Pattern Recognition Example: Handwritten Digit Classification

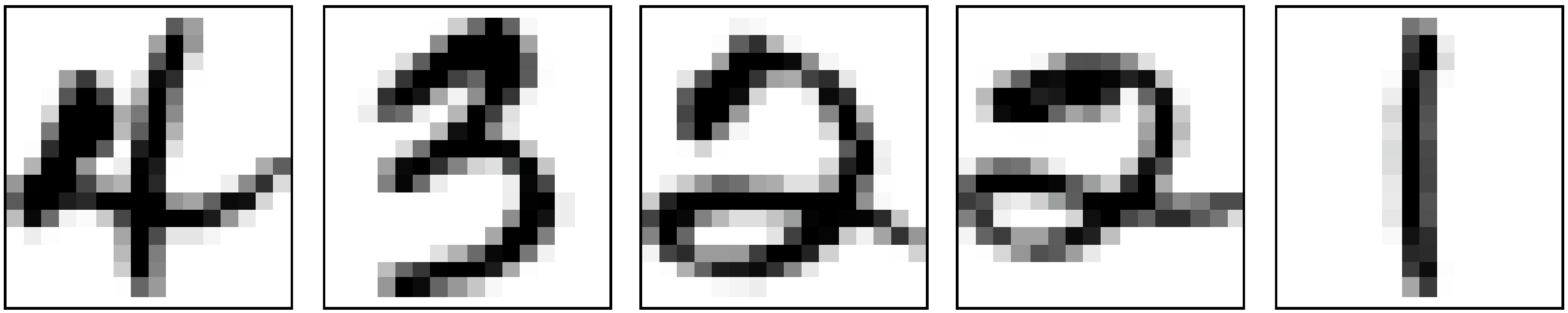
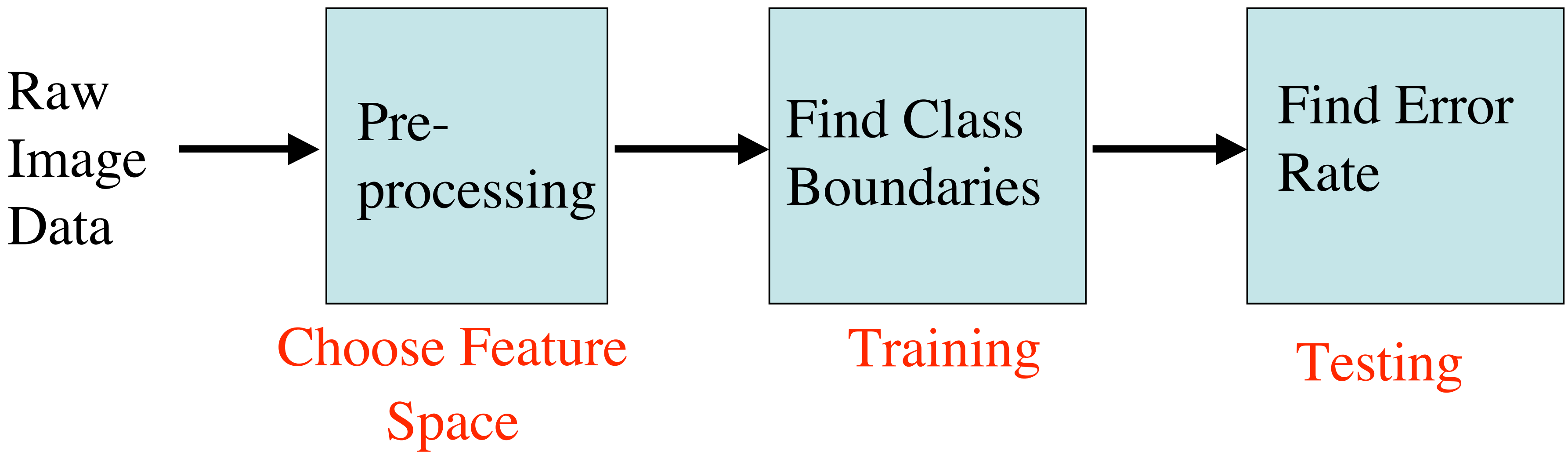


Figure 1: *Segmented handwritten zip codes. Each digit is represented by a* 16×16 *greyscale image, after size and orientation normalization.*

Flow Diagram



Almost every Patt. Rec. Method has been thrown at this problem.

METHOD	TEST ERROR RATE (%)
linear classifier (1-layer NN)	12.0
linear classifier (1-layer NN) [deskewing]	8.4
pairwise linear classifier	7.6
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
40 PCA + quadratic classifier	3.3
1000 RBF + linear classifier	3.6
K-NN, Tangent Distance, 16x16	1.1
SVM deg 4 polynomial	1.1
Reduced Set SVM deg 5 polynomial	1.0
Virtual SVM deg 9 poly [distortions]	0.8
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [distortions]	3.6

http://yann.lecun.com/exdb/mnist/

2-layer NN, 300 HU, [deskewing]	1.6
2-layer NN, 1000 hidden units	4.5
2-layer NN, 1000 HU, [distortions]	3.8
3-layer NN, 300+100 hidden units	3.05
3-layer NN, 300+100 HU [distortions]	2.5
3-layer NN, 500+150 hidden units	2.95
3-layer NN, 500+150 HU [distortions]	2.45
LeNet-1 [with 16x16 input]	1.7
LeNet-4	1.1
LeNet-4 with K-NN instead of last layer	1.1
LeNet-4 with local learning instead of ll	1.1
LeNet-5, [no distortions]	0.95
LeNet-5, [huge distortions]	0.85
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7
K-NN, shape context matching	0.67

Feature Representations

LeCun, 1998
Higher dimensional linear feature space
(convolution with many linear features)

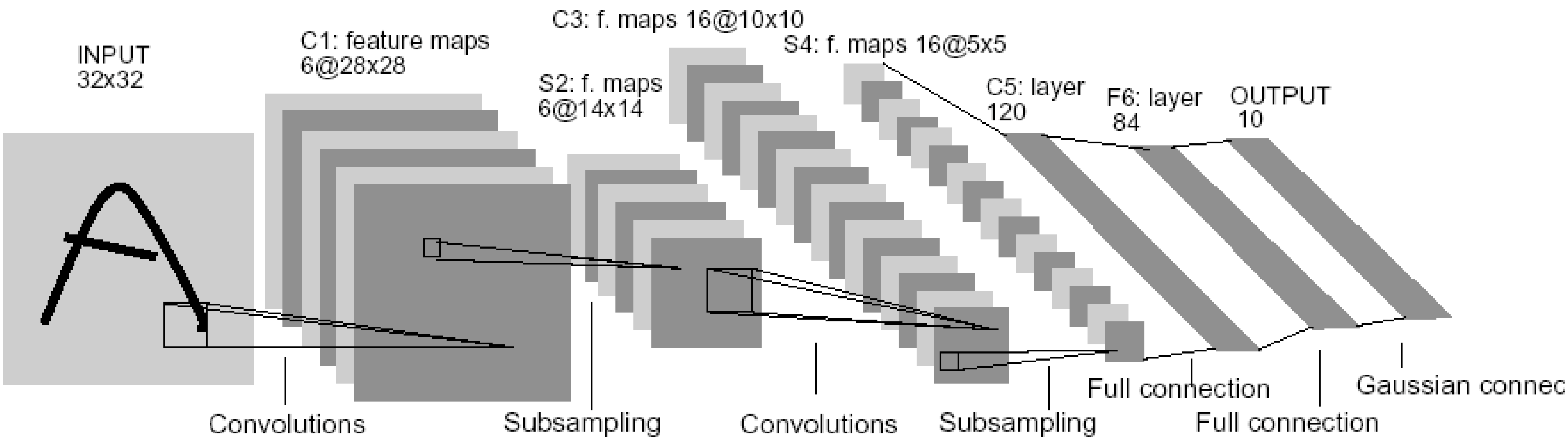


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of whose weights are constrained to be identical.

Feature Representations

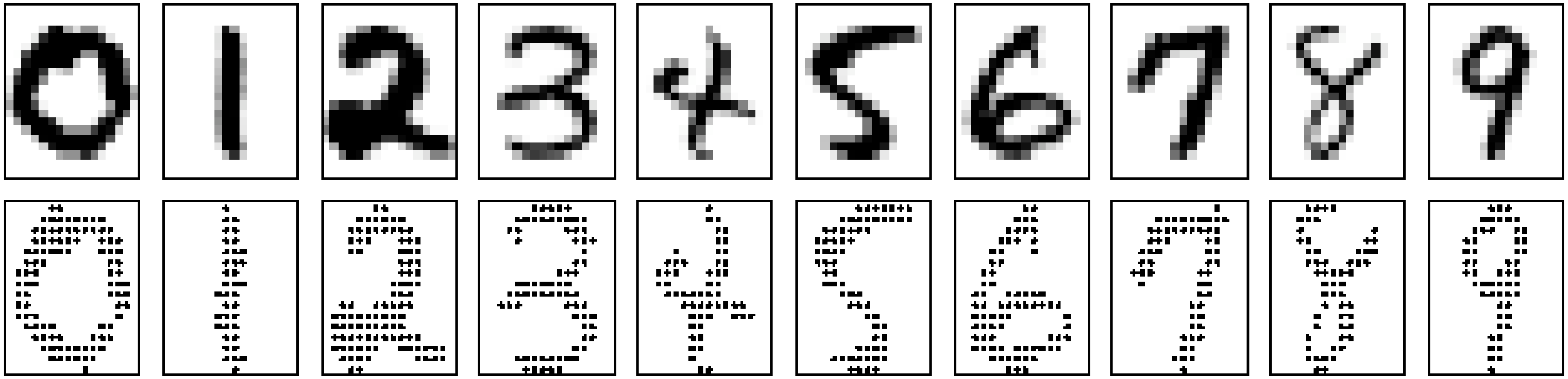


Figure 2: *A random selection of handwritten digits from the Buffalo database. The top row shows the* 16×16 *greyscale representations of the digits, and the bottom row shows the thresh-holded images represented as a point set.*

Thinning

Prototypes

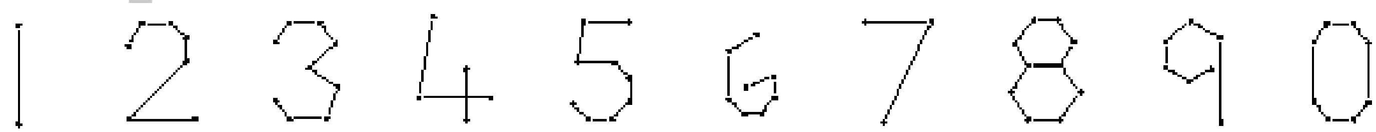


Figure 3: *Prototypes for the digits 0–9*

Hastie, Tibshirani, 1994

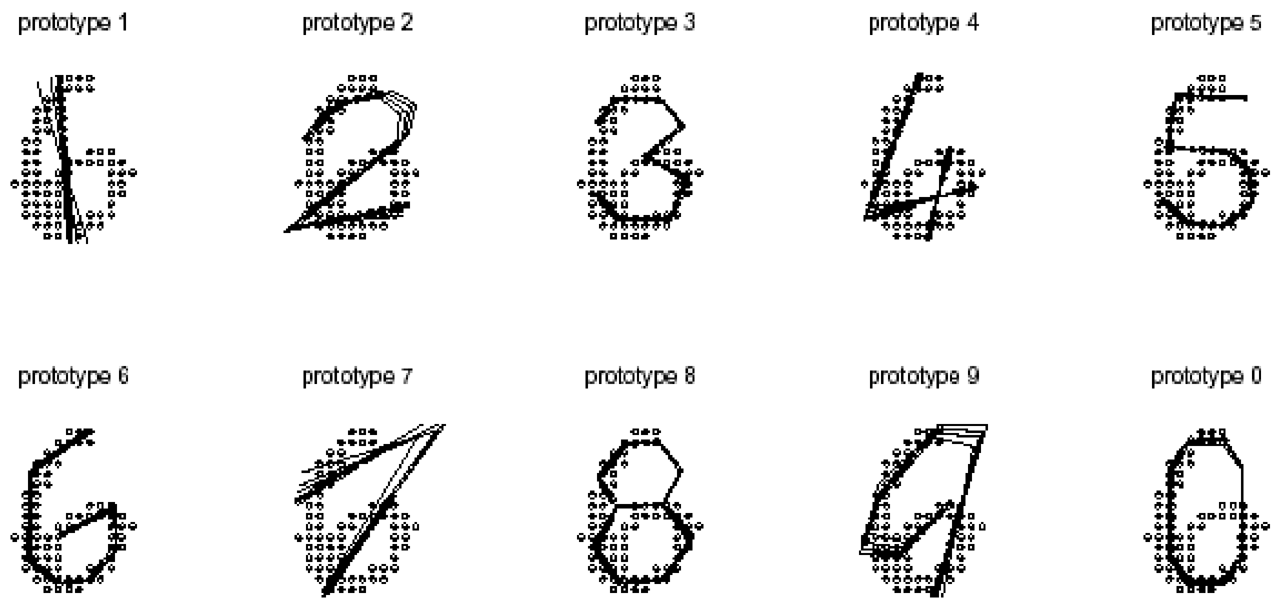


Figure 4: *Ten prototypes fitted to an image chosen at random from the training set. The different iterations are superimposed.*

Next do prototype fitting

What are the features? Fit statistics

Save a series of goodness of fit measures for each prototype to the digit. Use a vector of fit statistics to classify.

Building Invariance into feature spaces

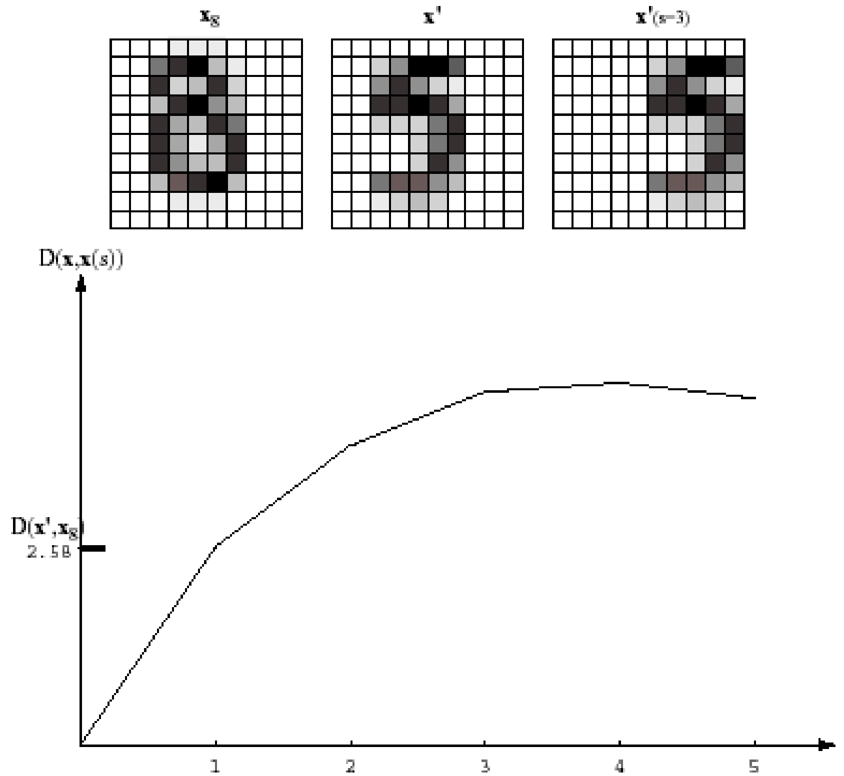


Figure 4.20: The uncritical use of Euclidean metric cannot address the problem of translation invariance. Pattern $\mathbf{x}'$ represents a handwritten 5, and $\mathbf{x}'(s=3)$ the same shape but shifted three pixels to the right. The Euclidean distance $D(\mathbf{x}', \mathbf{x}'(s=3))$ is much larger than $D(\mathbf{x}', \mathbf{x}_8)$, where $\mathbf{x}_8$ represents the handwritten 8. Nearest-neighbor classification based on the Euclidean distance in this way leads to very large errors. Instead, we seek a distance measure that would be insensitive to such translations, or indeed other known invariances, such as scale or rotation.

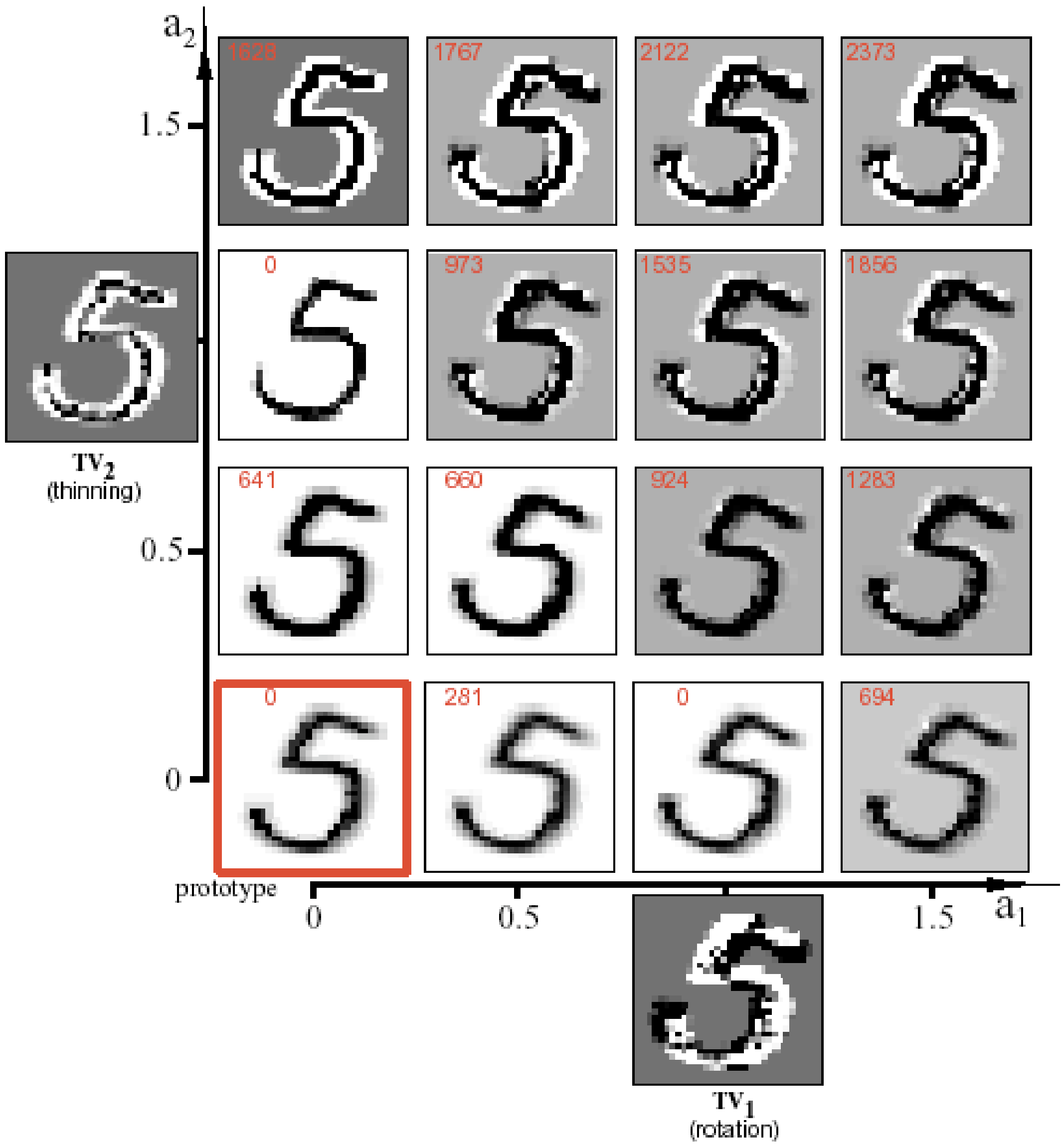
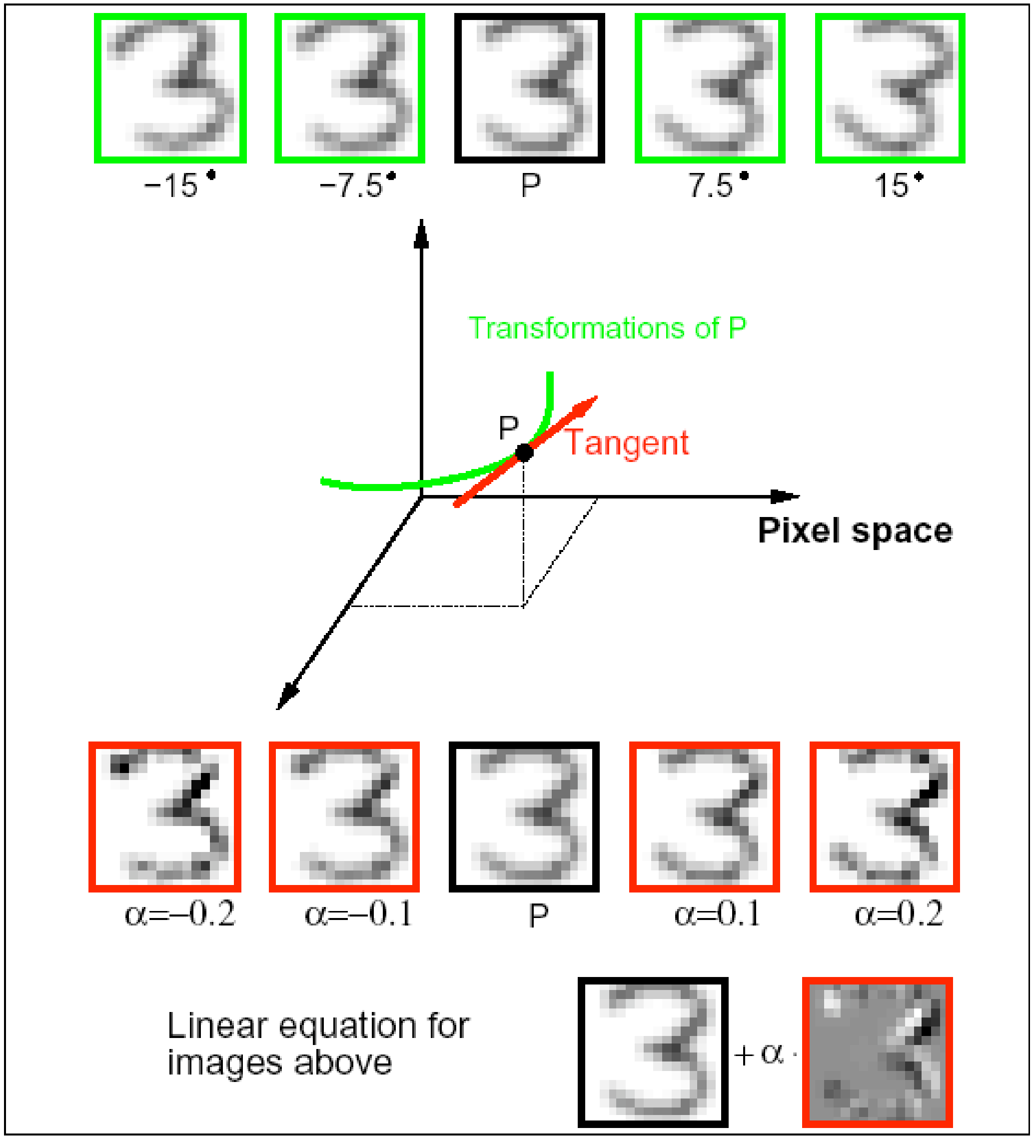


Figure 4.21: The pixel image of the handwritten 5 prototype at the lower left was subjected to two transformations, rotation, and line thinning, to obtain the tangent vectors $\mathbf{TV}_1$ and $\mathbf{TV}_2$; images corresponding to these tangent vectors are shown outside the axes. Each of the 16 images within the axes represents the prototype plus linear combination of the two tangent vectors with coefficients a_1 and a_2. The small

Other Feature Spaces



Hastie & Simard, 1997

In the tangent distance approach, prototypes are transformed (e.g. rotated, translated, thinned, etc.)

A linear approximation to that transformation is produced. That linear transformation is an axis in the feature space. New prototypes can be constructed by adding increments to the prototype vector in the transformation direction.

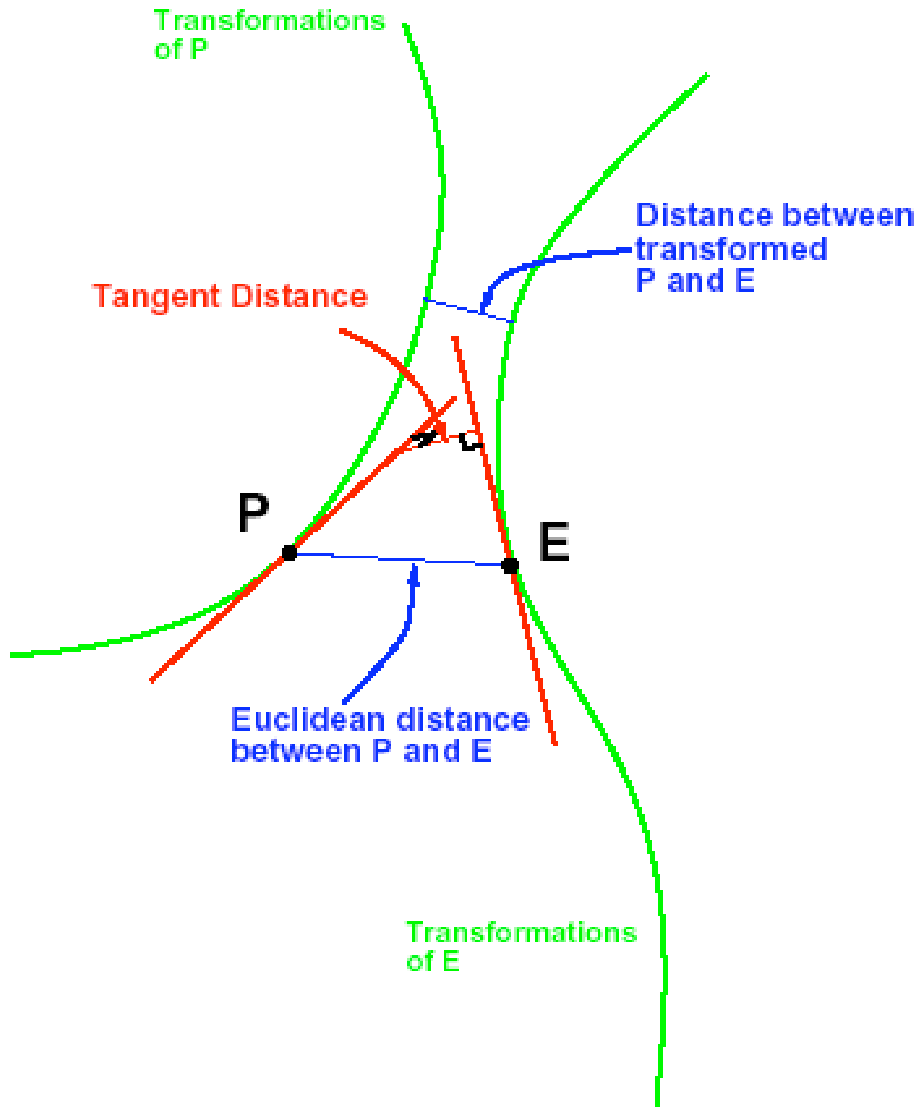


Figure 4: *Associated with each image is a manifold of dimension 7 corresponding to the the seven transformations such as rotation, scaling, etc. In principal we would like to compute the shortest distance between the manifolds of two images. Tangent distance approximates these manifolds by their tangent hyperplanes, which simplifies the distance calculations dramatically.*

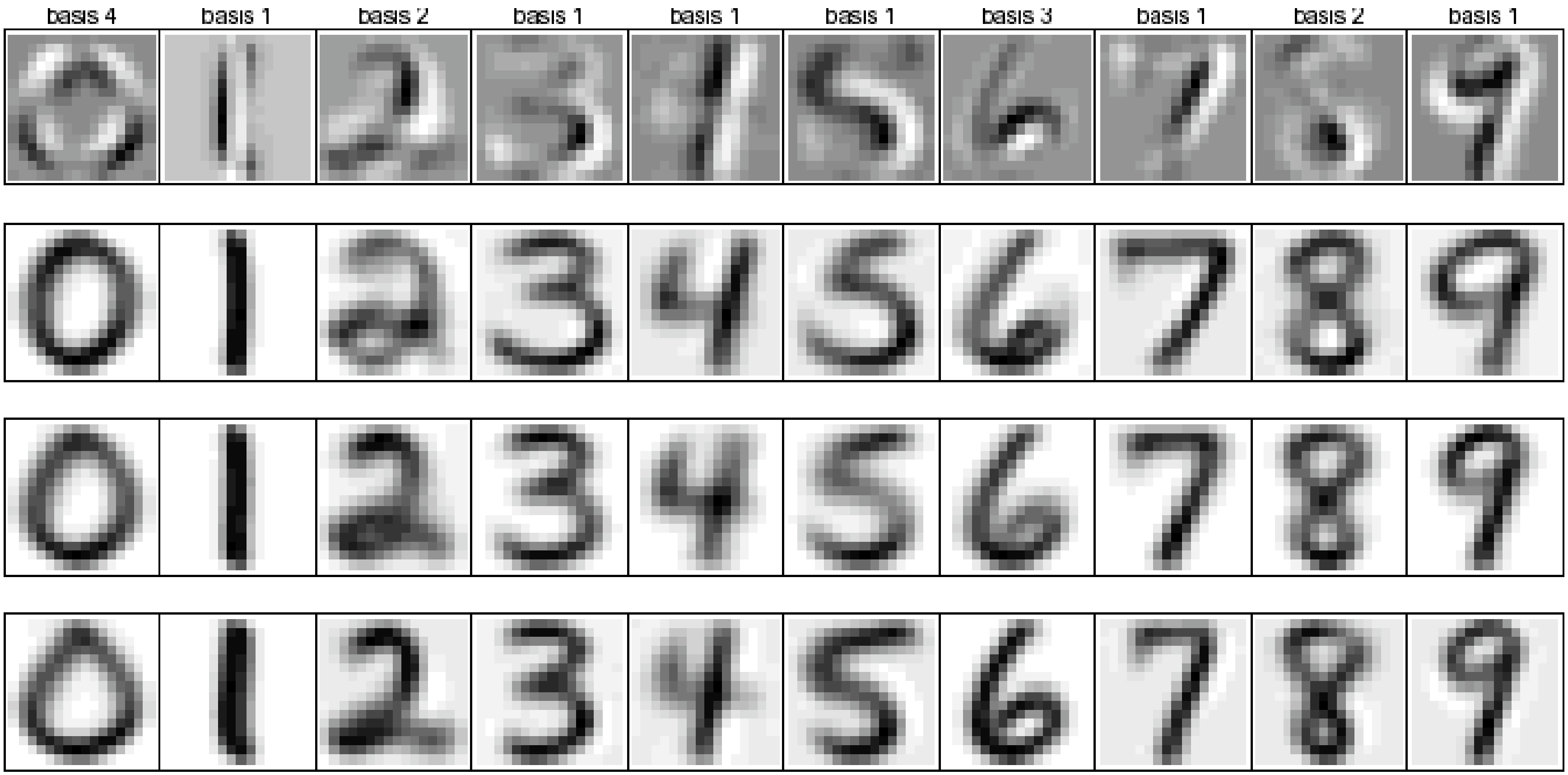


Figure 6: *Each column corresponds to a particular tangent subspace basis vector for the given digit. The top image is the basis vector itself, and the remaining 3 images correspond to the* 0.1, 0.5 *and* 0.9 *quantiles for the projection indices for the training data for that basis vector, showing a range of image models for that basis, keeping all the others at* 0.