Vision Interface '99, Trois-Rivières, Canada, 19-21 May

Test Feature Classifiers and its Applications

V. Lashkia
Department of Inf. and Computer Eng.
Okayama University of Science
1-1 Ridai-cho, Okayama, 700-0005 Japan
lashkia@ice.ous.ac.jp

S. Aleshin
Department of Mathematical Theory
of Intellectual Systems
Moscow State University

Abstract

In this paper, we present a class of combinatorical-logical classifiers called test feature classifiers. Introducing kernels and a rejection option we discuss the properties and performance of the proposed classifiers. To test the performance of the classifiers, we apply them to a well-known phoneme database and textual region location problem. Our experimental results show that the proposed classifiers have very high stability and performance and suggest that they can be used in a variety of pattern recognition applications.

1. Introduction

Since the birth of electrical computing machines important advances have been made in the field of classifier learning from examples. However many classifiers in the learning phase require optimization methods, and have problems with convergence, stability and time efficiency. Statistical and structural methods learn badly when exact statistical or structural knowledge is not available. Any method based on metrics uses the hypothesis that the proximity in the data space generally expresses the membership of the same class, and therefore a data set which does not satisfy this condition cannot be treated by such an approach.

In this paper, we present a class of combinatorical-logical classifiers called test feature classifiers. These classifiers allow us to avoid many of the above drawbacks. Classifiers are generated directly from training samples using so-called *tests*, sets of features that are sufficient to distinguish patterns from different classes of training samples. The concept of the test was first introduced in [1] for the purpose of digital logic circuit analysis. It was then realized that tests could be very useful in pattern recognition. First, as the pattern recognition tool tests were used in [2]. The concept of test feature classifiers was presented in [3].

Test feature classifiers are m-degree polynomials, and can be used for partitioning the n-dimensional feature space, $m \leq n$, whose features are assumed to be binary-valued. Optimization methods, statistical, structural or metrical characteristics of patterns are not required. In this work, we have attempted to address some issues relevant to test feature classifiers which have not previously received attention. The performance of the proposed classifiers depend on the selection of tests and training samples. Introducing kernels and a rejection option we discuss the properties and performance of the proposed classifiers. We describe cases when a 100% of recognition rate can be achieved.

We applied the proposed classifiers to a well-known phoneme database and textual region location problem.

The choice of the phoneme database was dectated by two factors: the phoneme database is one of the largest available in the Internet, and it is known to be a difficult classification problem. We compere the performance of proposed classifiers to the performance of the conventional classifiers. Our simulations show that the proposed classifiers have much better performance than conventional methods.

Automatic textual region location deals with extracting image regions that just contain text. This is very useful in a number of applications such as image database retrieval, identification of products (such as books, CDs, etc), and reading street signs and notices which is an important part of the automatic function of robots and navigation systems. Several Approaches to text location in complex images have been proposed [7, 6, 8, 9]. In [7] a connected component approach in [8] a connected component approach in combination with OCR system have been proposed. In [6] a hybrid method has been proposed which combines the connected component approach and a spatial variance approach. In [9] the method of multivalued image decomposition was proposed. The connected component method does not work well with natural images, where the characters have blurred boundaries, images with non-uniform illumination, images where characters have similar color as a background, and images with characters of different color components. The connected component method is also sensitive to the heuristic threshold values such as the size and diameter of component, and length of aligned characters. The spatial variance approach is direction sensitive and does not work well with natural images with high variance background, or images where characters are not aligned. The multivalued image decomposition method is also based on the connected component analysis, it is direction sensitive, and can only locate horizontal text, which is relatively large size.

In an attempt to avoid the above shortcomings we introduce a new approach for locating the text, which is based on Fourier spectrum features and test feature classifiers. The main characteristic of text is its regularity. We propose that frequency space characteristics are more appropriate for text region recognition. We assume that the text's regularities are implicitly contained in the Fourier spectrum features. In a typical classification system, feature extraction is usually of key significance to the overall system performance. However the performance of the system also strongly depends on the type of classifier used. We show that even with a primitive feature extraction procedure the proposed $TFNN$ classifier yields surprisingly stable and high recognition rates. The major advantage of the proposed method locating text compared with the previous approaches is its, orientation, alignment, size and color independence.

2. Test Feature Classifiers

Assume that P is an n-dimensional feature space, $P = \{\bar{t} = (t_1, ..., t_n)\}$, and each pattern is represented as a binary-valued feature vector in this space $t_i \in \{0, 1\}$. Let us also assume that there are 2 possible classes I_1 and I_2. The problem of designing a classifier for pattern recognition can be stated as follows: a function V must be found such that a pattern $\bar{x}$ is in the class I_1 (in the class I_2) if and only if $V(\bar{x}) \geq 0$ ($V(\bar{x}) < 0$).

Let us denote $B_1 = \{\bar{x}_1, ..., \bar{x}_{m_1}\}$ as a set of training samples from the class I_1 and $B_2 = \{\bar{y}_1, ..., \bar{y}_{m_2}\}$ as a set of training samples from the class I_2, where $\bar{x}_i = (x_1^i, ..., x_n^i), i = 1, ..., m_1$, $\bar{y}_i = (y_1^i, ..., y_n^i), i = 1, ..., m_2$, and $B_1 \cap B_2 = \emptyset$. A collection of features, $\tau = \{i_1, ..., i_k\}, (1 \leq k \leq n)$ is called a *test feature* (or test) of B_1 and B_2 if for any p $(1 \leq p \leq m_1)$ and any q $(1 \leq q \leq m_2)$ there exist some $i_s \in \tau$ $(1 \leq s \leq k)$ such that $x_{i_s}^p \neq y_{i_s}^q$. In other words, a test is a collection of features (a projection) which is sufficient to distinguish vectors from different classes of training samples. If for a test τ, the set $\tau - \{i_s\}$ is not a test for any s $(1 \leq s \leq k)$, then τ is called a *prime test feature* (or prime test).

A test $\tau = \{i_1, ..., i_k\}$, can be considered as an n-tuple vector, $\bar{\tau} = (\tau_1, ..., \tau_n)$, where τ_i is 1 if $i \in \{i_1, ..., i_k\}$, and 0 otherwise. Introducing tests allows us to construct different types of classifiers for the purpose of pattern recognition.

For a given test $\bar{\tau}$ we can measure the degree of similarity of an unknown pattern $\bar{t}$ to the training pattern $\bar{x}$ by

$$\prod_{i=1}^{n}(1 - \tau_i|t_i - x_i|) \qquad (1)$$

This expression takes the value 1 if and only if $\bar{t}$ and $\bar{x}$ coincide in the features defined by test $\bar{\tau}$, and takes the value 0 otherwise. In this case no metric is used and only an equivalence relation is required. The degree of similarity can also be measured in the classical way, using distance

$$d(\bar{\tau} \circ \bar{t}, \bar{\tau} \circ \bar{x}), \qquad (2)$$

where d is a distance metric, and the operation $\circ$ has the following meaning, $\bar{a} \circ \bar{b} = (a_1 \cdot b_1, ..., a_n \cdot b_n)$.

Denote T to be a set of tests. Taking (1) as a measure of similarity we calculate votes $V_1(\bar{t})$ and $V_2(\bar{t})$ for the classes I_1 and I_2 in the following way

$$V_1(\bar{t}) = \frac{1}{m_1} \sum_{\bar{\tau} \in T} \sum_{\bar{x} \in B_1} \prod_{i=1}^{n}(1 - \tau_i|t_i - x_i|)$$

$$V_2(\bar{t}) = \frac{1}{m_2} \sum_{\bar{\tau} \in T} \sum_{\bar{y} \in B_2} \prod_{i=1}^{n}(1 - \tau_i|t_i - y_i|).$$

We call a classifier based on the discriminant function $V(\bar{t}) = V_1(\bar{t}) - V_2(\bar{t})$ as *test feature classifier* [3] and denote it by TF_T. We extend TF to reject patterns $\bar{t}$ for which $V(\bar{t}) = 0$, and denote TF_T classifier as TFR_T for the rejection option.

Similarly, taking (2) as a measure of similarity and using the nearest neighbor concept we introduce a new classifier. For each test $\bar{\tau} \in T$ we calculate votes $W_1^{\bar{\tau}}(\bar{t})$ and $W_2^{\bar{\tau}}(\bar{t})$ for the classes I_1 and I_2 by the nearest neighbor rule, using (2) as a measure of similarity. Let

$$W_1(\bar{t}) = \frac{1}{m_1} \sum_{\bar{\tau} \in T} W_1^{\bar{\tau}}(\bar{t})$$

$$W_2(\bar{t}) = \frac{1}{m_2} \sum_{\bar{\tau} \in T} W_2^{\bar{\tau}}(\bar{t}).$$

We denote the classifier based on the discriminant function $W(\bar{t}) = W_1(\bar{t}) - W_2(\bar{t})$ by $TFNN_T$.

Denote the number of features (1s) in a test $\bar{\tau}$ as $|\bar{\tau}|$. We say $|\bar{\tau}|$ is the length of $\bar{\tau}$.

2.1. Performance

It is easy to prove that TF, TFR and $TFNN$ have no error on the training samples. As seen from the definition of the test feature classifier the classification performance of TF on the test samples depends on the set of tests T, and on the set of training samples B_1 and B_2. Let us call a test (a prime test) of I_1 and I_2 a *kernel* (a prime kernel). To provide the maximum classification rate for TF we need to find a set of kernels. If $\bar{\kappa}$ is a kernel then obviously $\bar{\kappa}$ is a test for B_1 and B_2 and the following relation holds

$$\bar{\kappa} \in \bigcap_{B_1 \in I_1, B_2 \in I_2} T \quad (3)$$

where T is the set of all tests for B_1 and B_2. In general, it is impossible to find a set of kernels for unknown I_1 and I_2, but we can estimate it from the training sets using relation (3). Suppose that a set of kernels K for unknown I_1 and I_2 is found. It is easy to see that TFR_K has a 100% recognition rate on any test samples for any training set. Even if we know a kernel, we need an appropriate training set to obtain a recognition rate of 100% for TF classifiers.

We say that pair (B, K), $B = B_1 \cup B_2$, covers a pattern $\bar{z}$ if there exist $\bar{x} \in B$ and $\bar{\kappa} \in K$ such that $\bar{z} \cdot \bar{\kappa} = \bar{x} \cdot \bar{\kappa}$. Denote by $C(B, K)$ the set of all $\bar{z}$ that are covered by (B, K). We call a set B as a *prototype* set for K if $C(B, K) \supseteq I_1 \cup I_2$. It is easy to see that if B is a prototype set for K then TF_K will have a 100% recognition rate on any test samples.

Let us describe two important properties of coverings. If $K_1 \subseteq K_2$ then $|C(B, K_1)| \leq |C(B, K_2)|$. If $|\bar{\kappa_1}| \leq |\bar{\kappa_2}|$ and $\bar{\kappa_1} \cdot \bar{\kappa_2} = \bar{\kappa_1}$, then $|C(B, \bar{\kappa_2})| \leq |C(B, \bar{\kappa_1})|$. From these properties we can conclude that in general to achieve a high recognition rate for test feature classifiers we should select a set of short tests using (3), and the more short tests we have the better the recognition rate is.

The approach with kernels is suitable when the set of all kernels is different from the trivial set $\{(1, 1..., 1)\}$. Below, we estimate the number of classes which have non-trivial sets of kernels.

Denote $\hat{T}$ to be the set of all prime tests. Denote $\hat{T}_i, i = 1, ..., n$, to be the set of all prime tests containing the i-th feature. Let $p_i = |\hat{T}_i|/|\hat{T}|$. We call p_i an info weight [3]. The classes I_1 and I_2 can be represented as a Boolean partial function $f(x_1, ..., x_n)$ such that $f(\bar{x}) = 0$ when $\bar{x} \in I_1$, and $f(\bar{x}) = 1$ when $\bar{x} \in I_2$. The ith variable of f is called an essential variable if there exists $(a_1, ..., a_n)$ and $(b_1, ..., b_n)$ such that $a_1 = b_1, ..., a_{i-1} = b_{i-1}$, $a_i \neq b_i$, $a_{i+1} = b_{i+1}, ..., a_n = b_n$ and $f(a_1, ..., a_n) \neq f(b_1, ..., b_n)$. It can be easily proved [3] that the ith variable of f is an essential variable if and only if the info weight $p_i = 1$.

Denote $S(n)$ to be the number of all partial functions with at least one essential variable defined on m number of n-tuples. Denote $S_{m_1 m_2}(n)$ to be the number of all partial functions with at least one essential variable that takes the value 0 m_1 times, and the value 1 m_1 times. Let $m = m_1 + m_2$. First, we estimate $S_{m_1 m_2}(n)$.

A set of $m_1 - 1$ number of n-tuples $\bar{\alpha_1}, ..., \bar{\alpha}_{m_1 - 1}$ with the property $f(\bar{\alpha_i}) = 0$ can be selected in $\binom{2^n}{m_1 - 1}$ different ways. From the remaining $2^n - (m_1 - 1)$ tuples, a set of $m_2 - 1$ number of n-tuples $\bar{\beta}_1, ..., \bar{\beta}_{m_2 - 1}$ with the property $f(\bar{\beta}_i) = 1$ can be selected in $\binom{2^n - (m_1 - 1)}{m_2 - 1}$ different ways. Suppose that the remaining two tuples (where f is defined) guarantee the essential dependence on one of the variables $x_1, ..., x_n$. The essential variable x_i can be selected in n different ways. Let $\bar{\gamma}$ be an n-tuple from the remaining $2^n - (m_1 + m_2 - 2)$ tuples. $f(\bar{\gamma})$ can be equal to 0 or 1. Since x_i is an essential variable then the last remaining $(m_1 + m_2)$th n-tuple $\bar{\delta}$ will be determined uniquely. Thus

$$S_{m_1 m_2}(n) \leq \binom{2^n}{m_1 - 1} \binom{2^n - (m_1 - 1)}{m_2 - 1} \times 2n(2^n - (m_1 + m_2 - 2))$$

Note that the above expression is an upper bound, since we take into account even impossible cases when for example $\bar{\delta}$ is already selected between $(m_1 + m_2 - 2)$ n-tuples.

Denote $R(n)$ to be the number of all partial functions defined on m tuples. It is easy to see that $R(n) = \binom{2^n}{m} 2^m$. The following lemma gives asymptotic growth ratios of $S(n)$ and $R(n)$ in the cases, when the domain of partial functions consists of n^k tuples, $k > 1$, i. e., $m_1 + m_2 = n^k$.

Lemma.

$$\lim_{n \to \infty} \frac{S(n)}{R(n)} \to 0.$$

Proof. Without loss of generality we consider $k = 2$.

$$\frac{S(n)}{R(n)} \leq$$

$$\frac{\sum_{m_1 + m_2 = n^2} 2n(2^n - (m_1 + m_2 - 2)) \binom{2^n}{m_1 - 1} \binom{2^n - m_1 + 1}{m_2 - 1}}{\binom{2^n}{n^2} 2^{n^2}} = \frac{2n(2^n - n^2 + 2)) \sum_{m_1 + m_2 = n^2} \binom{2^n}{m_1 - 1} \binom{2^n - m_1 + 1}{m_2 - 1}}{\binom{2^n}{n^2} 2^{n^2}}.$$

Since

$$\sum_{m_1 + m_2 = n^2} \binom{2^n}{m_1 - 1} \binom{2^n - m_1 + 1}{m_2 - 1} =$$

$$\sum_{m_1=0}^{n^2} \binom{2^n}{m_1 - 1}\binom{2^n - m_1 + 1}{n^2 - m_1 - 1} \le$$

$$n^2 \binom{2^n}{n^2/2 - 1}\binom{2^n - n^2/2 + 1}{n^2/2 - 1},$$

it follows that

$$\frac{S(n)}{R(n)} \le \frac{2n(2^n - n^2 + 2)n^2 \binom{2^n}{n^2/2-1}\binom{2^n - n^2/2+1}{n^2/2-1}}{\binom{2^n}{n^2} 2^{n^2}} =$$

$$\frac{2n^3(n^2)!}{2^{n^2}(2^n - n^2 + 1)((n^2/2 - 1)!)^2}.$$

From the well known Stirling formula we have

$$\frac{(n^2)!}{2^{n^2}} \sim \frac{\sqrt{2}(n^2/2)!(n^2/2)!}{\sqrt{\pi} n}.$$

Thus,

$$\frac{S(n)}{R(n)} \preceq C \frac{n^3((n^2/2)!)^2}{n(2^n - n^2 + 1)((n^2/2 - 1)!)^2} =$$

$$C \frac{n^3(n^2/2)(n^2/2)}{n(2^n - n^2 + 1)} = C' \frac{n^6}{2^n - n^2 + 1}.$$

Since $2^{(1-\epsilon)n} \preceq 2^n - n^2 + 1$ then

$$\frac{S(n)}{R(n)} \preceq C' \frac{n^6}{2^{(1-\epsilon)n}} \to 0. \qquad \square$$

From the lemma we can conclude that almost all functions (with a domain of n^k tuples, $k > 1$) do not have essential variables. Therefore, for each ith feature the info weight $p_i \neq 1$ and the set of kernels is non-trivial. Thus, in the cases when $|I_1 \cup I_2| \sim n^k, k > 1$, (these are cases which we encounter in reality when n becomes large) almost all classes have a non-trivial set of kernels and if we are able to select a set of kernels (from a set of tests) we will have a 100% recognition rate.

3. Experiments on Phoneme database

We applied the proposed classifiers to the phoneme data (Table 1.a) which is available via ftp at: ftp.dice.ucl.ac.be. The classification problem of the phoneme data is to distinguish between nasal and oral vowels. There are 5404 samples in this database. The phoneme database is known to be a difficult classification problem [4]. The dispersion matrix computed on the phoneme databese is given at Table 1.b, and Fischer's coefficient is equal to 0.0756. As seen from dispersion matrix and from value of Fischer's coefficient classes are very overlapped.

Num	Features	Class
1	1.239670 0.874530 -0.20510 -0.078137 0.066867	0
2	0.268281 1.351780 1.035080 -0.331522 0.216897	0
.....	......	
5404	0.136604 0.714084 1.349810 0.972467 -0.630074	1

a) Phoneme database.

Class	0	1
0	0.0	0.62
1	0.62	0.0

b) Dispertion matrix computed on the phoneme database.

K-Nearest Neighbor	87.76%
Neural Network	79.21%
C4.5 Decision Tree	83.92%
Quadratic Bayes	75.41%
Linear Bayes	73.00%

c) The best results for each classifier.

Table 1.

There meny papers related to the phoneme data recognition problem (see [4]). In Table 1.c we list the results of the different classifiers, which were obtained in [5]. Each classifier was optimized with respect to selecting "good" values for the parameters which govern its performance. The database was divided into two equal halves. One half was used as training set and the classification accuracy was then evaluated using the other half. The recognition results in Table 1 show that conventional methods incounter difficultes on large complex data set like phoneme.

To apply TF to the phoneme data features were rounded to two decimal places, Table 2.a (two decimal places is a minimal representation of the features by which patterns from different classes can be distinguished), and we used a binary representation of them. We divided the phoneme database into equal halves for training and testing, keeping same conditions as in [5]. In Table 2.b we show TFs recognition rates, where T_i is a set of randomly chosen short prime tests (with no kernels) of training samples, and T_i^* is a set of kernels. Even when a lot of information is lost due to the rounding feature space, recognition rate of TFs are much higher than conventional ones. TFs can achieve high recognition rate even with no kernels. The second half of Table 2.b shows what would happen if we were able to select even a small part of kernels. The more short kernels we have the better is the performance of TF and the less rejections we obtain for TFR.

Num.	Features	Class
1	1.24 0.87 -0.21 -0.08 0.07	0
2	0.27 1.35 1.04 -0.33 0.22	0
....	....	
5404	0.14 0.71 1.35 0.97 -0.63	1

a) Phoneme database with rounded features.

TF $_{T_1}$ $\lvert T_1 \rvert = 400$	89.32%
TFR $_{T_1}$	89.52%, 176 rejected
TFNN $_{T_1}$	95.65%
TF $_{T_2}$ $\lvert T_2 \rvert = 800$	90.27%
TFR $_{T_2}$	90.34%, 125 rejected
TFNN $_{T_2}$	95.58%
TF $_{T_1^*}$ $\lvert T_1^* \rvert = 400$	96.41%
TFR $_{T_1^*}$	100%, 728 rejected
TF $_{T_2^*}$ $\lvert T_2^* \rvert = 800$	99.09%
TFR $_{T_2^*}$	100%, 309 rejected

b) Results for TF classifiers on the phoneme database with rounded features.

Table 2.

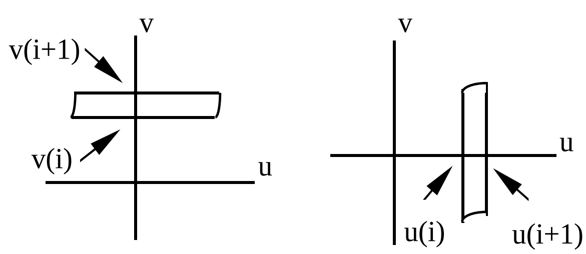

Figure 1. Fourier transform feature masks.

4. Textual Region Locating

In this section, we describe an application of the test feature classifier to the problem of textual region location. Our approach is to find all text areas and as few spurious non-text areas as possible, without actually classifying the characters.

At the beginning, the input is divided into 64×64 size blocks. For each block, the Fourier spectrum is calculated and the feature extraction procedure is applied.

4.1. Feature Extraction

As we mentioned in the introduction the main characteristic of text is its regularity. Our intention is to express these regularities in a feature vector and then apply test feature classifiers. By considering text regions as having a certain texture, textural feature extraction methods can be useful for the text regularity extraction.

Texture plays an important role in image analysis and understanding. Many papers have been devoted to the area of textural feature extraction and a number of various approaches have been proposed [10, 11]. A two-dimensional power spectrum of a texture image is often employed for texture feature extraction [12, 13]. Our choice of the power spectrum features is dictated by the following reasons. First, a power spectrum is commonly used for the selection of periodic information in the original spatial-domain image [13], and Fourier transform based methods usually perform well on textures showing periodicity [12]. Second, we wanted to have a robust system. Therefore, we chose to avoid methods based on spatial intensity measures. Such methods require many parameters and threshold values that must be set empirically and which can be tuned only for individual classes of images [6].

A peak in the power spectrum image represents any periodic structure of the original spatial-domain image. Therefore finding the peaks of the power spectrum is important for the texture description. The power spectrum $M(u, v)$ is examined in specific regions to isolate image features. Fig. 1 defines regions for the following Fourier features:

$$c_1(S) = \sum_{(u,v)\in S} M_1(u, v)$$

$$c_2(S) = \sum_{(u,v)\in S} M_2(u, v).$$

and $c_3(S)$, $c_4(S)$ which are standard deviations of the values $u'_i - u'_{i-1}$ and $v'_i - v'_{i-1}$ correspondingly, where $M_1(u, v) = M(u, v)$ when $M(u, v)$ is a local maximum and $M_1(u, v) = 0$ otherwise. (u', v') are the coordinates of these maxima. $M_2(u, v)$ is the difference between $M_1(u, v)$ and the minimal value of the middle of the neighborhood valleys when $u = u', v = v'$, and 0 otherwise.

The first feature c_1 expresses the energy of peaks in the region and is expected to be high for the text components images. By calculating c_2, c_3 and c_4 we attempt to identify the regularities of the image. We assume that the above features can implicitly express the regularities of the text.

4.2. Experimental Results

In order to demonstrate the performance of the proposed algorithm, the following simulations were performed on a database of images. The database consisted of various outdoor images including images with non-uniform illumination, images where characters had

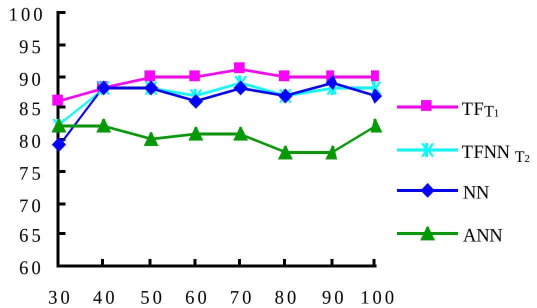

Figure 2. Recognition rates of TF, $TFNN$, NN, and ANN on test samples.

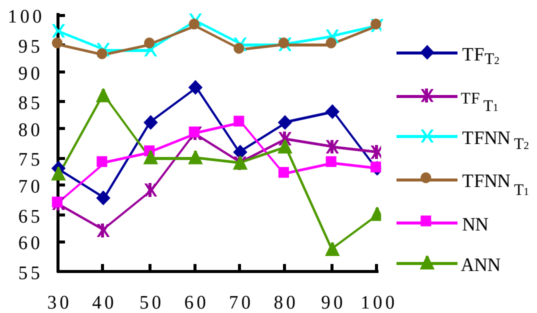

Figure 3. Recognition rates of TF, $TFNN$, NN, and ANN on test samples for the text component.

blurred boundaries, images with characters of different color components, and images with high variance background. The database was divided into two sets of data, one for training and the other for testing. In the experiment, we varied the number of randomly chosen training samples from 30 to 100, and more than 650 blocks were tested. The number of features used in experiments was 24. To use real-valued features in TF, we quantized the feature space into the five nonuniform intervals and then binary encoded them. The intervals used in experiments were $[0, \min], [\min, m-\sigma/2], [m-\sigma/2, m+\sigma/2], [m+\sigma/2, \max], [\max, \infty]$, where m and σ denote the average and the standard deviation value of a feature for the text component training set, respectively.

On a given training set, we used the sets of randomly chosen prime short tests T_1 and T_2, where $|T_1| = |T_2| = 800$.

The proposed TF and $TFNN$ classifiers were compared with two of the most popular non-parametric classifiers, a single nearest neighbor (NN) classifier, and a backpropagation artificial neural network (ANN). The neural network consisted of one hidden layer (nets with two hidden layers learned badly and were excluded from the experiment), and the number of hidden nodes ranged from 24 to 100. In order to reduce the randomness of the ANN performance due to different initial conditions, ten networks with different initial weights were trained. The ANN with the maximum performance was selected.

Results of the TF_{T_1}, $TFNN_{T_1}$, NN and ANN performance on test samples are shown in Fig. 2. It can be seen that even when information is lost due to the quantizing feature space, recognition rates of the proposed TF are higher than those of NN and ANN, which were applied directly to the real data.

As mentioned earlier, the main purpose of text locating method is to find all text areas and as few spurious non-text areas as possible. This means that for the estimation of a classifier, not only the overall recognition rate, but also the recognition rate for the text component is very important. Because non-text objects can be subsequently rejected by an OCR system, for a text region locating system it is important to have minimal loss on text regions even at the cost of increasing the number of non-text areas. Fig. 3 shows the recognition rates of TF , $TFNN$, NN and ANN for the text component. As we can see the performance of $TFNN$ is very stable and high, whereas all other classifiers have very unstable and low performance. Although all training samples in training sets and sets of tests T_1 and T_2 were selected randomly, $TFNN$ shows surprisingly stable and high performance. This demonstrates that the proposed method is robust and efficient for text region locating. Although TF has the best overall performance (Fig.2), the results in Fig. 3 suggest that $TFNN$ is the most suitable classifier for the purpose of textual region recognition.

Fig. 4 contains examples of test images and results of the classification by $TFNN$. Method gives good results on images where neither connected component method nor spatial variance based method can be applied.

5. Conclusions

The classifiers that have been presented are simple and robust. Optimization methods, statistical, structural or metrical characteristics of patterns are not required. We discuss the properties and performance of the proposed classifiers. Experiments on real data show that TFs can achieve a high recognition rate even with no kernels. Especially high and stable performance is shown by the $TFNN$ classifier which give almost a 96% of recognition rate on the phoneme data.

Based on test feature classifiers we also introduced a new method for text region locating. The proposed textual locating method is robust to variations in font,

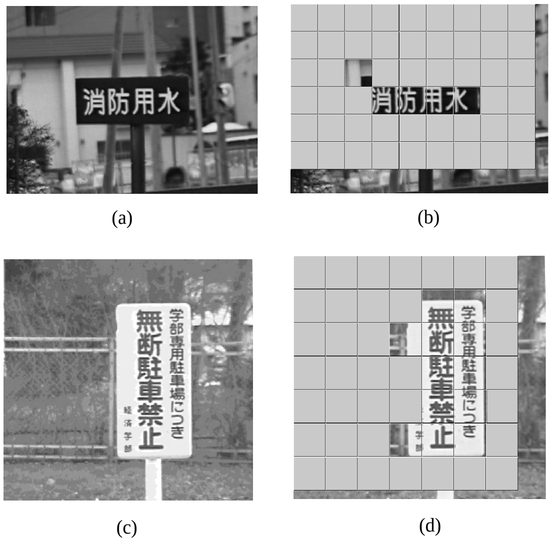

Figure 4. Recognition results of some test images. (a) and (c) test images; (b) and (f) results of applying $TFNN$ to (a) and (c).

color, size, or orientation of the text. It does not requires parameters and threshold values that must be set empirically as in previous approaches and allow us to avoid many of drawbacks of the popular connected component method and the spatial variance method. The performance of proposed classifiers as a function of various parameters were compared to the performance of the conventional ones. The proposed $TFNN$ classifier yields surprisingly stable and high recognition rates for text components.

Experimental results provide evidence of the practical utility of the test feature classifiers and they are shown to have much better recognition ability than conventional methods.

References

[1] I. Chegis and S. Yablonsky, "Logical Methods for Controlling Electric Circuit Function", Proceedings of V. A. Steklov Inst. of Maths., Vol 51 (in Russian), 1958.

[2] Yu. Zhuravlov, A. Dmitriev and F. Krendelev, "Mathematical Principles of the classification of Objects and Scene", Discrete Analyse, Vol 7 (in Russian), 1966.

[3] S. V. Aleshin, "Raspaznavanie Dinamicheskix Obrazov", Moscow University Press, (in Russian) 1996.

[4] Enhanced Learning for Evolutive Neural Architecture, ESPRIT Basic Research Project, No. 6891, 1995.

[5] K. Woods, W. Kegelmeyer, and K. Bowyer: "Combination of Multiple Classifiers Using Local Accuracy Estimates", IEEE Trans. PAMI-19, 1997.

[6] Y. Zhong, K. Karu and A. K. Jain: Locating text in complex color images, Pattern Recognition. Vol. 28. No. 10, 1995.

[7] B. Yu, A. Jain and M. Mohiuddin, Address block location on complex mail pieces, Proc. of the 4th Int. Conf. on Document Analysis and Recognition, 897-901,1997.

[8] J. Ohya, A. Shio and S. Akamatsu: Recognizing characters in scene images, IEEE Trans. PAMI-16, 1994.

[9] A. K. Jain and B. Yu, Automatic text location in images and video frames, IAPR ICPR98, 1497-1499, 1998.

[10] R. M. Haralic, K. Shanmugam and I. Dinstein, Textural features for image classification, IEEE trans. Syst. Man Cybernetics 3, 610-621, 1973.

[11] C. M. Wu and Y. C.Chen, Statistical feature matrix for texture analysis, CVGIP: Graphical Models Imag Proc. 54, 407-419, 1992.

[12] S. S. Liu and M. E. Jernigan, Texture analysis and discrimination in additive noise, Computer Vision Graphics Image Proc. 49, 52-67, 1990.

[13] J. C. Russ, The Image Processing Handboo, CRC Press, 1995.