EECS 442 Computer Vision: HW4

Term: Fall 2017
Instructor: Jason J. Corso, EECS, University of Michigan
Due Date: 11/03 23:59 Eastern Time

Constraints: This assignment may be discussed with other students in the course but must be written independently. Over-the-shoulder Matlab coding is strictly prohibited.

Goals: Deepen understanding of image as points concepts.

Data: No supplemental data on Canvas. Follow the links in the spec to download data needed.

Problem 1 (15): Haar Wavelet Transform

1. (5) Complete the Haar wavelet representation of following 1D image. Show the 8-element wavelet image.

2	4	2	0	6	2	1	7

2. (5) The wavelet basis provides a plausible means of compression by dropping certain coefficients. The most straightforward way of doing this is to cull the full set of coefficients of highest detail, i.e., set the coefficient layer immediately computed from the image to zero. For the 1D image above, compute the reconstructed image after compression.
3. (5) Consider a 1D image with 2^n pixels. Assume that the bit-depth is b (grayscale image values with range 0 through $2^b - 1$). Using the compression scheme above, derive an expression for the maximum error per pixel between the reconstructed image and original image. (The error here is computed by using l2 norm).

Problem 2 (30): House Number Classification

Recall the eigenface example discussed in the class. Now apply the same technique to classify house numbers. Below are some example images in the Street View House Number (SVHN) Dataset.

First of all, to get the basis of the digits, we use MNIST Dataset as training data. MNIST is a large database of handwritten digits. Download its training set images and labels from Yann LeCun's website. Every image contains a digit and has a size of 28×28. The original file is not in the standard image format (see how the pixels are arranged on the website). You may search for helper functions to read the images and labels (one can be found on this site).

Then download the SVHN test images from its official site. There are two formats of data, original images and cropped and centered images. Format 1 (`test.tar.gz`) is what we are going to use for classification. The annotation of images, including the labels and bounding boxes, are stored in `digitStruct.mat`, which can also be found in the downloaded folder. See detailed description about the format of the annotation on the official site.

You are required to implement a nearest neighbor classifier to recognize the house numbers for the **first 300 images** in the test dataset. Make use of the bounding box information to crop the image to the region of a specific digit and resize it to 28×28 (think about why). Note that there may be multiple digits in the same image and they are classified one by one to form a house number. Compare the house number generated by your program with the groundtruth and calculate the Number Error Rate and Digit Error Rate using the formula below.

$$\text{Number Error Rate} = \frac{\text{number of wrong house numbers}}{\text{total number of house numbers}} = \frac{\text{number of wrong house numbers}}{300}$$

$$\text{Digit Error Rate} = \frac{\text{number of wrong digits}}{\text{total number of digits in all images}}$$

Submission requirement:

Visualize first 10 basis calculated from MNIST that are associated with 10 largest eigenvalues and attach the images to your pdf writeup. Include one example of the house number that can be correctly classified and one example that fails. Also, report the error rate as described above. Submit the original program files to Canvas (do not include the dataset). Add a readme file to clarify the usage of your code. (See extra requirements below.)

Update: You should test your basis on the MNIST test set first, i.e., select a subset of test images (first 500, for example) and report the error rate in the pdf writeup. Then apply it to SVHN. Discuss why or why not your program works well on SVHN.

Optional: If you get very high error rate on SVHN dataset, you could try using a KNN set within SVHN but still using the MNIST basis, i.e., find the neighbors within SVHN instead of MNIST. But this is optional.

Submission Process (updated):

Submit a single pdf with your answers to these problems, include the code verbatim as text in the pdf. Include all plots and discussion in the pdf. Submit the pdf to Gradescope. The entry code of this course is **M5VRZV**.

Pack the original program files into one file and upload your code to Canvas. The problem description will clarify whether you should turn in the code for that problem.

Grading and Evaluation: The credit for each problem in this set is given in parentheses at the stated question (sub-question fraction of points is also given at the sub-questions). Partial credit will be given for both paper and Matlab questions. For Matlab questions, if the code does not run, then limited or no credit will be given.