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

Instructor: Jeanette Mumford
Due May 6th *by* noon

Please turn in your solution via Canvas, but simply upload MATLAB code in a .m file instead of mlx or pdf, since I will be running your script all at once (e.g. see `run` function).

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 using these 36 images in a cross validation. In the second step you will supply code that trains the classifier on all 36 images that you were given and then tests on a data set you have never seen. 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. I highly recommend looking at papers on the topic to get some tips for extracting the features. To make the challenge easier, I'm only having you classify between healthy and retinopathy, instead of classifying 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.

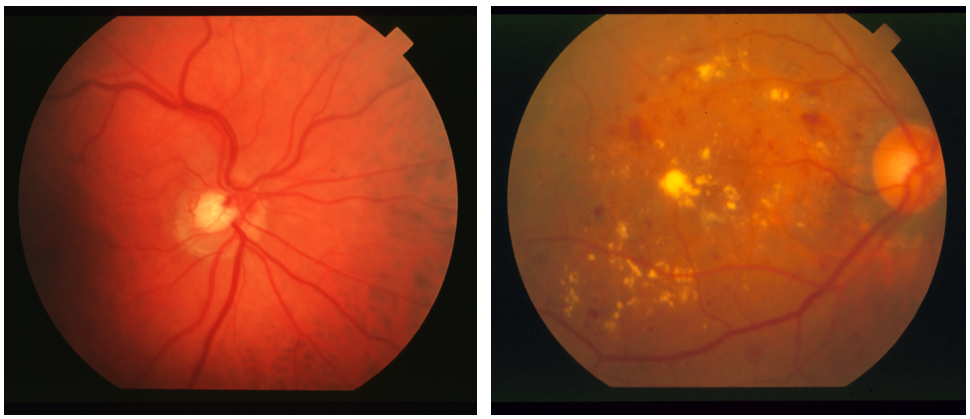

Figure 1: Left: Healthy retina, Right: Retinopathy

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

- 36 retina images (ppm format)
- The first 18 images are healthy retinas and the second set of images show retinopathy

What is required

1. Your code must be written 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, I will be running your code. Your code should be set up so I only add label information for a data set you didn't have access too (test set) and I'll change 2 paths: one to the data that were distributed (so I can check your feature extraction and cross validation) and the second will be to the data set you have not seen, which I will 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.

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 **two** features to extract from each image. Creation of a single feature will 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 MATLAB toolboxes that came with the University installation.
    - 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 add a few comments to your code that explain the method and how it relates to what we've learned in class.

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. Since there are 36 images, a 6-fold CV probably makes the most sense. Your code should display the classification accuracy. I would expect your classification accuracy to be higher than 60% and it shouldn't be difficult to get even higher than that. If your accuracy is lower than this you will not get full credit. You will also be graded upon whether you did your cross validation correctly.

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 that you didn't have access to. As I mentioned above, your code should be set up so I only need to change the image labels for the test set and 2 paths: the path to the distributed data directory and the path to the final test data set that was not distributed. There will also be 36 images in this data set and the naming convention will be the same, but they will not necessarily be ordered the same (referring to healthy/unhealthy labels). 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. Your code should output the classification accuracy for the test set.