MPCS 53112 – Final Project Proposal
Team:
Riley Wetzel, Rehan Balagamwala

Project:
Image Profiling System: Using Neural Networks for facial detection and recognition to organize photos.

Background and Motivation:

Our motivation for choosing neural networks to perform facial detection and recognition for our project has a lot to do with the recent resurgence of both in the scientific community. Neural networks are incredibly powerful tools in their ability to handle a huge number of inputs and carry out many layers of multiplication. Until recently though, hardware has not been readily available at a cheap enough cost to make them a practical technique for most. Further motivation pertains more directly to facial recognition technology. As photos continue to increase in resolution, we continue to require more from our algorithms, in terms of sizes of input, however we also gain a lot of power to detect features within the extra pixels. With neural networks it is often the case that bigger is better. As well, newer software is able, quite accurately, to correct for lighting differences and shadows on faces and other features of images, a previous stumbling block with the problem. Finally, as the world becomes increasingly digitized there are many calls and opportunities for facial recognition programs. They have wide ranging potential applications in security (user authentication through facial recognition), and even law enforcement functions.

Our motivation for this particular application is the mass of unorganized photos on our computers we have all likely encountered at one point or another. We are hoping to use our detection and recognition algorithms to implement a system that lends well to photograph organization. It would facilitate easy compilation of all photos of your family or your best friend, or all photos of nature shots (no faces) to be put in one place. Apple's iPhoto has some of these capabilities but we are motivated by the idea of a slightly more regulated and personalizable structure to the photo library.

Description:

To reiterate, we wish to implement an analysis system of sorts utilizing neural networks with an end goal of an application that works to personalize organization of local photo libraries. Implementation of such a system consists of two general tasks contributing to the overarching goal.

1. Facial Detection: This is an initial binary question asked to a photo: is there a face in this image? If so, it is desirable to box the area of the image containing the face. From there depending on applications, continue to search the photo for more faces.

2. Facial Recognition: This is a separate but related question, once you have an image of a face: Does this face match any face in your library of images?

We wish to proceed by prioritizing these two tasks and hopefully implementing the more complete system in the end. We are going to prioritize the task of facial recognition and hopefully take advantage of some prepackaged python facial detection algorithms to bring everything together.

Methodology:

As mentioned in the description, there are a number of phases to implementing this system. At this point, the algorithm we wish to implement is based on one described in Tom Mitchell's machine learning book which can be found here, http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/mlbook.html. There is a database referenced therein that we would like to use, but no longer seems to be available. As such we plan to move forward with the set of images available here http://pics.psych.stir.ac.uk/2D_face_sets.htm. We will proceed with preprocessing of images that will include mostly normalization and compression. For example, for our network it will be important to feed in images that are uniform in pixel dimension. Our initial goal is a basic implementation of the system, which we can hopefully beef up as we go along.

For facial recognition we wish to implement the neural network method described in the above source. We will implement a supervised from of this algorithm in which we hand tag images with the name of the person pictured and use those as targets. We plan on using multilayer networks learned by the Backpropagation algorithm, which is capable of expressing a rich variety of nonlinear decision surfaces (as opposed to expressing linear decision surfaces if using single perceptron units). The Backpropagation algorithm learns the weights for a multilayer network with a fixed set of activation units and interconnections. It employs gradient descent to attempt to minimize the squared error between network output values and target values for these outputs.

We plan on using gray scale images to train a facial recognition network. We have yet to decide if we are going to use one big network to train the network with each person getting his own output node, or if we are going to train an independent network for each person separately. This determination will be made based on model performance. Once we have raw image files, we will compress them using k-means clustering algorithm. The pixels in each of the compressed images will then be normalized (from 0-255 to between 0-1 to match the target vector scaling). Each pixel will then be fed into its own input node. Input units will connect to hidden layers via full mesh (we will test the model performance based on 1+ hidden layers). The hidden units in turn will connect to several output units. Finally, we will experiment with different error minimization procedures (most interestingly, minimizing the 'cross entropy' of the network) and dynamic network structures ('Cascade Correlation').

For the task of facial detection, for which we will not initially prioritize a unique implementation, we will build off of what is available in the open source computer vision library OpenCV. There is a neat implementation of facial detection in python that can be found here https://realpython.com/blog/python/face-recognition-with-python.

Putting the whole system together is the ultimate goal for the project, if we can integrate all of these components together. This is a little more difficult task since our image recognition algorithm, as of now, will rely on relatively clean and uniform, in terms of illumination and facial expressions, images and the faces picked out of images by the OpenCV algorithm may not be very clean. The hope is that we could extract those segments of images and compress them to uniform size and then run them through our network to find photos with the same face in them.

Evaluation:

Evaluation will have to be on the grounds of an f1-score. This isn't a life or death situation, but we still want quality and consistency in our product. Thus, we do not favor false positives or false negatives over one another as you neither would necessarily be better than the other for organizing photos. So we will initially use cross-validation and aim to optimize the f1_score with respect to correct classification in facial recognition tasks and number of accurately found faces in facial detection tasks.

Deliverables:

By mid-quarter we aim to have our facial recognition algorithm functioning on a clean data set (for training) to the point that we can run images through it. This may depend on how much image preprocessing we must do though. And, if not at the same time, soon after we will also have a functioning facial detection module.

It's hard to say what will be in order at the final stage, but our ultimate goal is a system that can handle the organization of photos based on who is in the photo or based on the presence or absence of faces in a photo. Given the time constraints of the quarter, it is unlikely we will be able to integrate the entire system, but a basis of complete modules for both detection and recognition would be nice so that integration of the two with a view toward a final product could be a (hopefully less analytically technical) next step to continuing the system, perhaps after the course has concluded.

Collaboration:

We have a joint github repository/org that we are going to use to code our algorithms together.