1. Pixel clustering. Suppose you want to represent a color image as accurately as possible but have a limit on the number of colors you can display. If you have $k$ colors available you could cluster the $(r,g,b)$ values of the pixels in the image into $k$ clusters, and then replace each pixel value by the mean of the cluster it is assigned to. Implement this and experiment with various values of $k$.

2. Image clustering. Cluster the MNIST dataset using pixel intensities as features. Observe if images contained in the same cluster represent the same digits. More formally, partition the dataset into $k$ clusters (experiment with $k = 10$ and $k = 20$) and determine the fraction of the instances in each cluster that belong to the majority class in that cluster.

3. Image clustering. Cluster the CIFAR-10 dataset using color histograms as features. As in the previous question, observe if images contained in the same cluster represent objects of the same class.

4. Image clustering. Cluster the CIFAR-10 dataset histograms of gradients as features. As in the previous question, observe if images contained in the same cluster represent objects of the same class.

5. Image classification. Download the scikit machine learning toolbox and classify the MNIST dataset using the algorithm and feature set of your choice. Make sure you don’t use the same examples for training and testing as this will give you overoptimistic results. Extra credit points will be given to the best results obtained in the class. For reference, state of the art error rates for this dataset are around 1%.

6. Image classification. Classify the CIFAR-10 dataset using the algorithm and feature set of your choice. Make sure you don’t use the same examples for training and testing. Extra credit points will be given to the best results obtained in the class. For reference, state of the art error rates for this dataset are around 5%.