A Skew-tolerant Strategy and Confidence Measure for k-NN Classification of Online Handwritten Characters

Vandana Roy and Sriganesh Madhvanath
HP Laboratories
HPL-2008-52
May 21, 2008*

Online Handwritten Character Recognition, Confidence measures, Skewed distribution, k-NN

Confidence measures for k-NN classification are an important aspect of building practical systems for online handwritten character recognition. In many cases, the distribution of training samples across the different classes is marked by significant skew, either as a consequence of unbalanced data collection or because the application itself incrementally adds samples to the training et over a period of use. In this paper, we explore the adaptive k-NN classification strategy and confidence measure in the context of such skewed distributions of training samples, and compare it with traditional confidence measures used for k-NN classification as well as with confidence transformations learned from the data. Our experiments demonstrate that the adaptive k-NN strategy and confidence measure outperforms other measures for problems involving both large and small sets of training data.

External Accession Date Only Approved for External Publication

Submitted to International Conference on Frontiers on Hand-writing Recognition (ICFHR 2008)

A Skew-tolerant Strategy and Confidence Measure for k-NN Classification of Online Handwritten Characters

Vandana Roy and Sriganesh Madhvanath
Hewlett-Packard Labs, Bangalore, India
vandana.roy, srig@hp.com

Abstract

Confidence measures for k-NN classification are an important aspect of building practical systems for online handwritten character recognition. In many cases, the distribution of training samples across the different classes is marked by significant skew, either as a consequence of unbalanced data collection or because the application itself incrementally adds samples to the training set over a period of use. In this paper, we explore the adaptive k-NN classification strategy and confidence measure in the context of such skewed distributions of training samples, and compare it with traditional confidence measures used for k-NN classification as well as with confidence transformations learned from the data. Our experiments demonstrate that the adaptive k-NN strategy and confidence measure outperforms other measures for problems involving both large and small sets of training data.

Keywords: Online Handwritten Character Recognition, Confidence measures, Skewed distribution, k-NN

1. Introduction

Nearest-neighbor (NN) and k-nearest neighbors (kNN) based recognizers have widely been used for handwritten character recognition. When used in applications, it is very important to compute reliable confidences corresponding to the recognition results. The confidence values are typically computed during the post-processing phase of the recognizer. They are the measures of correctness of output of a recognizer. The estimation of confidences requires higher values to be assigned to the correct recognition results, and lower values to the incorrect recognition results.

In many cases, the distribution of training samples across the different classes is marked by significant skew. For example, for English character recognition, word samples are collected, and the characters are segmented for training. As the frequency of some characters (e.g. a, i, o, etc.) is higher than others (e.g. z, q, etc.), this leads to a skewed distribution of samples across different classes. Another example where the training samples can be skewed is Indic scripts. In Indic scripts, the diacritical marks, like matras, cannot be collected in isolation, they are associated with consonants. Thus, while collecting the data at the character level, the matras occur more frequently than the consonants, hence leading to skewed distribution of samples.

Other applications such as those that allow users to define their own gestures for carrying out different commands, typically have to work with very small initial numbers of these user-defined gestures, and add more samples over a period of use. The distribution of samples at a given point in time is a function of when different gestures were created and how often they were used, and may be highly skewed across the different gesture classes.

In this paper, we explore the issue of confidence measures for k-NN classification of online handwritten character recognition when the distribution of training samples across different classes is heavily skewed or biased. We look at the traditional strategies and ways of computing confidences for k-NN classifiers and discuss some of their shortcomings with respect to skewed distributions. One shortcoming is the use of a fixed number of nearest neighbors (k).

Whereas confidence measures are typically computed as a postprocesing step, we also consider variations to the k-NN strategy itself that results in improved confidence measures. In particular, we explore the Adaptive-kNN strategy [3] and confidence measure proposed by Baoli *et al.* for text category classification, which uses different number of nearest neighbors for each class as opposed to a fixed number.

The paper is organized as follows. In Section 2, we discuss prior work in the area of confidence measures in related areas. Section 3 provides a detailed description of the popular nearest neighbor based recognizers, and associated confidence measures. In this section, we also discuss shortcomings of the traditional k-NN based confidence measures. In Section 4 we discuss the adaptive-kNN strategy and confidence measure and how it addresses some of the shortcomings of traditional k-NN. We then discuss the experiments conducted and analyze the results obtained in Section 5. We offer some conclusions and discuss future directions for this work in Section 6.

2. Related Work on Confidence Measures

There has been considerable work on computing and estimating confidences in related areas such as natural language processing, automatic speech recognition, optical character recognition, etc. Simona *et al.* [6] gave an overview of application of confidence estimation in various fields of natural language processing, including speech recognition, spoken language understanding, and statistical machine translation. Campbell *et al.* [5] proposed a framework for confidence estimation based upon scores and meta-information, such as utterance duration, channel type, and signal-to-noise ratio for automatic speech recognition. The proposed framework used regression techniques with multilayer perceptron (MLP) to estimate confidences using a data-driven methodology. Confidence measures for large vocabulary for automatic speech recognition [2] were discussed by Wendemuth *et al.* Data-dependent and data-independent measures were investigated. The transformations to single measure, and linear combination of these measures were computed using neural networks. Gillick *et al.* [7] proposed a probabilistic approach to confidence estimation for words recognized by a speech recognition system. The approach was based on an interpretation of a confidence as the probability that the corresponding recognized word is correct, and made use of generalized linear models as a means for combining various predictor scores so as to arrive at confidence estimates.

A comparison of various methods of classification and learning for offline character recognition is discussed by Liu *et al.* [9]. A brief description of confidence and rejection methods used in this area is presented. Class conditional probability and posterior probability based confidence measures are reported to be used extensively in the area of OCR.

Arlandis *et al.* [1] discussed rejection strategies and confidence measures for offline handwritten character recognition. They proposed a two-level confidence computation scheme to first reject the sample if the test sample was noisy, and to compute the confidence in the next level. The confidence value is computed by estimating the *a posteriori* probability taking into account the distances of the nearest neighbors. Various confidence measures for offline handwritten word recognition applied to postal address reading system are discussed by Brakensiek *et al.* [4]. An HMM classifier was used for recognition of words. Four different likelihood ratio based confidence measures specifically for HMM were compared.

Pitrelli *et al.* [10] applied confidence scoring techniques to verify the output of an offline handwritten character recognition system. They evaluated various confidence measures including raw recognition score, likelihood ratio, estimated posterior probability, and negative entropy.

Lin *et al.* [8] proposed an adaptive confidence transform (ACT) method for mapping distances to probabilities, in the context of Chinese character recognition. A two-phase transform was proposed to map the distances obtained from the classifier to the *a posteriori* probabilities. In the first phase, generalized confidence values are computed using the distances of the test sample to the training samples. In the second phase, the confidence values are converted to the *a posteriori* probabilities through a trained transform.

There has not been significant work on confidence measures for the online handwritten character recognition for skewed distribution of samples. Rather than mapping the result of recognizer to confidences during the post-processing phase, some research effort has gone into modification of the kNN recognition strategy to get better results. Baoli *et al.* describe a strategy Adaptive-kNN [3] to handle skewed distribution of samples for the problem of large scale text categorization. In this paper, we use this particular recognition strategy and associated confidence measure for classification of online handwritten characters in the presence of skewed distribution of training samples.

3. Nearest Neighbors based Recognizers for Online Handwritten Character Recognition

The nearest-neighbor and k-nearest neighbors classifiers have widely been used in the area of handwritten character recognition. For recognition of online handwritten characters, the ink-samples are first preprocessed by resampling and normalization. Features, e.g. coordinate values, curvature, etc. are then extracted from these preprocessed ink samples. Distances are then computed between the features extracted from training and test samples. Euclidean, Dynamic Time Warping (DTW), Manhattan, Mahalanobis distances, etc. are popularly used. Smith *et al.* [11] investigated three different distance metrics, namely, Hamming distance metric, pixel distance metric, and a metric based on the extraction of penstroke features for use in nearest neighbor classifier. Raw distances between the feature vectors are not a good indicator of the recognition reliability. Also, for comparing or combining different classifiers, the confidence values must be comparable. Hence it is desirable to convert the raw distances obtained to the more meaningful confidence values.

The traditional NN is the simplest classifier which assigns the label ω_j of the nearest sample to the test sample

t.

$$\omega_j = argmin_{\omega_i}(min_{x \in \omega_i}(distance(t,x))), i = 1 \ldots c \quad (1)$$

where c is the number of classes. The traditional k-NN classifier assigns the label of the majority of the samples of the same class within the k-nearest neighbors.

$$\omega_j = argmax_{\omega_i} \Sigma_{x \in kNN} y(x, \omega_i) \quad (2)$$

where the function y is defined as:

$$y(x, \omega_i) = \begin{cases} 1 & \text{if } x \in \omega_i \\ 0 & \text{otherwise} \end{cases}$$

Thus, the class which has the maximum number of samples within the k-nearest neighbors to the test sample is the winner. The confidence value of the sample belonging to the class ω_j is typically computed as:

$$Conf(\omega_j) = \frac{\Sigma_{x \in kNN} y(x, \omega_j)}{k} \quad (3)$$

Variants of these NN and kNN classifiers use similarity measures, instead of the distance values for recognition. Any similarity measure, such as inverse exponential distance, or inverse distance values can be used. Such a variant of NN classifier assigns the label of the sample with the maximum similarity to the test sample.

$$\omega_j = argmax_{\omega i}(max_{x \in \omega_i}(similarity(t,x))) \quad (4)$$

The confidence of the nearest sample can be computed as the ratio of similarity value of the nearest sample to the sum of similarity values of nearest samples of each class.

$$Conf(\omega_j) = \frac{max_{x \in \omega_j}(similarity(t,x))}{\Sigma_{\omega_k, k=1 \ldots c}(max_{y \in \omega_k}(similarity(t,y))} \quad (5)$$

Similarly, a variant of the k-NN classifier assigns the label of the class with the maximal sum of similarities.

$$\omega_j = argmax_{\omega_i} \Sigma_{x \in kNN} similarity(x,t) y(x, \omega_i) \quad (6)$$

The confidence associated with each unique class ω_i present in k-nearest neighbors is computed by:

$$Conf(\omega_i) = \frac{\Sigma_{x \in S_i} similarity(t,x)}{\Sigma_{i=1}^{k} similarity(t, x_i)} \quad (7)$$

Here, S_i is the subset of samples in the k-nearest neighbors, that belong to class ω_i.

3.1. Problems with kNN for Skewed Distribution of Training Samples

The traditional kNN algorithm computes the optimal value of "k" over a validation set, and uses this fixed value for all the classes during recognition. However, in many scenarios, as discussed earlier, the distribution of samples may not be uniform across all classes. As the distribution of samples is skewed, the same distribution cannot be assured over validation and training sets. Also, in applications that allow incremental learning of the samples to improve the accuracy of the recognizer, training samples are added to the set over time. Hence, the value of "k" computed over the validation set does not remain valid for the training set.

When the distribution of samples is skewed, it is very likely that a fixed k value will result in a bias towards classes with more number of samples. For example, consider the following scenario: the test sample belongs to a class which has only two samples in the training set, while all the other classes have large number of samples. The value of k is fixed to 10. Even if the two samples of the correct class are present in the k nearest neighbors, they are in the minority, resulting in incorrect recognition. On the other hand, if k (=2) to favor the class with the smallest number of samples, consider the case when the test sample belongs to a class having 15 samples in the training set. The value of k (=2) loses most of the discriminatory information present in the rest of samples of the class. Hence, fixing a value of k in the case of skewed distributions of samples leads to either a bias to larger classes, or loss of discriminatory information.

4. The Adaptive-kNN Strategy

To make full use of the discriminatory information in the training set, without biasing the results towards the classes having large numbers of samples, an adaptive-kNN strategy was proposed by Baoli *et al.* [3] for use in text categorization. As opposed to the fixed number of nearest neighbors used for all the classes, this strategy uses different number of nearest neighbors to classify the test sample. For each class ω_i, n_i nearest neighbors is used for computation of confidence, rather than a fixed k nearest neighbors. The value of n_i is determined based on the number of training samples of that class, N_i, and the value of k. The number of nearest neighbors for each class is based on its sample distribution in the training set. This allows smaller numbers of nearest neighbors to be used for classes with smaller training set size, and larger number of samples to be used for classes with large number of samples.

The confidence of the class ω_i is computed from the ratio of the number or similarity sum of neighbors belonging to the class in the n_i nearest neighbors to the total number or similarity sum of all the n_i selected nearest neighbors for that class. This is similar to the way confidence is computed for a k-nearest neighbor classifier, except that number of nearest neighbors considered for each class is different. The class with the highest confidence is the winner. The expression to compute the confidence of class ω_i is given by

$$Conf(\omega_i) = \frac{\Sigma_{x \in S_i} similarity(t, x)}{\Sigma_{i=1}^{n_i} similarity(t, x_i)} \quad (8)$$

We compute the value of n_i for the class ω_i using the expression:

$$n_i = max(\alpha, min(\lceil \frac{k * N_i}{max(N_j), j = 1..c} \rceil, N_i)) \quad (9)$$

Here, N_i is the number of available training samples of class ω_i. The denominator $max(N_j), j = 1..c$ is the size of the largest class; c is the number of classes, and α is a non-negative integer. The parameter α ensures a minimum value of nearest neighbors n_i of class ω_i to be used, in case the second value is very low. Without α, n_i may be too small or even equal to 1 for some smaller classes for a training set with a skewed class distribution.

The expression to compute n_i for every class is slightly different from that used in [3]:

$$n_i = min(\alpha + \lceil \frac{k * N_i}{max(N_j), j = 1..c} \rceil, k, N_i) \quad (10)$$

Instead of adding α to the second term to ensure a minimum number of nearest neighbors, we compute the minimum of weighted proportional size of the class in the dataset ($\frac{k * N_i}{max(N_j), j=1..c}$), and the size of that class to compute the number of nearest neighbors to be considered while recognition. Then we compute the maximum of α and the computed minimum value to ensure a minimum number of nearest neighbors in order to handle the small-scale datasets. It can also be noted, that in our formulation, we omitted the term k for computation of number of neighbors, as $\frac{k * N_i}{max(N_j), j=1..c}$ is always less than k.

When the class distribution in a training set is absolutely uniform, the adaptive-kNN reduces to traditional k-NN. As in the case of k-NN strategy, the value of k is determined using a validation set, but as we show in the experiments later, performance of the adaptive-kNN strategy is not very sensitive to this choice. Algorithm-1 describes the procedure to recognize a test sample using the Adaptive-kNN confidence measure.

5. Experiments and Results

In order to evaluate the performance of different strategies and associated confidences in the presence of skew, we created two kinds of training datasets of online handwritten characters - "small-scale" datasets which had a small number of training samples, and "large-scale" ones involving larger numbers. These are representative of the different scenarios described in the introductory section.

Algorithm 1 Adaptive-kNN for Online Handwritten Symbol Recognition

t: Test Sample to be recognized
c: Total number of classes
N_i: Total number of samples of class ω_i, i = 1…c
N_{max}: Maximum class size in the training set
kNN: Set of k-nearest neighbors of t
n_i: Number of nearest neighbors to be considered for class ω_i, i = 1…c
$n_i = max(\alpha, min(\lceil \frac{k*N_i}{N_{max})} \rceil, N_i))$
for each unique class $i \in kNN$ samples **do**
  Compute the confidence of the test sample belonging to class i as:
  $Conf(\omega_i) = \frac{\Sigma_{x \in S_i} similarity(t,x)}{\Sigma_{i=1}^{n_i} similarity(t,x_i)}$
end for
Sort the classes based on increasing order of confidences.
Classify the test sample based on the computed confidences.

We artificially introduced skew into these datasets and evaluated the performance of three different strategies (i) 1-NN, (ii) k-NN, (iii) Adaptive k-NN, and the respective confidence measures as in Equations 5, 7, and 8. We also compared the adaptive-kNN strategy and related confidence measure with the Adaptive Confidence Transform method applied to the confidence measure from the NN strategy. The similarity measure used in all our experiments is the inverse of distance. In the following paragraphs, we first describe the construction of the datasets used for our experiments. Then we evaluate the effect of changing the value of k on k-NN and adaptive-kNN strategies, followed by comparison of effectiveness of the different strategies and associated confidences on skewed large-scale and small-scale datasets.

Datasets We used numerals, lower case and upper case letters from the IRONOFF [12] dataset collected at IRESTE, University of Nantes (France) for our experiments. These character sets were comprised respectively of 4,086 isolated digits, 10,685 isolated lower case letters and 10,679 isolated upper case letters. The data was collected from different writers, and hence each class consisted of various different writing styles. In each category (lower, upper, numerals), the datasets were partitioned into three groups - validation set, training set, and test set.

In order to simulate skew in the validation and training set, we randomly selected samples from the above validation and training sets to create skewed datasets. The small-scale skewed training sets consisted of $1 - 3$ samples per class, whereas the large-scale skewed training sets consisted of $5 - 125$ samples per class. Thus the small-scale training sets had a small number of samples per class with less skew, while the large-scale training sets had large numbers of samples per class with large skew across classes. The test sets were left unmodified and consisted of 140 samples per class.

In the experiments described below, the process of generating skewed training sets was repeated thrice for each of lower case, upper case and numerals datasets to create three different small scale and large scale skewed datasets. Three different testing sets were used. Each experiment was run on three (training set, testing set) pairs, and all the results shown are the average of these three runs.

The validation sets were used to determine the optimal value of the parameter k. This value of k was then used for both the k-NN and Adaptive k-NN schemes.

Effect of choice of k on performance We conducted experiments using the different skewed datasets described earlier to study the impact of the choice of k on the accuracy of classification in the presence of skewed distributions. By way of illustration, the results on the skewed large-scale and small-scale datasets for lowercase characters is shown in Figure 1. The plot shows the affect of increasing the value of the k parameter on the accuracy of the recognizer. The accuracy is computed as the fraction of number of samples correctly recognized over the total number of samples in the testing dataset (i.e. without rejection). The figure shows that while the performance of traditional kNN is highly sensitive to the value of k, that of adaptive-kNN is much less so. When the distribution is skewed and potentially changing over time as new training samples are added, the optimal value of k to be used for kNN changes. Adaptive kNN is much more useful since it is not very sensitive to the value of k chosen.

Comparison of Confidence Measures using ROC curves In order to compare the different strategies and confidence measures, we classified the test samples and used the confidence values for the top choices to plot the Receiver Operating Characteristic plots. To show the effectiveness of the adaptive-kNN strategy and associated confidence measure over traditional NN, kNN, and ACT based confidence measures, we plotted the error vs. reject rate ROC curves to obtain a graphical depiction of the relationship between the error rate and reject rate on the test sets, as a function of the confidence threshold's value. Error rate is defined as the ratio of number of false accepts to total number of test samples with confidence greater than or equal to the confidence threshold. The reject rate is defined as the ratio of the samples rejected (i.e., with confidence value less than confidence threshold over the total number of samples). There is a trade-off between the two; our goal is to simultaneously minimize both error rate as well reject rate. In other words, the closer the ROC is to the axes, the better the confidence measure.

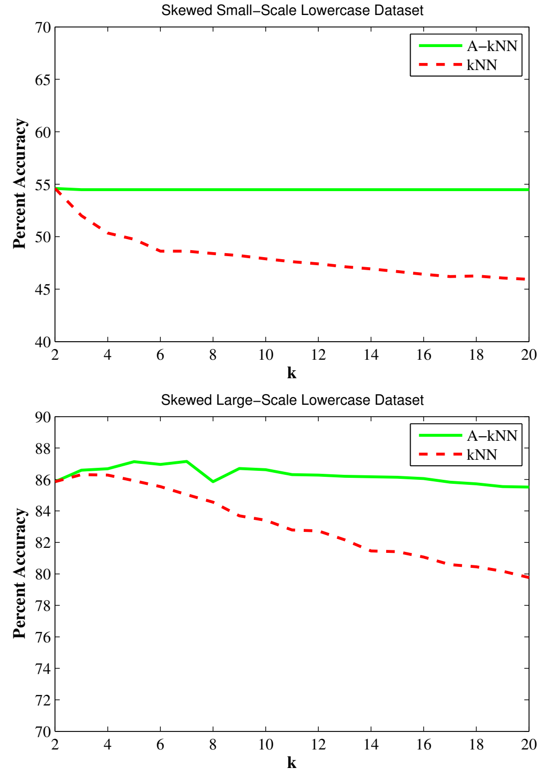

Figure 1. Effect of varying the value of k on English lower case dataset

The experiment was conducted on the small scale and large scale datasets of lower case, upper case and numerals categories. The results on the skewed small-scale and large-scale numerals datasets are illustrated by Figure 2. This plot shows that for both the small-scale and large-scale skewed datasets, the ROC curves plotted for the adaptive-kNN is closest to the axes, as compared to NN, kNN and ACT based confidence measures. This means that the confidence computed using adaptive-kNN method gives minimum error and reject rates. As these ROC curves were plotted for confidence threshold values in the interval of (0,1), some of the curves do not touch the axes. For the small-scale case, the ROC curve of adaptive-kNN is very close to that of kNN confidence measure. This is because of the smaller skew of the samples in the training dataset. As the distribution approaches uniform, the adaptive-kNN reduces to the confidence value computed for kNN strategy. For the large-scale case, the ROC curve of adaptive-kNN is very close to the ROC curve for ACT. This is possibly because, with large number of samples, the confidence transform learned is good. However unlike the ACT, the adaptive-kNN requires no training on the specific dataset.

These plots clearly show that adaptive-kNN strategy and associated confidence measure performs best among the confidence measures studied, for both small and large scale datasets with skewed distribution of samples.

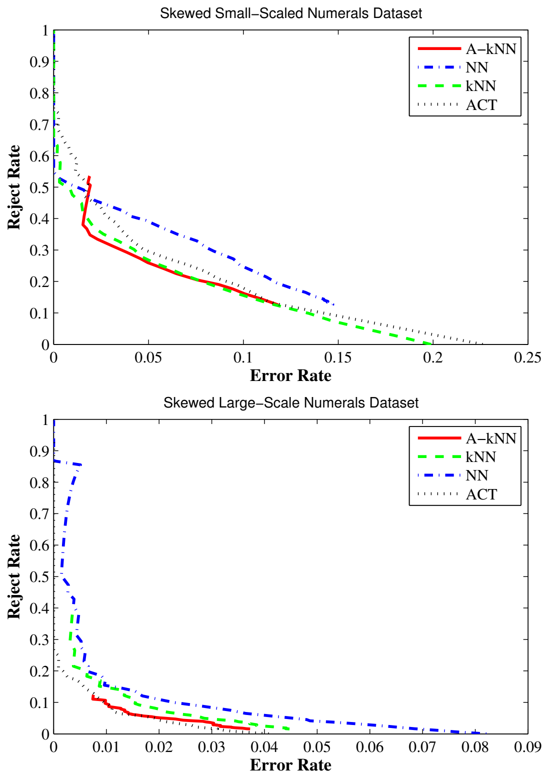

Figure 2. ROC curve for numerals dataset

6. Conclusions and Future Work

In this paper, we explored the Adaptive-kNN strategy and confidence measure to address the problem of on-line handwritten character recognition problem in presence of skewed training sets. We showed through experiments, that performance of the traditional kNN recognition strategy and confidence measure is highly sensitive to the value of k, while adaptive-kNN strategy and confidence measure is not so sensitive to the value of k. Hence, adaptive-kNN works well on the skewed training sets. We also compared the performance of Adaptive-kNN strategy and related confidence measure with various nearest neighbor based strategies and confidences, namely NN, kNN, ACT, by plotting the ROC curves, and observed that Adaptive-kNN strategy and confidence measure outperforms the NN, kNN, and ACT, when the distribution of samples across classes is skewed. This is a very promising technique for use in applications where the distribution of training samples is skewed due to unbalanced data collection or due to samples getting added over a period of use.

As future research work, we plan to formulate the confidence computed using the adaptive-kNN strategy into posterior probability for use in combination of classifiers. We also plan to conduct rigorous experiments on the datasets with the realistic skewed distribution of samples.

References

[1] J. Arlandis, J. C. Perez-Cortes and J. Cano, ”Rejection strategies and confidence measures for a k-NN classifier in an OCR task”, *International Conference on Pattern Recognition (ICPR)*, 2002, volume 1, pp 576–579.

[2] A.Wendemuth, G. Rose and J. Dolfing, ”Advances in Confidence Measures for Large Vocabulary”, *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 1999, volume 2, pp 705–708.

[3] L. Baoli, L. Qin and Y. Shiwen, ”An Aadaptive k-Nearest Neighbor Text Categorization Strategy”, *ACM Transactions on Asian Language Information Processing*, December 2004, volume 3, pp 215–226.

[4] A. Brakensiek, J. Rottland and G. Rigoll, ”Confidence Measures for an Address Reading System”, *International Conference on Document Analysis and Recognition (ICDAR)*, August 2003, volume 1, pp 294–298.

[5] W. M. Campbell, D. A. Reynolds, J. P. Campbell and K. J. Brady, ”Estimating and Evaluating Confidence for Forensic Speaker Recognition”, *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2005, volume 1, pp 717–720.

[6] S. Gandrabur, G. Foster and G. Lapalme, ”Confidence Estimation for NLP Applications”, *ACM Transactions on Speech and Language Processing (TSLP)*, October 2006, volume 3, pp 1–29.

[7] L. Gillick, Y. Ito and J. Young, ”A Probabilistic Approach to Confidence Estimation and Evaluation”, *International Conference on Acoustics, Speech and Signal Processing*, April 1997, volume 2, pp 879–882.

[8] X. Lin, X. Ding, M. Chen, R. Zhang and Y. Wu, ”Adaptive confidence transform based classifier combination for Chinese character recognition”, *Pattern Recognition Letters*, August 1998, volume 19, pp 975–988.

[9] C.-L. Liu and H. Fujisawa, ”Classification and Learning for Character Recognition: Comparison of Methods and Remaining Problems”, *Neural Networks and Learning in Document Analysis and Recognition*, August 2005, pp 1–7.

[10] J. F. Pitrelli and M. P. Perrone, ”Confidence-scoring post-processing for off-line handwritten-character recognition verification”, *International Conference on Document Analysis and Recognition (ICDAR)*, August 2003, volume 1, pp 278–282.

[11] S. Smith, M. Bourgoin, K. Sims and H. Voorhees, ”Handwritten Character Classification Using Nearest-Neighbor in Large Databases”, *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, September 1994, volume 16, pp 915–919.

[12] C. Viard-Gaudin, P. M. Lallican, P. Binter and S. Knerr, ”The IRESTE On/Off (IRONOFF) dual handwriting database”, *International Conference on Document Analysis and Recognition (ICDAR)*, September 1999, pp 455–458.