Selection of windows for W-operator combination from entropy based ranking

Marta M. Dornelles*†, Nina S. T. Hirata†
*Department of Exact and Technological Sciences
Universidade Estadual de Santa Cruz, Bahia, Brazil
Email: mmbertoldi@uesc.br
†Institute of Mathematics and Statistics
University of São Paulo, São Paulo, Brazil
Email: nina@ime.usp.br

***Abstract*—When training morphological operators that are locally defined with respect to a neighborhood window, one must deal with the tradeoff between window size and statistical precision of the learned operator. More precisely, too small windows result in large restriction errors due to the constrained operator space and, on the other hand, too large windows result in large variance error due to often insufficient number of samples. A two-level training method that combines a number of operators designed on distinct windows of moderate size is an effective way to mitigate this issue. However, in order to train combined operators, one must specify not only how many operators will be combined, but also the windows for each of them. To date, a genetic algorithm that searches for window combinations has produced the best results for this problem. In this work we propose an alternative approach that is computationally much more efficient. The proposed method consists in efficiently reducing the search space by ranking windows of a collection according to an entropy based measure estimated from input-output joint probabilities. Computational efficiency comes from the fact that only few operators need to be trained. Experimental results show that this method produces results that outperform the best results obtained with manually selected combinations and are competitive with results obtained with the genetic algorithm based solution. The proposed approach is, thus, a promising step towards fully automating the process of binary morphological operator design.**

***Keywords*-binary image; morphological operator design; W-operator combination; conditional entropy; sequential forward selection**

I. Introduction

Morphological operators are widely used in several image processing and analysis problems [1], [2], [3], [4]. Manual design of these operators consists on trial and error approaches to find compositions of simpler operators and suitable structuring elements until the desired processing results are achieved.

An alternative design approach rely on learning techniques from input-output pairs of training images. The goal of the learning process is to determine an image operator that maps input images to the corresponding output images. Most learning based approaches [5], [6], [7] consider translation-invariant and locally defined morphological operators, called W-operators. The local definition property of an operator means that its output at a given pixel can be determined by examining a finite region, delimited by a non-empty window W, around the pixel. Furthermore, all structuring elements that characterize the elementary operators of the learned operator will necessarily be constrained within W [8]. Windows should not be too small nor too large. Small windows result in large restriction errors and large windows result in large variance errors [9]. Therefore, specifying the right window W is important for the learning process to be able to produce a useful operator. An example of a pair of training images for the segmentation of handwriting is shown in Fig. 1.

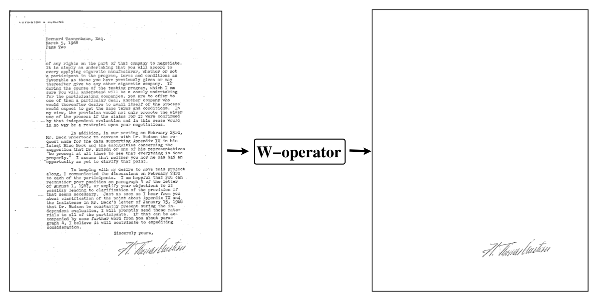


Fig. 1. Input-output pair of training images. Learning is concerned with determining an operator that maps the input to the output.

Given that the amount of training data is usually limited, once an optimal window in the sense of best balancing the tradeoff between the two types of errors is found, there is no much room to reduce the error of the trained operator. However, if more than one operator is allowed, one consistent way of reducing error is to consider combinations of operators [9]. The basic idea is to have several operators trained on distinct windows, and then in a second level of training combine their outcomes to reach a final output value.

However, such approach poses a new challenge. How many operators should be combined and how their individual windows should be chosen? Some recent works have addressed these questions [10], [11], [12]. Among them, a genetic algorithm (GA) based approach that searches for combinations in a reduced space of candidate windows has produced very interesting results, outperforming all manually selected combinations [12].

A major drawback of the GA based approach is, however, its computational cost. The high computational cost is due to the fact that the training process, which is a computationally expensive procedure, need to be executed each time the fitness function is evaluated. Fitness function must be evaluated through the GA iterations for each of the individuals (possible combinations) that compose the population.

Given this context, our main aim in this work is to propose an alternative approach to find window combinations that result in good combined operators.

We propose an approach that further reduces the search space by assigning a quality value to individual candidates in a collection of windows. The quality value is computed from training images without training, and thus very efficiently, using an entropy based measure. Then a search for good combinations is performed among the best ranked windows in a sequential forward fashion, similar to the well known sequential forward feature selection (SFS) algorithm [13]. This procedure requires only a few trainings. We also propose a slight modification in the entropy computation, in order to take into consideration the fact that we use estimated joint probabilities. The proposed modification is an attempt to better predict the generalization error of the learned operators.

A first set of experiments is carried out to verify the validity of the proposed method with respect to its ability to advantageously reduce the search space. Then a second set of experiments is performed to verify how effective are the combinations selected by the proposed approach in terms of producing good operators. Operators obtained using the selected window combinations are compared both to those selected by the GA based approach and to those selected manually.

The remaining of this paper is organized as follows. In Section II we introduce some background material and review related works for window selection in W-operator training. In Section III we detail the proposed method, including the window ranking method, the modified definition of the conditional entropy used to perform ranking, and a description of the whole procedure for two-level operator design from a set of ranked windows. In Section IV we present and discuss the experimental results and in Section V we present the conclusions of this work.

II. BACKGROUND

Binary images defined on a discrete grid (for instance, on $\mathbb{E} = \mathbb{Z}^2$) can be represented as elements of the power set $\mathcal{P}(\mathbb{E})$. Binary morphological operators $\Psi : \mathcal{P}(\mathbb{E}) \rightarrow \mathcal{P}(\mathbb{E})$ that are translation-invariant and locally defined with respect to a window W are called W-operators. They can be characterized by a local function $\psi : \mathcal{P}(W) \rightarrow \{0, 1\}$ such that $z \in \Psi(S) \iff \psi((S - z) \cap W) = 1$, for any $S \in \mathcal{P}(\mathbb{E})$ and $z \in \mathbb{E}$, where $(S - z) \cap W$ represents image S translated by $-z$ and restricted to W.

The problem of designing binary morphological operators can be reduced to the problem of learning its characteristic function [5], [6], [9]. Given pairs (S, I) of input-output sample images, the goal is to find a W-operator Ψ that, when used to process S, generates a result as close as possible of I. Closeness can be measured, for instance, computing the mean absolute error (MAE) between $\Psi(S)$ and I.

Assuming that samples (S, I) are realizations of a jointly stationary pair $(\mathbf{S}, \mathbf{I})$ of random processes with a local conditional probability distribution denoted by $P(\mathbf{y}|\mathbf{X})$, the MAE optimum W-operator is the one characterized by the function ψ given, for any $X \in \mathcal{P}(W)$, by

$$\psi(X) = \begin{cases} 1, & \text{if } P(1|X) > P(0|X), \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Note that when $P(1|X) = P(0|X)$, any of the two values can be attributed to $\psi(X)$ without affecting the error. We opt to attribute 0.

Since the conditional probabilities $P(y|X)$ are not known, in the learning process they are estimated from the input-output pairs of training images. The learning process also takes care of the generalization process, i.e., it assigns a value in $\{0, 1\}$ for each of the patterns X that are not present in the training images [9]. This is important because the learned operator Ψ must be able to assign an output value for each pattern that may be observed during its application. We call this process that takes as input a set of training images and a window and returns an image operator as the **basic training procedure**. Details of the procedure can be found in [9].

As mentioned in the introduction, too small windows should be avoided because the probability of having $|y - \psi(X)| \neq 0$ is high. On the other hand, too large windows may result in large imprecision of estimated conditional probabilities $P(y|X)$. Moreover, many low probability patterns X hardly are observed in the training images. These facts usually lead to large generalization error.

Combination of W-operators is an approach that has been proposed to better deal with the tradeoff between these two types of errors [9]. It consists in designing a number of operators using distinct windows, and then combining these operators. The combination process can be accomplished by first applying the basic training procedure for each of the windows, and then repeating it once again for patterns obtained by concatenating the output values of each of the operators. More specifically, if k operators are to be combined, then the basic training procedure must be executed $k+1$ times (one for each of the k windows and one for the combiner). This training method is called two-level training and the operators that are combined are called first-level operators while the operator that defines how they are combined (the combiner) is called second-level operator. The final two-level operator, which is a combined operator, is also a W-operator. The neighborhood considered by the combined operator, although not directly, is the union of the neighborhoods considered by the individual operators.

Empirical evaluation shows that combined operators resulting from the two-level training method consistently outperforms operators obtained with the basic training procedure for single windows [9]. This motivated studies on methods to help determining the number of operators to be combined and their respective windows. The problem of choosing an adequate window, even for the design of a single operator, is still an open problem.

A. Related works

One of the first attempts to automatically determine window combinations for two-level operators is described in [10]. A number of windows are created, from points in a given window domain, by using an algorithm that minimizes the magnitude of interaction information computed from training images. Operators are trained for each of the windows and ordered according to a criterion based on Conditional Mutual Information Maximization (CMIM) [14]. Then, two-level operators are designed using the first two windows, the first three windows, the first four windows and so on. The operators resulting from this process are then evaluated following the Minimum Description Length principle [15] and the best combination is selected. No conclusive comparison with manually designed combinations, described in [9], has been presented.

More recently, a genetic algorithm based solution, here to be called WGA, has been proposed [11], [12]. The main idea is to first constrain the search space by defining a collection of windows and then perform a genetic algorithm based search of window combinations in the constrained space. Each solution is encoded as a combination of windows and in order to evaluate fitness, the corresponding two-level operator is trained and its MAE is computed with respect to a validation set. Window combinations encoded on the individuals of the population are changed along iterations, and at the end, the one that resulted in minimum MAE is returned as the solution. Although WGA has shown to consistently produce better combinations than manually designed combinations, its execution time is high because evaluation of the fitness function requires training, which is a time consuming process. The collection of windows is built in such a way as to be coherent with the problem in hand. Three collections are considered: the first one, CJB, consists of basic shapes frequently used as structuring elements such as line segments, disks, and rectangles of different sizes and/or orientations, and occupying different positions relative to the origin within the window domain; the second one, CJI, consists of the windows obtained following the method based on minimizing the magnitude of interaction information as proposed in [10]; the last one, CJBI, is the union of the first two.

Combined classifiers have been also recently applied to the problem of removing staff-lines in music score images [16], [17]. Concerning methods for window definition, besides those pointed above, approaches based on feature selection has been also explored in [18].

III. Window selection using entropy based ranking

In classifier combination, by adding a new classifier to a group, often a final model with better performance can be obtained [19]. However, in order to improve a model by adding a new element in the combination, diversity is an important aspect to be considered. It is expected that each of the classifiers in the combination should be a specialist in some area of the problem to be solved. In general, the larger the diversity the better the model [19].

In our context, a major issue is how to measure the diversity of a set of windows and the quality of individual windows. A good measure of quality for a window is the MAE of the corresponding operator. However, computing the MAE requires training the operator. As for diversity, it could be ensured during the definition of a collection of windows to be used for searching the window combinations.

In order to choose a window combination, we use the same principle described above of adding new elements to a group. First, candidate windows are ordered according to some criterion. Then, the two top-ranked windows are used to train a two-level operator. Then, the process is repeated by taking at each iteration a new window combination obtained by adding the next ranked window to the previous combination, until a combination of a given maximum size is processed. At the end of the iterations, the combination that resulted in an operator with minimum MAE is chosen.

A key point in this approach is to use an efficient ranking criterion. We propose the use of conditional probability entropy.

A. Ranking criterion

Shannon's entropy [20], [21] has been used in previous works [22], [18] to evaluate candidate windows for the design of morphological operators. Recall that an optimal operator is estimated from conditional probabilities $p(y/X)$ (see Eq. 1). It is clear that when the conditional probability $p(y/X)$ is concentrated on one of the possible values (0 or 1), we have the best scenario with zero error. Conversely, when $p(1/X) = p(0/X) = 0.5$ we have the worst scenario in terms of error. This information can be captured by the mean conditional entropy [18], given by,

$$\begin{aligned} H_{(\mathbf{X},\mathbf{y})} &= -\sum_{X\in\mathbf{X}} p(X) \sum_{y\in\{0,1\}} p(y|X) lg(p(y|X)) \\ &= -\sum_{X\in\mathbf{X}} p(X) H(y|X). \end{aligned} \tag{2}$$

Note that in a theoretical formulation, both MAE and $H_{(\mathbf{X},\mathbf{y})}$ are equivalent in the sense that minimizing MAE corresponds to minimizing the conditional entropy and vice-versa. In other words, as we increase window size, MAE decreases or stays equal. Similarly, as we increase window, entropy tends to diminish because each conditional probability tends to concentrate on one of the possible outputs (i.e., there is a decrease in output confusion).

However, the direct use of this definition to estimate the quality of a window is not adequate since it neglects the fact that probabilities are estimated from a limited amount of training data. In the context of image operator design, it is expected that as we increase the window size the training error decreases as well, until reaching zero error. However, this decrease in error does not follow the same pattern with regard to test error; for test error, there is decrease of error until a certain point and after that the error starts to increase. The reason for such behavior lies in the fact that the larger the window is, the larger the variance of the designed operators and, thus, the chance of a non optimal operator to be produced increases as this variance increases. This is the so called generalization error that, once a minimum error point is reached, tends to increase with increasing window size.

Thus, the mean conditional entropy given by Eq. 2 computes a value for windows that is consistent with the expected training error of corresponding operators, but it is not adequate for predicting the error behavior of these operators with respect to test error.

1) Corrected conditional entropy: To cope with the above described issue, a correction term in the mean conditional entropy equation is proposed in [18]:

$$\hat{H}_{(\mathbf{X},\mathbf{y})} = \frac{2^{|W|} - N}{2^{|W|} + T} + \sum_{i=1}^{N} \frac{(N_{X_i} + 1)}{2^{|W|} + T} \hat{H}(y|X_i) \quad (3)$$

where $|W|$ is the window size, N_{X_i} is the number of times X_i has been observed, N is the total number of observed distinct patterns, T is the number of samples, and $\hat{H}(y|X_i)$ is the estimated conditional entropy of y given X_i.

This equation attributes conditional entropy 1 for all patterns that are not observed in the training data. When the window increases, the first term in the right side of the equation dominates the second one, and therefore minimum of $\hat{H}_{(\mathbf{X},\mathbf{y})}$ is achieved only by very small windows.

In [23], a new correction term is proposed based on an idea similar to the one proposed in a completely different application context [24]. Likewise [24], in [23] the non observed patterns are not included in the equation. Correction is applied to the so called **unique patterns**, that is, those that are observed only once in the training data:

$$\hat{H}^U_{(\mathbf{X},\mathbf{y})} = \frac{U}{T} + \sum_{\hat{p}(X_i) > \frac{1}{T}} \hat{p}(X_i)\hat{H}(y|X_i), \quad (4)$$

where U is the number of unique patterns. A conditional entropy equal to 1 is attributed to unique patterns, with the argument that unique patterns carry no sufficient information.

We propose, alternatively, to assign an entropy value H^c close to zero (e.g., $H^c = 0.001$) rather than equal to 1 (as in Eq. 4) to the unique patterns. This is motivated by the fact that although a single observation may not be sufficient, it does carry some information than no observation at all; it is more likely that its real conditional entropy is closer to zero than it is to 1. Thus, we have:

$$\hat{H}^*_{(\mathbf{X},\mathbf{y})} = H^c * \frac{U}{T} + \sum_{\hat{p}(X) > \frac{1}{T}} \hat{p}(X)\hat{H}(y|X). \quad (5)$$

Note that Eq. 4 is a particular case of this, where $H^c = 1$.

B. Procedure

The mean conditional entropy does not measure diversity between two windows. However it does associate a quality measure for each individual window. A possible way to guarantee some diversity is to consider windows with distinct shapes and sizes.

Let $CJ = \{W_1, W_2, \ldots, W_n\}$ be a collection of windows, each one of them defined on a common window domain. As stated before, evaluation of window combinations is performed by training the corresponding two-level operator. For training and evaluation, pairs of input-output images are divided into four groups: $(\mathbb{S}^1, \mathbb{I}^1)$ denotes the pairs of images to be used to train the first-level operators, $(\mathbb{S}^2, \mathbb{I}^2)$ denotes the pairs to be used to train the second-level operator (the combiner), and $(\mathbb{S}, \mathbb{I})$ denotes the pairs to be used to validate the trained operator in order to select the best one. The fourth group, denoted $(\mathbb{S}^t, \mathbb{I}^t)$, is used to test the selected operator.

This division may require a large number of pairs of input-output images. Ideally the four groups should be disjoint, as it will become clear below. However, the procedure can be applied even if the three first groups overlap each other. Only the fourth group, to be used to test the selected operators should not overlap the first three.

The proposed procedure, which we call WER (**W**indow selection using **E**ntropy based **R**anking), consists of the following steps:

1) Compute, for each window W_i in CJ, the value of a previously established criterion. We use as a criterion the corrected mean conditional entropy $\hat{H}^*_{(\mathbf{X},\mathbf{y})}$ defined by Eq. 5, with $H^c = 0.001$.
2) Rank windows in CJ according to the criterion function value, from the best to the worst one.
3) Train the two level operator for the first two windows, and denote it $\hat{\Psi}_2$; train the two-level operator for the three first windows and denote it $\hat{\Psi}_3$; repeat this process adding successively one window at a time, following the ranking order, until the combination with the first top-ranked J windows is processed.
   The basic training procedure must be executed once for each of the first J windows, and the two-level operator must be trained for $J-1$ combinations (with 2 to J windows).
   Number J is the maximum number of operators allowed in a combined operator. Reports in [12] suggest that 15 is a sufficient number for J and thus we adopt it here.
4) Apply each of the two-level operators $\hat{\Psi}_2, \hat{\Psi}_3, \hat{\Psi}_4, \cdots, \hat{\Psi}_J$ to the images in $\mathbb{S}$. The processed images will be denoted $\hat{\Psi}_2(\mathbb{S})$, $\hat{\Psi}_3(\mathbb{S})$, $\hat{\Psi}_4(\mathbb{S}), \cdots$, $\hat{\Psi}_J(\mathbb{S})$.
5) Compute MAE comparing $\hat{\Psi}_2(\mathbb{S}), \hat{\Psi}_3(\mathbb{S}), \cdots, \hat{\Psi}_J(\mathbb{S})$ to the respective ideal output images in $\mathbb{I}$. Denote the computed MAEs as $MAE_2, MAE_3, \cdots, MAE_J$.
6) Let $k = argmin\{MAE_i, i = 2, \ldots, J\}$. Select the combination with the first k windows and return.

Note that the above procedure uses the three first groups of pairs of input-output images. Once the optimal combined operator Ψ_k is selected, its expected performance can be computed on the images in group $(\mathbb{S}^t, \mathbb{I}^t)$.

IV. Experimental results

The procedure described in the previous section were implemented in Python, version 2.7. The code for generating the collection of windows were implemented in Java. Finally, for training operators and computing the MAE, we used TRIOSlib [25].

A preliminary set of tests were performed in order to assess the ability of WER in properly constrain the search space. Then, additional tests to compare the proposed method, WER, with manual selection and with WGA, a genetic algorithm based search method, were carried out for several datasets, using three different collections of windows. The details are presented below.

A. Datasets

The datasets used in the experiments are described in Table I. They are the same datasets used in [9]. Column **Dataset** indicates the name of the dataset and the respective window domain considered; **Description** briefly describes the task; $(\mathbb{S}^1, \mathbb{I}^1)$ indicates the number of images used for training first-level operators, $(\mathbb{S}^2, \mathbb{I}^2)$ indicates the number of images used to train second-level operators. The MAE for selecting a combination is estimated on images in $(\mathbb{S}, \mathbb{I})$. Column with header $(\mathbb{S}^t, \mathbb{I}^t)$ indicates the number of test images.

TABLE I
Description of datasets and the number of images in each of the groups (training, validation and test sets).

Dataset	**Description**	$(\mathbb{S}^1, \mathbb{I}^1)$	$(\mathbb{S}^2, \mathbb{I}^2)$	$(\mathbb{S}, \mathbb{I})$	$(\mathbb{S}^t, \mathbb{I}^t)$
Char_s (9×7)	Segmentation of character **s**	6	4	10	5
CircCirc (9×9)	Segmentation of circular objects	5	3	10	2
TexCirc (9×9)	Segmentation of characters	5	3	10	2
BoolNoise (9×9)	Boolean noise filtering	3	2	5	5
TexRev (9×7)	Text segmentation	3	2	5	5

Figure 2 shows pairs of input-output images from each of the datasets. The purpose is to just show the type of image transformations considered. Scale of the image has been reduced to better fit page width.

B. Window collections

The same window collections described in [12] have been used. The three collections are recalled briefly.

CJI: windows are generated using the algorithm proposed in [10] exploiting minimization of magnitude of interaction information. All windows are within a predefined window domain and are obtained by applying the algorithm to the training subset $(\mathbb{S}^1, \mathbb{I}^1)$. Collections with 81 and 105 windows were generated. Some windows in these collections are shown in Fig. 3.

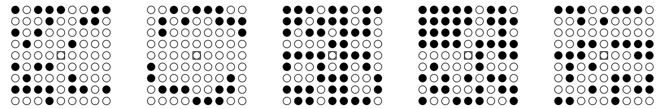


Fig. 3. Windows in the CJI collection, using the 9×9 window domain, for dataset *Char_s*.

CJB: Contains 24 windows corresponding to predefined basic shapes that are within a fixed window domain. Sample windows of this collection are shown in Fig. 4. In this case, window definition does not depend on the images.

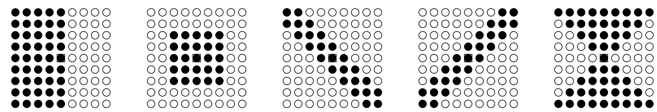


Fig. 4. Windows in the CJB collection, using the 9×9 window domain.

$CJBI$: Union of the two first sets.

C. Preliminary WER evaluation

Here we consider results obtained by WER considering different rankings: using the top $J = 15$ windows as described in the WER procedure, reversing the order of the top $J = 15$ windows, and considering $J = 15$ arbitrarily selected arbitrarily ranked windows. The results, using the CJB collection, are shown in Table II, where the first 10 rows refer to 10 runs of the arbitrary ranking and WER-r refers to WER with reversed ordering of the 15 top-ranked windows. The subscripts on MAE values indicate the number of windows in the selected combination.

TABLE II
Performance of WER (see details in the text)

	MAE over $(\mathbb{S}, \mathbb{I})$			
	Char_s	*CircCirc*	*TexCirc*	*BoolNoise*
1	0.004526_{10}	0.006862_{15}	0.041663_{8}	0.003099_{12}
2	0.004799_{11}	0.007444_{14}	0.041818_{10}	0.003116_{7}
3	0.004551_{12}	0.007153_{15}	0.043571_{13}	0.002779_{9}
4	0.004839_{11}	0.006394_{13}	0.041993_{13}	0.003249_{14}
5	0.004531_{12}	0.007309_{14}	0.040085_{11}	0.003157_{14}
6	0.004713_{8}	0.006862_{13}	0.039494_{9}	0.002631_{7}
7	0.004713_{11}	0.007454_{15}	0.039620_{9}	0.002896_{8}
8	0.004627_{11}	0.008161_{15}	0.042661_{9}	0.003101_{15}
9	0.004768_{10}	0.007038_{15}	0.040666_{7}	0.002959_{13}
10	0.004541_{10}	0.007423_{15}	0.041925_{14}	0.002960_{11}
WER	$\mathbf{0.00446_{8}}$	$\mathbf{0.006279_{13}}$	$\mathbf{0.039465_{7}}$	$\mathbf{0.002420_{9}}$
WER-r	0.004945_{14}	0.007569_{14}	0.040414_{10}	0.002829_{8}

Note that, for all four datasets, WER produced better results. That indicates that the restricted set of windows selected by WER is effective in generating good combinations. Another point to be noticed is that, in general, combinations selected by WER contain less windows than those selected by the alternative options.

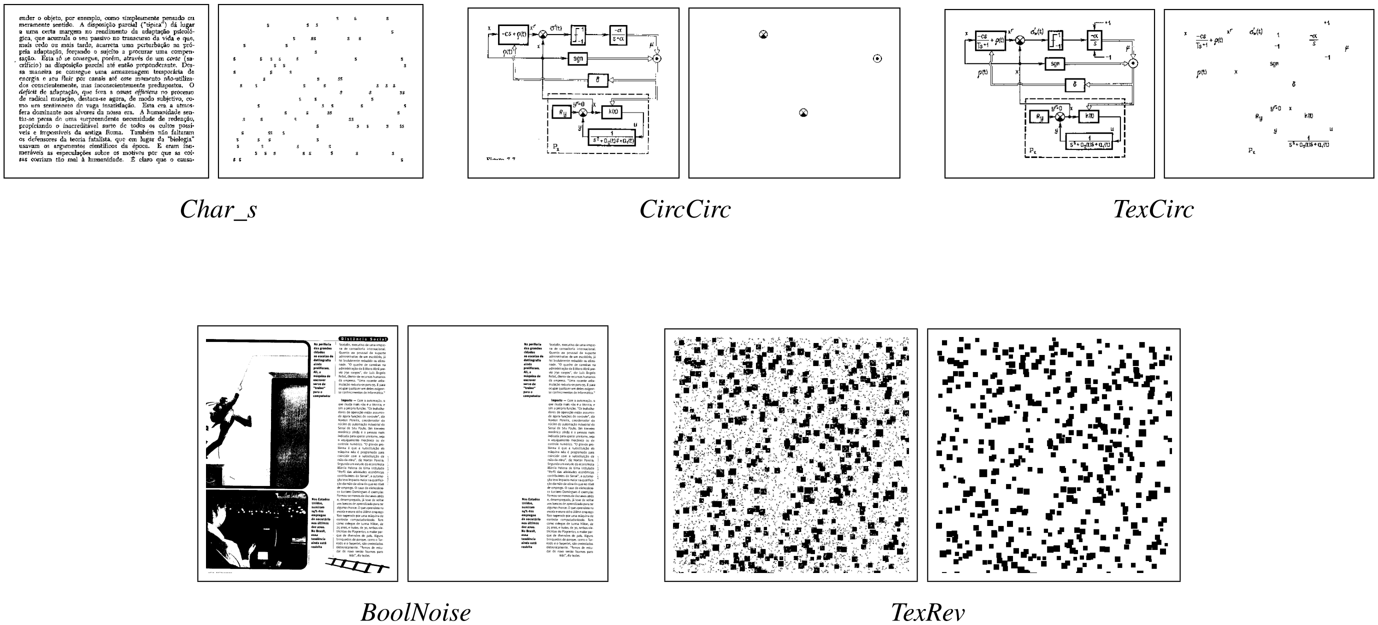


Fig. 2. Samples of the datasets used in the experiments. See also Table I.

An additional preliminary evaluation comparing the two corrections in the mean conditional entropy criterion (see Eq. 5), with $H^c = 1$ and $H^c = 0.001$ (ours) were also performed. Table III shows the MAE of the combinations selected by WER, using the CJB collection, when windows were ranked using $H^c = 1$ and $H^c = 0.001$.

TABLE III
WER PERFORMANCE, USING $H^c = 1$ AND $H^c = 0.001$ IN THE EQUATION OF THE RANKING CRITERION

Dataset	**Window domain**	MAE over $(\mathbb{S}^t, \mathbb{I}^t)$	
		$H^c = 0.001$	$H^c = 1$
Char_s	9×7	0.00446_8	0.00409_7
CircCirc	9×9	0.00628_{13}	0.00707_{15}
TexCirc	9×9	0.03946_7	0.04241_{15}
BoolNoise	9×9	0.00242_9	0.00315_{15}
TexRev	9×7	0.03739_{10}	0.03928_9

Except for the first dataset, $H^c = 0.001$ produced better results. This indicates that the newly proposed correction term in the mean conditional entropy equation better captures the expected generalization error of an operator.

D. Comparison with other methods

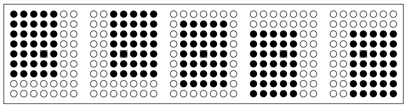
We compare the results obtained by WER with those reported in [9] for manually combined windows and those reported in [12] using WGA. Table IV shows the MAE relative to the images in the test set $(\mathbb{S}^t, \mathbb{I}^t)$ for the best combinations obtained by WER and by WGA, and also for the manually selected combination, with respect to each of the window collections.

The best results are highlighted in light gray. As can be seen, they all were produced by WGA. Although operators produced by combinations selected by WER are not as good as the ones obtained with WGA, we note that they are better than the manual choice. Another interesting observation is that best results of WER are all obtained when using the CJB window collection, while for WGA the best solutions are from either CJB or CJBI. A possible explanation for this is the diversity of windows. In CJB, windows are diverse by construction (at least in terms of their shape). On the other hand, in CJI many of the windows share several points and the results observed in this experiment may be indicating that windows are not as diverse as they ought to be. Further investigation is necessary to better understand this property.

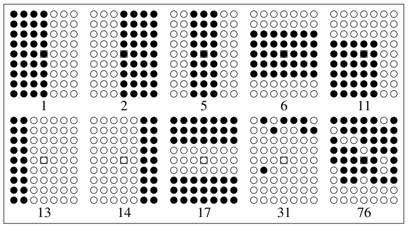
TABLE IV
MAE COMPUTED OVER $(\mathbb{S}^t, \mathbb{I}^t)$. SUBSCRIPT INDICATES THE NUMBER OF WINDOWS. FOR THE CJBI COLLECTION, THE FIRST SUBSCRIPT REFERS TO WINDOWS IN CJB AND THE SECOND ONE TO WINDOWS IN CJI

Method	*Char_s*	*CircCirc*	*TexCirc*	*BoolNoise*	*TexRev*
Manual	0.00561_5	0.00651_6	0.04196_6	0.00331_5	0.02600_5
CJI					
WER	0.00734_8	0.00562_{15}	0.04794_8	0.00337_7	0.03580_3
WGA	0.00672_8	0.00461_{15}	0.04685_8	0.00312_8	0.03280_3
CJB					
WER	0.00460_8	0.00470_{13}	0.04535_7	0.00224_9	0.02198_{10}
WGA	0.00461_8	0.00510_{15}	0.03784_8	0.00224_8	0.01248_3
CJBI					
WER	$0.00466_{6,1}$	$0.00573_{3,7}$	$0.04648_{6,2}$	$0.00237_{4,2}$	$0.02490_{4,5}$
WGA	$0.00433_{8,2}$	$0.00396_{2,13}$	$0.04147_{5,3}$	$0.00215_{6,2}$	$0.01248_{3,0}$

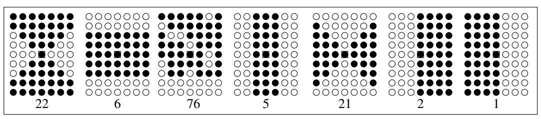
Examples of selected window combinations are shown in Fig. 5. They are the combinations selected manually, by WGA and by WER, for the *Char_s* dataset. In general, we have observed that the number of windows in combinations selected by WGA and by WER are close each other.



Manually selected



Selected by WGA, from CJBI



Selected by WER, from CJBI

Fig. 5. Window combinations selected for dataset *Char_s*.

As for the processing time, while WER requires, after window ranking, the training of J first-level operators ($J = 15$ in our experiments) and the training of $J - 1$ second-level operators, WGA requires much more trainings. For instance, let be a window collection of size 24, an initial population with 20 individuals and 200 iterations in WGA. Then, 24 trainings must be executed, one for each of the windows in the collection, and during the execution of the iterations, the second-level training must be executed $20 * 200$ times corresponding to the number of times the fitting function must be evaluated. Thus, the execution time of WGA also depends on the population size and the number of iterations. Table V shows execution times of WER and WGA. Running time measurements were taken on a machine with Ubuntu 4.11 Linux operating system and with 4 Intel i5-2410M processors of 2.30GHz. For the WGA, we present the number of windows in the collection, the number of iterations and the final size of the population. No clear patterns of execution times are observed because window sizes vary much from one collection to another. Nevertheless, it is clear that WER requires much less time than WGA.

A comparative example of results for an image in the *CircCirc* dataset is shown in Fig. 6. The image sections shown are, respectively, from the input, ideal output, result of the operator corresponding to manually chosen windows, result of the operator corresponding to windows selected by WGA (using the CJBI collection) and result of the operator corresponding to windows selected by WER (using the CJB collection).

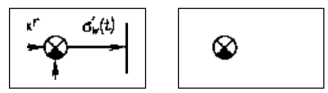
It is possible to observe that, visually, the result of WER is close to the one obtained with WGA and superior to the one obtained manually. We have observed that, using the CJB collection, the results obtained with WER were consistently

TABLE V

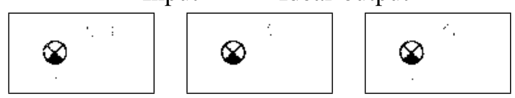
EXECUTION TIMES FOR WGA AND WER. FOR WGA, THE *window collection size / number of iterations / final population size* IS INDICATED BELOW EACH EXECUTION TIME (SEE MORE DETAILS IN THE TEXT)

	WER - Time (hours)			**WGA** - Time (hours)		
Dataset	**CJI**	**CJB**	**CJBI**	**CJI**	**CJB**	**CJBI**
Char_s	0.08	0.07	0.11	3.13 62/200/50	1.72 24/143/37	3.24 86/171/58
CircCirc	0.56	0.28	0.59	1.42 81/84/45	1.03 24/200/48	1.84 105/119/58
TexCirc	0.23	1.00	0.68	3.44 81/200/54	3.22 24/170/48	2.85 105/111/34
BoolNoise	2.38	2.06	2.12	14.60 81/137/38	5.09 24/157/37	18.09 105/147/47
TexRev	2.19	3.97	4.05	9.95 62/104/30	7.52 24/120/33	17,37 86/136/50

better than the results of manually chosen combinations for all tested datasets.



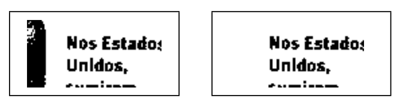
Input Ideal output



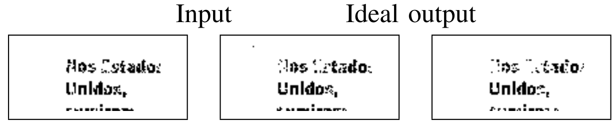
Manual WGA: CJBI WER: CJB

Fig. 6. Sections of an image in *CircCirc* dataset.

Further examples of results are shown in Fig. 7 and Fig. 8.

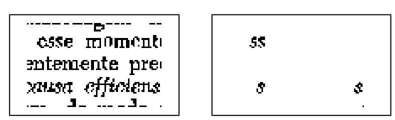


Input Ideal output

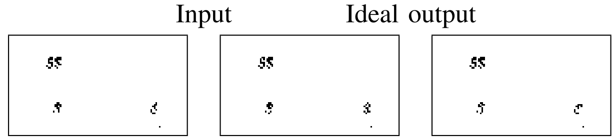


Manual WGA: CJBI WER: CJB

Fig. 7. Results for dataset *TexRev*.



Input Ideal output



Manual WGA: CJBI WER: CJB

Fig. 8. Results for dataset *Char_s*.

V. Conclusion

We have presented a method for selecting a window combination, to be used to train the operators to be combined in the process of designing two-level morphological operators. The main idea is to rank windows in a collection according to a quality criterion that may be useful to predict good combinations. Then combinations are searched among top ranked windows only. In this way, we drastically reduce the search space. Another key point in the proposed method is the use of a newly proposed modified mean conditional entropy as the criterion for ranking windows. Since mean entropy computation does not require training, it can be efficiently computed.

Experimental results obtained using a collection of windows with varying shapes, sizes and relative positions show that the method selects combinations that are better than those manually selected. At the same time, its results are only slightly below the results produced by WGA that employs a genetic algorithm based search. A great advantage of the proposed method over WGA is its computational efficiency.

Thus, the results obtained in this work indicate that the proposed method has potential to automate most of the binary morphological image operator design process. Using the proposed approach, besides the training images an user would need to provide only the specification of a window domain.

For further improvement, the proposed ranking criterion can be combined with other criteria that take diversity of a group of windows in consideration. By doing that, chances of finding better combinations will increase without leading to an increase in the number of combinations to be tested. Future steps of this research include the development of methods to automatically determine a window domain, improvements with regard to the creation of window collections, and the development of indexes to measure diversity of window combinations.

Acknowledgment

Marta M. Dornelles thanks Universidade Estadual de Santa Cruz for the financial support. Nina S. T. Hirata thanks CNPq (grants 484572/2013-0 and 307357/2012-0).

References

[1] P. Soille, *Morphological Image Analysis*, 2nd ed. Berlin: Springer-Verlag, 2003.

[2] A. Tuzikov, P. Soille, D. Jeulin, H. Bruneel, and M. Vermeulen, "Extraction of grid patterns on stamped metal sheets using mathematical morphology," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision and Applications*, 1992, pp. 425–428.

[3] R. Hirata-Jr, J. Barrera, R. F. Hashimoto, D. O. Dantas, and G. H. Esteves, "Segmentation of microarray images by mathematical morphology," *Real-Time Imaging*, vol. 8, no. 6, pp. 491–505, 2002.

[4] S. S. Basha and K. S. Prasad, "Automatic detection of breast cancer mass in mammograms using morphological operators and fuzzy C-means clustering." *Journal of Theoretical and Applied Information Technology.*, pp. 704–709, 2009.

[5] E. R. Dougherty, "Optimal mean-square N-observation digital morphological filters : I. optimal binary filters," *CVGIP: Image Understanding*, vol. 55, no. 1, pp. 36–54, 1992.

[6] J. Barrera, E. R. Dougherty, and N. S. Tomita, "Automatic programming of binary morphological machines by design of statistically optimal operators in the context of computational learning theory," *J. Electronic Imaging*, vol. 6, no. 1, pp. 54–67, 1997.

[7] J. Barrera, R. Terada, R. Hirata Jr., and N. S. T. Hirata, "Automatic programming of morphological machines by PAC learning," *Fundam. Inform.*, pp. 229–258, 2000.

[8] G. J. F. Banon and J. Barrera, "Minimal representations for translation-invariant set mappings by mathematical morphology," *SIAM J. Appl. Math.*, vol. 51, pp. 1782–1798, 1991.

[9] N. S. T. Hirata, "Multilevel training of binary morphological operators," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 707–720, 2009.

[10] C. S. Santos, N. S. Hirata, and R. Hirata Jr., "An information theory framework for two-stage binary image operator design," *Pattern Recognition Letters*, vol. 31, no. 4, pp. 297–306, 2010.

[11] M. M. Dornelles and N. S. T. Hirata, "A genetic algorithm based method for determining two-level morphological operators," in *Proceedings of 17th International Conference on Systems, Signals and Image Processing (IWSSIP)*, vol. 1, 2010, pp. 453–456.

[12] ——, "A genetic algorithm based approach for combining binary image operators," in *Proceedings of 21st International Conference on Pattern Recognition (ICPR)*, vol. 1, 2012, pp. 3184–3187.

[13] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recogn. Lett.*, vol. 15, no. 11, pp. 1119–1125, Nov. 1994.

[14] F. Fleuret and I. Guyon, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, vol. 5, pp. 1531–1555, 2004.

[15] P. D. Grünwald, *The Minimum Description Length Principle*. The MIT Press, 2007.

[16] I. S. Montagner, R. Hirata Jr., and N. S. T. Hirata, "A machine learning based method for staff removal," in *22nd International Conference on Pattern Recognition (ICPR)*, 2014, pp. 3162 – 3167.

[17] ——, "Learning to remove staff lines from music score images," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 2614 – 2618.

[18] D. C. Martins Jr., R. M. Cesar Jr., and J. Barrera, "W-operator window design by minimization of mean conditional entropy," *Pattern Analysis and Applications*, vol. 9, no. 2-3, pp. 139–153, 2006.

[19] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[20] C. E. Shannon, "A Mathematical Theory of Communication," *Bell system technical journal*, vol. 27, 1948.

[21] M. A. Hall and L. A. Smith, "Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper," in *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*. AAAI Press, 1999, pp. 235–239.

[22] D. Vaquero, J. Barrera, and R. Hirata, "A maximum-likelihood approach for multiresolution w-operator design," in *18th Brazilian Symposium on Computer Graphics and Image Processing*, Oct 2005, pp. 71–78.

[23] M. S. Reis, C. E. Ferreira, and J. Barrera, "The u-curve optimization problem: improvements on the original algorithm and time complexity analysis," *CoRR*, vol. abs/1407.6067, 2014.

[24] A. Porta, G. Baselli, D. Liberati, N. Montano, C. Cogliati, T. Gnecchi-Ruscone, A. Malliani, and S. Cerutti, "Measuring regularity by means of a corrected conditional entropy in sympathetic outflow," *Biological Cybernetics*, vol. 78, no. 1, pp. 71–78, 1998.

[25] I. S. Montagner, R. Hirata Jr, and N. S. T. Hirata, "TRIOS - an open source toolbox for training image operators from samples," in *Workshop of Works in Progress (WIP) in SIBGRAPI 2012 (XXV Conference on Graphics, Patterns and Images)*, 2012.