Learning Iterative Strategies in Multi-Expert Systems Using SVMs for Digit Recognition

Donato Barbuzzi[1], Donato Impedovo[2],
Francesco Maurizio Mangini[1], and Giuseppe Pirlo[1]

[1] Department of Computer Science, University of Bari, Bari, Italy
{donato.barbuzzi,francescomaurizio.mangini,
giuseppe.pirlo}@uniba.it
[2] Department of Electrical and Electronic Engineering, Polytechnic of Bari, Bari, Italy
impedovo@deemail.poliba.it

Abstract. This paper presents three different learning iterative strategies, in a multi-expert system. In first strategy entire new dataset is used. In second strategy each single classifier selects new samples starting from those on which it performs a misclassification. Finally, the collective behavior of classifiers is studied to select the most profitable samples for knowledge base updating. The experimental results provide a comparison of three approaches under different operating conditions and feedback process. A classifier SVM and four different combination techniques were used by considering the CEDAR (handwritten digit) database. It is shown how results depend by the iterations on the feedback process, as well as by the specific combination decision schema and by data distribution.

Keywords: Feedback-based strategies, Instance Selection, Multi Expert Systems.

1 Introduction

While in the on-line handwriting recognition temporal and spatial information about each stroke is available, in off-line handwriting recognition only the image of the written character is used to perform the classification task. Indeed, off-line handwriting recognition is still considered a difficult problem that is only partially solved.

It has been observed, in the last few decades, that the accuracy of handwriting character recognition can be also improved by multiple expert fusion. The idea is not to rely on a single decision making scheme but to use several designs (experts) for decision making [1, 2, 3]. In fact, the collective behavior of a set of classifiers can convey more information that those of each classifier of the set, and this information can be exploited for classification aims [4, 5].

More specifically this paper proposes to select those samples, to be used for retraining specific experts of the set, misclassified by the multi-expert system [6, 7, 8, 9]. This approach is compared to situation in which the entire new dataset is used for

A. Petrosino (Ed.): ICIAP 2013, Part I, LNCS 8156, pp. 121–130, 2013.
© Springer-Verlag Berlin Heidelberg 2013

learning as well as the case in which specific samples are selected by the individual classifier.

Furthermore at the three standard retraining rules is computed an iteration each on the feedback process. A significant decrease in the error rate is reported considering the approach feedback-based ME.

Tests have been performed on the task of handwritten digit recognition, on the CEDAR database, by considering different types of features and a state of the art classifier (Support Vector Machine). Four different combination techniques (Majority Vote, Weighted Majority vote, Sum Rule and Product Rule) have been used between abstract and measurement level. It is shown how results depend by the feedback process, as well as by the specific combination decision schema and by data distribution.

The paper is organized as follows: Section 2 presents an overview of retraining rules and the different strategies feedback-based. Experiments and results are in Section 3 and 4, respectively. Section 5 reports a discussion and the conclusion of the work.

2 Learning Strategies

2.1 Related Work

When new labeled data became available, the following fundamental question arises: "how to use new data?". The simplest thing is probably to use the entire new set to update the knowledge base of each expert in the system, already trained and in its working phase. On the other hand, many interesting algorithms can be adopted in order to select (or focus the attention on) specific samples.

In particular, the algorithm AdaBoost [10, 11] is able to improve performance of a classifier on a given data set by focusing the learner attention on difficult instances. Even if this approach is very powerful, it works well in the case of weak classifiers, moreover not all the learning algorithms accept weights for the incoming samples. Another interesting approach is the bagging one: a number of weak classifiers trained on different subset (random instance) of the entire dataset are combined by means of the simple majority voting [1, 2]. Bagging and AdaBoost algorithms are adapted when considering a single classifier but applied to a ME system, them performance are boosted [12].

From these observations, specific strategies are depicted in the next paragraph taking into account of a multi-expert system that works in supervised learning.

2.2 Selecting Instances

Let be:

- C_j, for j=1,2,…,M, the set of pattern classes;
- $P = \{x_k \mid k = 1,2,...,K\}$, a set of pattern to be feed to the Multi Expert (ME) system. P is considered to be partitioned into S subsets $P_1,P_2,\ldots,P_s,\ldots,P_S$, being

$P_s=\{x_k \in P \mid k \in [N_s \cdot (s-1)+1, N_s \cdot s]\}$ and $N_s=K/S$ (N_s integer), that are fed one after the other to the multi-expert system. In particular, P_1 is used for learning only, whereas P_2, P_3,…,P_s,…,P_S are used both for classification and learning (when necessary);

- $y_s \in \Omega$, the label for the x_s pattern, $\Omega=\{C_1, C_2, ..., C_M\}$;
- A_i the *i-th* classifier for $i = 1,2,\ldots,N$;
- $F_i(k) = (F_{i,1}(k), F_{i,2}(k), \ldots, F_{i,r}(k), \ldots, F_{i,R}(k))$ the feature vector used by A_i for representing the pattern $x_k \in P$ (for the sake of simplicity it is here assumed that each classifier uses R real values as features);
- $KB_i(k)$, the knowledge base of A_i after the processing of P_k. In particular $KB_i(k) = (KB_i^1(k), KB_i^2(k), ..., KB_i^M(k))$;
- E the multi expert system which combines H_i hypothesis in order to obtain the final one.

In first stage (s=1), the classifier A_i is trained using the patterns $x_k \in P^*_i=P_1$. Therefore, the knowledge base $KB_i(s)$ of A_i is initially defined as:

$$KB_i(s)=(KB^1_i(s),KB^2_i(s),\ldots,KB^j_i(s),\ldots,KB^M_i(s)) \quad (1a)$$

where, for j=1,2,…,M:

$$KB^j_i(s)=(F^j_{i,1}(s),F^j_{i,2}(s),\ldots, F^j_{i,r}(s),\ldots, F^j_{i,R}(s)) \quad (1b)$$

being $F^j_{i,r}(s)$ the set of the *r-th* feature of the *i-th* classifier for the patterns of the class C_j that belongs to P^*_i.

Successively, the subsets $P_2,P_3,\ldots,P_s,\ldots,P_{S-1}$ are provided one after the other to the multi-classifier system both for classification and for learning. P_S is just considered to be the testing set in order to avoid biased or too optimistic results. When considering new labeled data (samples of $P_2,P_3,\ldots,P_s,\ldots,P_{S-1}$), a naïve and two not naïve strategies can be used.

The naïve strategy uses all the available new patterns to update the knowledge base of each individual classifier:

- $\forall x_t \in P_s : update\_KB_i$ (2)

Of course, in order to select patterns from Ps to train A_i, in the first strategy (not naïve) A_i is updated by considering all misclassified samples:

- $\forall x_t \in P_s \ \exists' A_i(x_t) \neq y_t : update\_KB_i$ (3)

The second approach (not naïve) is derived from AdaBoost and bagging. A_i is updated by considering all its misclassified samples if and only if these produce (or contribute to) a misclassification of the ME:

- $\forall x_t \in P_s \ \exists'\left(A_i\left(x_t\right) \neq y_t \wedge E\left(x_t\right) \neq y_t\right): update\_KB_i$ (4)

In order to inspect and take advantage of the common behavior of the ensemble of classifiers, the following simple strategy is evaluated and compared to the previous two.

3 Experiments

3.1 CEDAR Database

In the experimental session, a multi-expert system for handwritten digit recognition has been considered [6, 8] and the CEDAR Database of handwritten digits has been used [13]. In this case $P=\{x_k \mid k=1,2,\ldots,20351\}$ (classes from "0" to "9").

The DB has been initially partitioned into 6 subsets:

- $P_1=\{x_1,x_2,x_3,\ldots, x_{12750}\}$,
- $P_2=\{x_{12751},\ldots, x_{14119}\}$,
- $P_3=\{x_{14120},\ldots, x_{15488}\}$,
- $P_4=\{x_{15489},\ldots, x_{16857}\}$,
- $P_5=\{x_{16858},\ldots, x_{18223}\}$,
- $P_6=\{x_{18224},\ldots, x_{20351}\}$.

In particular, $P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5$ represent the set usually adopted for training when considering the CEDAR DB. P_6 is the testing dataset. Each digit is zoned into 16 uniform (regular) regions [14, 15], successively, for each region, the following set of features have been considered [16, 17, 18, 19]:

F_1: features set 1 (geometric features): hole, up cavity, down cavity, left cavity, right cavity, up end point, down end point, left end point, right end point, crossing points, up extrema points, down extrema points, left extrema points, right extrema points;

F_2: features set 2 (contour profiles): max/min peaks, max/min profiles, max/min width, max/min height;

F_3: features set 3 (intersection with lines): 5 horizontal lines, 5 vertical lines, 5 slant -45° lines and 5 slant +45° lines.

3.2 Setup

The classifier used for the experimentation is a SVMs (Support Vector Machines). This is a classifier that separates a given set of binary labeled training data with a hyper-plane that is maximally distant from them. For no linear separation, SVM can work in combination with the technique of "kernels", that automatically realizes a

non-linear mapping to a feature space. Here, for multi-class recognition, more binary SVM are performed and the kernel function adopted is the rbf gamma [3].

Also many approaches have been considered so far for classifiers combination. These approaches differ in terms of type of output they combine, system topology and degree of a-priori knowledge they use [1,2,3]. The combination technique plays a crucial role in the selection of new patterns to be feed to the classifier in the proposed approach. In this work the following decision combination techniques have been considered and compared: Majority Vote (MV), Weighted Majority Vote (WMV), Sum Rule (SR) and Product Rule (PR). MV just considers labels provided by the individual classifiers, it is generally adopted if no knowledge is available about performance of classifiers so that they are equal-considered. The second approach can be adopted by considering weights related to the performance of individual classifiers on a specific dataset. Given the case depicted in this work, it seems to be more realistic, in fact the behavior of classifiers can be evaluated, for instance, on the new available dataset. In particular, let ε_i be the error rate of the i-th classifier evaluated on the last available training set, the weight assigned to

$$\text{is, } w_i = \log(1/\beta_i) \text{ being } \beta_i = \varepsilon_i/(1-\varepsilon_i) \tag{5}$$

Sum Rule (SR) and Product Rule (PR) take into account the confidence of each individual classifier given the input pattern and the different classes [1]. Before the combination, confidence values provided by different classifiers were normalized by means of Z-score [20, 21].

4 Experimental Results

This section presents the results in terms of both error rate percentage (ER) and number of selected samples (SS), for each different learning strategies, obtained using binary images on CEDAR database. We combined, adopting a multi-expert system, the three set of features (F_1, F_2 and F_3) and a classifier SVM. Values of the similarity index (SI) [22, 23] are reported in the last row of each table.

The label "X-feed" refers to the use of the X modality for the feedback training process: "All" is the feedback of the entire set, "C" is feedback at classifier level. "MV", "WMV", "SR", "PR" are feedback at ME level adopting, respectively, the majority vote, the weighted majority vote, the sum rule and the product rule schema.

Tables 1, 2 and 3 show results related to the use of SVM classifier. The three set of features F_1, F_2 and F_3 are represented here as: SVM_1, SVM_2 and SVM_3. P_1 is used for training and P_6 for testing. $P_2 \cup P_3 \cup P_4 \cup P_5$ is used for feedback learning. The first column (No-feed) reports results related to the use of P_1 for training and of P_6 for testing, without applying any feedback (0 Selected Samples), while the approach All-feed uses all samples belonging to the new set in order to update the knowledge base of each single classifier (All Selected Samples). Depending by the combination technique, a specific strategy can outperform the others. More specifically, applying two iterations in the feedback process, WMV-feed, SR-feed and PR-feed, respectively, an

improvement of 0,05%, 0.09% and 0,05% compared to the use of the entire new data-set (All-feed), while for WMV-feed and SR-feed, respectively, an improvement of 0,10% and 0,09% compared to the use of the feedback at single expert level. While, iterating the feedback process in three steps, MV-feed, WMV-feed and SR-feed, respectively, improvement of 0.14%, 0.05% and 0.09% compared to All-feed and only SR-feed improvement of 0.04% compared to C-feed.

Finally using the feedback-based strategies at ME level, it is of interest the fact that a very restricted subset of samples are selected for retraining.

Table 1. SVM, Feedback - $P_2 \cup P_3 \cup P_4 \cup P_5$, One Iteration

	No-feed	**C-feed**		**MV-feed**		**WMV-feed**		**SR-feed**		**PR-feed**		**All-feed**	
	ER	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**
SVM_1	2.94	2.82	335	3.01	143	3.01	93	2.96	76	2.96	64	2.92	5466
SVM_2	8.37	8.13	572	8.36	167	8.55	117	8.22	91	8.22	72	7.79	5466
SVM_3	4.09	4.35	225	4.46	124	4.46	124	4.18	69	4.18	52	4.23	5466
MV	2.54	2.49		**2.35**		X		X		X		2.58	
WMV	**1.69**	1.83		X		1.79		X		X		1.74	
SR	1.46	1.41		X		X		**1.36**		X		1.41	
PR	1.22	**1.17**		X		X		X		1.22		**1.17**	
SI	91.29	91.30		90.98		90.91		91.32		91.24		91.55	

Table 2. SVM, Feedback - $P_2 \cup P_3 \cup P_4 \cup P_5$, Two Iterations

	No-feed	**C-feed**		**MV-feed**		**WMV-feed**		**SR-feed**		**PR-feed**		**All-feed**	
	ER	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**
SVM_1	2.94	2.82	642	3.05	327	2.87	327	3.01	197	3.01	159	2.68	10932
SVM_2	8.37	8.32	988	8.46	418	8.36	283	8.41	221	8.60	171	7.47	10932
SVM_3	4.09	4.23	1127	4.37	430	4.37	295	4.51	181	4.37	174	4.14	10932
MV	**2.54**	**2.54**		2.63		X		X		X		**2.54**	
WMV	1.69	1.74		X		**1.64**		X		X		1.69	
SR	1.46	1.50		X		X		**1.41**		X		1.50	
PR	**1.22**	**1.22**		X		X		X		**1.22**		1.27	
SI	91.29	91.29		91.02		91.12		90.99		90.85		92.04	

Table 3. SVM, Feedback - $P_2 \cup P_3 \cup P_4 \cup P_5$, Three Iterations

	No-feed	**C-feed**		**MV-feed**		**WMV-feed**		**SR-feed**		**PR-feed**		**All-feed**	
	ER	**ER**	**SS**	**ER**	**SS**	**ER**	SS	**ER**	**SS**	**ER**	**SS**	**ER**	**SS**
SVM_1	2.94	2.96	879	3.10	427	2.96	425	2.96	254	3.05	204	2.87	16398
SVM_2	8.37	8.04	1296	8.55	552	8.60	366	8.79	287	8.88	217	7.85	16398
SVM_3	4.09	4.23	1616	4.51	574	4.32	391	4.61	299	4.51	228	4.37	16398
MV	**2.54**	**2.54**		2.63		X		X		X		2.77	
WMV	**1.69**	1.74		X		1.74		X		X		1.79	
SR	**1.46**	1.50		X		X		**1.46**		X		1.55	
PR	1.22	1.22		X		X		X		1.27		**1.17**	
SI	91.29	91.35		90.85		91.02		90.68		90.66		91.57	

Figures 1, 2 and 3 represent the performances of the tables: Table 1, Table 2 and Table 3, respectively, show that in any case a combination technique exists that definitively outperforms the other two approaches, adopting an iteration on the feedback process.

In particular, the strategy WMV-feed, after two iterations on the feedback process, reduces the error rate both compared to one iteration and three iterations, respectively, of 0.15% and 0.10% and respect to other two feedback-based strategies C-feed and All-feed.

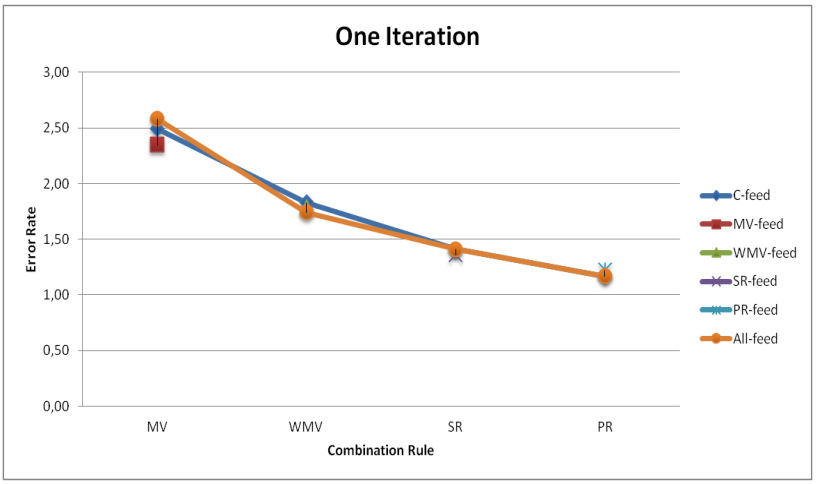

Fig. 1. SVM, Feedback - $P_2 \cup P_3 \cup P_4 \cup P_5$, One Iteration

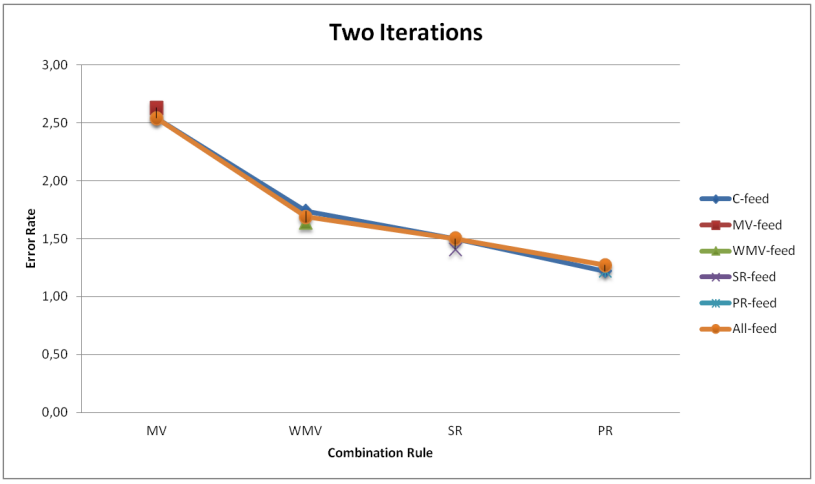

Fig. 2. SVM, Feedback - $P_2 \cup P_3 \cup P_4 \cup P_5$, Two Iterations

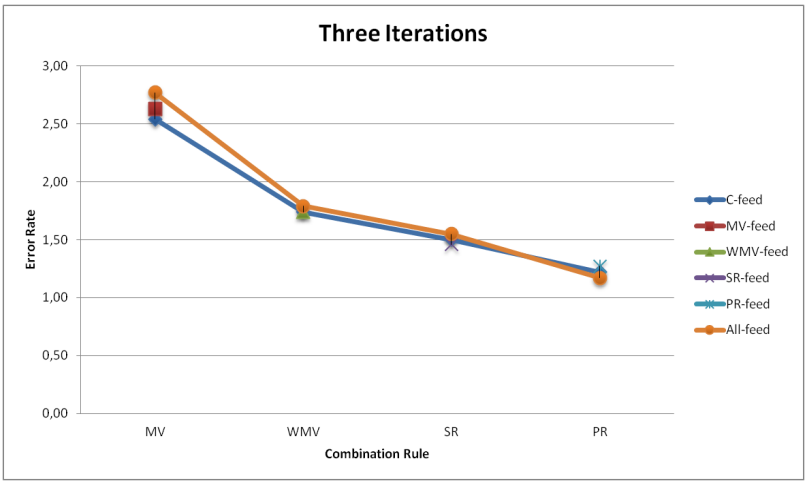

Fig. 3. SVM, Feedback - $P_2 \cup P_3 \cup P_4 \cup P_5$, Three Iterations

5 Discussion and Conclusion

This paper provides a comparison of three iterative learning strategies for multi-expert systems, when new labeled data become available. More precisely, the feedback-based strategies are used to the optimal way in which new selected samples must be used to update the knowledge base of the individual classifiers. For the purpose, four different combination techniques between abstract and measurement level strategies have been investigated under different operating conditions.

The experimental results have shown that performance of feedback-based training strictly depend by the iteration on the feedback process, by the combination strategy of the ME, but also by the data distribution and the similarity between samples in the feedback set and samples of the testing set. In particular, the not naïve strategy proposed in this paper (see eq. (4)) is able to select not only samples to be used for the updating process, but also the classifiers (see eq. (5)) to which those samples must be feed. Of course, considering initially trained classifiers, the multi-expert will return few instances for the retraining process if the classification performances on new data available are high. This can happen depending by performances of classifiers, by iterations on the feedback process as well as by the ratio new/old data. Especially, given a specific classifier, the difference between the confidence value in the case of misclassification and in the case of correct one could be imputed to the fact that the specific classifier (features, matching technique, etc.) is unable to represent it, and no improvements would be obtained by introducing the new sample in the knowledge base. This is particularly true under the assumption that strong (not weak) classifiers are used. Finally, the result shown that also when the cardinality of the new selected training set is negligible if compared to that of the initial training set, the feedback strategy is able to produce improvements.

Future work will inspect deeply the possibility of evaluate the approaches on the task of semi-supervised learning as well as in unsupervised learning [24, 25, 26].

References

1. Kittler, J., Hatef, M., Duin, R.P.W., Matias, J.: On combining classifiers. IEEE Trans. on PAMI 20(3), 226–239 (1998)
2. Suen, C.Y., Nadal, C., Legault, R., Mai, T.A., Lam, L.: Computer Recognition of unconstrained handwritten numerals. Proc. IEEE 80(7), 1162–1180 (1992)
3. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognition 36(10), 2271–2285 (2003)
4. Pirlo, G., Impedovo, D.: Fuzzy-Zoning-Based Classification for Handwritten Characters. IEEE Trans. on Fuzzy Systems 19(4), 780–785 (2011)
5. Suen, C.Y., Tan, J.: Analysis of errors of handwritten digits made by a multitude of classifiers. Pattern Recognition Letters 26(3), 369–379 (2005)
6. Impedovo, D., Pirlo, G.: Updating Knowledge in Feedback-based Multi-Classifier Systems. In: Proc. of ICDAR, pp. 227–231 (2011)
7. Barbuzzi, D., Impedovo, D., Pirlo, G.: Feedback-Based Strategies In Multi-Expert Systems. In: Sesto Convegno del Gruppo Italiano Ricercatori in Pattern Recognition (2012)
8. Impedovo, D., Pirlo, G., Barbuzzi, D.: Supervised Learning Strategies in Multi-Classifier Systems. In: Proceedings of 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2012), pp. 1215–1220 (2012)
9. Barbuzzi, D., Impedovo, D., Pirlo, G.: Benchmarking of Update Learning Strategies on Digit Classifier Systems. In: Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition, pp. 35–40 (2012)
10. Freud, Y., Schapire, R.E.: Decision-theoretic generalization of on-line learning and an application to boosting. J. of Computer and System Sciences 55(1), 119–139 (1997)
11. Schapire, R.E.: The strength of weak learnability. Machine Learning 5(2), 197–227 (1990)
12. Polikar, R.: Bootstrap-Inspired Techniques in Computational Intelligence. IEEE Signal Processing Magazine 24(4), 59–72 (2007)
13. Hull, J.: A database for handwritten text recognition research. IEEE T-PAMI 16(5), 550–554 (1994)
14. Impedovo, S., Modugno, R., Ferrante, A., Pirlo, G.: Zoning Methods for Hand-written Character Recognition: An Overview. In: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR 2012), Kolkata, India, November 16-18, pp. 329–334 (2010)
15. Impedovo, D., Modugno, R., Pirlo, G.: New Advancements in Zoning-Based Recognition of Handwritten Characters. In: Proc. XIII International Conference on Frontiers in Handwriting Recognition (ICFHR 2012), Monopoli, Bari, Italy, September 18-20, pp. 661–665 (2012)
16. Impedovo, S., et al.: Feature Membership Functions in Voronoi-Based Zoning. In: Serra, R., Cucchiara, R. (eds.) AI*IA 2009. LNCS, vol. 5883, pp. 202–211. Springer, Heidelberg (2009)
17. Impedovo, S., Modugno, R., Pirlo, G.: Analysis of Membership Functions for Voronoi-based Classification. In: Proceedings of the 12th Interational Conference on Frontiers in Handwriting Recognition (ICFHR 2012), November 16-18, pp. 220–225. IEEE Computer Society Press, Kolkata (2010)

18. Pirlo, G., Impedovo, D.: Adaptive Membership Functions for Hand-Written Character Recognition by Voronoi-based Image Zoning. IEEE Transactions on Image Processing 21(9), 3827–3837 (2012)
19. Impedovo, S., Pirlo, G.: Tuning between Exponential Functions and Zones for Membership Functions Selection in Voronoi-based Zoning for Handwritten Character Recognition. In: Proc. of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011), September 18-21, pp. 997–1001. IEEE Computer Society, Beijing (2011) ISBN: 978-0-7695-4520-2
20. Impedovo, D., Modugno, R., Pirlo, G.: Score Normalization by Dynamic Time Warping. In: Proceedings of the International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA), Taranto, Italy, September 6-8, pp. 82–85. IEEE Computer Society Press, Taranto (2010) ISBN: 978-1-4244-7229-1
21. Pirlo, G., Impedovo, D.: Adaptive Score Normalization for Multi-Classifier Systems. IEEE Signal Processing Letters 19(12), 837–840 (2012) ISSN: 1070-9908
22. Impedovo, D., Pirlo, G., Sarcinella, L., Stasolla, E.: Artificial Classifier Generation for Multi-Expert System Evaluation. In: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR 2012), November 16-18, pp. 42–426. IEEE Computer Society Press, Kolkata (2010) ISBN: 978-0-7695-4221-8
23. Bovino, L., Dimauro, G., Impedovo, S., Lucchese, M.G., Modugno, R., Pirlo, G., Salzo, A., Sarcinella, L.: On the Combination of Abstract-Level Classifiers. International Journal on Document Analysis and Recognition 6, 42–54 (2003) ISSN 1433-2833
24. Frinken, V., Bunke, H.: Evaluating Retraining Rules for Semi-Supervised Learning in Neural Network Based Cursive Word Recognition. In: Proc. of ICDAR, pp. 31–35 (2009)
25. Frinken, V., Fischer, A., Bunke, H., Fornes, A.: Co-Training for Handwritten Word Recognition. In: Proc. of ICDAR, pp. 314–318 (2011)
26. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training. In: ACM Proc. of COLT, pp. 92–100 (1998)