Classification and Learning Methods for Character Recognition: Advances and Remaining Problems

Cheng-Lin Liu[1] and Hiromichi Fujisawa[2]

[1] National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, Beijing 100080, P.R. China
liucl@nlpr.ia.ac.cn

[2] Central Research Laboratory, Hitachi Ltd., 1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan
hiromichi.fujisawa.sb@hitachi.com

Summary. Pattern 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 kind of methods include statistical methods, artificial neural networks, support vector machines, multiple classifier combination, etc. In this chapter, we briefly review the learning-based classification methods that have been successfully applied to character recognition, with a special section devoted to the classification of large category set. We then discuss the characteristics of these methods, and discuss the remaining problems in character recognition that can be potentially solved by machine 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 intermediate approach between them is feature analysis, also referred to as feature matching. The templates or prototypes in these early methods were either designed artificially, selected or averaged from few samples. As the number of samples increases, these simple methods are insufficient to accommodate the shape variability of samples, and so, are not able to yield high recognition accuracy. To take full advantage of large sample data, the character recognition community turned attention to learning-based classification methods, especially artificial neural networks (ANNs) from the late 1980s and the 1990s. Due to the close connection between ANNs and statistical pattern recognition, statistical classification methods are also considered seriously from then. Meanwhile, the research activities in pattern recognition and machine learning communities are becoming close to each other. New

C.-L. Liu and H. Fujisawa: *Classification and Learning Methods for Character Recognition: Advances and Remaining Problems*, Studies in Computational Intelligence (SCI) **90**, 139–161 (2008)
www.springerlink.com

learning methods, especially support vector machines (SVMs, and more generally, kernel methods) and ensemble methods (multiple classifier systems), are now actively studied and applied in pattern recognition.

Learning methods[3] have benefited character recognition tremendously: they release engineers from painful job of template selection and tuning, and the recognition accuracies have been improved significantly because of learning from large sample data. Some excellent results have been reported by, e.g. [2, 3, 4]. While enjoying the benefits of learning-from-examples, we are aware that the problem is far from being 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 large category set; 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 chapter, we first give a brief survey of classification methods in character recognition. A special section is devoted to the classification of large category set. We then discuss the strengths and weaknesses of these methods, identify the needs of improved performance in character recognition, and suggest some research directions of pattern 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. When we discuss classification, it is assumed that pre-processing and feature extraction procedures have been performed appropriately.

2 Brief Survey of Classification Methods

The classification methods for character recognition can be roughly categorized into feature-vector-based methods and structural methods. Feature-vector-based methods are prevailing, especially in off-line character recognition, because of their simple implementation and low computational complexity. Whereas a feature vector can be easily extracted from character images, for structural methods, the extraction of components or strokes are rather difficult. Meanwhile, there is not an off-the-shelf method for learning structural models from examples. Hence, we mainly discuss feature-vector-based methods, including statistical classification methods, ANNs, SVMs, and multiple classifier combination. A comprehensive survey of classification methods has been given by Jain et al. [6]. Statistical methods and ANNs are systematically

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

treated by Fukunaga [7] and Bishop [8], respectively. The textbook of Duda et al. [9] emphasizes statistical methods, but covers other methods as well. In the following, we briefly review the methods that have been successfully applied to character recognition.

2.1 Statistical Methods

Statistical classification methods are rooted in the Bayes decision rule. In the case of 0-1 loss, the input pattern is classified to the class of maximum a posteriori (MAP) probability, which is computed by the Bayes formula from the a priori probability (usually assumed equal for defined classes) and the conditional probability density. Statistical classifiers are divided into parametric ones and non-parametric ones depending on the probability density estimation approach. Parametric classifiers assume for each class a known form of density function, usually a Gaussian function, with unknown parameters estimated on training samples by maximum likelihood (ML). Non-parametric classifiers approximate arbitrary density functions by interpolating the local densities of training samples (Parzen window), or estimate the a posteriori probabilities directly from samples (k-nearest neighbor (k-NN)). Non-parametric methods are expensive in both storage space and execution, however. Though parametric methods assume restrictive density functions, they perform fairly well for practical problems.

When assuming Gaussian density and equal a priori probabilities, the Bayesian discriminant function is equivalent to a quadratic discriminant function (QDF), which is often taken as a standard classifier in benchmarking. When further assuming that the Gaussian density functions of all classes share a common covariance matrix, the QDF is reduced to a linear discriminant function (LDF). If more restrictively, the conditional density function is spherical Gaussian with equal variance, the discriminant function is reduced to the Euclidean distance from class mean, which was often taken in early feature matching methods. The QDF does not necessarily outperform the LDF because it has as many parameters as square of feature dimensionality, and so, is sensitive to the training sample size. The regularized discriminant analysis (RDA) method [10] alleviates this problem by smoothing the covariance matrices. On the other hand, Kimura et al. replace the minor eigenvalues of covariance matrix of each class with a constant [11]. The resulting modified quadratic discriminant function (MQDF) involves less parameters and lower computation than the QDF, and results in improved generalization accuracy (accuracy on un-trained samples). The MQDF is popularly used, especially in handwritten Chinese/Japanese character recognition. An improvement of MQDF with elaborate parameter estimation is called as modified Bayes discriminant function (MBDF) [12]. Another method that is often referred is the projection distance [13], in which the distance of input pattern from a linear subspace of each class serves a reasonable discriminant function. An

improvement, called modified projection distance (MPD), has a functional form similar to the MQDF [14].

Other than regularizing Gaussian density, the Gaussian mixture model (mixture of Gaussians) can model multi-modal distributions. For high-dimensional feature space as is the case of character recognition, however, it does not generalize well. Using a mixture of low-dimensional linear subspaces lowers the complexity of Gaussian mixture while maintaining the multi-modal nature, and the classification performance can be largely improved, as have been demonstrated in handwritten numeral recognition [15, 16].

Under the umbrella of statistical pattern recognition are also feature selection and transformation methods. Feature transformation can reduce the dimensionality of feature space and often improve the classification accuracy. Principal component analysis (PCA) and Fisher discriminant analysis (FDA) are two linear subspace learning methods that have been popularly used. PCA is effective mainly in the recognition of small character set, whereas for large character set, FDA is more efficient [12, 17]. Heteroscedastic discriminant analysis and nonlinear dimensionality reduction have been actively studied in pattern recognition, but are rarely applied to pratical character recognition.

Feature selection is also an active research field in pattern recognition and machine learning. It benefits recognition when there is a large number of features containing redundant and/or noisy ones. Extracting various types of features followed by feature subset selection may yield higher performance than classification on the original feature set or an artificially selected subset. Promising results of handwritten digit recognition using feature subset selection have been reported in [18].

Both parametric and non-parametric classifiers estimate the density parameters of each class independently without considering the separability of different classes. Some methods have been proposed to improve the classification accuracy by modifying the parameters according to the recognition errors on training samples made by the ML classifier, like the LDA method [19] and the mirror image learning method [20]. Parameter optimization methods by error minimization will be reviewed in the context of neural networks.

2.2 Artificial Neural Networks

The connecting weights of artificial neural networks (ANNs) are adjustable to fit an objective of functional approximation, e.g. minimum regression error. Feedforward neural networks, including single-layer perceptron (SLP), multilayer perceptron (MLP), radial basis function (RBF) network, higher-order neural network (HONNs), etc., have been widely applied to pattern recognition. Usually, each output node of the network corresponds to a class, and the maximum output gives the decision of classification. The connecting weights are usually adjusted to minimize the square error between the outputs and target values on training samples (supervised learning). The minimum square

error training algorithm for MLP is referred to as back-propagation (BP) in particular. /indexback-propagation

The descriptions of supervised learning, SLP, MLP, and RBF network can be found in most neural networks textbooks. The output of SLP can be viewed as a linear discriminant function, with the weights estimated by error minimization instead of maximum lieklihood (ML), and so, the SLP often gives higher classification accuracy than the LDF with ML estimation. The MLP is flexible to approximate nonlinear functions and capable of separating patterns of complicated distributions. This power makes it a popular tool for pattern classification. Many works in character recognition have taken the MLP as a standard classifier or benchmark. The generalization performance of MLP can be improved by weight decay, local connection (local receptive fields), weight sharing, structure selection and stopping by cross-validation, etc. A network using local connection and shared weights, called convolutional neural network, has reported great success in character recognition [2, 21]. It directly works on character image and the hidden nodes with local connection can be viewed as trainable feature extractors. For feature vector-based classification, using a modular network for each class can also improve the accuracy [22].

The RBF network has one hidden layer of Gaussian functions, which are combined linearly by the output nodes. In early stage, the parameters of RBF networks were usually estimated in two phases: Gaussian parameter estimation by clustering and weight learning by error minimization. Since the clustering procedure does not consider the speparability of patterns, the Gaussian parameters learned this way do not lead to good classification performance. A substantial improvement is to adjust all the parameters simultaneously by error minimization [8]. This makes the RBF network competitive with the MLP in classification accuracy.

The HONN is also referred to as functional-link network [23], polynomial network, or polynomial classifier [24]. Its output is a weighted combination of pattern features and their polynomial expansions. For high-dimensional features, the number of (even 2nd-order) polynomial terms is extremely large. This complexity can be reduced by dimensionality reduction before polynomial expansion [25] or polynomial term selection [26]. A recently proposed class-specific feature polynomial classifier (CFPC) improves the classification accuracy by polynomial expansion on class-specific linear subspaces [27].

Some unsupervised learning methods have also been applied to pattern recognition, among them are competitive learning for vector quantization (VQ, can be used for learning prototypes for each class) and auto-association network (an application to character recognition can be seen in [28]). On the other hand, Zhang et al. learn mixtures of linear subspaces using neural networks for classification [29].

The learning vector quantization (LVQ) algorithm of Kohonen [30] learns class prototypes with the aim of separating the samples of different classes. LVQ is a supervised learning method and can give higher classification accuracy than VQ. We view VQ and LVQ as neural-like methods because like

neural networks, the parameters (prototypes) are adjusted in online mode (stochastic gradient descent, iteratively on training samples). Some improvements of LVQ learn prototypes by minimizing classification or regression error instead of heuristic adjustment [31].

The discriminative learning quadratic discriminant function (DLQDF) [32] can also be viewed as a neural-like classifier. The DLQDF inherits the structure and initial parameters from the MQDF, but the parameters are optimized on training samples by minimizing the classification error by stochastic gradient descent. In experiments of handwritten numeral recognition, the DLQDF was shown to outperform most statistical and neural classifiers.

2.3 Kernel Methods

Kernel methods, including support vector machines (SVMs) [33, 34] primarily and kernel PCA, kernel FDA, etc., are receiving increasing attention and have shown superior performance in pattern recognition. Kernel methods use a kernel function to represent the inner product of two patterns in expanded nonlinear feature space (possibly of infinite dimensionality). Both training and classification are performed via the kernel function without explicit access of the nonlinear space. An SVM is a binary classifier with discriminant function being the weighted combination of kernel functions over all training samples. The weights (coefficients) are learned by quadratic programming (QP) with the aim of maximizing the margin in feature space. After learning, the samples of non-zero weights are called support vectors (SVs), which are stored and used in classification. The maximal margin criterion of SVM learning leads to good generalization performance, but the resulting large number of SVs brings about heavy storage and computation in classification.

For multi-class classification, binary SVMs can be combined in two ways: one-versus-all (one-against-others) or one-versus-one (pairwise). The pairwise combination scheme was shown to outperform one-versus-all when using linear kernel [35]. When nonlinear kernels are used, the one-versus-all scheme performs sufficiently. In recent years, many results of character recognition using SVM classification have been reported, mostly for small category set problems like numeral recognition. The results (e.g. [4]) show that SVMs indeed yield higher accuracies than statistical and neural classifiers, but the storage and computation of large number of SVs are expensive. 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 [36]. Dong et al. used a one-versus-all scheme for large set Chinese character recognition with fast training [37]. They speed up the recognition by using a coarse classifier for candidate selection, but cannot avoid the problem of storing large number of SVs.

2.4 Multiple Classifier Systems

Combining multiple classifiers has been long pursued for improving the accuracy of single classifiers [38, 39]. Rahman et al. give a survey of combination methods in character recognition, including various structures of classifier organization [40]. Moreover, other chapters of this book are dedicated to this subject. Parallel (horizontal) combination is more often adopted for high accuracy, while sequential (cascaded, vertical) combination is mainly used for accelerating large category set classification. According to the information level of classifier outputs, the decision fusion methods for parallel combination are categorized into abstract-level, rank-level, and measurement-level combination. Measurement-level combination takes full advantage of output information, and many fusion methods have been proposed to it [41, 42, 43]. Some character recognition results using multiple classifiers combined at different levels are reported by Suen and Lam [44].

The classification performance of multiple classifiers not only depends on the fusion strategy, but also relies on the complementariness (also referred to as independence or diversity) of the classifiers. Complementariness can be yielded by varying training samples, pattern features, classifier structure, learning methods, etc. In recently years, methods for generating multiple classifiers (called an ensemble) by exploring the diversity of training samples based on a given feature representation are receiving high attention, among them are the Bagging [45] and the Boosting [46]. For character recognition, combining classifiers based on different pre-processing and feature extraction techniques is effective. Yet another effective method uses a single classifier to classify multiple deformations (called perturbations or virtual test samples) of the input pattern and combine the decisions on multiple deformations [47, 48]. The deformations of training samples can also be used to train the classifier for improving the generalization performance [48, 21].

3 Strategies for Large Category Set

Unlike numerals and English letters that have only tens of classes, the character sets of some oriental languages, like Chinese, Japanese, and Korean, have thousands of daily-used characters. A standard of Chinese, GB2312-80, contains 3,755 characters in the level-1 set and 3,008 characters in the level-2 set, 6,763 in total. A general-purpose Chinese character recognition system needs to deal with an even larger set because those not-often-used characters should be recognized as well.

For classifying a large category set, many classifiers become infeasible because either the training time or the classification time becomes unacceptably long. Classifiers based on discriminative supervised learning (called discriminative classifiers hereof), like ANNs and SVMs, are rarely used to directly classify a large category set. Two divide-and-conquer schemes are often used

to accelerate classification. In one scheme, a simple and fast classifier is used to select a dynamic subset from the whole class set such that the input pattern belongs to the subset with high probability. In another scheme, the class set is divided into static (possibly overlapping) clusters and the input pattern is assigned to one or several clusters, whose unification gives the subset of classes for further discrimination. A hierarchical classification method using both schemes was reported in [49]. Tree classifiers were ever pursued for fast classification of large character set (e.g. [50]) but the accumulation of error along hierarchies makes them insufficient in accuracy, especially for recognizing handwritten characters.

In divide-and-conquer schemes, the second-stage classifier for discriminating a subset of classes (called fine classifier) can be a quadratic classifier or a discriminative classifier. The main advantage of quadratic classifiers is that the parameters of each class are estimated independently using the samples of one class only. The training time is hence linear with the number of classes (NoC). Successful quadratic classifiers include the MQDF of Kimura et al. [11, 12] and some modifications of Mahalanobis distance, which have lower complexity and yield higher accuracy than the original QDF. A further improvement is the compound discriminant functions [51, 14], which discriminate pairs of confusing classes without extra parameters compared to the baseline quadratic classifier. The asymmetric Mahalanobis distance of Kato et al. [52] yields superior recognition accuracy, though with higher complexity than the MQDF.

The training time for a discriminative classifier is square of the NoC since the total number of samples is linear with the NoC, and each sample is used for training the parameters of all classes. To alleviate this problem for large category set, neural networks are usually trained with a subset of samples. Fu and Xu designed probabilistic decision-based neural networks for discriminating groups of classes divided by clustering [53], with each network trained with the samples of the classes in a group. Kimura et al. design an MLP for each of confusing classes, which are determined from the classification on training samples using a statistical classifier [54]. Each MLP discriminates one target class from some rivals that are confused to the target class by the statistical classifier. In classification, an MLP is activated only when its target class is the top-rank class given by the statistical classifier. Saruta et al. design an MLP for each class, but the MLP is trained with the samples of a few classes only [55].

Training SVMs with all samples for Chinese character recognition has been attempted by Dong et al., who designed a fast training algorithm [37]. Though the training with all samples is now feasible due to the increasing power of computers, reducing the complexities of training, storage and classification is concerned for practical applications.

As a discriminative classifier, the LVQ classifier has moderate complexity for large category set [17, 31]. Fukumoto et al. has used a generalized LVQ (GLVQ) algorithm for discriminatively adjusting the class means of quadratic classifiers for large character set recognition [56]. The DLQDF [32], discrim-

inatively adjusting all the parameters of quadratic classifier, provides more accurate classification than LVQ, but its training is very computationally expensive for large category set. By introducing hierarchical rival class search for acceleration, the training of DLQDF on large category set is feasible [57]. Compared to the ML-based MQDF, however, the DLQDF improves the accuracy of handwritten Chinese character recognition only slightly [57, 58].

The mirror image learning method of Wakabayashi et al. [20], for adjusting the covariance parameters of quadratic classifier, was recently applied to handwritten Chinese character recognition with success [59]. Running quadratic classification and modifying covariance matrices for five cycles on training samples, the accuracy of MQDF on test samples was improved from 98.15% to 98.38%. Using compound quadratic discriminant functions for pair discrimination, the test accuracy was further improved to 98.50%.

Feature dimensionality reduction also plays an important role in large character set recognition, since it reduces the classifier complexity (both parameter storage and computation) and possibly, improves the classification accuracy. The Fisher discriminant analysis (FDA) has shown success in many recognition systems [12, 58], though it assumes equal covariance for all classes and tends to blur the difference between nearby classes. Previous hetorescedastic discriminant analysis (HDA) methods are computationally formidable for large category set. A new HDA method was proposed recently and applies effectively to Chinese character recognition [60].

A feature subspace learning method by error minimization, called discriminative feature extraction (DFE) [61], has been tried to improve the accuracy of Chinese character recognition [17, 62, 63, 57]. DFE optimizes the subspace vectors and classifier parameters simultaneously by stochastic gradient descent. With a classifier of single prototype per class, the optimization for thousands of classes is computationally feasible, and the simultaneous optimization of class prototypes and subspace can be viewed as a combination of LVQ and DFE. Using a quadratic classifier on the feature subspace learned by DFE with a prototype classifier, the accuracy of handwritten Chinese character recognition is improved significantly compared to classification on FDA subspace [57].

4 Comparison of Classification Methods

In this section we collect some character recognition results reported in the literature for comparing the performance of the classification methods reviewed above, and we discuss the characteristics of these methods regarding their impacts on practical applications.

4.1 Performance Comparison

The various experiments of character recognition differ in many factors such as the sample data, pre-processing technique, feature representation, classifier

structure and learning algorithm. It is hard to assess the performance of a special classification or learning method from the recognition accuracies reported by different works since the other factors are variable. Only a few works have compared different classification/learning methods based on the same feature data.

For handwritten character recognition, more experiments have been reported for off-line recognition than for on-line recognition. Regarding the target of recognition, the 10 Arabic numerals are most often tested, while Chinese characters or Japanese Kanji characters are often tested in large character set recognition. The numeral databases that have been widely tested include the 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, unlike that the CENPARMI and MNIST databases are partitioned into standard training and test sets.

Performance on Handwritten Numerals

We first collect some high recognition accuracies reported on standard numeral databases, then summarize some results of classification on common feature data.

The CENPARMI database contains 4,000 training samples and 2,000 test samples. Early works using structural analysis hardly reached 95% of correct recognition on this test set [64]. In recently years, 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 the standard 4,000 samples, correct rates over 99% have been given by polynomial classifier (PC) and SVMs with efficient image normalization and feature extraction [4, 65].

The MNIST database contains 60,000 training samples and 10,000 test samples. Each sample was normalized to a gray-scale image of 20×20 pixels, which is located in a 28×28 plane. The pixel values of normalized image are used as feature values, on which different classifiers and learning algorithms can be fairly compared. 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% [21]. Instead of the trainable feature extractors in CNN, extracting heuristically discriminating features also lead to high accuracies. Without training with distorted samples, Teow and Loe obtained a test accuracy 99.57% by extracting local structure features and classification using triowise linear SVMs [66]. On 200D 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].

On the MNIST database, training classifiers without feature extraction show worst performance. Since image pre-processing and feature extraction

are both important to character recognition, a better scheme to compare classifiers is to train them on a common discriminating feature representation. Holmström et al. compared various statistical and neural classifiers on PCA features extracted from normalized images [67]. However, the PCA feature does not perform satisfactorily. In the comparison studies of Liu et al. [68, 4], the features used, chaincode and gradient direction features, are widely recognized and well-performing 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), which is far less complex in storage and execution than SVMs. And the RBF network mostly outperforms the MLP when training all its parameters discriminatively.

A citation of error rates from [4] is shown in Table 1, where "4-grad" and "8-grad" stand for 4-orientation and 8-direction gradient features, respectively; and "SVC-poly" and "SVC-rbf" denotes one-versus-all support vector classifiers with polynomial kernel and RBF kernel, respectively. In this table, the RBF network is shown to be inferior to the MLP on the MNIST dataset, but on many other datasets, the RBF network outperforms the MLP [4].

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

Feature	pixel	PCA	4-grad	8-grad
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

Performance on Large Character Sets

In the area of Chinese/Japanese character recognition, a public handprinted (constrained handwriting) database ETL9B has been widely tested. Various classification methods have been proposed, but they have never been compared on a common feature representation of samples.

The ETL9B database contains 200 samples for each of 3,036 classes, including 2,965 Kanji and 71 hiragana characters. Early works often used 100 samples of odd number from each class for training and the even-numbered samples for testing, and focused on image normalization and feature extraction for improving the performance of feature matching. Nonlinear normalization based on line density equalization [69, 70] and edge direction feature

extraction are now widely accepted. Using the class means of training samples as prototypes, the recognition accuracy on test samples was hardly over 95%. On this sample partitioning scheme, Saruta et al. achieved a correct rate of 95.84% by using class-modular neural networks for fine classification [55]. Using FDA for dimensionality reduction and GLVQ for optimizing the class means, Fukumoto et al. reported a correct rate of 97.22% for Euclidean distance, 98.30% for projection distance (PD) and 98.41% for modified PD (MPD) [56]. The PD and MPD classifiers have comparable complexity with the MQDF, however.

High accuracies have been reported on ETL9B by using quadratic classifiers and SVMs. Nakajima et al. used 160 samples per class for training and the remaining 40 samples for testing, and reported a correct rate 98.90% using MPD and compound MPD [14]. Dong et al. tested on a partially different set of 40 samples per class, and reported a correct rate 99.00% by using SVMs trained on enhanced samples for fine classification [37]. 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 [12]. Suzuki et al. [51] and Kato et al. [52] tested on 20 samples per class in rotation, and both used partial inclination detection for improving normalization. Using compound Mahalanobis distance for fine classification, Suzuki et al. improved the recognition rate from 99.08% to 99.31%. Kato et al. reported a correct rate 99.42% by using asymmetric Mahalanobis distance for fine classification.

Some works reported results on ETL9B as well as databases of handwritten Chinese characters, say, HCL2000 [58] and CASIA [57]. The Chinese databases are not available for free use, however. From the reported results, the Chinese samples turn out to be more difficult to recognize than the samples of ETL9B. Based on nonlinear normalization and gradient direction feature extraction, the accuracies on ETL9B (with samples partitioned as [14]) are as high as 99.33% and 99.39%, while the accuracies on HCL2000 and CASIA databases are 98.56% and 98.43%, respectively. The underlying classification methods are DLQDF+compound quadratic discrminant [58] and DFE+DLQDF [57], respectively.

4.2 Statistical vs. Discriminative Classifiers

We refer to statistical classifiers as those based on parametric or non-parametric density estimation, and discriminative classifiers as those based on minimum (regression or classification) error training. Discriminative classifiers include neural networks and SVMs, for which the parameters of one class are trained on the samples of all classes or selected confusing classes. For statistical classifiers, the parameters of one class are estimated from the samples of its own class only. Non-parametric classifiers like Parzen window method and k-NN rule are not practical for real-time applications, and so, are

not considered in the following discussions. We compare the characteristics of statistical and discriminative 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 since the parameters of the new class are estimated from the new samples only. 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 generalization accuracies than statistical classifiers. This is because discriminative classifiers are trained to separate the samples of different classes in the feature space, while the pre-assumed density form of statistical classifiers limits its capability to accommodate large variability of samples.
- *Dependence on training sample size.* The generalization accuracy of regularized statistical classifiers (like MQDF and RDA) are more stable against the training sample size than discriminative classifiers (see [68]). 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 less parameters than statistical classifiers. Hence, discriminative classifiers are more economical 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. On the other hand, 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) [68]. Parametric statistical classifiers are resistant to outliers because of the assumption of compact density functions, whereas discriminative classifiers are susceptible to outliers because of open decision regions [71]. Outlier rejection is important to integrated segmentation and recognition of character strings [72]. The rejection capability of discriminative classifiers can be enhanced by training with outlier samples.

4.3 Neural Networks vs. SVMs

In addition to the common properties of discriminative classifiers as above, 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 with the aim of optimizing an objective function on training samples. By feeding the training samples a fixed number of epochs, 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 (for character classification) can be adjusted in string-level or layout-level training by gradient descent with the aim of optimizing the global recognition performance [2, 73]. 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 the network structure, and the selection of an appropriate structure relies on cross-validation. The performance of SVMs also depends on the selection of kernel type and kernel parameters, but this dependence is not so influential as the structure selection of neural networks.
- *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 are easy to control. For reducing the execution complexity of SVMs, SV reduction techniques are effective, but may sacrifice the classification accuracy to some degree.

5 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 list these problems and discuss the future research directions of classification and learning that can potentially solve or alleviate them.

5.1 Improvements of Accuracy

Recognition rates over 99% have been reported to handwritten numeral recognition and handprinted Chinese character recognition, but accuracies lower than 90% are often reported to some difficult cases like English letters, cursive words, unconstrained Chinese characters, etc. The recognition rate, even as high as 99.9%, is never sufficient. Any improvement to accuracy will make the recognition system more welcome by users. Improved accuracy can be

achieved by carefully tuning every processing task: pre-processing, feature extraction, sample generation, classifier design, multiple classifier combination, etc. We hereof only discuss some issues related to classification and learning.

- *Feature transformation.* 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.
- *Feature selection.* Character classification has been mostly performed on a limited number of features, which are usually artificially selected. Increasing the number of features complicates the design of classifier and may deteriorate the generalization performance. It is now possible to automatically select a good feature set from huge number of candidate features. With the aim of optimizing separability or description, the selected features may lead to better classification than artificially selected ones.
- *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. Since very large number of distorted samples can be generated and some of them may be misleading, the selection of samples then becomes 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 yielder high accuracy than both the pure statistical and the pure discriminative classifier [74]. A way to design such classifiers is to adjust the parameters of parametric statistical classifiers discriminatively on training samples [75, 32], to improve both generalization accuracy and resistance to outliers. 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 combine classifiers with different properties: training data, pre-processing, feature extraction, classifier structure, learning algorithm, etc. 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.

5.2 Reliable Confidence and Rejection

Since we cannot achieve 100% correct recognition in practice, 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) and posterior probability-like (posterior confidence). Rejecting ambiguous patterns (those confused between different classes) is generally based on posterior confidence, and rejecting outliers (those out of defined classes) is generally based on conditional confidence. If we can estimate the conditional confidence reliably, it would help reject ambiguous patterns as well. Both confidence measures can be unified into the posterior probabilities of open world: normalization to unity for defined classes and an outside world. 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 character recognition methods.

- *Elaborate density estimation.* Probability density estimation is a traditional problem in statistical pattern recognition, but is not well-solved yet. Good density models for character classes can yield both high classification accuracy and rejection capability, especially outlier rejection. The Gaussian mixture model is being studied intensively, and many efforts are given to automatically estimating the number of components. 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 (as discussed above) or support vectors in kernel space [76]. Structural analysis, though do not compete with statistical and discriminative classifiers in classification accuracy, may serve as good verifiers.
- *Hybrid statistical/discriminative learning.* Hybrid statistical/discriminative classifiers, as discussed in 5.1, may yield both high classification accuracy and resistance to outliers. This principle of learning is to be extended to more 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 [77]. Generally, combining complementary classifiers can improve the classification accuracy and the tradeoff between error rate and reject rate.

5.3 Improvements to Large Category Set

Discriminative learning methods have not been extensively applied to the recognition of large character set. Using quadratic classifiers with sophisticated normalization and feature extraction, high accuracies have been reported to handprinted sample databases like the ETL9B. However many misrecognized samples are easily recognized by humans and are potentially solvable by discriminative learning. The probable reasons that neural networks could not perform competitively (e.g. [55]) are: (1) There are few training samples per class (less than 200 in ETL9B); (2) The class-modular network only takes the samples of confusing classes as negative samples, so the resulting network is not resistant to the samples of un-trained classes. The application of SVMs to Chinese character recognition [37] is successful though its accuracy cannot be compared directly to other works because of different pre-processing and feature extraction procedures. To better utilize discriminative classifiers for discriminating similar characters, several issues should be considered seriously.

- *Training sample size.* To demonstrate the benefit of discriminative classifiers, we should use a large number of samples per class for training. If using the discriminative classifier for discriminating a subset of classes only, these selected classes should have more samples than other classes. A new public database released in Japan, called JEITA-HP, contain more than 500 samples per class. The Chinese database HCL2000 contains over 1,000 samples for each of 3,755 classes.
- *Confusing set selection.* The subset of confusing classes is usually selected heuristically according to the classification results of a statistical classifier on training samples. This procedure need to be considered more rigorously, say, from probabilistic view.
- *Type of discriminative classifier.* When the discriminative classifier is used to discriminate a subset of classes, the resistance to outliers is preferable because the patterns of un-trained classes are often presented to the discriminative classifier in execution. In this respect, the hybrid statistical/discriminative classifier is a good choice.
- *Samples for training discriminative classifier.* If a neural or SVM classifier is used to discriminate a subset of classes, it should be trained with samples of other classes too, in order to enhance the resistance to outliers.
- *Fusion of cascaded classifiers.* Using either a multi-class classifier or pairwise classifier for the second-stage classification, fusing the decisions of first-stage and second-stage classifiers probabilistically will benefit the global recognition accuracy.

5.4 Incremental Learning

As discussed in Section 4.2, when adding new classes or new samples to defined classes, discriminative classifiers need to be re-trained with all the

accumulated samples. Incremental learning for adapting existing classifiers to new classes and new samples has rarely been considered in character recognition. Some published works of incremental learning in the neural networks community can be referred for our application. Statistical models can be adapted on new samples without forgetting past data distribution in the framework of Bayesian learning, and hybrid statistical/discriminative models can also be stabilized to past distribution while adapting to new data. For all classifier models, an ensemble classifier can be generated by combining new classifiers trained with new samples with existing ones [78].

The samples for classifier training and adaptation are ever increasing. In addition to adaptation to new labeled data, training with unlabeled data is another topic that is intensively studied in machine learning, called semi-supervised learning [79]. This learning scheme is applicable to character recognition since we cannot attach class labels to all samples artificially.

5.5 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 [80]. 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 sets, researchers often compare the performance at system level: the final recognition rate obtained by integrating all the processing steps. In these circumstances, it is hard to decide what method at which step is the most influential to the final result.

To conduct a fair comparison of methods instead of overall systems, we recommend to use standard techniques for all steps except the step under comparison. 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. It is hoped that open source codes of standard techniques for every processing step of character recognition are released, such that other researchers can fairly compare the methods of a special step. For comparing classifiers, to release common feature data instead of sample images is meaningful.

Acknowledgements

This work was supported by the Hundred Talents Program of CAS, and the Natural Science Foundation of China under grants No.60543004 and No.60121302.

References

1. Mori, S., Suen, C.Y., Yamamoto, K.: Historical review of OCR research and development. Proc. IEEE 80(7) (1992) 1029–1058
2. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE 86(11) (1998) 2278–2324
3. Suen, C.Y., Liu, K., Strathy, N.W.: Sorting and recognizing cheques and financial documents. In: Lee SW, Nakano Y (eds) Document Analysis Systems: Theory and Practice. Springer, LNCS 1655 (1999) 173–187
4. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: Benchmarking of state-of-the-art techniques. Pattern Recognition 36(10) (2003) 2271–2285
5. Marinai, S., Gori, M., Soda, G.: Artificial neural networks for document analysis and recognition. IEEE Trans. Pattern Anal. Mach. Intell. 27(1) (2005) 23–35
6. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Anal. Mach. Intell. 22(1) (2000) 4–37
7. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, 2nd edition (1990)
8. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley Interscience, 2nd edition (2001)
10. Friedman, J.H.: Regularized discriminant analysis. J. Am. Statist. Ass. 84(405) (1989) 165–175
11. Kimura, F., Takashina, K., Tsuruoka, S., Miyake, Y.: Modified quadratic discriminant functions and the application to Chinese character recognition. IEEE Trans. Pattern Anal. Mach. Intell. 9(1) (1987) 149–153
12. Kimura, F., Wakabayashi, T., Tsuruoka, S., Miyake, Y.: Improvement of handwritten Japanese character recognition using weighted direction code histogram. Pattern Recognition 30(8) (1997) 1329–1337
13. Ikeda, M., Tanaka, H., Motooka, T.: Projection distance method of recognition of handwritten characters. Trans. IPS Japan 24(1) (1983) 106–112
14. Nakajima, T., Wakabayashi, T., Kimura, F., Miyake, Y.: Accuracy improvement by compound discriminant functions for resembling character recognition. Trans. IEICE Japan J83-D-II(2) (2000) 623–633
15. Hinton, G.E., Dayan, P., Revow, M.: Modeling the manifolds of images of handwritten digits. IEEE Trans. Neural Networks 8(1) (1997) 65–74
16. Kim, H.C., Kim, D., Bang, S.Y.: A numeral character recognition using the PCA mixture model. Pattern Recognition Letters 23 (2002) 103–111
17. Tsay, M.K., Shyu, K.H., Chang, P.C.: Feature transformation with generalized LVQ for handwritten Chinese character recognition. IEICE Trans. Information and Systems E82-D(3) (1999) 687–92
18. Zhang, P., Bui, T., Suen, C.Y.: Hybrid feature extraction and feature selection for improving recognition accuracy of handwritten numerals. In: Proc. 8th ICDAR, Seoul, Korea, 1 (2005) 136–140
19. Kawatani, T., Shimizu, H.: Handwritten Kanji recognition with the LDA method. In: Proc. 14th ICPR, Brisbane, 2 (1998) 1031–1035
20. Wakabayashi, T., Shi, M., Ohyama, W., Kimura, F.: Accuracy improvement of handwritten numeral recognition by mirror image learning. In: Proc. 6th ICDAR, Seattle (2001) 338–343

21. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: Proc. 7th ICDAR, Edinburgh, UK, 2 (2003) 958–962
22. Oh, I.S., Suen, C.Y.: A class-modular feedforward neural network for handwriting recognition. Pattern Recognition 35(1) (2002) 229–244
23. Pao, Y.H.: Adaptive Pattern Recognition and Neural Networks. Addison-Wesley, MA (1989)
24. Schürmann, J.: Pattern Classification: A Unified View of Statistical and Neural Approaches. Wiley Interscience (1996)
25. Kreßel, U., Schürmann, J.: Pattern classification techniques based on function approximation. In: Bunke H, Wang PSP (eds) Handbook of Character Recognition and Document Image Analysis, World Scientific (1997) 49–78
26. Franke, J.: Isolated handprinted digit recognition. In: Bunke H, Wang PSP (eds) Handbook of Character Recognition and Document Image Analysis, World Scientific (1997) 103–121
27. Liu, C.L., Sako, H.: Class-specific feature polynomial classifier for pattern classification and its application to handwritten numeral recognition. Pattern Recognition 39(4) (2006) 669–681
28. Kimura, F., Inoue, S., Wakabayashi, T., Tsuruoka, S., Miyake, Y.: Handwritten numeral recognition using autoassociative neural networks. In: Proc. 14th ICPR, Brisbane, 1 (1998) 166–171
29. Zhang, B., Fu, M., Yang, H.: A nonlinear neural network model of mixture of local principal component analysis: Application to handwritten digits recognition. Pattern Recognition 34(2) (2001) 203–214
30. Kohonen, T.: The self-organizing map. Proc. IEEE 78(9) (1990) 1464–1480
31. Liu, C.L., Nakagawa, M.: Evaluation of prototype learning algorithms for nearest neighbor classifier in application to handwritten character recognition. Pattern Recognition 34(3) (2001) 601–615
32. Liu, C.L., Sako, H., Fujisawa, H.: Discriminative learning quadratic discriminant function for handwriting recognition. IEEE Trans. Neural Networks 15(2) (2004) 430–444
33. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New Work (1995)
34. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Knowledge Discovery and Data Mining 2(2) (1998) 1–43
35. Kressel, U.: Pairwise classification and support vector machines. In: Schölkopf B, Burges CJC, Smola AJ (eds) Advances in Kernel Methods: Support Vector Learning, MIT Press (1999) 255–268
36. Bellili, A., Gilloux, M., Gallinari, P.: 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) (2003) 244–252
37. Dong, J.X., Krzyzak, A., Suen, C.Y.: An improved handwritten Chinese character recognition system using support vector machine. Pattern Recognition Letters 26(12) (2005) 1849–1856
38. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. System Man Cybernet. 22(3) (1992) 418–435
39. Ho, T.K., Hull, J., Srihari, S.N.: Decision combination in multiple classifier systems. IEEE Trans. Pattern Anal. Mach. Intell. 16(1) (1994) 66–75

40. Rahman, A.F.R., Fairhurst, M.C.: Multiple classifier decision combination strategies for character recognition: A review. Int. J. Document Analysis and Recognition 5(4) (2003) 166–194
41. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20(3) (1998) 226–239
42. Duin, R.P.W.: The combining classifiers: To train or not to train. In: Proc. 16th ICPR, Quebec, Canada, 2 (2002) 765–770
43. Liu, C.L.: Classifier combination based on confidence transformation. Pattern Recognition, 38(1) (2005) 11–28
44. Suen, C.Y., Lam, L.: Multiple classifier combination methodologies for different output levels. In: Kittler J, Roli F (eds) Multiple Classifier Systems, Springer, LNCS 1857 (2000) 52–66
45. Breiman, L.: Bagging predictors. Machine Learning 24(2) (1996) 123–140
46. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Computer and System Sciences 55(1) (1997) 119–139
47. Ha, T., Bunke, H.: Off-line handwritten numeral recognition by perturbation method. IEEE Trans. Pattern Anal. Mach. Intell. 19(5) (1997) 535–539
48. Dahmen, J., Keysers, D., Ney, H.: Combined classification of handwritten digits using the virtual test sample method. In : Kittler J, Roli F (eds) Multiple Classifier Systems, Springer, LNCS 2096 (2001) 99–108
49. Tang, Y.Y., et al.: Offline recognition of Chinese handwriting by multifeature and multilevel classification. IEEE Trans. Pattern Anal. Mach. Intell. 20(5) (1998) 556–561
50. Wang, Q.R., Suen, C.Y.: Analysis and design of a decision tree based on entropy reduction and its application to large character set recognition. IEEE Trans. Pattern Anal. Mach. Intell. 6(4) (1984) 406–417
51. Suzuki, M., Omachi, S., Kato, N., Aso, H., Nemoto, Y.: A discrimination method of similar characters using compound Mahalanobis function. Trans. IEICE Japan J80-D-II(10) (1997) 2752–2760
52. Kato, N., Suzuki, M., Omachi, S., Aso, H., Nemoto, Y.: A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance. IEEE Trans. Pattern Anal. Mach. Intell. 21(3) (1999) 258–262
53. Fu, H.C., Xu, Y.Y.: Multilinguistic handwritten character recognition by Bayesian decision-based neural networks. IEEE Trans. Signal Processing 46(10) (1998) 2781–2789
54. Kimura, Y., Wakahara, T., Tomono, A.: Combination of statistical and neural classifiers for a high-accuracy recognition of large character sets. Trans. IEICE Japan J83-D-II(10) (2000) 1986–1994
55. Saruta, K., Kato, N., Abe, M., Nemoto, Y.: High accuracy recognition of ETL9B using exclusive learning neural network-II (ELNET-II). IEICE Trans. Information and Systems 79-D(5) (1996) 516–521
56. Fukumoto, T., Wakabayashi, T., Kumura, F., Miyake, Y.: Accuracy improvement of handwritten character recognition by GLVQ. In: Proc. 7th IWFHR, Amsterdam (2000) 271–280
57. Liu, C.L.: High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction. In: Proc. 18th ICPR, Hong Kong, 2 (2006) 942–945

58. Liu, H., Ding, X.: Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes. In: Proc. 8th ICDAR, Seoul, Korea, 1 (2005) 19–23
59. Liu, H., Ding, X.: Handwritten Chinese character recognition based on mirror image learning and the compound Mahalanobis distance. J. Tsinghua Univ. (Sci & Tech) 46(7) (2006) 1239–1242 (in Chinese)
60. Liu, H., Ding, X.: Improve handwritten character recognition performance by heteroscedastic linear discriminant analysis, In: Proc. 18th ICPR, Hong Kong, 1 (2006) 880–883
61. Biem, A., Katagiri, S., Juang, B.H.: Pattern recognition using discriminative feature extraction. IEEE Trans. Signal Processing 45(2) (1997) 500–504
62. Huo, Q., Ge, Y., Feng, Z.D.: High performance Chinese OCR based on Gabor features, discriminative feature extraction and model training. In: Proc. ICASSP'01, Salt Lake City, Utah, 3 (2001) 1517–1520
63. Liu, C.L., Mine, R., Koga, M.: Building compact classifier for large character set recognition using discriminative feature extraction. In: Proc. 8th ICDAR, Seoul, Korea (2005) 846–850
64. Suen, C.Y., Nadal, C., Legault, R., Mai, T.A., Lam, L.: Computer recognition of unconstrained handwritten numerals. Proc. IEEE 80(7) (1982) 1162–1180
65. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognition 37(2) (2004) 265–279
66. Teow, L.N., Loe, K.F.: Robust vision-based features and classification schemes for off-line handwritten digit recognition. Pattern Recognition 35(11) (2002) 2355–2364
67. Holmström, L., Koistinen, P., Laaksonen, J., Oja, E.: Neural and statistical classifiers—taxonomy and two case studies. IEEE Trans. Neural Networks 8(1) (1997) 5–17
68. Liu, C.L., Sako, H., Fujisawa, H.: Performance evaluation of pattern classifiers for handwritten character recognition. Int. J. Document Analysis and Recognition 4(3) (2002) 191–204
69. Tsukumo, J., Tanaka, H.: Classification of handprinted Chinese characters using non-linear normalization and correlation methods. In: Proc. 9th ICPR, Rome (1988) 168–171
70. Yamada, H., Yamamoto, K., Saito, T.: A nonlinear normalization method for hanprinted Kanji character recognition—line density equalization. Pattern Recognition 23(9) (1990) 1023–1029
71. Gori, M., Scarselli, F.: Are multilayer perceptrons adequate for pattern recognition and verification? IEEE Trans. Pattern Anal. Mach. Intell. 20(11) (1998) 1121–1132
72. Liu, C.L., Sako, H., Fujisawa, H.: Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings. IEEE Trans. Pattern Anal. Mach. Intell. 26(11) (2004) 1395–1407
73. Liu, C.L., Marukawa, K.: Handwritten numeral string recognition: Character-level training vs. string-level training. In: Proc. 17th ICPR, Cambridge, UK, 1 (2004) 405–408
74. Raina, R., Shen, Y., Ng, A.Y., McCallum, A.: Classification with hybrid generative/discriminative models. In: Advances in Neural Information Processing System 16 (2003)

75. Dahmen, J., Schluter, R., Ney, H.: Discriminative training of Gaussian mixtures for image object recognition. In: Proc. 21st Symposium of German Association for Pattern Recognition, Bonn, Germany (1999) 205–212
76. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Machine Learning 54(1) (2004) 45–66
77. Suen, C.Y., Tan, J.: Analysis of error of handwritten digits made by a multitude of classifiers. Pattern Recognition Letters 26(3) (2005) 369–379
78. Polikar, R., Udpa, L., Udpa, A.S., Honavar, V.: Learn++: An incremental learning algorithm for supervised neural networks. IEEE Trans. System Man Cybernet. Part C 31(4) (2001) 497–508
79. Chawla, N.V., Karakoulas, G.: Learning from labeled and unlabeled data: An empirical study across techniques and domains. J. Artificial Intelligence Research 23 (2005) 331–366
80. Duin, R.P.W.: A note on comparing classifiers. Pattern Recognition Letters, 17 (1996) 529–536