Name: ________________________________

Score: /12

CSE 5524 Computer Vision for HCI

Homework Assignment #4

Due: See Carmen for due date

---

1) Write a function to compute the 7 **similitude** moment shape descriptors. Test and compare results on the rectangle box images 'boxIm[1-4].bmp' on the website (provide the computed moment values). Normalize each image before computing the moments so that the range of grayscale values is between 0-1. How do the moments change across the box images? Why are some moments zero? Please make sure your function will work with non-binary (grayscale) imagery, as you will need this for later assignments (do not use Matlab's regionprops function). [4 pts]

```
def similitudeMoments(image):
  # Your code to calculate 7-element similitude
  # moments vector here
  return Nvals
```

2) Using the datafile (eigdata.txt) provided on the WWW site, perform the following MATLAB commands [1 pt]:

```
#Load the data
import pandas as pd
eigdata = pd.read_csv("eigdata.txt", sep="\s+", header=None)
X=eigdata.to_numpy()

_, (plt1, plt2) = plt.subplots(1, 2)
plt1.plot(X[:,0], X[:,1], 'b.')
plt1.axis('equal')

#Mean-subtract data
m=np.mean(X,axis=0)
Y=X-m.reshape((1,-1))
plt2.plot(Y[:,0], Y[:,1], 'r.')
plt2.axis('equal')
plt.show()
```

[NEXT PAGE]

3) Compute the eigenvalues (V) and eigenvectors (U) of the data (stored in Y) using the function eig() in Matlab (*recall that you use either the covariance matrix or the inverse-covariance matrix of the data – see class notes*). Plot the mean-subtracted data Y and the 2-D Gaussian ellipse axes for given the eigenvectors in U (*you can use the plot command in Matlab for this. Make sure the axes have equal scale in the plot!)*. Use the eigenvalues in V to give the appropriate 3σ (*standard deviation - not variance!*) length to each axis (*did you compute the eigenvalues from the covariance or inverse covariance of Y? The eigenvalues will be related but different! See class notes*). [4 pts]

   [Note: it would also be nice to draw the 3σ ellipse around Y if you can – Google 'matlab ellipse.m' for some Matlab code if you are interested.]

4) Rotate Y using the eigenvectors to make the data uncorrelated (i.e., project data Y onto the eigenvectors – see class slides). Plot the results (using equal scale axes as before). [2 pts]

5) Perform a simple data reduction technique by keeping only the values resulting from projection of Y onto the eigenvector corresponding the largest eigenvalue of the covariance (not inverse-covariance) matrix. Plot a 1-D histogram of the values. Does it look like a 1-D Gaussian? [1 pt]

6) Submit a report containing all code, printouts of images, and discussion of results. Make a script to do the above tasks and call needed functions. Upload your report, code, and images to Carmen as usual. [No free points for this last step anymore!]

From now on you MUST submit (on time) your report, all code, and imagery on Carmen to receive points on the homework. Any portion missing will result in a 0 for the assignment!