(6)

7

CVPR'96

Competitive Mixture of Deformable Models for Pattern Classification *

Kwok-Wai Cheung Dit-Yan Yeung Roland T. Chin

Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
{william, dyyeung, roland}@cs.ust.hk

Abstract

Following the success of applying deformable models to feature extraction, a natural next step is to apply such models to pattern classification. Recently, we have cast a deformable model under a Bayesian framework for classification, giving promising results. However, deformable model methods are computationally expensive due to the required iterative optimization process. The problem is even more severe when there are a large number of models (e.g., for character recognition), because each of them has to deform and match with the input data before a final classification can be derived. In this paper, we propose to combine the deformable models into a mixture, in which the individual models compete with each other to survive the matching process during classification. Models that do not compete well are eliminated early, thus allowing substantial savings in computation. This process of competition-elimination has been applied to handwritten digit recognition in which significant speedup can be achieved without sacrificing recognition accuracy.

1 Introduction

Recently, some deformable models (DM) have been proposed and successfully applied to extract non-rigid objects from imagery, e.g. [1, 2]. A natural next step following extraction is deformable pattern classification. A number of systems combining DM extraction with statistical classifiers [3] or artificial neural networks [4] have been attempted for deformable object recognition. These systems are seemingly *ad hoc*, in that extraction and classification are treated as disjoint components. Bayesian inference, which provides probabilistic interpretation and a unified framework, has recently been used to integrate deformable extraction and classification tightly together. Moreover, an efficient implementation known as the *evidence framework* made popular by MacKay [5], where classification is regarded as model comparison, has been adopted for off-line isolated handwritten digit recognition achieving about 95% recognition accuracy [6].

Model-based pattern classification is known to have limitation in scaling up, in that the recognition time increases linearly with the number of model candidates. The problem becomes more serious for DMs where each model has to iteratively deform, fit to the data, and then converge. This iterative process is computationally expensive hindering its potential usefulness. Although some efficient techniques such as geometric hashing have been proposed to tackle the problem, they basically assume that the object to be recognized is not highly deformed and is represented by a set of pre-extracted salient points like corners (i.e. not working in pixel level). In order to alleviate the scale-up limitation of model-based deformable pattern classification, we propose to use competitive mixture of DMs for classification. During the process of competition, model candidates that do not show high potential of success are eliminated early, with elimination being an integral part of the optimization process. Experiments have been performed using the proposed mixture model for handwritten digit recognition. Results show that such an approach provides significant speedup without sacrificing accuracy.

2 Bayesian Framework for Deformable Models

Deformable models are formulated by a model deformation energy function and a data-misfit energy function [1]. Using the Gibbs distribution, they can

*The research work reported in this paper has been supported by the Hong Kong Research Grants Council under grant HKUST 614/94E.

be given probabilistic meanings. Under the Bayesian framework, the model deformation energy corresponds to the prior distribution for the model parameters and the data-misfit energy corresponds to the likelihood of the data. Optimal feature extraction then becomes parameter estimation while classification becomes model selection. Both of them can be computed using Bayesian inference procedures [5]. The following provides a brief overview of the Bayesian framework in the context of deformable pattern classification.

2.1 Three Levels of Inference

Let H_i denote the model for class i, $\mathbf{D}$ the input image data, $\mathbf{w}$ the model parameter vector describing object shape, α the regularization parameter, and β the data signal strength. α and β are known as hyperparameters.

Level 1. Modeling: A number of models $\{H_i\}$, each for a particular class i, are constructed. Training is typically involved.

Level 2. Feature Extraction: Optimal parameters $\{\mathbf{w}^*, \alpha^*, \beta^*\}$ for the model H_i are extracted by a best fit of H_i to the input image data. The process is equivalent to first (i) maximizing the posterior probability density $p(\alpha, \beta|\mathbf{D}, H_i)$ and then (ii) maximizing $p(\mathbf{w}|\mathbf{D}, \alpha, \beta, H_i)$, resulting in a maximum of $p(\mathbf{w}, \alpha, \beta, |\mathbf{D}, H_i)$.

Level 3. Classification: The best model is determined by selecting the model with the maximum likelihood $p(\mathbf{D}|H_i)$, under the assumption of uniform model prior, that is, to maximize

$$p(\mathbf{D}|H_i) = \int_{\alpha,\beta}\int_{\mathbf{w}} p(\mathbf{w}|\alpha, H_i)p(\mathbf{D}|\mathbf{w}, \beta, H_i) \; p(\alpha|H_i)p(\beta|H_i)d\mathbf{w}d\alpha d\beta \quad (1)$$

2.2 Model Formulation

2.2.1 Deformation Model (Prior Distribution)

Handwritten characters are modeled as splines, which are parameterized by a small set of k control points. The amount of deformation, referred to as deformation energy, $E_w(\mathbf{w})$, is measured by the Mahalanobis distance of the control point vector, $\mathbf{w} \in \Re^{2k}$, from the pre-defined home location vector, $\mathbf{h} \in \Re^{2k}$. The vectors are formed by concatenating the x and y coordinates of all the k control points, i.e., $\mathbf{w} = [x_1, y_1, x_2, y_2, ...x_k, y_k]^t$. Specifically, the deformation energy is expressed as

$$E_w(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{h})^t \mathbf{\Sigma}^{-1}(\mathbf{w} - \mathbf{h}) \quad (2)$$

where $\mathbf{\Sigma}$ is the $2k \times 2k$ covariance matrix of $\mathbf{w}$. Subsequently, the prior probability distribution of $\mathbf{w}$ is given by

$$p(\mathbf{w}|\alpha, H_i) = \frac{1}{Z_w(\alpha)} \exp(-\alpha E_w(\mathbf{w})) \quad (3)$$

where

$$Z_w(\alpha) = \left(\frac{2\pi}{\alpha}\right)^k |\mathbf{\Sigma}|^{1/2}. \quad (4)$$

The components of $\mathbf{h}$ and $\mathbf{\Sigma}$ are computed by maximum likelihood estimation during the training process (Level 1 inference). Figure 1 shows the digit models after training.

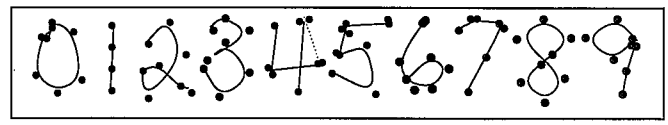


Figure 1: Ten digit models after training (Black dots represent the spline control points).

2.2.2 Data-Misfit Model (Likelihood of Data)

Let the input image be binary. The distribution of black pixels is modeled using a uniformly weighted mixture of Gaussians with their centers equally spaced along the spline. The misfit is represented by the data-misfit energy

$$E_D(\mathbf{w}) = -\log\left[\prod_{l=1}^{N} \frac{1}{N_g} \sum_{j=1}^{N_g} \exp\left(-\beta \frac{\|\mathbf{m}_j - \mathbf{y}_l\|^2}{2}\right)\right] \quad (5)$$

and subsequently, the likelihood of the data is given by

$$p(\mathbf{D}|\mathbf{w}, \beta, H_i) = \frac{1}{Z_D(\beta)} \exp(-E_D(\mathbf{w})) \quad (6)$$

where

$$Z_D(\beta) = \left(\frac{2\pi}{\beta}\right)^N, \quad (7)$$

$$\mathbf{m}_j = \mathbf{S}_j^t \mathbf{w}, \quad (8)$$

N is the number of black pixels, N_g is the number of Gaussians on the spline, $\mathbf{S}_j$ is a $2k \times 2$ matrix containing the cubic B-spline coefficients, β is the inverse of the Gaussian variance which models the character stroke width, $\mathbf{m}_j \in \Re^2$ is the center of the j^{th} Gaussian, and $\mathbf{y}_l \in \Re^2$ is the location of an individual black pixel.

2.2.3 Affine Transform

In order to achieve invariance in shifting, rotation, shearing and size change, affine transform $\{\mathbf{A}, \mathbf{T}\}$ is used to map the control point vector $\mathbf{w}$ in the model frame to the Gaussian mean $\mathbf{m}_j$ in the data frame.

$$\mathbf{m}_j^t = \mathbf{S}_j^t(\mathcal{A}\mathbf{w} + \mathcal{T}) \tag{9}$$

where $\mathcal{A}$ is a $2k \times 2k$ block diagonal matrix with k submatrices $\mathbf{A}$ placed on its diagonal and $\mathcal{T}$ is a $2k \times 1$ vector formed by concatenation of k $\mathbf{T}$ vectors.

2.2.4 Overall Model (Posterior Distribution)

Combining the deformation energy and the data-misfit energy, given by

$$E_M(\mathbf{w}) = \alpha E_w(\mathbf{w}) + E_D(\mathbf{w}) \tag{10}$$

the posterior distribution of $\mathbf{w}$ is defined as

$$p(\mathbf{w}|\mathbf{D}, \alpha, \beta, H_i) = \frac{1}{Z_M(\alpha, \beta)} \exp(-E_M(\mathbf{w})) \tag{11}$$

where $Z_M(\alpha, \beta) = \int \exp(-E_M(\mathbf{w}))d\mathbf{w}$. This posterior distribution is used in subsequent extraction and classification.

2.3 Feature Extraction

2.3.1 MAP Estimate of Spline Control Points

The MAP estimate of the spline control point vector $\mathbf{w}$ is obtained by maximizing $p(\mathbf{w}|\mathbf{D}, \alpha, \beta, H_i)$ in EQ.(11). The *Expectation-Maximization* (EM) algorithm is known to be efficient for such maximization and is used here. It is an iterative process consisting of two steps:

- *Expectation*-step

$$h_j^l(\hat{\mathbf{w}}) = \frac{\exp(-\beta \frac{\|\mathbf{m}_j - \mathbf{y}_l\|^2}{2})}{\sum_p \exp(-\beta \frac{\|\mathbf{m}_p - \mathbf{y}_l\|^2}{2})} \tag{12}$$

$$Q(\mathbf{w}|\hat{\mathbf{w}}) = -\alpha E_w(\mathbf{w}) - \beta E'_D(\mathbf{w}) \tag{13}$$

$$E'_D(\mathbf{w}) = \sum_{l=1}^{N} \sum_{j=1}^{N_g} \frac{h_j^l(\hat{\mathbf{w}}) \|\mathbf{m}_j - \mathbf{y}_l\|^2}{2} \tag{14}$$

- *Maximization*-step

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} Q(\mathbf{w}|\hat{\mathbf{w}}) \tag{15}$$

where $\hat{\mathbf{w}}$ is the control point vector obtained in the previous EM iteration.

The two-step process iterates to reach a converged solution, which corresponds to a local minimum of $E_M(\mathbf{w})$ or equivalently a local maximum of $p(\mathbf{w}|\mathbf{D}, \alpha, \beta, H_i)$. The parameters of the Q-function in EQ.(14), in fact, involve both the affine transform $\{\mathbf{A}, \mathbf{T}\}$ and the control point vector $\mathbf{w}$, which make $Q(\mathbf{w}|\hat{\mathbf{w}})$ a fourth order function. To reduce it to quadratic form so that it can be solved using linear techniques, a two-stage process as in [4] is used. For initialization, it is done by yet *another* EM-step, where the spline control points are fixed and the Q-function is maximized with respect to the affine transform parameters.

2.3.2 Regularization Parameter and Stroke Width Estimation

By maximizing the posterior probability density $p(\alpha, \beta|\mathbf{D}, H_i)$, the MAP estimates of α and β can be determined. First, $\mathbf{w}$ is transformed to an orthogonal basis such that $\nabla\nabla E_w(\mathbf{w}) = \mathbf{I}$. Then, setting the first derivative of $p(\alpha, \beta|\mathbf{D}, H_i)$ to zero, the MAP estimates α^* and β^* must satisfy

$$\alpha^* = \frac{\gamma}{2E_w(\mathbf{w}^*)} \tag{16}$$

$$\beta^* = \frac{2N - \gamma}{2E'_D(\mathbf{w}^*)} \tag{17}$$

where $\gamma = 2k - \alpha Trace(\nabla\nabla E'_M(\mathbf{w}^*)^{-1})$ and $\nabla\nabla E'_M(\mathbf{w}^*) = \alpha\mathbf{I} + \beta\nabla\nabla E'_D(\mathbf{w}^*)$.

Since there exist no closed-form solutions for α^* and β^*, the model fitting step and the $\{\alpha^*, \beta^*\}$ estimation step are implemented in an iterative fashion where EQ.(16) and (17) are used as the convergence criteria and some initial values of α and β are required. Figure 2 illustrates the steps of the extraction (matching) process.

2.4 Classification (Model Selection)

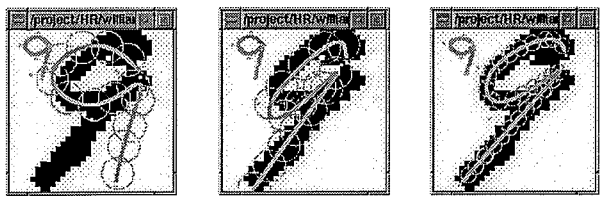
Classification involves computing the evidence $p(\mathbf{D}|H_i)$ based on $\{\mathbf{w}^*, \alpha^*, \beta^*\}$ for each model i, given by

$$p(\mathbf{D}|H_i) \propto \frac{Z_M(\alpha^*, \beta^*)}{Z_w(\alpha^*) Z_D(\beta^*)} \sqrt{\frac{2}{\gamma}} \sqrt{\frac{2}{2N - \gamma}} \tag{18}$$

where $Z_M(\alpha^*, \beta^*)$ is approximated by

$$Z_M(\alpha^*, \beta^*) \simeq \exp(-E_M(\mathbf{w}^*))(2\pi)^k |\nabla\nabla E'_M(\mathbf{w}^*)|^{-1/2} \tag{19}$$

Finally, classification is determined as $\arg\max_i p(\mathbf{D}|H_i)$.



(a) (b) (c)

Figure 2: Illustration of deformable pattern extraction. The small character near the upper left corner of each figure is the model. (a) Initial position of the model. (b) Model initialization using the proposed EM step for affine transform estimation. (c) Final fit.

3 Deformable Model-based Classification

A multi-class DM-based recognition system can be realized directly as described above. It is apparent that this approach has limitation in scaling up because the recognition time is proportional to the number of model candidates. Using the example of recognizing alphanumerals, the problem requires a 36-class system. Direct computation to recognize an unknown character as described in Section 2 requires matching 36 deformable models independently. We propose using a competitive mixture of models so that unlikely models can be eliminated early, leaving a (hopefully) small set of candidate models competiting for the best match.

3.1 Mixture of Deformable Models

Let us assume that the input image was generated uniquely by *one and only one* of the deformable models. Let $\mathcal{H} = \{H_1, H_2, ...H_M, \pi_1, \pi_2, ..., \pi_M\}$ denote a mixture of M models. A probability density function $p(\mathbf{D}|\mathcal{H})$ is defined by combining the evidence values of individual models, given as

$$p(\mathbf{D}|\mathcal{H}) = \sum_{i=1}^{M} \pi_i \; p(\mathbf{D}|H_i) \tag{20}$$

subject to the constraints $\pi_i \geq 0, \forall i$ and $\sum_i \pi_i = 1$. The mixture coefficient π_i is a measure which indicates how relevant is the particular model H_i to the input data $\mathbf{D}$.

Instead of computing the evidence values of all the model candidates for classification, we use a mixture of DMs to account for the input image. Due to the *one-and-only-one* assumption and the constraints on π_i, the ideal outcome of maximizing $p(\mathbf{D}|\mathcal{H})$ is to have only one of the mixture coefficients being one and all others being zero. In other words, the competition for final selection becomes a process of adjusting the coefficients under the constraints that all the coefficients are non-negative and sum to unity.

Updating the mixture coefficients based on the input image is done by maximizing $p(\mathbf{D}|\mathcal{H})$ using the EM algorithm. The associated Q-function is defined as

$$Q(\pi_i|\hat{\pi}_i) \;=\; \sum_{i=1}^{M} h_i \left(\log \pi_i + \log p(\mathbf{D}|H_i)\right) \tag{21}$$

where

$$h_i = \frac{\hat{\pi}_i \; p(\mathbf{D}|H_i)}{\sum_j \hat{\pi}_j \; p(\mathbf{D}|H_j)} \tag{22}$$

To maximize $Q(\pi_i|\hat{\pi}_i)$ subject to the constraints on π_i, the updating rule for π_i is

$$\pi_i = \frac{\hat{\pi}_i \; p(\mathbf{D}|H_i)}{\sum_i \hat{\pi}_i \; p(\mathbf{D}|H_i)} = h_i \tag{23}$$

Thus, π_i is re-estimated as the probability that the data $\mathbf{D}$ was generated by digit model i.

3.2 Elimination Process

Using the mixture of DMs as defined in EQ.(20) while keeping the full model set in $\mathcal{H}$ does not yield any computational advantage. It merely represents a change of convergence criteria, from the convergence of the parameters α, β and $\mathbf{w}$ to that of the mixture coefficients π_i. However, the updating of π_i during the optimization process provides an informative indicator showing the relevancy of H_i to the given data. This set of mixture coefficients is the basis for elimination. First, deformable pattern extraction (as described in Section 2.3) is initialized simultaneously for all models, involving the EM iterative procedure to perform model fitting. The mixture coefficients of all the models are then computed as an intermediate indicator during the iteration according to EQ.(23), after which some model candidates are eliminated from further matching. The removal is based on the relative value of π_i to all the other coefficients as well as its proximity to zero. The simplest removal rule is to eliminate at the end of each iteration the model that has the smallest coefficient. Other removal rules are also possible, and in general, the process involves the elimination of $(M - R)$ model candidates after a certain number of iterations until only one candidate remains with its coefficient larger than a value θ which is close to 1. In one experiment, $(M - R)$ models were eliminated after the first iteration of model fitting, and the

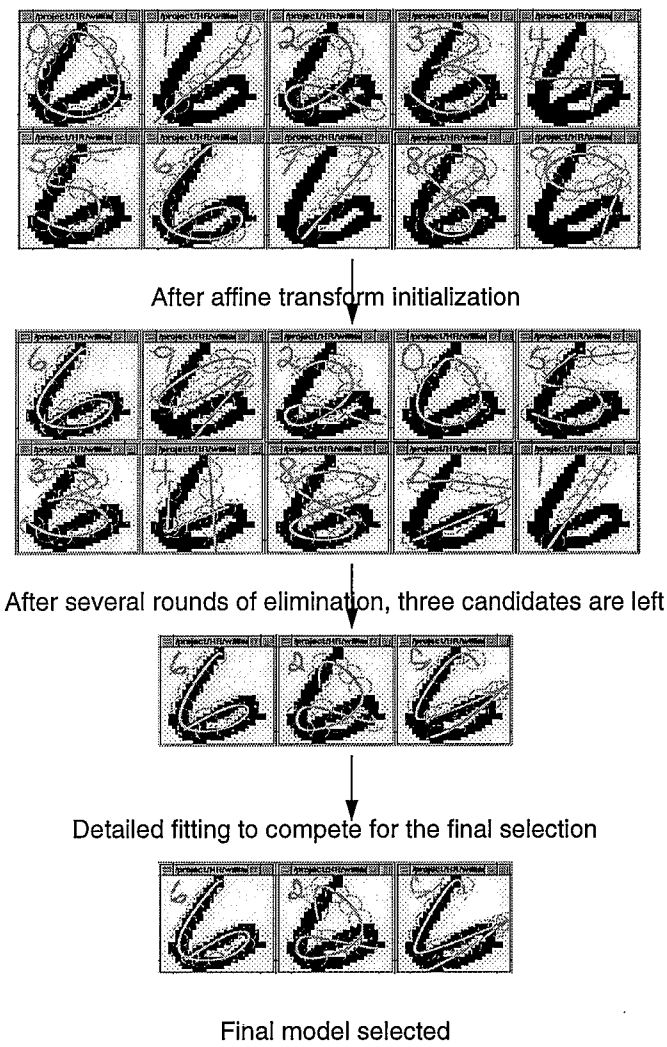


Figure 3: Illustration of the elimination process using the competitive mixture of models. After affine transform initialization for each digit model, the models are ranked according to their evidence estimates given by EQ.(18). The models that do not show potential of success early on in the process are eliminated. In our experiment, only the best 3 models are retained after the first iteration. The remaining models continue to fit the image data more closely and compete for the final selection.

process continued with the remaining R candidates until it converged when one of the mixture coefficients exceeded $\theta = 0.8$ (see Figure 3). Some preliminary results will be discussed in the next section.

The accuracy and computational cost of the proposed competitive classification process depend on the choices in the removal rule, such as the number of retained model candidates R, the removal criterion (e.g., π_i becomes close to zero), the convergence criterion (e.g., π_i becomes close to one), and the frequency of removal (e.g., removal after every three iterations). The frequency of removal can be indirectly controlled by a temperature factor τ to adjust the convergence rate of EM. For example, a modified updating rule for π_i is

$$\pi_i = \frac{\hat{\pi_i} \ \exp(\log p(\mathbf{D}|H_i)/\tau)}{\sum_i \hat{\pi_i} \ \exp(\log p(\mathbf{D}|H_i)/\tau)} \tag{24}$$

The following section depicts some preliminary experimental results showing the effectiveness of the proposed competitive classification approach.

3.3 Results

3.3.1 Recognition Accuracy

Experiments were performed by applying the proposed competitive mixture of DMs to recognize digits in the NIST Special Database 1. The subset used contains 3,471 digits (32x32 binary images) written by 49 different individuals. 2,044 digits were used for training and the rest for testing. In one experiment, a single elimination rather than repetitive removal was applied to evaluate the speedup and accuracy separately, and the experiment was repeated with different values of R. The recognition accuracy results are shown in Figure 4. As expected, the recognition accuracy generally drops when R decreases. With a small number of models retained for the final matching, say $R = 1$, the accuracy suffers. However, a slight increase to $R = 2$ yields a substantial improvement in accuracy. Further increase in the number of retained models only yields diminishing increase in the accuracy. This simple experiment shows that at least half of the candidate models can be eliminated without significantly affecting the recognition performance.

3.3.2 Computational Cost

The same experiment was conducted to investigate the speedup performance. To measure speedup using the proposed mixture model, we define a speedup metric S_R as

$$S_R = \frac{t_{all}}{t_R} \tag{25}$$

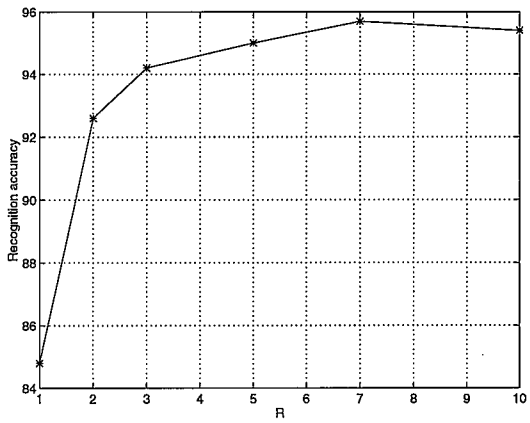


Figure 4: A plot of accuracy against R ($\theta = 0.8$ and $\tau = 200$).

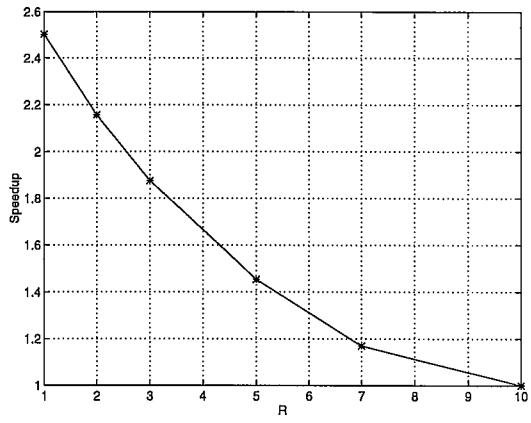


Figure 5: A plot of speedup against R ($\theta = 0.8$ and $\tau = 200$).

where t_{all} is the computational time required by the proposed mixture models without any elimination (more or less the same computational requirement as the original method) and t_R is the time required when R models are retained. As evident in Figure 5, speedup is achieved by eliminating candidate models that do not compete well. Based on Figures 4 and 5, this set of digits requires only 3 model candidates for the final matching to achieve close to the highest accuracy with a speedup of 1.9 times. Further speedup can be achieved if we reduce the computational cost of model initialization which is a major cost in the algorithm. In one experiment (see Table 1), affine transform initialization was applied to a coarse image (one quarter of full resolution), speedup was then raised to around 2.9 with a slight drop in accuracy. It should be noted that speedup is expected to be more significant when there are more models.

4 Conclusion and Future Work

Competitive mixture of deformable models is proposed for multi-class non-rigid pattern classification where each model can deform for robust recognition. When the number of classes is large and the patterns within each class vary a great deal, deformable pattern classification becomes very computationally intensive and hence hinders its practical usefulness. This proposed competitive approach enables the elimination of model candidates early on in the matching process, and thus achieves significant speedup without sacrificing accuracy. Handwriting recognition using a subset of the NIST Special Database 1 was used to demonstrate the proposed approach. The preliminary results based on this application show the potential usefulness of such an approach. Furthermore, this idea can be extended to include both competitive and cooperative mixtures in order to further achieve both speedup and accuracy improvement.

Different ways of initialization	Speedup	Accuracy
Affine transform at full resoln.	1.9	94.2%
Affine transform at low resoln.	2.9	92.2%
Without affine transform	3.4	88.0%

Table 1: Performance (speedup and accuracy) comparison of different methods for model initialization (R=3).

References

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snake: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.

[2] A. Yuille, D. Cohen, and P. Hallinan, "Feature extraction from faces using deformable templates," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, pp. 104–109, 1989.

[3] T. Hastie and R. Tibshirani, "Handwritten digit recognition via deformable prototypes," Technical Report, Statistics and Data Analysis Research Dept., AT&T Bell Labs., Murray Hill, NJ, USA, July 1992.

[4] C. K. I. Williams, *Combining deformable models and neural networks for handprinted digit recognition.* PhD thesis, University of Toronto, Nov. 1994.

[5] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.

[6] K. W. Cheung, D. Y. Yeung, and R. T. Chin, "A unified framework for handwritten character recognition using deformable models," *Proceedings of 2nd Asian Conference on Computer Vision*, vol. I, Singapore, pp. 344–348, Dec. 1995.