Homework 2

Student Name: **SHAURYA PANTHRI**

AuE 8930: Machine Perception and Intelligence
Instructor: Dr. Bing Li, Clemson University, Department of Automotive Engineering

* Refer to Syllabus for homework grading, submission and plagiarism policies;
* Submission files includes (Due March. 6, 2020 11:59 pm):

- This document file (with answers), and with your program results/visualization.
- A .zip file of source code (and data if any) with names indicating question number.

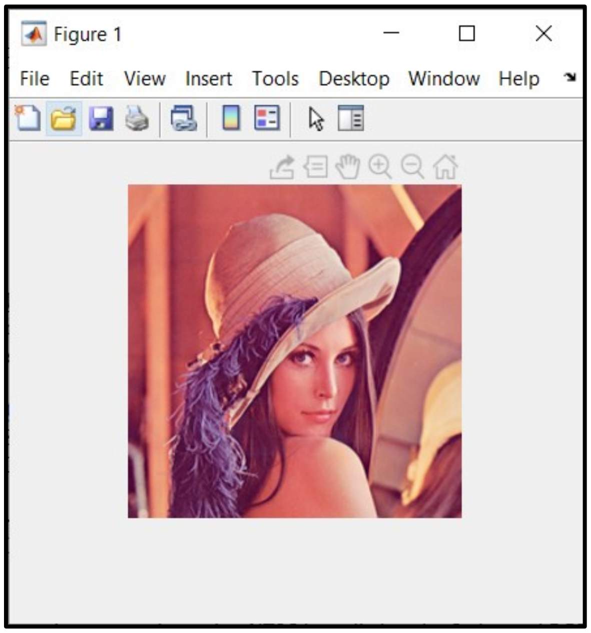
Note: For questions 1) and 2), you are required to write your own code rather than using any direct build-in implementation from 3rd party (like Matlab, Python, or others) libraries. You may use 3rd party built-in functions to check your results if you would like.

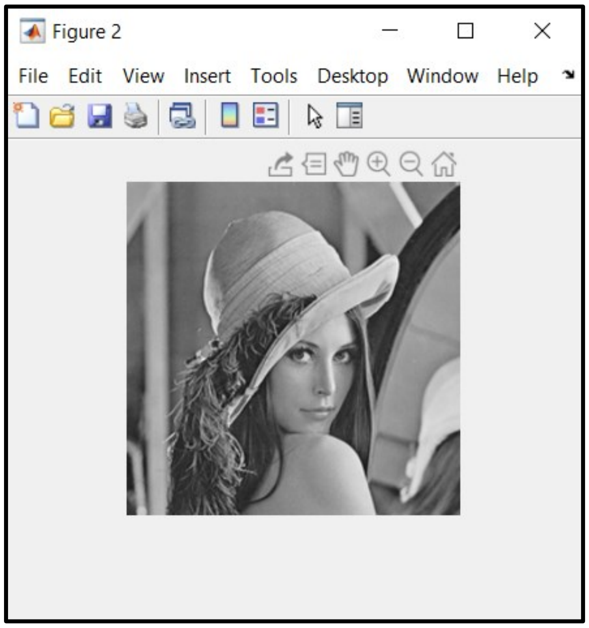
Question 1)

[Sampling/2D-Convolution – 15 pts] Download the image "Lenna.jpg" from the hyperlink.
(Lenna or Lena image is a standard test image widely used for image processing since 1973.)

1-1) Convert the image from RGB to gray, using a standard RGB-intensity conversion approach like NTSC, and store the converted image "LennaGray.jpg" as an 8-bit gray image. (2 pts)

ANSWER 1-1) The "Lenna.jpg" image was converted into "LennaGray.jpg" using standard RGB intensity conversion using NTSC by splitting the 3 channel RGB image into single gray scale by converting using the formula: 0.2989 * r + 0.5870 * g + 0.1140 * b. where, r denotes red, g denotes green and b indicates blue. A function was created in MATLAB by me called grayscale that accepts the RGB image and return the grayscale image.





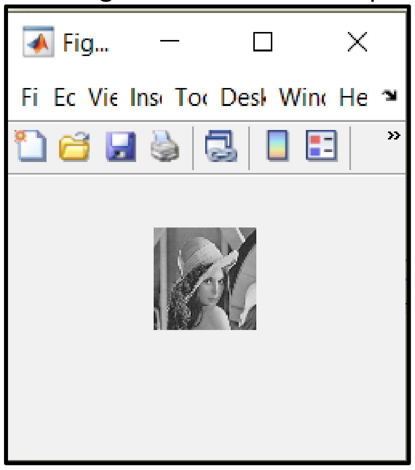
Function:

```
function [gray] = gray_scale(gray)
r= gray(:,:,1);
g= gray(:,:,2);
```

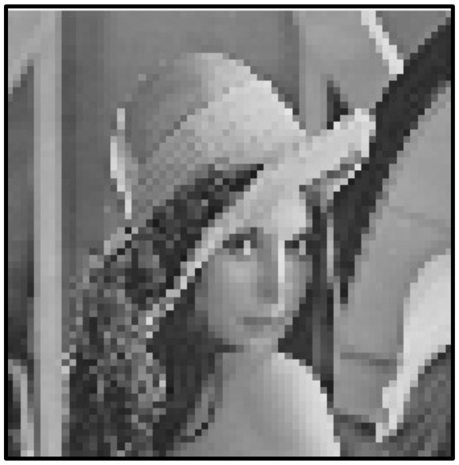
```
b= gray(:,:,3);
gray = 0.2989 * r + 0.5870 * g + 0.1140 * b
% gray = 0 * r + 0 * g + 0 * b
```

1-2) Down-sampling image "LennaGray.jpg" from size 256x256 to 64x64. (3 pts)
Perform the down-sampling and visualize your result.

ANSWER 1-2) In order to down sample the image a step function in matlab was used to iterate through the pixels in x and y direction with a step of 2 to first convert the 256x256 image into 128x 128 and then again the step was repeated to resample the image to 64x64. This can also be done by using a step of 4. The image was saved as resampled in MATLAB.



downsampled →



CODE:

```
resampled=gray1(1:2:length(gray1),1:2:length(gray1))
resampled=resampled(1:2:length(resampled),1:2:length(resampled))
figure()
imshow(resampled)                    % Down sampled image
```

1-3) Implement the convolution (using basic arithmetic operations only, rather than build-in conv()) of Sobel kernel on the "LennaGray.jpg" for edge detection, visualize and comment your detection result. (10 pts)

ANSWER 1-3) The convolution was performed by using a mask in x and y direction throughout the image pixels and the resultant of the two convolutions was calculated for edge detection of the image. The final image was type casted into uint8 format and saved as variable "G" in MATLAB while the convoluted images for GX and GY in x and y direction respectively were saved as "Gx" and "Gy".

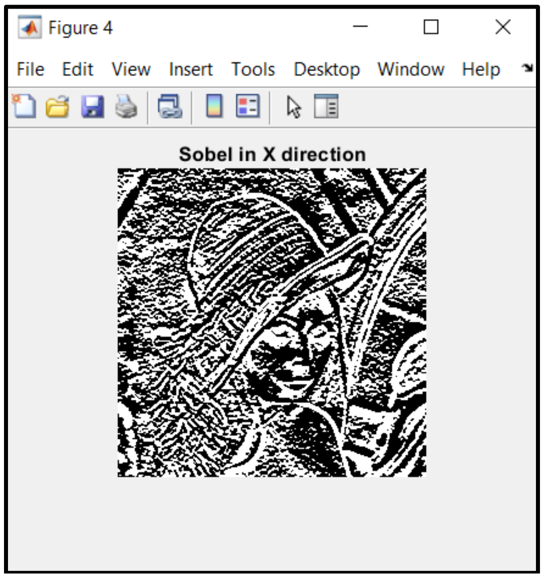
For vertical direction

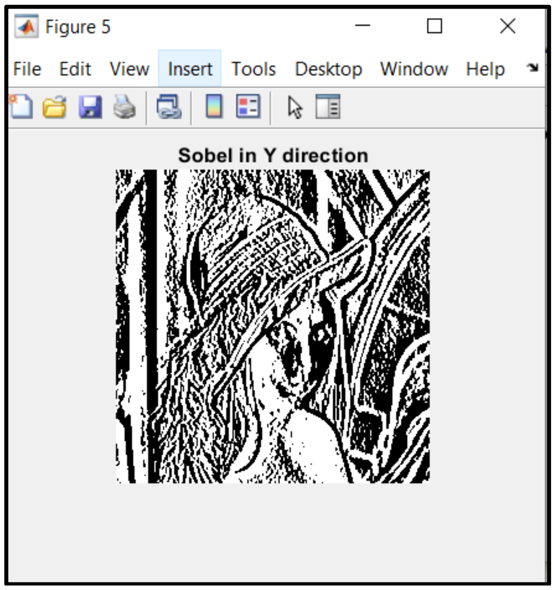
$$S_1 = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} * I$$

For horizontal direction

$$S_2 = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} * I$$

$$|G| = \sqrt{Gx^2 + Gy^2}$$





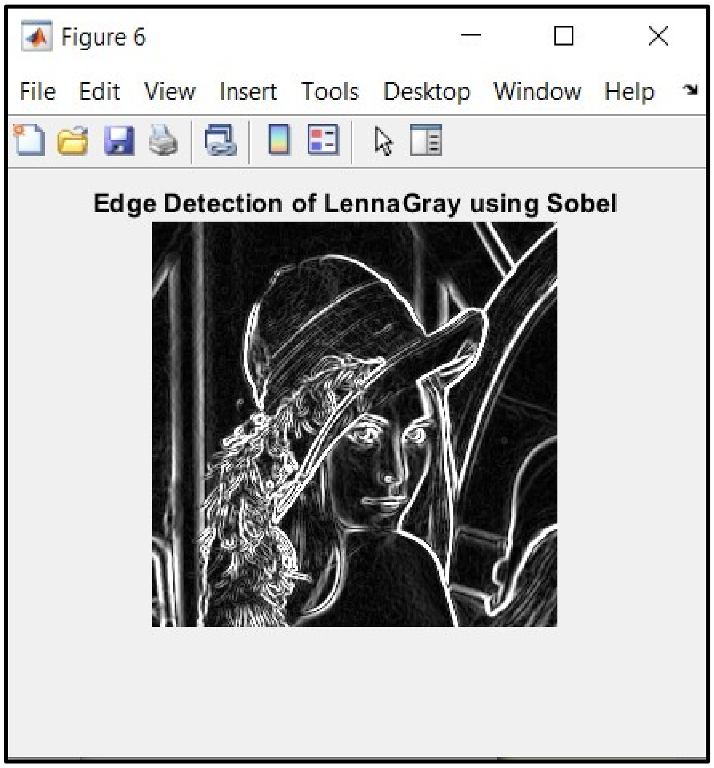


Figure 6
File Edit View Insert Tools Desktop Window Help
Edge Detection of LennaGray using Sobel

Edge detected in the image

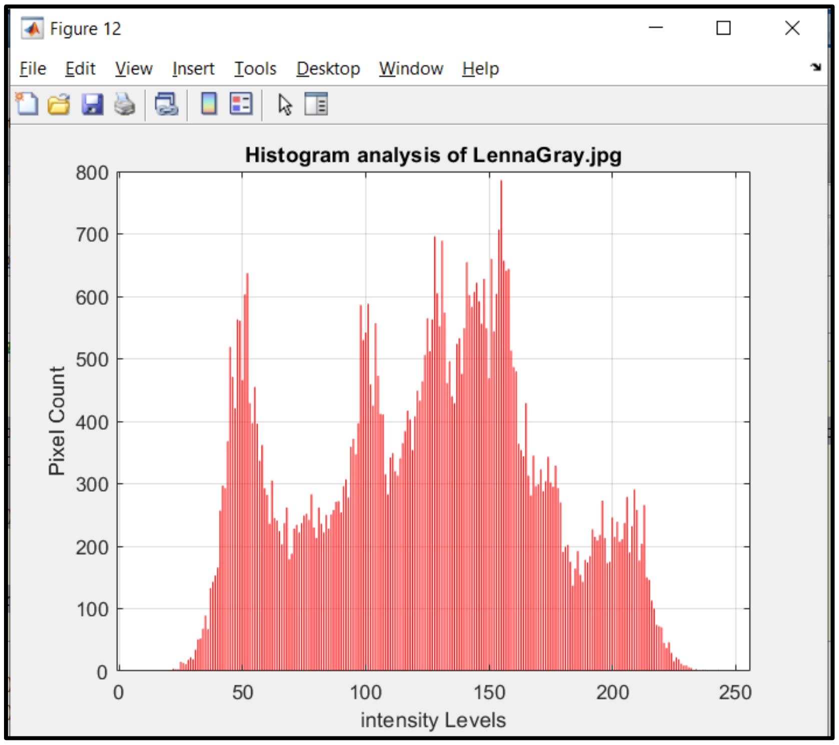
CODE:
```
%% Question 01.03 Edge detection using SOBEL
maskx=[-1 -2 -1;
        0  0  0;
        1  2  1];              % go to the image and multiply with
mask and keep shifting and save the result.

grayc=double(gray1)
[row,column]=size(gray1)  % store the rows and columns of the
GrayLenna image.
conv=zeros(row-3,column-3);
for i=1:(row-3);
    for j=1:(column-3);                        % last index is row and
column minus 3 because the length of the map is 3x3
%         cla
%         imshow(gray1)
%         rectangle('Position',[i j 3 3])       % Visualize the
movement of the kernel
%         drawnow
        gray_extract= grayc(i:(i+2),j:(j+2));
        r=maskx.*gray_extract;
        conv(i,j)=sum(sum(r));
    end
end
Gx= conv;
figure()
imshow(Gx)
title("Sobel in X direction")


% MASK in Y direction
masky=[-1 0 1;
        -2 0 2;
        -1 0 1];
conv=zeros(row-3,column-3);
for i=1:(row-3);
    for j=1:(column-3);                        % last index is row and
column minus 3 because the length of the map is 3x3
%         cla
%         imshow(gray1)
%         rectangle('Position',[i j 3 3])       % Visualize the
movement of the kernel
%         drawnow
        gray_extract1= grayc(i:(i+2),j:(j+2));
        r1=masky.*gray_extract1;
        conv(i,j)=sum(sum(r1));
    end
end
Gy= conv;
figure()
imshow((Gy))
title("Sobel in Y direction")

% Resultant of Gx and Gx for normalizing
G=uint8(sqrt((Gx.^2) +(Gy.^2)));
figure()
imshow(G)
title('Edge Detection of LennaGray using Sobel')
```

Question 2)

[Histogram Equalization – 15 pts.] Take the converted from above gray image “LennaGray.jpg”.

2-1) Perform histogram analysis and visualize histogram distribution (2 pts);

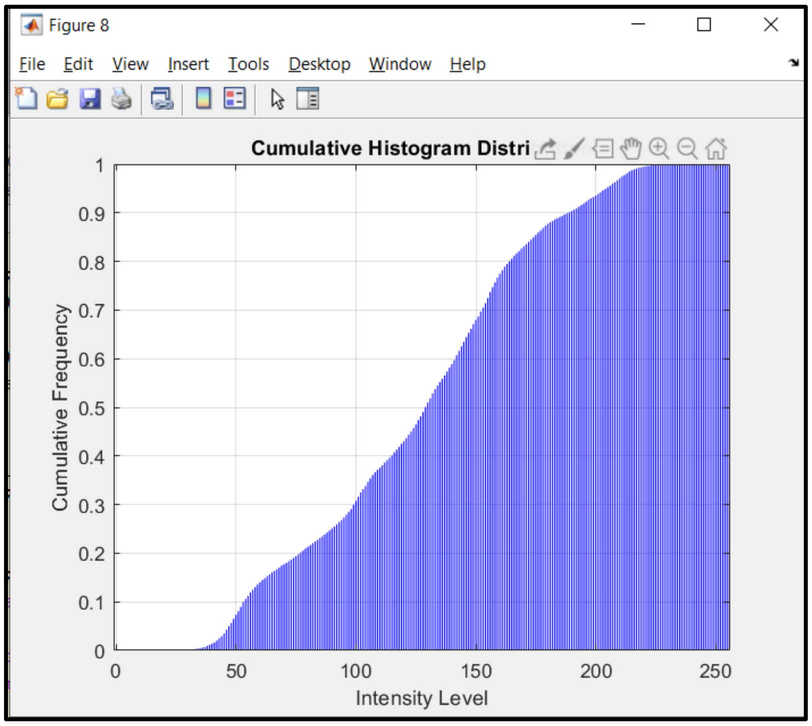
ANSWER 2-1) The histogram of the image provided the graphical representation of the tonal distribution of the image. Histogramz function was created in MATALB to perform the histogram analysis and visualize the histogram distribution.



CODE:

```
% [counts,grayLevels]=Histogramz(gray1)        % Computes the
histogram analysis of the image
function [counts, grayLevel] = Histogramz(gray1)
[rows, columns] = size(gray1);
counts = zeros(1, length(gray1));
for col = 1 : columns
  for row = 1 : rows
    grayLevel = gray1(row, col);                % Get the gray level.
    counts(grayLevel+1) = counts(grayLevel+1) + 1;    % Add 1
because graylevel zero goes into index 1 and so on.
  end
end
% Plot the histogram.
intensity_levels = 0 : 255;
figure();
bar(intensity_levels, counts, 'BarWidth', 0.5, 'FaceColor',
'r');
xlabel('intensity Levels');
ylabel('Pixel Count');
title('Histogram analysis of LennaGray.jpg');
grid on;
end
```

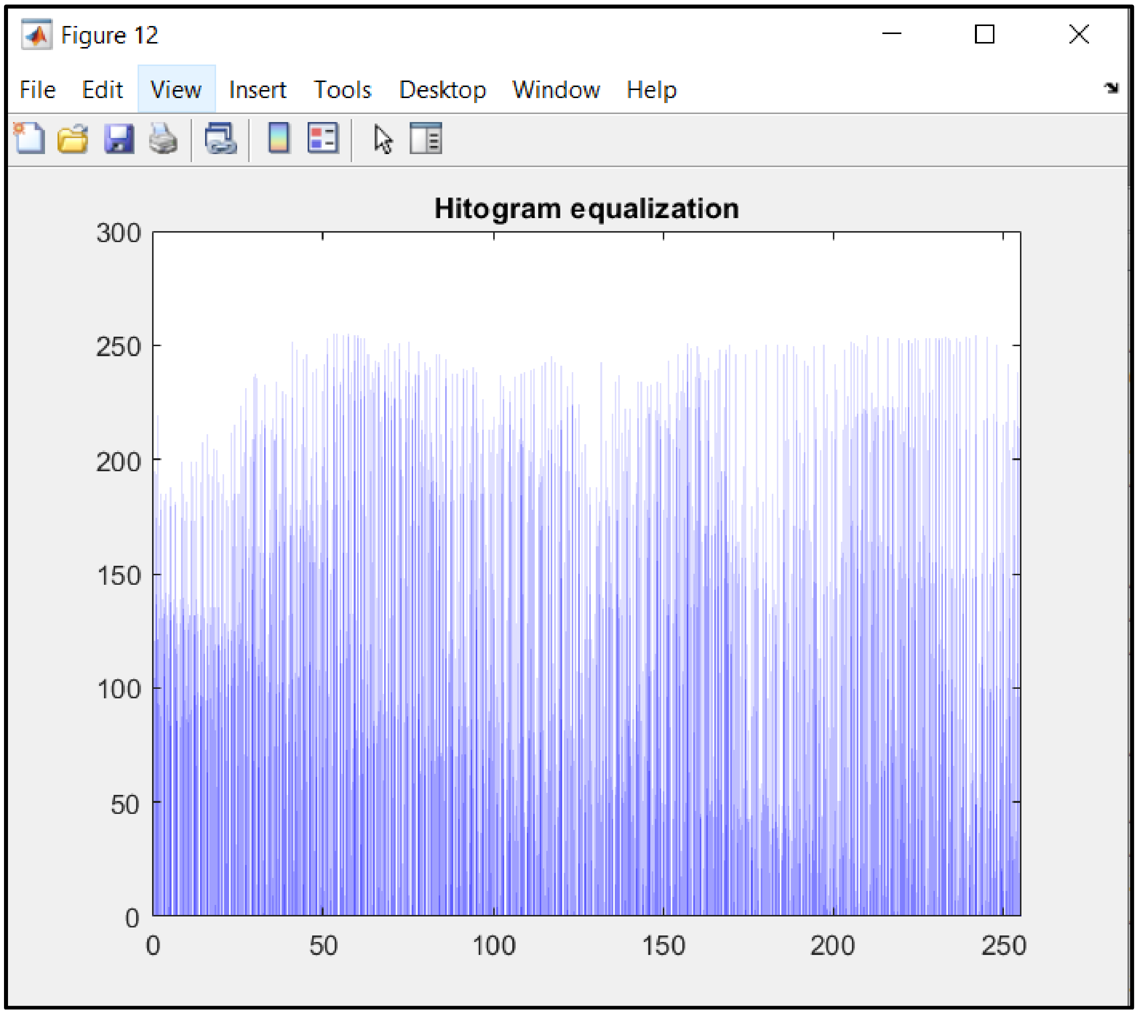
2-2) Calculate and visualize accumulative histogram distribution (3 pts);
ANSWER 2-2) The pdf function was created in MATLAB and its cdf that is, the cumulative distribution function was visualized as the cumulative histogram distribution of the image using MATLAB whose summation was overall 1 over the intensity level 0-255 of the image.

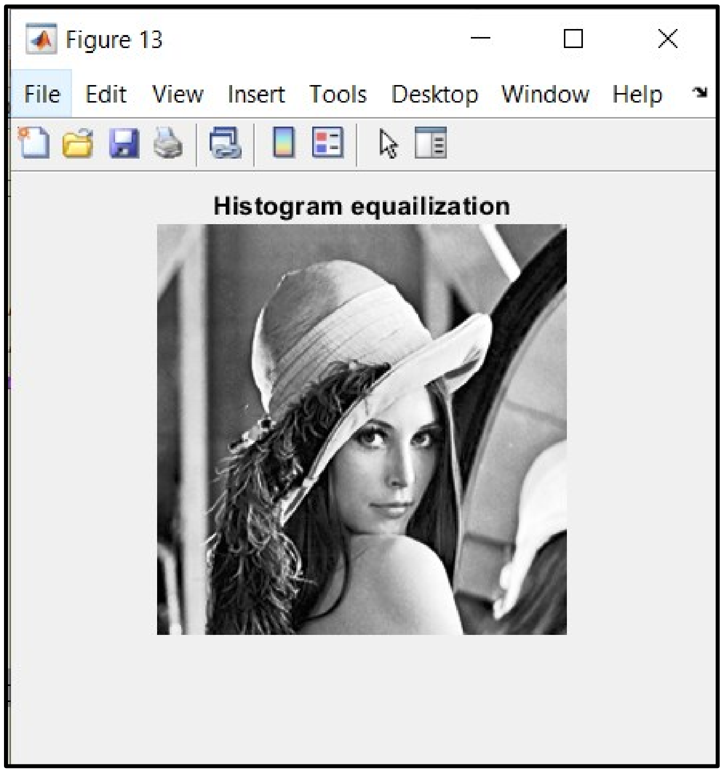


CODE:
```
cf=zeros(1,length(counts));
pdf=zeros(1,length(counts));
size=size(gray1);
pixel_total=size(1)*size(2);
pdf(1)=counts(1)/pixel_total;
cdf(1)=pdf(1);
for i=2:length(counts);
    pdf(i)=counts(i)/pixel_total;
    cf(i)=pdf(i-1)+cf(i-1);
end
figure();
intensity_levels = 0 : 255;
bar(intensity_levels,cf,'Barwidth',0.5,'FaceColor','b');
xlabel('Intensity Level');
ylabel('Cumulative Frequency');
title('Cumulative Histogram Distribution');
grid on;
```

2-3) Implement a function to perform histogram equalization for this image, visualize your histogram-equalized image and its histogram distribution. Comments the difference between the two images before/after histogram equalization. (10 pts);

Answer 2-3) The histogram equalization was performed by sampling the cumulative function multiplied with 255 as the highest pixel intensity, then type casted into integer format that is unit8. The histogram-equalized image and its histogram distribution were visualized.
The histogram equalization increased the global contrast of the image. Through this the intensities were better distributed on the histogram. It allowed the areas for lower contrast to gain somewhat higher contrast. This is done effectively by spreading out the most recent frequency intensity using the cumulative distribution function.





CODE:

```
%% Question 2.3
func=uint8(zeros(1,length(counts)));
l1=length(counts)
for i=1:l1;
    func(i)=uint8(255* cf(i));
             % sampling can also use round()
end
% figure();
% bar(intensity_levels,func,'Barwidth',0.5,'FaceColor','b');
% xlabel('Intensity Level');
% ylabel('Cumulative Frequency');
% title('CF');
% grid on;


for i=1:row;
for j=1:column;
 gray2(i,j)=func(gray1(i,j)+1);
end
end
figure()
bar(intensity_levels,gray2,'Barwidth',0.5,'FaceColor','b');
title('Hitogram equalization')
```

```
figure,imshow(gray2);
title('Histogram equailization')


% figure()
% a=histeq(gray1)              %Using built in function
% bar(intensity_levels,a,'Barwidth',0.5,'FaceColor','b');
% title('HIstogram equlization using matlab function')
```

Question 3)
[Line Detection – 30 pts] Download the image "ParkingLot.jpg" from the hyperlink.
Note: For this question, you are free to use any 3rd party libraries.

3-1) Apply and visualize histogram analysis, then find a proper threshold to convert the image to a binary image. (2 pts)

Answer 3-1) The histogram equalization was performed in MATLAB using the histeq() built in function in MATALB and to convert the image into Binary im2bw function was used that requires two parameters that is the image and the threshold. After using a loop the threshold was selected as 0.93.

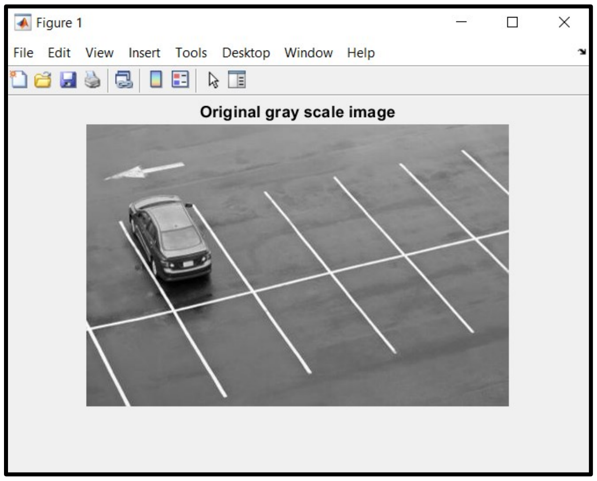


Figure 1
File Edit View Insert Tools Desktop Window Help
Original gray scale image

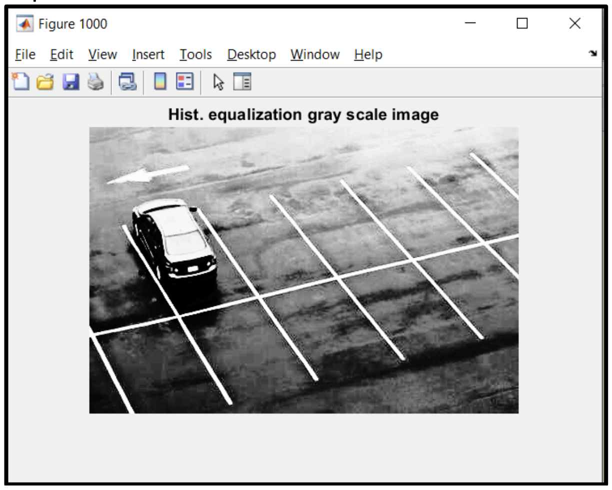
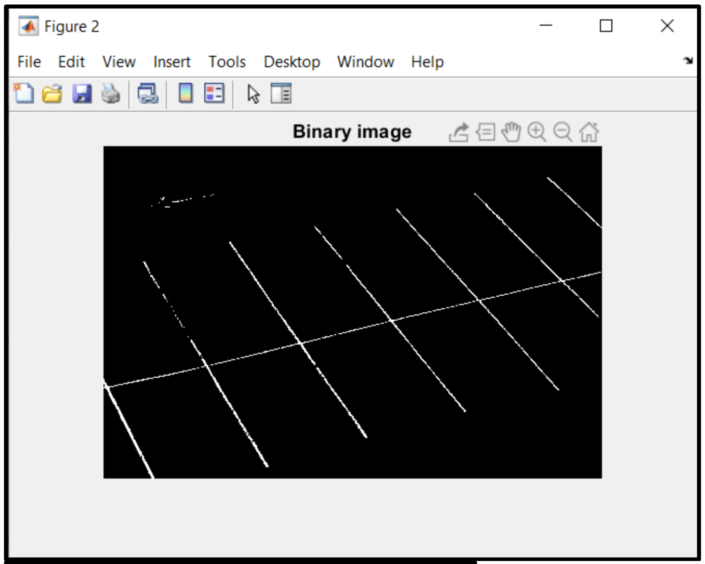
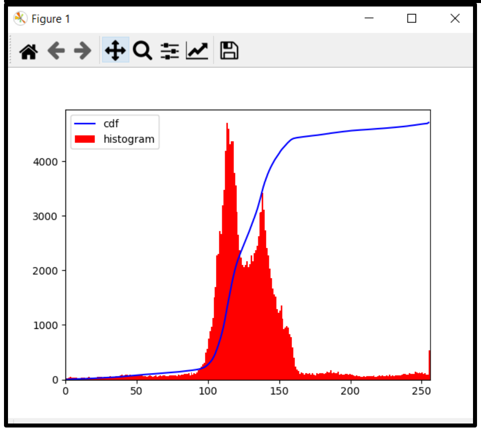
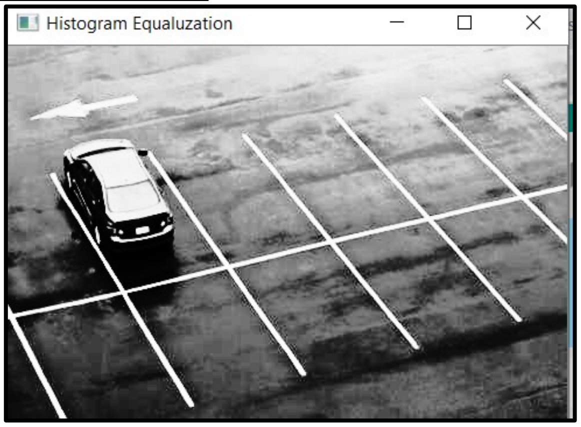


Figure 1000
File Edit View Insert Tools Desktop Window Help
Hist. equalization gray scale image







Results in python Question3.py

CODE:

```
%% Question 3.1
% grays=rgb2gray(img1);
% figure();
% imshow(img1)
% ALready a grayscale image

figure(1000)
histeq(img1) ;
%hist eq. gives a well distributed intensity image
title('Hist. equalization gray scale image')
% figure()
% for i=0.1:0.01:1;
% BW = im2bw(img1,i)
% print(i)
% imshow(BW)
% end
        % Select the threshold of the image using the loop
        %COnvert into Binary
figure()
BW=im2bw(img1,0.93);
BW=imfill(BW,'holes')
imshow(BW)
title('Binary image')
```

3-2) Apply Hough transformation or other line detection approach to detect multiple lines in the image (You select a threshold for the voting matrix). Visualize the lines in the image space and in the transformed space (like Polar space) respectively. (5 pts)

ANSWER 3-2) The Hough transformation was used to extract the lines from the image using the MATALB function hough() that accepts the image, rho resolution and theta. In order to calculate the hough peaks a threshold value of 0.3 of max(h) was used and the lines were extracted from the image using the function houghlines. The lines were not distributed chronologically in the image therefore they were sorted using the sort() function and plotted.
The intersection point between the horizontal and the vertical lines were not extracted using the hough transform. The visualization of the lines in the image space and the transformed space are as follows:

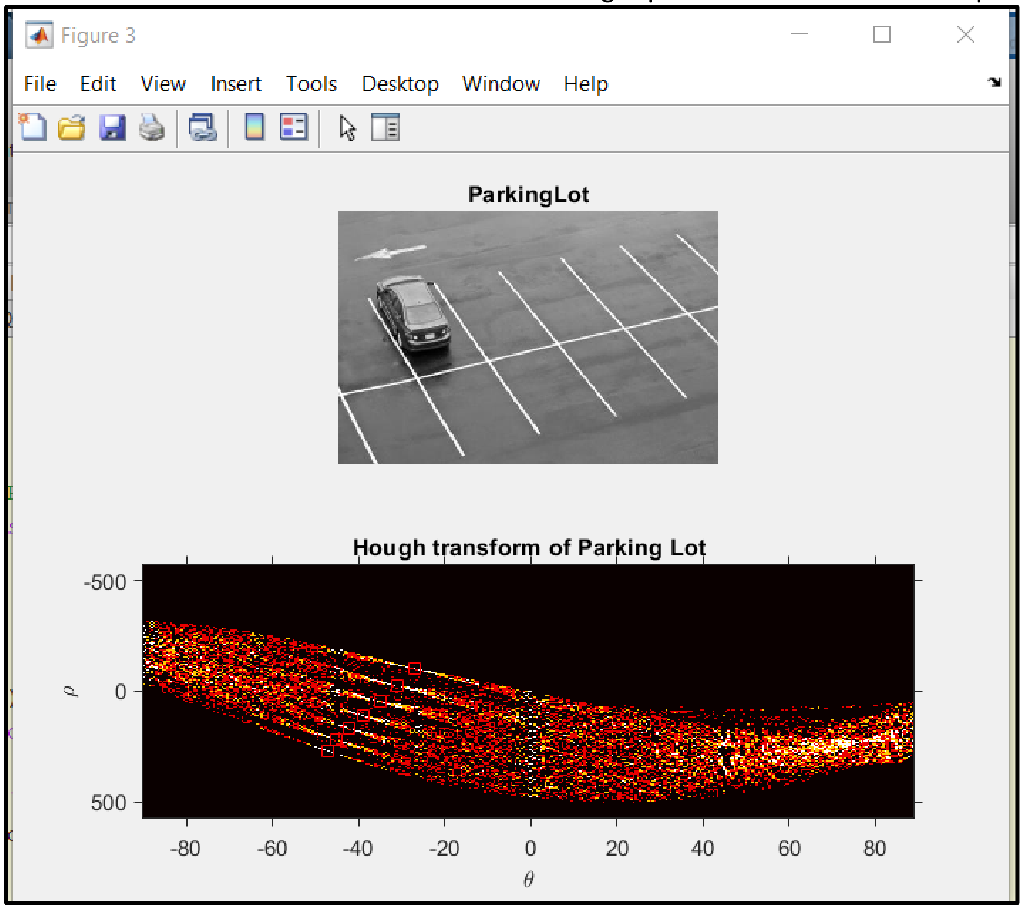
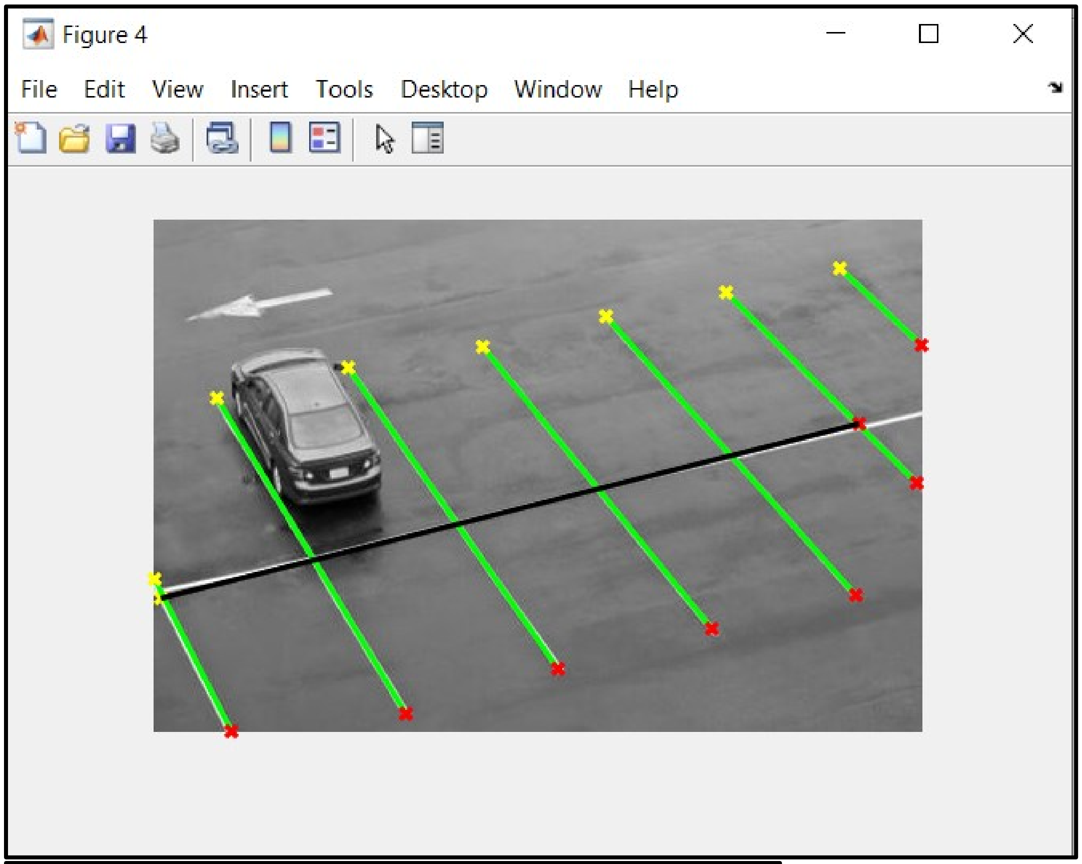
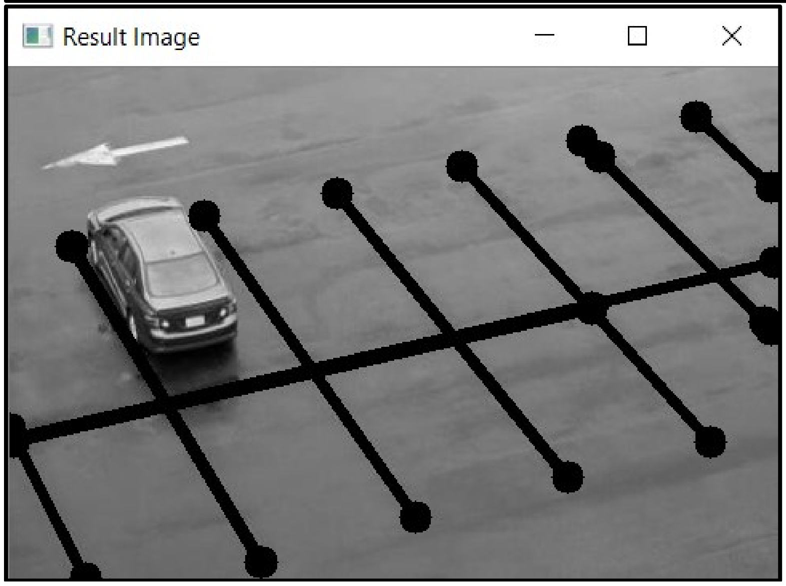


Figure 3
File Edit View Insert Tools Desktop Window Help
ParkingLot
Hough transform of Parking Lot
-500
0
500
ρ
-80 -60 -40 -20 0 20 40 60 80
θ





Result in python

CODE:

```
% Question 3.2
% BW = edge(BW,'canny');
% figure();
% imshow(BW);
```

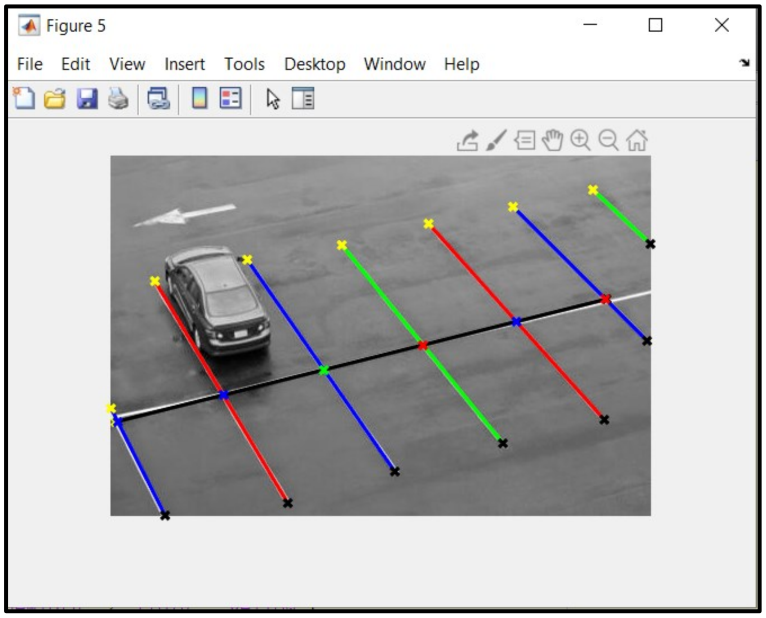
```
figure()
% [H,T,R] = hough(BW,'RhoResolution',0.5,'Theta',-90:0.5:89);
[H,T,R] = hough(BW,'RhoResolution',0.1,'Theta',-90:0.5:89);
subplot(2,1,1);
imshow(img1);
title('ParkingLot');
subplot(2,1,2);
imshow(imadjust(rescale(H)),'XData',T,'YData',R,...
      'InitialMagnification','fit');
title('Hough transform of Parking Lot');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;
colormap(gca,hot);

[H,theta,rho] = hough(BW);
P = houghpeaks(H,10,'threshold',ceil(0.3*max(H(:))));
x = theta(P(:,2));
y = rho(P(:,1));
plot(x,y,'s','color','red');


lines = houghlines(BW,theta,rho,P,'FillGap',400,'MinLength',50);


figure, imshow(img1), hold on
max_len = 0;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

   % Plot beginnings and ends of lines
   plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

   % Determine the endpoints of the longest line segment
   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
      max_len = len;
      xy_long = xy;
   end

end
% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','black');
```

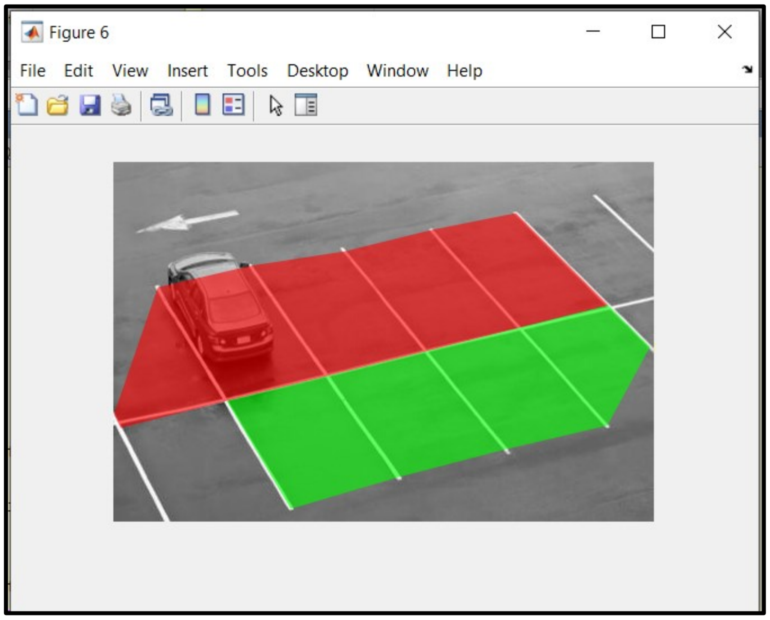
3-3) Comment on: will the two lines as two sides of a particular park space be parallel or not, explain why? (3 pts)

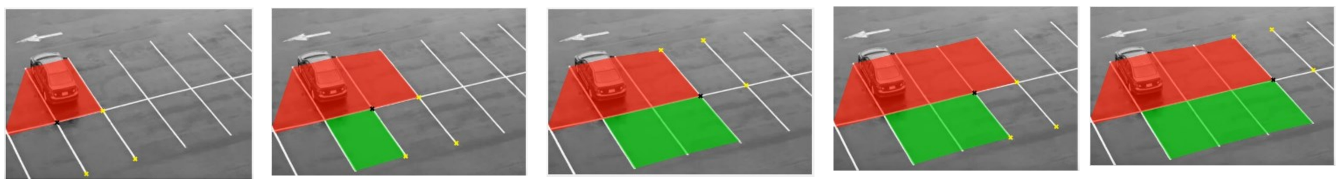
ANSWER 3-3) The two dimensional prospective projections of the lines on the two sides of a particular park space do not appear to be parallel and appear to converge in infinity. The objects that are near to the camera appear to be diverging while the ones away from the camera appear to converge and appear closer to each other than they really are.

3-4) Design and implement the approaches to find all parking space polygons with the four vertex points for each parking space. Describe your approaches and visualize all detected polygons with different colors overlaid on the original image. The TA will check your code. (20 pts)

ANSWER 3-4) The hough lines were sorted as the x variables in ascending and the y in descending because this was the trend in the image. Then these lines were visualized and found that they worked. The longest line that is the horizontal line was visualized with different color to separate it from the rest x_long, y_long and hence the intersection between all the vertical lines of the parking lot were indexed in a for loop that used a polyxpoly() function that returns the intersection points between them. These points were plotted with different color by indexing an array of different color for each iteration of the loop. In the end another for loop was used to insert the polygon on different parking space. The parking space above the horizontal line were plotted with different color as compared to the ones below the horizontal line. A video of the animation for each polygon is also provided.







Snips from the video for few iterations.

Question 4)

[Survey – 40 pts] Write a 2~3 pages survey report on a specific 2D-data measurement/detection problem related to automotive engineering (e.g.: lane detection, traffic sign detection, drivable area detection, a B-scan inspection for a manufacturing component, material characterization using microscopy image, et al. It is not limited to camera data. A negative example: 'Obstacle detection in 2D image' is not considered to be a specific problem because obstacle is not a specific target).

The grading of this question is based on the contents which the survey covers:
- The importance of this measurement (5);
- The challenges of measuring this target (5);
- Existing solutions of measuring this target (15);
- Existing problems of measuring this target (5);

There will be other grading factors (such as novelty, organization, et al) (10);
* You are encouraged to include any drawing/table in the report;
* Attention: use "…" [1] to cite any sentence you literally copied and use … [1] to cite a content you referred to, with reference list in the end;

ANSWER:

1. The importance of this measurement.
   Lane Detection is required for various ADAS features and purely autonomous vehicles. The lane detection is a virtual assistant to improve driving safety and reduce traffic accidents. The lane boundaries are detected to avoid collisions and issue warnings if a vehicle passes a lane boundary. It is important to precisely detect the lanes by applying computer vision because at times the lane boundaries are not clearly visible.
   - Lane Keep Assist: The lane detection helps the vehicle to avoid drifting out of the lane unintentionally by continuously providing inputs to the steering of the vehicle. Also, it has features to warn the driver in condition of lane departure through audio, visual or vibration signals (seat pulses in the direction of the lane and steering wheels).
   - Eliminate illegal parking of vehicles: In certain countries like Korea, a surveillance vehicle is used to collect pictures of illegally parked vehicles for penalizing these incidents and regulation of law and order. This vehicle uses a camera that detects if the vehicle is parked outside the dedicated lane for parking.
2. The challenges in measuring this target.
   Despite the perceived simplicity of finding the white markings on a dark road, it can be very difficult to determine the lane markings on different types of roads due to the following reasons.
   a. The biggest challenge with detecting lanes is that, often the lanes are not clearly visible. This is due to poor road conditions, insufficient quantity of paints used for making the lane boundaries.
   b. Often the environment factors like hail, snow, road potholes filled with water make it difficult to detect the lane lines in such regions

c. Shadow and other opaque objects: Sometimes there are objects on the road, occlusion by other vehicles ahead of the car or littered objects on the road that cover the lane markings or the shadows from trees or the vehicles that makes it difficult to detect the lanes. [1]
d. Illumination conditions: Inadequate streetlight intensity, nighttime condition, fog makes it challenging to detect lane markings on the road. The sunlight affects the road color during the day compared to dusk or dawn time. The RGB gray value will be nearly 130-170 more and 80-130 with shadow. Approaching headlight or the back light also affects detection.
e. Different type of lane markings: No governing standard for the geometry of road markings makes it difficult to consider geometry as the only parameter to discriminate the lanes.
f. The road splitting or merging and the interferences from the roadside objects worsen the detection successfully.

All these challenges make it difficult to discriminate between the road lane from the background in a captured image.

3. Existing solutions of measuring this target.
The erroneous findings will generate wrong steering commands that might jeopardize the vehicle safety. Therefore, a robust and reliable algorithm is the minimum requirement.
    a. Distortion removal with camera calibration: Most of the vehicle-mounted cameras are wide angle lens in order to obtain an enlarged field of vision and the image distortion changes appeared remarkable. In the actual road scene edge distortion of lanes, vehicles and background is called radial distortion. And if the camera is not parallel to image plane of sensor, image is tilted that makes the scene appear closer than its 3-D world position. This is more disadvantages tangential distortion. The mathematical model of the two distortions are as follows:

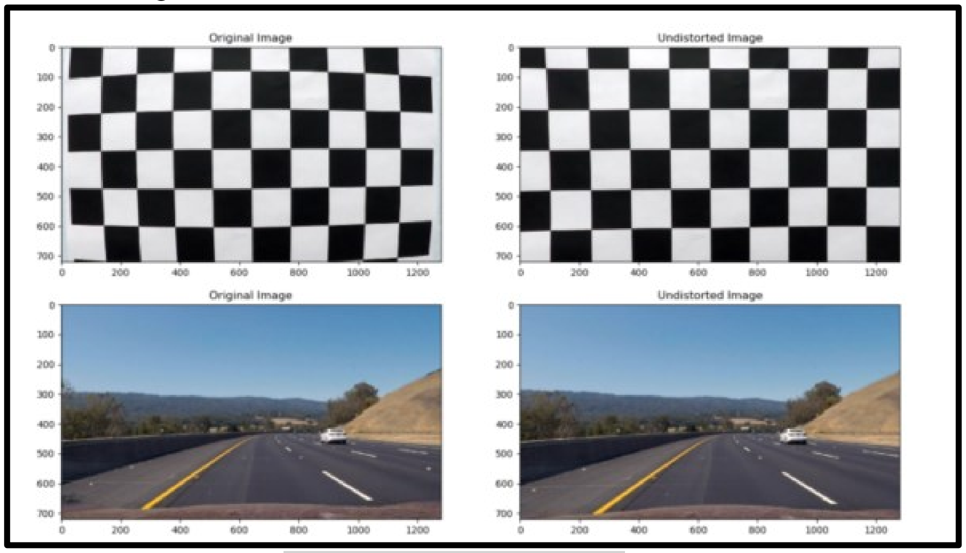
$$\boxed{\begin{cases} u' = u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ v' = v(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases}}$$

1. Mathematical model of radial distortion

$$\boxed{\begin{cases} u' = u + \left[2p_1 v + p_2(r^2 + 2u^2)\right] \\ v' = v + \left[p_1(r^2 + 2v^2) + 2p_2 u\right] \end{cases}}$$

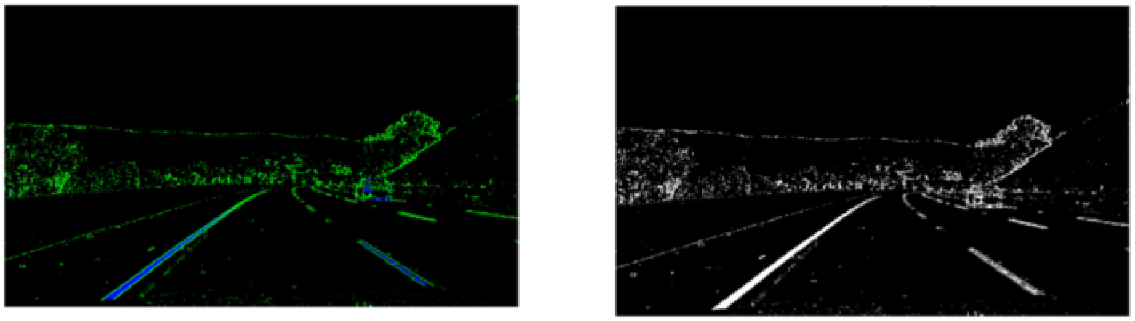
2. Mathematical model of tangential distortion

The parameters k_1, k_2, k_3, p_1, p_2 are acquired by camera calibration and using a checkboard calibration plate composed of black and white grids with regular shapes to easily find corner points and observe the correction conveniently. A transformation matrix is created to map the distorted points to the undistorted points and the distortion is removed using coordinate transformation. [2]



2. The correction of the distorted image

b. Edge detection solution by superimposing the threshold of Sobel operator and HSL color space and less computational cost: The Sobel operator performs edge detection of the image from horizontal and vertical directions and sufficiently filters the interference noise with the improved image processing effect. Only Sobel operator has low detection accuracy and rough edges. Therefore, HSL color space where Hue, saturation and lightness can be used. The position of spectral color represented by angle and different color values correspond to different angles. S refers to the degree of color to describe the similarity between the image and the standard. S [min=0 (grayscale image) where H is undefined to max =1, fully saturated]. L indicates the degree of the image color which changes along x axis. The HSL color space better reflects the visual field perception characteristics and each can be processed independently. By binarizing the image, useless interference information can be removed to speed up the operation of the detection algorithm.



(a) Threshold superimposition (b) Threshold superimposition binarization

3. Gradient threshold and color threshold superposition.

c. ROI Extraction and Inverse perspective Transformation: The effective portion of a frame that is lane accounts for two-third of the area detected is extracted. Using the vanishing

point setting extracted from the Hough transform an adaptive ROI can be established to reduce computational cost and time for lane detection.
Also, since the inverse perspective transformation is based on the inverse coordinate transformation from world coordinate to image coordinates, it transforms the perspective image into aerial view and restore the parallel relationship between the lane lines. As compared to the distortion removal, the IPT maps the position points in detected image to the new position points overlooking the perspective to obtain the aerial view of the road image.

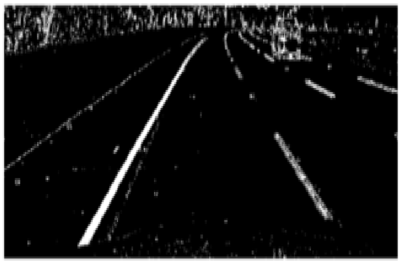


Figure 6. The extracted ROI.

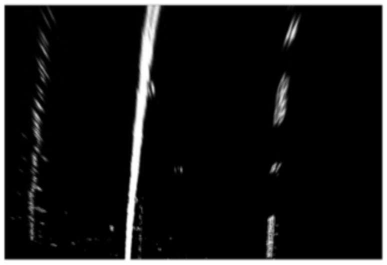


Figure 7. The aerial view of the lane.

Lines are parameterized by (θ_i, λ_i) where N is the number of lines, and $i = 1, 2, 3, \ldots, N$. Then, we can identify the Vanishing point $Vp(x_{vp}, y_{vp})$ of lane boundary as follows:

$$x_{vp} = \frac{AE - CD}{AB - C^2}, y_{vp} = \frac{BD - CE}{AB - C^2}$$

d. Mask Operation, B-spline Curve and RANSAC: The road conditions that are susceptible to environment factors and for dynamic driving while maintaining a high recognition rate with high accuracy. The interfering pixels that can cause false detection in the next lane frame can be done by masking the detected necessary image. The mask operation clearly shows the target pixel point by weighting and averaging the pixels around the target pixel that contributes to sharpening the detection image. [2] The actual lane under complicated road conditions tend to vary therefore a third-order B spline curve can be adjusted accordingly for best fit. The RANSAC that is a randomly sampling algorithm that first obtains random points from a large number of observation data points and then the valid points are selected that meet the hypothesis to estimate the corresponding models after optimization through continuous iteration for accurately fitting based on the effective parameters.

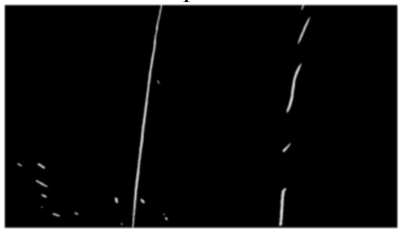


Figure 8. The detection image after mask operation.

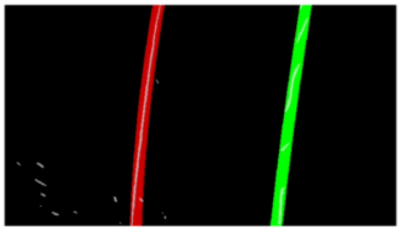
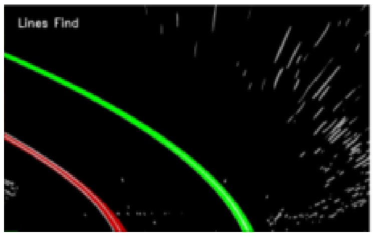


Figure 10. The lane line fitting result.



4. Existing problems of measuring this target.
   a. Image Distortion: Although most of the camera lens enable rapid generation of the images, the images get distorted. Image Distortion: Although most of the camera lens

enable rapid generation of the images, the images get distorted. The distortion changes the shape and size of the lane, vehicle, and background in the road scene. These changes are not conducive to judging the correct driving direction and determining the exact location of the vehicle. Therefore, it is essential to remove image distortion. The changes are not conducive to judging the correct driving direction and determining the exact location of the vehicle. Therefore, it is essential to remove image distortion.

b. Edge Detection: The edges of an image and the structure intensity surrounding them contains an essential information about the scene, the edge detection is important step in scene analysis for lane detection. There are certain problems while detection the edges. Firstly, the edge detectors are not robust to changes against the background noise or the noise introduced by the image capturing system. That is, it should not falsely detect edge caused by noises nor it should miss weak actual edges. Secondly, the edge detectors fail to localize the edges correctly. And third, if the brightness in the scene are non-homogenous their boundaries are not indicated by perfectly closed contour. Although many algorithms are robust but single-edge detection operators (Canny, Prewitt, LOG etc.) have many problems including a too-wide detection range, poor anti-noise interference and prolonged calculation time. [3]
c. ROI extraction: The substantial redundant information such as sky, tress and vehicles remain in the image and also the two lane lines that appear near-wide and far-narrow gradually meet and make it difficult to detect the lane at later stage. This extra information requires unnecessary computation and also increase the time for computation of the subsequent image processing step.
d. Vanishing point: The closer is the object to the camera the bigger it gets and vice versa. The parallel lines of the lanes merge into a point in the distant field of vision that is known as vanishing point. The distance between the adjacent lanes near the vanishing point decreases gradually that is detrimental to the effective lane detection.

References

[1] Y. e. a. ". D. U. B.-S. P. 1. I. C. o. I. I. a. S. (. N. d. Wang.

[2] S. S. X. &. P. (. L. D. A. f. I. V. i. C. R. C. a. D. E. S. 1. 3. d. 1. Cao.

[3] M. (. E. d. p. a. s. 1. I. I. C. o. S. M. a. C. C. C. a. S. d. 1. Bennamoun.

[4] Y. e. a. ". D. U. B.-S. P. 1. I. C. o. I. I. a. S. (. N. d. Wang.