Written Homework 10

Due Friday, March 19 at 5pm PST
(Reminder: Daylight Savings Time starts this week, so clocks move ahead by 1 hour)

As many of you have noticed, I have two cats. Their names are Boris and Nandor.

(a) Boris

(b) Nandor

In this assignment, you are going to write facial recognition code similar to what we made in lecture 31 to identify my cats.

I collected 75 pictures of Boris and 56 pictures of Nandor. Each picture is black and white and 200×200 pixels. I randomly chose 70% of the pictures from each cat to serve as a training data set. The other 30% will be used for testing.

Problem 1: Training

If you are using MATLAB, you can find the training pictures in the file `training_images.mat`. If you save this file in the same directory as your script, you can load the data with the code

```
load('training_images.mat')
```

If you are using python, you can find the training pictures in the file `training_images.csv`. If you save this file in the same directory as your script, you can load the data with the code

```
training_data = np.genfromtxt('training_images.csv', delimiter=',')
```

This will create a $93 \times 40,000$ matrix named `training_data`. The first 53 rows are pictures of Boris and the last 40 are pictures of Nandor.

(a) You can look at the picture in row `k` with the code

```
img = reshape(training_data(k, :), [200, 200])
imshow(img, [])
```

in MATLAB, or

```
img = np.reshape(training_data[k, :], (200, 200), order='F')
plt.imshow(img, cmap='gray')
```

in python. Look at all of the pictures, then choose your favorite one and include it in your writeup. (You do not need to include all of your code for this part - just the code used to show your favorite picture.)

(b) Find the average of all 93 images and display it using `imshow`. Attach the picture to your writeup. Make a new $93 \times 40,000$ matrix named `X` that is the same as `training_data`, but with the average image subtracted from each row.

(c) Calculate the **reduced** SVD of `X`. Calculate the scores for the training images by multiplying `X` by `V`. I will refer to this matrix as `scores` below. (You need to include your code in the writeup, but you don't need to answer anything else for this part.)

(d) Display the first two eigenfaces using `imshow`. Attach these images to your writeup.

(e) Calculate the re-scaled energies

$$E_k = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{93} \sigma_i^2}$$

and plot them versus k. Include your plot in the writeup (with an appropriate title and axis labels).

(f) Find the number of singular values that capture more than 99% of the information in this data set. That is, find the smallest k such that $E_k > 0.99$. Based on this number, do you think we could meaningfully compress this data set with the SVD? Why/why not?

(g) Reconstruct your favorite picture (the one you chose in part (a)) using the first 40 singular values/vectors. Display the image using `imshow` and attach the reconstructed image to your writeup. (Don't forget to add the average image after reconstructing your data.)

Problem 2: Testing

If you are using MATLAB, you can find the testing pictures in the file `testing_images.mat`. If you save this file in the same directory as your script, you can load the data with the code

```
load('testing_images.mat')
```

If you are using python, you can find the testing pictures in the file `testing_images.csv`. If you save this file in the same directory as your script, you can load the data with the code

```
testing_data = np.genfromtxt('testing_images.csv', delimiter=',')
```

This will create a $38 \times 40,000$ matrix named `testing_data`. The first 22 rows are pictures of Boris and the last 16 are pictures of Nandor.

(a) Make a new $38 \times 40,000$ matrix `Y` that is the same as `testing_data`, but with the average image from the previous problem subtracted from each row. Calculate the scores for the test data by multiplying `Y` by `V`. I will refer to this matrix as `scores_test` below. (You need to include your code in the writeup, but you don't need to answer anything else for this part.)

(b) Now we will identify the cat in the first testing image. Find the distance between the first row of `scores_test` and each of the 93 rows of `scores`. You can find the distance between two row vectors `x` and `y` in MATLAB with the code `norm(x - y)` and two 1D arrays `x` and `y` in python with the code `np.linalg.norm(x - y)`. Which row of `scores` gave you the smallest distance? Did this successfully identify the correct cat? (Remember, the first test image is a picture of Boris, and the first 53 training images are of Boris.)

(c) Repeat part (b) for each of the first 22 test images (all of which are pictures of Boris). How many did you successfully identify? What percentage were successful? Do you think this is a good success rate? (Keep in mind that

53/93 images in the training data are of Boris, so we would get this correct $53/93 \approx 57\%$ of the time by guessing randomly.)

(d) Repeat part (b) for each of the last 16 test images (all of which are pictures of Nandor). How many did you successfully identify? What percentage were successful? Do you think this is a good success rate? (Keep in mind that 40/93 images in the training data are of Nandor, so we would get this correct $40/93 \approx 43\%$ of the time by guessing randomly.)