Handwritten Digit Recognition Using SVM Binary Classifiers and Unbalanced Decision Trees

Adriano Mendes Gil[1], Cícero Ferreira Fernandes Costa Filho[2](✉),
and Marly Guimarães Fernandes Costa[2]

[1] Instituto Nokia de Tecnologia, Manaus, Brazil
adrianomendes.gil@gmail.com
[2] Universidade Federal do Amazonas/Centro de Pesquisa e Desenvolvimento em Tecnologia Eletrônica e da Informação , UFAM/CETELI, Manaus, Brazil
{ccosta,mcosta}@ufam.edu.br

Abstract. In this work, we use SVM binary classifiers coupled with a binary classifier architecture, an unbalanced decision tree, for handwritten digit recognition. According to input variables, two classifiers were trained and tested. One using digit characteristics and the other using the whole image as input variables. Developed recently, the unbalanced decision tree architecture provides a simple structure for a multiclass classifier using binary classifiers. In this work, using the whole image as input, 100% handwritten digit recognition accuracy was obtained in the MNIST database. These are the best results published in the literature for the MNIST database.

Keywords: Handwritten digit recognition · MNIST database · Support vector machine · Unbalanced decision tree · Binary classifiers

1 Introduction

In recent decades, character recognition technology has been driven by the increasing demand of converting an enormous amount of printed or handwritten information to a digital format [1]. This conversion from paper to computer in the past required human operators who processed billions of checks, mail correspondence, etc. This process was time consuming and error prone, motivating the development of optical character recognition (OCR), a technique for reading data and recognizing one character after another. OCR is an important pattern recognition technique. There are vast amounts of historical, technical and economic documents only in a printed form. An OCR system drastically reduces cost of digitalizing them. There are some successful techniques for OCR implementation applied in digitalization of handwritten and mechanical printed texts, and musical scores.

Character recognition is a very difficult problem, due to the great variability in writing styles, in other words, wide interclass variability: the same character can be written in different sizes and orientation angles.

As shown if Fig. 1, an OCR system is comprised of certain steps: image acquisition – a color, gray level or binary image is acquired; pre-processing – image processing

A. Campilho and M. Kamel (Eds.): ICIAR 2014, Part I, LNCS 8814, pp. 246–255, 2014.
DOI: 10.1007/978-3-319-11758-4_27

techniques are applied to improve image quality; layout analysis – the text structure is understood to facilitate text interpretation; word segmentation in characters; classification – pattern recognition is employed for character recognition and post-processing – gather the recognized characters to obtain the original words (opposite for word segmentation).

In this work we focused attention only on the classification step of digit recognition. Table 1 provides details about some digit recognition studies published in the literature. The columns of this table include: database, input data, classifier used and results.

Concerning input characteristics, the studies can be divided into two main groups: the first group consists of studies using digit extracted characteristics as input data [5,6,9,10,11] and the second one consists of studies using the whole image as input data [2,3,4,7,8].

Concerning the databases used, the studies shown in Table 1 can be divided into four groups: MNIST database, proprietary databases, CENPARMI database and NIST-SD19 database.

For performance comparison between different studies it is necessary that a common database be used for all them. In this work the MNIST handwritten digits database is adopted as the common database [12]. This database is suited for training and testing digit recognition algorithms and consists of 60,000 training patterns and 10,000 testing patterns. The patterns were obtained from 250 different authors. One digit is centralized in a gray level figure with 20x20 pixel size. This database presents two advantages: the digits need not be pre-processed and it is extensively used in the literature, enabling a performance comparison between different algorithms.

Consulting the web site of MNIST database, it can be verified that a total of 68 classifiers have been used for digit recognition [12]. The most used are: SVM, MLP and neural networks using convolutional algorithms.

In general, neural classifiers perform better than other classifiers. Convolutional algorithms have the best classifier performance. The best results for the accuracy in the classification step using convolutional algorithms, 99.73%, were obtained by Ciseran et al. [4]. In this study, the authors expanded the training and testing database, including elastic distortions. Deng [13] concluded that the use of distortion to expand the database is necessary to obtain high accuracy in digit classification. Studies that do not use distortion obtained low accuracy rates, varying between 99.47% and 99.65%.

Concerning the MNIST database and Table 1, it should also be noted that classifiers that use a whole image as input characteristics perform better than those that use digit characteristics as input.

Although impressive results for digit recognition using the MNIST database have been reported in the literature, this work focuses on improving state-of-the-art digit recognition, investigating the use of SVM.

In the literature, using SVM, the best results for digit recognition in the MNIST database, an accuracy of 99.44%, was obtained by Decoste and Scholkpf [2]. The authors employed a multiclass SVM classifier associated with the support virtual-vectors technique.

In this work, we intend to use SVM binary classifiers associated with a multiclass binary architecture, the unbalanced decision tree. According to input variables, two classifiers were trained and tested. One of them used digit characteristics and the other used the whole image as input variables.

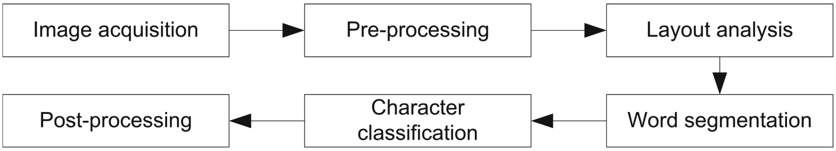


Fig. 1. Block diagram of an optical character recognition system

Table 1. A brief review of digit recognition

Reference	Database	Input data	Classifier	Results (accuracy)
[2]	MNIST Database	Whole image	SVM	99.44%
[3]	MNIST Database	Whole image	Combination of Convolutional Neural Networks	99.73%
[4]	MNIST Database	Whole image	Combination of Convolutional Neural Networks	99.77%
[5]	Proprietary Database	Fourier Descriptors + Border Transition Technique	MLP	96%
[6]	Proprietary Database	Fourier Descriptors	MLP + Models Previously Defined	90%
[7]	MNIST Database	Whole image	Perceptron	99.37%
[8]	Proprietary Database	Whole image	MLP	90%
[9]	Not cited	Hough Transform	MLP + Dempster-Shafer Theory	Not cited
[10]	NIST-SD19 Database	Kirsch Masks and Elliptic Fourier Descriptors	Combination of SVM Classifiers	98.55%
[11]	CENPARMI Database	Directional Distances	Modular Neural Networks	97.30%

2 Methods

2.1 Multiclass Binary Architectures

In both items 2.2 and 2.3, which address, respectively, the use SVM classifiers for digit recognition using digit characteristics and the whole image as input data, unbalanced decision trees, a type of multiclass binary architecture, is employed for digit recognition. So, in this item, we briefly review the different architectures of binary classifiers and, particularly, unbalanced decision trees.

According to Hassan and Demper [14], there are four different multiclass architectures using binary classifiers: one-against-rest, one-against-one, acyclic direct graph - ADG and unbalanced decision tree - UDT. Fig. 2 shows these architectures for a special case of four classes. In each one of these architectures, the output is the selection of only one class.

To distinguish between m classes, the architecture one-against-rest requires the training of m classifiers. Each classifier C_i is trained for recognizing class i. C_i returns a 1 if a given sample belongs to class i and 0 if a given samples does not belong to class i. It is only necessary to train m classifiers. When the training set is highly unbalanced, the performance of this architecture can be seriously affected. For a sample classification, m classifiers are used.

The one-against-one architecture uses the major voting rule. One sample is defined as belonging to class i if there are more votes for this class than for the others. A total of $m(m-1)/2$ binary classifiers are constructed, one for each different class pair. These classifiers are evaluated in parallel. Each classifier C_{ij} is trained using only samples of classes i and j. If a sample x is recognized by classifier C_{ij} as belonging to class i, a vote is assigned to class i. Otherwise, if it is recognized as belonging to class j, a vote is assigned to class j. After the sample is classified by all classifiers, the class that received more votes is considered the one to which the sample belongs. For a sample classification, $m(m-1)/2$ classifiers are used.

A set of binary classifiers can also be structured as an ADG. For this architecture, $m(m-1)/2$ binary classifiers are also necessary. In the architecture shown in Fig. 2, it can be observed that if the output of a classifier C_{ij} is class i, in the following node the class j is no longer considered a possible output class. This is why only $m-1$ classifiers are used for a pattern classification. Differing from the one-against-one architecture, only m-1 classifiers are evaluated to obtain a sample classification.

UDT was proposed by Ramanan et al. [15]. In each node, a decision is made regarding the type one-against-rest. Comparing with the architecture one-against-rest previously presented, this architecture uses only *m-1* classifiers. A sample classification begins in the node located on the top of the tree, using the classifier C_i. If the sample does not belong to class *i,* the decision process follows with the next right classifier of the tree. The classification process finishes when the sample is recognized as belonging to class n, by classifier C_n. As noted in Fig. 2, the lowest node of the tree decides only between two classes. According to Hassan & Damper [14], UDT follows a knockout strategy that, in the worst case, for a sample classification, requires $m-1$ classifiers. For a sample classification, on average, $(m-1)/2$ classifiers are used. Table 2 summarizes the main information of the four multiclass binary architectures. As shown, the UDT classifiers require a smaller number of classifiers both for training and classification. This is why in this paper we used SVM binary classifiers with a UDT multiclass architecture for digit recognition.

2.2 Digit Recognition Using Multiclass Binary Architecture with SVM Binary Classifiers and Digit Characteristics as Input Data

The block diagram of Fig. 3 shows a block diagram of the pattern recognition system used for digit recognition, using digit characteristics as inputs.

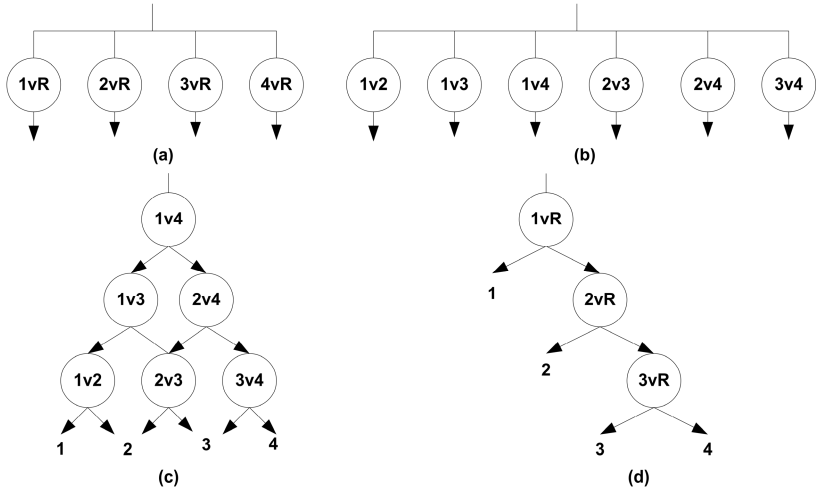


Fig. 2. Multiclass binary architectures: (a) one against rest; (b) one against one, (c) acyclic direct graph, (d) unbalanced decision tree

Table 2. Summary binary classifier architectures

Architecture	Number of classifiers	Classifiers used for a sample classification
one-versus-rest	*m*	*m*
one-versus-one	*(m*(m-1))/2*	*(m*(m-1))/2*
acyclic direct graph	*(m*(m-1))/2*	*m-1*
unbalanced decision tree	*m-1*	*(m-1)/2* *

* Average value

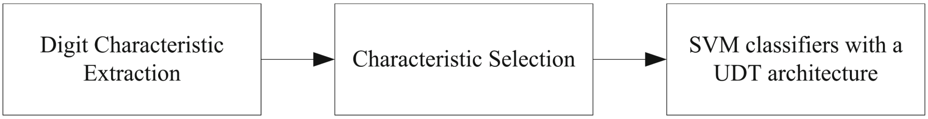

Digit Characteristic Extraction
Characteristic Selection
SVM classifiers with a UDT architecture

Fig. 3. Block diagram of a digit recognition system using multiclass binary architecture with SVM binary classifiers and digit characteristics as input data

Digit Characteristic Extraction. A set of 28 characteristics was used: twenty parameters corresponding to Fourier descriptors and eight parameters associated with border transition technique.

The twenty Fourier descriptors selected were the low frequency ones. The higher frequencies coefficients were discarded because they have insignificant values.

The border transition technique divides the digit image into four quadrants. For each quadrant, it calculates the transitions of pixel values from 0 to 1. In other words, a summation of the first order gradient in vertical and horizontal directions is done, totaling 8 parameters. In this work this complementary technique was used associated with Fourier descriptors, because the latter is invariant with rotation and displacement, impairing the distinction between ´6´and ´9´.

Characteristic Selection. Not all the 20 Fourier descriptors were used for classification. To select the best Fourier descriptors the scalar characteristic selection was used [16]. This is an "ad-hoc" technique that incorporates correlation information combined with criteria tailored for scalar characteristics. The procedure is divided into three parts. The first part is devoted to selecting only the first characteristic. The second part is devoted to selecting the second characteristic and the third part is used to select the other characteristics. In the first part, a class separability measure is selected and its value is computed for all the available characteristics. These values are ranked in descending order and the characteristic with higher value is chosen. In this paper, for this first part, a Fisher´s Discriminant Ratio (FDR) was used.

According to Theodoridis and Koutroumbas [16], FDR is sometimes used to quantify the separability capabilities of individual characteristics in a two-class problem, as is the case in this paper (pixels belong to bacillus or to background). FDR is defined as:

$$FDR = \frac{(\mu_1 - \mu_2)}{\sigma_1^2 + \sigma_2^2} \tag{1}$$

Where μ_1 and σ_{12} represent the mean value and standard deviation, respectively, of a characteristic in class ω_1; μ_2 and σ_{22} represent the mean value and standard deviation, respectively, of the same characteristic in class ω_2.

In the second and third parts, two other separability class measures are used: the divergence separability measure and the cross-correlation coefficient. The divergence measure between two classes ω_i and ω_j, for a given characteristic with mean value and standard deviation μ_i and σ_{i2} and μ_j and σ_{j2}, respectively, is defined as:

$$d_{ij} = \frac{1}{2}\left(\frac{\sigma_j^2}{\sigma_i^2} + \frac{\sigma_i^2}{\sigma_j^2} - 2\right) + \frac{1}{2}\left(\mu_i - \mu_j\right)^2\left(\frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2}\right) \tag{2}$$

To define the cross-correlation coefficient between two characteristics, let x_{nk}, n =1,2,….N and k=1,2,….m, be the kth characteristic of the nth pattern. The cross-correlation coefficient between any two characteristics is defined as [11]:

$$\rho_{ij} = \frac{\sum_{n=1}^{N} x_{ni} x_{nj}}{\sqrt{\sum_{n=1}^{N} x_{ni}^2 \sum_{n=1}^{N} x_{nj}^2}} \tag{3}$$

The second part selects x_{i_2} which

$$i_2 = \arg\max_j \{\alpha_1 \min_{i,j} d_{ij} - \alpha_2 \,|\, \rho_{i_1 j} \,|\}, \ \mathit{for} \text{ all } j \neq i \tag{4}$$

where α_1 and α_2 are weighting factors that determine the relative importance given to the two terms inside the brackets.

The third part selects x_{i_k}, k=3,...l, which

$$i_k = \arg\max_j \left\{ \alpha_1 \min_{\mathrm{i,j}} \mathrm{d}_{\mathrm{ij}} - \frac{\alpha_2}{k-1} \sum_{r=1}^{k-1} | \rho_{i_r\, j} | \right\} \tag{5}$$

With this technique, sets with the best 18, 17, 16, 15, 14, 13, 12 ,11, 10 and 9 Fourier descriptors were selected.

SVM Classifiers. Support vector machines (SVM) can be defined as binary learning machines used to separate data belonging to two classes using a hyperplane that maximizes the separation margin [17].

According to Theodoridis and Koutroumbas [16], for separable classes, the parameters of the hyperplane that maximize the margin are calculated through the determination of weight vector w and polarization w0, such that expression (6) is minimized and the Karush-Kuhn-Tucker (KKT) conditions are satisfied.

$$J(\mathbf{w}) \equiv \frac{1}{2} \| \mathbf{w} \|^2 \tag{6}$$

For nonseparable classes, the same parameters can be calculated minimizing expression (7), where new variables ξi, known as slack variables, are introduced. The goal now is to make the margin as large as possible but at the same time to keep the number of points with $\xi > 0$ as small as possible [16].

$$J(\mathbf{w}, w_0, \xi) = \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \xi_i \tag{7}$$

Parameter C in expression (7) is a constant positive that controls the tradeoff between the slack variable penalty and the margin. The value of the C parameter used in this work was 0.5.

SVMs use kernels to map the characteristic vector into a high dimensional space to exploit the nonlinear power of this tool. In this work, radial base function kernels were used, as shown in expression (8).

$$\exp(-\gamma \| \mathbf{x} - \mathbf{z} \|^2)^d, \gamma > 0 \tag{8}$$

3 Results

For SVM binary classifiers with an UDT architecture and digit characteristics as input data, the best results were obtained using the set of the best nine Fourier descriptors selected with the scalar selection technique, with the eight parameters obtained with a border transition technique, totaling 16 input variables for the SVM classifier. The nine best Fourier descriptors were the ones corresponding to the nine lower frequencies. For pattern classification, nine SVM binary classifiers were used with a UDT architecture, as shown in Fig. 4. Fig. 5 shows the confusion matrix obtained with the ORL test set. With the ORL training set, the accuracy was 85.27%. Table 3 shows the accuracy obtained for the ten digit classification.

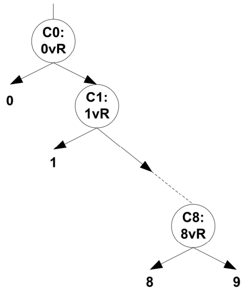
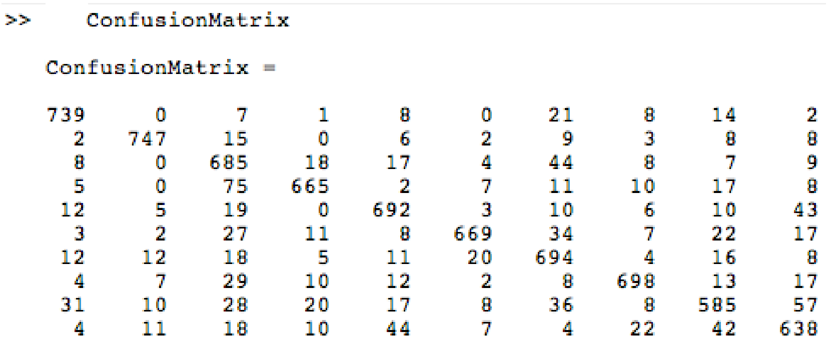


Fig. 4. UDT architecture used with SVM classifiers $C_0, C_1 \dots C_8$



```
>>    ConfusionMatrix

   ConfusionMatrix =

   739      0      7      1      8      0     21      8     14      2
     2    747     15      0      6      2      9      3      8      8
     8      0    685     18     17      4     44      8      7      9
     5      0     75    665      2      7     11     10     17      8
    12      5     19      0    692      3     10      6     10     43
     3      2     27     11      8    669     34      7     22     17
    12     12     18      5     11     20    694      4     16      8
     4      7     29     10     12      2      8    698     13     17
    31     10     28     20     17      8     36      8    585     57
     4     11     18     10     44      7      4     22     42    638
```

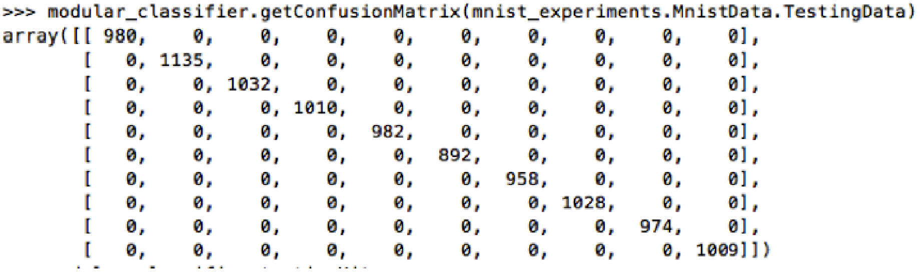
Fig. 5. Confusion matrix for multiclass binary architecture with SVM binary classifiers and digit characteristics as input data

For digit recognition using multiclass binary architecture with SVM binary classifiers and the whole image as input data, the number of inputs of each SVM classifier used in the UDT architecture (shown if Fig. 4) was 400, which corresponds to the pixels of an image with 20x20 pixels. Fig. 6 shows the confusion matrix obtained with the ORL test set. As shown in Fig. 6 no classification error occurred. So the obtained accuracy with the ORL test set was 100%. With the ORL training set, the accuracy was also 100%.

For SVM binary classifiers with an UDT architecture and digit characteristics as input data, the training time was 25h, while the answer time is about 1s. For SVM binary classifiers with an UDT architecture and the whole image as input data, the training time was 6h, while the answer time is less than 1s.

Table 3. Accuracy obtained for digit recognition using multiclass binary architecture with SVM binary classifiers and digit characteristics as input data

Digit	Accuracy	
	Training	**Test**
0	97.43%	93.50%
1	98.62%	96.56%
2	90.31%	85.31%
3	90.25%	91.44%
4	93.97%	90.50%
5	96.59%	93.87%
6	93.63%	91.63%
7	96.49%	92.94%
8	89.75%	89.56%
9	94.42%	90,28%
Mean Value	93.85%	91.56%



```
>>> modular_classifier.getConfusionMatrix(mnist_experiments.MnistData.TestingData)
array([[ 980,    0,    0,    0,    0,    0,    0,    0,    0,    0],
       [   0, 1135,    0,    0,    0,    0,    0,    0,    0,    0],
       [   0,    0, 1032,    0,    0,    0,    0,    0,    0,    0],
       [   0,    0,    0, 1010,    0,    0,    0,    0,    0,    0],
       [   0,    0,    0,    0,  982,    0,    0,    0,    0,    0],
       [   0,    0,    0,    0,    0,  892,    0,    0,    0,    0],
       [   0,    0,    0,    0,    0,    0,  958,    0,    0,    0],
       [   0,    0,    0,    0,    0,    0,    0, 1028,    0,    0],
       [   0,    0,    0,    0,    0,    0,    0,    0,  974,    0],
       [   0,    0,    0,    0,    0,    0,    0,    0,    0, 1009]])
```

Fig. 6. Confusion matrix for multiclass binary architecture with SVM binary classifiers and the whole image as input data (400 pixel values as input data)

4 Conclusion

Developed recently, UDT architecture provides a simple structure for a multiclass classifier using binary classifiers. In this work, the association of SVM binary classifiers with a UDT architecture, using the whole image as input, makes it possible to obtain an accuracy of 100% with handwritten digit recognition, using the MNIST database. This is the best recognition rate found in the literature for the MNIST database. As stated earlier, before this work, the best results for the accuracy in MNIST database was obtained by Ciseran et al. [4], 99.73%, using neural networks with convolutional algorithms.

The large number of support vectors used is a drawback of this approach. These vectors must be available at classification time, requiring nearly 1GB of memory.

The UDT architecture can be explored using any type of binary classifier, such as MLP and neural networks using convolutional algorithms, etc.

Acknowledgments. We thank AcademicEnglishSolutions.com for revising the English.

References

1. Cheriet, M., Kharma, N., Liu, C.L., Suen, C.: Character Recognition Systems. Wiley, New Jersey (2007)
2. Decoste, D., Schölkopf, B.: Training Invariant Support Vector Machines. Kluwer Academic Publishers, The Netherlands (2002)
3. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Convolutional Neural Network Committees for Handwritten Character Classification. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 1135 –1139 (2011)
4. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column Deep Neural Networks for Image Classification. Dalle Molle Institute for Artificial Intelligence. IDSIA/USI-SUPSI, Manno, Switzerland (2012)
5. Chung, Y.Y., Wong, M.T.: Handwritten character recognition by Fourier descriptors and neural network. In: IEEE Region 10 Annual Conference on Speech and Image Technologies for Computing and Telecommunications, vol. 1, pp. 391–394 (2007)
6. Poon, J.C., Man, G.M.: An enhanced approach to character recognition by Fourier descriptor. In: ICCS/ISITA 1992, vol. 2, pp. 558–562 (1992)
7. Kussul, E., Baidyk, T.: Improved method of handwritten digit recognition tested on MNIST database. In: 15th International Conference on Vision Interface, vol. 22, pp. 971–981 (2004)
8. Masmoudi, M., Samet, M., Taktak, F., Alimi, A.M.: A hardware implementation of neural network for the recognition of printed numerals. In: The Eleventh International Conference on Microelectronics, pp. 113–116 (1999)
9. Mandalia, A.D., Pandya, A.S., Sudhakar, R.: A hybrid approach to recognize handwritten alphanumeric characters. In: International Conference on System, Man and Cybernetics, vol. 1, pp. 723–726 (1992)
10. Travieso, C.M., Alonso, J., Ferrer, M.A.: Combining different off-line handwritten character recognizers. In: 15th International Conference on Intelligent Engineering Systems, Propad, pp. 315–318 (2011)
11. Oh, I.-S., Suen, C.Y.: A class-modular feedforward neural network for handwriting recognition. Pattern Recognition **35**(1), 229–244 (2002)
12. LeCun, Y., Cortes, C., Burges, C.J.: The MNIST database of Handwritten Digits, http://yann.lecun.com/exdb/mnist/ (accessed in January 02, 2014)
13. Deng, L.: The MNIST Database of Handwritten Digit Images for Machine Learning Research. IEEE Signal Processing Magazine, 141–142 (2012)
14. Hassan, A., Damper, R.I.: Classification of emotional speech using 3DEC hierarchical classifier. Speech Communication **54**, 903–916 (2012)
15. Ramanan, A., Suppharangsan, S., Niranjan, M.: Unbalanced Decision Trees for Multi-class Classification. In: International Conference on Industrial and Information Systems, Sri Lanka, pp. 291–294 (2007)
16. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Elsevier Academic Press, San Diego (2006)