# AMATH 582/482 Rahman

Coding Project 4

Due Friday March 10th at 11:59pm

## Teaching a Computer to Recognize Written Numbers

We're going to take a subset of Yann LeCun's famous dataset, which I preprocessed to reduce the amount of busy work. I saved the training datasets as `CP4_training_images.mat` and `CP4_training_labels.mat`, and the test datasets as `CP4_test_images.mat` and `CP4_test_labels.mat`.

We will write a few different learning algorithms; some from scratch and some using native MATLAB functions (or associated Python libraries).

- Reshape the images into column vectors (like the datasets we used in the lecture codes), and take the SVD of the images (like we did in the lecture codes).
- Observe how the singular values decrease. What is a good rank to reconstruct the images? I'll put more guidance in the code template.
- Play around with the projections: project the data onto three right singular vectors (columns of $V$).
- Pick two digits (any two). Use Linear Discriminant Analysis (just like what we did with dogs and cats) to classify them.
- Now do this for three digits.
- Which two digits are hardest to separate using this method? (i.e., which two digits have the highest percentage of data on the wrong side of the threshold)
- Which two digits are the easiest to separate?
- Now let's use the built-in MATLAB functions for support vector machines (`fitcsvm()`) and decision tree (`fitctree()`) to separate the digits (I'll provide more details in the template, but feel free to play around with these two functions, which have very detailed documentation on the MATLAB website).
- How do the three methods compare?