Proceedings of the 8th International Symposium on Artificial Intelligence, 124–131
Monterrey, Mexico, October 17-20, 1995

LEARNING SYMBOLIC DESCRIPTIONS OF 2D SHAPES FOR OBJECT RECOGNITION IN X-RAY IMAGES

Marcus A. Maloof **Ryszard S. Michalski***

Machine Learning and Inference Laboratory
George Mason University
Fairfax, VA 22030-4444
{maloof, michalski}@aic.gmu.edu

* Also, Institute of Computer Science
Polish Academy of Science

ABSTRACT

This paper describes a method for learning shape descriptions of 2D objects in x-ray images. The descriptions are induced from shape examples using the AQ15c inductive learning system. The method has been experimentally compared to *k*-nearest neighbor, a statistical pattern recognition technique, and artificial neural networks. Experimental results demonstrate strong advantages of the AQ methodology over the other methods. Specifically, the method has higher predictive accuracy and faster learning and recognition rates. The application considered is detecting blasting caps in x-ray images of luggage. An intelligent system performing this detection task can be used to assist airport security personnel with luggage screening.

INTRODUCTION

This paper concerns the development of a methodology for shape learning and recognition, and its application to learning symbolic descriptions of blasting caps in x-ray images under varying perceptual conditions. Task-oriented segmentation and event extraction are used to isolate objects of interest in images and to form a set of training examples in a pre-defined representation space. The training examples are passed to a learning system that produces general descriptions of the concepts to be learned.

The concept descriptions are then validated using testing examples of blasting caps and non-blasting caps, prepared apriori by a human operator. After validation, the learned descriptions are ready to be used for classifying unseen objects into blasting caps and other objects. The classification process uses a *flexible matching* method that determines a degree of match between an object to be classified and the obtained concept descriptions, rather than a strictly logical yes-or-no match.

This research intends to both make a contribution to the methodology of vision through learning and solve an important practical problem. An intelligent system capable of identifying blasting caps in x-ray images could assist airport security personnel with luggage screening.

The proposed method uses the AQ15c inductive rule learning system for generating blasting caps descriptions. To evaluate the proposed method, the descriptions obtained by the proposed learning system and other systems were compared in terms of their predictive accuracy and learning and recognition rates. Experimental results demonstrated strong advantages of the proposed method over *k*-nearest neighbor [17], a statistical pattern recognition technique, and artificial feed-forward neural networks [20]. Other experiments using this methodology have been reported by Maloof and Michalski [8] and Bala et al. [2].

The next section provides background on the AQ15c learning system and flexible matching, and reviews relevant applications of machine learning techniques to problems in machine vision. Section 3 describes the proposed method, and section 4 details obtained results. Section 5 discusses the contribution and significance of the results and outlines plans for future research.

BACKGROUND

An emerging new research area, Machine Vision through Learning (MVL), investigates the applicability of modern machine learning methods to problems in machine vision and the development of vision systems with learning capabilities [14]. From the computer vision standpoint, research in this area may help simplify the development of vision systems and increase their flexibility and adaptability. From the machine learning standpoint, its goal is to produce new learning methodologies capable of dealing with the complexities of visual perception. Potentially, machine learning methods can be used for a great variety of computer vision problems. This paper concentrates on one specific problem area, that of application of symbolic learning to shape description of 2D objects.

The AQ Learning Approach

In the AQ learning approach that is used in this study, a visual concept description is in the form of decision rules. Each rule is a conjunction of relational statements, each involving typically (but not necessarily) one attribute. The description is induced from a set of training examples and problem domain knowledge. Each training example is a vector of values of multitype discrete attributes (nominal, linear, or structured) with an indication of the decision class (visual concept) it belongs to.

Advantages of this approach include relatively high speed of learning, the possibility of parallel rule execution (and thus high speed of object recognition), ease of introducing and utilizing domain knowledge, the ability to work with large number of attributes to detect irrelevant attributes, high understandability of the learned concept descriptions, and the ability to generate descriptions of different types and different levels of generalization. Disadvantages include the need for quantizing continuous attributes, and a limited power of the descriptive language (the use of axis-parallel discriminating surfaces). These disadvantages can be reduced by introducing *flexible concept matching* techniques (which create much more complex, non-axis parallel concept boundaries), and *constructive induction* which allows the system to create derived attributes that can represent complex representation space transformations and concept boundaries.

In this study, we use inductive learning system AQ15c, in which concept examples, domain knowledge, and concept descriptions are built using an attributional representation language, called Variable-Valued Logic One, or VL_1 [1]. To make this paper self-contained, we begin by characterizing very briefly the description language. VL_1 decision rules (that can represent concept examples, domain knowledge and concept descriptions) are in the form:

$$D_i \; <:: \; C_i$$

where

D_i is the *decision* part of the rule, and is typically in the form of one *elementary statement* that assigns a value to a decision variable,

C_i is the *condition* part of the rule, and stated in the form of a conjunction of elementary statements (such a conjunction is also called a “*complex*”), and

<:: is the *decision assignment operator* (logically equivalent to implication).

An elementary statement (also called an *elementary condition* or “*selector*”) is in the form:

‘[’ <*referee*> <*relation*> <*referent*> ‘]’

where

<*referee*> is a member of the finite set of attributes,

<*relation*> is a relational operator (=, <>, >, <, >=, <=), and

<*referent*> is a subset of the domain of <*referee*>.

For example, [length > 2mm] and [color = red v blue] are elementary conditions. An elementary condition is satisfied by an object, if the value of the attribute stated in the condition for this object satisfies the <*relation*> between the <*referee*> and the <*referent*>.

In a *crisp* or *strict* matching convention, a given decision class is assigned to an object if the properties of an object satisfy the condition part of the rule. In a *flexible* matching convention, the rule is satisfied if the degree of match between an object and the rule is higher than the degree of match between the object and other candidate rules.

In the default parameter setting, AQ15c creates a maximally general hypothesis (a ruleset) that describes all the training examples and no negative examples. Negative examples of a given concept either are explicitly labeled as such, or in the case of multiple concept learning, are positive examples of all other concepts to be learned.

Finding a hypothesis that contains the minimum number of rules is a form of the general set covering problem [9]. Since this problem is NP-hard, the AQ algorithm (that underlies program AQ15c) solves this problem in a quasi-optimal manner, that is, finds a solution that is optimal or near-optimal.

Briefly, the AQ algorithm randomly selects one of the positive training examples (referred to as the *seed*) and then builds a set of alternative, maximally general descriptions of this seed (referred to as the *bounded star*). A domain-dependent preference criterion is used to select the most preferable rules from the bounded star. If the current description (the set of rules obtained so far) covers all positive examples, then the algorithm stops; otherwise, a new seed is selected from the yet-uncovered positive examples and the process repeats.

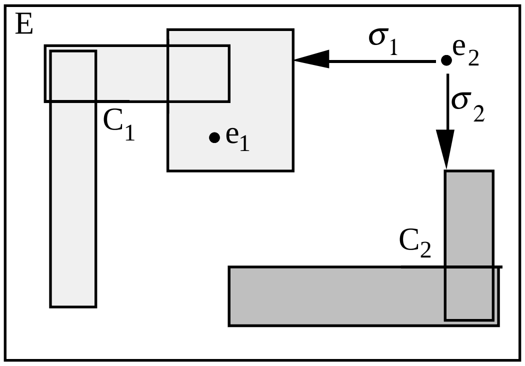
The AQ algorithm guarantees completeness and consistency of learned concepts. (Completeness means that the learned concept covers all positive examples. Consistency means that a learned concept does not cover any negative examples.) When examples are noisy, the system may make hypotheses that are partially incomplete or inconsistent.

Recognition Through Flexible Matching

After learning and validation, generated concept descriptions are incorporated into a system and can be deployed for concept recognition. Under a strict matching convention, if the example’s attribute values satisfy the condition part of a rule, then the decision class to which the rule belongs is assigned.

Conceptually, rules carve out decision regions in an event (representation) space, defined by the attributes chosen to characterize objects. Figure 1 illustrates such regions: the concept description C_1, consists of three rules (illustrated by overlapping rectangles), and concept description C_2 consists of two rules.

Figure 1: An illustration of concepts examples and their descriptions.



If an unclassified example, such as example e_1 in Figure 1, falls within one of the decision regions, it is assigned the decision class associated with that region. Thus, example e_1 would be assigned to decision class D_1. If an example does not fall within any decision region, as is the case with example e_2, then in the *strict matching* mode it is not assigned any decision. In the *flexible matching* mode, a degree of match (some function of the distance) between the example and all the concept regions is computed, and the best match indicates the decision class. Flexible matching thus helps to alleviate the brittleness associated with strict rule-based reasoning systems.

Several flexible matching schemes exist to calculate the degree of match between examples and concepts. The method employed here works as follows [18]**.** The degree of match σ_i between the example e_2 and the concept description C_i consisting of n complexes is given by:

$$\sigma_i = \sum_{j=1}^{n} \frac{\alpha_{ij}}{\beta_{ij}} - \prod_{j=1}^{n} \frac{\alpha_{ij}}{\beta_{ij}} \qquad (1)$$

where

α_{ij} is the number of conditions in rule j of concept C_i satisfied by example e_2, and
β_{ij} is the total number of conditions in rule j of concept C_i.

Formula 1 yields a real number in the range [0, 1], where 0 represents no match and 1 represents complete match.

Related MVL Work

Shepherd [16] used a decision tree learning algorithm to classify shapes of chocolates for an industrial vision system. Using feature vectors to represent examples, Shepherd compared a decision tree algorithm, *k*-nearest neighbor (*k*-nn), and a minimum distance classifier using classification accuracy. Classification accuracies for these learning methods were comparable with the minimum distance classifier, which produced the highest accuracy of 82%.

Cromwell and Kak [5] characterized object shapes using feature vectors for images containing electrical components such as resistors, capacitors, and transistors. Concepts were learned by applying inductive generalization rules and selecting the concept that explains the most examples from the training set. Their induction methodology was based on Michalski [12]. The average classification accuracy for their system was 72%. No comparisons were made to other learning systems.

The foundations for applying AQ learning to recognition problems in vision were laid by Michalski [10, 11]. In those seminal papers, AQ was used to learn relationships between image objects and concepts for discriminating between classes of images (textures and simple structures). Windowing operators were used to extract low-level features from texture samples, which were presented to the AQ learning algorithm. These ideas were further developed by Channic [4], who used convolution operators (e.g., the Kirsch operator) in conjunction with windowing operators for feature extraction. Channic proposed learning from sequences of images and used iterative and incremental learning for inducing multilevel texture descriptions from ultrasound images of laminated objects. Pachowicz and Bala [15] used the above AQ methodology with a modified set of Laws' masks for texture feature extraction and applied various description optimization techniques (e.g., the SG-TRUNC method [19]) for alleviating problems encountered by the introduction of noise. Bala [1] also introduced and applied to texture recognition methods for concept optimization (AQ-GA), and learning a large number of classes (PRAX).

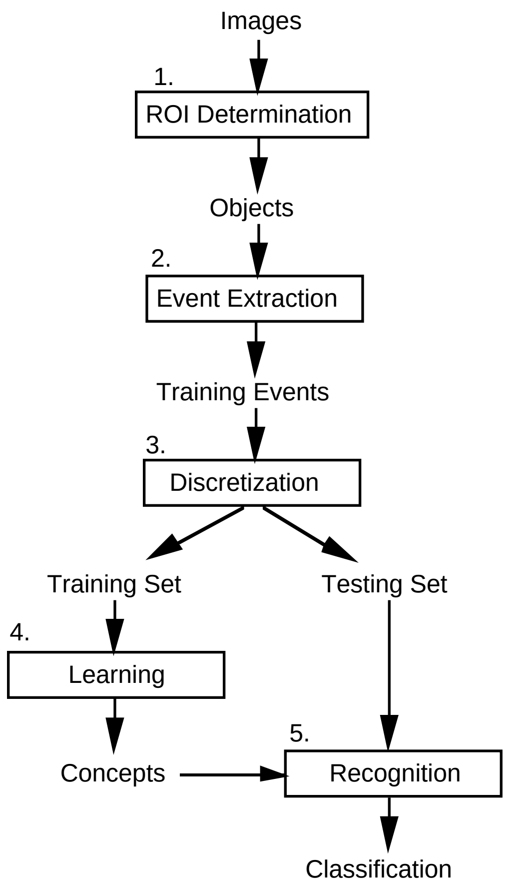
METHODOLOGY

The symbolic learning methodology used here closely parallels Michalski's [11] and Bala's [1], and proceeds through a five step process: (1) Region of Interest (ROI) Determination, (2) Event Extraction, (3) Discretization, (4) Learning, and (5) Recognition (see Figure 2). These steps are described in the following subsections.

Determination of Regions of Interest (ROI)

The first step involves determining which image regions are of interest, i.e., likely contain blasting caps. For illustration, Figure 3 shows two sample images of luggage containing blasting caps. In the experiments, we used an image set consisting of 30 images. The images were obtained by x-raying luggage

Figure 2: Basic steps of the learning and recognition methodology.



containing blasting caps, as it be would in an airport scenario: flat in relation to the x-ray source, but rotated in the plane orthogonal to the x-ray source.

Examples of airport luggage were constructed by placing different types of blasting caps at different positions and orientations in the bag and adding other objects, such as shoes, calculators, bolts, pens, and the like. For this initial study, 5 of the original 30 images were selected. The selected images were of low to moderate complexity in terms of blasting cap positional variability, degree of occlusion, and clutter.

Regions of interest were isolated and selected in the following manner:

1. Convolve image with 5x5 Gaussian.
2. Convolve image with 5x5 Laplacian.
3. Equalize Histogram.
4. Threshold image at the mode of the pixel distribution.
5. Select user-identified ROI or objects.

Operations 1–4 yielded binary images, from which an expert selected 53 objects and divided into two classes: blasting caps, containing 22 objects, and non-blasting caps, containing 31 objects.

Event Extraction

After ROI determination, image objects are described in terms of values of the attributes defining the representation space for learning. The attributes included:

1. `Area`: Area of an object
2. `Length`: Length of the object's perimeter
3. `Major`: Length of the major axis of a fitted ellipse,
4. `Minor`: Length of the minor axis of a fitted ellipse
5. `Compactness`: Ratio between area and perimeter.

Table 1 describes the representation space for this problem (the attributes and their ranges).

Table 1. Representation space for blasting caps.

Attribute	Possible Range
Area	37....675
Length	22.40....169.80
Major	8.04....58.35
Minor	3.39...25.72
Compactness	0.16....0.87

Values of the selected attributes were computed for the 53 regions identified in the ROI determination phase. Each object example was thus represented by a vector of real-valued attributes, except for the `Area` attribute, which is integer-valued .

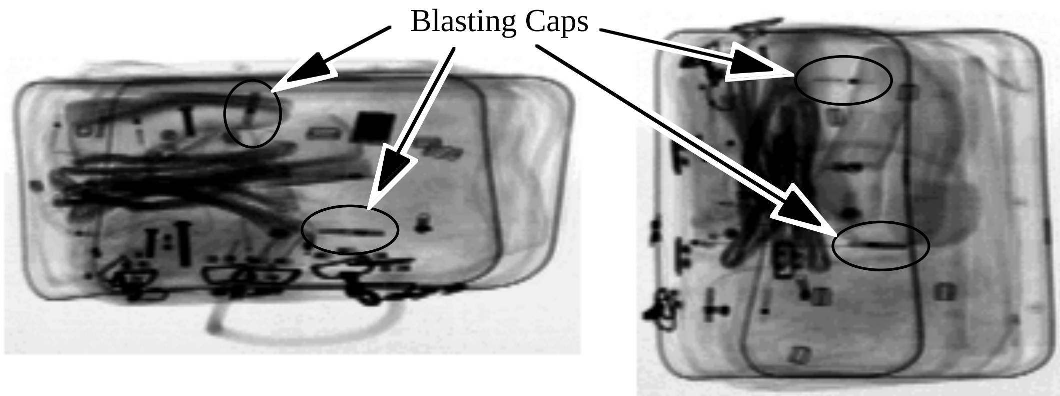
Discretization

Since AQ15c [18] operates on discrete attributes, the real-valued attributes, such as `Length` and `Major` (Table 1), require discretization. To further optimize the representation space, integer-valued attributes, like `Area`, can be projected into a smaller range. Such a process represents an abstraction operation on the representation space (as defined in the Inferential Theory of Learning [13]).

Several techniques exist for discretizing real-valued attributes including equal-width-intervals and equal-frequency-intervals [7]. With equal-width-intervals, the real range is divided into *n* equal-sized intervals and real values are mapped into the first *n* integers. A problem with this approach is that if the classification algorithm needs to discriminate between two real values and these values are mapped into the same range, any basis for discrimination is destroyed. In other words, the scaling procedure excessively abstracts the data.

Equal-frequency-intervals involves discretizing based on the frequency distribution of attribute values over the real-valued range. A problem associated with this technique is that a small group of important outliers could be grouped with a larger cluster of attribute values. Conversely, an important grouping of

Figure 3: Sample x-ray images of luggage containing blasting caps.



attribute values could be divided and mapped into different intervals because of outliers. In both instances, the scaling procedure excessively abstracts the representation space.

The ChiMerge discretization algorithm [7] uses the χ^2 statistic to merge real and integer attribute values into statistically relevant intervals. In other words, the algorithm groups and separates attribute values into intervals based on a statistical measure of the correlation between attribute values and their associated class labels.

AQ15c, using the SCALE implementation [3] of the ChiMerge algorithm, discretized all attribute values into at least 5 intervals using a 99% significance level. ChiMerge algorithm has the freedom to construct any number of intervals; however, one of its parameters is a lower bound on number of intervals. The significance level determines how parsimonious ChiMerge behaves when grouping real-valued attributes. For higher significance levels (e.g., 99%), ChiMerge tends to construct a small number of large intervals [7].

Table 2 illustrates the discretization of the `Major` attribute by the ChiMerge method. Notice that ChiMerge partitioned attribute ranges into differing widths for the attribute (e.g., scaled intervals 1 and 2). ChiMerge constructed discretization ranges, similar to those in Table 2, for each of the 5 attributes.

Table 2: Discretization of the attribute `Major`.

Value Range of Attribute "Major"	Abstracted Attribute Value
8.03...13.50	0
13.51...14.72	1
14.73...22.49	2
22.50...23.03	3
23.04...26.94	4
26.95...38.29	5
38.30...58.35	6

Learning

ROI determination, event extraction, and discretization produced 53 training examples divided between two distinct classes: caps and noncaps. Each training example consisted of 5 linear multivalued attributes, ranging between 6 and 13 value levels, a result of the ChiMerge discretization.

Preliminary experiments were conducted to establish learning parameters. The best performance resulted when AQ15c was set to generate *characteristic* rules. After learning parameters were determined, they were held constant and additional experiments consisted of 500 learning and recognition runs using a 2-fold cross validation methodology [17]. For each run, the complete set of pre-classified training examples was divided randomly and evenly into training and testing sets. These sets were given to AQ15c, which learned a set of decision rules from the training examples. Figure 4 presents one of the hypotheses learned by AQ15c.

Figure 4: Example of a hypothesis induced by AQ15c.

```
Cap <:: [Area=1 ∨ 3 ∨ 5 ∨ 7] &
        [Length=9..13] & [Major=5..6] &
        [Minor=0 ∨ 2 ∨ 4..6] &
        [Compact=2..4]
```
(t-weight:10, u-weight:10)

```
Cap <:: [Area=1] & [Length=5 ∨ 7] &
        [Major=5] & [Minor=0..1 ∨ 4] &
        [Compact=2..4]
```
(t-weight:3, u-weight:3)

The hypothesis in Figure 4 consists of two rules. Each rules is annotated by two numbers: the t-weight and the u-weight. The t-weight indicates the total number of the positive training examples the rule covers. The u-weight indicates how many of those examples are covered only by this rule (different rules can potentially overlap, therefore some examples may

Table 5: Performance summary for classification technique.

Learning Method	Average Recognition Accuracy	Best Single Performance	Average Learning Time	Average Recognition Time
AQ15c	95%	100%	0.1s	0.02s
ANN	79%	95%	7.5s	0.003s
k-nn (*k* = 1)	69%	88%	0.07s	0.02s

be covered by more than one rule). These weights indicate the strength of rules.

Recognition

For each run, following the learning step, the induced decision rules were used to classify the examples in the testing set. This produced a classification rate for the run. Four statistics were computed for each 500 run experiment: average recognition rate, best single performance, and average learning and recognition times. The average recognition rate for an experiment is the average of the recognition rates for each of the 500 runs. The best single performance is the highest classification rate achieved by the learner on any single run of an experiment. The average learning time is the average time spent learning a concept description from training examples for each of the 500 runs. Finally, the average recognition time is the average time spent testing the concept on testing examples for each of the 500 runs. This testing and validation methodology was also used for experiments involving other learning methods. When testing examples using AQ15c, the flexible matching scheme described in section 2.2 was used.

EXPERIMENTAL RESULTS

Three experiments were conducted using AQ15c, *k*-nn, and an artificial feed-forward neural network, using the testing and validation method described above. These learning methods were compared using average classification accuracy, best single performance, and average learning and recognition times.

k-nn works by taking a testing example and finding its *k* closest neighbors in the representation space using some distance measure. Typically *k* is odd to prevent ties. The class with the most closest neighbors is assigned as the decision class for the testing example. *k*-nn is an example of a memory-based learning technique. Preliminary experiments were conducted using 2-fold cross-validation, with *k* = 1, 3, 5, 7, 9, 11, 13 and 15 using an Euclidean distance measure. *k* = 1 produced the best results. A subsequent experiment consisting of 500 runs using 2-fold cross validation resulted in an average recognition accuracy of 69% and a best single performance of 88%.

The second experiment was with the Quickprop implementation [6] of a feed-forward neural network [20]. An artificial neural network (ANN) is a non-symbolic learning model inspired by the neuronal architecture of the human brain. The class of multilayer feed-forward networks is capable of learning non-linear statistical regularities from pre-classified examples. These models are considered non-symbolic since learned concepts are represented as real-valued weights distributed throughout the network's connections.

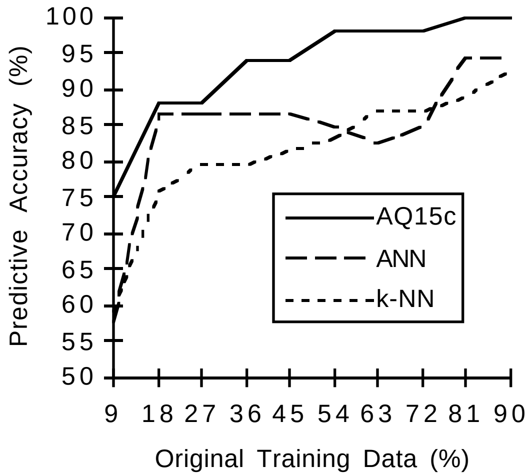
The neural network architecture chosen was a 1 hidden layer network with 5 input, 4 hidden, and 2 output units. Training and testing data were uniformly mapped into a continuous real range of [0, 1]. Output patterns were encoded as a linear representation of each of the two possible classes of objects. Again, preliminary experiments were conducted to determine the network's architecture and learning parameters. Subsequently, 500 learning and recognition runs were conducted using 2-fold cross-validation. The average recognition accuracy was 79%, while the best single performance was 95%.

The final experiment involved AQ15c, described earlier. After preliminary experiments helped determine learning parameters, 500 runs were conducted using 2-fold cross-validation. The average recognition accuracy was 95%, while the best single performance was 100%. Figure 4 shows an induced hypothesis for the cap class that achieved 100% classification accuracy on testing data. Table 5 summarizes the performance of the three learning methods. Figure 5 shows the learning curves for the three methods. This graph represents how the classification accuracy increases with respect to increasing amounts of training data.

DISCUSSION

In our experiments, AQ15c, a symbolic rule learning method used in this study, achieved significantly better average predictive accuracy on testing data than both *k*-nn and a neural network (95% vs. 69% and 79%). It learned somewhat more slowly than *k*-nn (0.1s vs. 0.03s), but an order of magnitude faster than neural network (0.1s vs. 7.5s). The recognition times of new examples were comparable in for AQ15c and *k*-nn, but significantly slower than for

Figure 5: Learning curves for the three classification methods.



neural net (20ms vs. 3ms). One may observe, however, that AQ15c decision rules can be executed in parallel, for example, by executing them on a neural network, in which case the recognition rates would be equally fast [2].

As mentioned earlier, one of the most important trademarks of the AQ15c learning system is high comprehensibility of knowledge (decision rules) it generates. VL_1 decision rules are easy to understand and interpret by a human expert. They can be easily translated into English. Neural networks lack this feature, because its knowledge resides in real-valued weights distributed throughout the network's connections, and thus carry little meaning for an expert. High understandability of the VL_1 rules makes it possible for human experts to modify and improve them. For example, human experts drawing on their domain knowledge might modify the ranges of rule conditions determined by the ChiMerge discretization step or remove some spurious condition. Since experts can understand the rules, they can also estimate the consequences of changes made to the rules.

It is precisely this kind of understanding and a possibility of human control that is needed for some applications, especially those in which a system is supposed to assist humans in making decisions affecting other humans.

CONCLUSIONS

The work presented in this paper is a natural progression and extension of our previous research on applying symbolic machine learning to vision problems [10, 11, 4, 15, 1, 2, 8]**.** Among the main advantages of the method proposed here are relatively high learning speed, high prediction accuracy and high understandability of the decision rules.

Machine learning can be applied potentially at differing levels of object representation (i.e., the pixel level, the feature level, and so on), and can be used in conjunction with a variety of vision processes, such as model formulation, pose estimation, and segmentation. This paper has demonstrated one such application, specifically, to acquiring symbolic descriptions of 2D shapes for object recognition.

The primary weakness of the current implementation is the need for human involvement in ROI determination. Consequently, this is one of the tasks of future research. Another weakness (shared also by many other learning programs) is the assumption that the given representation space (as defined by the chosen attributes and their domains) is sufficiently relevant to the problem at hand. This assumption can be weakened by the application of a constructive induction program, such as AQ17 [3], able to automatically improve the knowledge representation space. In future work, we intend to investigate the applicability of constructive induction to this and related problems. We also plan to apply the proposed method to other computer vision problems, such as gesture recognition, medical image analysis, and satellite image interpretation.

ACKNOWLEDGMENTS

The authors wish to thank Eric Bloedorn for reading preliminary drafts of this paper. The authors greatly appreciate the support of a Doctoral Fellowship from the School of Information Technology and Engineering at George Mason University. This research was conducted in the Machine Learning and Inference Laboratory at George Mason University. The Laboratory's research is supported in part by the Advanced Research Projects Agency under Grant No. N00014-91-J-1854, administered by the Office of Naval Research, and the Grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, in part by the Office of Naval Research under Grant No. N00014-91-J-1351, and in part by the National Science Foundation under Grants No. IRI-9020266 and DMI-9496192.

REFERENCES

[1] Bala, J. W. (1993) Learning to recognize visual concepts: development and implementation of a method for texture concept acquisition through inductive learning. Ph.D. Dissertation, George Mason University, Fairfax, VA.

[2] Bala, J. W.; Michalski, R. S.; and Pachowicz, P. W. (1994) Progress on vision through learning at George Mason University. *Proceedings of the 1994 Image Understanding Workshop*, 191–207.

[3] Bloedorn, E.; Wnek, J.; Michalski, R. S.; and Kaufman, K. (1993) AQ17 — A multistrategy learning system: the method and user's guide. *Reports of the Machine Learning and Inference Laboratory*, MLI 93–12. Machine Learning and Inference Laboratory, Department of Computer Science, George Mason University, Fairfax, VA.

[4] Channic, T. (1985) TEXPERT: An application of machine learning to texture recognition. Master's Thesis, University of Illinois, Urbana.

[5] Cromwell, R. L., and Kak, A. C. (1991) Automatic generation of object class descriptions using symbolic learning techniques. *AAAI-91*, 710–717.

[6] Fahlman, S. E. (1988) *An empirical study of learning speed in back-propagation networks*. Technical Report CMU-CS-88-182. Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

[7] Kerber, R. (1992) ChiMerge: discretization of numeric attributes. *AAAI-92*. 123–128.

[8] Maloof, M. A., and Michalski, R. S. (1994) Learning descriptions of 2D blob-like shapes for object recognition in x-ray images: an initial study. *Reports of the Machine Learning and Inference Laboratory*, MLI 94–4. Machine Learning and Inference Laboratory, Department of Computer Science, George Mason University, Fairfax, VA.

[9] Michalski, R. S. (1969) On the quasi-minimal solution of the general covering problem. *Fifth International Symposium on Information Processing*, A3: 125–128.

[10] Michalski, R. S. (1972) A variable-valued logic system as applied to picture description and recognition. *IFIP Working Conference on Graphic Languages*, 21–47.

[11] Michalski, R. S. (1973) AQVAL/1 — computer implementation of a variable-valued logic system VL_1 and examples of its application to pattern recognition. *First International Joint Conference on Pattern Recognition*, 3–17.

[12] Michalski, R. S. (1980) Pattern recognition as rule-guided inductive inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2.4:349–361.

[13] Michalski, R. S. (1994) Inferential Theory of Learning: developing foundations for multistrategy learning. In Michalski, R. S., and Tecuci, G., eds., *Machine Learning: A Multistrategy Approach*, Vol. IV, 3–61. San Francisco, CA: Morgan Kaufmann.

[14] Michalski, R. S.; Rosenfeld, A.; and Aloimonos, Y. (1994) Machine vision and learning: research issues and directions. *Reports of the Machine Learning and Inference Laboratory*, MLI 94–6. Machine Learning and Inference Laboratory, Department of Computer Science, George Mason University, Fairfax, VA.

[15] Pachowicz, P. W., and Bala, J. W. (1991) Texture recognition through machine learning and concept optimization. *Reports of the Machine Learning and Inference Laboratory*, MLI 95–4. Machine Learning and Inference Laboratory, Department of Computer Science, George Mason University, Fairfax, VA.

[16] Shepherd, B. A. (1983) An appraisal of a decision tree approach to image classification. *IJCAI-83*, 473–475.

[17] Weiss, S. M., and Kulikowski, C. A. (1992) *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning and expert systems*. San Mateo, CA: Morgan Kaufmann.

[18] Wnek, J.; Kaufman, K.; Bloedorn, E.; and Michalski, R. S. (1995) Selective induction learning system AQ15c: the method and user's guide. *Reports of the Machine Learning and Inference Laboratory*, MLI 95–4. Machine Learning and Inference Laboratory, Department of Computer Science, George Mason University, Fairfax, VA.

[19] Zhang, J., and Michalski, R. S. (1989) Rule optimization via the SG-TRUNC method. *Proceedings of the Fourth European Working Session on Learning*.

[20] Zurada, J. M. (1992) *Introduction to artificial neural systems*. St. Paul, MN: West Publishing.