Part-Based Statistical Models for Object Classification and Detection

Elliot Joel Bernstein and Yali Amit

Department of Statistics, University of Chicago
E-mail: {bernstei,amit}@galton.uchicago.edu

Abstract

We propose using simple mixture models to define a set of mid-level binary local features based on binary oriented edge input. The features capture natural local structures in the data and yield very high classification rates when used with a variety of classifiers trained on small training sets, exhibiting robustness to degradation with clutter. Of particular interest are the use of the features as variables in simple statistical models for the objects thus enabling likelihood based classification. Pre-training decision boundaries between classes, a necessary component of non-parametric techniques, is thus avoided. Class models are trained separately with no need to access data of other classes. Experimental results are presented for handwritten character recognition, classification of deformed LaTeX symbols involving hundreds of classes, and side view car detection.

1. Introduction

Two subproblems of computer vision have received considerable attention in recent years: classification of objects in pre-segmented images and detection of a pre-specified class of objects in large images. Ultimately a comprehensive vision system will need to perform both tasks. A useful step in this direction involves the formulation of object representations that can be used for both tasks and allow easy transition between the two. This raises several issues of primary importance.

Scalability: Dealing with large numbers of classes poses problems not apparent in small-scale problems. We believe it is important that a system be able to add new classes without access to the training data for previously learned classes. Moving from classification to detection, the existence of multiple objects in a large image leads to an explosion in computation. Efficient multi-class classification will require sub-linear growth in computation as a function of the number of classes.

Modularity: This is raised as a key challenge in [7] and [4]. If a representation learned using pre-segmented training data is to be useful for multi-class vision, we will need a way of composing classifiers to choose among different scene labellings, each involving multiple objects. Probability models for objects provide a natural framework for achieving this.

Subclass Discovery: The partition of the set of all objects into classes is somewhat arbitrary, and elements of the same class may be topologically quite different. A vision system should be able to discover coherent subclasses.

We believe the issues of scalability and modularity can only be addressed by discovering a *compact and reusable* collection of mid-level local features, and using them to define simple, parsimonious statistical object models. These models can be easily composed to describe whole scenes and settle competitions among various scene interpretations, and they can be naturally incorporated in a coarse-to-fine computational strategy (see [4]). Furthermore, by using a mixture of distributions to represent each class, we are able to automatically discover subclasses. For example, Figure 3 demonstrates the discovery of a model for American sevens and one for European sevens. These issues have guided our thinking in developing the ideas presented in this paper.

1.1. Overview of the models

At the heart of our method is a model for local image patches. We propose to model *non-background* local image patches through simple, classical, statistical mixture models based on coarse, photometrically invariant, binary oriented edge features which are assumed independent conditional on the component. Each component of the mixture defines a new binary local feature and inherits the photometric invariance of the original edge features. The number of components is on the order of several tens or hundreds, depending on the size of the image patches. The primary motivation for defining these features is to achieve high invariance to object deformations without paying a heavy price in discriminatory power.

Spreading each feature into a neighborhood of its detected location greatly increases the stability of our representation to object deformations. However, since each local feature is a very rare event, on both generic background images and other object structures, this spreading operation does not produce large numbers of false object detections. Nor is there a significant loss in the discriminatory power of the individual features. Achieving the same level of invariance with simpler, more common local features, such as the original edges, would yield large numbers of false detections and discriminatory power would be lost.

Although it is possible to use the resulting features in any type of classifier, we extend the idea of mixture models from the feature level to the object level, now assuming conditional independence of the *features* in each component. With a statistical model defined for each class, classification reduces to maximizing likelihoods. Since the local features are common to all classes, and since training only requires examples from one class at a time, new classes can be added and learned without retraining the entire classifier. This is in contrast to non-parametric approaches which estimate decision boundaries directly. The use of a mixture model also provides a natural method for discovering any subclass structure that may be present in a class.

1.2. Overview of results

We demonstrate the utility of the local features and the object models in the problem of handwritten digit recognition using the MNIST dataset. We also demonstrate the universality of our feature set and the utility of our method for large numbers of classes on a dataset of artificially deformed LATEX characters, *using features learned from MNIST training data.* Finally, we demonstrate the ability of the method to deal with gray-level images by applying it to the problem of side view car detection, using generic features learned from non-car images.

We compare the model-based approach to a support vector machine (SVM) based on the same features. The conclusion from this experiment is that using simple and quite classical statistical models, on rather classical input (oriented edges), we can discover powerful local features from small amounts of training data. Furthermore, using a statistical model for each class, we achieve classification rates near the state of the art for nonparametric classifiers. The potential gain is that these models provide possible building blocks for a statistical model of scenes, something which can not be easily achieved with non-parametric classifiers.

2. Prior Work

2.1. Learning parts

In recent years, emphasis has shifted from predefining specific collections of features to discovering the features through training. For example in [11, 15] Haar-type features are used. In [3] informative features with low background probabilities are built from conjunctions of edges in a specific configuration. In [14, 15] features are chosen separately for each class to maximize discrimination, while in [13] a training procedure discovers informative features shared across classes, yielding more efficient algorithms.

The SIFT descriptor [10] is widely used and bears some similarity to our method. However, the SIFT descriptor has a very large support, even at the highest resolution, while our features are much more local. Our features also produce a much denser image representation, assigning features at all but the flattest image locations.

The convolutional neural network described in [9] discovers local features with shared weights across the image. The local image patches are learned indirectly as a result of the global optimization of the weights of the multilayer net with a cost determined by the object-level classification rate.

An alternative is to learn local structures from unlabeled data. For example, a sparse representation can be found for image patches as linear combinations of independent variables [5]. We have chosen a similar approach but prefer a more traditional and transparent form of learning using mixture models on *non-background* image patches.

2.2. Object models

In recent years there has been growing interest in constellation models under which objects are described in terms of sparse spatial configurations of local features. See for example the sparse models of [2] and [6]. These are used primarily in detection tasks and in themselves do not carry sufficient detail to classify between object classes with significant similarities such as characters. The models we propose are ‘dense’ and yield high classification rates. However sparse models can be easily extracted as approximations in a coarse-to-fine computational framework. This, however, is beyond the scope of this paper. There are also several approaches to statistical modeling of object classes that are highly dedicated to a particular category of objects, such as handwritten characters [12, 8].

3. Models for local features and objects

The point of departure for defining our local features is a set of eight oriented binary edges denoted $X_e(x)$, $e =$

$1, \ldots, 8$, at each pixel x. These are coarsely defined in terms of their orientation, are highly robust to photometric variations, and after a small amount of 'spreading' (a local MAX operation) are also robust to local deformations. They have been used extensively in recognition and detection experiments (see [2]).

3.1. A mixture model on non-background patches

We describe the distribution of edge maps on subwindows of size W centered anywhere in the image:

$$X_{W+y} = \{X_e(x+y), x \in W, e = 1, \ldots, 8\}.$$

Since we are not interested in 'wasting' features to model generic unstructured background, we define an elementary background model where all edges are conditionally independent with uniform probability p_{bgd}. For each image patch encountered, we count the number of edges n_W in the patch and compute the probability P_{n_W} of observing at least n_W edges under the background model. We reject the background hypothesis and model that patch as part of the *non-background* population if $P_{n_W} < 0.01$.

We use a mixture of conditional independence models on the binary edge variables. Assuming K_F components to the mixture, we define the probability of a particular configuration X_W

$$P(X_W) = \sum_{f=1}^{K_F} \tau_f P_f(X_W) \tag{1}$$

$$P_f(X_W) = \prod_{z \in W} \prod_{e} p_{z,e,f}^{X_e(z)} (1 - p_{z,e,f})^{(1 - X_e(z))}$$

where τ_f are the mixing probabilities. In words, an edge of orientation e at location z occurs with probability $p_{z,e,f}$, and all edges are conditionally independent given the model component f. These are classical models and can be trained with a straightforward implementation of the EM algorithm.

The resulting features are very easy to interpret. In Figure 1 we show the mean gray-level image for several MNIST parts, as this is easier to visualize than the actual probability maps $p_{z,e,f}$. Note that the unsupervised clustering process has discovered local structures such as curves, endings, and even junctions.

Estimation of this model requires that we specify the number of model components K_F. We have found experimentally that good classification results are obtained with about 100 features. Ideally, K_F would be internally estimated, using only the training data, possibly by means of either cross-validation or an information criterion like BIC.

3.2. Feature detection and spreading

Having learned a set of local features, we transform each image from its edge representation $X_e(x)$ to a local feature representation $Y_f(x),\ f = 1, \ldots, K_F$. A window, the same size as W, is swept across the edge map, and at each *non-background* region $x + W$ the most likely feature under the mixture model is recorded, i.e.

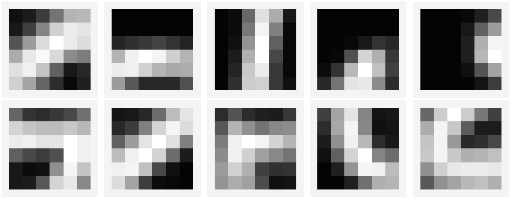

Figure 1. Mean images of ten sample clusters from MNIST feature learning.

$$Y_{f*}(x) = \begin{cases} 1 & \text{if } f^* = \arg\max_f \ \log P_f(X_{x+W}) \\ 0 & \text{otherwise} \end{cases}$$

Note that the computation of the log-likelihood at all locations is simply a linear convolution on the *binary edge data*, not the original image data.

The result of the image transformation is a new set of feature maps on the same image lattice L as the the original edge maps. Since each feature encodes an entire local structure, its exact position is no longer as important as the exact edge positions. We take advantage of this fact by spreading the detected features to a neighborhood of the original location. This defines a set of spread features

$$Y_f^s(x) = \max_{\xi \in B(x)} Y_f(\xi) \tag{2}$$

for $x \in L$ and $B(x)$ a neighborhood of x. In experiments, we took $B(x)$ to be a small square grid centered at x.

After spreading, the features are mapped to a coarser grid by dividing all coordinates by some factor. Note that after spreading and rescaling several features can be found at the same location. This greatly increases on object stability. However, since the features do not occur on generic background and at most one out of K_F features is allowed at each point on the original grid, each feature type remains a rare event.

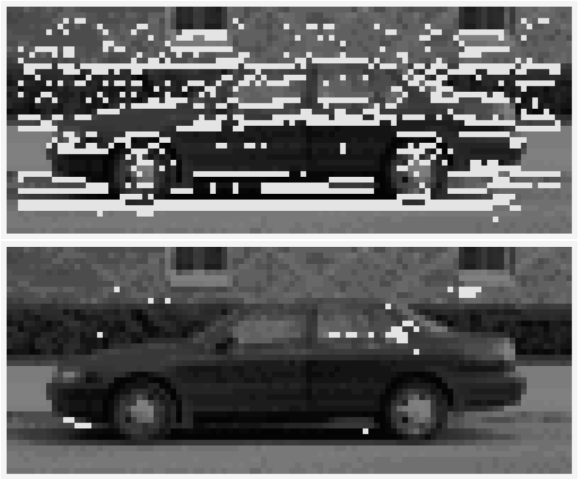

Figure 2. Locations of horizontal edges (top) and a typical feature with primarily horizontal structure (bottom) on a training image.

Figure 2 shows a training example from the side view car experiments along with the locations of all horizontal edges (top) and a particular local feature with predominantly horizontal structure (bottom). The contrast of the image has been reduced for display, and the edges used in this experiment are polarity-insensitive. Clearly, the presence or absence of a feature at a particular location on the grid carries much more information about the class variable than the presence or absence of an edge. Thus, we are able to work on a coarser grid, dramatically decreasing training and testing times without degrading classification or detection results. Note that, in the extreme, rescaling to a 1×1 grid is similar to the well known 'bag of features' paradigm: only the identities of the detected features are retained, without any information about their position.

3.3. Object models

Each class c was modeled as a mixture of K_c components of the same form as (1). That is, conditional on the model component m, each spread feature Y_f^s was assumed to occur independently at each location z in L with some probability $p_{z,f,m,c}$:

$$\begin{aligned} &P(Y^s|M=m, C=c) \\ &\quad = \prod_{z \in L} \prod_{f} p_{z,f,m,c}^{Y_f^s(z)} (1 - p_{z,f,m,c})^{(1 - Y_f^s(z))} \end{aligned}$$

Again the mixture is fit using the EM algorithm. Once the model for each class is estimated, testing proceeds by calculating

$$(\hat{C}, \hat{M}) = \operatorname*{arg\,max}_{c,m} \ \log P(Y \mid M=m, C=c).$$

Because of the sharp differences between the likelihood of the data under the different component models this is essentially the same as maximizing the posterior on class.

A significant computational advantage is gained from the use of binary features because the log-likelihood is of the form

$$\begin{aligned} &\log P(Y|M=m, C=c) \\ &\qquad = \alpha_{c,m} + \sum_{z,f} w_{z,f,c,m} \mathbf{1}\{Y_f^s(z) = 1\} \end{aligned}$$

and can thus be computed efficiently once the feature locations are established. While a similar model can be used with the original edges as features, the use of mid-level features as an intermediate representation incorporates some inter-edge correlations into the model while maintaining the conditional independence structure of the model specification (see Section 4.1).

4. Experimental Results

We have used these features in experiments with several datasets. Results are reported using the proposed statistical models and compared to an SVM with a quadratic kernel. It is clear from the results that the features are powerful representations for the purpose of classification even in the presence of hundreds of classes.

4.1. MNIST Digit Data

The MNIST dataset of handwritten digits contains 60,000 training and 10,000 test images that have been subject to some preprocessing. We have performed experiments using only a portion of the training set to demonstrate the ability of our feature learning technique to extract powerful features for classification from relatively few training examples.

One hundred features were learned from the first 1,000 images in the training set. Ten features are shown in Figure 1, each represented by the mean of all subimages whose conditional likelihood was maximized by the corresponding model component. Clearly the simple mixture model on *non-background* subimages yields features that capture local shape structure.

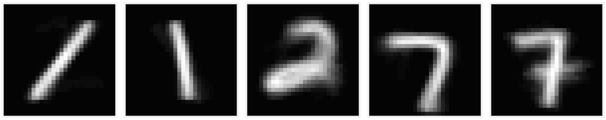

Figure 3. Mean images of five sample MNIST object models.

The features were then used to train several classifiers, using between 1,000 and 10,000 training examples taken

Classifier	Deshearing	Training Examples		
		1,000	5,000	10,000
Parts				
Mixture	On	2.82%	1.67%	1.60%
	Off	3.76%	2.06%	1.64%
SVM	On	2.37%	1.40%	1.06%
	Off	3.16%	1.59%	1.27%
Edges				
Mixture	On	4.47%	2.81%	2.17%
	Off	6.94%	3.61%	3.04%
SVM	On	3.07%	1.65%	1.26%
	Off	3.81%	2.04%	1.70%

Table 1. Error rates for MNIST classification.

from the beginning of the MNIST training set. Table 1 shows classification results using the various classifiers, with and without additional preprocessing by a 'deshearing' step. Some examples of the resulting object models (without deshearing) are presented in Figure 3. Again we show the mean images of the clusters, rather than the probability maps.

The two models of class '7' demonstrate the ability of our method to discover natural subclasses, one of the principles outlined in Section 1. Note that the clustering procedure also picks up on effects that might be better handled explicitly, like rotation. Contrasting results, using only edges as the features, are also reported in Table 1. We used the same number of model components for each class, and the number was tuned using a validation set of unused images from the MNIST training set.

Classification times: Part-based classification times with the mixture model ranged from approximately 1ms - 7ms per digit, depending on the number of model components. For the SVM, classification time ranged from 12ms with 1,000 training data to 42ms with 10,000. The transformation from gray-level images to parts takes an additional 11ms per digit. All experiments were conducted using a Pentium-M 1.7GHz CPU.

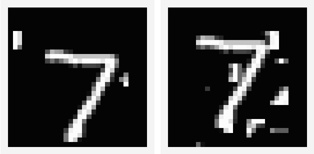

Figure 4. A test digit with clutter at levels two and ten.

We also synthesized noisy test images and attempted to classify them. This was done by selecting random 4×4 non-background regions from MNIST digits and placing them at random locations in each test digit. Gray-level values were combined by maximization. Figure 4 shows two examples of the same digit with different amounts of clutter added. Although the 4×4 windows are smaller than the window used to train our parts, this gives realistic-looking clutter which is generally assigned some part label.

Classifier	Features	Level of Clutter		
		2	6	10
Mixture	Parts	3.61%	5.82%	8.76%
	Edges	5.46%	9.53%	14.06%
SVM	Parts	2.35%	4.91%	8.16%
	Edges	5.37%	8.79%	13.30%

Table 2. Error rates for cluttered MNIST digit classification, using 5,000 training digits.

Deshearing is not performed since it is very sensitive to clutter. The data are, however, centered in the frame (after the addition of clutter) by calculating the center of mass of the gray-level data and translating it to the center of the image grid. The idea was to mimic the effect of attempting to properly align a selected region of interest for classification. Since the MNIST data have already been centered in this manner, failure to recenter after adding clutter would artificially reduce error rates. Table 2 shows error rates for classification of artificially cluttered data using classifiers trained with 5,000 data points. For comparison, we also show results using the edge data alone. As predicted, classification with edges degrades faster. Furthermore, mixture model performance with parts is competitive with the SVM, especially for high levels of clutter.

Although the SVM results with edges are comparable to mixture model results with parts (on clean data), the latter retains several advantages. The modular nature of the mixture model makes it better suited for use in scene interpretation. Since the parts themselves occur much more rarely than edges, they are inherently more useful for detection at low false negative rates. Also, the parts exhibit superior classification performance in the presence of clutter – a consideration that is important when dealing with realistic scenes.

We are most interested in performance with small training sets, but the power of our features is further demonstrated by the fact that, if we train an SVM using parts and 20,000 training digits, we achieve a classification error rate of *0.90% with deshearing and 0.96% without*.

4.2. Artificially Deformed LaTeX Characters

To test the scalability of this approach to a large number of classes we employed the randomly deformed LaTeX data set of [2]. A small sample from the dataset is presented in Figure 5, with classification rates presented in Table 3. The model was trained using only thirty examples from each class to estimate a two-component mixture.

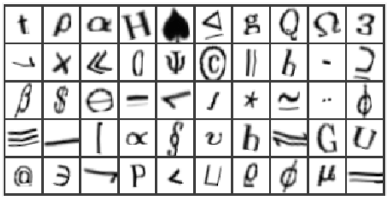

Figure 5. Deformed LaTeX characters.

This example also demonstrates the universality of our feature set, since we did not estimate a new set of features from the LaTeX training data. Rather, we used the same local features from the previous experiments, estimated from the first 1,000 MNIST training examples. In essence, these local features have captured the generic local structure of line drawings.

Classifier	Number of Classes		
	100	200	293
Mixture	3.63%	3.62%	4.72%
SVM	1.80%	1.50%	2.58%

Table 3. Error rates for the LaTeX experiment.

4.3. Side View Car Detection

Finally, we tested our method in a full detection framework on the side view car dataset of [1]. First, a collection of fifty 10×10 local features was learned from only the negative training examples. In order to develop local features for use in a variety of detection tasks, no images of cars were used at this stage. Then, a two-component mixture model for the car class was estimated using *only the first fifty* positive training examples. No additional use was made of the negative examples. In particular, while the method of [1] requires explicit training of a classifier to discriminate between positive and negative examples, we require no such step.

Tests for detection were constructed by first using the two-component clustering results to learn models for the object at two resolutions: 10×4 and 50×20. (The original training images were 100×40.) The feature detection algorithm was applied to each training image, the resulting feature maps were rescaled to the required resolution, and the model parameters were estimated.

Detection proceeded by first applying a test for the coarse model at every image location, followed by a test for the finer model at all locations that passed the initial test. The test was based on the log-likelihood ratio test statistic, $\log (P_o(Y)/P_b(Y))$, where $P_o(\cdot)$ denotes the likelihood under the object model and $P_b(\cdot)$ denotes the likelihood under a generic background model. Note that the background model is *not* learned from negative examples. Rather, it is based on all features occurring independently at all locations with some fixed probability. The universal background probability p_b can be adjusted to tune the sensitivity of the detector. Furthermore, since the distribution of the test statistic can be estimated on the training data, it is possible to select a reasonable threshold *a priori*, without reference to validation data.

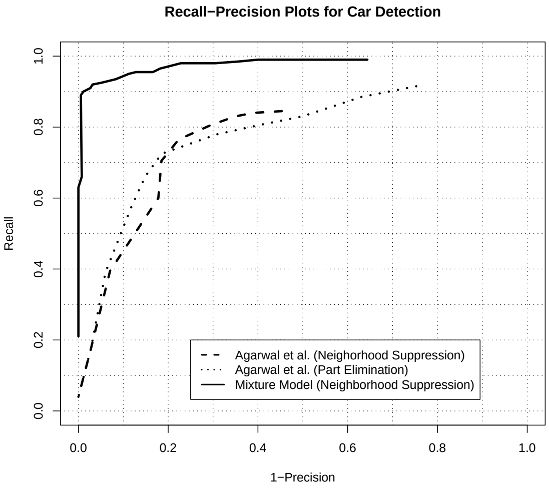

Figure 6. Recall-precision curves for side view car detection.

A recall-precision curve of our results on test data with fixed-size cars is shown in Figure 6, along with two corresponding curves from [1]. ‘Neighborhood Suppression’ and ‘Part Elimination’ refer to two methods for pruning multiple competing detections. We used only the neighborhood suppression method, which assigns to each detection a region of influence within which all weaker detections are eliminated. In our framework, the strength of a detection is simply its likelihood ratio. Figure 7 shows two example test images with cars correctly detected.

5. Discussion and Future Work

We have shown that a simple clustering procedure on the population of local image patches yields local features that efficiently describe local object structures, even when trained on a generic collection of images. Simple classifiers defined in terms of these features yield very low error rates on standard problems with small training sets. In particular, using these features, it is possible to describe classes as mixtures of conditional independence models and achieve low error rates with likelihood based classification. The classes

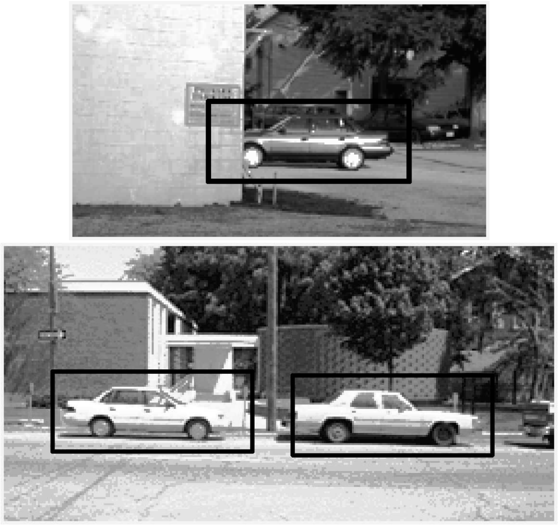

Figure 7. Detected cars in test images.

can be learned sequentially, and it is not necessary to train on background images for detection.

As mentioned our method bears some similarity to the convolutional neural network described in [9], which alternates between convolutional layers that extract local features with shared weights across the image and sub-sampling layers that sub-sample the image. A major difference is the form of learning and how classification is performed. The local features are learned directly from the population of local image patches, object models are learned separately, and classification is likelihood based. Furthermore very competitive results are obtained with very small data sets. In the present setting we use only one level of features between the elementary edges and the objects. Using the same mechanism one can imagine discovering higher level features as components of mixture models on local configurations of the mid-level features.

Currently, we classify by comparing an observation to each mixture component and choosing the component that maximizes the likelihood. It would be more efficient to eliminate multiple hypotheses simultaneously, in a coarse-to-fine framework, as in [4], reserving computation of the component likelihoods for only a few likely candidates.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, November 2004.

[2] Y. Amit. *2D Object Detection and Recognition: Models, Algorithms, and Networks*. MIT Press, 2002.

[3] Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11:1691–1715, 1999.

[4] Y. Amit, D. Geman, and X. D. Fan. A coarse-to-fine strategy for multi-class shape detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.

[5] A. J. Bell and T. J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[6] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the International Conference on Computer Vision*, volume 1, 2003.

[7] S. Geman, D. F. Potter, and Z. Chi. Composition systems. *Quarterly of Applied Mathematics*, LX:707–736, 2002.

[8] I.-J. Kim and J.-H. Kim. Statistical character structure modeling and its application to handwritten Chinese character recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(11):1422–1436, November 2003.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[11] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of International Conference on Computer Vision*, 1998.

[12] M. Revow, C. K. Williams, and G. E. Hinton. Using generative models for handwritten digit recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(6):592–606, 1996.

[13] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. AI Memo 2004-008, Massachusetts Institute of Technology, April 2004.

[14] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[15] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.