PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100 Ft. Road, BSK III Stage, Bengaluru – 560 085

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Title: Image Processing and Data Visualization Using MATLAB		
Course code: -UE19CS257B		
Semester : 4th sem	**Branch:** CSE	**Team Id:** 57
SRN: PES1UG19CS491	**Name:** Snigdha Sinha	
SRN: PES1UG19CS592	**Name:** Yashi Chawla	

PROJECT REPORT

Problem Statement:

To detect number of free parking spaces in a parking lot using image processing, by comparing images of an empty parking lot as reference against the image of a parking lot with cars.

Objectives:

To obtain the correct number of free parking spaces and thus help decrease traffic congestion in a parking lot.

Description:

We aim to build a system for the detection of parking space with the help of image processing technique. The system will detect cars through images instead of electronic sensors. A camera is installed at a high and fixed position in the parking lot. An image of empty parking lot will be taken as reference and then an image of parking lot with cars will be taken. Both the images will be subtracted to find the numbers of parking slots available. Therefore, the system will help in counting the number of parked vehicles and, identifying the number of spots available.

New Concept Learnt(Explanation):

Imfill: a function that performs flood fill operation on the background pixels of the input binary image bw. **Imfill(BW, ”holes”)** has been used in this project implementation, which fills holes in the binary image BW, wherein a hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image.

Blob Analysis

The method of analyzing an image that has undergone binarization processing is called "blob analysis". A blob refers to a lump. Blob analysis is image processing's most basic method for analyzing the shape features of an object, such as the presence, number, area, position, length, and direction of lumps. It calculates statistics for labelled regions in a binary image. The block returns quantities such as the centroid, bounding box, label matrix, and blob count.

Learning Outcome:

- While researching for this project, we learnt about the various domains and applications of Image Processing.
- We learnt about the different steps involved in Image Processing, such as:
    1. Adjusting Image Intensity Values using ‘imadjust’
    2. Analysing images using thresholding techniques
    3. Filtering the image using Weiner Filter
    4. Blob detection
- We learnt about the various applications of Blob Analysis, such as:
    1. Analysing the shape features of an object
    2. Calculating the statistics for labelled regions in a binary image
    3. Detecting the presence or absence of lumps in the processed image
    4. Detecting the count and size of objects which are to be inspected in the image

Code:

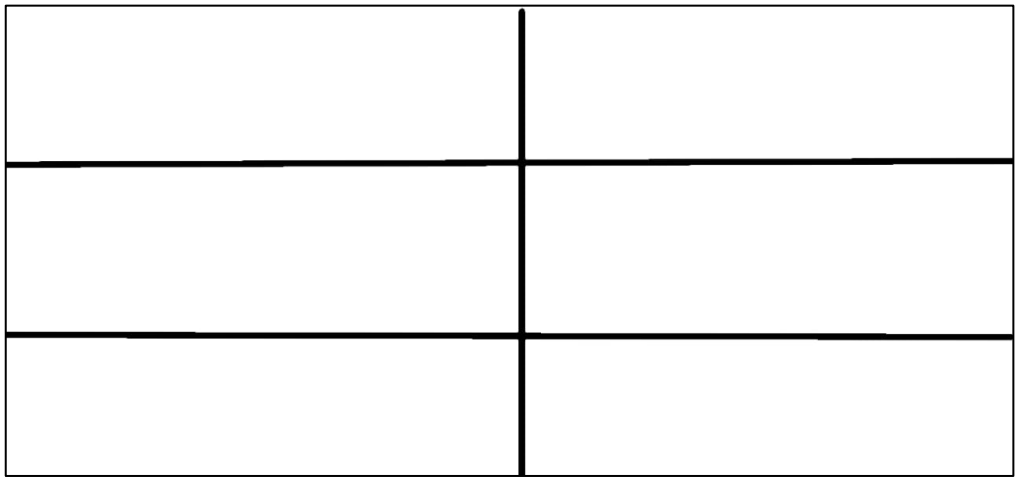
```
image = imread('D:\Snigdha\2nd year\4th sem\SpecialTopics\Matlab
project\Main Project\car-2.jpg');
background = imread('D:\Snigdha\2nd year\4th sem\SpecialTopics\Matlab
project\Main Project\bg.jpg');
img = double(rgb2gray(image));%convert to gray
bg = double(rgb2gray(background));%convert 2nd image to gray
[ht, wdth] = size(img); %image size?
slot_total=36;    %Given Total number of slot in the parking area.
%Foreground Detection
threshold=11;
diff = abs(img-bg);
for x = 1:wdth
for z = 1:ht
if (diff(z,x)>threshold)
fground(z,x) = img(z,x);
else
fground(z,x) = 0;
end
end
end
subplot(2,2,1) , imshow(image), title (sprintf('Parking Area with %d
slots (The original frame)',slot_total));
subplot(2,2,2) , imshow(mat2gray(img)), title ('Converted Frame');
subplot(2,2,3) , imshow(mat2gray(bg)), title ('Background Frame ');

adj=imadjust(fground);% adjust the image intensity values to the color
map
level=graythresh(adj);
nse=imnoise(adj,'gaussian',0,0.025);% apply Gaussian noise
flt=wiener2(nse,[5,5]);%filtering using Weiner filter
bw=im2bw(flt,level);
fill_holes=imfill(bw,'holes');
open = bwareaopen(fill_holes,5000);
labeled = bwlabel(open,8);
blobs = regionprops(labeled,'all');
cars_total = size(blobs, 1);
subplot(2,2,4) , imagesc(labeled), title (sprintf('(Foreground) Total
space available is %d',slot_total-cars_total));
hold off;


%{
CONDITION TO CHECK THE VACANT SPACE
IF YES then it divide it into 6 parts as 6 LANEs are there
then for each lane image processing is applied as before and lane with
vacant space comes first with their space.
LANE number are like
   LANE 1       LANE 2
   LANE 3       LANE 4
   LANE 5       LANE 6
%}

 if((slot_total-cars_total)>0);
  fprintf('You can enter into the parking area');
  fprintf('\n Total number of cars present');
```

```
  disp(cars_total);% display number of cars
  fprintf('Total number of vacant spaces present');
  disp(slot_total-cars_total);
  fprintf('PARKING AREA STRUCTURE:- \n LANE 1\t\t LANE 2 \n LANE 3\t\t
LANE4 \n LANE 5\t\t LANE 6');

%dividing the image of the parking area into 6 parts as 3 rows and 2
coloums.
rows=int32(ht/3);
columns=int32(wdth/2);
img_1=img(1:rows,1:columns);
img_2=img(1:rows,columns+1:end);
img_3=img(1+rows:2*rows,1:columns);
img_4=img(1+rows:2*rows,columns+1:end);
img_5=img(2*rows+1:end,1:columns);
img_6=img(2*rows+1:end,columns+1:end);
new_img={img_1 img_2 img_3 img_4 img_5 img_6};   %An Array to store 6
images.
bg_1=bg(1:rows,1:columns);
bg_2=bg(1:rows,columns+1:end);
bg_3=bg(1+rows:2*rows,1:columns);
bg_4=bg(1+rows:2*rows,columns+1:end);
bg_5=bg(2*rows+1:end,1:columns);
bg_6=bg(2*rows+1:end,columns+1:end);
new_bg={bg_1 bg_2 bg_3 bg_4 bg_5 bg_6};

%{
LOOP is taken from LANE 6 to LANE 1.
And the previous process is repeated for each lane for vacant space
detection.
%}
for i=6:-1:1
 lane_total=6;
 img_lane=new_img{i};
 bg_lane=new_bg{i};
 [ht_lane,wdth_lane] = size(img_lane);
 threshold_lane=11;
 diff_new = abs(img_lane-bg_lane);
 for x = 1:wdth_lane
  for z = 1:ht_lane
   if (diff_new(z,x)>threshold_lane)
    fg_new(z,x) = img_lane(z,x);
   else
    fg_new(z,x) = 0;
   end
  end
 end

 adj_lane=imadjust(fg_new);% adjust the image intensity values to the
color map
 level_new=graythresh(adj_lane);
 nse_new=imnoise(adj_lane,'gaussian',0,0.025);% apply Gaussian noise
 flt_new=wiener2(nse_new,[5,5]);%filtering using Weiner filter
 bw_new=im2bw(flt_new,level);
 fill_holes=imfill(bw_new,'holes');
 open = bwareaopen(fill_holes,5000);
 labeled_new = bwlabel(open,8);
 blobs_lane = regionprops(labeled_new,'all');
```

```
 cars_total_lane = size(blobs_lane, 1);
 %If lane is available with vacant slot then this statement is true and
car will get its direction.
 if((lane_total-cars_total_lane)>0)
     fprintf('\n \nGo to Lane %d',i);
     fprintf('\n Number of cars present: %d\n',cars_total_lane);
     fprintf('Number of vacant spaces present: %d\n',lane_total-
cars_total_lane);
     figure, subplot(2,2,1), imshow(image), title('Whole parking area');
     lane=sprintf('Lane %d is availabe with vacant slot',i);
     subplot(2,2,2) , imshow(mat2gray(img_lane)), title (lane);
     subplot(2,2,3) , imshow(mat2gray(bg_lane)), title ('Background Frame
');
     cars=sprintf('Total space available in lane %d is %d',i,lane_total-
cars_total_lane);
     subplot(2,2,4) , imagesc(labeled_new), title (cars);
     hold off;
 break;
 end
 end
 else
   fprintf('\n No space available in parking area.\n Exit'); % If parking
area is full then this statement executes.
 end
```

Output Screenshots

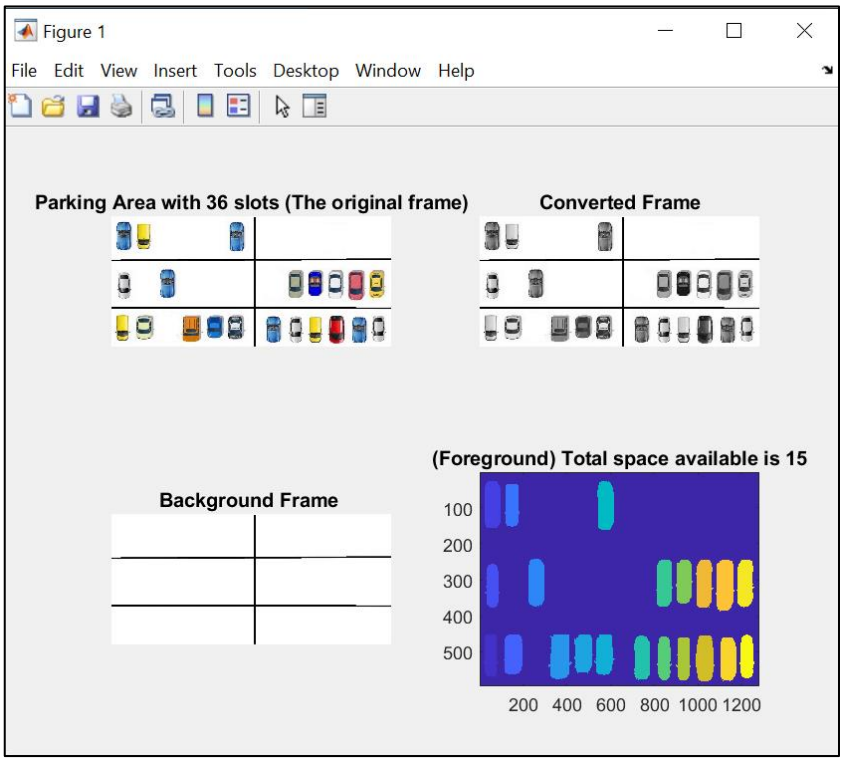
Input Image:



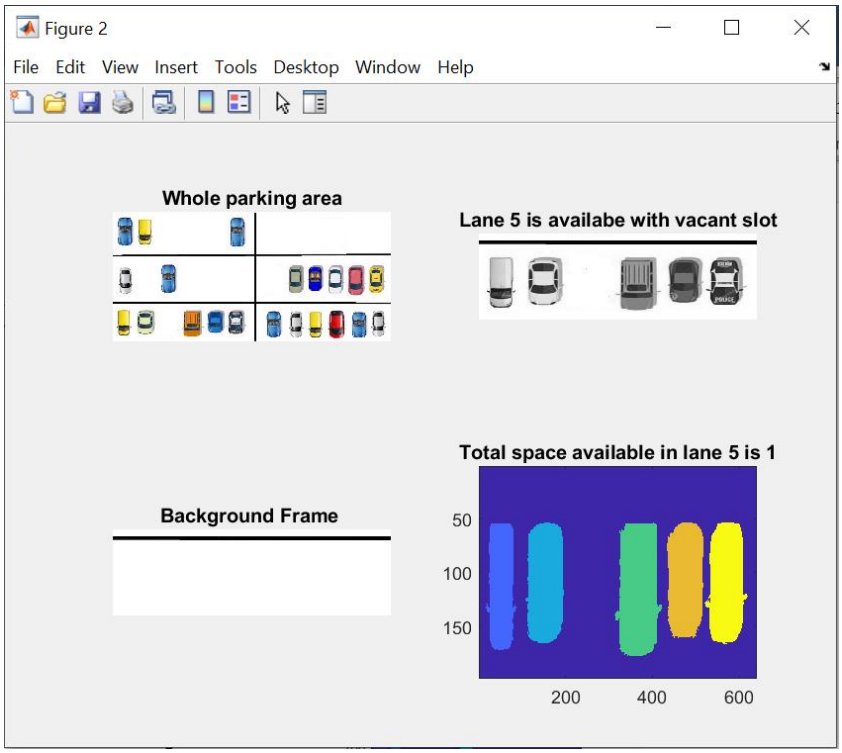
Empty Parking Lot

Parking Lot with cars

Output Image:



The figure shows: i) The Entire parking area
ii) The Converted Frame
iii) The background frame
iv) The total space available in the parking area



The figure shows: i) the image of the parking lot
ii) the lane which has available space
iii) the total space available in that lane

Name and Signature of the Faculty