Homework 3

Student Name: Utkarsha Chaudhari

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 Feb. 25, 2021 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.

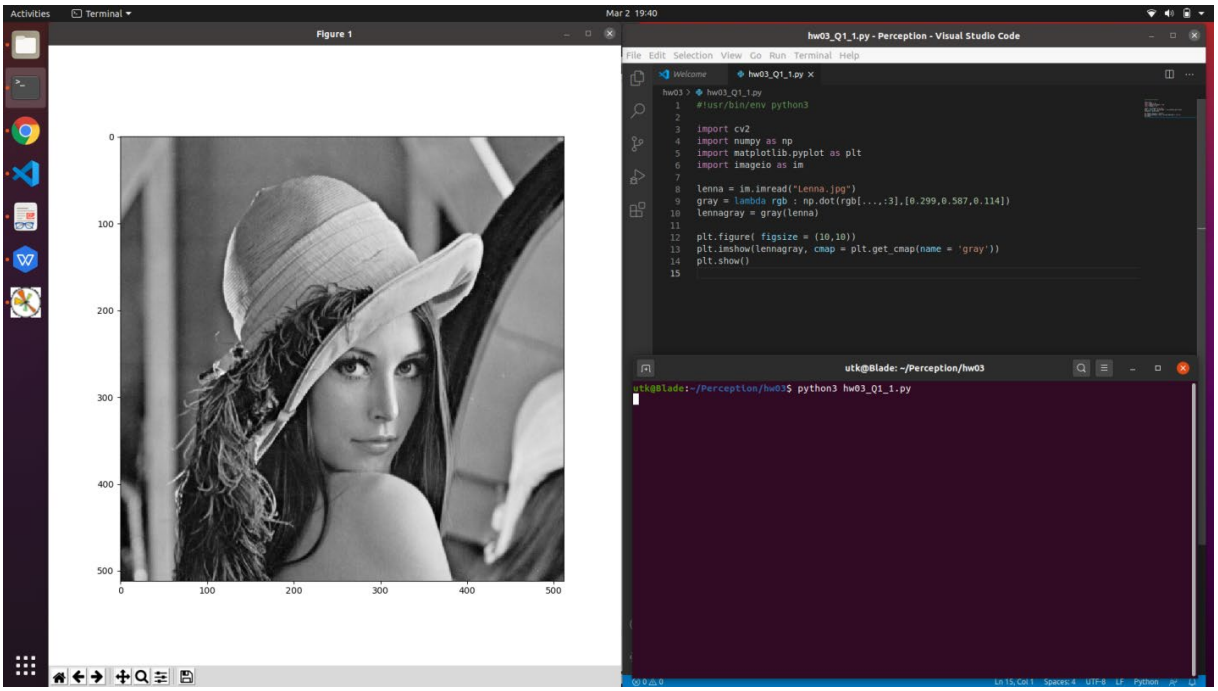
Question 1)

[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:



1-2) Down-sampling image “LennaGray.jpg” from size 256x256 to 64x64. (3 pts)

Perform the down-sampling and visualize your result.

Answer:

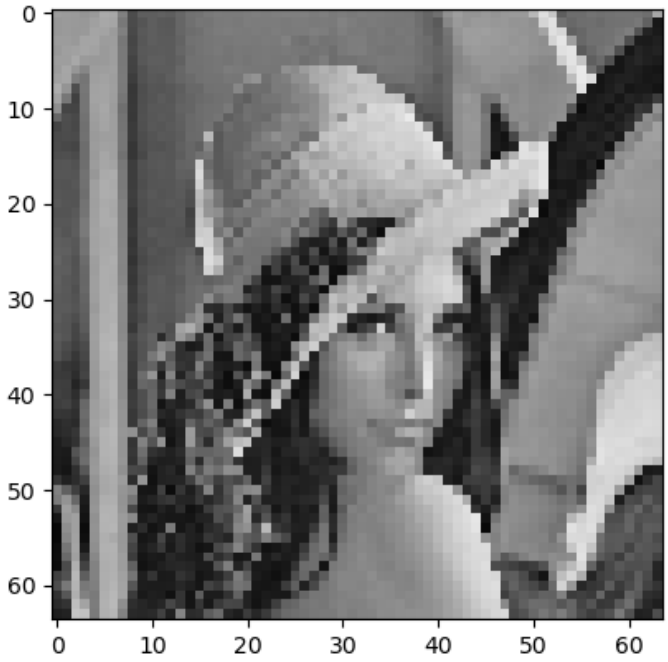


Figure 1: Down sampled image

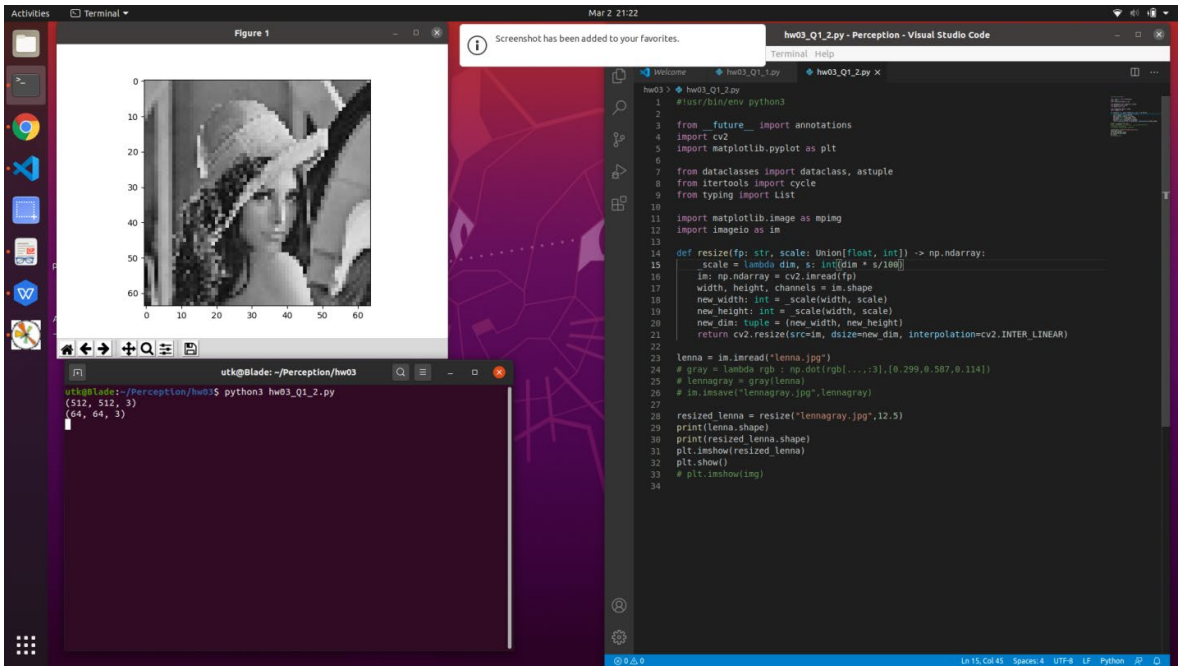
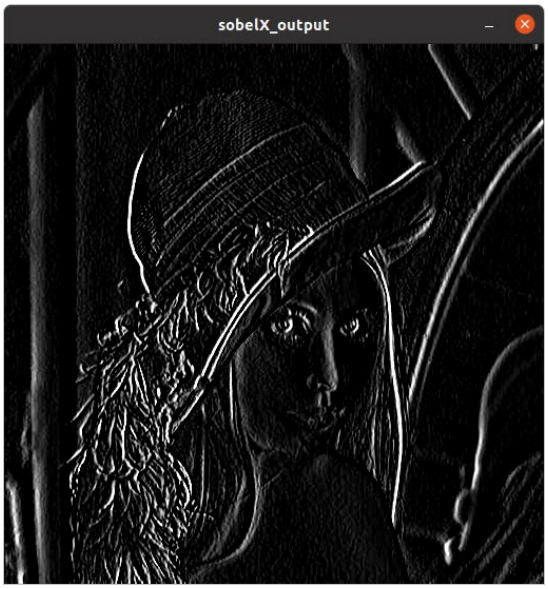


Figure 2: Code and output

1-2) 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:

I have implemented the Sobel kernel and extracted the edges in the horizontal and vertical direction, respectively.



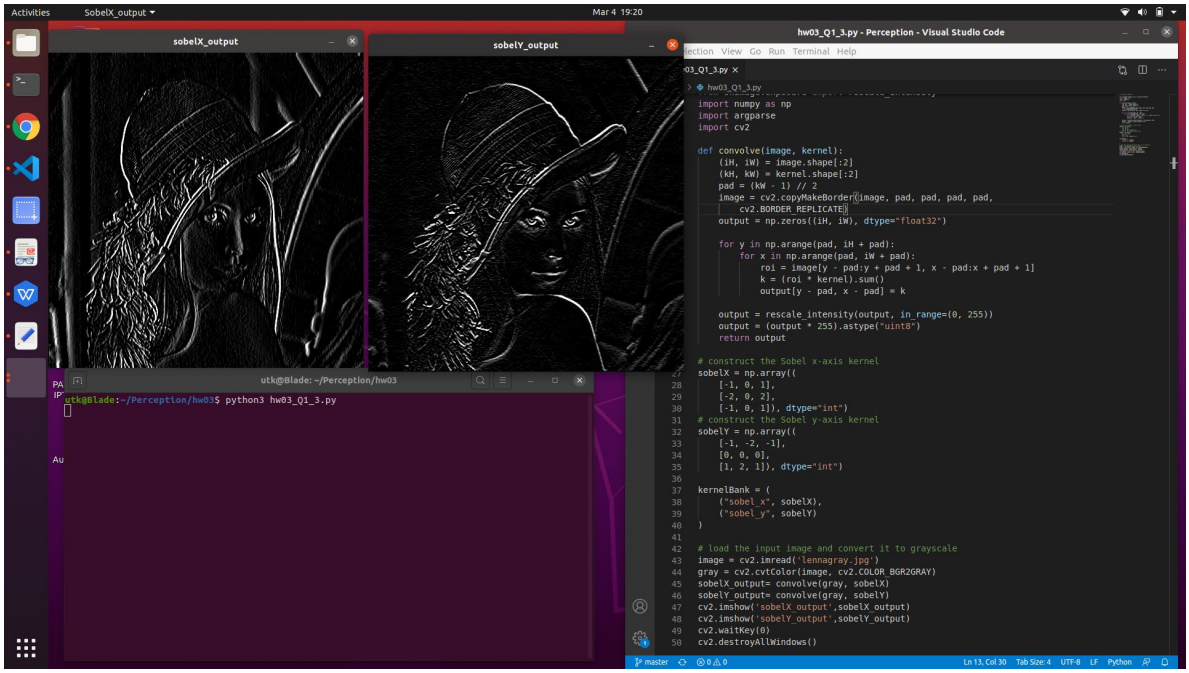


Figure 3: Sobel code and output

Question 2)

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

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

Answer:

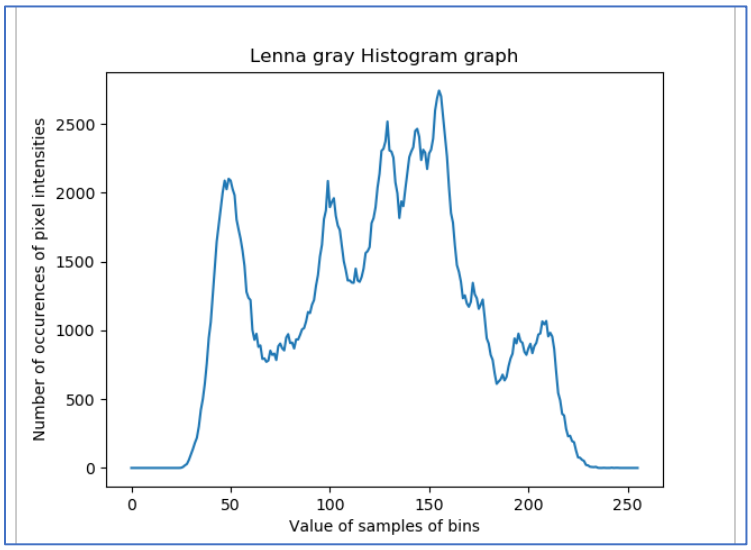


Figure 4: Histogram graph

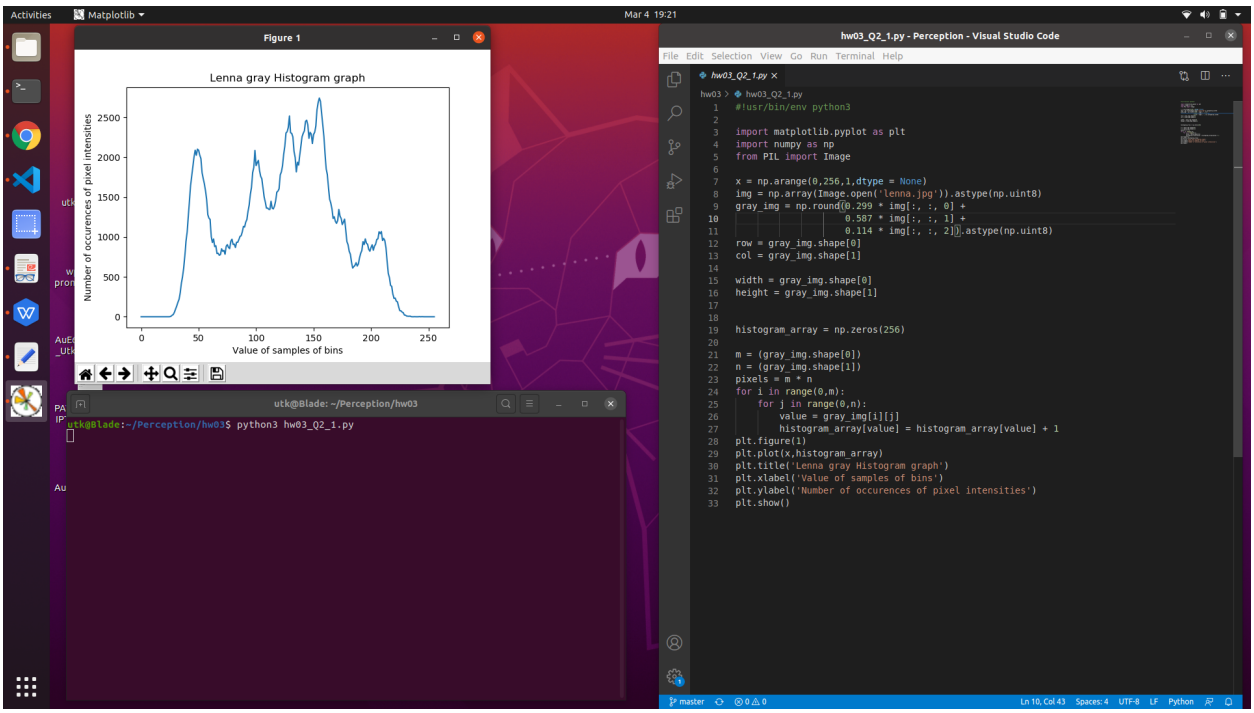


Figure 5: Histogram code and output

2-2) Calculate and visualize accumulative histogram distribution (3 pts)

Answer:

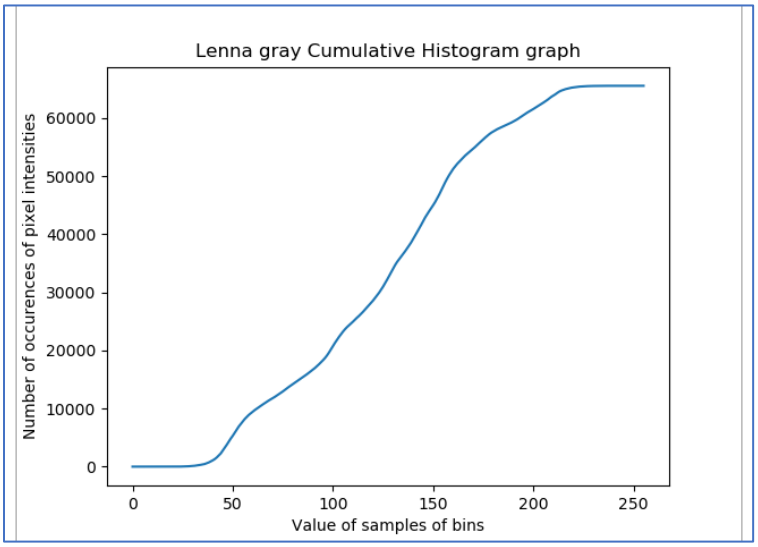


Figure 6: Accumulative Histogram distribution

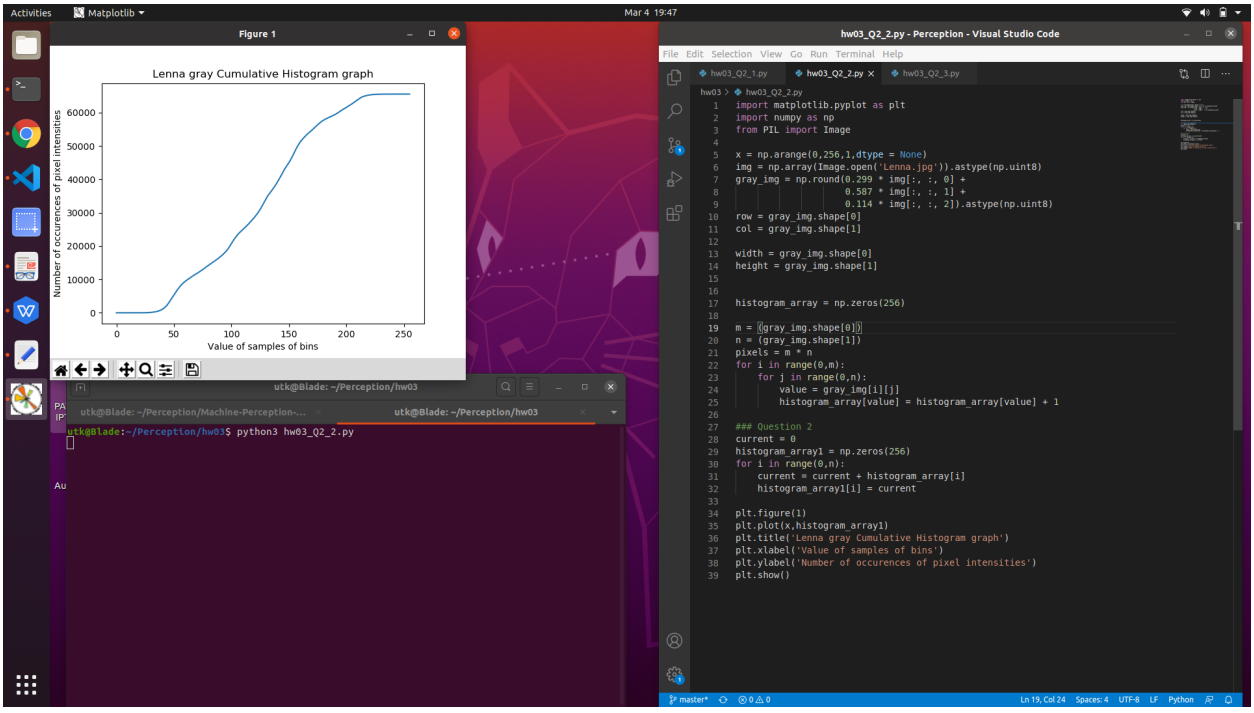


Figure 7: Accumulative histogram code and output

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:

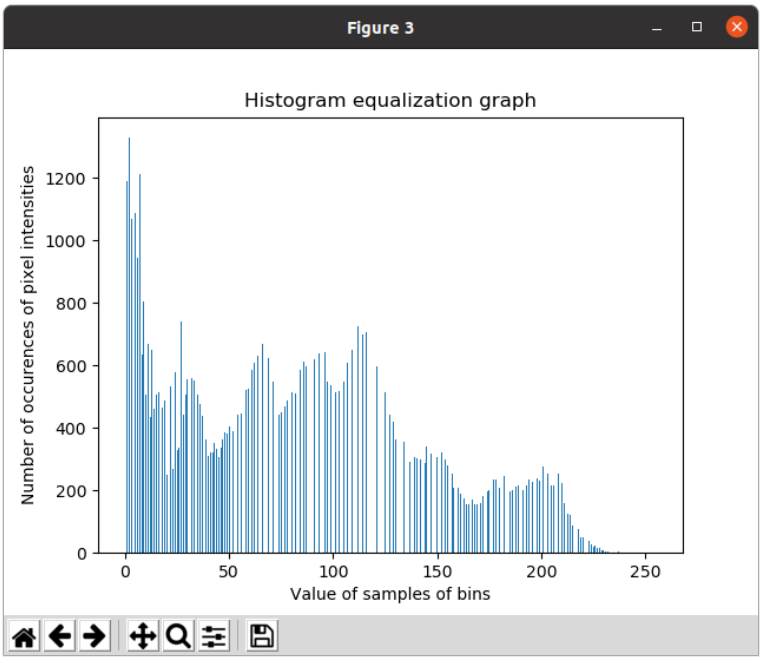
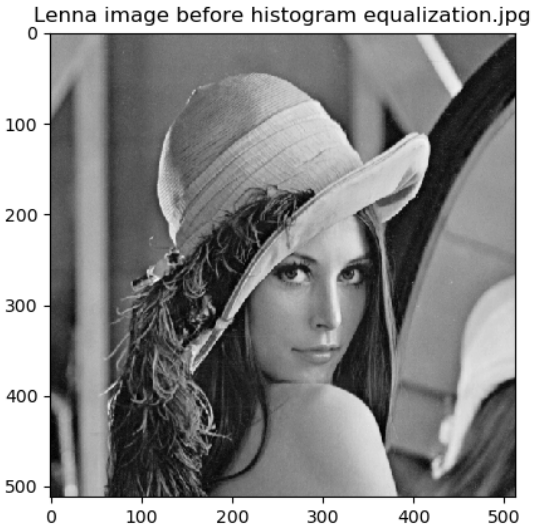
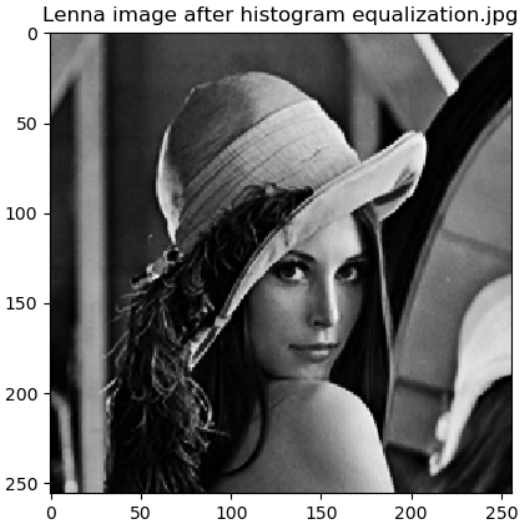


Figure 8: Histogram equalization





Comments on the before and after image:

- As we can see below, there is a difference between the two images.

- In the first method with histogram distribution, the intensities are distributed in the image and it represents the number of pixels for each of the intensity values which are considered.
- The histogram determines the number of pixels for each brightness level considering from the black and white to the color channels.
- Since we work on the gray scale image of Lenna, the peaks are reached since the image has high brightness level in most of the pixels in the image.
- While considering the second image after performing the histogram equalization, the contrast present in the images is increased to a great extent.
- This is accomplished by effectively spreading out the most frequent intensity values which means the stretching of the intensity range present in the image.
- By doing histogram equalization, the global contrast of images increases when the data is represented by close contrast values.
- Moreover, the areas of lower local contrast help in gaining a higher contrast. From, the graph it can be observed that the intensity is equally spaced but has peaks in the first part of the image because of the sharp hair contrast lines in the image.

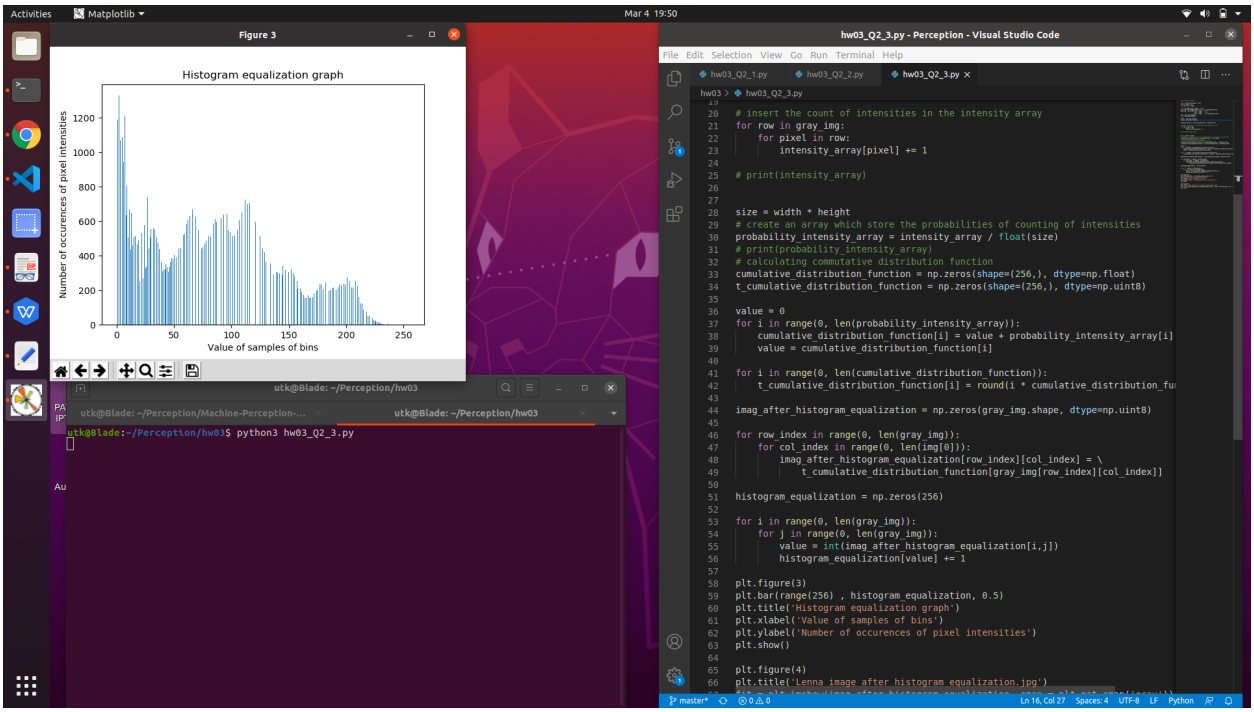


Figure 9: Histogram equalization code and output

Question 3)

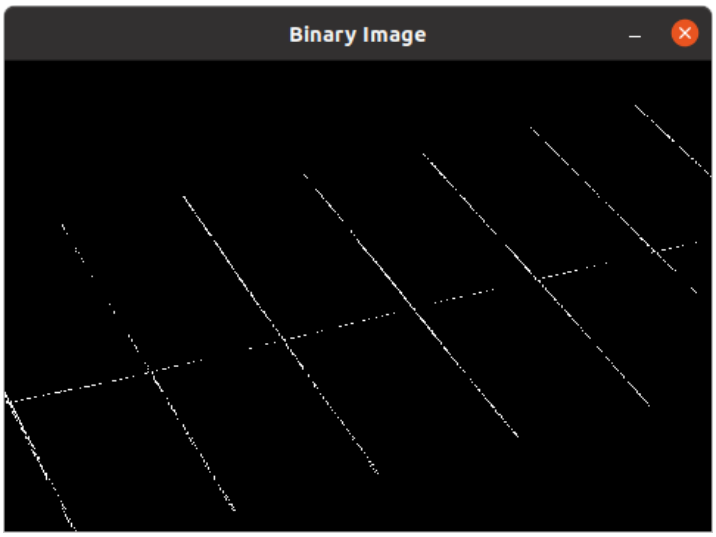
[Line Detection – 30 pts] Download the image "ParkingLot.jpg" from the hyperlink.

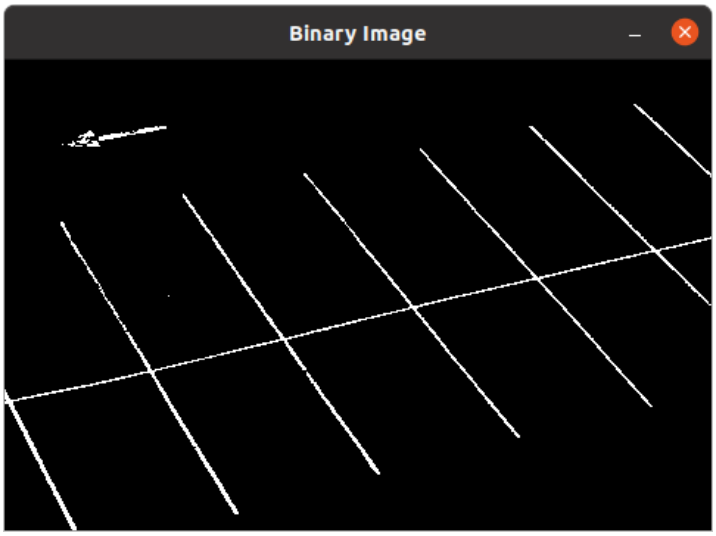
Note: For this question, you are free to use any 3rd party libraries.

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

Answer:

Threshold used 222. I used the Inverse binary type so that the lines would be white and the background black. I tried to remove the white arrow completely on the first image as shown below but ended up getting discontinuous parking lines. Hence, I decreased the threshold and let the arrow be in the processed image to get continuous lines.





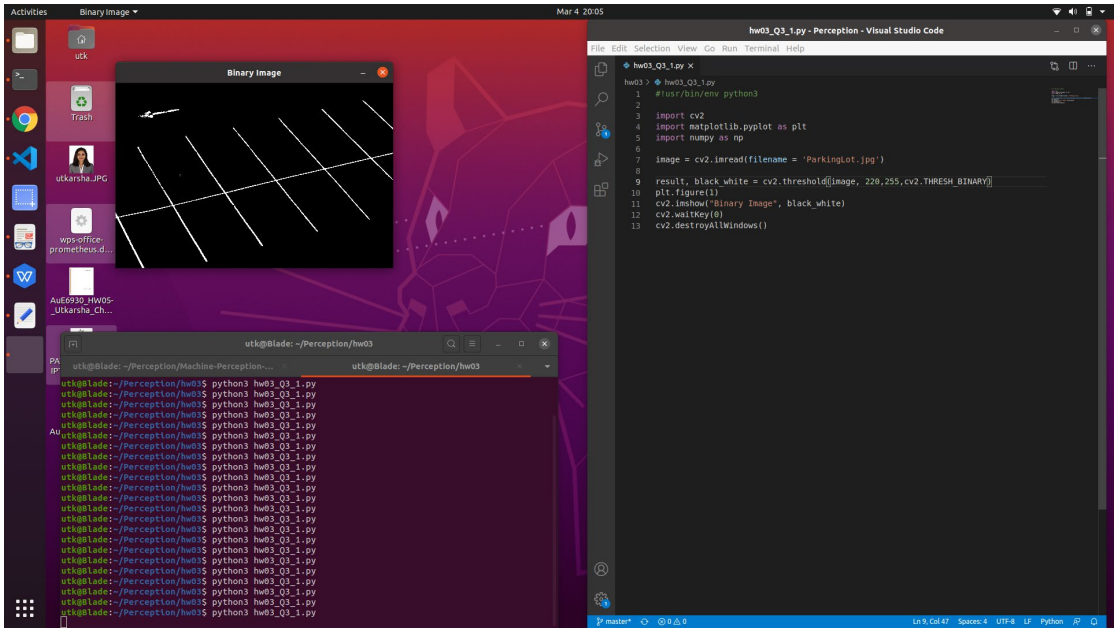
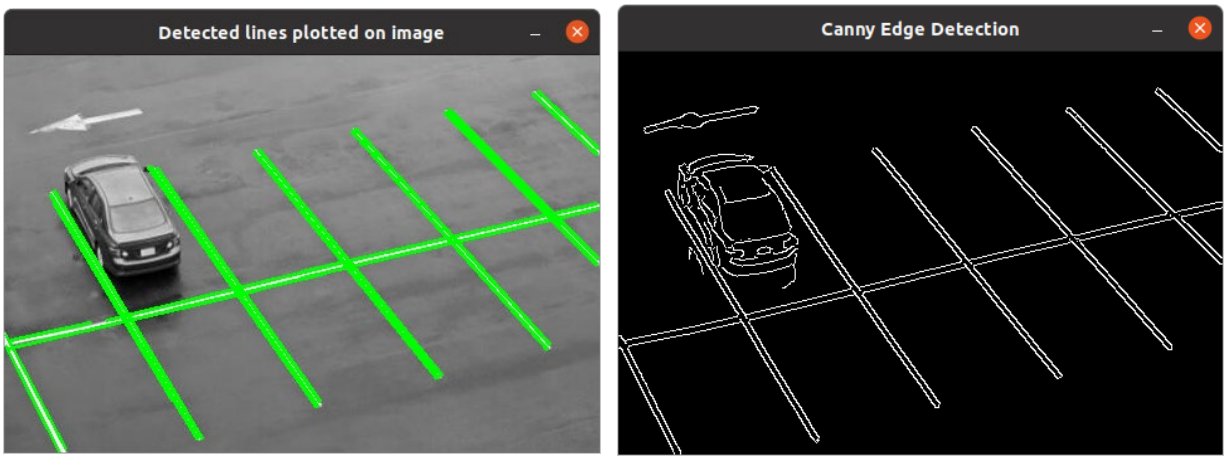


Figure 10: Code and output

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:

I used Canny edge detection and then applied Hough transform to plot the lines. As we were able to extract continuous lines from the image, the Hough transform was plotted perfectly.



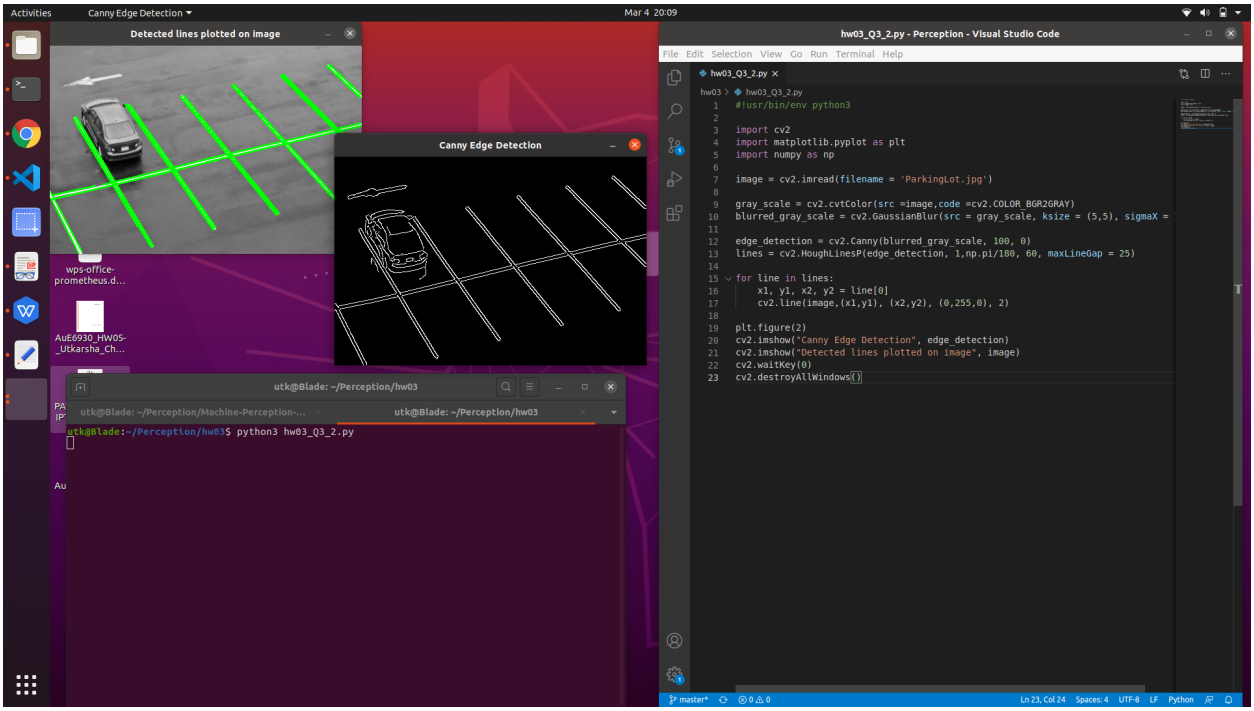
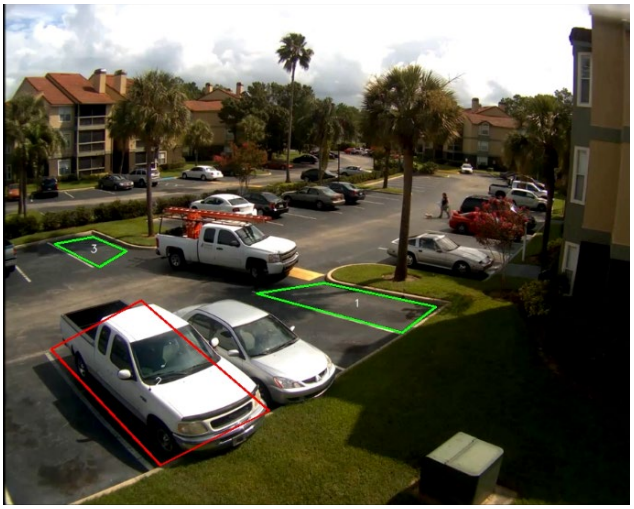


Figure 11: Hough transform code and output

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:

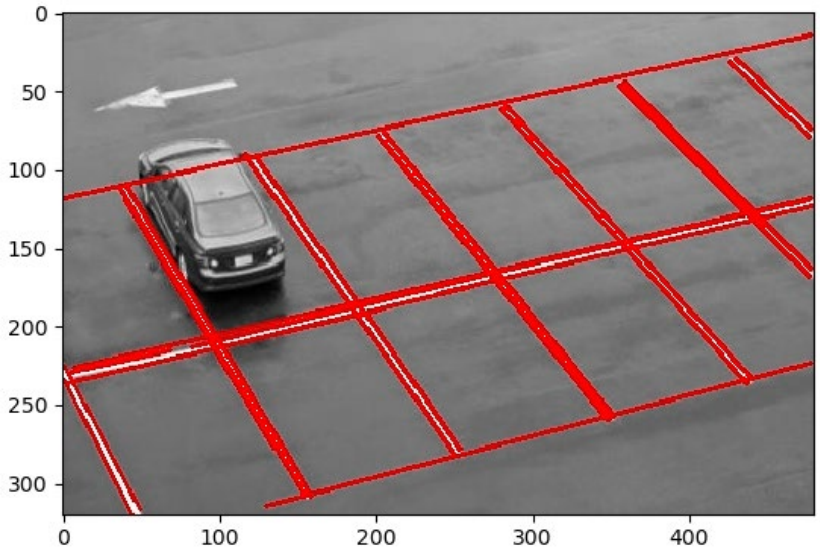
The two lines of a parking spot should always be parallel as this represents the space that a vehicle is supposed to be parked. The parking spots can be any shape but for packing more vehicles in a space, rectangle is the most efficient shape. In image processing, the two lines will not be always parallel as this depends on the perspective. See example below for reference.



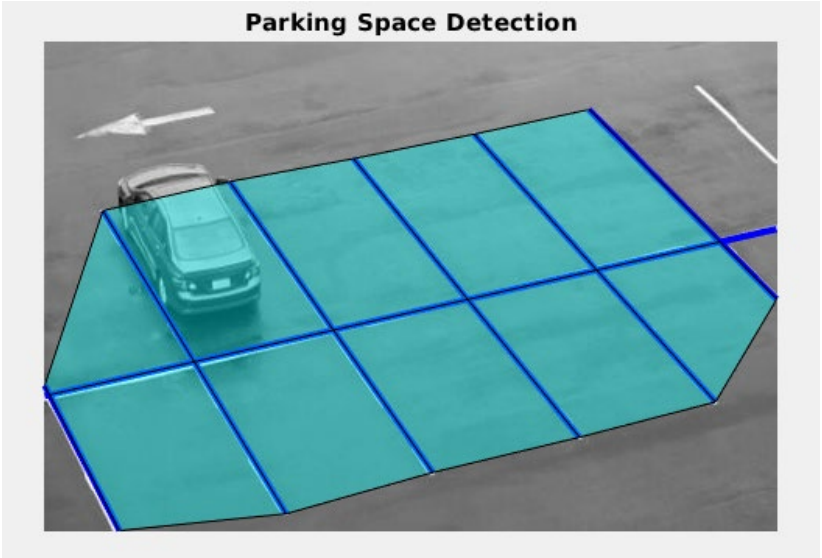
3-4) Design and implement the approaches to find all parking space polygons with the four vertex points for each parking space. Describe your approaches and visualize all detected polygons with different colors overlaid on the original image. The TA will check your code. (20 pts)

Answer:

I tried the code for this question in python, but I could not come up with the correct logic. My result was as follows:



Therefore, I coded this question in matlab and the result was as follows:



Question 4)

[Survey – 40 pts] Write a 2~3 pages survey report on a specific 2D-data measurement/detection problem related to automotive engineering.

- Please targeting at a specific 2D-detection goal/object. e.g.: lane detection, traffic sign detection, pedestrian 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.
- The detection target needs to be specific, rather than generic such as 'Obstacle' since it is not a specific target.

The grading of this question is based on the contents in the aspects of:

- The importance of measuring this target (5);

- The challenges of measuring this target (5);

- Existing approaches of measuring this target (15);

- Existing problems of these existing approaches (10);

- There will be other grading factors (such as novelty, organization, et al) (5);

* Attention: You are encouraged to include any drawing/table in the report;

* This survey is more focusing for the sensing and measurement of a 2D physical quantity or object, rather than comparing multiple 2D sensing modalities.

* You should not literally copy sentences from reference, and use "…" [1] to mark it if you really have to literally cite few sentences. For citations, use brackets （e.g. [1]) in the end of your statements, with reference list in the end of the report.

Answer:

The 2D quantity that I have surveyed is "Road sign detection".

1. <u>Importance of this measurement</u>

- When a driver is driving a car, he/she constantly monitors the road signs and behaves accordingly to ensure road safety.
- There are 3 types of road signs, regulatory, warning and guide signs all having equal importance for detection.

- Road signs are utilized as a method of warning and guiding drivers, helping regulate the traffic flow so that all the other components like pedestrians, motorcycles, bicycles etc. can use roads with safety and caution.
- For example, 25 years ago, 30% of the accidents occurred at intersections.
- This number has now gone down significantly due to the use of STOP signs at intersections.
- Hence, it is important for an autonomous vehicle to detect the road signs and act accordingly.

2. <u>Challenges of detecting road signs.</u>

- There are numerous challenges for detecting the road signs.
- Large datasets need to be created with all types of signs and from different perspectives so that the sign can be detected from any view.
- Different datasets need to be created for different regions or countries.
- This requires a lot of data collection and background work before you can implement the sign detection algorithm.
- These image sets work well for colorful or unique road signs but do not work very well for speed limits or signs with less features.
- The sign detection is mostly vision based and hence the camera data must be processed. Camera data is usually heavy and hence requires computation power for image segmentation and then identifying the region of interest and hence the road sign.
- The road sign recognition is a difficult task if we are attempting to detect and recognize sign image captured with unfavorable background. Complex background, weather conditions, lighting and shadows make the task complicated and difficult.
- Presence of objects in backgrounds with likely shape and color can also hinder the detection.[1]
- Disoriented or damaged signs making it hard for the system to detect and recognize.[1]
- Images acquired are usually blurred because of car vibration and speed.
- Poor visibility because of lighting and bad weather conditions.[1]
- Positing of road sign is also important for the system to detect the sign. Signs placed near trees often have portions hided by tree branches.[1]
- Color fading because of constant exposure to sunlight.[1]
- These are some of the challenges being faced for sign detection of autonomous vehicles.

3. **Existing solutions for measuring this target**

- These are some of the challenges being faced for sign detection of autonomous vehicles.
- Most common sensor used for traffic signs is vision based.

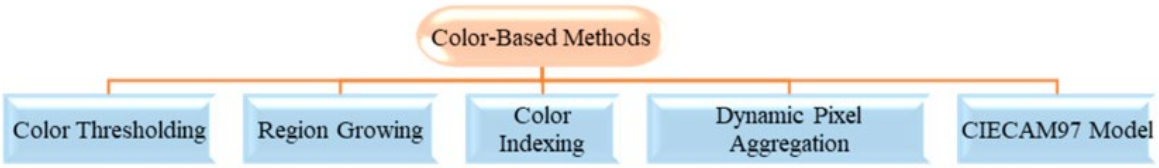
1) Neural network training

- Modern traffic-sign recognition systems are being developed using convolutional neural networks.
- The data sets are fed to the neural network and the algorithm learns which sign means what and hence recognizes it.
- This is a widely used deep learning technique.
- The same traffic sign is pictured in different lighting and from different perspectives so that the algorithm understands and is able to recognize it.

- the same traffic sign is pictured in different lighting and from different perspectives so that the algorithm understands and can recognize it.

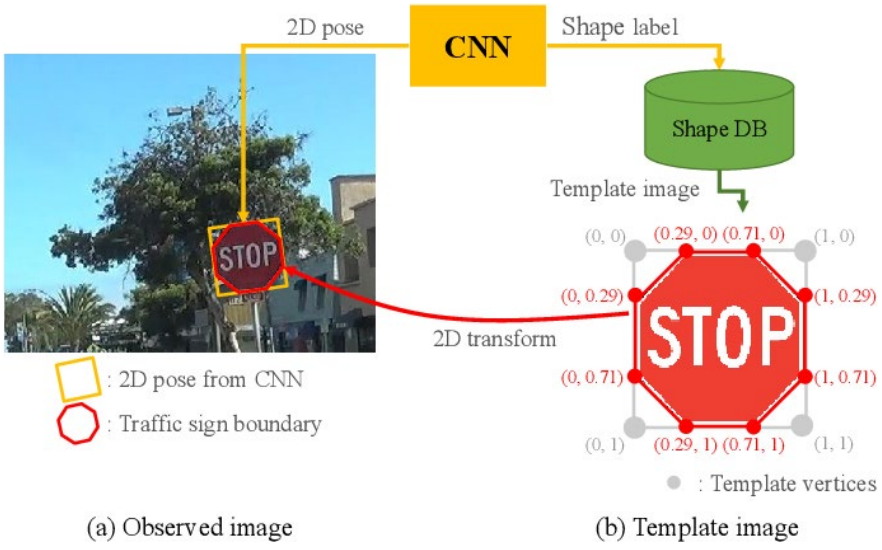
2) Detection depending on the colour of the traffic sign.

- Color-based methods take advantage of the fact that traffic signs are designed to be easily distinguished from the environment.
- These colors are then extracted to detect ROI within an input image based on the different image-processing methods.
- Detection methods based on color characteristics have low computing, good robustness and other characteristics that can improve the detection.

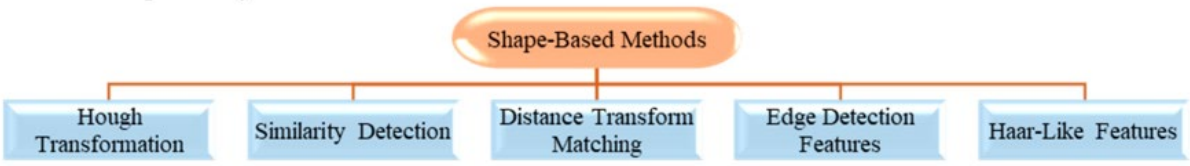


3) <u>Detection depending on the shape of the traffic sign.</u>

- Typical sign board shapes like hexagons, circles, and rectangles define different types of signs, which can be used for classification.
- The most common shape-based approach is the Hough transformation.

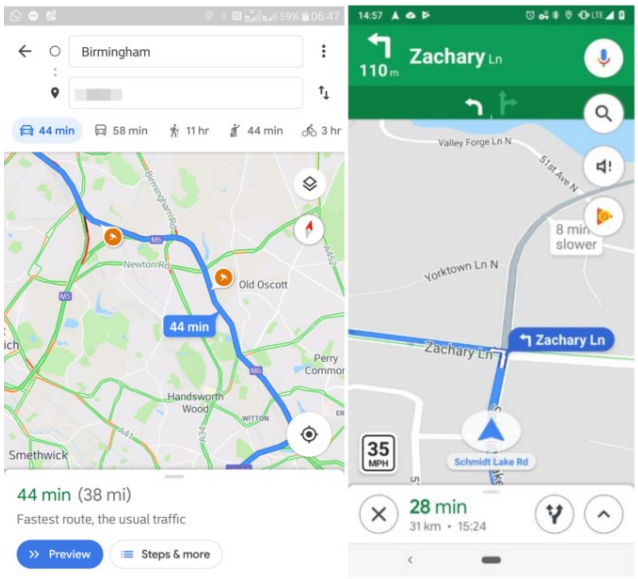


(a) Observed image (b) Template image



4) <u>GPS</u>

- Road signs information like the speed limit or the traffic signal can be pulled from the GPS.
- This is beneficial as detecting the speed limit road signs is rather difficult for a vision-based system.



4. <u>Existing problems the above methods</u>

1) <u>Neural network training</u>

- Even though the traffic signs have been generalized by the Vienna Convention, some regions have different signs. This makes it necessary to create new country-specific datasets.
- A lot of images in different weather and lighting conditions must be fed to the algorithm which makes. This means a lot of pre-processing is required.
- Even though thousands of pictures are fed to the algorithm, there might be unanticipated conditions due to which the algorithm is not able to recognize the signs.
- Hence all the pre-processing goes to waste.

2) <u>Detection depending on the colour of the traffic sign.</u>

- Methods based on the colour of the sign work with high-resolution colored dataset but not with grayscale images.
- In addition, the main problem with using colour as the parameter is it's sensitivity to various factors like weather conditions, time of the day as well as reflection, age and condition of the signs.

3) <u>Detection depending on the shape of the traffic sign.</u>

- For this method, the memory and computational requirement is quite high for large images.
- In addition, damaged, partially obscured, faded and blurred traffic signs may cause difficulties in detecting traffic signs accurately, leading to low accuracy rate.

4) GPS data

- GPS gives only limited data such as speed limit and traffic lights.
- Due to this limitation this is not a full proof method but just a supplement for any other vision-based methods.

References:

[1] https://www.ijert.org/research/image-segmentation-and-shape-analysis-for-road-sign-detection-IJERTV3IS120641.pdf