CMPT 412 Fall 2017 Assignment 5: Face Recognition using Neural Networks

Due: 11:59pm December 4, 2017
(Late penalty will increment W, F as usual; however, it must be done by 11:59pm Sat. Dec. 9th)

The goal of this assignment is to give you just a little experience with designing, training and testing a relatively small neural network. Although this isn't a machine learning course, machine learning is now widely used to solve computer vision problems. Since the training phase can take hours or weeks even when multiple GPUs are available in order to obtain good results for many deep learning applications, this assignment uses a very small dataset and a not very deep network structure.

Create and test a neural network to identify peoples' faces. The recognition task and database are the same as used in your previous assignment on Fourier-based face recognition.

If you need another copy of the AT&T face database it's available here:
http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.tar.Z

A good example of how to create a simple neural network in MATLAB is available here:

https://www.mathworks.com/help/releases/R2017a/nnet/examples/create-simple-deep-learning-network-for-classification.html

That example is for the classification of handwritten digits. The face recognition/classification problem is similar and so this example should be quite useful to you. I suggest understanding that example in detail and then moving on to modify if for the face domain.

Note that the Neural Network Toolbox in MATLAB versions R2017a (the one that's currently installed on the CSIL machines) and R2017b are somewhat different. The example at the above link is based on R2017a. If you install R2017b on your own machine you'll find a variant of that example that uses the feature "batchNormalizationLayer". You're welcome to experiment with it, but make your assignment run work **without using** batchNormalizationLayer. This restriction is simply to make the assignment the same for everyone, whether they're using R2017a or R2017b.

Once you get your basic network running, experiment with preprocessing of the input images and the network architecture to see how this affects the performance.

- Image preprocessing
    - Use only simple preprocessing operations such as smoothing, contrast enhancement, normalization, etc. It generally does not make sense to use complex preprocessing techniques such as the Fourier Transform in this context. The input to the network should still be images of the faces (possibly preprocessed) at their original resolution.
    - Implement preprocessing using the ReadFcn attribute of the imageDatastore class

        - imagedatastore reference: https://www.mathworks.com/help/matlab/ref/imagedatastore.html
- Different network architectures
    - Numbers and types of layers
    - For each 2D convolution layer, the size of kernel, number of filters, padding, stride, etc.
    - Max pooling
- Different training settings
    - Regularization
    - Learning rate (can be dynamic too)
    - Training options reference material is available here:
        - https://www.mathworks.com/help/nnet/ref/trainingoptions.html

Use only 5 randomly chosen images from each class (i.e., person) for training and use the other 5 for testing.

As a starting point, you can use the same network as in the R2017a example, but with the AT&T faces dataset. You will of course have to change the dataset path, input layer size for your network (adjusted according to the size of the input images) and the fully connected layer size (based on the number of classes you have). After some experimentation with different network architectures you should get better accuracy than with your initial network.

You might also consider doing some tests similar to what you did for the previous assignment such as rotating the images slightly to see how such changes affect the performance.

There **will not be demos** for this assignment since classes are ending soon. Therefore, please provide a more detailed report on your work than for the previous assignments. In particular, submit via Canvas a single PDF file containing:

1. Your Matlab Code
2. Documentation containing:
    a. A description of the various network architectures (layers) you tried along with the different parameters you used and how they affected the classification accuracy.
    b. A description of the testing you have done with your network and the results you obtained in terms of recognition rate.
    c. Your thoughts on this method and what else it might be good for (if anything).

You don't need to use them but if you want faster training times then there are several CSIL computers equipped with GPUs. They are asb9840-a02 to a08 and asb9840-b##.