Name: ______________________________

Score: /15

CSE 5524 Computer Vision for HCI AU’22

Homework Assignment #2

Due: Tuesday 9/6

---

1) Perform Gaussian smoothing on the grayscale image affleck_gray.png (or affleck_gray_flip.png). Try with multiple sigma values, starting with larger values (e.g., 20 to .5). **When does the face become recognizable to your friends?** [2 pts]

```
sigma=1.0;  % use different values
G = fspecial('gaussian', 2*ceil(3*sigma)+1, sigma);
faceIm = double(imread('affleck_gray.png'));
gIm = imfilter(faceIm, G, 'replicate');
imshow(gIm/255);  % double images need range of 0-1
imwrite(uint8(gIm), 'gIm.bmp');
```

NOTE: If you choose to use Python, find similar function calls/operators.

2) Write a function to compute and display the 2D Gaussian derivative masks Gx and Gy for a given sigma. Sample the Gaussian derivative/gradient (2D) **equation** directly (see class notes) at the appropriate x,y (col, row!) locations. Note: each mask is a square 2D matrix. Please ensure that the positive derivative lobe is on the side of the increasing direction of each axis for each mask (see notes regarding the negative sign in the equations). Plot each mask (either in 2D or 3D). [3 pts]

```
[Gx, Gy] = gaussDeriv2D(sigma);
```

3) Compute and display the gradient magnitude of an image - **search the web** for an interesting image that has strong vertical and horizontal boundaries/edges; convert to grayscale if necessary (you know how to do this!); make sure to upload the image with your code in the Carmen submission. [2 pts]

```
gxIm = imfilter(myIm, Gx, ‘replicate’);
gyIm = imfilter(myIm, Gy, ‘replicate’);
magIm = sqrt(gxIm.^2 + gyIm.^2);
imagesc(gxIm);
imagesc(gyIm);
imagesc(magIm);
```

[next page]

4) Threshold and display the “gradient magnitude” image with different threshold T levels. [2 pts]

```
tIm = magIm > T;
imagesc(tIm);
```

5) Compare the above results with the Sobel masks. [2 pts]

```
Fx = -fspecial('sobel')';
fxIm = imfilter(Im,Fx);
Fy = -fspecial('sobel');
fyIm = imfilter(Im,Fy);
magIm = sqrt(fxIm.^2 + fyIm.^2);
tIm = magIm > T;
imagesc(tIm);
```

6) Run the MATLAB canny edge detector, edge(Im,'canny'), on your image and display the default results. How does it compare? (Python: you can use the scikit-image package) [2 pts]

7) Make a HW2.m (or HW2.py) script to do the above tasks and call needed functions. Create a report (PDF desired) with all results, printouts of images, and **discussion** of results. Zip all content (report, code, images) into Lastname_osudotnumber_HW2.zip and upload to Carmen for the grader. [2 pts]