Parking Spot Detection using Image Processing in MATLAB

1st Aman Gupta
19105007
Electronics and Communication Dept.

2nd Aman Bhatt
19105013
Electronics and Communication Dept.

3rd Shubham Kaushik
19105014
Electronics and Communication Dept.

4th Ayush Sharma
19105016
Electronics and Comm. Dept.

5th Raghav Bansal
19105018
Electronics and Comm. Dept.

6th Vibhanshu Kaushal
19105036
Electronics and Comm. Dept.

7th Aditya Pandey
19105097
Electronics and Comm. Dept.

***Abstract*—This is a software that has been created and tested using MATLAB. The objective of the project is basically to use image processing to identify or detect parking spots in a parking lot. It uses multiple steps from reading the image to giving total number of parking spots available. It can be implemented on a higher level using cameras covering the parking lot to acquire images of the lot which can be processed further in the software. This project can be a great economic alternative option for parking detection .**

***Index Terms*—Smart Parking, Parking Spots Detection, Image Processing, Image Segmentation, Image Collection**

I. Introduction

As the number of vehicles are increasing day by day the need of parking spots is also increasing rapidly. To deal with this issue large parking lots are being constructed. Specifically in metropolitan cities multi storey parking are becoming common but managing these parking lot is not an easy task as it is not humanly possible to keep a track of total number of cars in the parking lot. According to a survey conducted by GLG Surveys most of the people spend around 15 minutes of their time in a day on an average and wastes around half a liter of fuel just to find a vacant parking spot . This might seem like a small number but if we compute this wastage of fuel and time for a month , for a year then we will truly get the overall scenario of the situation..

Therefore , finding a parking spot is one of the major problem in public places. Traffic jams are caused due to wrong car parking. Due to limited space a number of cars entering at a time may cause problem in full parking spots. In order to solve this problem we are proposing a system that can analyse the parking area based on the image processing technique using MATLAB. Then , It will be able to identify that how many parking spaces are available in the parking area. Accordingly , the driver will be conveyed about the available space and it will save time and increase the affectivity. Irrelevant vehicle crowding to find a parking place would be reduced.

II. Methodology and Framework

This system will consist of acquiring the image of the empty parking lot and subsequently acquiring image of the parking lot at the time of interest. These two images will go through the segmentation process. Afterwards the images with the cars will be passed through image enhancement steps to reduce noise. Total vehicles will then be detected and number of empty spots will be displayed.

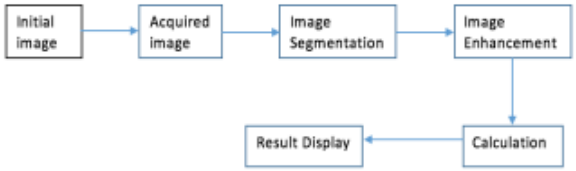

Fig. 1. Block Diagram of System.

A. Obtaining the one-time input

We will acquire the initial parking lot image with no cars. This image is used as a reference image in the next steps. The image has been loaded inside MATLAB using the “imread” function.

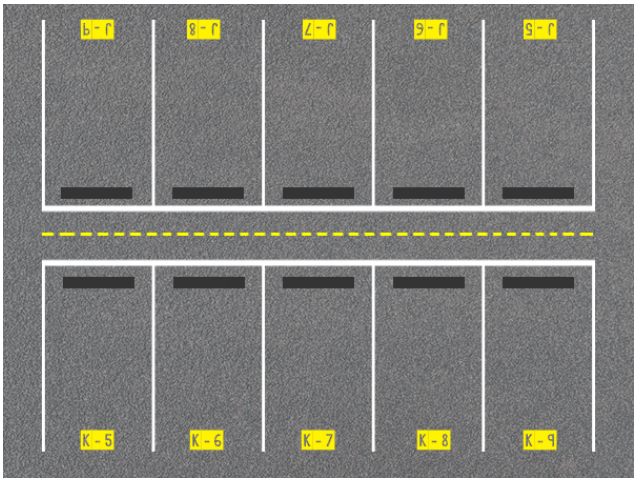

Fig. 2. Initial image with empty parking lot

```
%% read empty parking space

empty_img= imread('NoCarParked.jpg');
empty_img= imresize(empty_img,[400 nan]);
```

Retrieve image for empty parking lot

B. Obtaining the input image

An image of the parking lot is taken on time of interest. It is crucial that this image is taken from the exact same position as the initial image. The image is loaded inside MATLAB using "imread". Both of the input images are resized to predetermined dimensions.

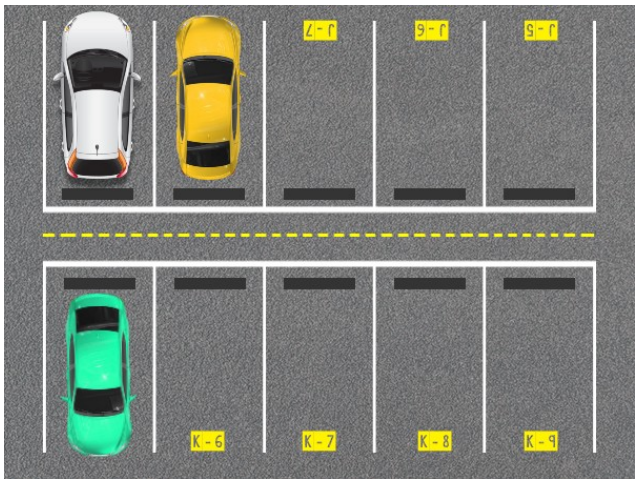

Fig. 3. Image of some cars parked in parking lot

```
%% read image

raw_img = imread('ThreeCarsParked.jpg');
raw_img = imresize(raw_img,[400 nan]);
```

Retrieve raw image for parked cars

C. Image Subtraction:

The initial image of the parking lot is now subtracted from the input image using the "imsubtract" function. This outputs a resultant image which is rid of all the unwanted details such as parking spot separator lines, cones etc.

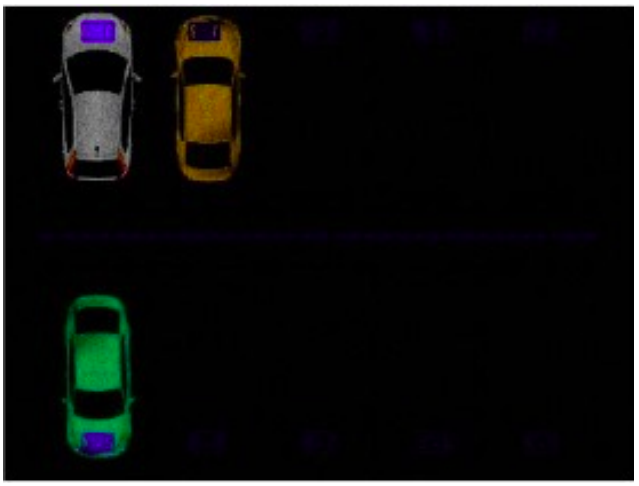

Fig. 4. Subtraction Result

```
%% subtract to get rid of parking lines

img = imsubtract(raw_img,empty_img);
```

Image Subtraction

D. Image Segmentation:

The input image is RGB. So it is converted to grayscale using "rgb2gray" function. This eliminates the hue and saturation data and only preserves the luminance data of the image. Image segmentation is used to simplify and change the representation of the image to make it easier to analyze. We use thresholding technique to segment our image. The image is converted from grayscale to binary "im2bw" function. The selected threshold value is 0.02.

Fig. 5. GrayScale Image

```
%% convert from RGB to grayscale

igray = rgb2gray(img);
```

Image RGB to Grayscale Operation

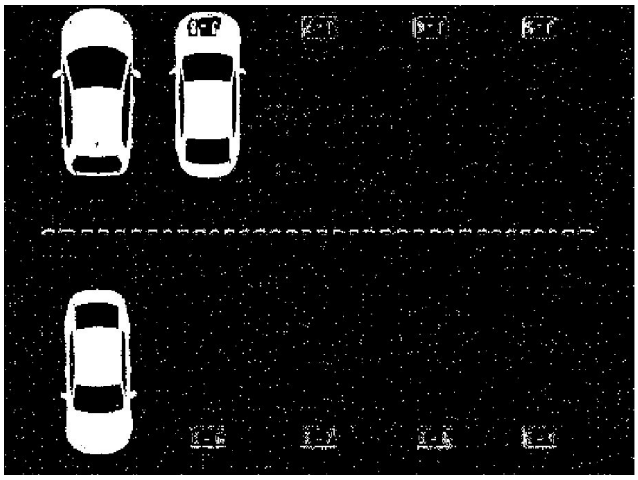

Fig. 6. Binary Image

```
%% convert from grayscale to binary with
    specified threshold

level = 0.02;
ithresh = im2bw(igray,level);
```

Image Thresholding Operation

E. Image Enhancement:

After the segmentation process the image consists of 'holes' and noise. To remove these we will use image enhancement techniques known as Morphological Transformations. Specifically, we will use dilation and opening operator i.e erosion and then dilation. In the erosion step, the pixels white pixels are eroded using a strel object to reduce noise. In the dilation step, white pixels are added using the strel object to fill the holes, such as the ones caused due to windscreens of the vehicles. This is done using the "imfill" and "imopen" functions. This will eliminate the noise and the output image will contain white objects.

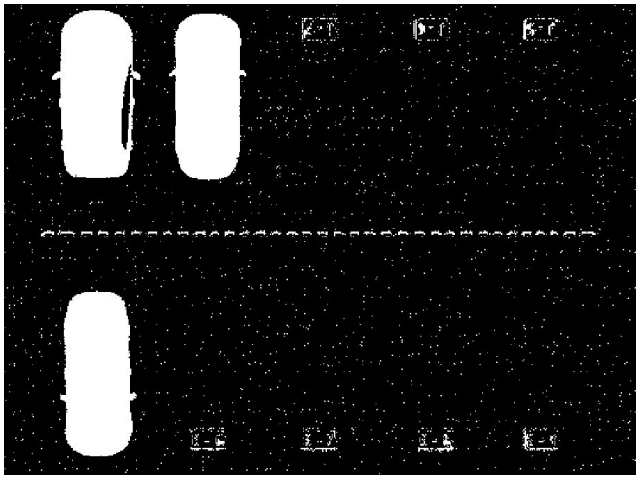

Fig. 7. Image Holes Filled

```
%% fill holes

ifilled = imfill(ithresh,'holes');
```

Image Holes filling Operation

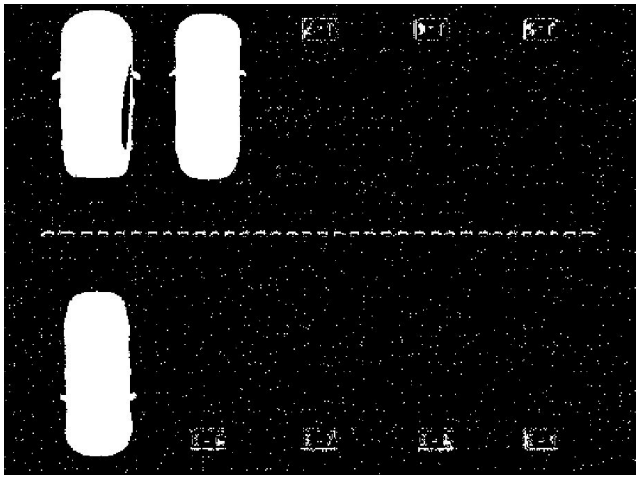

Fig. 8. Image Structured Elements and Opening

```
%% shaping every object with specified
    disk size

se = strel('disk',5);
iopen = imopen(ifilled,se);
```

Image Structuring and Opening Operation

Fig. 9. Image Small Objects Removed

```
%% remove small unnecessary objects

iclear = bwareaopen(iopen,4,4);
```

Removing small objects Operation

F. Bounding Box detection:

In this step, exterior boundaries are created on the input images using data from the morphologically transformed image. We use "regionprops" to achieve this. It returns measurements for various properties of the objects in our binary image.

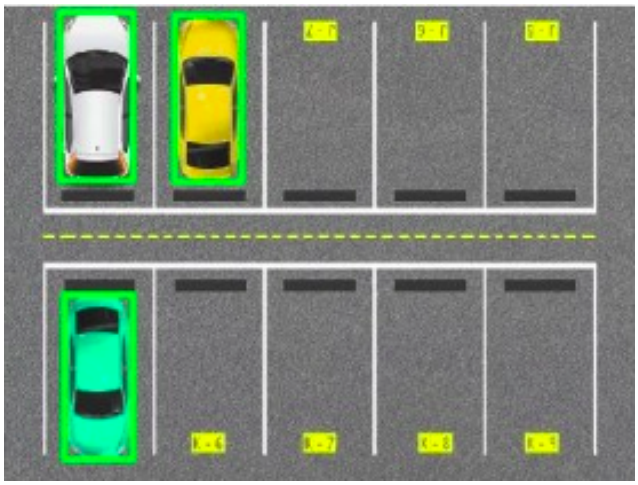

Fig. 10. Boundary Boxes for Detected Cars

```
%% making boundary boxes around detected
    cars

imshow(raw_img),title('boundary boxes');
props = regionprops(iclear,'BoundingBox',
    'Image');
for i=1:size(props,1)
rectangle('Position',props(i).BoundingBox
    ,'EdgeColor','g','LineWidth',2);
end
```

Detecting Cars and making Boundary Boxes

G. Displaying the output:

The total number of 'objects' are the total amount of vehicles from the image. Thus number of empty parking spots are (total parking spots - number of vehicles). This result is displayed using the "msgbox" function. This will create a dialogue box which displays the output.

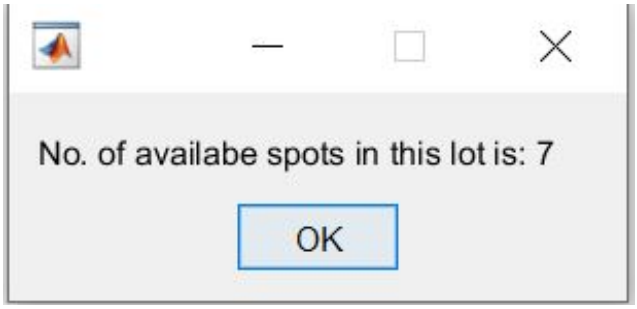

Fig. 11. Displaying Empty Spots in a Dialogbox

```
%% count no. of cars and print available
    spots

dialog = msgbox(['No. of availabe spots
    in this lot is: ',num2str(10-i)])
```

Counting Empty Parking Spots Operation

III. System Algorithm

The basic phases of the proposed algorithm for image processing-based parking lot detection. This is basically all the steps that are done by the software which are explained above in detail.

Use a dialogue box to ask the user for the number of spaces in the lot.

Take a picture of an empty parking spot, which is known as the starting image.

Take a photo of the car-filled parking lot.

Subtract the occupied and vacant parking images. Image segmentation will be used to process the image result as follows:

- Create a grayscale image from the TrueColor RGB image.
- Convert a grayscale image to a binary image at the specified level
- Reduce the noise in the figure using dilation ,erosion etc .
- As image is clear of all the noise , image region measuring bounding box is created for detected automobiles.
- These bounding box are examined and final image of parking spots occupied is given.
- Then these identified number of automobiles are subtracted from the total number of given as input and final number of available slots are displayed.

IV. Experimental Results

The Software was tested by giving dummy images of filled and empty parking lots. All the outputs given by the software were as we desired. The software successfully read all the images and was able to detect the number of available parking lots . The software was tested with multiple images . You can see the final output images provided below.

V. Conclusion

MATLAB was used to create and evaluate smart parking space detection systems based on image processing. Various types of image processing and segmentation were used in the process, allowing the system to learn and count items. In comparison to other systems that require more electronic hardware, such as sensors, we believe this is a more efficient solution. The system was put to the test with photos of fake cars and a parking lot. The system was able to detect the number of cars in various testing settings, which proved to be successful. The main goal of this research is to demonstrate a system that can count available parking spots in a parking lot. This method will assist you.

This Software in future can be linked to cameras which will feed the software with real time images and the software can detect the available spots in real time. This can be used for example in malls , a system can be attached with ticket machine that whenever we are taking a ticket the software analyzes the parking lot and gives you the floor number and parking number which is available as an output.