CS 560 Advanced Topics in Artificial Intelligence
Computer Vision (Winter 2015)

HW3 (12%) & HW4 (8%)

- Assigned date: 2/19 Thu.
- HW3 due date: 3/1 Sun.
- HW4 due date: 3/11 Wed.
- Electronic submission via CSNS. Include your name and CIN# in your program.
- Submit source codes and readme.doc file. Do not submit any data or intermediate/executable files.
- Submit `readme.doc` file in your submission. Explain how to compile, run and test your program in very detail. Also, explain your methods in detail.
- If you program produce any compile or run time errors, you will NOT receive full credit.

1. **HW3:** PCA (1%)

   Using OpenCV, implement PCA and demonstrate your program using the data (i.e. ten 2D points) given in the PCA lecture slides (Slide #9).

2. **HW3:** PCA for Moon Crater Recognition – Part1 (11%)

   Download the compressed folder of crater images and decompress the file. The `CraterRecognition` folder consists of two sub-folders: `craters` and `non-craters`. The `craters` folder includes 2,631 crater images. The folder `non-craters` includes 2,969 non-crater pgm images. So total 5,600 images are in the `CraterRecognition` foler. Dimension of each image is 28x28. If needed, download Irfanview (http://irfan-view.download-assist.com/) to see pgm images.

   Step1: Training (6%)

   - Allow the user to enter a value between 50 and 100 for M.
   - Create a training set **TD** by selecting M crater images randomly.
   - Using M image, calculate eigenvectors of the data by applying the same approach shown in the eigenface lecture note.

     Let $C=AA^T$ (of size 784x784) and $L=A^TA$, where each image is a column vector of A.
     - Calculate the matrix L of size MxM and find its eigenvectors and eigenvalues.
     - Allow the user to enter a value for M’. (e.g. 20)
     - Choose M’ eigenvectors of L associated with the highest M’ eigenvalues. Let’s denote each eigenvector of L v_i, i=1,…,M’.
     - Calculate M’ eigenvectors, u_i, of C by multiplying A and v_i.

   - Display the first five eigencrater(eigenvectors), *of size 28x28*, associated with the highest five eigenvalues. *Each eigencrater should be displayed as an image.*

Step 2: Classifying Craters (5%)

- Select 50% of crater images and 50% of non-crater images. This set of total 2800 images is your a classification data set **CD**.
- Transform all images in CD into eigenspace using M' eigenvectors of C. Let's say transformed images are the pattern vectors.
- Cluster pattern vectors in eigenspace using any algorithm that you feel suitable. For example, you can use *k*-Mean clustering algorithm. Make your program limit the number of clusters be between 2 and 10.
- After clustering,
  - For any pure cluster consisting of only one class/type (crater or non-crater) of images, label the cluster and calculate the cluster mean vector by averaging all pattern vectors in that cluster. Some clusters represent crater class and some clusters represent non-crater class.
  - If any cluster is a mixture of two classes, check which class is dominant and assign non-dominant pattern vectors to the closest cluster of same class. Update cluster mean vectors of those effected clusters.
- Display the mean vector of each cluster. *Backproject the mean vector to the image space and display as a 28x28 image.*

3. **HW4:** PCA for Moon Crater Recognition – Part2 (8%)

Step 3: Query Input and Recognition (6%)

- Allow the user to choose a query input image from the given original data set. It can be either a crater or non-crater image.
- Calculate its pattern vector Ω of the input. (Ω is the transformed image into eigenspace using M' eigenvectors of C.)
- Classify whether or not the query input is a crater.
  - Allow the user to choose a classification method: Nearest Mean or k - Nearest Neighbor.
  - Classify whether or not the query input is a crater:
    - (Nearest Mean) find and display a class that minimizes the Euclidian distance $d_i = |\Omega - \Omega_i|$, Ω_i is a mean vector describing the i-th cluster.
    - (Nearest Neighbor) among all data in **CD**, find *k*=50 nearest neighbors to Ω using the Euclidian distance. Find and display the cluster that has majority in the 50 nearest neighbors.

Step 4: Recognition Rate (2%)

- Perform your recognition on all images in CD and *display your classification results in the format of the confusion matrix.* Also, display how long the program took to produce the result on all the images.
- Perform your recognition on all images that were not used for CD and *display your classification results in the format of the confusion matrix.* Also, display how long the program took to produce the result on all the images.

4. **HW4: Extra-credit 1**

   Instead of using the 28x28 crater image itself as an input vector for PCA, propose a different way to represent inputs vectors for this crater recognition problem. Perform all recognition steps using this new input vectors and compare results.

5. **HW4: Extra-credit 2**

   Propose a method or methods that can change any of the Step 1 ~ Step 3 and improve your recognition rate while reducing false detection rate.