Natural Image Segment Classification

Stat 882 – Pattern Recognition

Brian Rigling

20 March 2002

I. Introduction

Methods for accurately classifying segments of images are of growing value in many fields. Researchers in the area of computer vision wish to use machine learning to enable artificial intelligence platforms to interact with a visually sensed environment. Those interested in performing lossy image compression also have interests in image segment classification. By successfully identifying the contents of an image, variable levels of compression may be applied to different portions of the image. This may also be extended to implementation in video coding algorithms.

Existing work in the field includes research done by Michael Revow, in the field of hand-written character classification. In his studies, a parametric model, such as a spline, would be fit to the curves composing the hand-written characters. The parameters of the model fit would then be used as extracted features fed into a neural network classifier [1-5]. Song Chun Zhu has done significant work in segmentation of full images, using Markov chain models [6]. By using a Markov model, he is able to jointly classify many adjacent segments of the image. Steve Waterhouse has done work in applying mixtures of experts to image classification [7]. An expert is an algorithm that is trained to recognize a particular object or event within an image. By applying multiple experts to a given image, each expert is able to decompose the portions of the image that it is able to represent more effectively than any of the other experts.

In this report, we will study the application of a variety of pattern recognition techniques to the image segment classification problem. In specific, the challenge is to independently classify segments extracted from outdoor images into one of m classes. The data set used in this study, obtained from the Delve [8] web site, has also been the

subject of experiments performed by many other researchers. Radford Neal applied the nearest neighbor classification method to this data set, possibly in the course of the study discussed in [9]. At one point, unknown parties applied CART [10-11] software, a data-mining tool that automatically searches for patterns and relationships in highly complex data. The aforementioned Michael Revow applied a *k*-nearest neighbor technique, and Steve Waterhouse also studied this data set in his Ph.D. thesis [7]. Unfortunately, results from none of these studies were readily available. While listings of the results were available on the Delve web site, no documentation to enable interpretation of these numbers was provided.

The next section of this report will detail the data set under consideration. Section III will briefly describe each classification algorithm applied and will present the results from each. Finally, Section IV will summarize our observations and conclusions.

II. Data Set

Carla Brodley originally created the natural image segment data set during her time with the Vision Group at the University of Massachusetts in 1990. The image segments were extracted from seven outdoor images. Each segment is a full-color nine-pixel square. Each of the image segments fits into one of seven classes: cement, brickface, foliage, grass, path, sky, or window. There were 210 samples, 30 of each class, partitioned for training, and 2100 samples, 300 of each class, were set aside for testing.

From each image segment, 16 continuously valued features were extracted, and it is these extracted features that are contained in the data set. The first two features, *short-line-density-5* and *short-line-density-2*, are continuously valued over the range [0,1] and

are defined to be the low and high contrast line counts, respectively. The next four features, *vedge-mean*, *vedge-sd*, *hedge-mean* and *hedge-sd*, are continuously valued over [0,∞) and specify the mean and standard deviation of the horizontal and vertical contrasts. The *intensity-mean*, continuously valued over [0,∞) is the average pixel intensity of the segment, defined as the mean of the next three attributes. The *rawred-mean* (R), *rawblue-mean* (B), and *rawgreen-mean* (G), defined over [0,∞), are the average red, blue, and green over the segment. Then, the *exred-mean*, *exblue-mean*, and *exgreen-mean*, defined over (-∞,∞), are the excess red (2R - (G + B)), excess blue (2B – (G + R)), and excess green (2G – (R + B)). Finally, the *value-mean*, *saturation-mean*, and *hue-mean* are a non-linear three-dimensional transformation of R, G, and B.

The first step in this study was to perform a multiple discriminant analysis on the features of the training data. Following the development in section 3.8.3 of [16], we determined the within-class scatter matrix S_W and the between-class scatter matrix S_B. This allows us to pose and solve the generalized eigenvalue problem

$$S_B w_i = l_i S_W w_i \qquad (1)$$

where w_i are the generalized eigenvectors and l_i are the generalized eigenvalues. The magnitude of the largest eigenvalue was *2.18* followed by *0.0055* as the next largest. This implies that a large majority of the differences between image classes is contained in the features indicated by the eigenvector w_1 corresponding to $l_1 = 2.18$. The indicated attributes, meaning those features whose corresponding elements in w_1 have magnitudes significantly greater than zero, were *intensity-mean*, *rawred-mean*, *rawblue-mean*, *rawgreen-mean*, *exred-mean*, *exblue-mean*, and *exgreen-mean*. These features will be hereafter referred to as the MDA features. Given that the intensity and the excess color

values are defined as linear combinations of R, G, and B, we may therefore conclude that average pixel color is the greatest discriminating factor between image segment classes.

III. Classification Methods Applied

Several different classification methods were applied to the image segments. These classifiers included Bayes' decision, *k*-nearest neighbor, probabilistic neural networks [12,13], minimum mean-squared error (MSE) [16], Lagrangian support vector machines (SVM) [14], and *m*-class SVM [15] For each of these approaches, the classification algorithm was first trained using the training data, and the algorithm was then applied to the testing data. Also, for each of these approaches, training and testing was done with the full set of image features and with just the MDA features.

A. Bayes' Decision

The Bayes' decision classifier is proven to give the best performance of any classifier, provided that an accurate definition of each class's posterior probability density function (PDF) is available. As analytical expressions for the PDFs of the image segment feature data is not available, we assumed that the features of each image class were jointly Gaussian. Therefore, in the training phase of this experiment, we computed a sample mean for each class

$$m_j = \frac{1}{N_j} \sum_{i \in Class\ j} x_i \tag{2}$$

and a sample covariance matrix for each class

$$K_j = \frac{1}{N_j} \sum_{i \in Class\ j} (x_i - m_j)(x_i - m_j)^T \tag{3}$$

where $N_j = 30$ is the number of samples in class j. We then removed rows and columns of each K_j until they became non-singular. The posterior PDF of class j may then be written as

$$p_j(x) = \frac{1}{(2p)^{n/2}|K_j|^{1/2}} \exp\left[-\frac{1}{2}(x-m_j)^T K_k^{-1}(x-m_j)\right], \quad (4)$$

and thus the Bayes' decision classifier is

$$D_i = \arg\max_{j \in 1..7} p_j(x_i). \quad (5)$$

This classifier, as described above, was applied to the full set of feature data and to the reduced set of MDA features. The results of these experiments are shown in the confusion matrices of Table 1 for the full set of features and Table 2 for the MDA features. The classifier using the full set of features achieved a probability of correct classification (Pcc) of 0.844, and the classifier using the MDA features achieved a Pcc of 0.845. This indicates that the MDA features capture all of the information relevant to the Bayes' decision classifier, with an assumed Gaussian PDF, and may even offer a slight reduction in noise, by discarding features that are not informative. In both cases, the cement, foliage, path, and window were the most difficult classes to recognize.

Table 1: Confusion matrix using Bayes' decision on the full feature set.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.76	0.00	0.04	0.00	0.13	0.00	0.07
	Brickface	0.00	0.98	0.00	0.00	0.01	0.00	0.01
	Foliage	0.05	0.00	0.58	0.00	0.00	0.00	0.37
	Grass	0.00	0.00	0.00	0.99	0.00	0.00	0.01
	Path	0.13	0.00	0.01	0.00	0.86	0.00	0.01
	Sky	0.01	0.00	0.00	0.00	0.00	0.99	0.00

Table 2: Confusion matrix using Bayes' decision on the MDA features.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.70	0.00	0.01	0.00	0.20	0.00	0.09
	Brickface	0.01	0.98	0.00	0.00	0.00	0.00	0.01
	Foliage	0.02	0.00	0.62	0.00	0.00	0.00	0.36
	Grass	0.00	0.00	0.01	0.98	0.00	0.00	0.00
	Path	0.18	0.00	0.03	0.00	0.79	0.00	0.00
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	Window	0.10	0.00	0.04	0.00	0.01	0.00	0.86

B. *k*-Nearest Neighbor

The *k*-nearest neighbor classifier attempts to estimate optimal decision boundaries by finding the *k* training samples nearest to the test sample, and then classifying the test sample by majority vote of the *k* training samples. In our experiments, the distance between samples was measured using an unweighted Euclidean norm

$$d(x, y) = \sqrt{(x - y)^T (x - y)}, \tag{6}$$

Then, given the *k* training samples closest to the test sample *x*, the decision rule may be written as

$$D_i = \arg\max_{j \in 1..7} Freq_j \{x_1, x_2, \ldots x_k\} \tag{7}$$

where $Freq_j$ measures the number of occurrences of class *j* in the set $\{x_1, x_2, \ldots, x_k\}$. To vary the tightness of the classifier's fit to the training data, we varied the number of nearest neighbors *k* from one to eight. The experimental results, achieved by varying *k*, are shown in Figure 1 for the full feature set and the MDA features. It can be seen in Figure 1 that the best performance is achieved with $k = 1$, the confusion matrices for which are given in Tables 3 and 4. The *k*-nearest neighbor classifier achieved Pcc of 0.854 for the full data set and 0.855 for the MDA features. As for the Bayes' decision classifier, the cement, foliage, path, and window were the most difficult to classify.

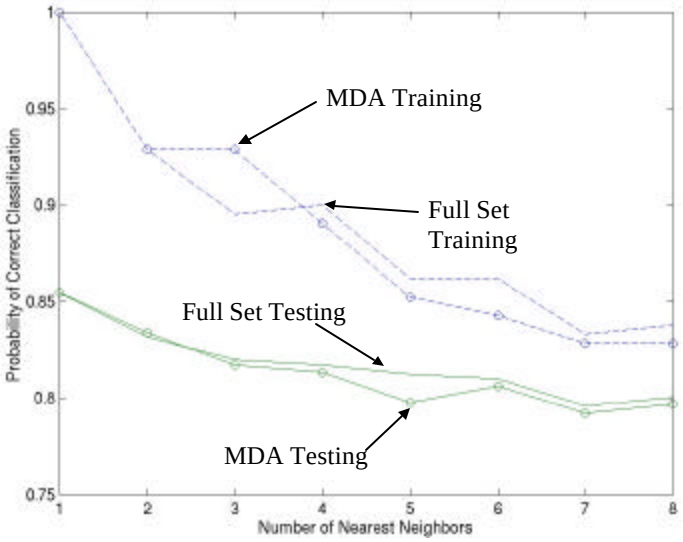

Figure 1: Varying *k* in the nearest neighbor classifier varies the tightness of the classifier's fit to the training data.

Table 3: Confusion matrix using *k* -nearest neighbor classifier (*k* =1) on the full feature set.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.63	0.01	0.02	0.00	0.27	0.00	0.08
	Brickface	0.00	0.99	0.00	0.00	0.00	0.00	0.01
	Foliage	0.02	0.00	0.74	0.00	0.02	0.00	0.23
	Grass	0.01	0.00	0.00	0.99	0.00	0.00	0.00
	Path	0.13	0.00	0.01	0.00	0.85	0.00	0.01
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Table 4: Confusion matrix using *k* -nearest neighbor classifier (*k* =1) on the MDA features.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.65	0.00	0.01	0.00	0.28	0.00	0.06
	Brickface	0.00	0.99	0.00	0.00	0.00	0.00	0.01
	Foliage	0.03	0.00	0.69	0.00	0.02	0.00	0.26
	Grass	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	Path	0.14	0.00	0.00	0.00	0.84	0.00	0.02
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	Window	0.10	0.01	0.08	0.00	0.00	0.00	0.82

C. Probabilistic Neural Network

Similar to the *k*-nearest neighbor classifier, the probabilistic neural network (PNN) classifier attempts to estimate optimal decision boundaries based on the Euclidean distance from the test sample to each training sample. The Matlab PNN is a neural

network in two layers. The first layer computes the distance from the test sample to each training sample. The second layer linearly combines the distances computed for the training samples of each class, to produce a posterior probability score for each class. A compete transfer function then selects the class with the highest posterior probability. The Matlab PNN function allows user control of a "spread" variable, which determines the tightness of the classifier's fit to the training data. The results obtained by varying this spread variable from 0 to 5 are shown in Figure 2. The classifier using the full data set and a spread of 2.8 achieved a Pcc of 0.846, and the classifier using the MDA features and a spread of 0.4 achieved a Pcc of 0.852. These results are shown in the confusion matrices of Tables 5 and 6, and it can be seen that the problematic image classes are consistent with the previous two classifiers.

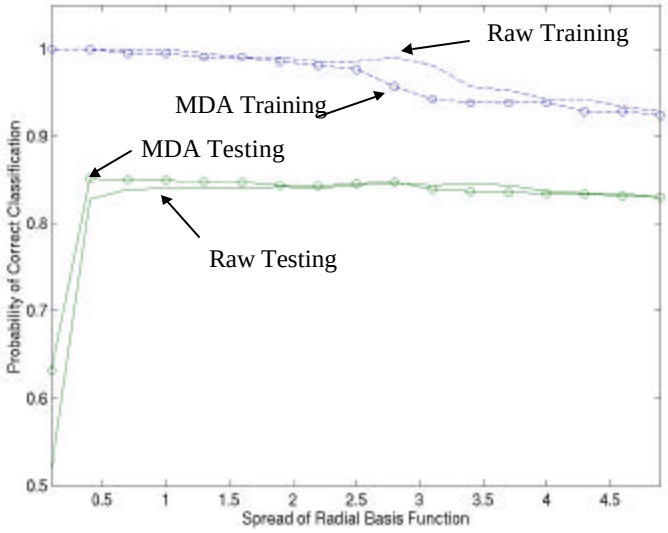

Figure 2: Varying the spread in the PNN classifier varies the tightness of the classifier's fit to the training data.

Table 5: Confusion matrix using PNN classifier (spread = 2.8) on the full feature set

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.64	0.01	0.02	0.00	0.26	0.00	0.08
	Brickface	0.00	0.99	0.00	0.00	0.00	0.00	0.01
	Foliage	0.06	0.00	0.74	0.00	0.02	0.00	0.18
	Grass	0.01	0.00	0.00	0.99	0.00	0.00	0.00
	Path	0.14	0.00	0.01	0.00	0.84	0.00	0.01
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Table 6: Confusion matrix using PNN classifier (spread = 0.4) on the MDA features.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.65	0.00	0.01	0.00	0.28	0.00	0.06
	Brickface	0.00	0.99	0.00	0.00	0.00	0.00	0.01
	Foliage	0.04	0.00	0.69	0.00	0.02	0.00	0.26
	Grass	0.01	0.00	0.00	0.99	0.00	0.00	0.00
	Path	0.14	0.00	0.00	0.00	0.84	0.00	0.02
	Sky	0.01	0.00	0.00	0.00	0.00	0.99	0.00
	Window	0.10	0.01	0.08	0.00	0.00	0.00	0.81

D. Minimum Mean-Squared Error

In a 2-class classification problem, the minimum MSE classifier decision may be expressed as

$$D_i = \begin{cases} 1, & a^T x_i \geq 0 \\ 2, & a^T x_i < 0 \end{cases}. \tag{8}$$

The linear discriminant coefficients a are determined by

$$a = \arg\min_a \left\| Ra - b \right\|^2 = \left(R^T R \right)^{-1} R^T b \tag{9}$$

where R contains the set of training features, such that R_{ij} is feature j of training sample i. The vector b is defined such that b_i is 1, if sample i is from class 1, and is –1, if sample i is from class 2. The least squares solution of (9) assumes that sufficient training samples are available such that $R^T R$ is full rank.

The minimum MSE classifier may be extended to an m-class problem by simply determining m-1 linear discriminants. In which case, each classification is handled sequentially by the decision rule

$$\begin{aligned} &for\ k = 1\ to\ m-1 \\ &\quad D_i = \begin{cases} k, & a_k{}^T x_i \geq 0 \\ \{k+1,\ldots,m\}, & a_k{}^T x_i < 0 \end{cases} \\ &if\ D_i \neq k, then\ k = k+1 \end{aligned} \tag{10}$$

where the order in which the classes are considered is significant. One must consider the clustering of the classes in feature space to decide this order. Classes that lie on the edge of the cluster should be classified first, thus allowing the discriminant of lowest complexity. The classifier may fit the training data more tightly by projecting the data samples into different orders of polynomial space. Figure 3 shows the classification results from varying the projected polynomial order from 1 to 8. The classifier using the full set of features and projecting into second order polynomial space achieved a Pcc of 0.823, and the classifier using the MDA features and projecting into third order polynomial space achieved a Pcc of 0.83. The confusion matrices for these results are given in Tables 7 and 8, where the typical classes remain problematic. It is important to highlight that the minimum MSE classifier is designed by minimizing the MSE, which does not necessarily carry any implications with respect to the probability of correct classification.

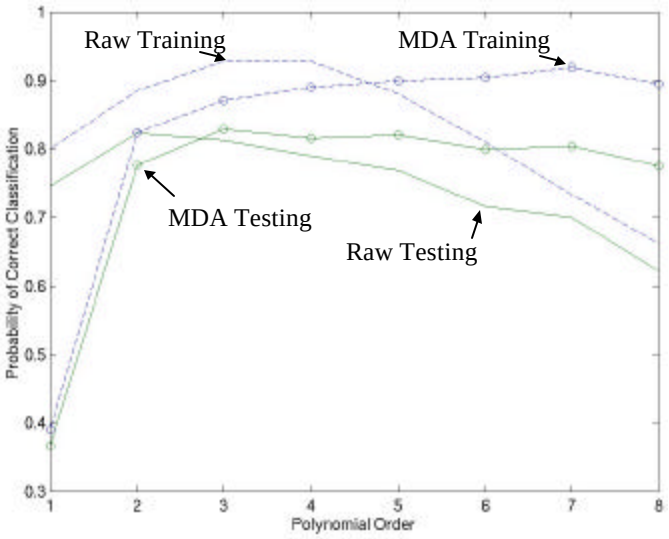

Figure 3: Varying the projected polynomial order in the minimum MSE classifier varies the tightness of the classifier's fit to the training data.

Table 7: Confusion matrix using the 2nd order minimum MSE classifier on the full feature set.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.50	0.00	0.03	0.00	0.35	0.00	0.12
	Brickface	0.00	0.97	0.00	0.00	0.01	0.00	0.02
	Foliage	0.01	0.01	0.79	0.00	0.04	0.02	0.13
	Grass	0.00	0.00	0.00	0.99	0.01	0.00	0.00
	Path	0.15	0.00	0.03	0.00	0.81	0.00	0.00
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Table 8: Confusion matrix using the 3rd order minimum MSE classifier on the MDA features.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.52	0.00	0.02	0.00	0.32	0.00	0.13
	Brickface	0.00	0.97	0.00	0.00	0.01	0.00	0.02
	Foliage	0.02	0.00	0.85	0.01	0.00	0.03	0.08
	Grass	0.00	0.00	0.01	0.99	0.00	0.00	0.00
	Path	0.16	0.00	0.03	0.00	0.77	0.00	0.04
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	Window	0.05	0.00	0.12	0.01	0.00	0.12	0.71

E. Lagrangian Support Vector Machine

Support vector machine classifiers are based on solving a constrained quadratic-programming problem that yields a maximum-margin separating surface between two classes. In [14], Mangasarian and Musicant reformulate the standard SVM programming problem to allow the minimization to be performed on an unconstrained and

differentiable convex function, in a space of dimensionality equal to the number of training samples. In application to our *m*-class image segment classification problem, *m*-1 Lagrangian SVMs were determined using radial basis functions. The class decision was then made sequentially, by considering one class at a time as in (10) for the minimum MSE classifier. The spread of the radial basis functions were varied from 0 to 500, in order to control the tightness of the classifier's fit to the training data. Figure 4 shows the results obtained by varying this spread for the full data set and the MDA features. The full data set classifier achieved a Pcc of 0.743 with a spread of 400, and the MDA feature classifier achieved a Pcc of 0.762 with a spread of 120. The confusion matrices for these results are given in Tables 9 and 10, where again the typical classes remain problematic.

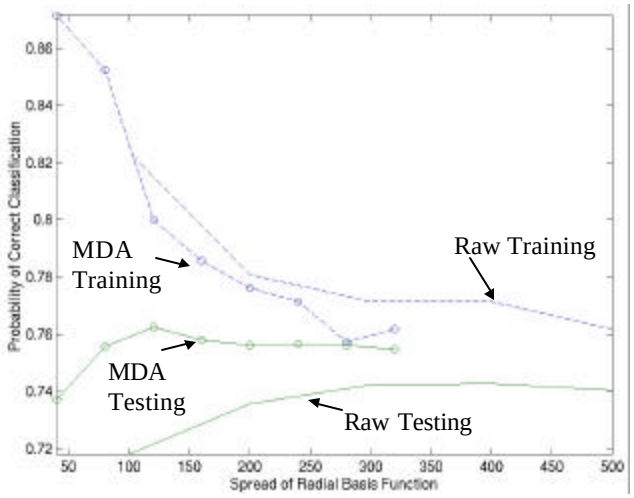

Figure 4: Varying the spread of the radial basis functions in the Lagrangian SVM classifier varies the tightness of the classifier's fit to the training data.

Table 9: Confusion matrix using the Lagrangian SVM classifier (spread=400) on the full feature set.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.19	0.04	0.00	0.00	0.63	0.05	0.08
	Brickface	0.02	0.95	0.02	0.00	0.00	0.00	0.01
	Foliage	0.04	0.01	0.59	0.00	0.02	0.12	0.21
	Grass	0.00	0.01	0.00	0.98	0.00	0.00	0.01
	Path	0.04	0.00	0.00	0.00	0.87	0.09	0.00
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Table 10: Confusion matrix using the Lagrangian SVM classifier (spread=120) on the MDA features.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.19	0.02	0.00	0.00	0.63	0.08	0.08
	Brickface	0.02	0.95	0.00	0.00	0.00	0.00	0.02
	Foliage	0.04	0.00	0.54	0.00	0.05	0.04	0.32
	Grass	0.02	0.00	0.00	0.96	0.00	0.02	0.00
	Path	0.03	0.01	0.00	0.00	0.96	0.00	0.00
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	Window	0.06	0.00	0.18	0.00	0.02	0.00	0.73

F. *m*-Class Support Vector Machine

The *m*-class SVM classifier built by Yi Zhao [15] performs joint *m*-class pattern recognition by solving the constrained quadratic-programming problem necessary for a maximum-margin separating surface. A radial basis function was used as the kernel for the SVM, and the variance of this basis function was varied from 10^{-5} to 10^{-1}, in order to control the tightness of the classifier's fit to the training data. These results are shown in Figure 5 for the full feature set and the MDA features. The full feature set classifier and the MDA features classifier both achieved their best performance with a variance of 10^{-3}, achieving probabilities of correct classification of 0.872 and 0.877, respectively. The confusion matrices for these results are shown in Tables 11 and 12. The cement remained the most difficult class to identify, but the other previously problematic classes did show some improvement. The *m*-class SVM classifier achieved the best performance of all of the classifiers considered.

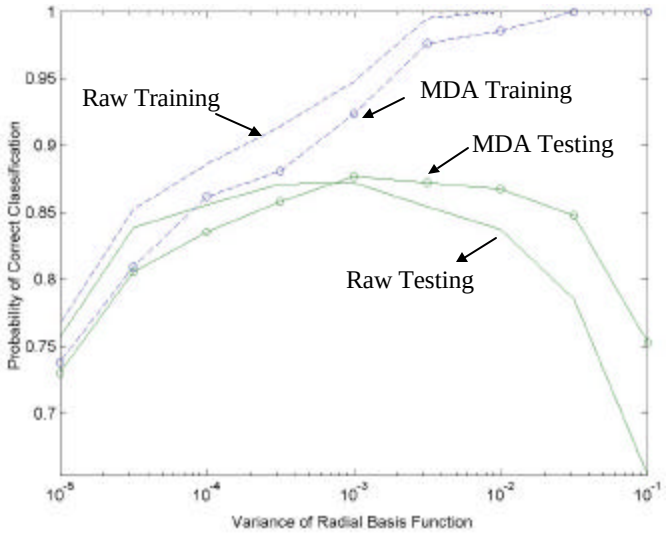

Figure 5: Varying the variance of the radial basis functions in the *m*-class SVM classifier varies the tightness of the classifier's fit to the training data.

Table 11: Confusion matrix using the *m* -class SVM classifier (variance=10^{-3}) on the full feature set.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.63	0.00	0.05	0.00	0.24	0.00	0.09
	Brickface	0.00	0.98	0.00	0.00	0.00	0.00	0.02
	Foliage	0.01	0.00	0.87	0.00	0.01	0.00	0.11
	Grass	0.00	0.00	0.01	0.99	0.00	0.00	0.00
	Path	0.13	0.00	0.05	0.00	0.81	0.00	0.01
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	Window	0.04	0.00	0.13	0.00	0.00	0.00	0.82

Table 12: Confusion matrix using the *m* -class SVM classifier (variance=10^{-3}) on the MDA features.

Probability of Classification		Decision						
		Cement	Brickface	Foliage	Grass	Path	Sky	Window
Truth	Cement	0.60	0.00	0.00	0.00	0.29	0.00	0.11
	Brickface	0.00	0.98	0.00	0.00	0.00	0.00	0.02
	Foliage	0.03	0.00	0.82	0.00	0.01	0.00	0.14
	Grass	0.00	0.00	0.00	0.99	0.00	0.00	0.00
	Path	0.12	0.00	0.00	0.00	0.87	0.00	0.01
	Sky	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	Window	0.04	0.01	0.07	0.00	0.00	0.00	0.88

IV. Conclusions

Based on the results discussed above, the following observation may be made. First, MDA successfully identifies the information rich features of a data set. This conclusion is supported by the fact that the classifiers considered performed as well on

the MDA features as the full data sets, if not slightly better. Second, as the Bayes' decision classifier did not achieve the best performance, we must conclude one or both of the following: 1) a jointly Gaussian PDF does not accurately model the feature data, and 2) the statistics computed from the training data were not precise enough. Third, given that the k-nearest neighbor classifier performed best with $k = 1$, we may conclude that there were not sufficient training samples available to accurately characterize the feature PDFs. Finally, the joint m-class SVM classifier significantly outperformed the m-1 sequential Lagrangian SVM classifiers, indicating the sub-optimal nature of either the Lagrangian SVM or the sequential architecture.

V. References

[1] G.E. Hinton, P. Dayan and M. Revow, "Modeling the Manifolds of Images of Handwritten Digits," *IEEE Trans on Neural Networks*, Vol 8, 65-74 1997.

[2] M. Revow, C.K.I. Williams and G.E. Hinton, "Using generative models for handwritten digit recognition," *IEEE Trans on Pattern Analysis and Machine Intellegince*, Vol 18(6), pp 592-606, 1996.

[3] C.K.I. Williams, M. Revow and G.E. Hinton, "Instantiating deformable models with a neural net," *Computer Vision and Image Understanding*, Vol 68(1) pp 120-126 1997.

[4] M. Revow, C.K.I. Williams and G.E. Hinton. "Using mixtures of deformable models to capture variations in hand printed digits," *Third International workshop on Frontiers in Handwriting Recognition*, Buffalo, USA. pp 142-152, 1993.

[5] G.E. Hinton, C.K.I. Williams and M. Revow. "Adaptive elastic models for hand-printed character recognition," In *NIPS 4*, J.E. Moody, S.J. Hanson and R.P Lippman (eds), 1992.

[6] Z.W. Tu and S.C. Zhu, "Image Segmentation by Data Driven Markov Chain Monte Carlo," *IEEE Trans on Pattern Analysis and Machine Intelligence* , Vol 24. no.5, May 2002.

[7] S. Waterhouse, "Classification and Regression using Mixtures of Experts," Ph.D. dissertation, Cambridge University Engineering Department, Oct. 1997.

[8] Delve web site, http://www.cs.toronto.edu/~delve/, University of Toronto, Department of Computer Science.

[9] Neal, R. M., "Assessing relevance determination methods using DELVE", in C. M. Bishop (editor), *Neural Networks and Machine Learning*, pp. 97-129, Springer-Verlag, 1998.

[10] CART v. 1.1, "Classification And Regression Trees", California Statistical Software, Inc. 961 Yorkshire Ct. Lafayette, California 94549.

[11] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth International Group: Belmont, California.

[12] P.D. Wasserman, Advanced Methods in Neural Computing, New York: Van Nostrand Reinhold, pp. 35-55, 1993.

[13] MATLAB, The MathWorks, Inc., Natick, MA, copyright 1984-2002.

[14] O.L. Mangasarian and D.R. Musicant, "Lagrangian Support Vector Machines," *Journal of Machine Learning Research*, Vol 1, March 2001, 161-177.

[15] Y. Zhao, *m-Class Support Vector Machine Classifier*, IPS Laboratory, Ohio State University, using *LIBSVM* by Chih-Chung Chang and Chih-Jen Lin, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[16] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification 2nd Ed.*, John Wiley & Sons, Inc., New York, NY, 2001.