University of San Francisco

Physics 301 – Computational Physics
Fall, 2015

HW08

Due: Thursday Dec 3, at 8 PM

Write a program and name it

`pca_letters.py`

For the images of the 26 letters in the English alphabet (provided), please do the following.

I. Image Preparation

For each image,

1. Display the image (`pyplot.imshow()`) and, by index slicing, crop the image — remove as much of the empty area around each letter as possible and remove a narrow band that sometimes appears near the right edge of the image.

2. Resample the images onto a coarser grid so that each image now has 16 × 16 pixels. You can use `scipy.interpolate.interp2d()` to do this. We will cover the usage of this function on Tuesday, Dec 1. If you want to get started sooner than that, feel free to consult

   http://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp2d.html

3. Subtract from each image its mean.

The following function should take an image file, `let_file`, and perform the above three tasks:

```
make_let_im(let_file, dim = 16, y_lo = 70, y_hi = 220, x_lo =
10, x_hi = 200, edge_pix = 148, plot_let = False)
```

The keyword arguments `y_lo, y_hi, x_lo, x_hi` contain the default values of the indices for cropping the image. The argument `edge_pix` is specified to remove the a narrow band that sometimes appears to the right of `edge_pix` (mentioned in 1. above) — the value of this argument varies depending on the image. If `plot_let` is `True`, plot the cleaned-up 16 × 16 image.

The function should return `let_im, let_im_flat`. The first is a cleaned-up 16 × 16 image of a letter and the second is that image reshaped into a 1d-array of 256 elements (a "flattened" image).

II. Data array construction

Construct a 2d-array that is 26 × 256 and consists of the image data for the 26 letters; call this array `X`.

III. Principal Component Analysis

Perform PCA on `X`. Call the PCA-projected array `Xproj`. Decide on how many PCA components are necessary for the reconstructed images for *all* 26 letters to be recognizable by eye.

First write a function to perform PCA on `X`:

```
alphabet_pca(X, n_comp = 5)
```

`n_comp` specifies the number of PCA components you want. It should return

```
pca, Xproj, pca_comps
```

where

`pca`: an object of the class `sklearn.decomposition.PCA`,
`Xproj`: contains the PCA projections (coefficients) in the form of an array with dimension `(26, n_comps)`,
`pca_comps`: PCA components (eigenimages).

Then write the following function that can be used to display the eigenimages and the PCA constructed image of a single letter, which is specified by `let_idx` (e.g., `let_idx = 0` corresponds to the letter "A"; `let_idx = 1`, the letter "B"; etc.):

```
show_pca_im(Xproj, pca_comps, dim = 16, let_idx = 0)
```

IV. Overall Structure

You should specific `let_idx` and `n_comp` at command line by using `argparse` in the program, and the program should take the following overall structure:

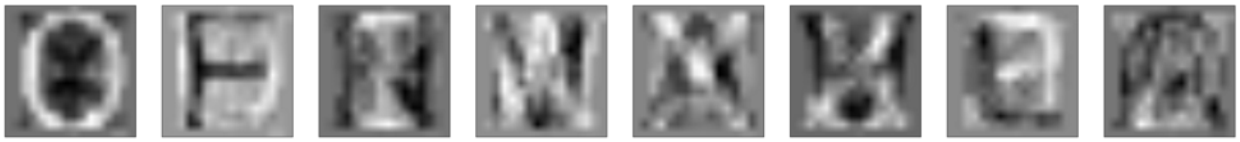
```
'''
Docstring

'''


# imports
import ...
import ...
import ...

# function definitions in alphabetic order
def alphabet_pca(X, n_comp = 5):

     ...



def (...):

     ...


def (...):

     ...

.
.
.


# ***************** Main Program *********************

# Input arguments

parser = argparse.ArgumentParser()
.
.
.


# Prepare letter images
A, A_flat = make_let_im('letterA.png', edge_pix = 135)
B, B_flat = make_let_im('letterB.png', edge_pix = 130)
```

```
.
.
.
Z, Z_flat = make_let_im('letterZ.png', edge_pix = 115)

# Construct data array for all 26 letters, X
...


# Perform PCA
alfbet_pca, Xproj, pca_comps = alphabet_pca(X, n_comp = n_comp)

# plot PCA reconstructed letters, one at a time.
# e.g., let_idx = 0 would show the image for letter A,
# and the PCA eigenimages.
let_coef = show_pca_im(Xproj, pca_comps, let_idx = let_idx)
```

At the command line, with

```
> python pca_letters.py -let_idx 5 -n_comp 8
```

You should get



The first image shows the 8 eigenimages, and the second image is the PCA reconstructed letter "F".

Please note that in order to do character recognition well, we really need a large data set that consists of all commonly used fonts of the 26 letters in the alphabet. In this homework, we have only looked at one font. It is a warmup exercise for you to become familiar with PCA.