Published in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000

Learning from One Example Through Shared Densities on Transforms

Erik G. Miller and Nicholas E. Matsakis and Paul A. Viola
Massachusetts Institute of Technology Artificial Intelligence Laboratory
Cambridge MA 02139
{emiller, matsakis, viola}@ai.mit.edu

Abstract

We define a process called congealing *in which elements of a dataset (images) are brought into correspondence with each other jointly, producing a data-defined model. It is based upon minimizing the summed component-wise (pixel-wise) entropies over a continuous set of transforms on the data. One of the biproducts of this minimization is a set of transforms, one associated with each original training sample. We then demonstrate a procedure for effectively bringing test data into correspondence with the data-defined model produced in the congealing process.*

Subsequently, we develop a probability density over the set of transforms that arose from the congealing process. We suggest that this density over transforms may be shared by many classes, and demonstrate how using this density as "prior knowledge" can be used to develop a classifier based on only a single training example for each class.

1 Introduction

There has been such great progress in the development of classifiers for problems such as handwritten digit recognition that the problem is for many researchers considered to be solved. Classifiers such as LeCun et al.'s convolutional networks [9, 12] achieve performance very close to that of human test subjects, a commonly assumed benchmark of "optimal" performance. It seems that performance cannot get much better. However, these methods require large training sets. For example, LeCun et al. use 6,000 samples of each character in training. This leaves open the question of whether these methods are appropriate when large amounts of training data are not available.

If we examine the performance of classifiers using a small amount of data, in the limit, one example per class, there still seems to be a large gap between the capabilities of machines and humans. Consider the symbol for the new European currency, the "Euro", shown in Figure 1. After seeing a single example of such a character, humans can recognize the character in a wide variety of contexts, styles, and positions.

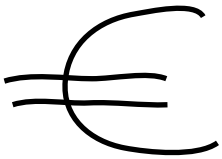

Figure 1. The new symbol for the standard European currency. After seeing a single example, people typically have no difficulty recognizing the symbol in a wide variety of styles and variations.

Clearly, this is due at least in part to generalization based on previously learned classes. That is, our knowledge about handwriting in general allows us to bring prior knowledge to the formation of our model for a new character based on a single example. A long-standing question in computer vision, and in AI in general, is what form this prior knowledge takes.

In this paper, we consider the value of knowledge about common deformations of images of objects. We show that such knowledge, in the form of a probability density on a space of deformations, can be used to greatly improve the behavior of a classifier on the handwritten digit recognition problem. Specifically, we discuss the performance of three classifiers. The first classifier is a variation on nearest neighbor using the Hausdorff distance between images [8]. It measures a distance directly to the original training samples. We use this classifier because it can be applied in very sparse data settings, i.e. with only a single training example. This classifier implicitly determines a probability measure on images for each class c_j, that is

$$P(I|c_j), \tag{1}$$

the probability of the observed image given the class.

For the second classifier, we give algorithms for finding transforms that place the training and test images in corre-

spondence with each other before doing the Hausdorff distance comparison. Here, our probability measure is over the corresponding, or latent, images. That is, we find

$$P(I_L|T,c_j), \tag{2}$$

the probability of the *latent* or "pre-transform" image, given a transform and a class. This step is closely related to previous work reported in [2, 13, 5] and will be discussed below.

In the third classifier, we make a substantial improvement by assigning a cost to the transformation which puts a test character into correspondence with a particular training set. This can be viewed as evaluating

$$P(T)P(I_L|T,c_j), \tag{3}$$

the probability of the correspondence transform *and* the latent image.

We then demonstrate how our techniques can be used to build a classifier with only a single training example from each *digit* class, by borrowing the first factor in Eq.(3) from another *non-digit* class. This classifier's performance (89.3%) is dramatically better than the basic Hausdorff distance classifier (29.7%) (one of the few that can be applied in such an extreme sparse data situation). We believe that the single sample learning problem is a good place to focus efforts on closing the gap between human-machine performance, given that the performance of classifiers based on large amounts of data is plateauing.

2 The Image Model

We adopt an image model similar to those adopted in papers on deformable templates as in [2]. Other researchers have recently had successes in using similar models as in [5] and [13].

The basic idea is that the image be understood as *texture* and *shape* as described in [13], or as a *latent image* that has undergone a *transformation*, as presented in [5]. We denote this as

$$I = T(I_L) \tag{4}$$

where I is an observed image, I_L is a latent image, and $T(\cdot)$ is a transformation which warps one image into another. If T is invertible, the latent image can be computed from the input image as

$$I_L = T^{-1}(I). \tag{5}$$

In the most general setting, $T(\cdot)$ could be any map from the coordinates of one image (values in $\mathcal{R}^2$) to the coordinates in the new image. For our development, we will assume that the transforms are affine, although we discuss extensions to arbitrary diffeomorphisms (smooth one-to-one mappings) below. Affine transforms may be represented as matrix operations as well as functions, so that

$$T^* = T_1 T_2 \tag{6}$$

represents both the matrix product of T_1 and T_2 as well as the composition of the two warps.

When the so-called shape of an image relative to a reference image or model has been eliminated, then the images are said to be in correspondence. For example, by deshearing the image of an italic character, we can put it in better correpondence with the canonical non-italicized version of the character.

3 Congealing for model formation

Vetter et al. coined the term *vectorization* for the process of aligning (bringing into correspondence) an image with a reference image or a model. We introduce the term *congealing* which we define as the simultaneous vectorization of each of a set of images to each other. This is similar in spirit to Vetter et al.'s bootstrap vectorization procedure [13] and Frey et al.'s Transformed Mixtures of Gaussians [5], although it differs in several details, discussed below.

Given a set of training images for a particular class, the process of congealing transforms each image, through an iterative process, so as to minimize the joint pixelwise entropies E of the resulting images. We define this joint entropy to be

$$E = \sum_{p=1}^{P} H(v(p)), \tag{7}$$

where $v(p)$ is the binary random variable defined by the values of a particular pixel p across all of the images, $H(\cdot)$ is the discrete entropy function of that variable [3], and $1 \leq p \leq P$ is the set of pixel indices for the image. For example, if pixel p^*, whose coordinates are $(7,13)$, is white in half of the images and black in the other half, then $v(p^*)$ is a Bernoulli random variable with mean 0.5, and $H(v(p^*)) = 1$ bit.

We treat the $v(p)$ as independent random variables in Eq.(7). This means that our entropy estimate from Eq.(7) is an upper bound on the true entropy of the image distribution. By minimizing this upper bound we hope to minimize the true entropy.

3.1 The congealing algorithm

The basic algorithm (Algorithm 1) proceeds as follows. It makes small adjustments in the affine transforms applied to each image to reduce the pixelwise entropy defined in Eq. (7). More formally:

1. We will maintain a transform matrix U_i for each image. Set all of these to the identity initially.

2. Compute the summed pixel-wise entropies E for the current set of images.

3. Repeat until convergence:

   (a) For each image I_i,

      i. For each affine parameter (rotation, x-shear, y-shear, x-translation, y-translation, x-scale, y-scale).

         A. Let A be the 2-D affine transform based on a nominal value of the current affine parameter. For example,

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

         represents a matrix that translates an image one pixel in the x direction.

         B. Let $U_i^{new} = AU_i$.

         C. Apply U_i^{new} to image I_i and recompute the pixel-wise entropy E.

         D. If E has been reduced, accept U_i^{new}, otherwise:

         E. Let $U_i^{new} = A^{-1}U_i$ and apply U_i^{new} to the image I_i. If E has been reduced, accept U_i^{new}, otherwise revert to U_i.

   (b) After each cycle through the image set, readjust the scale of all of the transforms U_i *by the same amount* so that the mean log-determinant of the transforms is 0 (discussed below).

   (c) After each cycle through the image set, store the current mean image for later use with test character congealing. We shall denote the kth mean image for class j as M_k^j.

4. At this point, each of the U_i is the transform such that

$$U_i(I) = I_{L_i}. \tag{8}$$

   That is, U_i is our best guess of the transformation which maps the observed data into the latent image. Hence, we can identify the final transforms from congealing as samples of T^{-1} from Eq.(5). Alternatively,

$$T_i = U_i^{-1}. \tag{9}$$

   These "samples" of T will be used in Section 6 to form a density on transforms.

The scale readjustment step (b) protects against the images all shrinking to improve the cost function. While the algorithm allows some images to shrink and others to grow, it forces the mean scale to be the same as in the original images.

The reader may notice that the number of affine parameters given in Algorithm 1 (7) is actually greater than the number of degrees of freedom in a two-dimensional affine transform (6). That is, the set of parameters overspecifies the transform. This is irrelevant for a coordinate-descent procedure, however. In fact, the overcomplete parameterization works significantly faster in practice than doing coordinate descent on the individual elements of the transform matrix, presumably because these parameters tend to point in directions closer to the transform gradient than the naive parameterization.

We demonstrate the properties of this algorithm on several real and artificial data sets. Figures 2(a) and 2(b) show samples from the NIST character database. The data vary in scale, position, rotation, shear, and position to a limited extent. Figures 2(c) and 2(d) show the congealed versions of these data sets, i.e. their positions after optimal alignment.

Figure 3 illustrates how the pixelwise entropies of the adjusted image sets change as the algorithm progresses. Figures 3(a) and (b) demonstrate that before the congealing process has begun, the pixelwise entropies are relatively high, and the high coherence of (c) and (d) show the improvement in pixelwise entropies due to the congealing pro-

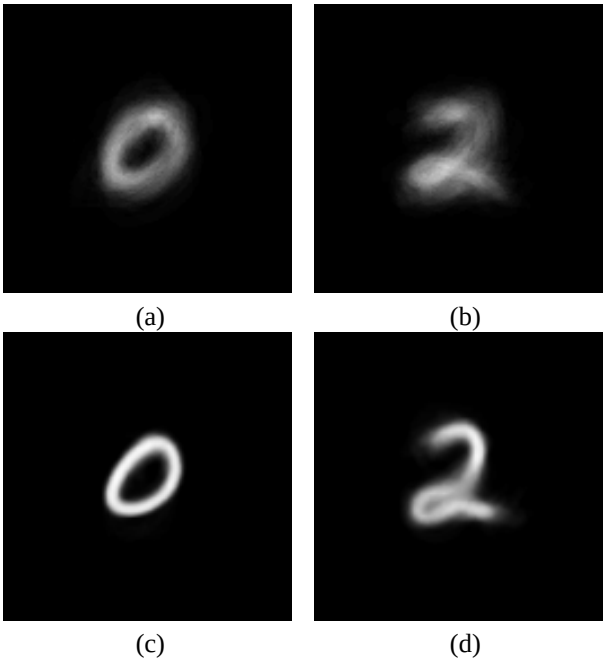

Figure 3. Mean images during the congealing process. (a) The initial mean image for the set of zeroes. The blurriness is due to the misalignment of the images. (b) The initial mean image for the set of twos. (c) The final mean image for the set of zeroes. The coherence of the aligned images is indicated by the increased sharpness of the image.(d) The final mean image for the set of twos.

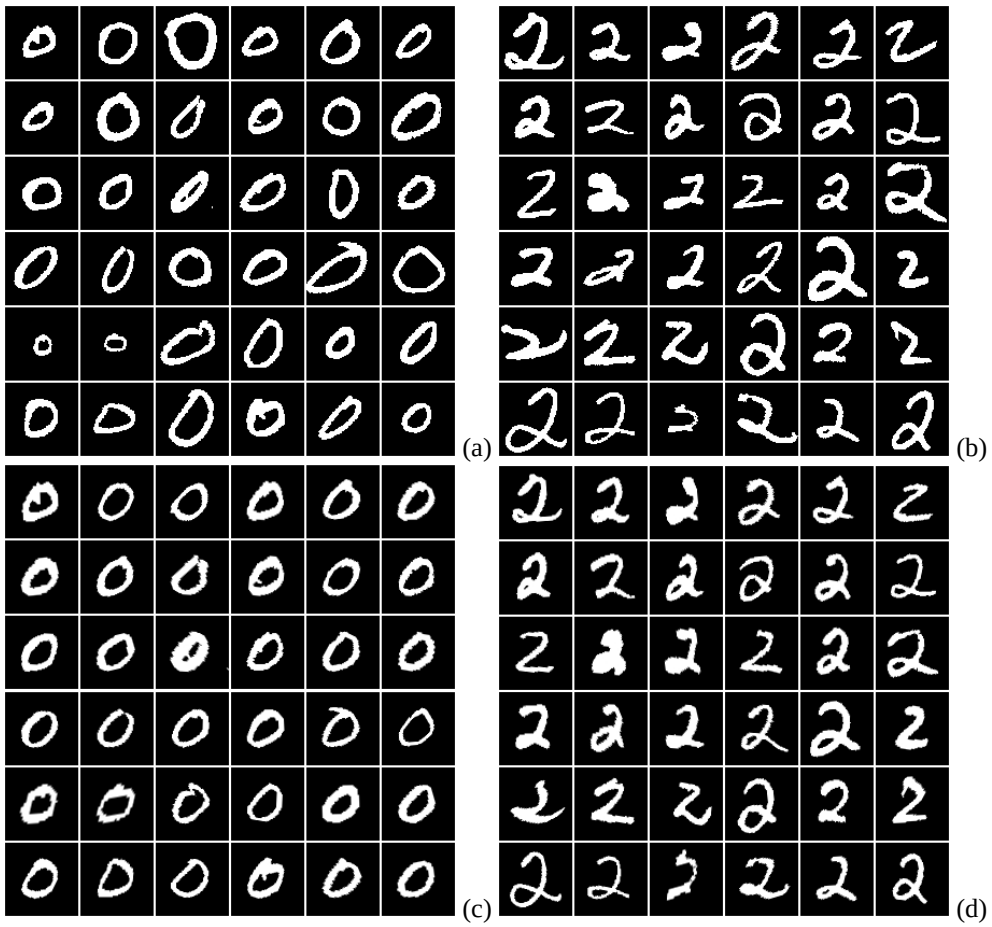

Figure 2. (a) Samples of zeroes from the NIST database. (b) Samples of twos from the NIST database. (c) The zeroes after congealing. (d) The twos after congealing.

cess. Notice that there is more entropy in the final mean "two" image than in the "zero" image due to the fact that zeroes can be better aligned through affine transforms than twos can.

3.2 Avoiding Local Minima

The congealing process has a serendipitous advantage in that it often circumvents two types of minimization problems, the so-called "zero-gradient" problem and the local minima problem. Because the alignment process is done over an ensemble of images which has a data-dependent smoothing effect, these two issues arise infrequently.

An example of the zero-gradient problem is shown in Figure 4. Note that the grey "X" and the white "X" do not overlap at all, despite the fact that their centroids are aligned. Thus a differential change in relative rotation of the two characters will not improve alignment according to our cost function. There is a local minimum problem here as well. It arises when one of the legs of the "X" overlaps, while the other does not.

A common solution to these problems is to blur the images before trying to align them, for example by convolving them with Gaussian kernels. This sometimes works well for the zero-gradient problem, but throws away high frequency information in the image which may be helpful for alignment. And it frequently cannot solve the local minimum problem at all.

The congealing method addresses both of these problems without any ad hoc blurring. Since the method aligns each example to the statistical model of the ensemble, the natural variability in the data causes smoothing in exactly the parts of the image where the images vary. In other words, it creates a kind of data-specific smoothing as seen in Figure 3. Given a reasonably large number of examples, this smoothing addresses both the "zero gradient" and local minima problems.

There are, however, situations in which our method fails

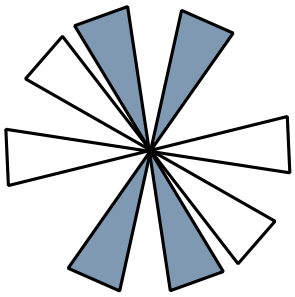

Figure 4. An example of two images that will not congeal to the global minimum of the correspondence fitness function. This problem can usually be alleviated in practice by congealing a large number of samples simultaneously.

(a) (b)

Figure 5. Two distinct centers of convergence for a set of rotated "4" images. The algorithm aligned the horizontal part of some fours with the vertical part of others and got stuck in this local minimum. However, since any test character which happens to be a four should rotate to one of these two positions, this can still make a good model for classification.

to converge to the globally optimum alignment for a set of characters. We conducted the following experiment to examine this issue more systematically on realistic data. Starting with a single image of a "4" from the NIST database, we generated a sequence of 100 images rotated at uniform intervals from $-\frac{\theta}{2}$ to $\frac{\theta}{2}$. For $\theta < 68$ degrees, the images congealed to a unique corresponding position, but when $\theta > 68$ degrees, two correspondence "centers" emerged. This is due to a local minimum in the congealing process, as illustrated in Figure 5. Although this lack of convergence to a single global "center" is not ideal, it does not preclude us from using the resulting density model, which has relatively low entropy. That is, even in the presence of multiple convergent "centers" we are performing an important dimensionality reduction in the data by congealing. The key property is that a test character will be congealed to a predictable location for comparison with the model (discussed below).

4 Using the Models for Classification

To perform classification, we wish to estimate the model with the maximum posterior probability give the observed image I. Assuming a uniform prior over the classes c_j, we have using Bayes' rule:

$$\arg\max_j P(c_j|I) = \arg\max_j P(I|c_j). \tag{10}$$

As in [5], we then introduce the transformation variable T and integrate over it:

$$\begin{aligned} & \arg\max_j P(I|c_j) && (11)\\ = \; & \arg\max_j \int_{T\in\mathcal{T}} P(T,I|c_j)\,dT && (12)\\ = \; & \arg\max_j \int_{T\in\mathcal{T}} P(T|c_j)P(I|T,c_j)\,dT && (13)\\ = \; & \arg\max_j \int_{T\in\mathcal{T}} P(T|c_j)P(I_L|T,c_j)\,dT && (14) \end{aligned}$$

Eq.(14) follows since the latent image I_L corresponding to the image I is a deterministic function of I, given the transform T.

We now make the key simplifying assumption that, with high probability,

$$\begin{aligned} & \arg\max_j \int_{T\in\mathcal{T}} P(T|c_j)P(I_L|T,c_j)\,dT && (15)\\ = \; & \arg\max_j \max_{T\in\mathcal{T}} P(T|c_j)P(I_L|T,c_j). && (16) \end{aligned}$$

This is an approximation to the full Bayesian treatment of the problem, and assumes that the joint distribution $P(T,I_L|c_j)$ peaks sharply around the maximum. This allows us to avoid the integral in Eq.(15) and search a full space of possible transforms, rather than a discrete set as some authors have done [5].

As an additional, but optional simplification, we assume that the transform probability is independent of the class, and so the final maximization expression becomes

$$\arg\max_j \max_{T\in\mathcal{T}} P(T)P(I_L(T,I)|T,c_j). \tag{17}$$

In the last equation, we have shown I_L as a function of T and I only to remind the reader that the latent image is defined by the observed image I and the current value of the transform T.

4.1 Finding the best transform of a test image

An additional challenge when dealing with an infinite number of allowable transforms is to find the maximum over T in Eq.(17). When dealing with only a finite number of transforms, this becomes just an iterative computation, linear in the number of allowed transformations. Obviously this is not possible in dealing with a continuously parameterized space of transforms.

One possibility would be to perform gradient descent on Eq.(17) in the space of affine parameters of the transforms. The problem with this is that this function may have local minima that cannot be circumvented easily, especially if the congealing process produced a low entropy model, as in Figures 3(c) and (d). That is, we may end up with local minima or zero-gradient problems as discussed in the previous section.

As stated above, elements of the training ensemble rarely get stuck during the congealing process. In order to improve convergence of a test example, it can be *inserted into the training ensemble*. Congealing can then be performed as before. Since the test sample was drawn from the same distribution as one of the training classes, the test sample should also successfully reach a "center" or local minimum which is on average as good as the local minimum achieved by the training data for the corresponding class.

Of course congealing all of the training data for each test character presented to the system would represent a great deal of redundant computation. Two insights allow for a significant reduction in computation. First, if the number of training examples is large, the addition of an addition example should not effect the convergence of the ensemble. Second, the elements of the ensemble only interact through the probabilistic model which is estimated at each time step. As a result the convergence behavior of a new example added to the ensemble is only a function of the sequence of probabilistic models, not the behavior of the entire ensemble. Alignment of a new test example proceeds by reducing the entropy, or equivalently by maximizing the likelihood, of the example with respect to a sequence of probabilistic models, which are saved during Algorithm 1.

5 Initial Experiments

We performed experiments on NIST Special Database 19 [7]. These are 128x128 binary images of characters that have been segmented and roughly centered but are otherwise unprocessed. For a baseline, we used 1000 examples of each training class and tested on 100 examples which did not overlap with the training data. For the following experiments, we implemented a symmetric Hausdorff distance classifier [8]. This is essentially a template match which ignores mismatched pixels near the boundaries of each character. We used a dilation of only a single pixel around each example. The base performance of the Hausdorff classifier in this experiment was 92.5%, as shown in Table 1.

Since we have demonstrated that we can congeal both training and test data into the forms seen in Figure 2, we could also classify in the congealed data space. For this experiment, the classifier actually degraded to 87.3%. We explain this as follows. Since we allowed any affine transform rather than one from a finite set as a possible solution to the minimization of Eq.(17), there were cases in which an inappropriate transform may have been used to minimize the joint entropy of the test sample and the training data for a particular class. In particular, examining the confusion matrix for this case, it became clear that many of the digits had been transformed into very good matches to the "1" model, by getting shrunk in the x-direction. For example, by scaling an "8" down by a factor of 10 in the x-direction, we get a character which matches the "1" model very well. This explains the poor performance of this version of the classifier.

We help alleviate this problem by estimating an explicit density on transforms, as suggested by Eq.(17). By modifying our likelihood estimates with the transform density, performance was increased to 96.4%. This is precisely because we are assigning a high cost to transforms such as one described above which transforms an "8" into a "1". A similar approach (with an implicit estimate) has been taken by [10].

6 Estimating a Density on Affine Transforms

Note that the congealing process implicitly produces a large set of possible transforms for each class. (Recall, each training image is mapped to a latent image by some transform U_i, thus giving a sample of T_i, its inverse.) Thus we can develop a kernel based estimator for the class of transforms that map latent images to observed images. Let us assume our estimator has the form:

$$P(T) = \frac{1}{N}\sum_{i=1}^{N} K(T, T_i). \tag{18}$$

One simple way to put a density on the space of affine transforms would be to consider each sample T_i as a vector in a six-dimensional space and then to use a Gaussian kernel in a Parzen style estimator. For example, setting the kernel function as in:

$$K(T, T_i) = Ce^{-\|T-T_i\|^2}. \tag{19}$$

Training Samples	Basic Hausdorff	With Congealing	With Transform Density
1000	92.5%	87.3%	96.4%
1	29.7%	60.0%	89.3%

Table 1. The first row of the table shows experiments using 1000 training examples. The first column shows the results for a basic Hausdorff classifier with centroid alignment. The second column shows the same technique applied to the congealed data. The third column shows the benefit from using the density on transforms. In the second row, we show the results from a classifier using only a single example. The last column of the second row used a previous computed density on transforms.

The problem with this approach is that it does not obey the so-called Affine Group Invariance property [1]. Intuitively, this just means that the above kernel behaves very differently in different regions of the space. For example, two shrinking transforms which vary by 1 degree of rotation would be treated as being much closer together than two expanding transforms separated by 1 degree of rotation. This violates the Affine Group Invariance property. See [6] and [1] (Section 7), for more details on this.

We propose an alternative kernel, namely

$$K(T,T_i) = Ce^{-F\left(T^{-1}T_i\right)}, \tag{20}$$

which obeys the Affine Group Invariance property. When the set of affine transforms is viewed as a group, the operator $T^{-1}T_i$ can be viewed as the natural difference between T and T_i in the sense that it is the operator which maps T to T_i (when applied on the right). We currently choose $F(M)$ to be $\|M-\mathbf{1}\|^2$, where $\mathbf{1}$ is the identity matrix. In other words, subtract the identity matrix from the argument M and take the sum of the squared magnitudes of the remaining components. This particular choice of $F(\cdot)$ in our estimator is, at the moment, an arbitrary decision, and we are currently investigating the option of learning a better function to replace it with.

Once we have a density on transforms, we can improve our performance by evaluating Eq.(3) rather than Eq.(2). This takes our performance from 87.3% to 96.4%. We now examine how this method can be applied in a setting where we have learned similar tasks, but have very little data for our new task.

7 A One Sample Classifier

Assume we are addressing the handwritten *digit* recognition problem. Suppose that previously we have learned a density on transforms in a separate problem, e.g. by examining handwritten *letters*. In the following experiments, we used a set of transforms B_i derived by congealing a set of 100 images of the letter "A" taken from the NIST Special Database 19.

We then make the assumption that this transform density can be substituted into Eq.(17) in place of a digit transform density. We then need only compute the probability of the maximum likelihood latent image under the current model and we have a classifier.

The major challenge here is to bring a test sample into correspondence with the single training sample of a digit. We achieve this in the following way. Assume that we have the set of transforms B_i derived from the set of "A"s.

1. Create an artificial data set for a class j from the single training example by operating on the example with each of the B_i. Since we can borrow an arbitrarily large number of transforms from another classifier, we can make this data set as large as we want.

2. Once we have this artificial data set, we can proceed exactly as if we had a real training set.

To make the problem easier, in choosing a single example of each digit on which to build a model, we chose a sample of each digit from the NIST database that represented our estimate of a "canonical" example of that particular character. This clearly makes the problem easier than if we had randomly selected a character from the same distribution as the test data. However, we note that this method of selecting a single canonical example fits very well with the model of a teacher teaching a student via a canonical example.

The method of creating artificial data discussed above is similar to the methods used in [11] and [12] in which extra data is created by randomly sampling from affine transforms and applying those transforms to data samples already observed. The difference here is that we are using exactly those transforms for which we have evidence, rather than sampling uniformly over some ad hoc set of transforms.

The simple Hausdorff classifier, applied to the single training example case gave us 29.7%. By creating an artificial data set and doing congealing of the test sample, we improved performance to 60.0%. Finally, by including the density on transforms derived from the set of "A"s, we improved the performance to 89.3%. We emphasize that this is just a starting point, and we hope to be able to use similar

methods to improve this number significantly. Also, notice that we used the "borrowed" set of transforms both to create an artificial dataset for the congealing process *and* to evaluate the probability of the transform applied to a test image.

8 Discussion and Conclusions

We have demonstrated a method which simultaneously estimates the geometric correspondences for an ensemble of training images and can also be used to put a test image in correspondence with a model. From the original images, a factored probabilistic model can be estimated that includes a model for the latent image as well as the distribution of geometric transformations. Once we have correspondences, it is easier to develop good density estimators and classifiers, even based on extremely simple methods such as a Hausdorff Distance classifier.

All of the experiments we described have been done on binary images, but there is no reason these methods cannot also be applied to other types of data. As long as a measure of entropy for each component of a data point can be developed, as in [14] for example, the method could be applied to greyscale and color images, as well as other types of data. We recently have had promising results in on-line handwriting, allowing transformations not only in the spatial domain, but also in time.

We also note that the congealing method and probability associated with the accompanying transform can be used in conjunction with any classifier, such as support vector machines [12]. We are currently investigating the effect on performance of these classifiers in sparse data settings.

An obvious question is whether affine transforms are the appropriate set of transformations to consider in modeling the observed image as a combination of a latent image and some arbitrary "warp". [4] discusses a Fourier representation for smoothly varying vector fields. We are currently investigating whether a family of transformations based on these representations can give us better performance on the one-sample classifier problem.

Finally, we note that our congealing process can be viewed as a type of non-linear projection on a manifold. That is, we can view the congealing process as a propagation of our training data (and then test data) into a lower dimensional manifold which spans only the variability in the data that can *not* be represented by affine transforms. This raises the question of whether other desired invariances can be dealt with via this data propagation approach. For example, consider the case of shadows on objects. In most cases, we would like recognition to proceed independent of a particular shadow cast on an object. Suppose we allow ourselves to smoothly change the brightness of a greyscale image in any region bounded by an edge to make a collection of images less entropic, in the sense defined by Eq.(7). This may allow us to remove shadows from images without adding many spurious features to an image. This would be a kind of congealing in the "texture" space, rather than in the "shape" space. We are currently investigating this possibility.[1]

References

[1] S. Amari, "Natural Gradient Works Efficiently in Learning," *Neural Comp.*, 10, 1998, pp.251-276.

[2] Y. Amit, U. Grenander, M. Piccioni, "Structural Image Restoration Through Deformable Templates," *J. Amer. Stat. Assoc.* , 86 (414), Jun., 1991, pp. 376-387.

[3] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, 1991.

[4] G. E. Christensen, "Consistent Linear-Elastic Transformations for Image Matching," In A. Kuba et al. (Eds.). *Inf. Proc. in Medical Imaging, Lect. Notes in Comp. Sci. 1613*, 16th Inter. Conf., pp. 224-237, Jun.-Jul., 1999.

[5] B. J. Frey and N. Jojic, "Estimating Mixture Models of Images and Inferring Spatial Transformations Using the EM Algorithm," *Proc. IEEE CVPR*, June, 1999, pp.416-422.

[6] U. Grenander, M. I. Miller, A. Srivastava, "Hilbert-Schmidt Lower Bounds for Estimators on Matrix Lie Groups for ATR," *IEEE PAMI*, 20 (8), Aug., 1998.

[7] P. Grother, "NIST Special Database 19 Handprinted Forms and Characters Database," *Technical Report on Special Database 19*, National Institute of Standards and Technology, March 1995.

[8] D. P. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE PAMI*, 15 (9), pp. 850-863, Sept., 1993.

[9] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Comparison of learning algorithms for handwritten digit recognition," In F. Fogelman and P. Gallinari, editors, *Inter. Conf. on Artificial Neural Networks*, pp. 53-60, Paris, 1995.

[10] M. Revow, C. Williams, and G. Hinton, "Using Generative Models for Handwritten Digit Recognition," *IEEE PAMI*, 18 (6), pp. 592-606, June, 1996.

[11] P. Simard, Y. LeCun, and J. Denker, "Efficient Pattern Recognition Using a New Transformation Distance," In *NIPS V*, Morgan Kaufmann Pub., 1993, pp. 50-58.

[12] V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, 1998.

[13] T. V. Vetter, M. J. Jones, T. Poggio, "A Bootstrapping Algorithm for Learning Linear Models of Object Classes," *Proc. IEEE CVPR*, 1997, pp. 40-46.

[14] P. Viola and W. M. Wells III. Mutual information: An approach for the registration of object models and images. *Inter. J. Comp. Vis.*, 1997.

[1] **Acknowledgments**: This work was supported by Microsoft Corp.