Feature selection with equalized salience measures and its application to segmentation

Davi P. Santos and João Batista Neto
Instituto de Ciências Matemáticas e de Computação - USP
CP 668, São Carlos, SP, Brazil 13560-970
daveps@gmail.com,jbatista@icmc.usp.br

Abstract

Segmentation is a crucial step in Computer Vision in which texture plays an important role. The existence of a large amount of methods from which texture can be computed is, sometimes, a hurdle to overcome when it comes to modeling solutions for texture-based segmentation. Following the excellence of the natural vision system and its generality, this work has adopted a feature selection method based on salience of synaptic connections of a Multilayer Perceptron neural network. Unlike traditional approaches [9, 21], this paper introduces an equalization scheme to salience measures which contributed to significantly improve the selection of the most suitable features and, hence, yield better segmentation. The proposed method is compared with exhaustive search according to the Jeffrey-Matusita distance criterion. Segmentation for images of natural scenes has also been provided as a probable application of the method.

1. Introduction

Segmentation is a task performed mainly through texture analysis [25, 1, 20, 22] for which several methods have been developed. A large amount of methods is based on texture.

Lacking of formal definition, texture nature leads to a great variety of features due to its inherent subjective properties [25]. While some features are mere calculations that mainly depict the statistical relationship of a pixel and a certain neighborhood [11, 8], others are inspired by the human visual system [6]. The latter mimics the way human perceive different aspects of texture: sometimes related to a tactile (e.g., harshness and smoothness), geometric (e.g., direction and regularity) or another paradigm.

These aspects provide the basis of mathematical models, empirically or biologically motivated. Such models follow a pattern recognition approach in which several texture features are combined in order to generate classification or segmentation. Depending on the application, some models succeed and some fail [12]. The success in discriminating texture is directly related to the number of features employed. If in one hand, a considerable number of features increases the generality of the model, it introduces the problem of high dimensionality and acquisition costs of all texture measures.

This paper proposes a bioinspired approach to feature selection in which 71 texture features are extracted from a window and combined into an Multilayer Perceptron (MLP). It is bioinspired in the sense that the intelligence within the internal connections of an MLP is used to rank the features.

A common way to analyze the neural network connections is to compute Garson salience measure [9]. It is defined over the input-output paths, by extracting feature relevance from neuron weights. However, a typical problem of this approach is that the measure is dependent on the mean value of each feature. This work introduces a normalization scheme, herein called Input Equalization, which handles data prior to passing them to the neural network so as to improve the efficiency of the measure for the discrimination process.

The assessment of the more relevant features, according to salience, is achieved by comparing them with features selected by an optimal method. Two distinct assessment strategies have been adopted in this work: classification success rate and quality of the feature subset. A more complex classifier - an MLP - and a simpler classifier - a minimum distance classifier. (MDC) [10] - have been used in the first and second strategies, respectively.

Training was carried out with the learning algorithm RPROP [23] for its power of generalization in the learning process and the lack of parameter adjustment that leads to a relevant simplification of the training stage. Another advantage of the RPROP algorithm is its aptitude for large training data sets, like the one employed in the experiments shown in this paper. Since batch mode training is deterministic, the order of the patterns is irrelevant. They do not compete during training [13].

This work has adopted four well-known texture extraction methods: Cooccurence matrices [11], Run Lengths [8], 1D [3] and 2D Fourier Transform [2], from which, a total of 71 texture features have been computed by a 13x13 template. For the sake of clarity, features will be represented, hereafter, by a number in the range $1, \ldots, 71$, distributed as follows: Run-Length(feature numbers: 1-10), first order statistics(11-14), Coocurrence matrices (15-58), 1D (59-66) and 2D (67-71) Fourier Transform features.

This paper is organized as follows. Section 2 gives a brief overview on feature selection, with special attention to the heuristic approach chosen for this work. The methodology adopted is explained in section 3. Section 4 and 5 brings experiments and results of the feature selection method, and its application to segmentation, respectively. Conclusions are finally given in section 6.

2 Feature selection

Feature selection is generally used to reduce the measure acquisition costs and also improve the precision of the classification system [15]. Both aspects are evaluated in this work.

2.1 Exhaustive search selection

Exhaustive search is an optimum feature selection method. To be computationally viable, it requires a low cost criterion and/or a low dimensionality problem. Given n features, the total number of subsets is $2^n - 1$. When the number of desired features is known, the total amount of combinations is $\binom{n}{m}$ (O(n^m)). A broadly used criterion is the interclass separability, like Jeffrey-Matusita (JM) distance [14], that uses the Bhattacharyya distance [7]:

$$JM_{h,k} = \sqrt{2\left(1 - exp\left(-B_{h,k}\right)\right)}$$

$$B_{h,k} = \frac{1}{8}\left(M_h - M_k\right)^T \left(\frac{C_h + C_k}{2}\right)^{-1} \left(M_h - M_k\right) + \frac{1}{2} ln \left(\frac{\left|\frac{C_h+C_k}{2}\right|}{\sqrt{|C_h|\,|C_k|}}\right)$$

where h and k are the classes and M_i and C_i are, respectively, the mean vector of the patterns and the covariance matrix of class i.

When more than two classes are presented, the summation of the distance of all pairs is performed. Prior probabilities (P_i) are then used when the pattern sets have different sizes:

$$JM = 2\sum_{i,j} P_i P_j JM_{i,j}$$

The feature subset that yields the highest JM value is the one that provides the best class separability.

2.2 Salience selection

The fact that neural networks do not make any prior assumption on the nature of the data distribution has been explored by many researchers to avoid the restrictive conditions of statistical methods [5].

Garson [9] proposed a salience measure that indicates the importance of each MLP input node. His heuristic approach, which is an estimate based on the cost of all connection paths from the input to the output layer, was shown to be computationally viable when compared with traditional selection methods [21].

First of all, a normalization factor N_h is calculated for each h hidden neuron. It is the sum of the absolute values of the neuron weights $w_{i,h}$ (index i corresponds to the input node number):

$$N_h = \sum_i |w_{i,h}|$$

Each possible path (where variable o stands for output neurons) returns a value $w*_{i,h,o}$ based on N_h and output weights:

$$w*_{i,h,o} = \frac{|w_{i,h}||w_{h,o}|}{N_h}$$

Finally, salience $S_{i,o}$ of input i, with respect to output neuron o, is given by:

$$S_{i,o} = \sum_o w*_{i,h,o}$$

With the salience measure it is possible to eliminate a feature with no knowledge on its related contribution to classification rate. Hence, instead of training the neural network several times to evaluate each feature contribution, a single training suffices to compute a ranking in which features with higher salience values are said to be more relevant. For better accuracy, some intermediate trainings can be performed until the end of selection process.

Salience is usually applied to reduce selection complexity. For instance, for each feature removed, a training could be performed; for n features, the complexity would be $O(n)$ (more specifically $O(n - m)$, because when a m-sized subset is reached, algorithm ends). This algorithm of linear complexity is preferable to an exhaustive search. However, there is no guarantee of optimality.

3 The proposed method

The work on salience [21] deals with a 2-class problem only, that is, a single MLP output. This work, on the other hand, proposes a new value ST_e - the total salience for each input e:

$$ST_e = \sum_s \frac{S_{e,s}}{\sum_f S_{f,s}}$$

The work of Ruck et al [24], in which the influence of the effective feature interval over salience (in that case, salience is the derivative of output with respect to each individual input) is mentioned, raised some concern on the weakness of Garson salience when applied over irregular features such as those described in section 1 (features of uneven levels of mean values and/or within different numerical ranges).

It is known that when the mean value of a certain feature is high, the weights directly connected to it are likely to possess lower values. This is due to the *counterbalance* performed by the MLP. This trade-off is a natural characteristic of MLP, giving equal chances to all entries, while rewarding the discriminating power of each feature as opposed to their sheer numerical values. Therefore, the salience measure is directly affected, because it depends on the values of such weights. Hence, without an appropriate treatment, the mean value of the entries becomes a component of traditional salience measure that may jeopardize its importance as a discriminant feature.

To work around this problem, we have opted for dealing with the input data. Masters [19] suggests the *centering* and *scaling* techniques to ease the training stage, although he aknowledges that these techniques are not always necessary. Nevertheless, in selection by salience, as will be demonstrated later, they are essential to compensate level differences between features. To provide all features with the same mean value $(0,5)$, the new $v(1)$ values of each class have been shifted and scaled according to the given formula:

$$v(1) = \frac{v(0) - \mu}{2m} + 0,5$$

$$m = \begin{cases} |min - \mu|, \text{ if } |min - \mu| > |max - \mu|, \\ |max - \mu|, \text{ if } |max - \mu| > |min - \mu|. \end{cases} \quad (1)$$

where μ, max and min are mean, maximum and minimum feature values, respectively. This special normalization has been named *input equalization* (IE).

In Figure 1, two out of the three curves correspond to the salience values computed for each feature: the traditional and the proposed one. The remaining curve depicts the mean value of each feature. Some inverse correlation can be observed between the traditional and the mean value curves. It is due to the signal trade-off introduced by the MLP. The curve for the new salience, however, shows a correlation with value near zero.

Another issue closely related with the performance of the selection by salience measure is that of partial retraining. Leray and Gallinari [18] have suggested the removal of just a single feature at a time, followed by an immediate retraining. By doing so, correlations between features can be identified. To speed up the selection process, the partial retraining technique has been adopted. This practice feeds the learning process with information from a previous training stage so that the convergence of learning process is attained earlier. This technique has been suggested by Laar et al. [17, 16]. They assume that the weight values of the previous network still contain useful information for the current network. The computational cost of such approach, according to preliminary experiments, were $1,4$ times lower than those obtained by an MLP in which all weights were reset at each iteration. The success rates *with* or *without* this technique have been very similar.

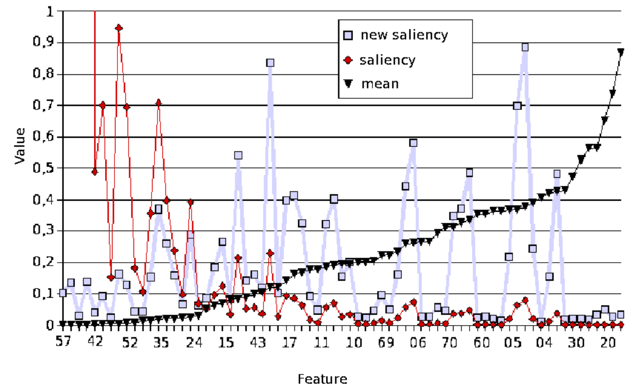

Figure 1. New salience compared with the original one: correlation with mean value, originally $-0,22$, changed to $0,06$ with equalization. Some feature numbers are omitted.

Bringing all these suggestions together, a new system has been implemented, as shown in Figure 2.

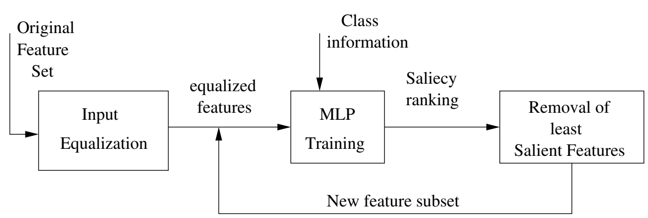

Figure 2. Selection by Saliency (SS).

The feature selection and classifier training are implemented separately. As a result, the selection by salience, which is the main contribution of this work, can be assessed isolately. Such assessment consists of comparing the proposed technique with the optimal selection, according to the interclass separability measure *Jeffrey-Matusita Distance*. Therefore, an extra MLP is necessary as a classifier. Two variations of selection by salience have been tested.

4 Experiments and results

The experiments presented in this section aims to evaluate the feature selection by salience, which is an heuristic process, against optimal selection according to *Jeffrey-Matusita distance*. Both approaches will be detailed in the next paragraphs and will be hereafter referred to as SS and SJM, respectively.

A variation of the SS approach has also been assessed. It consists of interrupting the selection process after the first iteration, so that the best salience measures would then indicate the best features. It is clear, according to this approach that the complexity of the selection process drops drastically, changing from linear $O(n)$ to constant order $O(1)$. This alternative approach is referred to as SSCO. (selection by salience of constant order).

Next, selection by salience (SS) is performed to select the best subset of features and, for comparison purposes, the best subsets from SJM and SSCO approaches are also selected. To investigate the relevance of the Input Equalization (IE), as explained in section 3, the best subset of SS without IE is also computed. All approaches are evaluated in terms of classification success and computational costs.

A Minimum Distance Classifier (MDC) was then used to determine which subset (from SS or SJM) produced the largest performance gain against the original feature set. Finally, SS is qualitatively assessed as a segmentation tool for natural scenes images by visual inspection of the resulting segmented images.

In all experiments, the criterion adopted to determine the class to which every new pattern belonged to has been the number of the output neuron with signal above $0,5$. In cases where none (or more than one) output neuron was above this value, then the class was labeled "unknown". The processing times, in seconds, correspond to the CPU occupation period, that is, parallel processes had their times summed up to give a more realistic idea of how much CPU has been used by them.

4.1 The input data

A sample of 1607824 patterns has been used to create training and testing sets. It has been computed from four distinct natural texture images, taken from Brodatz album [4] combined as illustrated in Figure 3. Hence, the testing set is 52 times larger than the training set, demanding a stronger generalization power from the classifier. 20416 patterns and 4 classes have been chosen so that a viable processing time could be fulfilled.

Since the heuristic approach was also governed by this choice, the new amount of patterns was limited to be, at least, 5 times larger than the total of free parameters of the neural network. This limit L has been computed as follows:

$$L = 5((N_e + 1)N_o + (N_o + 1)N_s)$$

with N_e being the number of entries, N_o the number of hidden neurons and N_s the number of output neurons. Replacing by the values adopted in this work, yields:

$$L = 5((71 + 1)5 + (5 + 1)4) = 1920$$

The total number of training patterns, although small, remained fully satisfactory: $20416 >> 19200$.

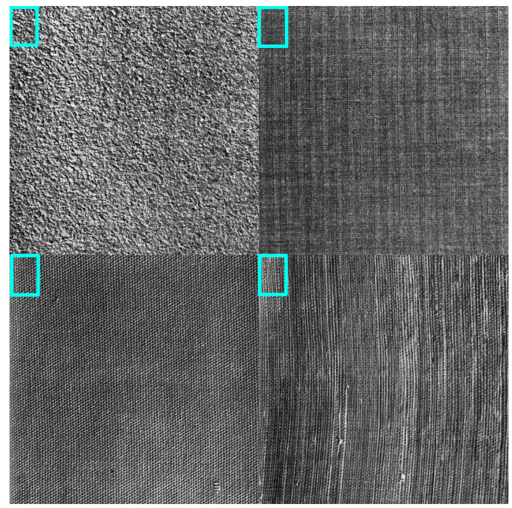

Figure 3. Testing sample: training areas are indicated by rectangles.

For each approach (SJM, SS and SSCO), the selected features and the training success rate are reported. A comparison between the differents testing success rates is also given.

4.2 SJM

The SJM approach indicated the best subset as the one formed by features number 10, 12, 19 and 22, originated from *Run Length*, Cooccurrence Matrices and *first order statistics*. The normalized JM distance was $0,983772$, computed in $8857s$. Combinations which involved zero determinant matrices have been discarded. It is possible to track the evolution of the JM distance against the growth of the feature subset (figure 4). The selection of a subset feature size larger than 4 would make computation practically infeasible.

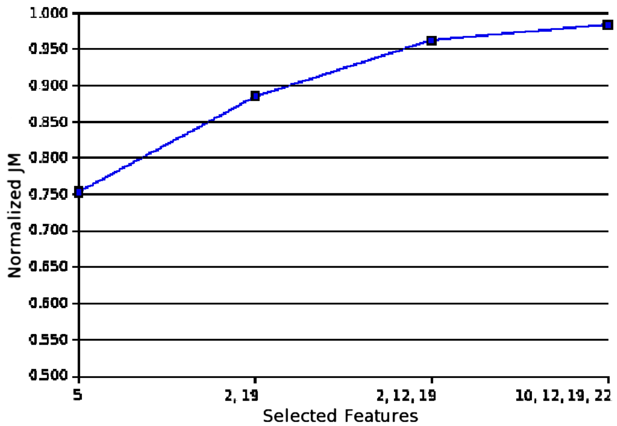

Figure 4. Jeffrey-Matusita distance: method is computationally viable up to 4 features (sample size = 30624).

4.3 SS

Five MLP neural networks have been initially trained with all 71 texture features. As training errors settled [1], the shortest salience feature was removed. When the size of the remaining subset was reduced to 4 features, the process was halted.

The best outcome showed a success training rate of $98,0\%$. The resulting features were 12, 16, 36 and 52 (*pixel value standard deviation* and Coocurrence matrices).

4.4 SS without IE

The SS without IE (Input Equalization) was performed 5 times. The best subset yielded a training error rate of $70,8\%$. Cooccurrence matrices features (37, 48, 52 and 57) were the only texture measures present in this subset.

4.5 SSCO

As said before, SSCO is a variation of the SS approach in which salience is attained right after the first iteration of the algorithm. The complexity is then of order $O(1)$. An experiment has been conducted and consisted of evaluating on the 4-sized subset of the most salient features after the first iteration.

In the beginning, all features were still present. That explains the 100% success training rate in the selection process. Hence, the criterion for choosing the best subset of features, by running the process 5 times, has been to perform the training step each time and choose the subset that delivered the lowest possible error $(4,0\%)$. This sort of training is equivalent to that attained with the SS approach, in its last iteration.

The following subset has been obtained in this experiment: 7, 19, 20 and 52 (*Run Length* and Cooccurrence Matrices).

4.6 Comparing the approaches

The data set used in feature selection stage has been the same data set employed to train [2] each of the subsets selected: SJM, SS, SS without IE and SSCO.

The average training processing time for all approaches was $922s$ (with 1.776 standard deviation) and the segmentation time was, in average, $4,6s$ $(0,07s$ standard deviation). Texture feature extraction took $17s$ for the training set, while $840s$ were necessary for the testing set. Figure 5 illustrates the resulting SS-based segmentation.

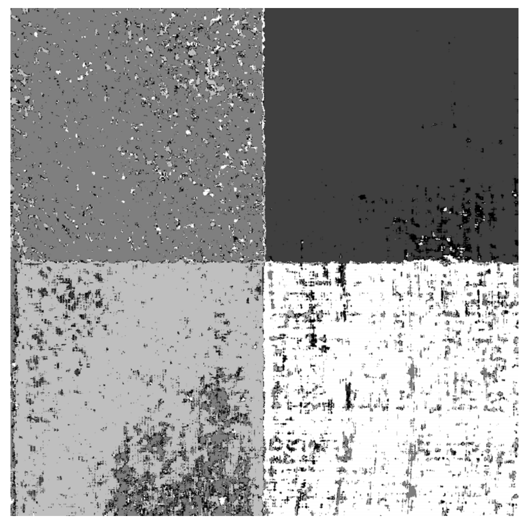

Figure 5. SS-based segmentation: best result.

Table 1 describes the success testing rates and the processing times of the four approaches. *Total time* is the sum of all five selections used to chose the best subset.

Clearly, SSCO arises as the best choice among the salience-based methods for both image and classifier used in the experiment, since the success rate is very close to that attained by SS (the best result), not to mention the selection processing time which is much inferior than the others. As for SJM approach, the addition of an extra feature to the desired subset would make it computationally infeasible for the exhaustive search. The addition of 10 or even 20 extra

[1]interruption of training process after 100 cycles without a 1% drop (at least) in the sum of quadratic error.

[2]interruption of training process has been altered so as to happen after 1000 cycles, when a 1% drop (at least) in the sum of quadratic error is not noticeable.

Method	SJM	SS	SS withou IE	SSCO
% success rate	86,2 (0,57)	86,6 (0,64)	71,0 (1,69)	86,4 (0,61)
selection time	8857	16548 (3672)	18882 (3641)	2052 (348)
total time	8857	82740	94410	10260
memory used	1,6GB	624MB	624MB	624MB

Table 1. Comparing all approaches: success rate for the testing set, selection time and RAM memory used. The standard deviation is given in parenthesis and time is given is seconds.

features to the desired subset size would make SJM impracticable, even for *branch and bound* search [7]. On the other hand, the computational cost for SSCO would remain the same when features are added, whereas complexity would even drop for SS approach: each new feature in the desired subset size means one less iteration in the selection process.

Selection times could be shortened if one a) reduces the number of times the process is conducted, or b) interrupts the training stage (i.e. the retraining that occurs at each iteration of the selection step) earlier. It should be pointed out that the stop criterion adopted in this work was rather conservative. The SSE (Sum of Squared Error) yielded values around 10^{-31} when, in fact, a value of $1,0$ would suffice to correctly classify $99,995\%$ of the patterns. It is possible that a better stop criterion be obtained other than the SSE. Another means to cut processing times is to take advantage of the MLP selector weights, using them as initial weight values for the MLP classifier. This would substantially reduce the classifier training time, except, of course, for the SJM approach.

Notice that the Input Equalization (IE), used in both SS and SSCO approaches, strongly contributed to burst the success rate of the system.

It should also be pointed out that the experiments produced very similar results. This is due to the generalization power of the MLP network and also its ability to find complex relations in data. To assess the quality of the selected features rather than the intelligence of the classifier, an additional experiment has been conducted.

4.7 Feature assessment by Minimum Distance Classifier

A rule of thumb in pattern recognition context is that the better the features, the simpler the classifier can be. From this premise, four experiments have been conducted but, unlike the MLP outcome of the previous sections, a Minimum Distance Classifier has been used instead. Four feature schemes have been defined: SJM, SS, SSCO and all 71 features (table 2).

The best subset, according to the Minimum Distance Classifier is that obtained with SS. The success rate is approximately the same as the average success rate of an MLP classifier. Another important outcome is the Minimum Distance Classifier success rate for all features, which is very similar to a random classification (25%). This results indicates how inadequate either bad features or the high dimensionality can be. It also reinforces the need of a feature selection stage in a pattern recognition system.

5 Application to segmentation

This section shows some results of the feature selection method by salience for two natural scene images [26] in the context of image segmentation. The training data set was extracted from each sample image, by manual selection of small squared regions. Each region represented a different class. The structure employed in the previous experiments was also adopted here, except for: a) a longer training period in the *feature selection stage* (process halted after 1000 cycles when the sum of the quadratic error is not reduced in at least 1%); b) process halted after 10 cycles in the *training stage* of the two last images; c) hidden layers with size 2 for both natural scenes.

The feature extraction process took 50 seconds and remained constant in all cases. The pixels within the squared classes of interest, for example, *water*, *sand* and *sky* constituted the regions from which features have been extracted. The pixels outside these squared boxes were used as the testing set, since they have not been used in neither the training stage nor the feature selection process.

The feature selection task consisted in running the process three different times and choosing the shortest subset, with training error below 2%. Hence, the size of the subset varied throughout experiments.

Figures 6 and 7 show, respectively, the original grey level image and its segmented counterpart for the first natural scene. Two features have been selected (11 and 13): grey level mean and median.

In the same manner, figures 8 and 9 are for the second natural scene. Two features have been selected (12 and 18): grey level standard deviation and mean value of Cooccurrence matrices variance for width 2.

Method	SJM	SS	SSCO	Combined schemes
% training success rate	82,0	90,6	86,6	91,2
% testing success rate	70,0	86,7	75,3	25,1

Table 2. Approaches compared with a simpler classifier: success rate for training and testing.

The selection, training and segmentation times are given in table 3

Notice that, mainly, selected features were those from the first order statistics. Visual inspection of such images in fact indicates that the region grey level is, probably, the most discriminant feature. Can be observed, as well, that the more complex the texture the more complex the feature (e.g the rocks from figure 8 were segmented with a Haralick feature).

It is also possible to infer that the selection by salience (SS) also identifies the amount of essential features, depending on the complexity of the problem.

Figure 6. Natural scene 1: training data indicated by a rectangle (17952 pixels sample).

Figure 7. Segmented natural scene 1: each gray level represents a class. Black color stands for unknown class.

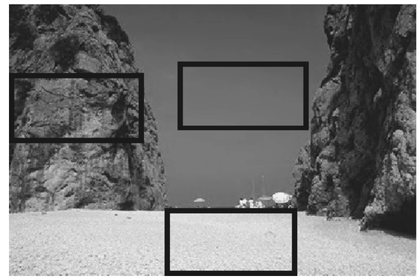

Figure 8. Natural Scene 2: training data indicated by a rectangle (15552 pixels sample).

Figure 9. Segmented natural scene 2: each gray level represents a class. Black color stands for unknown class.

6 Conclusions

This work explored the problem of feature selection by means of a MLP neural network with salience measure. Different variations of such approach have been shown and compared with an optimal statistical method.

One of the drawbacks of traditional statistical approaches is their complexity. Confronted with an MLP-based exhaustive search, the proposed method not only provides the ability to reduce the amount of training from 2^n to n as it also performs partial retraining. Results have shown that for subsets with more than 4 features (out of 71 computed texture features) the proposed method is more efficient than the exhaustive search based on the JM distance.

It has also been noticed that MLP-based classification systems do not benefit completely from a feature selection module. This is due to the fact that the MLP itself internally

Natural scene	1	2
selection time	120000	9180
training time	84	26
segmentation time	0,16	0,16

Table 3. Natural scene results: times are given in seconds. Selection time is the sum of all three computing times.

carries out this process when all features are present. Hence, when it comes to selecting features for an MLP, the main advantages are a) lower costs for feature extraction and b) drop in the overall complexity of the system. On the other hand, for systems based on *Minimum Distance Classifiers* and others of lesser complexity, the proposed feature selection approach also introduces the possibility of increasing the success rate of the classifier.

Salience measures as originally proposed [9] and used [21] so far, were highly correlated with the mean value of the features. The input equalization (IE) process, introduced in section 3 of this work, breaks this correlation. As a result, a boost in feature selection/segmentation performance has been achieved regardless the salience-based selection method adopted, as shown, for example, in table 1.

References

[1] A. K. Bachoo and J.-R. Tapamo. Texture detection for segmentation of iris images. In *SAICSIT 2005*, pages 236–243, Republic of South Africa, 2005. South African Institute for Computer Scientists and Information Technologists.

[2] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall Professional Technical Reference, 1982.

[3] O. Brigham. *The Fast Fourier Transform*. Prentice-Hall, 1974.

[4] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, NY, 1966.

[5] G. Castellano and A. M. Fanelli. Variable selection using neural-network models. *Neurocomputing*, 31(1-4):1–13, 2000.

[6] M. Clark, A. C. Bovik, and W. S. Geisler. Texture segmentation using gabor modulation/demodulation. *Pattern Recogn. Lett.*, 6(4):261–267, 1987.

[7] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[8] M. M. Galloway. Texture analysis using gray level run lengths. *Computer Graphics Image Processing*, 4:172–179, 1975.

[9] G. D. Garson. Interpreting neural net connection weights. *AI Expert*, 6(4):46–51, 1991.

[10] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 3rd edition, 1992.

[11] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. In *CMetImAly77*, pages 141–152, 1977.

[12] D.-C. He, L. Wang, and J. Guibert. Texture discrimination based on an optimal utilization of texture features. *Pattern Recogn.*, 21(2):141–146, 1988.

[13] T. Heskes and W. Wigerinck. A theoretical comparison of batch-mode, on-line, cyclic, and almost-cyclic learning. *Neural Networks, IEEE Transactions on*, 7(4):919–925, 1996.

[14] R. Huber and L. Dutra. Feature selection for ers 1/2 insar classification: High dimensionality case, 1998.

[15] A. K. Jain and D. Zongker. Feature selection: Evaluation, application and a small sample performance. *IEEE Transactions on PAMI*, 19:153–158, 1997.

[16] P. Laar, S. Gielen, and T. Heskes. Input selection with partial retraining. *Lecture Notes in Computer Science*, 1327:469–476, 1997b.

[17] P. Laar, T. Heskes, and S. Gielen. Partial retraining: A new approach to input relevance determination, 1997.

[18] P. Leray and P. Gallinari. Feature selection with neural networks. *Behaviormetrika (special issue on Analysis of Knowledge Repr*, 26(1), 1998.

[19] T. Masters. *Signal And Image Processing with Neural Networks*. John Wiley & Sons, Inc., New York, 1994.

[20] K. Muneeswaran, L. Ganesan, S. Arumugam, and K. R. Soundar. Texture image segmentation using combined features from spatial and spectral distribution. *Pattern Recogn. Lett.*, 27(7):755–764, 2006.

[21] R. Nath, B. Rajagopalan, and R. Ryker. Determining the saliency of input variables in neural network classifiers. *Computers & OR*, 24(8):767–773, 1997.

[22] L. Qin, W. Wang, Q. Huang, and W. Gao. Unsupervised texture classification: Automatically discover and classify texture patterns. *ICPR*, 2:433–436, 2006.

[23] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks*, 1993, 1993.

[24] D. W. Ruck, S. K. Rogers, and M. Kabrisky. Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, 2(2):40–48, 1990.

[25] M. Tuceryan and A. K. Jain. Texture analysis. *Handbook of Pattern Recognition and Computer Vision*, pages 235–276, 1998.

[26] J. Z. Wang. James Z. Wang's Research Group Image Repository, 2003. Accessed in 30/01/2007: http://wang.ist.psu.edu/docs/home.shtml.