Proceedings of the 1997 Image Understanding Workshop, 835–845.
San Francisco, CA: Morgan Kaufmann.

Learning to Detect Rooftops in Aerial Images*

Marcus A. Maloof† **Pat Langley**† **Stephanie Sage**† **Thomas O. Binford**‡

†Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, CA 94306
{maloof, langley, sage}@isle.org • http://www.isle.org

‡Robotics Laboratory, Department of Computer Science
Stanford University, Stanford, CA 94305
binford@cs.stanford.edu • http://robotics.stanford.edu

Abstract

In this paper, we examine the use of machine learning to improve the robustness of systems for image analysis on the task of roof detection. We review the problem of analyzing aerial photographs, and describe an existing vision system that attempts to automate the identification of buildings in aerial images. After this, we briefly review several well-known learning algorithms that represent a wide variety of inductive biases. We report three experiments designed to illuminate facets of applying machine learning methods to the image analysis task; one experiment focuses on within-image learning, another deals with the cost of different errors, and a third addresses between-image learning. Experimental results demonstrate that machine-learned classifiers meet or exceed the accuracy of handcrafted solutions and that useful generalization occurs when training and testing on data derived from different images.

1 Introduction

The number of images available to image analysts is growing rapidly, and will soon outpace their ability to process them. Computational aids will be required to filter this flood of images and focus the analyst's attention on interesting events, but current image understanding systems are not yet robust enough to support this process. Successful image understanding relies on knowledge, and despite theoretical progress, implemented vision systems still rely on heuristic methods that remain fragile. Handcrafted knowledge about when and how to use particular vision operations can give acceptable results on some images but not others.

In this paper we explore the use of machine learning as a means for improving knowledge used in the vision process, and thus for producing more robust software. Recent applications of machine learning in business and industry [Langley and Simon, 1995] hold useful lessons to its application in image analysis. A key idea in applied machine learning involves building an *advisory* system that recommends actions but gives final control to a human user, with each decision generating a training case, gathered in an unobtrusive way, for use in learning. This setting for knowledge acquisition is similar to the scenario in which an image analyst interacts with a vision system, finding some system analyses acceptable and others uninteresting or in error. The aim of our research program is to embed machine learning into this interactive process of image analysis.

This adaptive approach to computer vision promises to greatly reduce the number of decisions that image analysts must make per picture, thus improving their ability to deal with a high flow of images. Moreover, the resulting systems should adapt their knowledge to the preferences of individual users in response to feedback from those users. The overall effect should be a new class of systems for image analysis that reduce the workload on human analysts and give them more reliable results, thus speeding the image analysis process.

In the sections that follow, we report initial progress on using machine learning to improve decision making at one stage in an existing image understanding

*This research was supported by the Defense Advanced Research Projects Agency under grant N00014-94-1-0746, administered by the Office of Naval Research.

system. We begin by explaining the task domain — identifying buildings in aerial photographs — and then describe the vision system designed for this task. Next we review five well-known algorithms for supervised learning that hold potential for improving the reliability of image analysis in this domain. After this, we report the design of experiments to evaluate these methods and the results of those studies. In closing, we consider related work on learning for image understanding and some directions for future research.

2 Nature of the Image Analysis Task

The image analyst interprets aerial images of ground sites with an eye to unusual activity or other interesting behavior. The images under scrutiny are usually complex, involving many objects arranged in a variety of patterns. A typical image from the Fort Hood RADIUS repository, which contains satellite photographs of a military base, includes buildings in a range of sizes and shapes, major and minor roadways, sidewalks, parking lots, vehicles, and vegetation. A common task faced by the image analyst is to detect change at a site as reflected in differences between two images, as in the number of buildings, roads, and vehicles. This in turn requires the ability to recognize examples from each class of interest. In this paper, we focus on the performance task of identifying buildings in satellite photographs.

Aerial images can vary across a number of dimensions. The most obvious factors concern viewing parameters, such as distance from the site (which affects size and resolution) and viewing angle (which affects perspective and visible surfaces). But other variables also influence the nature of the image, including the time of day (which affects contrast and shadows), the time of year (which affects foliage), and the site itself (which determines the shapes of viewed objects). Taken together, these factors introduce considerable variability into the images that confront the analyst.

In turn, this variability can significantly complicate the task of recognizing object classes. Although a building or vehicle will appear different from alternative perspectives and distances, the effects of such transformations are reasonably well understood. But variations due to time of day, the season, and the site are more serious. Shadows and foliage can hide edges and obscure surfaces, and buildings at distinct sites may have quite different structures and layouts. Such variations serve as mere distractions to the human image analyst, yet they provide serious challenges to existing computer vision systems.

This suggests a natural task for machine learning: given aerial images as training data, acquire knowledge that improves the reliability of such an image analysis system. However, we cannot study this task in the abstract. We must explore the effect of specific induction algorithms on particular vision software. In the next two sections, we briefly review one such system for image analysis, followed by five learning methods that might give it more robust behavior.

3 An Architecture for Image Analysis

Lin and Nevatia [1996] report a computer vision system for the analysis of ground sites in aerial images. Like many programs for image understanding, their system operates in a series of processing stages. Each step involves aggregating lower level features into higher level ones, eventually reaching hypotheses about the locations of buildings. We will consider these stages in the order they occur.

Starting at the pixel level, the system uses an edge detector to group pixels into edgels, and then invokes a line finder to group edgels into lines. Junctions and parallel lines are identified and combined to form three-sided structures or "Us". The algorithm then groups selected Us and junctions to form parallelograms. Each such parallelogram constitutes a hypothesis about the position and orientation of the roof for some building, so we may call this step 'rooftop generation'.

After the system has completed the above aggregation process, a 'rooftop selection' stage evaluates each hypothesis to determine whether that candidate has sufficient evidence to be retained. The aim of this process is to remove hypotheses that do not correspond to actual buildings. Ideally, the system will reject most spurious hypotheses at this point, although a final 'verification' step may still collapse duplicate or overlapping rooftops. This stage may also exclude hypotheses if there exists no evidence of three-dimensional structure, such as shadows and walls.

Analysis of the system's operation suggested that rooftop selection held the most promise for improvement through machine learning, because this stage must deal with many spurious hypotheses. This process takes into account both local and global criteria. Local support comes from features such as lines and corners that are close to a given parallelogram. Since these suggest walls and shadows, they provide evidence that the hypothesis corresponds to an actual building. Global criteria consider containment, overlap, and duplication of hypotheses. Using these evaluation criteria, the set of hypotheses is reduced to a more manageable size.

The individual constraints applied in this process have a solid foundation in both theory and practice. The problem is that we have only heuristic knowledge about how to combine them. Moreover, such rules of thumb are currently crafted by hand, and they do not fare well on images that vary in their global characteristics, such as contrast and amount of shadow. However, methods from machine learning, to which we now turn, may be able to induce better conditions for selecting or rejecting candidate roofs. If these acquired heuristics are more accurate than the existing handcrafted solutions, they will improve the reliability of the rooftop selection process.

4 A Review of Learning Techniques

We can formulate the task of acquiring rooftop selection heuristics in terms of supervised learning. In this process, training cases of some concept are labeled as to their class. In rooftop selection, only two classes exist — rooftop and non-rooftop — which we will refer to as positive and negative examples of the rooftop concept. Each instance consists of a number of attributes and their associated values, along with a class label. These labeled instances constitute training data that are provided as input to an inductive learning routine, which generates concept descriptions designed to distinguish the positive examples from the negative ones. These knowledge structures state the conditions under which the concept, in this case 'rooftop', is satisfied.

For this study, we selected five well-known learning methods: Quinlan's [1993] C4.5, Clark and Niblett's [1989] CN2, Nearest neighbor [e.g., Aha et al., 1992], Naive Bayes [e.g., Langley et al., 1992], and Perceptron learning [e.g., Zurada, 1992]. We chose these methods because they represent a range of representations, performance schemes, and learning mechanisms for supervised concept learning. In addition, they exhibit different inductive biases, meaning that the algorithms acquire certain concepts more easily than others. As a result, they should provide insights about the types of machine learning algorithms that are useful for the rooftop selection task.

C4.5 [Quinlan, 1993] constructs a 'decision tree' from training data. Each nonterminal node in such a tree specifies an attribute, and the emanating links indicate a value (or range of values), whereas terminal nodes specify a class name. Classification occurs by sorting an instance downward through the tree until it reaches a terminal node. Algorithms for inducing decision trees operate by selecting the attribute whose values best discriminate among the classes, partitioning the training data into subsets for each value (or range), then applying the process in turn to each of the resulting subsets. This recursive partitioning process divides the training data into ever smaller sets, until each set contains only one class or no further splits are possible. Some variants, including C4.5, include a 'pruning' stage that cuts back the tree after its construction.

CN2 [Clark and Niblett, 1989, Clark and Boswell, 1991] learns a set of conjunctive rules from training examples. To classify an unknown instance, CN2 determines which rules the instance satisfies. When rules from different classes match an instance, the system uses a probabilistic conflict resolution scheme to select the single best rule. CN2 uses a covering algorithm, much like AQ [Michalski, 1969], that constructs rules one at a time. It specializes a maximally general description until it finds a "best" rule, as determined by some evaluation criterion. CN2 removes those training examples covered by the rule and repeats this process with the remaining examples, creating additional rules until all examples have been covered. The system copes with continuous data by dividing the continuous range into discrete sub-intervals.

Another approach is the nearest neighbor method [e.g., Aha et al., 1991], which uses an 'instance-based' representation of knowledge that simply retains training cases in memory. This approach classifies new instances by finding the 'nearest' stored case, as measured by some distance metric, then predicting the class associated with that case. For numeric attributes, a common metric (which we also use in our studies) is Euclidean distance. In this framework, learning involves nothing more than storing each training instance, along with its associated class. Although this method is quite simple and has known sensitivity to irrelevant attributes, in practice it performs well in many domains. Some versions select the k closest cases and predict the majority class; here we will focus on the 'simple' nearest neighbor scheme, which uses only the nearest case.

A fourth alternative is the naive Bayesian classifier [e.g., Langley et al., 1992], which stores a probabilistic concept description for each class. This description includes an estimate of the class probability and the estimated conditional probabilities of each attribute value given the class. The method classifies new instances by computing the probability of each class using Bayes' rule, combining the stored probabilities by assuming that the attributes are independent given the class and predicting the class with the highest probability. Like nearest neighbor, naive Bayes has known limitations, such as sensitivity to attribute correlations and an inability to represent multiple decision regions, but in practice it behaves well on many natural domains.

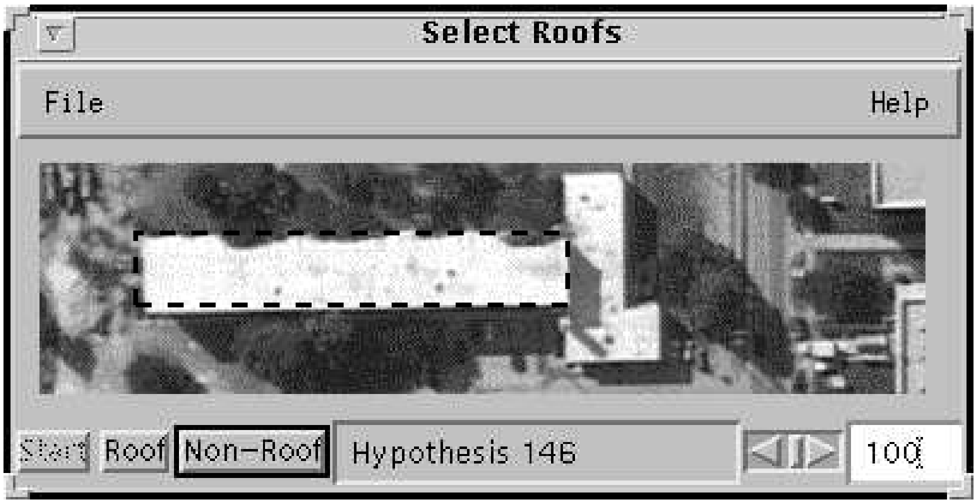

Figure 1: Visualization interface for roof hypotheses.

Finally, the Perceptron learning algorithm [e.g., Zurada, 1992] finds coefficients and a threshold for a linear discriminant function. The algorithm learns these values by applying a hill-climbing technique until it minimizes the error between the desired output and the actual output of the linear discriminant function. To classify unknown instances, the algorithm simply computes the weighted sum using the learned coefficients and the attribute values. If the weighted sum exceeds the threshold, then the system assigns the instance to one class; otherwise, it assigns the instance to the other class. No modifications are needed to handle continuous data (cf. CN2). This algorithm is well-studied and learns the same type of classifier as the Lin/Nevatia [1996] rule, which also takes the form of a linear discriminant function.

5 Representation and Labeling of Rooftop Hypotheses

To apply the above algorithms to the problem of roof hypothesis selection, we selected two data sets derived from aerial images of Fort Hood, Texas. The site contains 29 actual buildings. The first data set, derived from image FHOV1027, consists of 1179 roof hypotheses generated from a nadir view of the site; in contrast, the second data set, derived from image FHOV625, contains 2193 hypotheses produced from an oblique view taken at a different time. Each hypothesis in the two data sets is described in terms of nine features that summarize the evidence gathered from lower levels of analysis; these include evaluation of edge support, corner support, parallel support, orthogonal trihedral vertex support, shadow corner support, gap overlap, displacement of edge support, crossing lines, and existence of junctions. All nine features take on continuous values.

Before we could pass the hypotheses to a learning algorithm, we first had to label each one as either a positive or negative example of the desired concept. To accomplish this task easily, we implemented a visualization system, shown in Figure 1, that displays each roof hypothesis and lets the user label it as positive or negative. There are two problems with this approach. First, the user may have to label thousands of hypotheses, which would be time-consuming and tedious. To reduce the number of hypotheses a human has to label, we implemented a simple pre-screening algorithm that takes user-identified regions of interest (i.e., areas surrounding buildings) and determines how many corners of a rooftop hypothesis fall within this region. For example, if the user specified that two corners must fall within the region, then those hypotheses with fewer than two corners are labeled automatically as negative, and the remaining hypotheses are passed to the visualization system for human classification. For image FHOV1027, the pre-screening step reduced the number of hypotheses that required labeling from 1179 to 257.

The second problem is that the visualization system displays hypotheses in the visual space and not in the attribute space. Early experiences with the visualization system showed it was difficult to judge the quality of hypotheses simply by their visual characteristics (i.e., the set of four lines on the image). Consequently, we needed a feedback mechanism to show how the hypothesis looked in the visual and attribute spaces.

To address this problem, we incorporated a simple learning algorithm into the visualization system that uses a nearest neighbor classifier and its past experience to classify a new hypothesis. The system

Table 1: Experimental results for within-image learning using data from two Fort Hood images.

	(a) Image FHOV1027			(b) Image FHOV625		
	Positive Accuracy	Negative Accuracy	Overall Accuracy	Positive Accuracy	Negative Accuracy	Overall Accuracy
Lin/Nevatia	48.80±1.4	100.00±0.0	82.91±0.6	*69.64±1.8*	100.00±0.0	*91.58±0.3*
Naive Bayes	*74.56±1.5*	83.84±0.7	82.28±0.5	59.72±1.6	91.40±0.7	87.82±0.6
Nearest neighbor	60.08±1.9	93.00±0.6	87.50±0.5	54.32±1.9	94.00±0.3	89.63±0.4
C4.5 (w/pruning)	58.12±3.7	94.00±0.5	87.95±0.5	39.24±1.8	96.56±0.6	90.28±0.4
CN2	47.84±1.4	98.44±0.2	*89.99±0.4*	0.00±0.0	100.00±0.0	89.20±0.3
Perceptron	0.00±0.0	100.00±0.0	83.20±0.5	1.60±0.9	100.00±0.0	89.37±0.3

displays hypotheses classified to be roofs as green rectangles, non-roofs as blue rectangles, and atypical hypotheses as red rectangles. The user can set a "sensitivity threshold" that affects how distant from previous instances a hypothesis must be before it is labeled atypical. As the visualization system gains experience, it displays fewer and fewer atypical hypotheses. After the system classifies and displays a hypothesis, the user either confirms or overrides the system's decision. By the end of the labeling session, the user typically confirms most of the decisions made by the system.

6 Within-Image Learning

A typical machine learning experiment manipulates one or more independent variables and evaluates the effect of this manipulation by measuring one or more dependent variables. Since our hypothesis was that machine learning would produce classifiers that perform as well or better than handcrafted knowledge, the natural independent variable was the classifier used to label hypotheses, in particular whether one used the Lin/Nevatia heuristic versus a learned classifier. Because we were also interested in the behavior of different methods, we compared the Lin/Nevatia scheme to all the learning methods described in Section 4. The obvious dependent variable is overall accuracy of the learned knowledge on unseen instances, computed as the percentage of correctly classified test instances. However, the cost of discarding a hypothesis that corresponds to an actual rooftop is more expensive than retaining a hypothesis that does not. Consequently, we also measured accuracy on both the positive and negative instances.

Our first study took the form of a common type of experiment found in the machine learning literature and involved forming multiple partitions of a data set into training set/testing set pairs. For a given pair, one applies the learning algorithm to the training set and evaluates the acquired knowledge on the test set. Averaging results across all pairs provides a good estimate of the accuracy for a given algorithm, since it minimizes the effect of unrepresentative samples in the training or testing sets. We conducted this experiment using data derived from image FHOV1027 and image FHOV625. We randomly generated 25 partitions of the data set. For each partition, the training set consisted of 60% of the original data set, while the testing set consisted of the remaining 40%; we then applied all of the learning algorithms to each train/test pair. For this experiment, we used the MLC++ library of machine learning programs [Kohavi et al., 1996].

For comparison, we also ran the Lin/Nevatia [1996] selection criterion on each test set and computed its average accuracy. Because its authors constructed this heuristic manually, no training was involved. Of course, to the extent that Lin and Nevatia modified it in response to its behavior, they did 'train' their heuristic, but this training was conducted using separate images from a 'model board', which were photographs of a scale-model site for which ground truth was known. This does not provide the best comparison possible. However, our concern here is not with the origin of their heuristic but with trying to improve upon it.

Table 1 shows the results of the within-image learning experiments using data generated from images FHOV1027 and FHOV625. The table presents results for each image in terms of average positive, negative, and overall accuracy, with 95% confidence intervals. The highest accuracies for the positive class and overall are indicated by italics. For image FHOV1027 (Table 1a), naive Bayes performed the worst overall, but had the best performance for the positive class. On the other hand, CN2 performed best overall, but performed poorly in terms of positive accuracy. For image FHOV625 (Table 1b), the Lin/Nevatia classifier performed best in terms of overall accuracy and positive accuracy, whereas naive Bayes was second, but only in terms of positive accuracy.

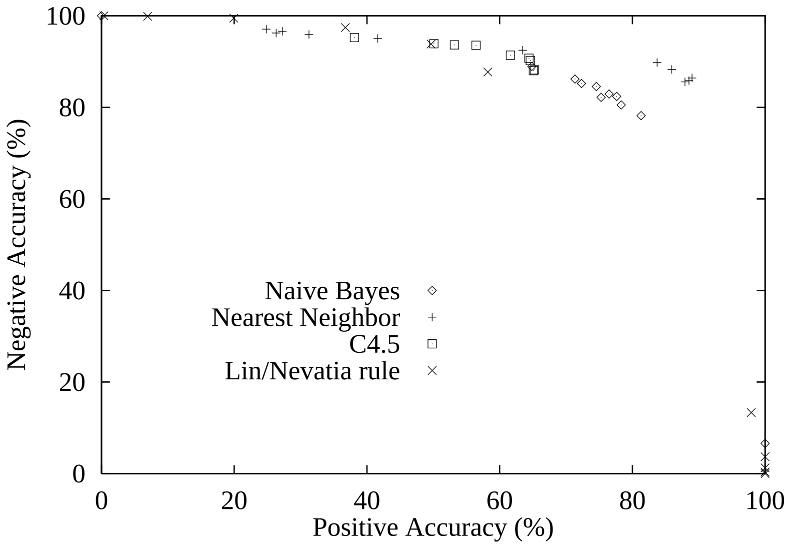

Figure 2: ROC curve for image FHOV1027.

To provide a context for these results, consider the performance of a frequency-based classifier that always predicts the most frequent class. Given the data generated from images FHOV1027 FHOV627, this rule would always predict a negative instance, resulting in a positive accuracy of 0%, a negative accuracy of 100%, and an overall accuracy reflecting the percentage of negative instances in the data set, or 87.1%. An extreme bias toward positive accuracy would lead one to always predict a positive instance, resulting in a positive accuracy of 100%, a negative accuracy of 0%, and an overall accuracy reflecting the percentage of positive instances in the data set, or 12.9%. The Lin/Nevatia rule, naive Bayes, and nearest neighbor all find comfortable trade-offs between positive and negative accuracy. These results are roughly consistent with our hypothesis, but they are hardly conclusive.

An important thing to note about the results on these two images is how the relative position of the machine learning methods stayed essentially the same. The one exception, on the second image, was that CN2 performed slightly worse than the Perceptron learning algorithm. This consistency is encouraging because it suggests that we can identify a single algorithm that will perform well across a range of images. Hopefully, experiments with more images will demonstrate the same trend.

7 The Cost of Misclassification

Given the above results, naive Bayes and nearest neighbor show promise of being comparable to the Lin/Nevatia rule in terms of accuracy on the positive instances. Ideally, we would like to improve upon their positive accuracy without losing much negative accuracy. To achieve this affect, the learning algorithm must be biased in favor of positive accuracy, but most machine learning methods do not provide ways to accomplish this. Pazzani et al. [1994] have done some preliminary work along these lines, which they describe as addressing differing costs of error types. The basic idea is to change the way the algorithm treats positive instances relative to negatives, either during the learning process or at the time of testing.

This approach should also give us more principled comparisons among the various learning methods. By systematically varying the relative costs, we can generate a Receiver Operating Characteristic (ROC) curve, which graphs negative accuracies as a function of positive accuracies. The ROC curve for each algorithm provides a cost-independent summary of its behavior, with curves that cover larger areas being generally better. This suggests a revision of our hypothesis from the previous section: as one varies error costs, machine learning will produce ROC curves with equal or larger areas than that covered by the Lin/Nevatia classifier.

To test this hypothesis, we implemented or obtained cost sensitive versions of the best performing algorithms from the previous experiments, namely naive Bayes, nearest neighbor, and C4.5. We defined a cost on the range $[-1.0, 1.0]$, where a negative setting means that mistakes on the positive class are more costly and a positive setting means that mistakes on the negative class are more costly. Although this formulation differs slightly from Pazzani et al.'s [1994], it is equivalent for two-class problems.

For naive Bayes, we used the cost sensitive measure to adjust the Bayesian posterior probabilities [Duda

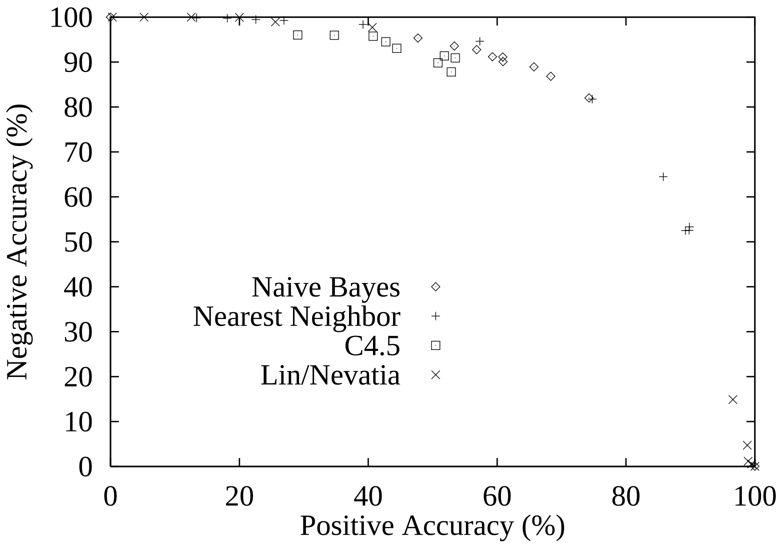

Figure 3: ROC curve for image FHOV625.

and Hart, 1973, Pazzani et al., 1994]. Specifically, the modified algorithm alters the posterior probability of the preferred class so that it becomes more probable. The class whose posterior probability is modified, and the degree to which it is changed, depends on the sign and magnitude of the cost metric.

We introduced a cost sensitive measure into the nearest neighbor algorithm by adjusting the distance from an unknown instance to its closest neighbor for each class, positive and negative. After making this adjustment based on the sign and magnitude of the cost measure, the classification process proceeds normally, assigning the class label of the "closest" neighbor. This modification also works for versions of the algorithm that consider more than one neighbor when classifying unknown instances.

We were able to make similar modifications to the Lin/Nevatia classifier [1996] for purposes of comparison. Since this classifier is a linear discriminant function, we adjusted the threshold such that the decision boundary is closer to the hypothetical cluster of positive examples or the cluster of negative examples. The direction and degree to which we adjust the threshold is again dependent on the sign and magnitude of the cost measure.

We also obtained a cost sensitive version of C4.5 [Grimmer, 1997], which takes a different approach to learning minimum cost classifiers than ours. Rather than using cost sensitive measures in the testing phase, it takes cost into account during the learning phase when it prunes the decision tree [Breiman et al., 1984]. Briefly, the pruning algorithm selects the least costly of three actions for each subtree: do nothing (i.e., leave the subtree unpruned); replace the subtree with a leaf node and assign the majority class label of the subtree to the leaf node; and replace the subtree with the subtree of one of its children. Costs for classes in this version of C4.5 are expressed on the integer range $[0, \infty]$.

To investigate the effect of misclassification costs, we conducted an experiment using the cost sensitive versions of nearest neighbor, naive Bayes, C4.5, and the Lin/Nevatia classifier. One condition used data from image FHOV1027, while another used data from image FHOV625. For each algorithm and image, we varied the cost metric and measured the resulting positive and negative accuracy for ten runs; each run involved partitioning the data set randomly into training (60%) and testing (40%) sets. For each cost setting and each classifier, we plotted the average true negative rate (i.e., accuracy on negative cases) against the true positive rate (i.e., accuracy on positive cases). Figures 2 and 3 show the ROC curves that resulted from this procedure for images FHOV1027 and FHOV625, respectively.

The most notable aspect of Figure 2 is that the curves for most of the learned classifiers are nearly identical with that for the Lin/Nevatia rule, giving support for our hypothesis that machine learning can at least match handcrafted knowledge. However, portions of the ROC curve for cost sensitive nearest neighbor stand out as substantially higher than others, suggesting that this method outperforms both the Lin/Nevatia and the alternative learning schemes. This difference was not apparent from our earlier experiment, and shows the clear advantage of using ROC curves to compare methods in cost sensitive domains.

Inspection of Figure 3 reveals additional support for our basic hypothesis, since again the Lin/Nevatia

Table 2: Experimental results for between-image learning using data from two Fort Hood images.

	(a) Train FHOV1027/Test FHOV625			(b) Train FHOV625/Test FHOV1027		
	Positive Accuracy	Negative Accuracy	Overall Accuracy	Positive Accuracy	Negative Accuracy	Overall Accuracy
Lin/Nevatia	*68.00±0.0*	100.00±0.0	*91.52±0.0*	50.00±0.0	100.00±0.0	83.12±0.0
Naive Bayes	33.72±0.5	98.12±0.1	91.32±0.1	*86.44±0.7*	64.12±1.4	67.88±1.0
Nearest neighbor	25.96±0.8	97.16±0.2	89.48±0.2	71.36±1.3	80.60±0.5	79.12±0.3
C4.5 (w/pruning)	24.56±1.5	98.00±0.3	89.98±0.2	58.60±2.5	84.40±0.9	80.10±0.7
CN2	60.00±0.0	100.00±0.0	89.60±0.0	0.00±0.0	100.00±0.0	83.30±0.0
Perceptron	0.00±0.0	100.00±0.0	89.15±0.0	1.28±0.8	100.00±0.0	*83.47±0.1*

curve runs the same course as those for most learned classifiers. But for this image, nearest neighbor does no better than the other induction algorithms, and portions of the C4.5 curve appear substantially worse. Clearly, we should replicate these results on more images, but the results so far are generally encouraging.

8 Between-Image Learning

We geared our final study more toward the goals of image analysis. Recall that our motivating problem is the large number of images that must be processed. In order to alleviate the burden on the image analyst, we want to apply knowledge learned from some images to many other images. We have already noted that several dimensions of variation pose problems to transferring learned knowledge to new images. For example, one viewpoint of a given site can differ from other viewpoints of the same site in both orientation and angle from the perpendicular. We need to better understand how the knowledge learned from one image generalizes to other images that differ along these dimensions. Images taken at different times and images of different sites present similar issues. Our hypothesis here was a stronger version of earlier ones: classifiers learned from one image would perform as well or better on unseen images than handcrafted classifiers. However, we also expected that such between-image learning would give less impressive results than the within-image situation.

To test our predictions, we simply examined generalization across the two images, which differ both in viewing angle and in time. Clearly, future experiments should systematically vary each of these independent variables, to determine their individual effect on transfer, but the current comparison should give us some insight into how well learned knowledge carries across images. For each learning algorithm, we conducted 25 runs in which 60% of the hypotheses from image FHOV1027 served as the training set and all hypotheses from image FHOV625 as the test set. In addition, we carried out the same procedure using 60% of the data from image FHOV625 for training and all of the data from image FHOV1027 for testing. For this experiment, we again took advantage of the MLC++ library of machine learning programs [Kohavi et al., 1996].

The results for this between-image experiment appear in Table 2. For the first condition (Table 2a), the Lin/Nevatia rule performed the same as in the second within-image learning condition (Table 1b), since the same data was used for testing. Recall that no comparable notion of generalization applies to the Lin/Nevatia classifier since it was handcrafted for unrelated data. The overall performance of naive Bayes was very high, but its predictive accuracy on the positive class was much less than for CN2 and the Lin/Nevatia classifier. CN2 achieved a respectable accuracy on the positive examples for this problem, in contrast to its performance in the experiment on within-image learning.

For the second between-image condition (Table 2b), the Lin/Nevatia rule again performed as it did in the first within-image condition (Table 1a), since the same data was used for testing. Although the Perceptron algorithm achieved the highest overall accuracy, CN2 and the Lin/Nevatia rule performed nearly as well. In terms of positive accuracy, naive Bayes and nearest neighbor performed highest; yet in terms of overall accuracy, these two methods fared the worst.

However, evaluating our hypothesis about between-image generalization involves comparing with accuracies from the within-image study, and here the results are mixed. Table 1b reports performance when we trained and tested the learning methods on image FHOV625, whereas Table 2a reports performance for training on FHOV1027 and testing on FHOV625. As predicted, the positive accuracies for naive Bayes, nearest neighbor, and C4.5 in the between-image case are lower than for the within-image condition, suggesting considerably less generalization across images than within them.

But, comparing Table 1a (training and testing on FHOV1027) with Table 2b reveals a different story. Here the positive accuracies are substantially higher for the between-image condition, suggesting that generalization was actually better across images than within them. This finding is more encouraging but runs counter to our expectation that training and testing on different images would be a more difficult learning task than training and testing on the same image. The results are further confused by CN2's behavior, which showed higher accuracies than expected in Table 2a but not in Table 2b.

Clearly, the results from this experiment are inconclusive, and we suspect that both the large decrease in FHOV1027 accuracies and the increase in FHOV625 accuracies are artifacts due to the particular distribution of positive and negative instances in these data sets. Repeating the study using cost sensitive versions of the learning algorithms, and calculating ROC curves for each pair of training and test images, should reveal the generalization for each case in a distribution-free manner. In this framework, we would expect that the area under the ROC curve for between-image learning on a given test image and a given method will be less than the area for within-image learning on the same image and method.

In summary, our experiments have revealed some factors that influence performance on the rooftop selection task — the learning algorithm used to acquire knowledge, the relative cost of classification errors, and the nature of the images themselves. We showed that, at least for positive accuracies, the naive Bayesian and nearest neighbor classifiers closely approach (and sometimes exceed) the handcrafted Lin/Nevatia heuristic. Although it is difficult to identify the conditions under generalization occurs between image, we have available a methodology that will help us investigate this issue further.

9 Related Work on Visual Learning

Research on learning in computer vision has become increasingly common in recent years. Papers by Conklin [1993], Sengupta and Boyer [1993], Cook et al. [1993], and Provan et al. [1996] all describe approaches to learning three-dimensional descriptions for use in object recognition. Another approach [e.g., Gros, 1993, Pope and Lowe, 1996] instead learns characteristic views for use in recognition, while still others focus on learning the appearances of objects or scenes [e.g., Nayar et al., 1996, Pomerleau, 1996, Viola, 1993].

Most work on visual learning ignores the importance of misclassification costs, but our work along these lines has some precedents. In particular, Draper et al. [1994] incorporate the cost of errors into their algorithm for constructing and pruning multivariate decision trees. They tested this approach on the task of labeling pixels from outdoor images for use by a road-following vehicle. They determined that, in this context, labeling a road pixel as non-road was more costly than the reverse, and showed experimentally that their method could reduce such errors on novel test pixels.

However, like much of the research on visual learning, Draper et al.'s work focused on image processing in complex scenes at eye level. One exception is Connell and Brady's [1987] work on learning structural descriptions of airplanes from aerial views. Their method converted training images into semantic networks, which it then generalized on comparison with descriptions of other instances. However, Connell and Brady do not appear to have tested experimentally their algorithm's ability to accurately classify objects in new images.

Draper [1996] reports a more careful study of learning in the context of analyzing aerial images. His approach adapts methods for reinforcement learning to assign credit in multi-stage image processing (for software similar to the Lin/Nevatia system), then uses an induction method (backpropagation in neural networks) to learn conditions on operator selection. He presents initial results on a RADIUS task that also involves the detection of roofs.

Our framework shares some features with Draper's approach, but assumes that learning is directed by feedback from a human expert. We predict that our supervised method will be more computationally tractable than his use of reinforcement learning, which is well known for its high complexity. Our approach does require more interaction with users, but we believe this interaction will be unobtrusive if cast within the context of an advisory system for image analysis.

10 Concluding Remarks

Although our initial studies have provided some insights into the role of machine learning in image analysis, much still remains to be done. For example, we may want to consider alternate measures of classification accuracy that take into account the presence of multiple valid hypotheses for a given rooftop. Classifying one of these hypotheses correctly is sufficient. In addition, although the rooftop selection stage was a natural place to start in applying our methods, we are also interested in working at both earlier and later levels of the process. Note that the goal here is not only to increase classification accuracy, which could be handled entirely by hypothesis selection, but to reduce the complexity of processing by removing bad hypotheses before they are aggregated into larger structures. With this aim in mind, we plan to extend our work to apply at all levels of the image understanding process.

We must address a number of issues before we can apply machine learning to other stages. One involves identifying the cost of different errors at each level, and taking this into account in our modified induction algorithms. Another concerns whether we should use the same induction algorithm at each level or use different methods at each stage. We should also explore using a number of learning methods in combination, either averaging their predictions (as in work on 'ensembles') or cascading the results (as in work on 'boosting').

As we mentioned earlier, in order to automate the collection of training data for learning, we also hope to integrate learning routines into the Lin/Nevatia software. This system was not designed initially to be interactive, but we would like to modify it so that the image analyst can accept or reject recommendations made by the image understanding system, generating training data in the process. At intervals the system would invoke its learning algorithms, producing revised knowledge that would alter the system's behavior in the future and, hopefully, reduce the user's need to make corrections. The interactive labeling system described in Section 5 could serve as an initial model for this interface.

In conclusion, our studies suggest that machine learning has an important role to play in improving the accuracy, and thus the robustness, of image analysis systems. However, we need additional experiments to give better understanding of the factors affecting between-image transfer and we need to extend learning to additional levels of the image understanding process. Also, before we can develop a system that truly aids the human image analyst, we must develop and implement unobtrusive ways to collect training data to support learning.

Acknowledgements

The authors thank Ram Nevatia, Andres Huertas and Andy Lin for their assistance in obtaining the images and data used for experimentation and providing valuable comments and advice. We would also like to thank Wayne Iba for his help with naive Bayes and Udo Grimmer for providing the cost sensitive version of C4.5. This research was conducted at the Institute for the Study of Learning and Expertise and in the Computational Learning Laboratory, Center for the Study of Language and Information, at Stanford University.

References

[Aha et al., 1991] Aha, D.W., Kibler, D. and Albert, M.K. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[Breiman et al., 1984] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. *Classification and regression trees.* Belmont, CA: Wadsworth, 1984.

[Clark and Boswell, 1991] Clark, P. and Boswell, R. Rule induction with CN2: some recent improvements. *Machine Learning — Proceedings of the Fifth European Conference (EWSL-91)*, 151–163. Berlin: Springer-Verlag, 1991.

[Clark and Niblett, 1989] Clark, P. and Niblett, T. The CN2 induction algorithm. *Machine Learning*, 3:261–284, 1989.

[Conklin, 1993] Conklin, D. Transformation-invariant indexing and machine discovery for computer vision. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 10–14. Menlo Park, CA: AAAI Press, 1993.

[Connell and Brady, 1987] Connell, J.H. and Brady, M. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31:159–183, 1987.

[Cook et al., 1993] Cook, D., Hall, L., Stark, L. and Bowyer, K. Learning combination of evidence functions in object recognition. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 139–143. Menlo Park, CA: AAAI Press, 1993.

[Draper et al., 1994] Draper, B.A., Brodley, C.E. and Utgoff, P.E. Goal-directed classification using linear machine decision trees. *IEEE Transactions on PAMI*, 16.9:888–893, 1994.

[Draper 1996] Draper, B. Learning grouping strategies for 2D and 3D object recognition. *Proceedings of the Image Understanding Workshop*, 1447–1454. San Francisco, CA: Morgan Kaufmann, 1996.

[Duda and Hart, 1973] Duda, R.O., and Hart, P.E. *Pattern classification and scene analysis.* New York, NY: Wiley, 1973.

[Grimmer, 1997] Grimmer, U. Personal communication. 1997.

[Gros, 1993] Gros, P. Matching and clustering: Two steps towards automatic object model generation in computer vision. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 40–44. Menlo Park, CA: AAAI Press, 1993.

[Kohavi et al., 1996] Kohavi, R., Sommerfield, D. and Dougherty, J. Data mining using MLC++: a machine learning library in C++. *Proceedings of IEEE Tools with Artificial Intelligence*, 234–245, 1996 (http://www.sgi.com/Technology/mlc).

[Langley et al., 1992] Langley, P., Iba, W. and Thompson, K. An analysis of Bayesian classifiers. *Proceedings of the Tenth National Conference on Artificial Intelligence*, 223–228. Menlo Park, CA: AAAI Press, 1992.

[Langley and Simon, 1995] Langley, P. and Simon, H.A. Applications of machine learning and rule induction. *Communications of the ACM*, 38 (November) 55–64, 1995.

[Lin and Nevatia, 1996] Lin, C. and Nevatia, R. Building detection and description from monocular aerial images. *Proceedings of the Image Understanding Workshop*, 461–468. San Francisco, CA: Morgan Kaufmann, 1996.

[Michalski, 1969] Michalski, R.S. On the quasi-minimal solution of the general covering problem. *Proceedings of the 5th International Symposium on Information Processing*, Vol. A3, 125–128, 1969.

[Nayar et al., 1996] Nayar, S.K., Murase, H. and Nene, S.A. Parametric appearance representation. In Nayar, S.K. and Poggio, T., eds., *Early visual learning*, 131–160. New York: Oxford University Press, 1996.

[Pazzani et al., 1994] Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T. and Brunk, C. Reducing misclassification costs. *Proceedings of the Eleventh International Conference on Machine Learning*, 217–225. San Francisco, CA: Morgan Kaufmann, 1994.

[Pomerleau, 1996] Pomerleau, D. Neural network vision for robot driving. In Nayar, S.K. and Poggio, T., eds., *Early visual learning*, 161–181. New York: Oxford University Press, 1996.

[Pope and Lowe, 1996] Pope, A.R. and Lowe, D.G. Learning probabilistic appearance models for object recognition. In Nayar, S.K. and Poggio, T., eds., *Early visual learning*, 67–97. New York: Oxford University Press, 1996.

[Provan et al., 1996] Provan, G., Langley, P. and Binford, T.O. Probabilistic learning of three-dimensional object models. *Proceedings of the Image Understanding Workshop*, 1403–1413. San Francisco, CA: Morgan Kaufmann, 1996.

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

[Sengupta and Boyer, 1993] Sengupta, K. and Boyer, K.L. Incremental model base updating: Learning new model sites. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 1–5. Menlo Park, CA: AAAI Press, 1993.

[Viola, 1993] Viola, P.A. Feature-based recognition of objects. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 60–64. Menlo Park, CA: AAAI Press, 1993.

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