

**CS 484, Fall 2017**  
**Homework Assignment 2: Edge Detection**  
**Due: November 29, 2017**

**Car Detection Function Code**

1. Read the image and converted it to greyscale[1][2].

```
function car_detect = car_detection(name)

cur_image = imread(name);
cur_image_greyscale = rgb2gray(cur_image);
```

2. Filtered the image with Gaussian Filter to remove noise

```
d_cur_img = imgaussfilt(d_cur_img);
```

3. Computed the horizontal and vertical edge responses (gradients) with the Sobel mask. Then, computed the directions and magnitudes of edge responses[2].

```
horizontal_gradient = d_cur_img;
vertical_gradient = d_cur_img;

for i=1:rows-2
    for j=1:cols-2
        % horizontal gradients with Sobel mask
        horizontal_gradient(i,j) = ((2*d_cur_img(i+2,j+1) + d_cur_img(i+2,j) +
d_cur_img(i+2,j+2)) - (2*d_cur_img(i,j+1) + d_cur_img(i,j) + d_cur_img(i,j+2)));
        % vertical gradients with Sobel mask
        vertical_gradient(i,j) = ((2*d_cur_img(i+1,j+2) + d_cur_img(i,j+2) +
d_cur_img(i+2,j+2)) - (2*d_cur_img(i+1,j) + d_cur_img(i,j) + d_cur_img(i+2,j)));
    end
end

% angles of the gradients
Gdir = atand(horizontal_gradient./vertical_gradient);
% magnitudes of the gradients
Gmag = sqrt(horizontal_gradient.^2 + vertical_gradient.^2);
% zeroing out the non-horizontal gradients
horizontal_gradients = ((Gdir > 80) & (Gdir < 110)) | ((Gdir < -80) & (Gdir > -110));
```

4. Summed up the number of horizontal edges column-wise[1]

```
% col-wise summation of number of horizontal edges
summed_up_horizontals = zeros(1, cols);
for j=1:cols
    for i=1:rows
        summed_up_horizontals(j) = summed_up_horizontals(j) +
horizontal_gradients(i,j);
    end
end
```

5. Smoothed the histogram of the horizontal edge numbers with morphological opening. Then, peaks were determined and they are set to be the vertical symmetry axes for cars[1].

```
se = strel('square', 40);

% morphological opening for image smoothing
summed_up_horizontals = imopen(summed_up_horizontals, se);

% determine important peaks
min_peak_height = max(summed_up_horizontals) - 11;
[peak_values,locations,widths,prominences] =
findpeaks(summed_up_horizontals,'MinPeakHeight',min_peak_height,
'MinPeakProminence',2);
[no_of_rows, no_of_cols] = size(locations);
```

6. Same process was applied row-wise to detect horizontal symmetry axes of cars[1].

```
% rowwise addition of horizontal edges
rowwise_summed_up_horizontals = zeros(1,rows);
for i=1:rows
    for j=1:cols
        rowwise_summed_up_horizontals(i) = rowwise_summed_up_horizontals(i) +
horizontal_gradients(i,j);
    end
end

% smooth the rowwise summed horizontal edge values histogram
se = strel('square',35);
rowwise_summed_up_horizontals = imopen(rowwise_summed_up_horizontals,se);

% determine important peaks
min_peak_height2 = max(rowwise_summed_up_horizontals) -
std(rowwise_summed_up_horizontals);
[peak_values2,locations2,widths2,prominences2] =
findpeaks(rowwise_summed_up_horizontals,'MinPeakHeight',min_peak_height2,
'MinPeakProminence',2);
if(numel(locations2) == 0)
    min_peak_height2 = max(rowwise_summed_up_horizontals) -
std(rowwise_summed_up_horizontals)*2;
    [peak_values2,locations2,widths2,prominences2] =
findpeaks(rowwise_summed_up_horizontals,'MinPeakHeight',min_peak_height2,
'MinPeakProminence',2);
end
[no_of_rows2, no_of_cols2] = size(locations2);
```

7. Rectangles were drawn on the original picture to show the locations of cars. Unnecessary symmetry axes were ignored.

```
%draw rectangles on the original image
hold on;
for i=1:no_of_cols
    if(bool == 1)
        for j=1:no_of_cols2
            cur_width = max(widths(i), widths2(j));
            if(no_of_cols2 >= 3)
                if(j == 2 ) %since there will not be any cars on top of eachother
                    rectangle('Position',[((locations(i)-widths(i)/2) (locations2(j)-
widths2(j)/2) cur_width*1.25 cur_width*1.25], 'EdgeColor','g', 'LineWidth', 3);
                    break;
                end
            elseif(no_of_cols2 == 2 && j < no_of_cols2)
                if(locations2(j) >= locations2(j+1)-widths2(j+1)/4 && locations2(j) <=
locations2(j+1)+widths2(j+1)/4) % number of horizontal symmetry axes equal to 2
(commonly found case)
                    rectangle('Position',[((locations(i)-widths(i)/2) (locations2(j)-
widths2(j)/2) cur_width*1.25 cur_width*1.25], 'EdgeColor','g', 'LineWidth', 3);
                    break;
                end
            elseif(no_of_cols == 2 && i <= no_of_cols) %number of vertical symmetry
axes equal to 2 (commonly found case)
                if(locations(j) >= locations(j+1)-widths(j+1) && locations(j) <=
locations(j+1)+widths(j+1))
                    rectangle('Position',[((locations(i)-widths(i)/2) (locations2(j)-
widths2(j)/2) cur_width*1.25 cur_width*1.25], 'EdgeColor','g', 'LineWidth', 3);
                    bool = 0;
                    break;
                else
                    rectangle('Position',[((locations(i)-widths(i)/2) (locations2(j)-
widths2(j)/2) cur_width*1.25 cur_width*1.25], 'EdgeColor','g', 'LineWidth', 3);
                    end
            elseif( no_of_cols ~= 2 || no_of_cols2 ~= 2 )
                rectangle('Position',[((locations(i)-widths(i)/2) (locations2(j)-
widths2(j)/2) cur_width*1.25 cur_width*1.25], 'EdgeColor','g', 'LineWidth', 3);
                end
            end
        end
    end
hold off;
end
```

## Results

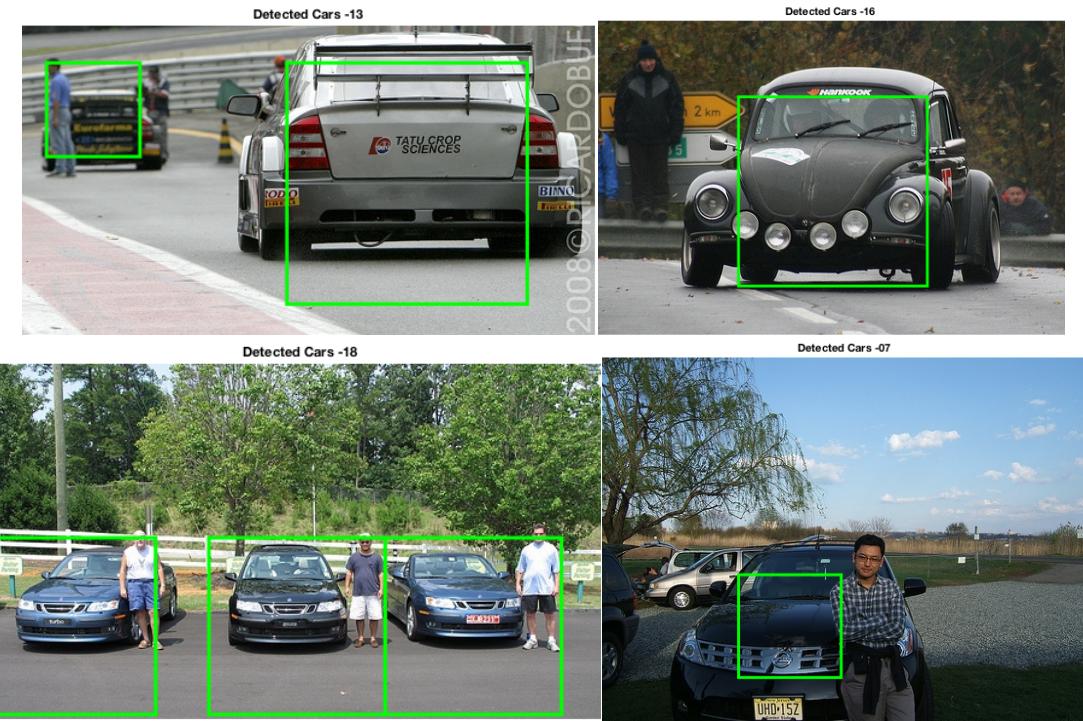
### Successful Results



In these results, mainly the cars were detected in their correct positions. A small error in 2 of these images was that the sizes of the bounding boxes (widths of the peaks) were not determined correctly. This is due to too much/less smoothing of the summation histograms. For some images, the opening structuring element size was too large, and for some, it was too small.



In the above images, the positions of cars were again determined correctly, however the peak widths were found small in most of the images. This is due to the fact that the size of the structuring element used when the histograms were smoothed out, was too small. In the fourth image, the perspective affected the histogram peak width and this caused the algorithm to detect the part of the car. In the last image, the car at the back was not detected, since its peak was lost when smoothing out the histogram.

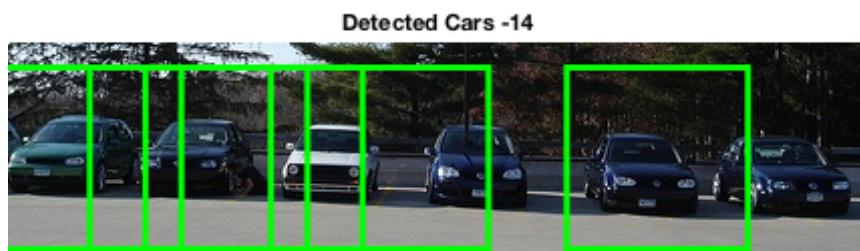


The most successful image here was the first image. The car in the back was detected, since its peak remained even after smoothing the histogram. Again the sizes of the bounding boxes were a bit off, but the cars were detected without any problem even considering the different perspectives.

### Unsuccessful Results



In the images above, the algorithm detected the floor as the maximum symmetry in the histogram peaks, because of the surrounding elements in the first image and because of the different perspective in the second image.



In the above image, most of the cars were detected, however two remained undetected due to close positioning of the cars and the sudden brightness change in the right side.



In the left image, the positions of the cars were detected higher than they really are. This is due to a sudden brightness change and the ceiling causing the horizontal symmetry axis to be translated higher. In the right image, because of the fact that the first, third and the fourth cars having a totally different perspective according to the camera, they were not detected since their peak heights were under a certain threshold.

## Resources

- [1] N. D. Matthews, P. E. An, D. Charnley, and C. J. Harris: Vehicle detection and recognition in greyscale imagery, Intelligent Autonomous Vehicles, pp. 1-6, 1996.
- [2] Y. Li and L. G. Shapiro, "Object Recognition for Content-Based Image Retrieval," in Lecture Notes in Computer Science, Springer-Verlag, 2004.