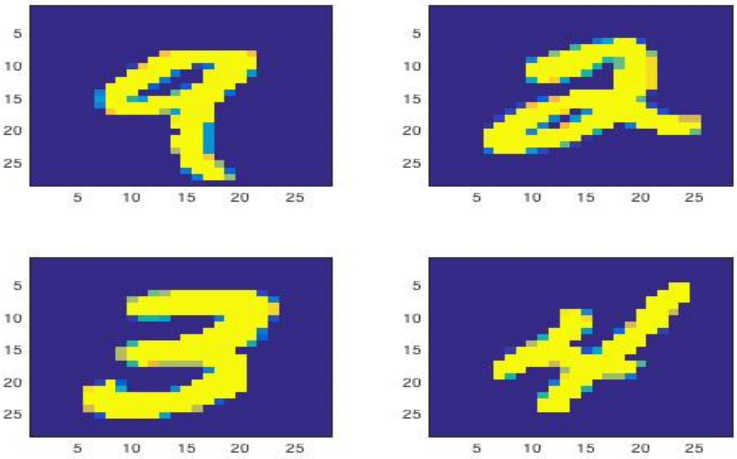
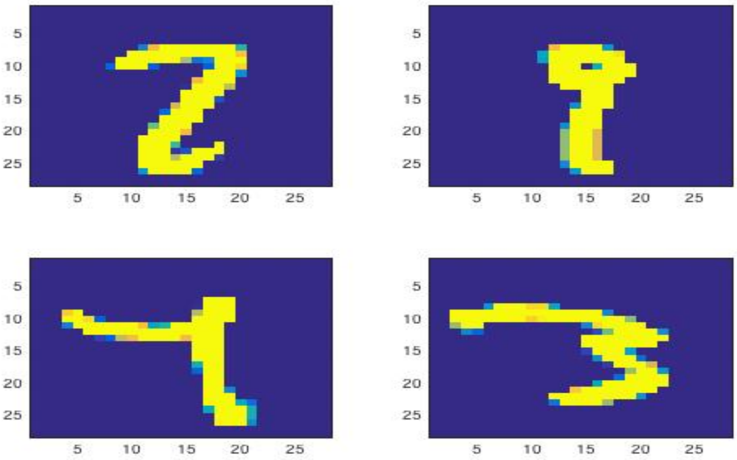
2 Classification of handwritten numbers 0-9

NIST is the USA variant of the Norges Forskningsråd (NFR). NIST has designed a database, called MNIST (see yann.lecun.com/exdb/mnist), with pixtures of handwritten numbers 0-9. The pixtures have dimension 28x28 pixels and are in 8-bit greyscale; i.e. pixel values between 0-255. For practical purpose one should note that the the pixtures have been ”preprocessed”; i.e. centred and scaled to prepare them for classification. Figure 3 shows four ”easy” examples, while figure 4 shows examples which are harder to classify correctly. A large amount of classifiers have been designed for this case, resulting in error rates between $1 - 10$ %. The state-of-the-art is (of course) a deep neural network (DNN).
The database consists of 60000 training examples written by 250 different persons and 10000 test examples written by 250 other persons.



Figur 3: ”Easy” examples of the numbers 9, 2, 3 og 4



Figur 4: ”Dubious” examples of the numbers 2, 9, 7 og 3. Or the numbers 7, 1, 4 og 7 ?

The task consists of two parts both using variants of a nearest neighbourhood classifier.

1. In the first part part the **whole** training set shall be used as templates.
   (a) Design a NN-based classifiser using the Euclidian distance. Find the confusion matrix and the error rate for the test set. The data sets should preferably be split up into chunks of images (for example 1000) in order to a) avoid too big distance matrixes b) avoid using excessive time (as when classifying a single image at a time)
   (b) Plot some of the misclassifed pixtures. Some useful Matlab commands for this are :
   - **x = zeros(28,28); x(:)= testv(i,:);** will convert the pixture vector (number i) to a 28x28 matrix
   - **image(x)** will plot the matrix x
   - **dist(template,test)** will calculate the Euclidian distance between a set of templates and a set of testvectors, both in matrix form.

   (c) Also plot some correctly classified pixtures. Do you as a human disagree with the classifier for some of the correct/incorrect plots?
2. In the second part you shall use clustering to produce a small(er) set of templates for each class. The Matlab function $[\mathbf{idx_i}, \mathbf{C_i}] = \mathbf{kmeans}(\mathbf{trainv_i}, \mathbf{M})$; will cluster training vectors from class ω_i into M templates given by the matrix C_i.
   (a) Perform clustering of the 6000 training vectors for each class into $M = 64$ clusters.
   (b) Find the confusion matrix and the error rate for the NN classifier using these $M = 64$ templates pr class. Comment on the processing time and the performance relatively to using all training vectors as templates.
   (c) Now design a KNN classifier with K=7. Find the confusion matrix and the error rate and compare to the two other systems.