Classification System Optimization with Multi-Objective Genetic Algorithms

†Paulo V. W. Radtke[1,2], Robert Sabourin[1,2], Tony Wong[1]

[1]École de Technologie Supérieure - Montreal, Canada
[2]Pontifícia Universidade Católica do Paraná - Curitiba, Brazil
†e-mail: radtke@livia.etsmtl.ca

Abstract

This paper discusses a two-level approach to optimize classification systems with multi-objective genetic algorithms. The first level creates a set of representations through feature extraction, which is used to train a classifier set. At this point, the most performing classifier can be selected for a single classifier system, or an ensemble of classifiers can be optimized for improved accuracy. Two zoning strategies for feature extraction are discussed and compared using global validation to select optimized solutions. Experiments conducted with isolated handwritten digits and uppercase letters demonstrate the effectiveness of this approach, which encourages further research in this direction.

Keywords: Classification systems, feature extraction, ensemble of classifiers, multi-objective genetic algorithms

1. Introduction

Image-based *pattern recognition* (PR) requires that pixel information be first transformed into an abstract representation (a feature vector) suitable for recognition with classifiers, a process known as *feature extraction*. A relevant classification problem is the *intelligent character recognition* (ICR), most specifically the offline recognition of isolated handwritten symbols on documents. A methodology to extract features must select the spatial location to apply transformations on the image [1]. The choice takes into account the *domain context*, the type of symbols to classify, and the *domain knowledge*, what was previously done in similar problems. The process is usually performed by a human expert in a trial-and-error process. We also have that changes in the domain context may manifest in the same classification problem, which also requires changes in the classification system.

To minimize the human intervention in defining and adapting classification systems, this problem is modeled as an evolutionary *multi-objective optimization problem* (MOOP), using the domain knowledge and the domain context. This paper details the two-level genetic approach to optimize classification systems in Fig. 1. The first level employs the *Intelligent Feature Extraction* (IFE) methodology to extract feature sets that are used on the second level to optimize an *ensemble of classifiers* (EoC) to improve accuracy.

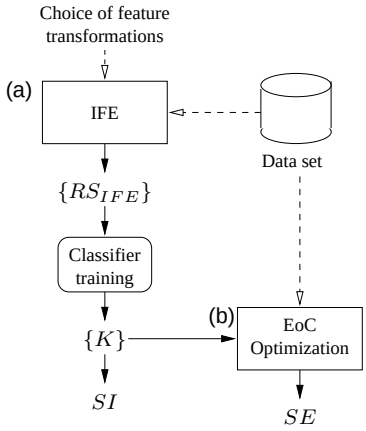


Figure 1. Classification system optimization approach. Representations obtained with IFE are used to further improve accuracy with EoCs.

This paper extends the work in [2]. New contributions lies in (1) the comparison of zoning operators for the IFE methodology with handwritten digits, and (2) the application of the most performing operator to optimize a classification system for uppercase letters. Another difference is the use of a *global validation* strategy [3] to select solutions during optimization. The global validation strategy improves average results obtained in comparison to the traditional validation approach used in [2]. The paper has the following structure. The approach to optimize classification systems is discussed in Section 2, and Section 3 discusses how the *multi-objective genetic algorithms* (MOGAs) were used. Section 4 details the experimental protocol and Section 5 presents the results obtained. Finally, Section 6 discusses the goals attained.

2. Classification System Optimization

Classification systems are modeled in a two-level process. The first level uses the IFE methodology to obtain the representation set RS_{IFE} (Fig. 1.a). The representations in RS_{IFE} are then used to train the classifier set K that is considered for aggregation on an EoC SE for

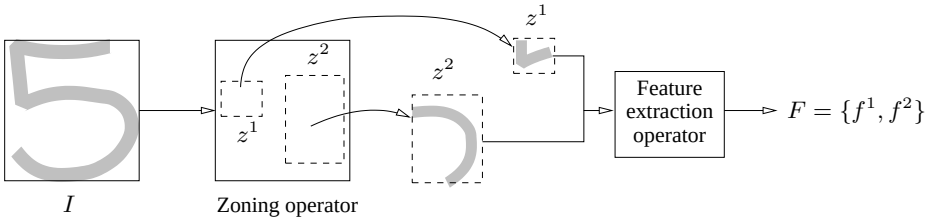


Figure 2. IFE structure.

improved accuracy (Fig. 1.b). Otherwise, if a single classifier is desired for limited hardware, such as embedded devices, the most accurate single classifier SI may be selected from K. The next two subsections details both the IFE and EoC optimization methodologies.

2.1. Intelligent Feature Extraction

The goal of IFE is to help the human expert define representations in the context of isolated handwritten symbols, using a wrapper approach with a fast training classifier. IFE models handwritten symbols as features extracted from specific *foci* of attention on images using *zoning*. Two operators are used to generate representations with IFE: a *zoning operator* to define foci of attention over images, and a *feature extraction* operator to apply transformations in zones. The choice of transformations for the feature extraction operator constitutes the domain knowledge. The domain context is introduced as actual observations in the *optimization* data set used to evaluate and compare solutions. Hence, the zoning operator is optimized by the IFE to the domain context and domain knowledge.

The IFE structure is illustrated in Fig. 2. The zoning operator defines the zoning strategy $Z = \{z^1, \ldots, z^n\}$, where $z^i, 1 \leq i \leq n$ is a zone in the image I and n the total number of zones. Pixels inside the zones in Z are transformed by the feature extraction operator in the representation $F = \{f^1, \ldots, f^n\}$, where $f^i, 1 \leq i \leq n$ is the partial feature vector extracted from z^i. At the end of the optimization process, the resulting representation set $RS_{IFE} = \{F^1, \ldots, F^p\}$ presents the IFE user with a choice among various trade-offs with respect to the optimization objectives.

The result set RS_{IFE} is used to train a discriminating classifier set $K = \{K^1, \ldots, K^p\}$, where K^i is the classifier trained with representation F^i. The first hypothesis is to select the most accurate classifier $SI, SI \in K$ for a single classifier system. The second hypothesis is to use K to optimize an EoC for higher accuracy, an approach discussed in Section 2.2. The remainder of this section discusses the IFE operators chosen for experimentation with isolated handwritten characters and the candidate solution evaluation.

2.1.1. Zoning Operators

Two zoning operators are compared, the *divider zoning operator* and the *hierarchical zoning operator*. Both are compared to a *baseline* representation with a high degree of accuracy on handwritten digits with a *multi-layer Perceptron* (MLP) classifier [4]. Its zoning strategy, detailed in Fig. 3.b, is defined as a set of three image dividers, producing 6 zones. The *divider zoning operator* expands the baseline zoning concept into a set of 5 horizontal and 5 vertical dividers that can be either *active* or *inactive*, producing zoning strategies with 1 to 36 zones. Fig. 3.a details the operator template, genetically represented by a 10-bit binary string. Each bit is associated with a divider's state (1 for active, 0 for inactive).

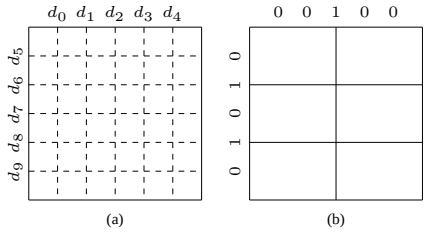


Figure 3. Divider zoning operator (a). The baseline representation in (b) is obtained by setting only d_2, d_6 and d_8 as active.

The *hierarchical zoning operator* is the second option, recursively defining a zoning strategy with the set of eight patterns in Fig. 4. Zones inside a root pattern are recursively partitioned with another pattern in the set, as illustrated in Fig. 5. This zoning strategy is described by the string *ba#eg*, where # is a pattern ignored in the root pattern.

For our experiments with the hierarchical zoning operator, only one level of recursion is allowed and a maximum of 16 zones can be defined. This choice is to avoid too small zones, close to pixel size, that would not contribute to classification. The operator is genetically encoded with a 15 bits binary string, where 5 patterns (one root plus four leaves) are encoded with 3 bits each. Unlike the divider zoning operator, the hierarchical zoning operator can not reproduce the baseline representation.

2.1.2. Feature Extraction Operator

Oliveira *et al.* used and detailed in [4] a mixture of concavities, contour directions and black pixel surface transformations, extracting 22 features per zone (13 for concavities, 8 for contour directions and 1 for surface). To allow a direct comparison between IFE and the base-

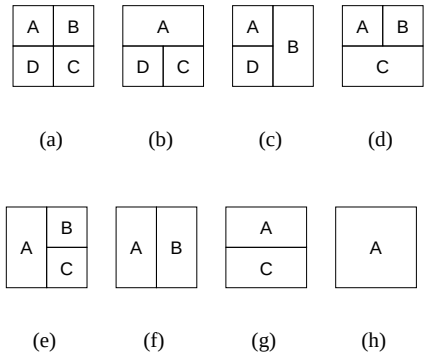


Figure 4. Hierarchical recursive patterns.

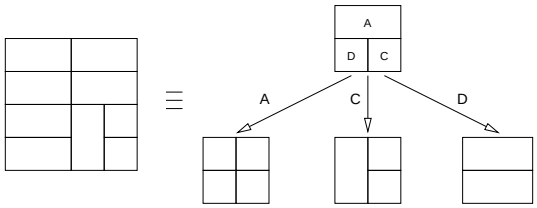


Figure 5. Hierarchical zoning example.

line representation, the same feature transformations (the domain knowledge) are used to assess the IFE.

2.1.3. Candidate Solution Evaluation

Candidate solutions are evaluated with respect to two objective functions, classification accuracy (*wrapper* mode) and cardinality. A lower representation dimensionality is associated to higher generalization power and to less processing time for feature extraction and classification. Thus, the objectives are to minimize both dimensionality (zone number) and the classification error rate on the *optimization* data set (the domain context).

The wrapped classifier needs to be computationally efficient and reasonably accurate to prototype IFE solutions. Kimura *et al.* discussed in [5] the *projection distance* (PD) classifier, which is fairly quickly to train and classify observations. Therefore, the PD classifier has been chosen to the IFE wrapper approach.

2.2. EoC Optimization

A recent trend in PR has been to combine several classifiers to improve their overall performance. Algorithms for creating EoCs will usually fall into one of two main categories. They either manipulate the training samples for each classifier in the ensemble (like Bagging and Boosting), or they manipulate the feature set used to train classifiers [6]. The key issue is to generate a set of diverse and fairly accurate classifiers for aggregation [7].

We create EoCs on a two-level process. The first level creates a classifier set K with IFE, and the second level optimizes the classifiers aggregated as a MOOP. We assume that RS_{IFE} generates a set K of p diverse and fairly accurate classifiers. To realize this task as a MOOP, the classifiers in K are associated with a binary string E of p bits, which is optimized to select the best combination of classifiers using a MOGA. The classifier K^i is associated with the i^{th} binary value in E, which indicates whether or not the classifier is active in the EoC.

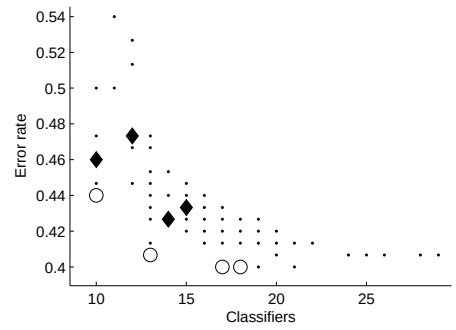
The optimization process is guided by two objectives, EoC cardinality and EoC quality. EoC cardinality is minimized to reduce classification time, and quality is measured through the combined classifier accuracy on the *optimization* data set, as discussed in [8]. The optimization goal is to minimize both EoC cardinality and the associated error rate on the *optimization* data set. Evaluating the EoC error rate requires actual classifier aggregation. The normalized continuous values of MLP outputs are aggregated by their output average [7]. To speed up the process, the MLP outputs are calculated once only and stored in memory for future aggregation. PD classifiers are aggregated by majority voting. As with MLP classifiers, PD votes are calculated once only and stored in memory.

3. Multi-Objective Genetic Optimization

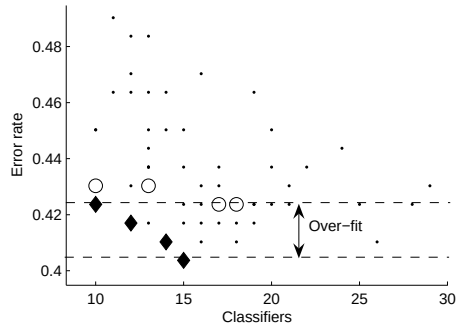
Two algorithms are used in the experiments. The first was designed for the IFE methodology, the *Multi-Objective Memetic Algorithm* (MOMA) [9]. The second algorithm is used for the EoC optimization, the *Fast Non-Dominated Sorting Genetic Algorithm* (NSGA-II) [10], a well known algorithm in the literature. MOMA was chosen over NSGA-II for the IFE methodology for it's higher solution diversity to optimize an EoC later. On the other hand, NSGA-II is chosen over MOMA for EoC optimization for being faster and producing comparable results.

IFE and EoC solutions obtained by the optimization algorithm may be over-fitted to the *optimization* data set. To avoid over-fit after the optimization process, resulting solutions are traditionally validated with a disjoint *selection* data set [11] to select the most accurate solution. However, our experiments indicate that a more robust validation process is needed. With NSGA-II to optimize an EoC, Fig. 6 details all individuals in the population at generation $t = 14$. Fig. 6.a is the objective function space used during the optimization process (*optimization* data set), and Fig. 6.b is the objective function space used for validation (with the *validation* data set). Points are candidate solutions in the current generation (MLP EoCs). Circles represent the best optimization trade-offs, and diamonds the best trade-offs in validation. Solutions with good generalization power may be eliminated by genetic selection, which emphasizes solutions with good performance on the *optimization* data set (memorization). The most appropriate is to validate all candidate solutions during the optimization process with a *selection* data set and store good solutions in an auxiliary archive. This process is referred as the *global validation* and is further detailed in [3].

An algorithmic template for MOGAs using global validation is detailed in Algorithm 1, requiring a *selection*



(a) Optimization



(b) Validation

Figure 6. MLP EoC solutions as perceived by the optimization and validation processes at generation $t = 14$ with NSGA-II.

data set and an auxiliary archive S to store the validated solutions. An MOGA evolves the population P_t during mg generations. At each generation, the population P_{t+1} is validated and the auxiliary archive S is updated with good solutions. As the validation strategy used to train classifiers, this validation stage provides no feedback to the MOGA. At the end of the optimization process, the best solutions are stored in S.

4. Experimental Protocol

The tests are performed as in Fig. 1, targeting both the PD and MLP classifiers. The IFE methodology is solved to obtain the representation set RS_{IFE}. This set is used to train the classifier sets K_{PD} and K_{MLP}, using the PD and MLP classifiers. For a single classifier system, the most accurate classifiers $SI_{PD}, SI_{PD} \in K_{PD}$ and $SI_{MLP}, SI_{MLP} \in K_{MLP}$ are selected. EoCs are then created with K_{PD} and K_{MLP}, producing SE_{PD} and SE_{MLP}. Zoning strategies are compared with handwritten digits and the most performing is then applied with handwritten uppercase letters. Solutions obtained are

Result: Auxiliary archive S
Creates initial population P_1 with m individuals;
$S = \emptyset$;
t=1;
while $t < mg$ **do**
  Evolves P_{t+1} from P_t;
  Validate P_{t+1} with the *selection data set*;
  Update the auxiliary archive S with individuals from P_{t+1} based on the validation results;
  t=t+1;
end

Algorithm 1: Algorithmic template for a MOGA with global validation.

compared to the baseline representation defined in [4]. All tests are replicated 30 times and average values are presented.

The data sets in Tables 1 and 2 are used in the experiments – isolated handwritten digits and uppercase letters from NIST-SD19. MLP hidden nodes are optimized as feature set cardinality fractions in the set $f = \{0.4, 0.45, 0.5, 0.55, 0.6\}$. Classifier training is performed with the *training* data set, except for handwritten digits with the PD classifier that uses the smaller *training'* data set (to implement a computationally efficient wrapper). The *validation* data set is used to adjust the classifier parameters (MLP hidden nodes and PD hyper planes). The wrapper approach is performed with the *optimization* data set, and the *selection* data set is used with the global validation strategy. Solutions are compared with the test data sets, $test_a$ and $test_b$ for digits, and $test$ for uppercase letters.

Table 1. Handwritten digits data sets extracted from NIST-SD19.

Data set	Size	Origin	Offset
training'	50000	hsf_0123	1
training	150000	hsf_0123	1
validation	15000	hsf_0123	150001
optimization	15000	hsf_0123	165001
selection	15000	hsf_0123	180001
$test_a$	60089	hsf_7	1
$test_b$	58646	hsf_4	1

Table 2. Handwritten uppercase letters data sets extracted from NIST-SD19.

Data set	Size	Origin	Offset
training	43160	hsf_0123	1
validation	3980	hsf_4	1
optimization	3980	hsf_4	3981
selection	3980	hsf_4	7961
test	12092	hsf_7	1

The parameters used with MOMA are the following: crossover probability is set to $p_c = 80\%$, and mutation is set to $p_m = 1/L$, where L is the length of the mutated

binary string [12]. The maximum number of generations is set to $mg = 1000$ and the local search will look for $n = 1$ neighbors during $NI = 3$ iterations, with deviation $a = 0\%$. Each slot in the archive S is allowed to store $max_{S^l} = 5$ solutions. These parameters were determined empirically. The same parameters ($p_c = 80\%$, $p_m = 1/L$ and $mg = 1000$) are used for NSGA-II. Population size depends on the optimization problem. To optimize the IFE with MOMA, the population size is $m = 64$. For the EoC optimization, $m = 166$ is used. Individual initialization is performed in two steps for both optimization algorithms. The first step creates one individual for each possible cardinality value, zone number for the IFE, and aggregated classifiers for EoC optimization. The second step completes the population with individuals initialized with a Bernoulli distribution.

Experiments are conducted on a Beowulf cluster with 25 nodes (Athlon XP 2500+ processors and 1GB RAM). The optimization algorithms were implemented using LAM MPI v6.5 in master-slave mode with a simple load balance. PD vote and MLP output calculations were performed once in parallel using a load balance strategy, and results were stored in files to be loaded into memory for the EOC optimization process.

5. Experimental Results

The classification system is first optimized for handwritten digits using both zoning operators. Results for both the PD and MLP classifiers are indicated in Table 3. Table columns are as follows: *zoning operator* is the IFE zoning operator used, *solution* indicates the solution name, $|S|$ the solution cardinality (features or classifier number), *HN* the MLP hidden nodes and the error rates on the $test_a$ and $test_b$ data sets are indicated as e_{test_a} and e_{test_b} respectively.

The first conclusion is that the optimized EoC SE provides lower error rates than the single classifier SI. Comparing zoning operators, the superiority of the divider zoning operator is clear, as the hierarchical zoning operator has higher error rates. Comparing results with the baseline representation defined by the human expert, we observe that the divider zoning operator outperform the baseline representation in both SI and SE solutions, with both classifiers, whereas the hierarchical zoning operator fails to do the same.

To verify these statements, a multiple comparison is performed. A Kruskal-Wallis nonparametric test is used to test the equality of mean values, using bootstrap to create the confidence intervals from the 30 observations in each sample. The conclusions presented regarding the zoning strategies and improvements obtained with EoCs were confirmed as true, with a confidence level of 95% ($\alpha = 0.05$). Thus, we choose the divider zoning operator to experiment with uppercase letters, also expecting accuracy improvements with the EoC optimization.

Results obtained with uppercase letters are indicated in Table 4. Again, the baseline representation is outperformed by solutions produced by our classification system optimization approach. We observe again that the optimized EoC SE is more accurate than the single classifier SI, further justifying the choice for EoCs on robust classification systems.

Comparing solutions with the baseline representation, average improvements obtained with the IFE and EoC approaches justify the methodology. The IFE produced the same results in the 30 replications, thus the SI error rates in Tables 3 and 4 are the most accurate single classifiers. For digits and the divider zoning operator, the lowest EoC error rates with PD are $e_{test_a} = 1.93\%$ and $e_{test_b} = 5.06\%$, and with the MLP they are $e_{test_a} = 0.73\%$ and $e_{test_b} = 2.31\%$. Finally, for uppercase letters the lowest EoC error rate with PD is $e_{test} = 6.22\%$ and $e_{test} = 3.89\%$ with MLP.

The global validation strategy outperformed the traditional validation approach used in [2]. With handwritten digits, selecting the best EoC validated in the last population P_t yields average error rates of $\overline{e_{test_a}} = 2.07\%$ and $\overline{e_{test_b}} = 5.37\%$ with PD, and $\overline{e_{test_a}} = 0.77\%$ and $\overline{e_{test_b}} = 2.42\%$ with MLP, higher values in comparison to EoCs in Table 3. A more complete analysis is presented in [3].

Finally, Fig. 7 details the zoning strategies used to train the SI classifier selected from K_{PD}/K_{MLP}. Figures 7.a and 7.b details the zoning strategy selected with handwritten digits, using the divider and hierarchical zoning operators respectively. For handwritten digits, the zoning representation were the same for both the PD and MLP classifiers. However, with uppercase letters the selected zoning representation depends on the classifier. With the PD classifier, the classifier SI_{PD} was trained using the representation in Fig. 7.c, while with the MLP classifier we had that SI_{MLP} used the representation in 7.d.

Comparing results obtained to other representations in the literature, we have the following scenario. Milgram *et al* experimented with isolated handwritten digits in [13]. Using the same baseline representation they obtained error rates of 1.35% on $test_a$ with a NN classifier and 0.63% on $test_a$ with a SVM (one against all). As for handwritten uppercase letters, it is difficult to compare results directly. Differences in the experimental protocol to train and test classifiers make a direct comparison unfeasible with the results in [14, 15]. The same protocol was used in [13] with the baseline representation, yielding error rates of 7.60% with a 3-NN classifier and 3.17% with a SVM classifier (one against all). These results indicate that the use of a more discriminant target classifier may improve results obtained with the proposed approach to optimize classification systems.

6. Discussion

The methodology to optimize classification systems outperformed the baseline representation defined by an human expert. Obtained solutions are suitable for two different situations. The single classifier SI can be applied on hardware with limited processing power, and the EoC SE is suitable for classification systems running on

Table 3. Handwritten digits results – mean values on 30 replications for SI and SE.

Zoning Operator	Solution	PD classifier			MLP classifier			
		$\|S\|$	e_{test_a}	e_{test_b}	$\|S\|$	**HN**	e_{test_a}	e_{test_b}
–	*Baseline*	132	2.96%	6.83%	132	60	0.91%	2.89%
Divider	SI	330	2.18%	5.47%	330	132	0.82%	2.51%
	SE	24.67	2.00%	5.19%	14.13	–	0.76%	2.36%
Hierarchical	SI	242	3.46%	8.30%	242	134	1.14%	3.31%
	SE	13.7	3.09%	7.33%	22.86	–	0.99%	2.99%

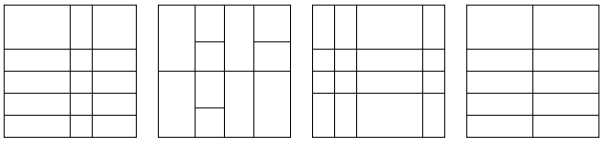


Figure 7. Zoning strategies.

Table 4. Handwritten uppercase letters results – mean values on 30 replications for SI and SE.

Solution	PD classifier		MLP classifier		
	$\|S\|$	e_{test}	$\|S\|$	**HN**	e_{test}
Baseline	132	9.20%	132	80	5.00%
SI	352	7.19%	220	88	4.29%
SE	14.41	6.43%	5.37	–	4.02%

server computers. Two IFE zoning operators were tested for feature extraction with handwritten digits, and the divider zoning operator outperformed the hierarchical zoning operator. The divider zoning operator was then used successfully with handwritten characters. Global validation also improved classification accuracy in comparison to the traditional selection method previously used.

Future works will extend the optimization of single classifier systems with feature subset selection, aiming to reduce representation complexity and classification time. Other zoning operators will be considered as well, to allow more flexible definition of *foci* of attention.

Acknowledgments

The first author would like to acknowledge the CAPES and the Brazilian government for supporting this research through scholarship grant BEX 2234/03-3. The other authors would like to acknowledge the NSERC (Canada) for supporting this research.

References

[1] Z.-C. Li and C. Y. Suen, "The partition-combination method for recognition of handwritten characters", *Pattern Recognition Letters*, Vol. 21(8): 701–720, 2000.

[2] P. V. W. Radtke, R. Sabourin and T. Wong, "Intelligent Feature Extraction for Ensemble of Classifiers", *Proceedings of the 8th International Conference on Document Analysis and Recognition*, Seoul, South Korea, 2005, pp 866–870.

[3] P. V. W. Radtke, T. Wong and R. Sabourin, "An Evaluation of Over-Fit Control Strategies for Multi-Objective Evolutionary Optimization", *Submitted to the 2006 International Joint Conference on Neural Networks*, Vancouver, Canada, 2006.

[4] L. S. Oliveira, R. Sabourin, F. Bortolozzi and C. Y. Suen, "Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24(11): 1438–1454, 2002.

[5] F. Kimura, S. Inoue, T. Wakabayashi, S. Tsuruoka and Y. Miyake, "Handwritten Numeral Recognition using Autoassociative Neural Networks", *Proceedings of the International Conference on Pattern Recognition*, 1998, pp. 152–155.

[6] L. I. Kuncheva and L. C. Jain, "Design Classifier Fusion Systems by Genetic Algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 4(4): 327–336, 2000.

[7] J. Kittler, M. Hatef, R. P. W. Duin and J. Matas, "On Combining Classifiers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20(3): 226–239, 1998.

[8] D. Ruta and B. Gabrys, "Classifier Selection for Majority Voting", *Information fusion*, Vol. 6: 63–81, 2005.

[9] P. V. W. Radtke, T. Wong and R. Sabourin, "A Multi-Objective Memetic Algorithm for Intelligent Feature Extraction", *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, Guanajuato, Mexico, 2005, pp 767–781.

[10] K. Deb, S. Agrawal, A. Pratab and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", *Proceedings of the Parallel Problem Solving from Nature VI Conference*, Paris, France, 2000, pp 849–858.

[11] C. Emmanouilidis, A. Hunter and J. MacIntyre, "A Multiobjective Evolutionary Setting for Feature Selection and a Commonality-Based Crossover Operator", *Proceedings of the 2000 Congress on Evolutionary Computation*, La Jolla, USA, 2000, pp 309–316.

[12] Á. E. Eiben, R. Hinterdind and Z. Michalewicz, "Parameter Control in Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 3(2):124–141, 1999.

[13] J. Milgram, R. Sabourin, M. Cheriet, "Estimating Posterior Probabilities with Support Vector Machines: A Case Study on Isolated Handwritten Character Recognition", submitted to the *IEEE Transactions on Neural Networks*, 2006.

[14] A. L. Koerich, "Large Vocabulary Off-Line Handwritten Word Recognition" (PhD thesis), *École de Technologie Supérieure*, Montreal, Canada, 314p, 2002.

[15] I.-S. Oh and C. Y. Suen, "Distance features for neural network-based recognition of handwritten characters", *International Journal on Document Analysis and Recognition*, Vol. 1(2): 73–88, 1998.