Assignment 2

Due Date: 2022/1/31

Assessment: 4% of total course mark

Instructions:

- For coding problems, please include the results as well as the screenshots of codes in the report
- Please upload source codes along with the report in Avenue (1 zip/rar file including codes, results and 1 PDF report file)
- The report MUST be written in Latex
- The codes MUST be written in Python language
- Please write comments for your codes!

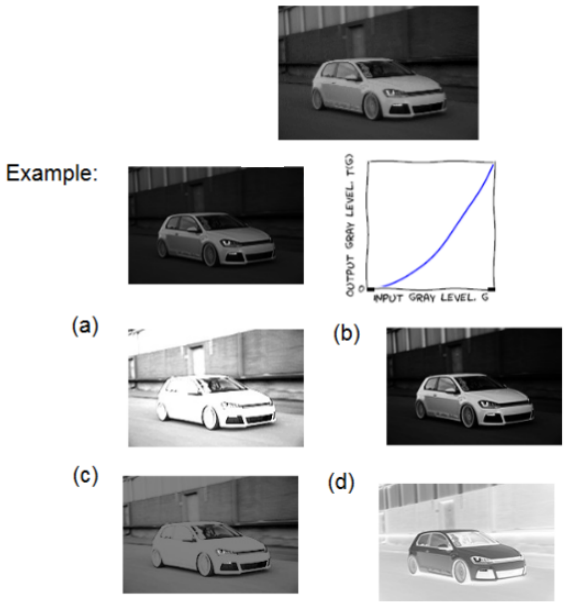
Theory (50 %)

1 Histogram Equalization, 10%

Consider the histogram (2, 2, 4, 8, 16, 32, 64, 128) where the number of gray levels is 8 (0-255 intensity levels). What is the output histogram of histogram equalization?

2 Transfer Function, 12%

Given an input image as follows, draw a rough sketch of the intensity transfer function that generates each of the following processed images.



3 Filtering in Spatial Domain, 12 %

Consider the image below.

$$\begin{bmatrix} 3 & 5 & 8 & 4 \\ 9 & 1 & 2 & 9 \\ 4 & 6 & 7 & 3 \\ 3 & 8 & 5 & 4 \end{bmatrix} \tag{1}$$

Show the results of 3×3 median filtering if the following masks are used. A "0" in a mask position means that the corresponding pixel is not used for median calculation. You can use zero padding of the image to have the output with the same size as input image.

a)

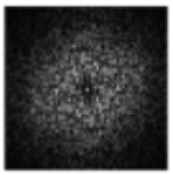
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2}$$

b)

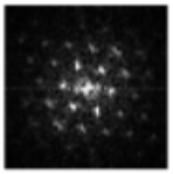
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3}$$

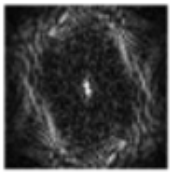
4 Fourier Domain, 16 %

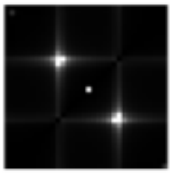
Pair the following frequency domain images with their corresponding temporal domain images. Briefly justify your answer for each pair.

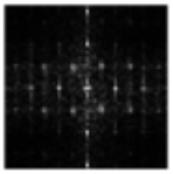
(a) 

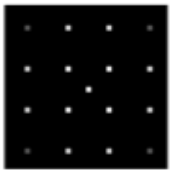
(b) 

(c) 

(d) 

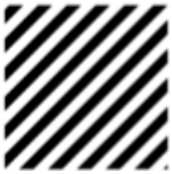
(e) 

(f) 

(g) 

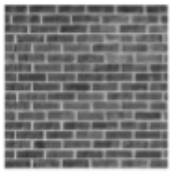
(h) 

A. 

B. 

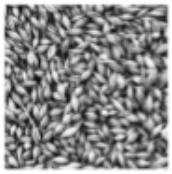
C. 

D. 

E. 

F. 

G. 

H. 

Implementation (50 %)

1 Background removal (10 %)

Explain how a static background can be estimated using median filtering. Now, write a program that reads "video.mp4", subtracts the background, and saves only moving vehicles. Save the estimated background as an image.

2 Image Manipulation in FFT domain (20 %)

Read "messi.png" and "ronaldo.png" as grayscale images. Take FFT of two images. Combine the magnitude of the first image with the phase of the second image and vice versa. Then apply inverse FFT to get two new images. Visualize the results. Explain the importance of phase and magnitude about the information that they contain.

2.1

Create a low pass filter in the frequency domain, then apply that on one of the images using logical functions (AND, OR). Then create a high pass filter (it can be easily obtained from the low pass filter you have created) and apply it to the other image. Take inverse FFT to get the resulted images and add them together ($\frac{1}{2}I_1 + \frac{1}{2}I_2$). Show the result in two versions of small and large images. Explain the results.
You may use the following functions: np.ifft, np.fft.fft2, np.fft.ifftshift, np.fft.fftshift, circle, bitwise_and, bitwise_not

3 Removing Noise (10 %)

Read "img3.png",

3.1

Remove noise from the image by applying a median filter

3.2

Remove noise from the image by applying a low pass filter (for example, Gaussian)

3.3

Which one works better? Why?

4 Histogram Equalization of Color Images (10 %)

Read "img4.png",

4.1

Apply global histogram equalization on each R, G, B channel and save the result. Show the histograms before and after histogram equalization side by side for each channel.

4.2

Convert image to LAB space, apply histogram equalization only on the L channel and save the result. Show the L channel histogram before and after histogram equalization.

4.3

Repeat part a and b with CLAHE function in openCV.

4.4

Compare previous results visually and explain which method works better.
