Milestone

Original Goal:

My original goal was to obtain a proper data set and narrow down my potential set of classification methods to one specific method.

Progress On Goal:

I tried numerous methods to try to obtain a data set for the Telugu Alphabet and ran into the following problems (essentially in order):

- The Telugu Alphabet has 58 (18 vowels and 40 consonants) letters, and each of these letters can be combined in a number of different ways, resulting in hundreds of different symbols that need to be distinguished. Therefore, any data set that contained all of these letters would have to be huge. I therefore, knew that I would have to work with a subset of these letters if I was to try to create my own data set.
- To create a data set, the simplest method would have been to get several people to write out the letters and then scan in the results. However, this method would case several problems.
    - Finding more than 15 different people in the Hanover area who actually read and write Telugu would be difficult. While I could potentially get non-Telugu speakers to write out the alphabet, this practice would defeat the purpose of the exercise since they would show little to no inconsistency.
    - In order for the data set to be usable, the letters would have to all be approximately the same size and position on the paper. Size, to maintain resolution and image size, and position, because many Telugu letters are very similar in shape and various letters can only be told apart due to the varying positions of different symbols.
- When my first idea failed, I attempted to get sample letters out of scanned texts on the internet. However, as I stated above, there are hundreds of different symbols in the Telugu language, since each consonant can be combined with up to 1 vowel and up to 1 consonant:

    So the number of symbols:
    18 vowels + 40 * 19 * 41 consonants = 31, 178 symbols
    Not all of these are practically used… but that still leaves nearly 25, 000 symbols

  It was therefore, difficult to consistently find the same symbols in different handwritings. I also once again, ran into problems with the image resolution.
- Finally I attempted to contact professors who were researching this particular problem. My quest for a data set still failed, however, in doing this I learnt that my inability to find a data set was not due to a lack of persistence on my part, but was rather, because there were no publically available data sets for me to use.

- Finally, I ended up understanding that sometimes, you fail, and you need to make the best of it and move on.

The New Problem:

Having put all this thought into this problem, I decided that it would be best for me to stick to a similar problem, and I have therefore moved onto trying to classify handwritten digits.

- **The Data Set:**

  The data set I'm using can be found at:
  And belongs to Semeion Research Center of Sciences Communication, Rome, Italy.
  It consists of 1593 images, each represented as a 1 by 256 matrix (to be reshaped into a 16 by 16 matrix).
  The images have been converted into black and white by rounding the value of each pixel into a 1 or a 0.

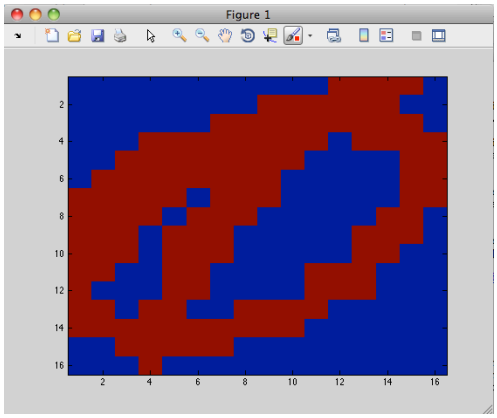

Figure 1: *An image from the data set representing the digit 0*

  I then split this data set into a training set and a testing set.

- **The Algorithm:**

  So far, I have used PCA and the Fisher Linear Discriminant to reduce the dimension of the images, and a k-Nearest Neighbor Classifier on the dimension reduced data.

  The Results of PCA:

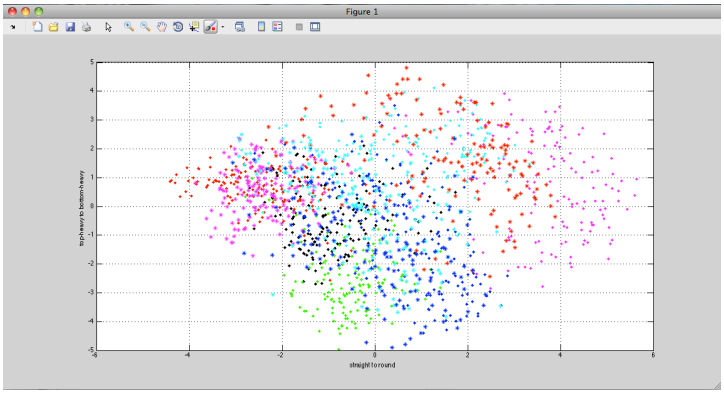

Figure 2: *PCA to 2 dimensions on the train dataset.*
The x-axis is roundess of the symbol and the y-axis is bottom-heaviness

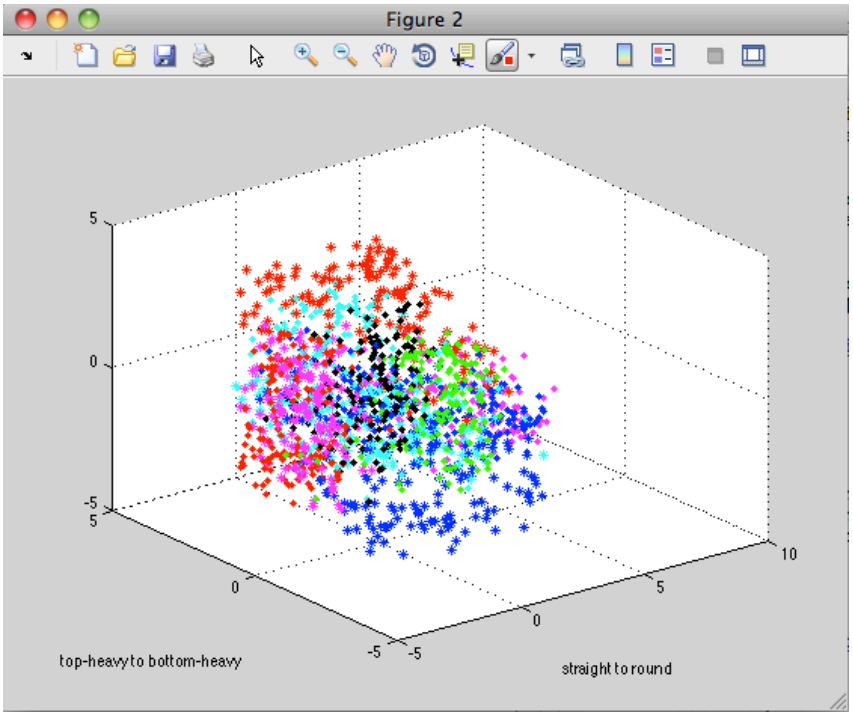

Figure 3: *PCA to 3 dimensions on the train dataset*

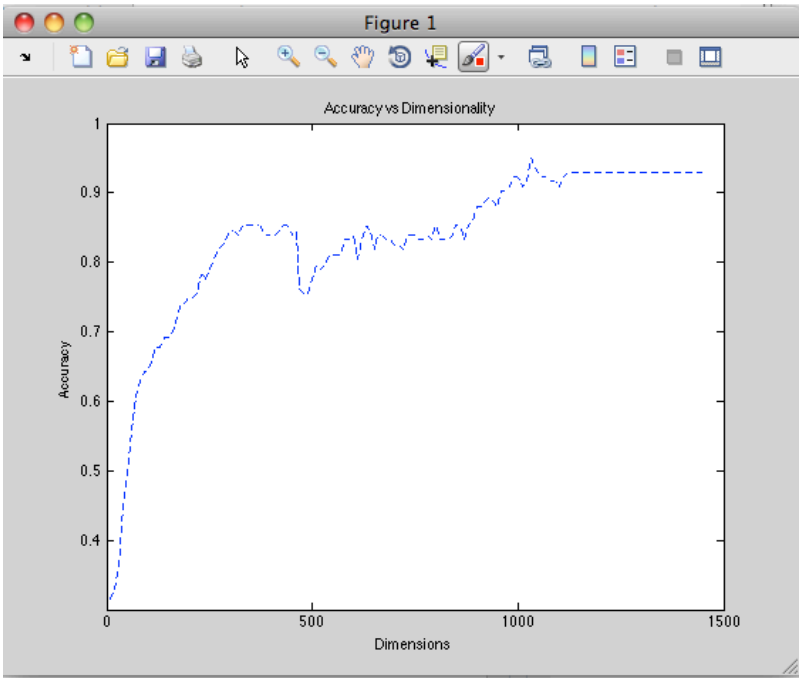

Figure 4: *Dimensions to which the data is reduced, vs the Accuracy of the FLD classifier.*

In addition to these, I have learned that the most common classifier used on handwritten digits is a 2-layer Neural Network, and I expect to implement that as well.

Where I am vs. Where I wanted to be:

In some ways, I am ahead of my goal, since I have in fact narrowed down which classifier I intend to use and have managed to obtain a fully functional data set. Ideally, I would have liked to also finish implementing the Neural Network before the milestone. However, I am glad to be in a position where I do not need to worry about finding a data set anymore.