty and full slots in the parking area.

We will try to develop the MATLAB code for the initialization part. Also, we have widely used the Image-Processing Toolbox.

B. Acquiring the Images and conversion part

Image captured by a high-definition camera is uploaded to the MATLAB program. Image will then be processed for further use. Here, we are going to provide or feed the images with cars parked.

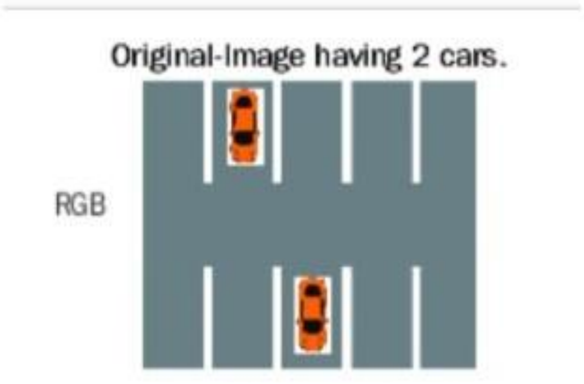

Fig. 3. Colored Image

Now, to analyze that image, first we need to convert all the images into the grayscale image.

The real image is in the form of RGB format having 3 different channels, but we need to convert it first into a grayscale image which will be in the 1 channel format having the range of [0,255].

0 indicates black and 255 indicates white color.

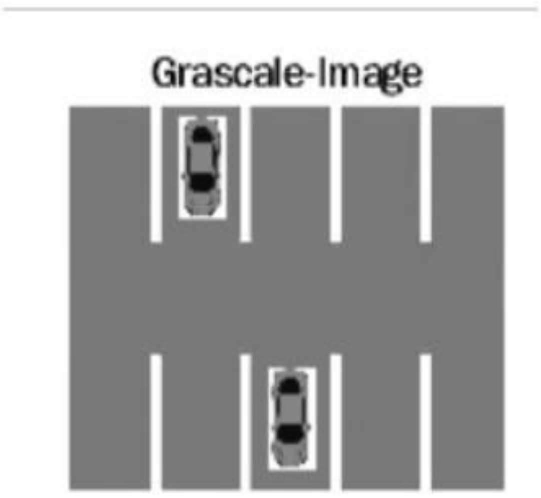

Fig. 4. Gray-Scale Image

After performing first 2 steps, these are the output images that we are getting. As you can see from the above picture, it has now been converted from RGB to Grayscale image.

C. Setting the Background image

Now, to detect the cars location, we have come up with an idea of the pixel differentiation. The logic is we try to find out the Pixels difference between the Background image (Having No cars) with images having cars. whenever it finds out the car the pixel value is totally different from the rest. Therefore, to implement this idea, we need to perform these many below steps,

- Import the background image.
- Both the images should have the same size, if not then make it similar.
- Convert the image into Grayscale Image.
- Perform the difference between these 2 images.

This is how the background image looks(After converting it into a grayscale image).

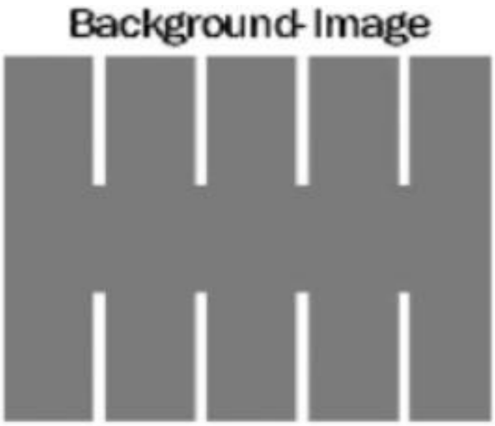

Fig. 5. Background-Image

D. Image enhancement and binary image

Here, suppose if we find any blur or hazy part in the image, then we will remove(dehazed) by imreducehaze method. Now, at last we need to convert our grayscale image into a binary image (0: black,1: white), by deciding the threshold value. To determine the threshold value, we have plotted a histogram of the image.
As you can see from the above histogram, we can set our threshold value as 50. And also by looking at the plot, we

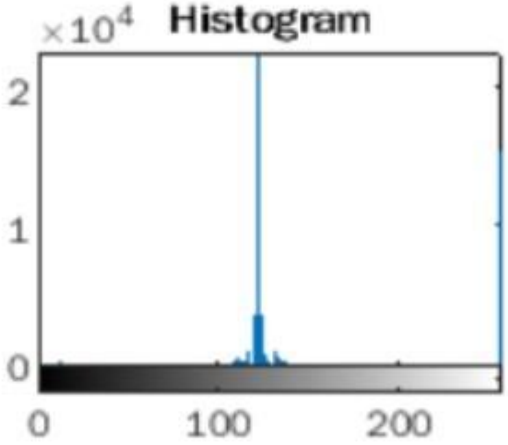

Fig. 6. Histogram

come to know that you will get an abundant number of pixels in the range of 100-130. Binary Image:

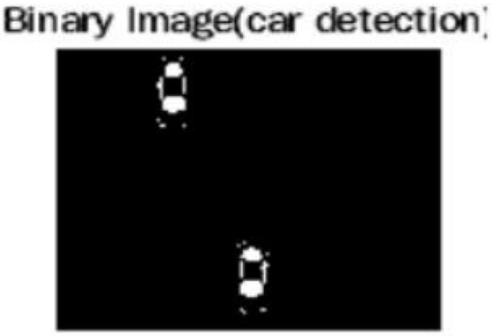

Fig. 7. Binary Image

E. Blob analysis

So, we have already converted it into a binary image. Now it's time to make the distinction. This basically finds the borderline between the object and the background image. Hence it helps us locate the car in the parking area.
have applied number of methods like,

bwareaopen (image, pixel): It will remove the small objects from the binary image which are having less than p pixels, here we have kept as 500.

bwlabel: label connected components in 2-D binary-image.

region props: measure the properties of the image of different regions.

F. Detection

After the blob analysis defines a distinct color for cars and background,
We can detect the car and put it separate from the empty image.
We are going to plot the blob-analyzed image with detection of cars also we will print out how many parking slots are available, actually over here the process goes like as follows.

Algorithm

- First, we have captured the images.
- Then, we have set the background Image.
- Make the same size of both the images.
- Transforming into Gray-scale-Images.
- Try to detect edges of the car.
- Removing the haze and other small objects.
- Based on the threshold value, Converted into binary image.
- Separate it with connected edges.
- Using blob analysis technique, detect number of empty slots.

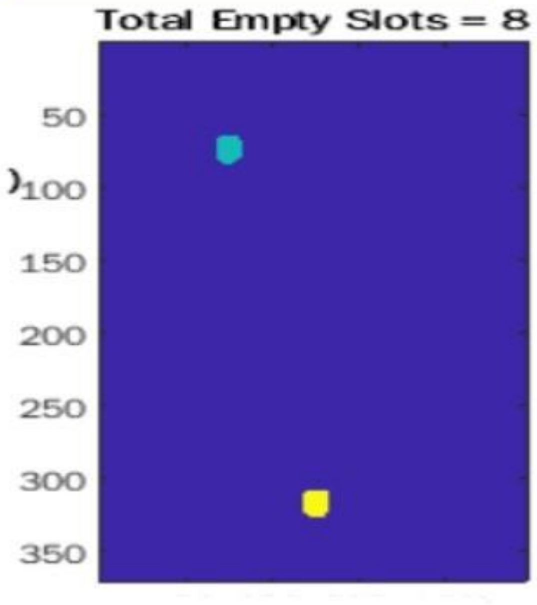

Fig. 8. Detected-Image

IV. Conclusion

The main motive to do this project was to perform image processing and detection. This project has been put to implementation using MATLAB software. People wishing to park in the area can get a real time situation of the parking lot. Situations where there are multiple parking areas in close proximity, the parking (if the use the same program), might be able to display each other's parking slots and redirect the traffic to close proper parking. Hence the mail target of this project, which was to help people save time by directing them to the closest parking spot possible by showing them the live car parking situation, has been fulfilled.

V. References

[1] https://in.mathworks.com/help/images/color.html

[2] https://in.mathworks.com/discovery/image-recognition-matlab.html

[3] https://in.mathworks.com/help/vision/object-detection.html

[4] https://in.mathworks.com/help/vision/ref/blobanalysis.html

[5] https://www.youtube.com/watch?v=UoRwjP-bWjA

VI. Screenshots-Output:(sample)

Detection of 2 cars (Having 8 slots free-Available)

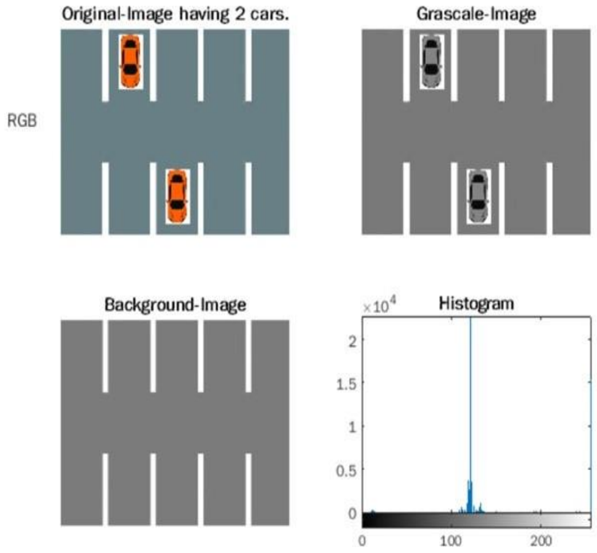

Fig. 9. Images

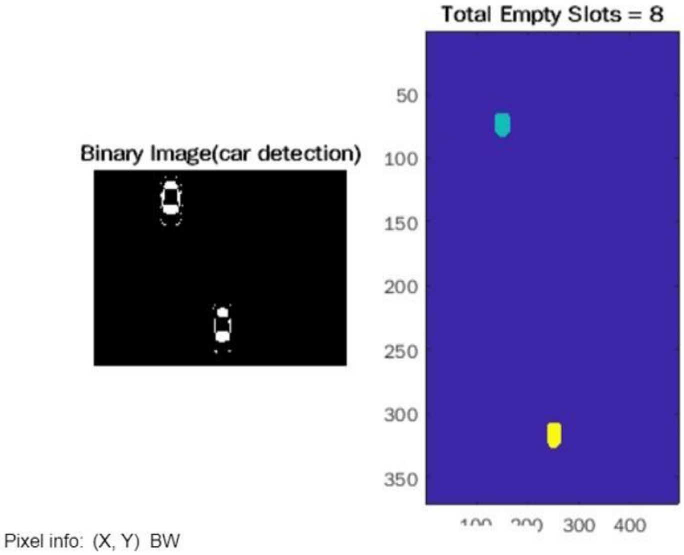

Fig. 10. Detection