BMI/CS 567 Medical Image Analysis
University of Wisconsin-Madison
Final Project

Instructor: Jeanette Mumford
Due May 10th *by* noon

Please turn in your solution via Canvas, but simply upload MATLAB code in a .m file instead of mlx or pdf, since we will be running your script all at once (e.g. see `run` function). You will also submit a word document (or something similar) that describes your approach. I'd expect this document to be around one page long.

For this final project you will be using multiple skills learned over the semester to run a classification analysis. The ultimate goal is to classify a set of data from a retinal imaging database from the STARE (STructured Analysis of the Retina) Project. I'm setting this up in the style of classification contests that I have seen in the past, here is an example of a recent one. You will get a subset of 36 images and their labels (normal/retinopathy) and you will need to analyze the images to extract features and then use these features to classify whether the image is normal or retinopathy. Then I will test your feature extraction code and trained classifier on a set of data you didn't have access to. Normally in the contests, this is what determines the winner. Feel free to look into retinopathy if you'd like, here's a short video about it, if you're interested. To make the challenge easier, I'm only having you classify between healthy and retinopathy, even though there are many types of diabetic retinopathy. Below is an example of a healthy (left) and unhealthy (right) retina. As you can see, there are some pretty striking differences. The challenge is to develop automated procedures to extract features that quantify these differences and successfully classify the image. A task that is fairly easy to do with our eyes, but more difficult to automate!

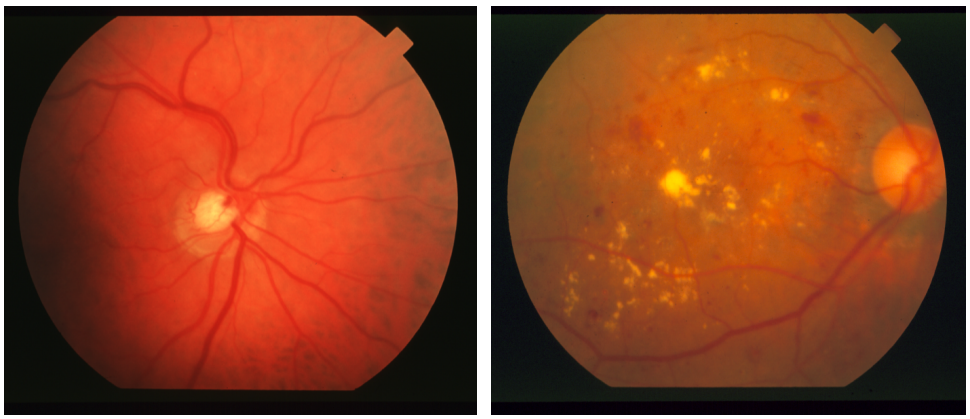

Figure 1: Left: Healthy retina, Right: Retinopathy

What you have been given in the zipped directory, `distributed.zip`:

- 36 retina images (ppm format)
- Text file containing the labels where half were diagnosed as healthy and half retinopathy.

What is required

1. Your code must be written very cleanly and I expect you to turn in a single, main .m file and possibly some extra .m files that contain subfunctions (not required). Unlike the homework, the grader and I will be running your code on our local machines. Your code should be set up so we only need to change 2 paths: one to the data that were distributed (so we can check your feature extraction and cross validation) and the second will be to the data set you have not seen, which will we use to test your approach. You should test your code thoroughly before submitting and can simply test the second part using the data you do have. It should run, seamlessly, by typing the path to your script at the MATLAB command line or, equivalently, using the `run` function. You will also submit a text document (word is fine) describing the methods used for feature selection, how they relate to what we've learned in class, what classifiers you chose and why, how many folds you used in your cross validation and why.

2. Feature extraction. Features are measures that you extract from an image. Basically, if you look at the images you will see things that will likely discriminate between the two types of images and must figure out a way to quantify it.

    - You need to generate at least **three** features to extract from each image. Creation of a single feature can often be a multi-step process and it is required that each feature use at least 1 method learned in class. You may use some of the built-in MATLAB functions in addition to a class approach for a single feature. Each feature estimation must use a unique method from class. For example, you cannot simply mean filter as a step for all features and count that as the class-based approach. You cannot download additional functions from the internet, but you can use anything that is included in the university-based MATLAB version.
    - You are also required to describe each method you used for feature selection clearly. If you used a new function that we didn't cover in class, you must define it and relate it to what we learned in class. Please include these in a separate text document (word is fine).

3. Classifier. Next you will run a cross validation using your favorite two classifiers that we covered in class. It is up to you how many folds you would like for your cross validation, but add a description to the word document writeup of the classifier you chose, the number of folds you chose and why. The output from this portion of code should be a classification accuracy within each category for each classifier. You should label this clearly in the output.

4. Left out test set evaluation. Last, you must choose which classifier you are using on the left out training set. You should supply code that will run the training step on the full set of distributed data and then run the test step on the left out data. As I mentioned above, your code should be set up so we only need to change 2 paths: the path to the distributed data directory and the path to the final test data set that was not distributed. You will not have this test set when you're doing the project, so I would just use the distributed data twice to ensure your code works.