

Published in Proceedings of "The Seventh IEEE Conference on Artificial Intelligence Applications", Feb. 24-28, 1991, Miami Beach, Florida

Application of Symbolic Machine Learning to the Recognition of Texture Concepts

J.W. Bala and P.W. Pachowicz
Center for Artificial Intelligence, George Mason University
4400 University Dr., Fairfax, VA 22030

Abstract: This paper presents an approach to the texture recognition problem that deals with noisy learning and testing data. The method incorporates symbolic machine learning to acquire texture descriptions. Then, these descriptions are optimized in order to remove some noisy/imperfect components. We present methodology and experimental results showing the increase in system recognition effectiveness when optimization of texture descriptions is proceeded continuously. Such a matching of partial concept prototypes with test data gives a recognition characteristics obtained for different concept optimization degrees. Then, the dynamics of this characteristics is used to make recognition decision.

1. Introduction

Since our environment contains textured objects and the texture data contains noise and other irregularities, the acquisition and robust recognition of such noisy and imperfect concepts is crucial for many vision tasks; e.g., inspection, object recognition, navigation, remote tracking and surveillance. The success of these tasks depends both on the effectiveness of image understanding processes and the quality of data a system has to deal with during both the learning phase and task execution phase. Once the system is trained and tuned to a given type of data, it is difficult to assume without testing that the system will be working with the same data but affected by a different noise. The problem is very difficult when training data contains noise and when one considers the development of autonomous systems working without human supervision.

The traditional approach to the texture recognition is focused on the following three phases; i.e. (i) designing and selection of a set of most representative features that preserve as much the discriminatory power of applied methodology as possible, (ii) acquiring descriptions of texture classes from teacher- or system-provided training data, and (iii) matching test data with derived class descriptions. The overall texture recognition effectiveness depends on the quality of all above presented phases.

The traditional approach applies classifiers in such a way that they are adapted on the feature set to take optimal advantage of the extracted information. Such a method belongs to the class of feature extraction oriented methods, where an extraction of relevant features plays a very important role [4]. The main problem with traditional approaches is that we do not have a universal feature extraction method that works effectively with noisy and imperfect data. When one considers that the training data

can be noisy and imperfect, one has to agree that the derived descriptions of texture classes consequently contain noisy components. Therefore, the recognition based on the direct matching of such noisy/imperfect description of texture classes with testing data follows all above mentioned disadvantages; i.e., the recognition results are affected by the noise and imperfectness of learning and testing data. This is one of the reasons why most efforts to the solution of the texture recognition problem (and other recognition problems) focus on the improvement of the feature extraction phase to reduce the influence of noise and data imperfectness on the system performance.

Contrary to the improvement of recognition effectiveness by designing more sophisticated feature extraction methods, we develop a methodology that improves the recognition effectiveness under the assumption that the training data provided during the learning phase is noisy/imperfect. Our methodology is based on: (1) the application of learning-based acquisition of texture class description, (2) the manipulation of acquired descriptions in order to optimize them, and (3) the recognition of test data by a dynamic tracking of classification decisions computed for the matched optimized descriptions.

In Section 2, we justify our approach to the problem of texture concept acquisition and recognition by the example of introductory experiments provided for the symbolic machine learning methodology. Modified system architecture and applied learning methodology to new experiments is discussed in Section 3. Section 4 presents the principle of an iterative manipulation of acquired texture concept descriptions in order to optimize them and apply to recognize test data. Selected experiment is presented in Section 5, while Section 6 summarizes our work.

2. Motivation and Justification

Considering the application of learning-based concept acquisition in texture domain, we have already shown the methodology and benefits of such an approach [10]. In one of our experiments, the system was able to improve the average recognition rate (for six classes of texture acquired from very poor image data) from 70% of correct recognitions obtained for the k-NN pattern recognition method [5], to 80% for the symbolic machine learning approach, and to 91% for the symbolic machine learning approach incorporating optimization of texture class descriptions and a single matching.

description improves the recognition effectiveness and smooths recognition rates.

In the experiment summarized above, the optimization of texture class description is controlled by parametric coefficients. If we request the higher optimization degree, we increase these coefficients. The increase or decrease of the optimization coefficients is based on the noise evaluation. While the noise is propagated from the input image data into the teaching examples, it is built into the components of texture description. Based on it, we tried to set up an optimization degree (coefficients) to the level of approximated noise in the input training data, for which the recognition effectiveness would be maximum. Such manipulation with the optimization coefficients has provided interesting observations presented in the next sections.

3. Modified System Architecture

New experiments were applied on the modified system architecture presented in Figure 2. The modifications included extraction of texture attributes, redesigning of the recognition system, implementation of new decision making methodology, and the creation of a control loop of the iterative optimization of rules.

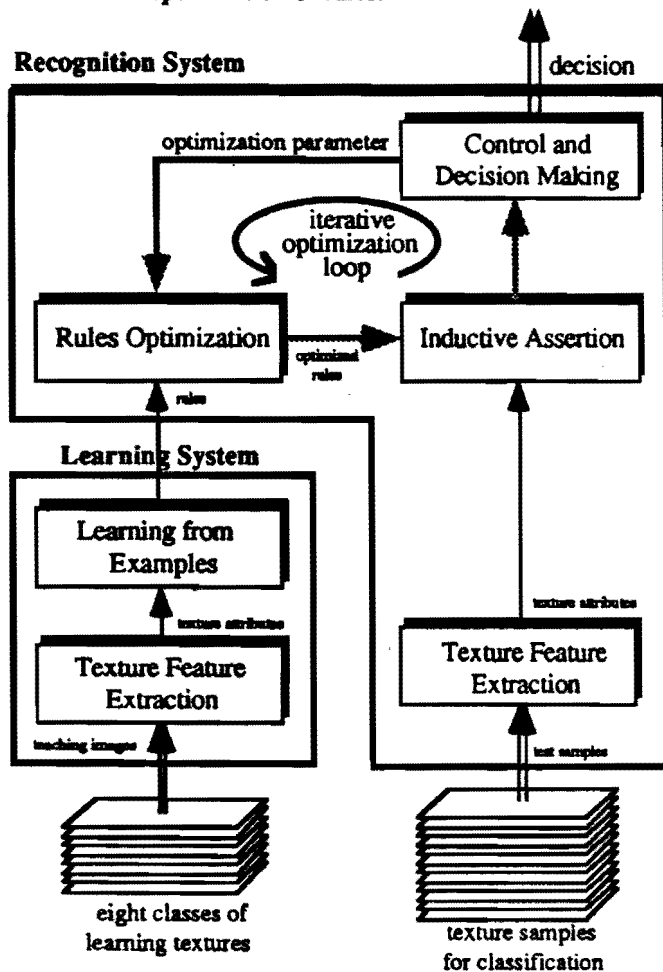


Figure 2: System architecture.

Presented to the system texture images were used to acquire attributional description of texture classes. Acquired rules were applied to the recognition phase, where testing texture samples have been shown to the system. The recognition process has been arranged into the iterative optimization loop. This loop consists of three modules: rules optimization, inductive assertion, and a module of control and decision making. The loop is controlled by an optimization parameter and it activates inductive assertion. Inductive assertion processes are performed each time for optimized rule descriptions of texture classes. The system increases the optimization degree for each iteration loop. The decision making module completes partial classification probabilities computed for each optimization loop. In this way, a recognition curve is created versus the optimization degree.

3.1 Extracting Texture Attributes

Contrary to the development of more powerful noise reducing methods, we have applied the simplest extraction of texture features that preserves the negative influence of noise. Texture attributes have been extracted from input image in the following way. A five by five feature extraction window, presented in Figure 3, was applied to extract eight local texture attributes as the absolute difference of a neighboring pixel and the central pixel grey values; i.e.,

$$a_k = |x_{\#} - x_{k-1}| \quad \text{for } k=1, \dots, 8 \quad (1)$$

where $x_{\#}$ is a grey level value of the central pixel, and x_k is a grey level value of a neighboring pixel. The event as a vector of eight attribute values $\langle a_1, a_2, \dots, a_8 \rangle$ was derived for each randomly selected pixel from the learning or testing image data.

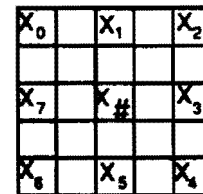


Figure 3: Feature extraction window.

3.2 Learning Texture Description from Examples

The attributional description of texture classes was acquired by the learning from examples methodology. We applied the AQ15 inductive learning program. The AQ programs perform a heuristic search through a space of symbolic expressions, and its goal is to find the most preferred expression according to a specified criterion [7,8]. The input consists of separated sets of learning events labeled by a class name and characteristic for each class of texture, where each event contains a given number of attributes. The set of events characteristic for one class is a set of positive examples. The program finds an optimal cover over only positive examples, where examples belonging to the other classes are negative examples. The learning

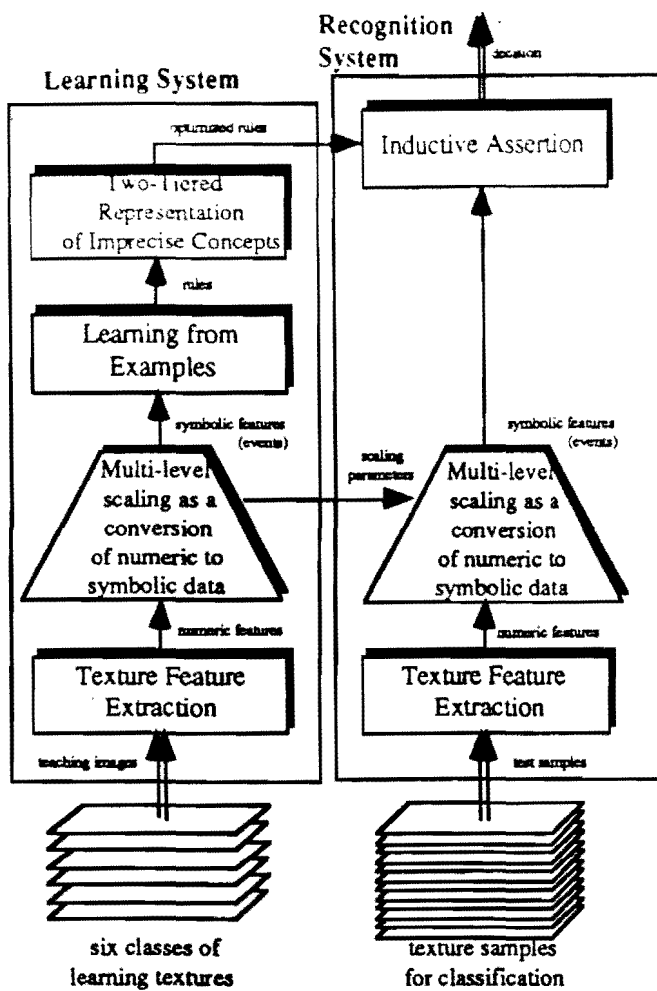
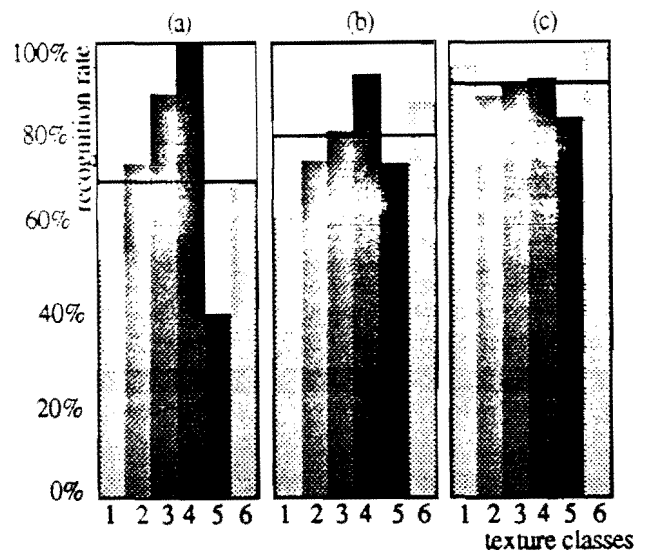


Figure 1: Open-loop architecture for texture concept acquisition and recognition.

The traditional open-loop architecture, presented in Figure 1, was used with distinction of learning and recognition systems. Image data for the learning phase was composed of six preclassified texture images from the Brodatz album [1], i.e., pressed cork, lawn, woolen cloth, water, pigskin and fur. The images were characterized by irregular lighting and several of them had smoothly changed resolution caused by the slanted projection. Input grey-level images were processed by well-known techniques for texture features extraction; i.e., Laws' masks [6] and co-occurrence matrices [3,13]. A vector of eight attributes was extracted for a single pixel, and for each method of feature extraction. The scaling module performed the conversion of numeric features into their symbolic intervals. The consistency of learning data was checked. The AQ inductive learning algorithm [9] was applied to learn texture attributional descriptions from symbolic events. Next, these rules were optimized incorporating two-tiered representation of imprecise concepts [9].

The texture recognition system was composed of feature extraction, scaling, and inductive assertion modules. The first two modules were similar to the corresponding modules of the learning phase. The inductive assertion module performed classification of test data samples using learned rule descriptions [11].



	PR approach k-NN (a)	ML approach rules (b)	ML approach optimal rules (c)
Average recognition	70%	80%	91%
Highest recognition	99%	93%	98%
Lowest recognition	40%	72%	83%
Averaged deviation	16.3	7.0	4.0

Table 1: Comparison of machine learning and pattern recognition approaches to texture recognition (a bar represents an average recognition).

The best results (Table 1) were obtained for eight Laws' masks. The average recognition was equal to 91% and the system recognized all test textures correctly. The highest recognition rate was equal to 98% and the lowest recognition rate was equal to 83%. The averaged deviation defined as $d = 1/N \sum |x_i - x_{il}|$ was the lowest and equal to 4.0. The experiments that applied texture classes description using inferred non-optimized rules show relatively worse results. The average, highest and lowest recognition rates were equal to 80%, 93% and 72%, respectively. The averaged deviation was greater and equal to 7.0. In comparison, the worse performance was received applying k-NN pattern recognition method (applied with different k values) that deals with complex distribution of feature space. The average, highest and lowest recognition rates for the k-NN (k=10) pattern recognition method were equal to 70%, 99% and 40%, respectively. The averaged deviation was greatest and equal to 16.3.

The analysis of this introductory experiment proved that (i) the application of machine learning methodology increases the recognition effectiveness, and (ii) optimal rule

process is repeated for each class (a set of positive examples) to create rule descriptions of all texture classes.

A description of a class is a disjunctive normal form which is called a cover. A cover is a disjunction of complexes. A complex is a conjunction of selectors, and a selector is a form:

$$[L \# R] \quad (2)$$

where, L is called the referee which is an attribute, R is called the referent which is a set of values in the domain of the attribute in L, # is one of the following relational symbols: =, <, >, >=, <=, <>. In the AQ15 program, each generated complex is associated with a pair of weights; i.e. total (t-weight) and unique (u-weight). The following is the example of a complex:

$$[x1=1..3][x4=0][x6=1..7][x8=1] \quad (\text{total:6, unique:2}) \quad (3)$$

where x_i is an attribute. The t-weight of a complex is the number of positive examples covered by the complex, and the u-weight is the number of the positive examples uniquely covered by the complex.

3.3 Rule Optimization

The requirement of "noise free domain" cannot be satisfied in the real world, especially in such areas like signal processing and understanding, autonomous robotics, and intelligent systems. The flexibility of a concept, the variability of concept occurrences (caused by external conditions like resolution, illumination and sensor positioning), and the noise in learning datasets prevent most learning systems from being applied to many real world learning tasks. On the other hand, challenging applications of machine learning methodologies in the engineering domain can give us the possibility of creating more intelligent machines.

Our approach to dealing with the negative influences of noise, the imperfection of models and acquired descriptions applies the rule optimization in order to remove less significant components (complexes). In this way, we assume that the lightest complexes can represent noisy components of the rule description and at least they represent the less typical characteristics of a texture class.

To optimize a rule description, we use the truncation method [9,14]. In the AQ15 program, each generated complex is associated with a pair of weights; i.e., the t-weight (as the total number of positive examples covered by a complex) and the u-weight (as the number of positive examples uniquely covered by the complex). The complexes of a concept description are ordered according to the decreasing values of the t-weight. The t-weight may be interpreted as the measure of typicality or the representativeness of a complex as a concept description. The complexes with the highest t-weight may be viewed as describing the most typical concept examples, and thus serves as its prototypic description.

Suppose we have a t-weight ordered disjunction of complexes, and we remove from it the "lightest" complex; i.e. the complex with the lowest t-weight. Such a truncated description will not strictly match events that do not uniquely satisfy the removed complex. However, by applying a flexible match (see next section), these events may still be closely related to the correct concept, and thus be correctly recognized.

We can proceed further and remove the second "lightest" complex from the cover, and observe the performance. Each such step produces a different trade-off between the complexity of the description on the one hand, and the risk factor along with the evaluation complexity on the other. At some step the best overall result may be achieved for a given application domain.

3.4 Flexible Matching and Inductive Assertion

There are two methods for classifying the concept membership of an instance; i.e. the strict match and the flexible match. In the strict match, one tests whether an instance strictly satisfies the condition part of a rule (i.e., one of rule complexes). Such matching gives the response of two possible logical values; i.e., true or false. In the flexible match, one determines the degree of closeness between the instance and the condition part. Such closeness is represented by a coefficient that can vary in range from 0. (does not match) to 1.0 (matches). In the strict matching, one recognizes a concept if it overlaps with the concept description. In the flexible matching, one determines the most closely related concept description.

The truncated description is the one that has some of its complexes with the lowest u-weight removed. Such truncated description will not strictly match events that uniquely satisfy the truncated complex. So, to evaluate the membership of an event to a texture class, one has to apply a flexible match. Such flexible matching can improve the classification decision when less significant components (i.e., possibly noisy components) are truncated from the class description.

In the recognition phase, we incorporated a special tool, ATEST program [12], that had been developed to test the performance of a rule base. Rule performance is measured by the degree of agreement between the system's and expert's classifications. ATEST views rules as the expressions when applying them on a vector of attribute values. The result of this evaluation is a real number which is the degree of consonance between the conditional part of the rule and the event.

4. Concepts Recognition via Iterative Optimization of Their Descriptions

4.1. Reversing Classification Decision

Existing approaches to the recognition problem assume that to recognize a concept one needs to match the observed data with a stored description of concepts. When the properties

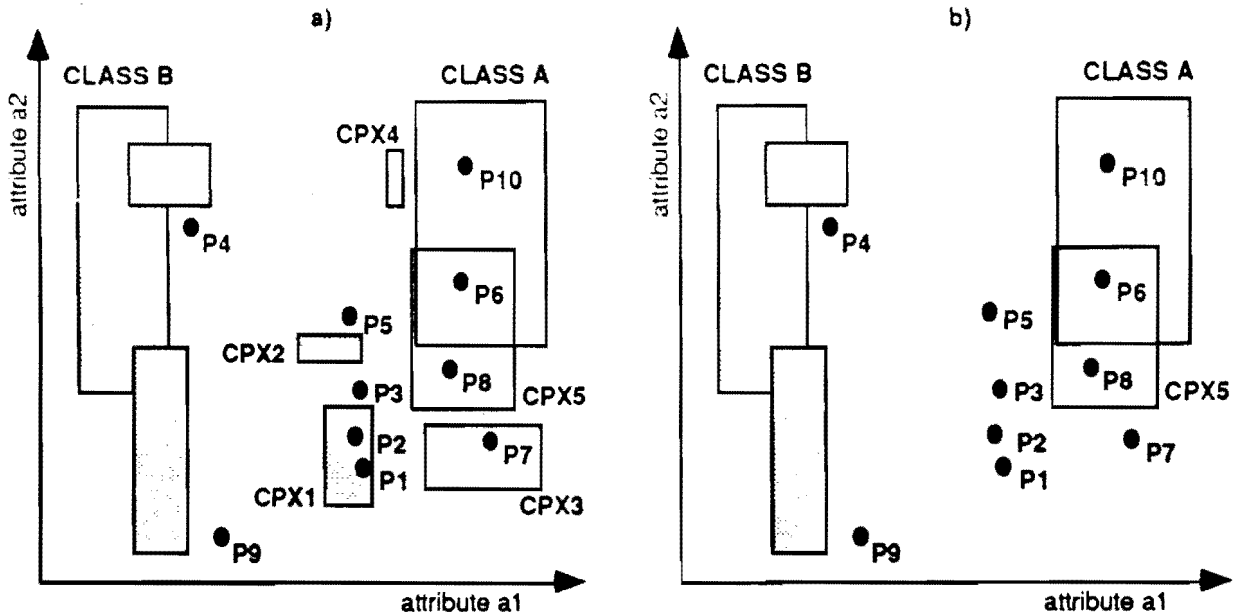


Figure 4: An example for reversing the recognition decision for a set of 10 testing events: a) before truncation 4 events are classified to the class A and 6 events to the class B, b) after truncation of CPX1, CPX2, CPX3 and CPX4 (i.e., truncation degree is equal 2) complexes classified to the class A and only 2 events to the class B.

of the observed object match the stored description of a concept, then the name of the concept is returned. An approximate match is computed and used as a measure of confidence in recognizing particular concept among others. A major problem with such a method is that in order to recognize an object one always needs to measure the same properties of it. Yet, people can recognize the same concept using many different subsets of properties, i.e., with truncated concept descriptions.

In our method, we recognize textures by matching unknown texture samples with learned descriptions on the different truncation levels. The description on a given level is obtained by truncating a given number of less significant complexes from the lower part (i.e., less significant part) of the description. At each level, we match testing samples with given descriptions of texture classes and complete recognition rates.

Let us demonstrate how such a truncation method (as the optimization of concept description) can reverse the classification decision. Figure 4 presents a graphical representation of two class descriptions, i.e. class A and class B, where the domain of events consists of two attributes. Class A is described by four complexes, and class B by six complexes. The black dots represent sample vectors to be recognized. For the initial description of these two classes (Figure 4a), four events (i.e., P6, P7, P8 and P9) are assigned to class A and six events (i.e., P1, P2, P3, P4, P5 and P9) are assigned to class B. By truncating CPX2 complex from class B, P5 event is assigned to class A by closeness with CPX5 complex, and the recognition membership (the number of samples classified to class A versus the number of samples classified to class B) is changed from 4-6 to 5-5. Further truncation of complex CPX1 of class B (see Figure 4b) changes the recognition membership to 8-2 and class A yields the recognition.

Notice that complexes are also simultaneously truncated from class A for the consecutive truncation degrees; i.e., complexes CPX3 and CPX4.

Theoretically, such reduction of rule description should lead to the extraction of main clusters of learning data. Applied flexible matching, then should classify test data to the nearest cluster. However, there is the question of what is the best optimization degree? It is rather impossible to answer to this question because the best optimization degree depends on characteristics of the feature distribution in the attribute space. If this distribution is complicated, for example, there are several local clusters in the description of one class, the optimization cannot be executed for relatively higher optimization degree.

4.2. Recognition Algorithm

If we accept that the truncation of less significant components from the concept description eliminates some noise, then the dynamic increase of the truncation degree (but performed in the lowest range of optimization values) should slightly increase the recognition effectiveness in this range. Such effect was observed in our first experiments [10]. We concluded then that a recognition curve created for each class description and for different values of the truncation degree could be a useful classification feature for the final decision making process.

Experiments with the learning and recognition of noisy and imperfect concepts provided for the texture domain have shown the behavior of such recognition curve. Generally, two patterns of the recognition curve are observed regarding the first phase of the increase of the truncation degree: (1) curves of the recognition uptrend, and (2) curves of the recognition downtrend. Based on the behavior of these curves, we define the following recognition algorithm that

incorporates iterative optimization of concept description and flexible matching with test data:

- Step 1: Label the first section of each recognition curve as uptrend or downtrend recognition pattern,
- Step 2: Select these recognition curves that have the uptrend recognition pattern only, and
- Step 3: Make the final decision indicating this class for which the uptrend pattern runs through the highest recognition rates.

5. Recognition Results and Their Interpretation

The experiment presented in this paper was carried out with four texture classes (Figure 5) taken from the Brodatz album [1]; i.e., reptile skin (class A), cotton canvas (class B), fur (class C), and beach sand (class D).

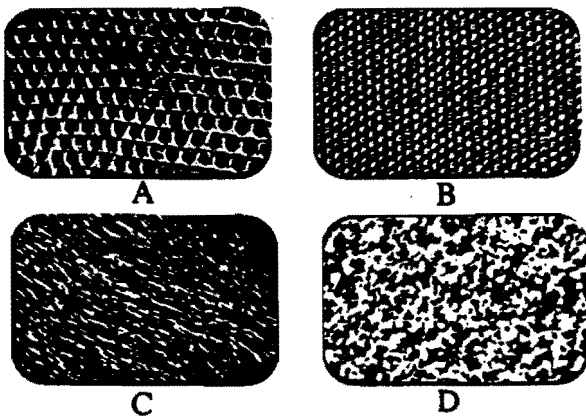


Figure 5: Samples of texture classes used for experiment.

A set of small sections of images was shown to the system during the teaching phase, while the other set of image sections was shown during the recognition phase. Before the feature extraction was executed (for learning and recognition phases), all textural images were preprocessed. The number of pixel grey levels was changed from 256 to a number of levels below 55. Training examples (i.e., 200 examples for each class) were selected randomly from texture images. Then, the system acquired texture class descriptions incorporating learning from examples methodology.

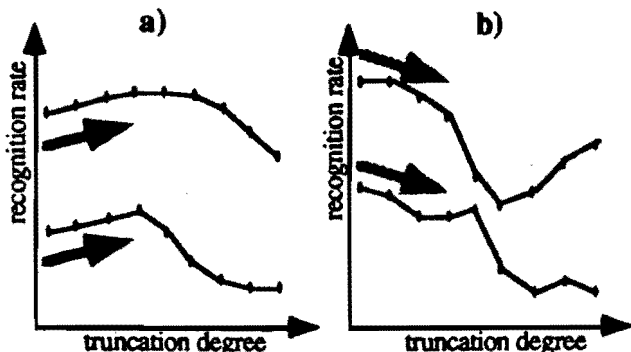


Figure 6: Examples for (a) the uptrend pattern and (b) the downtrend pattern of the recognition curve.

In the recognition phase, the system selected 100 testing events for each texture class but from different sections of texture image. The recognition process has been performed on such four sets of test data as explained in the above sections. Reconstructed recognition curves are presented for each testing dataset in Figure 7. All uptrend sections of the recognition curves are indicated by an arrow. The solid curve represents the class that should be recognized correctly and it was recognized by our method. Other classes are characterized by dotted lines. The truncation degree corresponds to the consecutive steps of rules optimization -- all rules were optimized in each step by the truncation of a given number of less significant complexes. The recognition rate was computed as the number of test samples recognized to a given class divided by the total number of test samples of a given dataset. It should be pointed out that a single testing event can be recognized to more than one class when it is covered by complexes of different classes or the distance to such complexes is the same.

Figure 7 shows that if one applies our methodology, all four classes are recognized correctly. But if one applies traditional single match method, two texture classes (i.e., classes B and D) are not recognized.

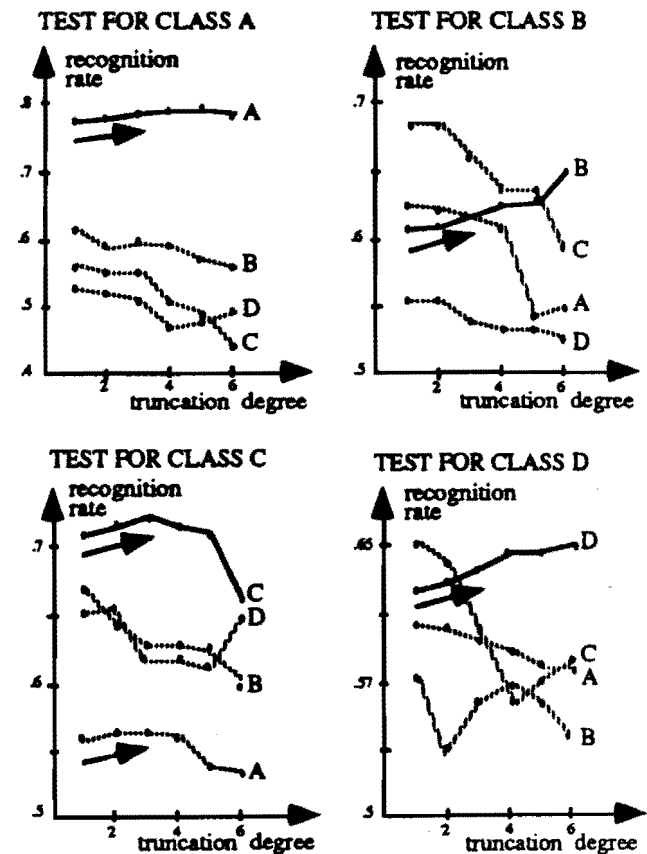


Figure 7: Recognition curves.

Based on these results, let us discuss the major advantages of introduced methodology following the recognition results for classes A and C versus the recognition results for classes B and D. Classes A and C are recognized correctly by the traditional single match approach without

consideration of the dynamic behavior of the recognition curves. Both of these classes have a typical uptrend pattern of the recognition curve, while other curves have the downtrend pattern. Classes A and C are recognized correctly by introduced new method, as well. The decision making process, however, performed for the test for class C considers two uptrend recognition curves; i.e. classes A and C. The final decision points out the class C because its recognition rates for the uptrend section of the recognition curve are higher than for the class A.

Classes B and D are not recognized if one applies traditional single match method. However, they are successfully recognized applying presented in this paper methodology based on the analysis of dynamic behavior of the recognition curve.

7. Conclusions and Future Work

We have presented a novel approach to the recognition of objects on the example of texture that incorporates machine learning methodology and deals with noise/imperfect data. This approach was tested on texture recognition problem, where features were extracted by a very simple method. Some of these results were presented in this paper to illustrate introduced methodology, to present its effectiveness in such cases where traditional approaches failed to recognize objects correctly.

This work follows our former observations and results [10]. Developed object recognition methodology is being applied to create an intelligent vision system for outdoor navigation of autonomous robots and for remote surveillance systems [11]. These systems have to acquire visual concepts both in the supervised and the unsupervised mode. While the supervised mode guarantee by a teacher a good representation of training data, the unsupervised acquisition must be based on the collection of training examples provided by the system itself. In such case, the influence of imperfectly collected data on the final decision has a crucial role in the later recognition phase. Our future work also includes the formalization of introduced methodology, the modification of truncation algorithm, and investigation of its effectiveness in various object recognition problems.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. Ryszard Michalski and Dr. Harry Wechsler for valuable comments, Janet Holmes and Janping Zhang for discussion and technical help. This research was done in the Artificial Intelligence Center of George Mason University. Research activities of the Center are sponsored in part by the Defense Advanced Research Projects Agency under grant, administrated by the Office of Naval Research, No. N00014-87-K-0874, in part by the Office of Naval Research under grant No. N00014-K0226, and in part by the Office of Naval Research under grant No. N00014-88-K0397.

REFERENCES

[1] Brodatz, P., "A Photographic Album for Arts and Design", Toronto, Dover Publishing Co., 1966.

- [2] Bergadano, F., Matwin, S., Michalski, R.S. and Zhang, J., "Learning Two-Tiered Descriptions of Flexible Concepts: The POSEIDON System", submitted to Machine Learning Journal, 1990.
- [3] Davis, L., Johns, S., and Aggarwal, J., "Texture analysis using generalized co-occurrence matrices", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-1, pp. 251-259, 1979.
- [4] DuBuf, J.M.H., Kardan, M. and Spann, M., "Texture Feature Performance for Image Segmentation", *Pattern Recognition*, Vol.23, No.3-4, pp.291-309, 1990.
- [5] Duda, O.R., Hart, P.E., *Pattern Classification and Scene Analysis*, John Wiley&Sons, 1973.
- [6] Laws, K.I., "Textured Image Segmentation", PhD Thesis, Dept. of Electrical Engineering, University of Southern California, Los Angeles, 1980.
- [7] Michalski, R. S., "A Theory and Methodology of Inductive Learning", in *Machine Learning: An Artificial Intelligence Approach*, TIOGA Publishing, Palo Alto, CA, pp 83-134, 1983.
- [8] Michalski, R.S., Mozetic I., Hong J.R., Lavrac N., "The AQ15 Inductive Learning System", Report No. UIUCDCS-R-86-1260, Department of Computer Science, University of Illinois at Urbane-Champaign, July, 1986.
- [9] Michalski, R.S., "Two-Tiered Concept Meaning, Inferential Matching and Conceptual Cohesiveness", in "Similarity and Analogy", S. Vosniadou and A. Orton (Eds.), Cambridge University Press, 1987.
- [10] Pachowicz, P.W., "Integrating Low-Level Features Computation with Inductive Learning Techniques for Texture Recognition", *International Journal of Pattern Rec. and Artificial Intell.*, Vol.4, No.2, pp147-165, 1990a.
- [11] Pachowicz, P.W., "Learning-Based Architecture for Robust Recognition of Variable Texture to Navigate in Natural Terrain", Proc. IEEE International Workshop on Intelligent Robots and Systems '90, Tsuchiura, Ibaraki, Japan, July 1990.
- [12] Reinke R. E., "Knowledge Acquisition and Refinement Tools for the Advice Meta-Expert System", ISG 84-4, Department of Computer Science, University of Illinois at Urbana-Champaign, July 1984.
- [13] Van Gool, L., Dewaele, P., Oosterlinck, A., "Texture Analysis Anno 1983", *Computer Vision, Graphics and Image Processing*, Vol.29, pp.336-357, 1985.
- [14] Zhang, J., Michalski, R.S., "Rule Optimization via SG-TRUNC Method", Proc. of the Forth European Working Session on Learning, pp.251-262, Montpellier, Dec. 1989.

