UNIVERSITY OF TEXAS AT ARLINGTON

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER VISION
ASSIGNMENT 4

Nudrat Nawal Saber

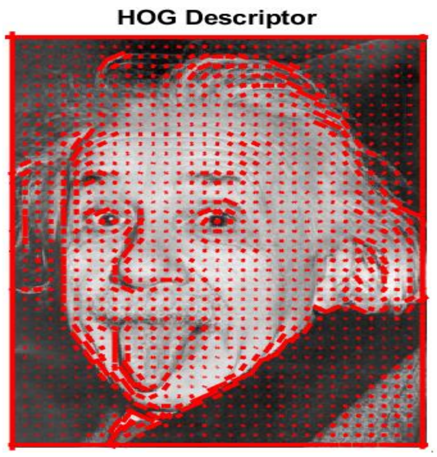
1001733394

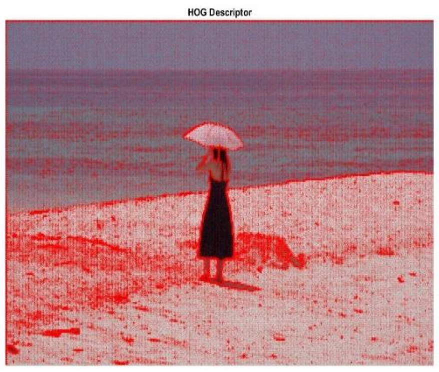
Problem 1:

Write a MATLAB function, [hog] = HOG(im), that implements Algorithm 1. The input, im, is a grayscale image and the output, hog, is a long vector of concatenated block descriptors.

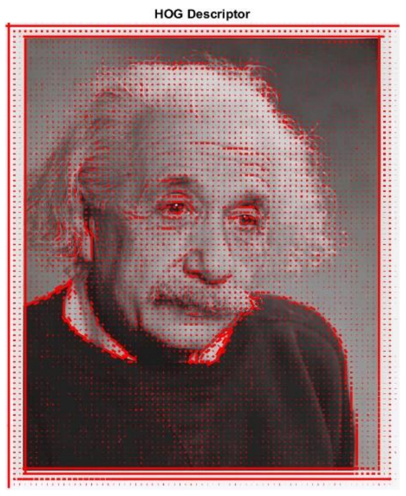
Ans:

I wrote the functions in MATLAB : HOG.m, that performs the operations in given Algorithm 1 .To run the HOG function ,we have to run the script main.m. I used the given image Einstein.png ,umbrella_woman.jpg, Einstein_2.jpg .
Hogdescriptors for those images look like the following :





a)Einstein b) umbrella woman

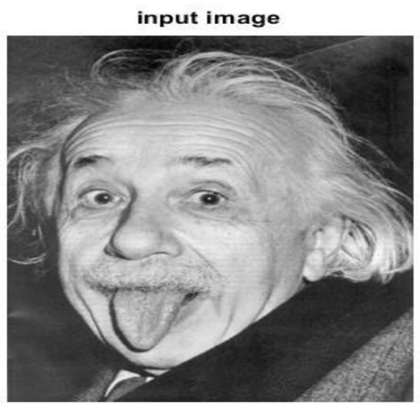


c) einstein_2

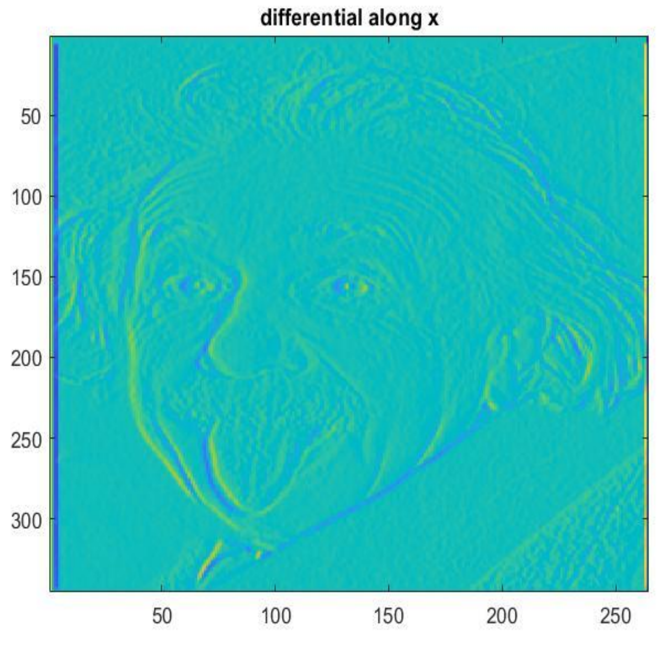
Problem 2:

Write a MATLAB function, [im dx, im dy] = filter image(im), that computes the gradient of the input image by differentiating the image along the x and y directions. The input, im, is a grayscale m × n image (Figure 2a) which must be first converted to double format using the built-in im2double function. The outputs, im dx and im dy, are the differentiated images along the x and y directions, respectively. To differentiate the image, use the 1D kernel [-1, 0, 1]. Note that you may need to pad zeros on the boundary of the input image to get the same size filtered images.

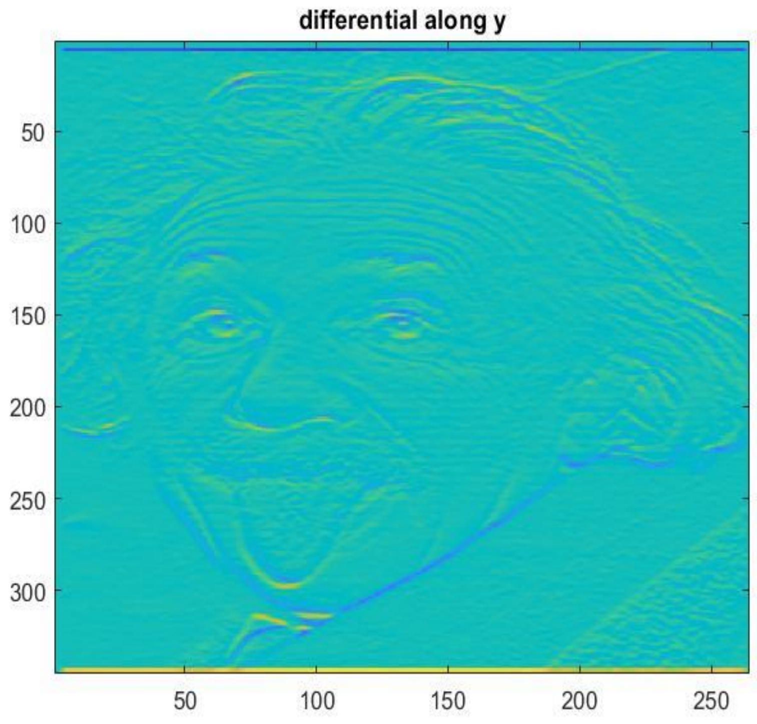
Ans:



a)Input image



b) Differential along the x direction

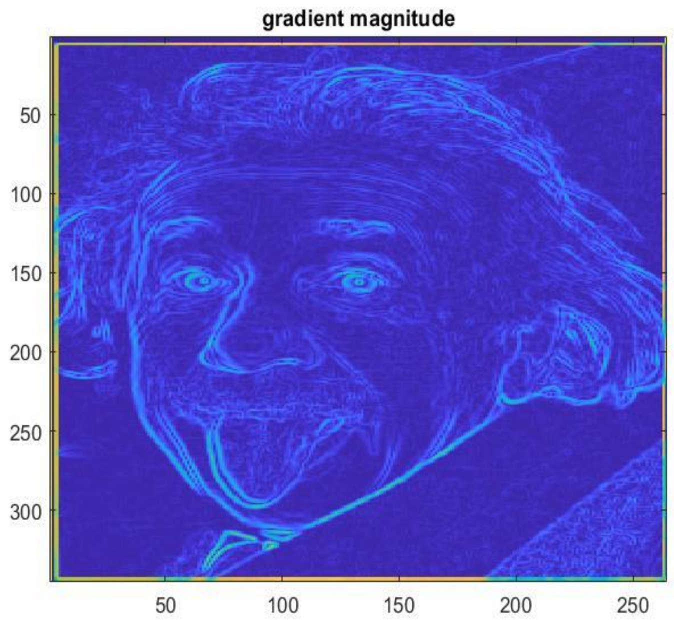


c) Differential along the y direction

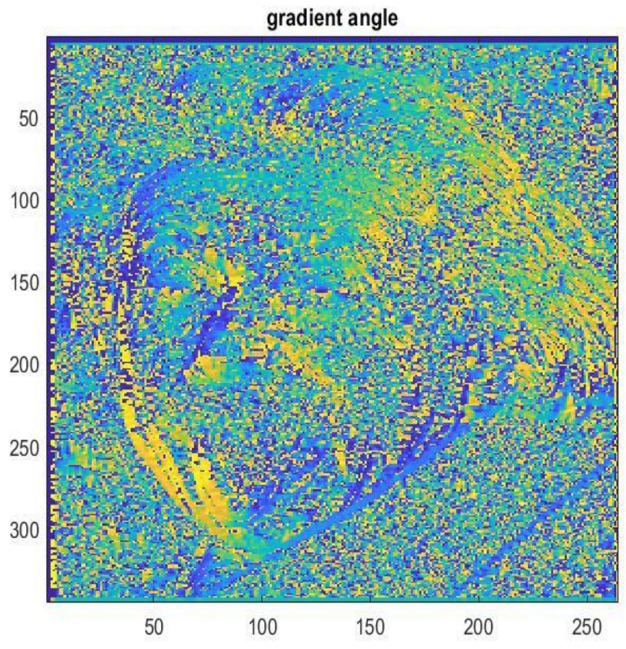
Problem 3:

Write a MATLAB function, [grad mag, grad ang] = get gradients(im dx, im dy), that computes the magnitude and angle of the gradient. The inputs, im dx and im dy, are the x and y differential images of size m×n. The outputs, grad mag and grad ang, are the magnitude and orientation of the gradient images of size m × n. Note that the range of the angle should be $\theta \in [0, \pi)$ and that the angle is unsigned (i.e. $\theta \equiv \theta + \pi$). By visualizing the gradients, you can see that the magnitude of the gradient is proportional to the contrast (edge) of the local patch and the orientation is perpendicular to the edge direction as shown in Figure 3.

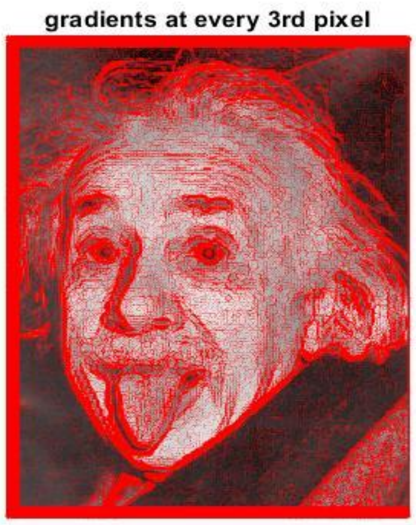
Ans:

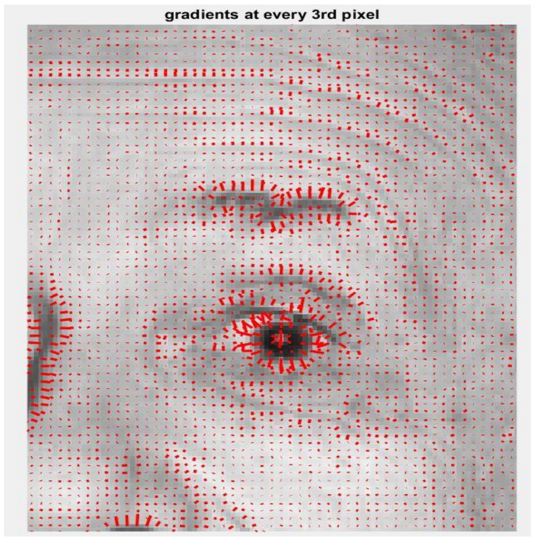


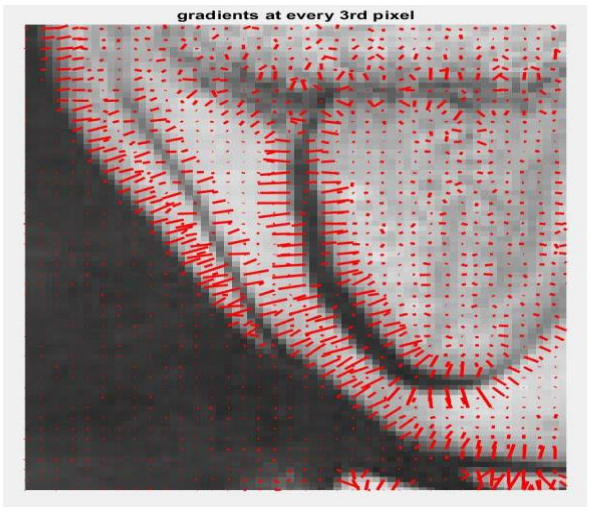
a) Visualization of the gradient magnitude



b) gradient orientation







c) gradients at every 3rd pixel

Problem 4:

rite a MATLAB function, ori histo = build histogram(grad mag, grad ang, cell size), that creates a histogram of oriented gradients for each cell. The first two inputs, grad mag and grad ang, are the magnitude and orientation of the m × n gradient images, respectively. The third input, cell size, is a positive integer that indicates the size of each cell. A typical cell size is 8. The output, ori histo, is a 3D tensor of size M × N × 6 where M and N are the number of cells along y and x axes, respectively. Specifically, M = bm/cell sizec and N = bn/cell sizec where b·c is the floor function as shown in Figure 4a.

Ans:

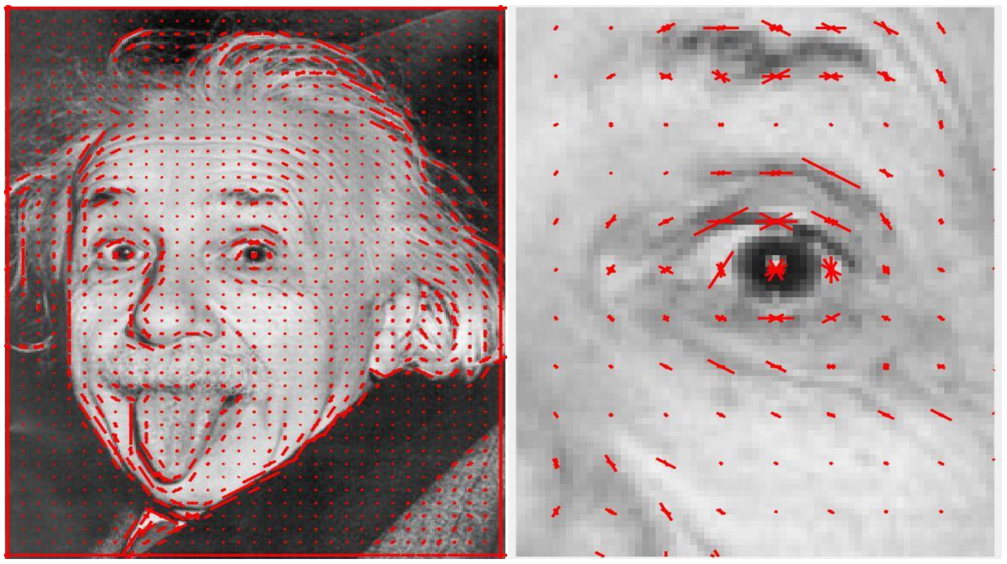

a. b,

Figure: Histogram of oriented gradients (HOG) features extracted and visualized for the entire image a) along with a zoomed in region (b).

Problem 5:

Write a MATLAB function, ori histo normalized = get block descriptor(ori histo, block size), that builds the normalized histogram. The inputs, ori histo and block size, are the histogram of oriented gradients without normalization and the size of each block (i.e. the number of cells in each row/column), respectively. The output, ori histo normalized, is the normalized histogram of size (M − (block size − 1)) × (N − (block size − 1)) × (6 × block size2).

Ans:

I wrote a MATLAB script, get_block_descriptor.m, that performs all the operations stated above.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in CVPR, 2005.