Classification and Learning for Character Recognition: Comparison of Methods and Remaining Problems

Cheng-Lin Liu
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation, Chinese Academy of Sciences
P.O. Box 2728, Beijing 100080, P.R. China
E-mail: liucl@nlpr.ia.ac.cn

Hiromichi Fujisawa
Central Research Laboratory, Hitachi, Ltd.
1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan
E-mail: fujisawa@crl.hitachi.co.jp

Abstract

Classification methods based on learning from examples have been widely applied to character recognition from the 1990s and have brought forth significant improvements of recognition accuracies. This class of methods includes statistical methods, artificial neural networks, support vector machines, multiple classifier combination, etc. In this paper, we discuss the characteristics of the classification methods that have been successfully applied to character recognition, and show the remaining problems that can be potentially solved by learning methods.

1. Introduction

The methods popularly used in the early stage of OCR (optical character recognition) research and development are template matching and structural analysis [1]. An approach intermediate between them is feature analysis, also called as feature matching. The templates or prototypes in these early methods were either designed artificially, selected or averaged from few samples. As samples increase, these simple design methods are insufficient to accommodate the shape variability of samples, and so, are not able to yield high recognition accuracies. To take full advantage of large sample data, the character recognition community has turned attention to classification methods based on learning-from-examples strategy, especially artificial neural networks (ANNs) from the late 1980s and the 1990s. New learning methods, especially support vector machines (SVMs), are now actively studied and applied in pattern recognition.

Learning methods[1] have benefited character recognition tremendously: they release we engineers from painful job of template selection and tuning, and the recognition accuracies have been promoted significantly because of learning from large sample data. Some excellent results have been reported by, e.g., [2, 3, 4]. Despite the improvements, we are aware that the problem is far from solved: the recognition accuracies of either machine-printed characters on degraded image or freely handwritten characters are insufficient, the existing learning methods do not work well on huge sample data and ever-increasing data, recognition errors cannot be eliminated even if we reject a large percentage of samples, etc. The solution of these remaining problems should still rely on learning: to better utilize knowledge and samples.

In this paper, we discuss the strengths and weaknesses of classification methods that have been widely used, identify the needs of improved performance in character recognition, and suggest some research directions of classification that can help meet with these needs. We will focus on the classification of isolated (segmented) characters, though classification methods are also important for other tasks like layout analysis and segmentation (see [5]). The classification of characters is also important for segmentation, when over-segmentation-based or character-model-based word/string recognition schemes are adopted.

[1] We refer to learning when classifier design is concerned, and refer to classification when the task of recognition is concerned.

2. State of the Art

We mainly discuss feature vector-based classification methods, which have prevailed structural methods, especially in off-line character recognition. These methods include statistical methods, ANNs, SVMs, and multiple classifier combination. After giving an overview of classification methods, we will compare the performance of popular methods on benchmark datasets, and discuss the properties of different types of classification/learning methods.

2.1 Overview of classification methods

We summarize the classification methods in categories of statistical methods, artificial neural networks (ANNs), kernel methods, and multiple classifier combination.

- *Statistical methods.* Statistical classifiers are rooted in the Bayes decision rule, and can be divided into parametric ones and non-parametric ones [6, 7]. Non-parametric methods, such as Parzen window and k-NN rule, are not practical for real-time applications since all training samples are stored and compared. Assuming Gaussian density with various restrictions, the Bayesian discriminant function is reduced to a quadratic discriminant function (QDF), linear discriminant function (LDF), and Euclidean distance from class mean. The regularized discriminant analysis (RDA) method [8] stabilizes the performance of QDF by smoothing the covariance matrices. The modified QDF (MQDF) of Kimura et al. [9] involves less parameters and lower computation than the QDF, and results in improved generalization accuracy. For modeling multi-modal distributions, the mixture of Gaussians in high-dimensional feature space does not necessarily give high classification accuracy, yet the mixture of linear subspaces has shown effects in handwritten character recognition [10, 11].

- *Artificial neural networks.* Feedforward neural networks, including multilayer perceptron (MLP), radial basis function (RBF) network, higher-order neural network (HONN), etc., have been widely applied to pattern recognition. The connecting weights are usually adjusted to minimize the squared error on training samples in supervised learning. Using a modular network for each class was shown to improve the classification accuracy [12]. A network using local connection and shared weights, called convolutional neural network, has reported great success in character recognition [2, 13]. The RBF network can yield competitive accuracy with the MLP when training all parameters by error minimization [14]. The HONN is also called as functional-link network, polynomial network or polynomial classifier (PC). Its complexity can be reduced by dimensionality reduction before polynomial expansion [15] or polynomial term selection [16]. Vector quantization (VQ) networks and auto-association networks, with the sub-net of each class trained independently in unsupervised learning, are also useful for classification. The learning vector quantization (LVQ) of Kohonen [17] is a supervised learning method and can give higher classification accuracy than VQ. Some improvements of LVQ learn prototypes by error minimization instead of heuristic adjustment [18].

- *Kernel methods.* Kernel methods, including support vector machines (SVMs) [19, 20] primarily and kernel principal component analysis (KPCA), kernel Fisher discriminant analysis (KFDA), etc., are receiving increasing attention and have shown superior performance in pattern recognition. An SVM is a binary classifier with discriminant function being the weighted combination of kernel functions over all training samples. After learning by quadratic programming (QP), the samples of non-zero weights are called support vectors (SVs). For multi-class classification, binary SVMs are combined in either one-against-others or one-against-one (pairwise) scheme [21]. Due to the high complexity of training and execution, SVM classifiers have been mostly applied to small category set problems. A strategy to alleviate the computation cost is to use a statistical or neural classifier for selecting two candidate classes, which are then discriminated by SVM [22]. Dong et al. used a one-against-others scheme for large set Chinese character recognition with fast training [23]. They used a coarse classifier for acceleration but the large storage of SVs was not avoided.

- *Multiple classifier combination.* Combining multiple classifiers has been long pursued for improving the accuracy of single classifiers [24]. Parallel (horizontal) combination is more often adopted for high accuracy, while sequential (cascaded, vertical) combination is mainly used for accelerating large category set classification. The decision fusion methods are categorized into abstract-level, rank-level, and measurement-level combination [25, 26]. Many fusion methods have been proposed to measurement-level combination [27, 28]. The complementariness (also called as independence or diversity) of classifiers is important to

yield high combination performance. For character recognition, combining classifiers based on different techniques of pre-processing, feature extraction, and classifier models is effective. Another effective method, called perturbation, uses a single classifier to classify multiple deformations of the input pattern and combine the decisions on multiple deformations [29, 30]. The deformations of training samples can also be used to train the classifier for higher generalization performance [30, 13].

2.2 Performance comparison

The experiments of character recognition reported in the literature vary in many factors such as the sample data, pre-processing technique, feature representation, classifier structure and learning algorithm. Only a few works have compared different classification/learning methods based on the same feature data. In the following, we first cite some high recognition accuracies reported on well-known sample databases, and then summarize some classification results on common feature data.

Handwritten numeral recognition has been most widely tested in pattern classification for its wide applicability and the ease of implementation. Some popular databases are CENPARMI, NIST Special Database 19 (SD19), MNIST, etc. The NIST SD19 contains huge number of character images, but researchers often use different partitions of data for training and testing. We hence collect some results reported on CENPARMI and MNIST databases, which are partitioned into standard training and test sets.

The CENPARMI database contains 4,000 training samples and 2,000 test samples segmented from USPS envelope images. This set was considered difficult, but it is easy to achieve a recognition rate over 98% by extracting statistical features and training classifiers. Suen et al. reported a correct rate 98.85% by training neural networks on 450,000 samples [3]. By training with 4,000 samples, correct rates over 99% have been given by polynomial classifier (PC) and SVMs [4, 31].

The MNIST database contains 60,000 training samples and 10,000 test samples selected from the NIST SD19. Each sample was normalized to a gray-scale image of 20×20 pixels, which is located in a 28×28 plane. LeCun et al. collected a number of test accuracies given by various classifiers [2]. A high accuracy, 99.30%, was given by a boosted convolutional neural network (CNN) trained with distorted data. Simard et al. improved both the distorted sample generation and the implementation of CNN and resulted in a test accuracy 99.60%. Instead of the trainable feature extractors in CNN, extracting heuristically discriminating features also lead to high accuracies. On gradient direction feature, Liu et al. obtained a test accuracy 99.58% by SVM classification, 99.42% by polynomial classifier, and over 99% by many other classifiers [4].

Table 1. A citation of error rates (%) on the MNIST test set.

Feature	pixel	PCA	grad-4	grad-8
k-NN	3.66	3.01	1.26	0.97
MLP	1.91	1.84	0.84	0.60
RBF	2.53	2.21	0.92	0.69
PC	1.64	N/A	0.83	0.58
SVC-poly	1.69	1.43	0.76	0.55
SVC-rbf	1.41	1.24	0.67	0.42

On the MNIST database, training classifiers without feature extraction performs inferiorly to that with feature extraction. A better scheme to compare classifiers is to train them on a common feature representation. Holmström et al. compared various statistical and neural classifiers on PCA features [32]. However, the PCA feature neither performs sufficiently. In the comparison studies of Liu et al. [33, 4], the features used, chaincode and gradient direction features, are widely accepted in practice. Their results show that parametric statistical classifiers (especially the MQDF) generalize better than neural classifiers when training with small sample data, while neural classifiers outperforms when training with large sample data. The SVM classifier with RBF kernel mostly gives the highest accuracy. The best neural classifier was shown to be the polynomial classifier (PC), and the RBF classifier was shown to perform comparably with or better than the MLP. A citation of error rates from [4] is shown in Table 1, where "grad-4" and "grad-4" stand for 4-orientation and 8-direction gradient features, respectively.

In the area of Chinese/Japanese character recognition, a handprinted database ETL9B has been widely tested. This database contains 200 samples for each of 3,036 classes. High accuracies have been reported by using quadratic classifiers and SVMs. Kimura et al. tested on 40 samples per class in rotation and reported average rate 99.15% by using modified Bayes discriminant function on enhanced training samples [34]. Kato et al. tested on 20 samples per class in rotation, and used partial inclination detection for improving normalization [35]. They reported a correct rate 99.42% by using asymmetric Mahalanobis distance for fine classification. Dong et al. tested on 40 samples per class and reported a correct rate 99.00% by using SVMs

trained on enhanced samples for fine classification [23]. In these works, the techniques of pre-processing and feature extraction and the option of distorted sample generation affect the recognition accuracy.

2.3 Statistical vs. discriminative classifiers

We refer to discriminative classifiers (also called margin-based classifiers) as those based on minimum (regression or classification) error training, including neural networks and SVMs, for which the parameters of one class are trained on the samples of all classes. For statistical classifiers, the parameters of one class are estimated from the samples of its own class only. We compare the characteristics of two kinds of classifiers in the following respects.

- *Complexity and flexibility of training.* The training time of statistical classifiers is linear with the number of classes, and it is easy to add a new class to an existing classifier. Also, adapting the density parameters of a class to new samples is possible. In contrast, the training time of discriminative classifiers is proportional to square of the number of classes, and to guarantee the stability of parameters, adding new classes or new samples need re-training with all samples.

- *Classification accuracy.* When training with enough samples, discriminative classifiers give higher accuracies than statistical classifiers. The accuracy of regularized statistical classifiers (like MQDF and RDA) are more stable against the training sample size (see [33]). On small sample size, statistical classifiers can generalize better than discriminative ones.

- *Storage and execution complexity.* At same level of classification accuracy, discriminative classifiers tend to have fewer parameters than statistical classifiers, and so, they are less expensive in storage and execution.

- *Confidence of decision.* The discriminant functions of parametric statistical classifiers are connected to the class conditional probability, and can be easily converted to a posteriori probabilities by the Bayes formula. In contrast, the outputs of discriminative classifiers are directly connected to a posteriori probabilities.

- *Rejection capability.* Classifiers of higher classification accuracies tend to reject ambiguous patterns better, but not necessarily reject well outliers (patterns out of defined classes) [33]. Parametric statistical classifiers are resistant to outliers, whereas discriminative classifiers are susceptible to outliers because their decision regions tend to be open [36]. The rejection capability of discriminative classifiers can be enhanced by training with outlier samples.

2.4 Neural networks vs. SVMs

Neural classifiers and SVMs show different properties in the following respects.

- *Complexity of training.* The parameters of neural classifiers are generally adjusted by gradient descent. By feeding the training samples a fixed number of sweeps, the training time is linear with the number of samples. SVMs are trained by quadratic programming (QP), and the training time is generally proportional to the square of number of samples. Some fast SVM training algorithms with nearly linear complexity are available, however.

- *Flexibility of training.* The parameters of neural classifiers can be adjusted in string-level or layout-level training by gradient descent with the aim of optimizing the global performance [2, 37]. In this case, the neural classifier is embedded in the string or layout recognizer for character recognition. On the other hand, SVMs can only be trained at the level of holistic patterns.

- *Model selection.* The generalization performance of neural classifiers is sensitive to the size of structure, and the selection of an appropriate structure relies on cross-validation. The convergence of neural network training suffers from local minima of error surface. On the other hand, the QP learning of SVMs guarantees finding the global optimum. The performance of SVMs depends on the selection of kernel type and kernel parameters, but this dependence is less influential.

- *Classification accuracy.* SVMs have been demonstrated superior classification accuracies to neural classifiers in many experiments.

- *Storage and execution complexity.* SVM learning by QP often results in a large number of SVs, which should be stored and computed in classification. Neural classifiers have much less parameters, and the number of parameters is easy to control. In a word, neural classifiers consume less storage and computation than SVMs.

3. Remaining Problems and Future Works

Though tremendous advances have been achieved in applying classification and learning methods to character recognition, there is still a gap between the needs of applications and the actual performance, and some problems encountered in practice have not been considered seriously. We discuss the future works of classification and learning that can potentially solve or alleviate these problems.

3.1 Improvements of accuracy

Accuracies lower than 90% are often reported to difficult cases like unconstrained cursive script recognition. Improved accuracy is always desired. It can be achieved via elaborating every processing task: preprocessing, feature extraction, sample generation, classifier design, multiple classifier combination, etc. We hereof only discuss some issues related to classification and learning.

- *Feature transformation and selection.* Feature transformation methods, including PCA and FDA, have been proven effective in pattern classification, but no method claims to find the best feature subspace. Generalized transformation methods based on relaxed density assumptions and those based on discriminative learning are expected to find better feature spaces. On the other hand, we can extract a large number of features, and automatic feature selection may lead to better classification than artificial selection.

- *Sample generation and selection.* Training with distorted samples has resulted in improved generalization performance, but better methods of distorted sample generation are yet to be found. Sample selection from very large data set is important to guarantee the efficiency and quality of training.

- *Joint feature selection and classifier design.* To select features and design classifier jointly may lead to better classification performance. The Bayesian network belongs to such kind of classifiers and is now being studies intensively.

- *Hybrid statistical/discriminative learning.* A hybrid statistical/discriminative classifier may yield higher accuracy than both the pure statistical and the pure discriminative classifier [38]. A way to design such classifiers is to adjust the parameters of parametric statistical classifiers discriminatively on training samples [39, 40]. Also, combining the decisions of statistical and discriminative classifiers is preferred to combining similar classifiers.

- *Ensemble learning.* The performance of combining multiple classifiers primarily relies on the complementariness of classifiers. Maximizing the diversity of classifiers is now receiving increasing attention. A heuristic is to generate classifiers with different properties at various processing steps. Among the methods that explore the diversity of data, the Boosting is considered as the best ensemble classifier. It has not been widely tested in character recognition yet.

3.2 Reliable confidence and rejection

It is desirable to reject or delay the decision for those patterns with low confidence. There maybe two kinds of confidence measures: class conditional probability-like (conditional confidence, on which outlier rejection is based) and posterior probability-like (posterior confidence, on which ambiguity rejection is based). Both confidence measures can be unified into the posterior probabilities for open world: defined classes plus an outside class. Transforming classifier outputs to probability measures facilitates contextual processing which integrates information from multiple sources. The following ways may help improve the rejection capability of current recognition methods.

- *Elaborate density estimation.* Probability density estimation is a traditional problem in statistical pattern recognition, but is not well-solved yet. The Gaussian mixture model is being studies intensively. For density estimation in high-dimensional spaces, combining feature transformation or selection may result in good classification performance. Density estimation in kernel space would be a choice to explore nonlinear subspace.

- *One-class classification.* One-class classifiers separate one class from the remaining world with parameters estimated from the samples of the target class only. Using one-class classifiers as class verifiers added to a multi-class classifier can improve rejection. The distribution of a class can be described by a good density model or support vectors in kernel space [41]. Structural analysis of character patterns may serve as good verifiers.

- *Hybrid statistical/discriminative learning.* Hybrid statistical/discriminative classifiers have good resistance to outliers. This principle is yet to be

extended to other statistical models than Gaussian discriminant function and may be combined with feature transformation.

- *Multiple classifier combination.* Different classifiers tend to disagree on ambiguous patterns, so the combination of multiple classifiers can better identify and reject ambiguous patterns [42]. Generally, combining complementary classifiers can improve the classification accuracy and the tradeoff between error rate and reject rate.

3.3 Incremental learning

Usually, classifiers are trained off-line and are fixed unchanged in execution. For continuously improving the classification performance, adapting classifiers to new samples faced in applications is desired [43]. In many cases, it is hard to store all the samples ever accumulated for re-training the classifier. Incremental learning for adapting existing classifier to new classes and new samples without the need of re-training with all samples has rarely been considered in character recognition. Some published works of incremental learning in the neural networks community can be referred to for our consideration.

Training with unlabeled data is another topic that is intensively studied in machine learning, called semi-supervised learning. This learning scheme is applicable to character recognition to utilize more unlabeled samples in practice for improving generalization.

3.4 Benchmarking of methods.

Fair comparison of classifiers is difficult because many classifiers, especially neural networks, are flexible in implementation and their performance are affected by human factors [44]. In the character recognition field, the comparison of methods is more difficult because many processing steps (pre-processing, feature extraction, classification) are involved. Even on experiments using the same training and test data, researchers often compare the performance at system level: the final recognition rate by integrating all techniques. It is hard to decide what method at which step is the most influential.

To conduct fair comparison of methods other than systems, we suggest to use standard techniques for each step except the step the method to compare is applied. For example, to compare classifiers, standard pre-processing and feature extraction techniques should be applied to all the classifiers to compare. Many techniques, e.g., nonlinear normalization and direction feature extraction, are variable in implementation details. Hence, we suggest to open source codes of standard techniques for every processing step of character recognition, such that researchers can fairly compare the methods of a special step. For benchmarking classification methods, we can alternatively release standard feature data other than image data.

References

[1] S. Mori, C.Y. Suen, K.Yamamoto, Historical review of OCR research and development, *Proc. IEEE*, 80(7): 1029-1058, 1992.

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE*, 86(11): 2278-2324, 1998.

[3] C.Y. Suen, K. Kiu, N.W. Strathy, Sorting and recognizing cheques and financial documents, *Document Analysis Systems: Theory and Practice*, S.-W. Lee and Y. Nakano (eds.), LNCS 1655, Springer, 1999, pp. 173-187.

[4] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: benchmarking of state-of-the-art techniques, *Pattern Recognition*, 36(10): 2271-2285, 2003.

[5] S. Marinai, M. Gori, G. Soda, Artificial neural networks for document analysis and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(1): 23-35, 2005.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, Academic Press, 1990.

[7] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, second edition, Wiley Interscience, 2001.

[8] J.H. Friedman, Regularized discriminant analysis, *J. Am. Statist. Ass.*, 84(405): 165-175, 1989.

[9] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(1): 149-153, 1987.

[10] G.E. Hinton, P. Dayan, M. Revow, Modeling the manifolds of images of handwritten digits, *IEEE Trans. Neural Networks*, 8(1): 65-74, 1997.

[11] H.-C. Kim, D. Kim, S.Y. Bang, A numeral character recognition using the PCA mixture model, *Pattern Recognition Letters*, 23: 103-111, 2002.

[12] I.-S. Oh, C.Y. Suen, A class-modular feedforward neural network for handwriting recognition, *Pattern Recognition*, 35(1): 229-244, 2002.

[13] P.Y. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis, *Proc. 7th ICDAR*, Edinburgh, UK, 2003, Vol.2, pp.958-962.

[14] C.M. Bishop, *Neural Networks for Pattern Recognition*, Claderon Press, Oxford, 1995.

[15] U. Kreßel, J. Schürmann, Pattern classification techniques based on function approximation, *Handbook of Character Recognition and Document Image Analysis*, H. Bunke and P.S.P. Wang (eds.), World Scientific, 1997, pp.49-78.

[16] J. Franke, Isolated handprinted digit recognition, *Handbook of Character Recognition and Document Image Analysis*, H. Bunke and P.S.P. Wang (eds.), World Scientific, 1997, pp.103-121.

[17] T. Kohonen, The self-organizing map, *Proc. IEEE*, 78(9): 1464-1480, 1990.

[18] C.-L. Liu, M. Nakagawa, Evaluation of prototype learning algorithms for nearest neighbor classifier in application to handwritten character recognition, *Pattern Recognition*, 34(3): 601-615, 2001.

[19] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New Work, 1995.

[20] C.J.C. Burges. A tutorial on support vector machines for pattern recognition, *Knowledge Discovery and Data Mining*, 2(2): 1-43, 1998.

[21] U. Kressel, Pairwise classification and support vector machines, *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C.J.C. Burges, A.J. Smola (eds.), MIT Press, 1999, pp.255-268.

[22] A. Bellili, M. Gilloux, P. Gallinari, An MLP-SVM combination architecture for offline handwritten digit recognition: reduction of recognition errors by support vector machines rejection mechanisms, *Int. J. Document Analysis and Recognition*, 5(4): 244-252, 2003.

[23] J.X. Dong, A. Krzyzak, C.Y. Suen, High accuracy handwritten Chinese character recognition using support vector machine, *Proc. Int. Workshop on Artificial Neural Networks for Pattern Recognition*, Florence, Italy, 2003.

[24] A.F.R. Rahman, M.C. Fairhurst, Multiple classifier decision combination strategies for character recognition: a review, *Int. J. Document Analysis and Recognition*, 5(4): 166-194, 2003.

[25] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, *IEEE Trans. System Man Cybernet.*, 22(3): 418-435, 1992.

[26] C.Y. Suen, L. Lam, Multiple classifier combination methodologies for different output levels, *Multiple Classifier Systems*, J. Kittler and F. Roli (eds.), LNCS 1857, Springer, 2000, pp.52-66.

[27] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3): 226-239, 1998.

[28] R.P.W. Duin, The combining classifiers: to train or not to train, *Proc. 16th ICPR*, Quebec, Canada, 2002, Vol.2, pp.765-770.

[29] T. Ha, H. Bunke, Off-line handwritten numeral recognition by perturbation method, *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5): 535-539, 1997.

[30] J. Dahmen, D. Keysers, H. Ney, Combined classification of handwritten digits using the virtual test sample method, *Multiple Classifier Systems*, J. Kittler and F. Roli (eds.), LNCS 2096, Springer, 2001, pp.99-108.

[31] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction techniques, *Pattern Recognition*, 37(2): 265-279, 2004.

[32] L. Holmström, P. Koistinen, J. Laaksonen, E. Oja, Neural and statistical classifiers—taxonomy and two case studies, *IEEE Trans. Neural Networks*, 8(1): 5-17, 1997.

[33] C.-L. Liu, H. Sako, H. Fujisawa, Performance evaluation of pattern classifiers for handwritten character recognition, *Int. J. Document Analysis and Recognition*, 4(3): 191-204, 2002.

[34] F. Kimura, T. Wakabayashi, S. Tsuruoka, Y. Miyake, Improvement of handwritten Japanese character recognition using weighted direction code histogram, *Pattern Recognition*, 30(8): 1329-1337, 1997.

[35] N. Kato, M. Suzuki, S. Omachi, H. Aso, Y. Nemoto, A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance, *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(3): 258-262, 1999.

[36] M. Gori and F. Scarselli, Are multilayer perceptrons adequate for pattern recognition and verification? *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11): 1121-1132, 1998.

[37] C.-L. Liu, K. Marukawa, Handwritten numeral string recognition: character-level training vs. string-level training, *Proc. 17th ICPR*, Cambridge, UK, 2004, Vol.1, pp.405-408.

[38] R. Raina, Y. Shen, A.Y. Ng, A. McCallum, Classification with hybrid generative/discriminative models, *Advances in Neural Information Processing System 16*, 2003.

[39] J. Dahmen, R. Schluter, H. Ney, Discriminative training of Gaussian mixtures for image object recognition, *Proc. 21st Symposium of German Association for Pattern Recognition*, Bonn, Germany, 1999, pp.205-212.

[40] C.-L. Liu, H. Sako, H. Fujisawa, Discriminative learning quadratic discriminant function for handwriting recognition, *IEEE Trans. Neural Networks*, 15(2): 430-444, 2004.

[41] D.M.J. Tax, R.P.W. Duin, Support vector data description, *Machine Learning*, 54(1): 45-66, 2004.

[42] C.Y. Suen, J. Tan, Analysis of error of handwritten digits made by a multitude of classifiers, *Pattern Recognition Letters*, 26(3): 369-379, 2005.

[43] U. Miletzki, T. Bayer, H. Schafer, Continuous learning systems: postal address readers with built-in learning capability, *Proc. 5th ICDAR*, Bangalore, India, 1999, pp.329-332.

[44] R.P.W. Duin, A note on comparing classifiers, *Pattern Recognition Letters*, 17: 529-536, 1996.