In Press: *Expert Systems with Applications*
Special issue on Intelligent Systems Applied in Industry and Business

LEARNING SYMBOLIC DESCRIPTIONS OF SHAPE FOR OBJECT RECOGNITION IN X-RAY IMAGES

MARCUS A. MALOOF AND RYSZARD S. MICHALSKI

Machine Learning and Inference Laboratory, George Mason University, Fairfax, VA 22030

Abstract—*In this paper, we describe a method for learning shape descriptions of 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, the C4.5 decision tree learning program, and a multilayer feed-forward neural network. 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. AQ's representation language, VL_1, was better suited for this problem, which can be seen by examining the empirical results and the learned rules. The method was applied to the problem of 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.*

1. INTRODUCTION

Despite many efforts, the problem of shape learning and recognition remains open. The goal of our research is to develop a methodology for shape learning and recognition, and apply it to a variety of practical problems. In this paper, we are concerned with the recognition 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 predefined representation space. The training examples are passed to a learning system that produces general descriptions of the visual concepts.

The learned concept descriptions are then validated using testing examples of blasting caps and non-blasting caps, prepared a priori 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 solve an important practical problem and make a contribution to the methodology of vision through learning. An intelligent system capable of identifying blasting caps in x-ray images could assist airport security personnel with luggage screening.

The described method uses the AQ15c inductive rule learning system (Wnek, Kaufman, Bloedorn, & Michalski, 1995) for generating blasting cap descriptions from training examples. To evaluate the method, the descriptions obtained by the AQ15c learning system and other systems were compared in terms of predictive accuracy and the speed of learning and recognition. Experimental results demonstrate strong advantages of the method over *k*-nearest neighbor (Weiss & Kulikowski, 1992), a statistical pattern recognition technique, C4.5 (Quinlan, 1993), a decision tree learning method, and a multilayer feed-forward neural network (Zurada, 1992). AQ's representation language was better suited for this problem than the representation languages of the other methods. This can be seen by examining the empirical results and the learned concept descriptions. Other experiments using this methodology for the problem of blasting cap detection have been reported by Maloof and Michalski (1994), Bala, Michalski, and Pachowicz (1994), Maloof and Michalski (1995), and Maloof, Duric, Michalski, and Rosenfeld (1996).

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 method, while section 4 details the obtained results. Section 5 discusses the contribution and significance of the results and outlines plans for future research.

2. 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 (Michalski, Rosenfeld, and Aloimonos, 1994). 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, this area will require the development of new learning methodologies capable of dealing with the complexities of visual perception.

Potentially, machine learning methods can be used for a variety of computer vision problems. This paper concentrates on the application of machine learning techniques to one specific computer vision problem: the acquisition and use of shape descriptions for object recognition.

2.1. The AQ Learning Approach

In the AQ learning approach that is used in this study, a visual concept description is in the form of VL_1 decision rules (Michalski, 1972). 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 (i.e., nominal, linear, or structured) with an indication of the decision class (visual concept) to which it belongs.

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

In this study, we use the inductive learning system AQ15c, in which concept examples, domain knowledge, and concept descriptions are expressed using an attributional representation language, called Variable-Valued Logic 1, or VL_1 (Michalski, 1972). To make this paper self-contained, we begin by briefly characterizing the description language. VL_1 decision rules, used to represent concept examples, domain knowledge and concept descriptions, are of the form:

$$D_i <:: C_j$$

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_j is the *condition* part of the rule 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 of 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 $\vee$ 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 the 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 above a threshold of acceptability and higher than the degree of match between the object and other candidate rules.

In the default parameter setting, AQ15c creates maximally general hypotheses (a ruleset) that describe all positive 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 the 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 (Michalski, 1969). Since this problem is NP-hard, the AQ algorithm (that underlies 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 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 uncovered positive examples and the process repeats.

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

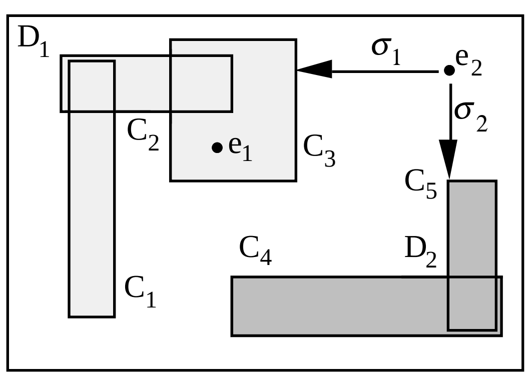

FIGURE 1. An illustration of decision regions, complexes, and examples.

2.2. 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 associated with the rule is assigned.

Conceptually, rules carve out decision regions in a representation (event) space, defined by the attributes and their domains chosen to characterize objects. Figure 1 illustrates such decision regions. The decision region D_1 consists of three complexes: C_1, C_2, and C_3 (illustrated by overlapping rectangles), while the decision region D_2 consists of two complexes: C_4 and C_5.

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 using a *strict matching* mode, no decision would be assigned. Using a *flexible matching* mode, a degree of match between the example and all decision regions is computed, and the best match above a threshold of acceptability indicates the decision class. Flexible matching thus helps alleviate the brittleness associated with rule-based reasoning systems.

Several flexible matching schemes exist to calculate the degree of match between examples and concept descriptions. The method employed here works as follows (Wnek et al., 1995). For a decision region D_i consisting of n complexes C_j, the degree of match σ_i for the example e_2 is given by:

$$\sigma_i = \begin{cases} \dfrac{\alpha_{ij}}{\beta_{ij}}, \text{ for } j = 1 \\ \sigma_{i-1} + \dfrac{\alpha_{ij}}{\beta_{ij}} - \sigma_{i-1}\dfrac{\alpha_{ij}}{\beta_{ij}}, \text{ for } j = 2..n \end{cases} \tag{1}$$

where

α_{ij} is the number of conditions in complex C_j satisfied by example e_2, and

β_{ij} is the total number of conditions in complex C_j.

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

In summary, the advantages of the AQ15c inductive learning system over the other methods include:

- can learn comprehensible concept descriptions,
- uses crisp or flexible matching techniques for recognition,
- learns different types of rules (discriminant or characteristic),
- learns different types of covers (disjoint, intersecting, or ordered),
- allows problem-dependent utility measures for attributes,
- allows problem-dependent measures of description quality, and
- guarantees completeness and consistency.

2.3. Relevant Past Work

There has been a number of papers on the topic of learning descriptions of shape using machine learning techniques. For example, Shepherd (1983) 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 producing the highest accuracy of 82%.

Connell and Brady (1987) describe a shape learning component using a semantic network representation. The vision system builds semantic networks from gray-scale images which are generalized using a modified version of Winston's (1984) ANALOGY program. Generalized representations are subsequently used for object recognition. The system was used to learn shapes of commercial aircraft and of three types of hammers. Experimental results were not presented.

Cromwell and Kak (1991) 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 covers the most examples from the training set. Their induction methodology was based on Michalski's (1980). The average classification accuracy for their system was 72%. No comparisons were made to other learning systems.

Segen (1994) uses a multistrategy learning approach for real-time recognition of hand gestures. Shape is represented by a hierarchical graph representation that captures local curvature, distance between features and angles between features. The representation is invariant to planar rotation and translation. These structural features are induced using an unsupervised constructive induction technique. Classified structural features are then generalized using a supervised learning algorithm into a graph model that consists of probabilities for structural features of a shape class. Reported errors rates were between 5% and 10%. Most errors were reported as unknown gestures rather than misclassifications.

Cho and Dunn (1994) introduce a property-based learning algorithm for learning shape in which concepts are represented as lists, or conjunctions, of local properties. Shape is approximated by a series of line segments. Local spatial properties are computed between pairs of line segments and are invariant to planar translation, rotation and scale. Reinforcement learning updates weights associated with property lists, while forgetting mechanisms remove concepts of low weight. Experimental results are presented for learning tool shapes and hand gestures, for which the method achieved 92% and 96% predictive accuracy. No comparisons were made to other learning methods.

The foundations for applying AQ learning to recognition problems in vision were laid by Michalski (1972, 1973). In those seminal papers, the Multilevel Logical Template (MLT) methodology used the AQ algorithm 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 (1985), 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 (1991) used the MLT methodology with a modified set of Laws' masks for texture feature extraction and applied various description optimization techniques (e.g., the SG-TRUNC method (Zhang & Michalski, 1989)) for alleviating problems encountered by the introduction of noise. Bala (1993) also introduced and applied to texture recognition methods for concept optimization (AQ-GA), and learning a large number of classes (PRAX).

The initial MLT methodology has been extended recently into the Multilevel Image Sampling and Transformation (MIST) methodology. One of its applications is to the problem of segmenting a natural scene into classes, such as sky, trees, grass, and the like (Michalski, Zhang, Maloof, & Bloedorn, 1996). In this study, three learning techniques were compared: AQ15c (Wnek et al., 1995), a backpropagation neural network (Zurada, 1992), and AQ-NN (Bala et al., 1994). AQ-NN is a multistrategy technique in which the AQ15c inductive learning system is used to structure a neural network architecture using decision rules. A secondary learning phase is used to adjust the network's weights, which results in higher predictive accuracy than the learned rules alone and recognition rates often faster than a neural network. Features, such as hue, intensity, vertical and horizontal line operators, were extracted from a 10x10 pixel window moved over user-designated training areas. AQ15c achieved a recognition rate of 94%, while both the neural network and AQ-NN achieved 100% on unseen testing data.

Maloof and Michalski (1994) used the MLT methodology for learning shape descriptions for discriminating among x-rays of blasting caps from an image database. 14 intensity and shape features were used to represent three characteristic blobs of x-rayed blasting caps. Experimental comparisons were made between AQ15c, a neural network, and *k*-nn, which achieved predictive accuracy rates of 94%, 96%, and 69%, respectively.

This initial study was followed by experiments using x-rayed luggage containing blasting caps and various forms of clutter, such as shoes, bolts, pens, and calculators. Five shape features invariant to planar rotation and translation were used to represent blasting cap and non-blasting cap objects. Comparisons were again made between AQ15c, a neural network, and *k*-nn, which achieved predictive accuracy rates of 95%, 79%, and 69%, respectively, as reported in (Maloof & Michalski, 1995).

The work presented here is an extension of our previous work in several ways. We consider an additional learning method, C4.5 (Quinlan, 1993), which is a symbolic inductive learning system, like AQ15c, but learns a different concept representation, namely decision trees. The inclusion of this method provides a broad comparison of learning techniques. We also provide a detailed analysis of why AQ15c performed better than other methods on this problem.

Additional work is reported in (Maloof et al., 1996) in which inductive learning is used to acquire object functionality. Recognition proceeds in a top-down and bottom-up manner. In a bottom-up step, low intensity blobs are used as attention-catching devices and posit object hypotheses. A top-down step attempts to match learned local models to image characteristics surrounding the blobs. Positive and negative examples were represented using 25 intensity, shape, and proximity features. AQ15c achieved recognition rates of 84%. No comparisions were made to other learning methods.

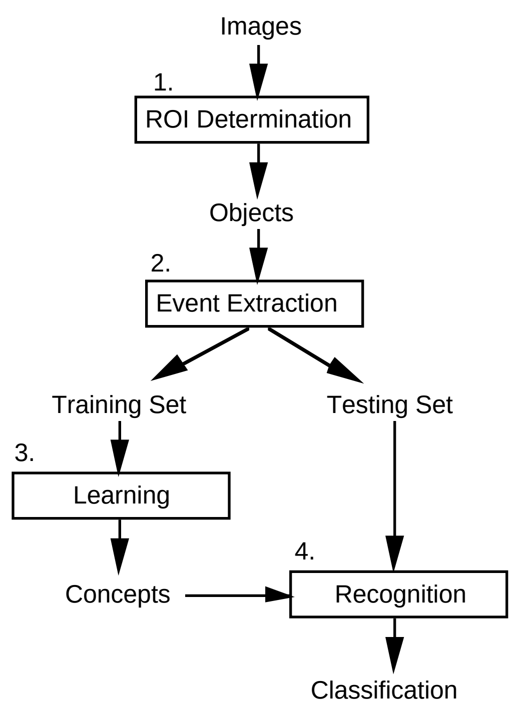

FIGURE 2. Basic steps of the shape learning and recognition methodology.

3. METHODOLOGY

The symbolic learning methodology used here closely parallels Michalski's (1973), Channic's (1985) and Bala's (1993) proceeding through a four step process: (1) Region of Interest (ROI) Determination, (2) Event Extraction, (3) Learning, and (4) Recognition (see Figure 2). These steps are described in the following subsections.

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

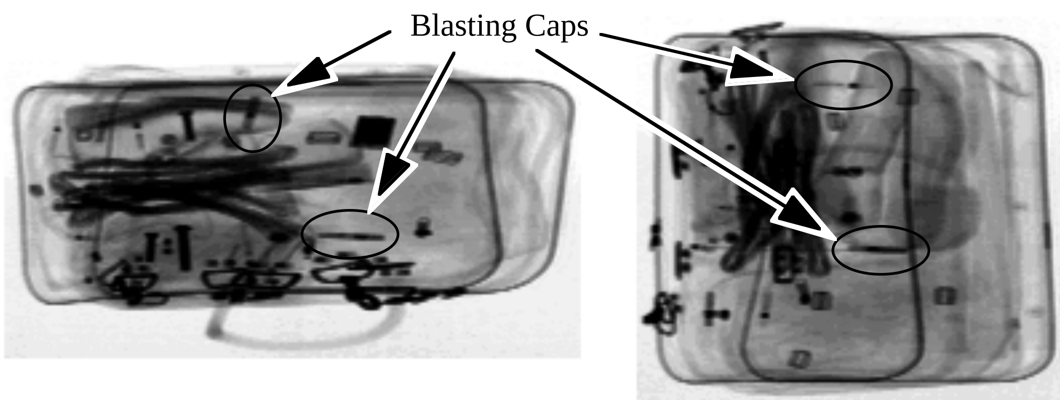

FIGURE 3. Sample x-ray images of luggage containing blasting caps.

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 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.

3.2. Event Extraction

After ROI determination, image objects are described in terms of the attribute values 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 was integer-valued .

3.3. Learning

The AQ15c symbolic inductive learning system (Wnek et al., 1995) was used to learn descriptions of shape from positive and negative training examples. Since AQ15c operates on discrete attributes, the real-valued attributes, such as `Length` and `Major` (Table 1), required 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 by the Inferential Theory of Learning (Michalski, 1994).

Several techniques exist for discretizing real-valued attributes including equal-width-intervals and equal-frequency-intervals (Kerber, 1992). 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 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 (Kerber, 1992) 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 (Bloedorn, Wnek, Michalski, & Kaufman, 1993) of the ChiMerge algorithm (Kerber, 1992), discretized all attribute values into at least 5 intervals using a 99% significance level. The 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 (Kerber, 1992).

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

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

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. All attributes achieved 1.0 using the entropy (Quinlan, 1986) and PROMISE (Baim, 1988) measures for attribute relevancy.

Preliminary experiments were conducted to establish learning parameters. The best performance resulted when AQ15c was set to generate *characteristic* rules, which are rules that consist of all known attributes and attribute values for a class of objects (Michalski, 1983). After learning parameters were determined, they were held constant and additional experiments consisted of 50 learning and recognition runs using a 2-fold cross-validation methodology (Weiss & Kulikowski, 1992). For each run, the complete set of preclassified 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.

The hypothesis in Figure 4 consists of two rules. Each rule 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 maybe covered by more than one rule). These weights indicate the strength of rules.

```
Cap <:: [Area=1 v 3 v 5 v 7] &
        [Length=9..11 v 13] &
        [Major=5, 6] &
        [Minor=0 v 2 v 5 v 6 v 8 v 10] &
        [Compact=1..4]
        (t-weight:8, u-weight:8)

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

FIGURE 4. Example of a hypothesis induced by AQ15c.

3.4. 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 or recognition rate for the run. Four statistics were computed for each 50 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 50 runs. A 95% confidence interval was also computed for the learning method. 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 50 runs. Finally, the average recognition time is the average time spent testing the concept on testing examples for each of the 50 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.

4. EXPERIMENTAL RESULTS

Four experiments were conducted using AQ15c, *k*-nn, C4.5, and a multilayer 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. Table 3 summarizes the experimental results. The predictive accuracy results are statistically significant at $p < .01$. Figure 5 shows the learning curves for the four methods. This graph indicates how the classification accuracy increases with respect to increasing amounts of training data.

TABLE 3
Performance summary for classification techniques

Learning Method	Average Recognition Accuracy*	Best Single Performance	Average Learning Time	Average Recognition Time
AQ15c	95.48±1.52%	100%	0.1 s	30 ms
C4.5	80.61±2.62%	96%	0.1 s	20 ms
ANN	75.2±2.46%	96%	37.5 s	4 ms
k-nn (k = 1)	70.88±1.84%	81%	0.05 s	20 ms

*$p < .01$

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 instance-based learning technique (Aha, Kibler, & Albert, 1991). 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 50 runs using 2-fold cross-validation resulted in an average recognition accuracy of 71% and a best single performance of 81%.

The second experiment was with the Quickprop implementation (Fahlman, 1998) of a feed-forward neural network (Zurada, 1992). An artificial neural network (ANN) is a nonsymbolic learning model inspired by the neuronal architecture of the human brain. The class of multilayer feed-forward networks is capable of learning nonlinear statistical regularities from preclassified examples. These models are considered nonsymbolic 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

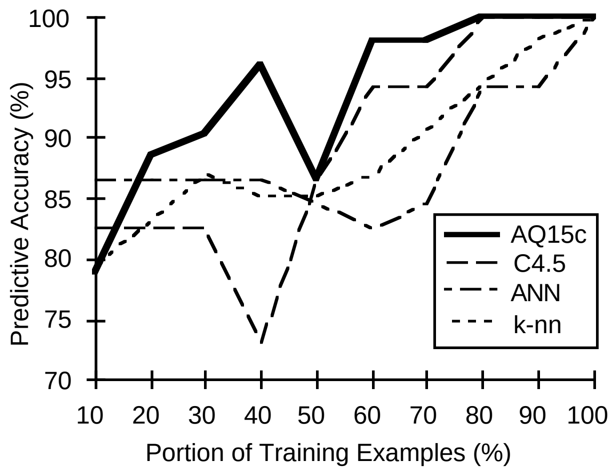

FIGURE 5. Learning curves for the four classification methods.

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, 50 learning and recognition runs were conducted using 2-fold cross-validation. The average recognition accuracy was 75%, while the best single performance was 96%.

In the third experiment, C4.5 (Quinlan, 1993) was used to induce decision trees. A decision tree is an *n*-ary tree having attributes as interior nodes and attribute values as branches. Leaf nodes in the tree are decision classes. Recognition involves starting at the root node and testing the attribute of an observation. The value of the observation's attribute dictates which branch of the decision tree is traversed. This proceeds recursively until a leaf node is reached. The class label of that leaf node is returned as the classification of the observation. Decision trees are induced by recursively selecting an attribute that best partitions training data by class using a disjointness criteria, such the entropy measure. Preliminary experiments were used to set C4.5's windowing and pruning parameters. 50 learning and recognition runs were conducted using 2-fold cross-validation. Average recognition accuracy for C4.5 was 81%, while its best single performance was 96%.

The final experiment involved AQ15c, described earlier. After preliminary experiments helped determine learning parameters, 50 runs were conducted using 2-fold cross-validation. The average recognition accuracy was 96%, 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 four learning methods. Average recognition accuracy is shown with a 95% confidence interval.

5. DISCUSSION

In our experiments, AQ15c, a symbolic rule learning method, achieved significantly better average predictive accuracy on testing data than *k*-nn, C4.5, and a neural network (96% vs. 81%, 75%, and 71%). It learned somewhat more slowly than *k*-nn (0.1 s vs. 0.05 s), but orders of magnitude faster than the neural network (0.1 s vs. 37.5 s). Learning times were comparable for AQ15c and C4.5. The recognition times of unseen examples were comparable for AQ15c, C4.5, and *k*-nn, but were significantly slower than the neural net (20 ms vs. 4 ms). One may observe, however, that AQ15c decision rules can be used as the basis for a neural network architecture in which the recognition rates are as fast or faster (Bala et al., 1994; Michalski et al., 1996).

A reason why AQ15c out-performed the other methods is that the VL_1 representation language was able to better represent the distribution of the training examples in the representation space. Refering to the first rule in Figure 4, we get an idea of the disjointness of the representation space by looking at the selectors for the `Area` and `Minor` attributes. The `Area` selector consists of odd values, while the `Minor` selector consists primarily of even values, creating a checkerboard-like pattern covering the representation space. The disjointness of the space expressed by the decision rules indicates the need for constructive induction (Michalski, 1980) to map the current representation space into one that is better suited for learning.

Because rules carve out decision regions in the representation space and leave portions of the space uncovered, AQ tends to generalize the training examples properly for this problem. During recognition, if an unknown example is from a part of the representation space uncovered by a rule, flexible matching routines (described in Section 2.2) compute the most likely class.

Decision tree learning, on the other hand, completely partitions the representation space. For this particular problem and problems like it in which concepts are checkerboard patterns, like the MONK2 problem (Thrun et al., 1991), decision tree learning tends to overspecialize its concept descriptions. While both C4.5 and AQ15c learn disjunctive normal form (DNF) concepts, because VL_1 expressions can potentially overlap, a richer set of DNF concepts can be represented than with a decision tree representation which cannot produce overlapping covers.

k-nn's poor performance on this problem is also likely due to the distribution of examples in the space. Blasting cap and non-blasting cap examples were sparsely distributed and intermixed throughout the representation space. There were no groupings of training examples into classes. Consequently, from any unknown point in the representation space, the distance metric could have found an example from either of the classes as its closest neighbor. This disjointness of the space also would have nullified any affect larger values of k would have had. Experimental results bear this out, since $k = 1$ produced the best classification accuracy.

Finally, the neural network may have also been affected by the distribution of the examples in the space. It is more difficult to precisely understand what the neural network learns since its concepts are nonsymbolic. Ultimately, the precise nature of the decision regions learned by a neural network is unknown (Michie, Spiegelhalter, & Taylor, 1994). A second possible explanation for the neural network's poor performance is that we may not have found the optimal set of network parameters, although considerable time was spent configuring and optimizing the network for these experiments. Unlike the other learning methods, the parameter space for neural networks is quite large and how to effectively search this space for the optimal set of parameters is itself a challenging research issue. Indeed, for sufficiently complex problems, this space can only be searched heuristically. Finally, because training a neural network is accomplished by a gradient descent algorithm, there is no termination point for training. This also is in contrast to the other methods investigated. The person training the network must make the decision when to stop training. If learning is stopped too soon, the intended concepts may not have been learned sufficiently. If learning is stopped too late, the network will overlearn the concepts.

As to why the representation space appears to be so disjoint could related to the ROI determination procedure. Human bias in this step can greatly affect learning results. For this problem, if the expert selects a representative set of blasting caps and then selects negative examples that are either much smaller or larger than most blasting caps, then the representation space would likely consist of two or three clusters of training examples isolated from each other. On the other hand, if the expert selects negative examples that are similar in shape to blasting caps, then the representation space would likely be more disjoint, especially after discretizing the space. By virtue of various discretization algorithms, especially those that correlate class label to attribute values when forming intervals, like ChiMerge (Kerber, 1992), two close points with different class labels in an *n*-dimensional continuous representation space will be close to each other in a discrete version of the space. So if an expert selects negative examples in this manner, it would result in a more disjoint, and more difficult, representation space for learning than if the examples were selected in the former manner.

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.

6. CONCLUSIONS

The work presented in this paper is an extension of our previous research on learning shape descriptions (Michalski, 1972, 1973; Bala et al., 1994; Maloof & Michalski, 1994, 1995). Further extensions to this work are reported in (Maloof et al., 1996). Among the main advantages of the method 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, of acquiring symbolic descriptions of shape for object recognition.

The primary weakness of the current implementation is the need for human involvement in ROI determination and the strictly bottom-up recognition strategy. Consequently, this is one of the tasks of future research and has been explored in (Maloof et al., 1996). We plan to investigate how incremental learning and user feedback can be used during ROI determination to produce a system that interacts with the expert and improves over time. 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 for the problem at hand. This assumption can be weakened by the application of a constructive induction program, such as AQ17 (Bloedorn et al., 1993), which is able to automatically improve the 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 method to other computer vision problems, such as gesture recognition, medical image analysis, and satellite image interpretation.

Acknowledgment—The authors wish to thank Eric Bloedorn, Ken Kaufman, and Ibrahim Imam for insights into decision tree learning.

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

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, **6**, 37–66.

Baim, P. W. (1988). A method for attribute selection in inductive learning systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(6), 888–896.

Bala, J. W. (1993). *Learning to recognize visual concepts: Development and implementation of a method for texture concept acquisition through inductive learning*. Doctoral dissertation, George Mason University, Fairfax, VA.

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

Bloedorn, E., Wnek, J., Michalski, R. S., & Kaufman, K. (1993). *AQ17 — A multistrategy learning system: The method and user's guide* (Tech. Rep. No. MLI 93–12). Fairfax, VA: George Mason University, Machine Learning and Inference Laboratory.

Channic, T. (1985). *TEXPERT: An application of machine learning to texture recognition*. Master's thesis, University of Illinois, Urbana.

Cho, K., & Dunn, S. M. (1994). Learning shape classes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(9), 882–888.

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

Connell, J. H., & Brady, M. (1987). Generating and generalizing models of visual objects. *Artificial Intelligence*, **34**(2), 159–183.

Fahlman, S. E. (1988). *An empirical study of learning speed in back-propagation networks* (Tech. Rep. No. CMU-CS-88-182). Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Kerber, R. (1992). ChiMerge: Discretization of numeric attributes. In *Proceedings of AAAI-92*, 123–128.

Maloof, M. A., & Michalski, R. S. (1994). *Learning descriptions of 2D blob-like shapes for object recognition in x-ray images: An initial study* (Tech. Rep. No. MLI 94–4). Fairfax, VA: George Mason University, Machine Learning and Inference Laboratory.

Maloof, M. A., & Michalski, R. S. (1995). Learning descriptions of 2D shapes for object recognition in x-ray images. In *Proceedings of the 8th International Symposium on Artificial Intelligence*, 124–131.

Maloof, M. A., Duric, Z., Michalski, R. S., & Rosenfeld, A. (1996). Recognizing blasting caps in x-ray images. In *Proceedings of the 1996 Image Understanding Workshop*, 1257–1261.

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

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

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

Michalski, R. S. (1980). Pattern recognition as rule-guided inductive inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**(4), 349–361.

Michalski, R. S. (1983). A theory and methodology of inductive learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: Vol. I. An artificial intelligence approach* (pp. 83–134). San Francisco, CA: Morgan Kaufmann.

Michalski, R. S. (1994). Inferential Theory of Learning: Developing foundations for multistrategy learning. In R. S. Michalski, & G. Tecuci (Eds.), *Machine learning: Vol. IV. A multistrategy approach* (pp. 3–61). San Francisco, CA: Morgan Kaufmann.

Michalski, R. S., Rosenfeld, A., & Aloimonos, Y. (1994). *Machine vision and learning: Research issues and directions* (Tech. Rep. No. MLI 94–6). Fairfax, VA: George Mason University, Machine Learning and Inference Laboratory.

Michalski, R. S., Zhang, Q., Maloof, M. A., & Bloedorn E. (1996). The MIST Methodology and its application to natural scene segmentation. In *Proceedings of the 1996 Image Understanding Workshop*, 1474–1479.

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine learning, neural, and statistical classification*. West Sussex: Ellis Horwood.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, **1**, 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA: Morgan Kaufmann.

Pachowicz, P. W., & Bala, J. W. (1991). *Texture recognition through machine learning and concept optimization* (Tech. Rep. No. MLI 91–4). Fairfax, VA: George Mason University, Machine Learning and Inference Laboratory.

Segen, J. (1994). GEST: A learning computer vision system that recognizes hand gestures. In R. S. Michalski, & G. Tecuci (Eds.), *Machine learning: Vol. IV. A multistrategy approach* (pp. 621–634). San Francisco, CA: Morgan Kaufmann.

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

Thrun, S. B., et al. (1991). *The MONK's problems: A performance comparison of different learning algorithms* (Tech. Rep. No. CMU-CS-91-197). Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Weiss, S. M., & Kulikowski, C. A. (1992). *Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning and expert systems*. San Francisco, CA: Morgan Kaufmann.

Winston, P. H. (1984). *Artificial intelligence* (2nd ed.). Reading, MA: Addison-Wesley.

Wnek, J., Kaufman, K., Bloedorn, E., & Michalski, R. S. (1995). *Selective induction learning system AQ15c: The method and user's guide* (Tech. Rep. No. MLI 95–4). Fairfax, VA: George Mason University, Machine Learning and Inference Laboratory.

Zhang, J., & Michalski, R. S. (1989). Rule optimization via the SG-TRUNC method. In *Proceedings of the Fourth European Working Session on Learning*, 251–262.

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