Adaptive Image Interpretation : A Spectrum Of Machine Learning Problems

Vadim Bulitko BULITKO@UALBERTA.CA
Lihong Li LIHONG@CS.UALBERTA.CA
Greg Lee GREGLEE@CS.UALBERTA.CA
Russell Greiner GREINER@CS.UALBERTA.CA
Ilya Levner ILYA@CS.UALBERTA.CA

Department of Computing Science
University of Alberta
Edmonton, Alberta T6G 2E8, CANADA

Abstract

Automated image interpretation is an important task in numerous applications ranging from security systems to natural resource inventorization based on remote-sensing. Recently, a second generation of adaptive machine-learned image interpretation systems have shown expert-level performance in several challenging domains. While demonstrating an unprecedented improvement over hand-engineered or first generation machine learned systems in terms of cross-domain portability, design cycle time, and robustness, such systems are still severely limited. In this paper we inspect the anatomy of a state-of-the-art adaptive image interpretation system and discuss the range of the corresponding machine learning problems. We then report on the novel machine learning approaches engaged and the resulting improvements.

Keywords: learning from labeled and unlabeled data, automated operator and feature set selection, reinforcement learning, Markov decision processes, adaptive image interpretation.

1. Introduction & Related Research

Image interpretation is an important and highly challenging problem with numerous practical applications. Handcrafted image interpretation systems suffer from expensive design cycle, a high demand for expertise in both subject matter and computer vision, and the difficulties with portability and maintenance. Over the last three decades, various *automated* ways of constructing image interpretation systems have been explored. The following brief account is based on (Draper, 2003).

One of the promising approaches to automatic acquisition of image interpretation systems lies with treating computer vision as a control problem over a space of image processing operators. Early attempts used the schema theory (Arbib, 1972; Arbib, 1978). While presenting a systemic way of designing image interpretation systems, the approach was still *ad-hoc* in its nature and required extensive manual design efforts (Draper et al., 1996).

In the 1990's the second generation of control policy based image interpretation systems came into existence. More than a systematic design methodology, such systems used theoretically well-founded machine learning frameworks for automatic acquisition of control strategies over a space of image processing operators. The two well-known pioneering examples are a Bayes net system (Rimey & Brown, 1994) and a Markov decision process (MDP) based system (Draper et al., 2000).

The latter system (called ADORE for ADaptive Object REcognition) learned dynamic image interpretation strategies for finding buildings in aerial images. As with many vision systems, it identified objects (in this case, buildings) in a multi-step process. The input data were raw images, and the output was an interpretation which identified buildings in the image; in between, the data could be represented as intensity images, probability images, edges, lines, or curves. ADORE modelled image interpretation process as a Markov decision process, where the intermediate representations were continuous state spaces, and the vision procedures were actions. The goal was to learn a dynamic control policy that selects the next action (i.e., image processing operator) at each step so as to maximize the quality of the final interpretation.

ADORE, which was a pioneering system, left several exciting directions for future work and improvement. In this paper we discuss a spectrum of machine learning and decision making problems that need to be addressed before ADORE-like systems can become truly hands-off machine-learned tools capable of robust image interpretation portable across a wide variety of domains. These directions are investigated in a project titled MR ADORE (Multi Resolution ADORE).

Section 2 reviews the requirements and design of MR ADORE thereby demonstrating the critical assumptions it

Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data, Washington DC, 2003.

Figure 1. Artificial tree plantations result in simple forest images. Shown on the left is an original photograph. The right image is its desired labeling provided by an expert as a part of the training set.

makes and the resulting difficulties. Section 3 then reports on the solution approaches employed to date and discusses the results. Throughout the paper the task of forest canopy interpretation from aerial photographs is used as the testbed.

2. MR ADORE Design Objectives

Our extension, MR ADORE (Bulitko et al., 2002), was designed with the following objectives as its target: (i) rapid system development for a wide class of image interpretation domains; (ii) low demands on subject matter, computer vision, and AI expertise on the part of the developers; (iii) accelerated domain portability, system upgrades, and maintenance; (iv) adaptive image interpretation wherein the system adjusts its operation dynamically to a given image; (v) user-controlled trade-offs between recognition accuracy and resources utilized (e.g., time required).

These objectives favor the use of readily available off-the-shelf image processing operator libraries (IPL). However, the domain independence of such libraries requires an intelligent policy to control the application of library operators. Operation of such control policy is a complex and adaptive process. It is *complex* in that there is rarely a one-step mapping from image data to image label; instead, a series of operator applications are required to bridge the gap between raw pixels and semantic objects. Examples of the operators include region segmentation, texture filters, and the construction of 3D depth maps. Figure 2 presents a partial IPL operator dependency graph for the forestry domain.

Image interpretation is an *adaptive* process in the sense that there is no fixed sequence of actions that will work well for all/most images. For instance, the steps required to locate and identify isolated trees are different from the steps required to find connected stands of trees. Figure 3 demonstrates two specific forestry images that require significantly different operator sequences for satisfactory interpretation results.

The success of adaptive image interpretation systems therefore depends on the solution to the control problem: for a given image, what sequence of operator applications will most effectively and reliably interpret the image?

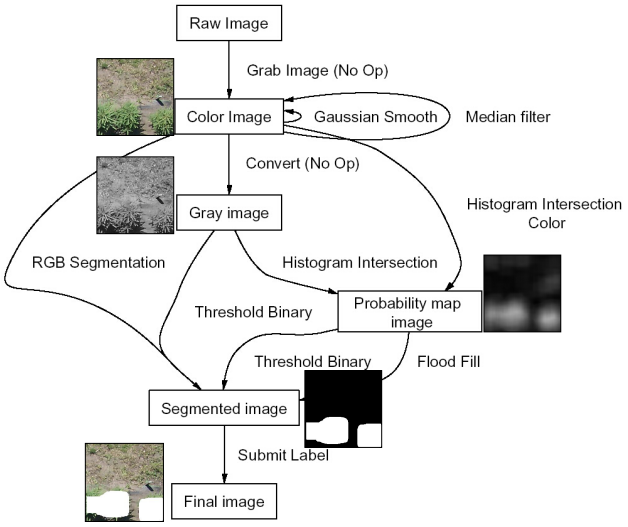

Figure 2. A small fragment of the operator graph for the domain of forest image interpretation. Most operators were ported from the freely available Intel Open CV and Intel IPL libraries. Sample images are shown next to the data types.

3. MR ADORE Operation

MR ADORE starts with the Markov decision process (MDP) as the basic mathematical model by casting the IPL operators as the MDP actions and the results of their applications as the MDP states. In the context of image interpretation, the formulation frequently leads to several challenges absent in the standard heuristic search/MDP domains such as the grid world, the 8 puzzle (Reinefeld, 1993), etc. (i) Each individual state is so large (on the order of several mega-bytes), that we cannot use standard machine learning algorithm to learn the heuristic function. Selecting optimal features for sequential decision-making is a known challenge in itself. (ii) The number of allowed starting states (i.e., the initial high-resolution images) alone is effectively unlimited for practical purposes. In addition, certain intermediate states (e.g., probability maps) have a continuous nature. (iv) There are many image processing operators (leading to a large branching factor); moreover, many individual operators are quite complex, and can take hours of computation time each. (v) Goal states are not easily recognizable as the target image interpretation is usually not known *a priori*. This renders the standard complete heuristic search techniques (e.g., depth-first, A*, IDA* (Korf, 1985)) inapplicable directly.

In response to these challenges MR ADORE employs the following off-line and on-line machine learning techniques. First, we can use training data (here, annotated images) to provide relevant domain information . Each training datum is a source image, annotated by a user with the desired out-

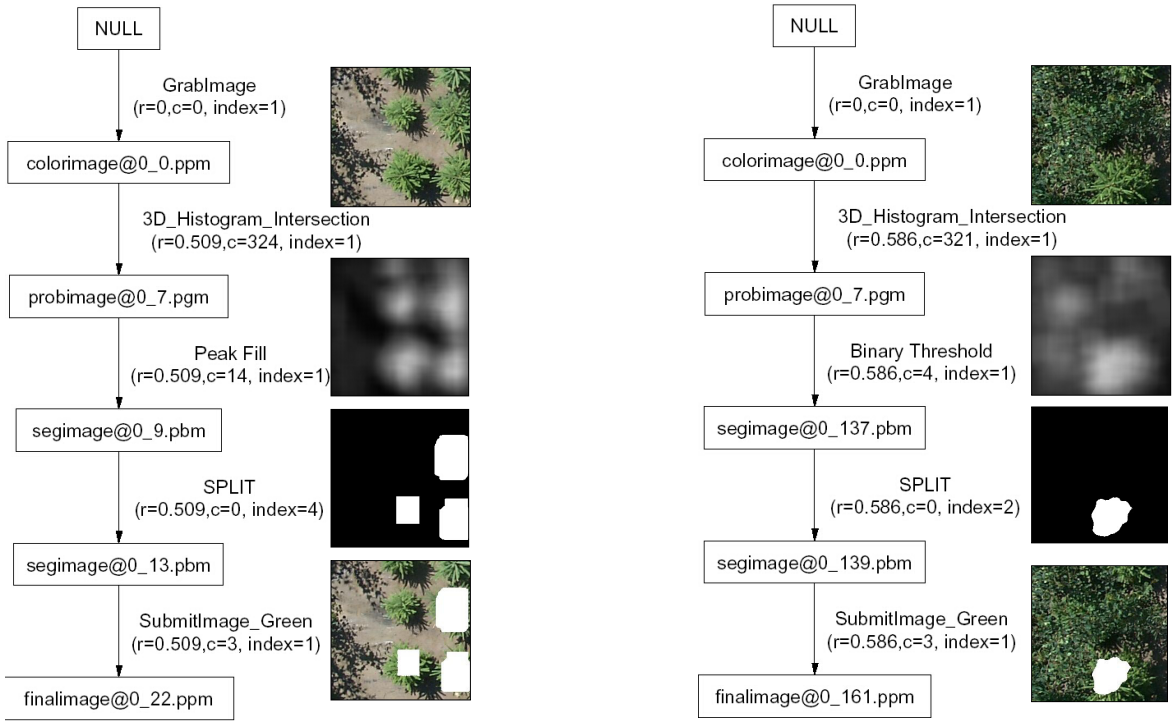

Figure 3. Adaptive nature of image recognition: two different input images require significantly different satisfactory operator sequences. Each node is labeled with its data type. Each arc between two data tokens is shown with the operator used.

put. Figure 1 demonstrates a training datum in the forestry image interpretation domain.

Second, during the off-line stage the state space is explored via limited depth expansions of training images. Within a single expansion all sequences of IPL operators up to a certain user-controlled length are applied to the training image. Since training images are user-annotated with the desired output, terminal rewards can be computed based on the difference between the produced labeling and the desired labeling. Then, dynamic programming methods (Barto et al., 1995) are used to compute the value function for the explored parts of the state space. We represent the value function as $Q : S \times A \rightarrow R$ where S is the set of states and A is the set of actions (here, IPL operators). The true $Q(s, a)$ computes the maximum cumulative reward the policy can expect to collect by taking action a in state s and acting optimally thereafter.

Automatically constructed features $f(s)$ are used to abstract relevant attributes of large domain states thereby making the machine learning methods practically feasible. Then supervised machine learning extrapolates the sample values computed by dynamic programming on the explored fraction of the state space onto the entire space.

Finally, when presented with a novel input image to interpret, MR ADORE first computes the abstracted version $f(s)$, then applies the machine-learned heuristic value function $Q(..)$ to compute $Q(f(s), a)$ for each IPL operator a; it then performs the action $a^* = \arg_a \max Q(f(s), a)$. The process terminates when the policy executes action `Submit(⟨ labeling ⟩)` where ⟨ `labeling` ⟩ becomes the system's output.

4. Machine Learning Problems

4.1. Automated Operator Selection

During the off-line phase, MR ADORE explores the state space by expanding the training data provided by the user. In doing so it applies all operator sequences up to a certain depth d. Larger values of d are preferable, as this allows more operators to be applied, which can lead to better performance. Even short sequences (4-6 operators) clearly exhibit the benefits; see Figure 4.

On the other hand, the size of the state space being explored increases exponentially with the depth d and therefore the search can become prohibitively expensive. With the current operator set used in MR ADORE for the tree canopy recognition task, the effective branching factor is approximately 26.5 which results in the sizes and timings shown in Table 1.

There are three conflicting factors at work: (i) larger off-the-shelf image processing operator libraries are required to make MR ADORE cross-domain portable, (ii) longer operator sequences are needed to achieve high interpretation quality, and (iii) combinatorial explosion during the off-line phase can impose prohibitive requirements on the storage and processing power. Fortunately, commercial

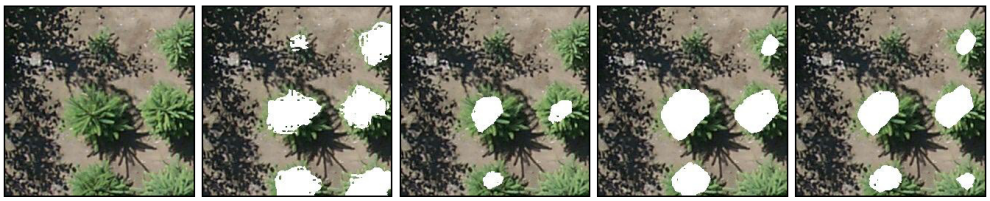

Figure 4. Longer operator sequences lead to better labeling. From left to right: the original image, desired user-provided labeling, the best labelings with an operator sequence of length 4, 5, and 6.

Table 1. Off-line state space exploration. All operator sequences up to a fixed length are applied to an image. The number of nodes and sequences, explored state space physical size (GBytes), and the expansion time are averaged over 10 images. A dual processor Athlon MP 1600+ running Linux was used.

Sequence Length	# of Nodes	# of Sequences	Size (GBytes)	Time
4	269	119	0.038	30 sec
5	7,382	3,298	1	10 min
6	192,490	86,037	26	8 hrs

domain-independent operator libraries almost invariably contain numerous operators that are redundant or ineffective *for some specific domain*. Therefore, the feasibility of the off-line learning phase as well as subsequent on-line performance critically depends on the selection of an efficient operator library for a particular domain. Figure 5 demonstrates the considerable difference in the best possible interpretations obtainable with three operator sets.

Previous systems (e.g., (Draper et al., 2000)) relied on *manual* selection of the highly relevant non-redundant operators thereby keeping the resulting IPL small and the off-line state space exploration phase feasible. Unfortunately, such solutions defeat the main objective of MR ADORE-like systems: *automatic* construction of an interpretation system for a novel domain. Therefore, we have implemented the following effective approach for *automated* operator selection.

Filter operator selection methods attempt to remove some operators based on system-independent criteria, such as operator redundancy, relevance, and other metrics. While being less expensive than running the target system, such optimization criteria may not deal well with the higher interdependence within operator sets in comparison to that of feature sets (for which filter methods are traditionally used).

Wrapper approaches account for the tight coupling of sequentially used operators by invoking the target system with a candidate operator set. While being more accurate, their optimization criteria can be too computationally expensive to be practically feasible. For instance, in the context of MR ADORE measuring the performance of a typical operator set on a test suite of 48 images takes around 12 hours on a dual AMD Athlon MP+ 2400 Linux server.

Our approach combines the strength of both schools by using wrapper-like heuristic search in the space of operator sets. Unlike traditional wrapper methods, however, we guide the search with a *fast* system-specific fitness function. In order to keep down the amount of human intervention, we employ machine learning methods to induce an approximation of the actual fitness function. This is done by evaluating a collection of randomly drawn operator sets via running the actual system (MR ADORE). The fitness of operator set $\mathcal{S}$ is defined as: $r(\mathcal{S}) - \alpha|\mathcal{S}|$, where $r(\mathcal{S})$ is the best reward MR ADORE is able to gain with the operator set $\mathcal{S}$ averaged over a suite of training images; $|\mathcal{S}|$ is the size of the operator set; and α is a scaling coefficient.

In the preliminary experiments to date (Bulitko & Lee, 2003), we have used various supervised machine learning approaches including decision trees, decision lists, naive Bayes, artificial neural networks, and perceptrons to approximate the fitness function. Genetic algorithms, simulated annealing, and forward pass greedy selection were used as the heuristic search techniques in the space of operator sets. The empirical evidence showed a 50% reduction in the operator set size while allowing MR ADORE to maintain the image interpretation accuracy of 97.8% of that of the full operator set.

4.2. Learning From Unlabeled Data

During the off-line phase MR ADORE samples the value function by computing the rewards of various interpretations of the same image resulting from all possible operator sequences. This is done by comparing the interpretations produced with the desired interpretation supplied by the user. The rewards are then propagated onto all intermediate data tokens using dynamic programming methods. The resulting collection of [data token, action, reward] tuples is used to train the function $Q(..)$, whose use was described in the previous sections.

Clearly, the efficacy of this approach depends on the sufficient volume of user-provided *labeled images*. In the domain of forest inventorization, manual labeling of tree canopies on aerial photographs is a fairly expensive (tens of thousands of dollars per 20-30 high resolution images), slow, and error prone process. One study (Gougeon, 1995) found the typical human interpretation error to be between 18 and 40%. On the other hand, unlabeled data can often be captured automatically from aerial and orbit-based platforms. Therefore, applications of MR ADORE-like sys-

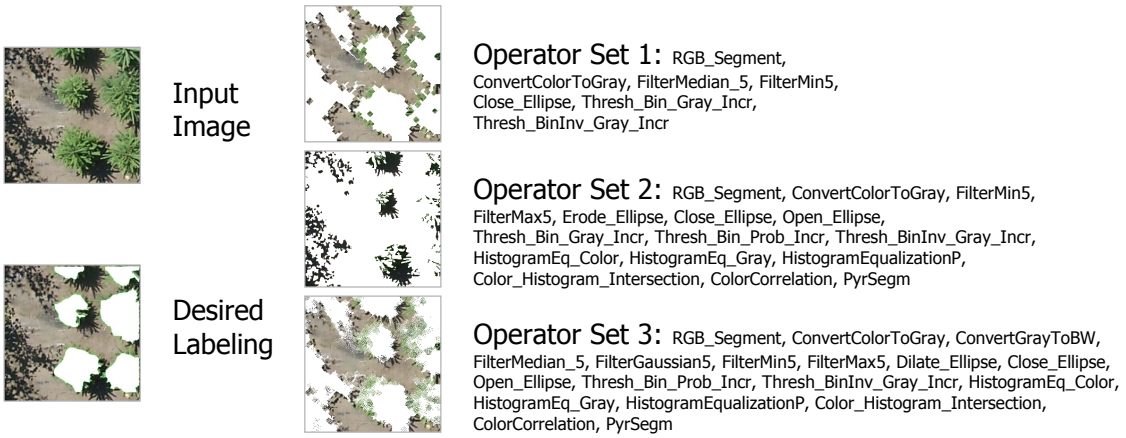

Figure 5. Three operator sets shown with their best possible labelings of a particular image.

tems can be widened significantly if the unlabeled data can be exploited.

A well-known and widely used approach to dealing with missing values is EM (Dempster et al., 1977). Typically there are two steps in each iteration of the algorithm — *expectation* (E-step) and *maximization* (M-step). Given a family model parameterized by some unknown parameter Θ and some data D, EM aims at finding the optimal $\hat{\Theta}$ to maximize the probability $P(D \mid \Theta)$. The algorithm computes the distribution over missing values in D, based on the Θ, then uses this completed data set to estimate the most likely parameter value. It was proved that unless Θ reaches a stable point, EM will always increase $P(D \mid \Theta)$.

Co-training (Blum & Mitchell, 1998; Abney, 2002) is a learning paradigm to address problems with strong structural prior knowledge. It assumes that: (i) there exist more than one *independent* feature representation of the data; (ii) these representations are *redundant* in that each one alone can be used for determining the labels; (iii) the representations are *consistent* in that the concept function defined over one representation agrees on the labels with the concept function that is defined over another representation. Under these assumptions, different learners are built for different representations. Then the classifier built by one learner can be used to estimate some of the unknown labels; another learner can uses these newly-labeled data points, along with the original labeled data, to produce its classifier. This new classifier can be used to produce labels for other unlabeled data points, which can be given to the first learner, etc.

Both EM and co-training can be viewed as bootstrapping algorithms in the sense that they work by repeatedly estimating missing values and learning from the estimates. Our approaches share the basic idea of bootstrapping. While many previous co-training systems (Blum & Mitchell, 1998; Nigam et al., 2000; Szummer & Jaakkola, 2001; Szummer & Jaakkola, 2002) focused on learning effective *classifiers* (for text or similar tasks), MR ADORE learns a *control policy* in the image interpretation domain. More

Algorithm I

Input: labeled data: D_L; unlabeled data: D_U; two learners: A and B.

1. **Initialize:** $D'_U \leftarrow \{\}$.
2. **Repeat until some termination condition is satisfied:**
   (a) Train A with $D_L \cup D'_U$;
   (b) Train B with D_L;
   (c) Select a random subset of unlabeled data: $\Delta D'_U \subset D_U$;
   (d) B labels the data in $\Delta D'_U$;
   (e) $D'_U \leftarrow D'_U \cup \Delta D'_U$;
3. **Output:** the classifier produced by learner A.

Figure 6. Algorithm I

importantly, we consider the problem of semi-supervised learning in the framework of sequential decision making, which means that the reward gained by the resulting policy is more important than other measurements such as mean-squared-errors of the Q-function.

Each of our approaches involved two learners; we considered neural networks (Hagan et al., 1996) and k-nearest-neighbors (Cover & Hart, 1967). In the algorithms, some data in D_U are labeled and denoted D'_U, D'_{UA}, or D'_{UB}.

Figure 6 illustrates the first algorithm. Labeled data D_L are used to train two weak learners A and B. In order to improve A, learner B repeatedly selects and labels randomly selected data samples in D_U. The data are then added to D'_U to train A in later iterations. The size of D'_U increases as training goes on.

The second algorithm (Figure 7) extends the idea and enables A to influence B. That is, in each iteration, B labels some randomly selected samples in D_U. The samples are then used to train A. Consequently, A labels some randomly selected data in D_U which are then used to train B. The algorithm, thus, proceeds in the co-training fashion. The sizes of D'_{UA} and D_{UB} are gradually increased.

We apply the two algorithms to learning Q-function in

Algorithm II
Input: labeled data: D_L; unlabeled data: D_U; two learners: A and B.

1. **Initialize:** $D'_{UA} \leftarrow \{\}$ and $D'_{UB} \leftarrow \{\}$.
2. **Repeat until some termination condition is satisfied:**
   (a) Train A with $D_L \cup D'_{UA}$;
   (b) Train B with $D_L \cup D'_{UB}$;
   (c) Select two subsets of unlabeled data $\Delta D'_{UA}$ and $\Delta D'_{UB}$ randomly from D_U;
   (d) Estimate the labels: A labels $\Delta D'_{UB}$; B labels $\Delta D'_{UA}$;
   (e) $D'_{UA} \leftarrow D'_{UA} \cup \Delta D'_{UA}$; $D'_{UB} \leftarrow D'_{UB} \cup \Delta D'_{UB}$;
3. **Output:** the classifier produced by learner A.

Figure 7. Algorithm II

MR ADORE and evaluate the resulting policy in terms of MSE and, more importantly, the relative reward. The latter is the actual interpretation reward gained by the policy learned relative to the maximum reward achievable in the system. We used texture features (e.g., local binary patterns) as well as other features (e.g., mean values and histograms of RGB/HSV) in the experiments to date. Two learners: a multi-layer feed-forward artificial neural network (ANN) and a k-nearest-neighbor (kNN) were used. Figure 8 demonstrates how well the ANN performs when trained on labeled data only.

Approximating the true Q-function with kNN works by interpolating the Q-values from the k state-action pairs nearest to the state-action pair at hand. Thus, training data resulting from the expert-labeled input images leads a database of cases for each operator in MR ADORE. Each case is recorded in terms of the features extracted off the data token and the actual Q-value of applying the operator on the token.

Initially, both D_L and D_U are constructed by selecting random subsets; for the nearest neighbor algorithm we used $k = 1$. We fix $|D_L| = 20$ and the number of unlabeled data added to D'_U (referred to as $\mathrm{unlabel\_n} = |\Delta D'_U|$) is also fixed. So in each iteration, a random subset of $\mathrm{unlabel\_n}$ data tuples are selected from D_U and added to D'_U for further training.

In the first set of experiments, kNN used the Euclidean distance (i.e., the l_2-norm of two vectors) in the HSV histogram space to measure the proximity of two data tokens (e.g., images or probability maps). Figure 9 illustrates the results. Remarkably, even though the mean-squared-errors with Algorithm I are worse than those in Figure 8, the average relative reward is improved by approximately 4% ($unlabel\_n = 20$), 15% ($unlabel\_n = 60$) and 8% ($unlabel\_n = 100$) which is equivalent to supervised training with 10, 80 and 20 *additional user-labeled* images, respectively.

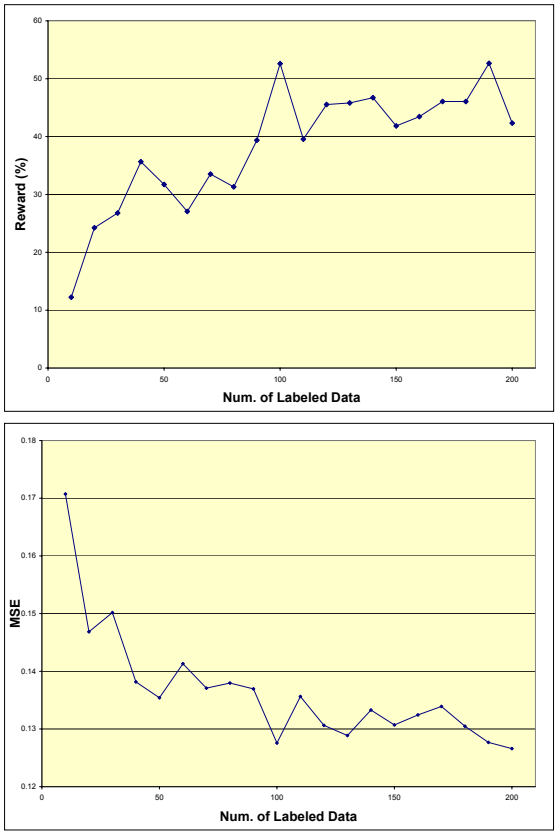

Figure 8. Training ANN-represented Q-function: the relative reward and the mean-squared-error vs. the amount of labeled data. Mean RGB values of the images were used as the features. Both the learning rate and momentum are fixed at 0.3.

Note that the Euclidean proximity of two data tokens in the feature space is not a reliable indication of similarity between their Q-values. Indeed, frequently in sequential decision-making the same operator should be favored on two images distant in the feature space if the features are suboptimal. Therefore, the optimal metric function to be used with a kNN approximator of the Q-function is non-trivial to hand-craft. Hence, we train a three-layer feed-forward ANN (referred to as ANN-k) to be the distance function used within the kNN co-training experiments.

Figure 10 shows the result of the second algorithm with a machine learned distance metric (ANN-k). With the same amount ($|D_L| = 20$) of labeled data, the accuracy was improved by approximately 20% ($unlabel\_n = 10$), 4% ($unlabel\_n = 15$), and 7% ($unlabel\_n = 20$), respectively, although the mean-squared-errors are slightly higher than those in Figure 8. These improvements are roughly equivalent to supervised training with 80, 10, and 30 additional expert-labeled images, respectively.

5. Conclusions and Future Research

Future research directions include further analysis of the relative contributions of the different ML techniques en-

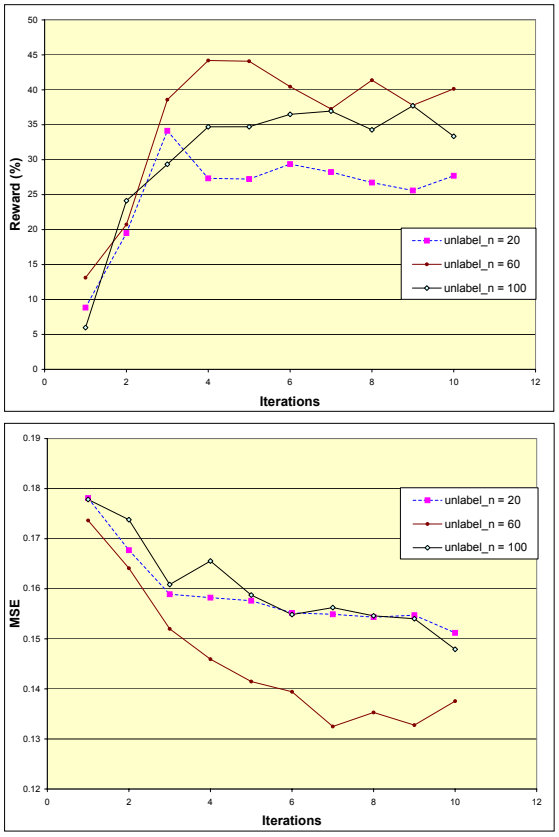

Figure 9. Experimental results using Algorithm I. The number of unlabeled data samples added to D'_U in each iteration is fixed at 20, 60 or 100, and $|D_L| = 20$. ANN uses mean RGB values of an image as features, while KNN uses the Euclidean distance in the HSV histogram space to measure the proximity of two data.

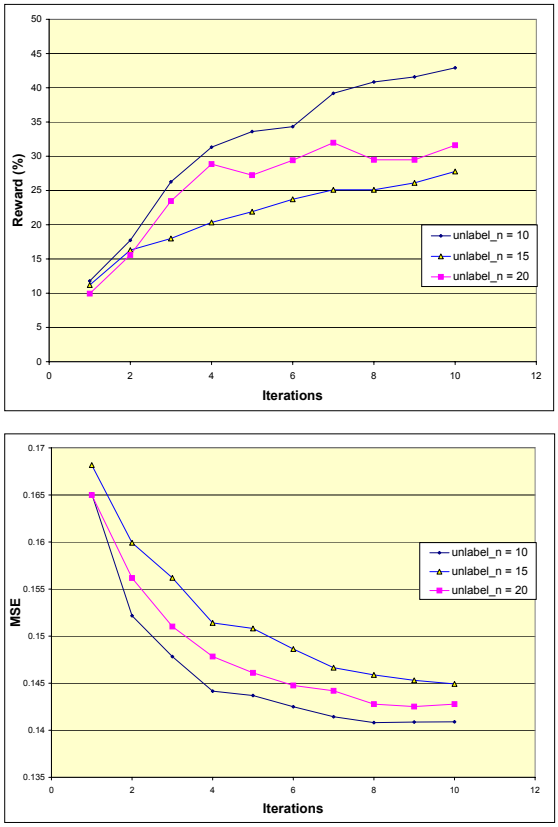

Figure 10. Experimental results using Algorithm II. The number of unlabeled data added to D'_U in each iteration is fixed at 10, 15 or 20, and $|D_L| = 20$. ANN uses mean RGB values of an image as features, while kNN uses a machine learned distance metric (ANN-k) to estimate the Q-value-relevant distance between two data in the texture feature space.

gaged, scaling up experiments, automated methods for policy construction, additional machine learning methods for the value functions, explicit time and resource considerations for real-time operation, explicit backtracking for the off-line dynamic programming methods, and RTA*-style lookahead enhancements (Korf, 1990).

Conventional ways of developing image interpretation systems usually require a significant subject matter and computer vision expertise from the developers. The resulting systems are expensive to upgrade, maintain, and port to other domains.

More recently, second-generation adaptive image interpretation systems (Bulitko et al., 2002) used machine learning methods to (i) reduce the human input in developing an image interpretation system for a novel domain and (ii) increase the robustness of the resulting system with respect to noise and variations in the data.

In this paper we presented and analyzed a state-of-the-art adaptive image interpretation system called MR ADORE and demonstrated several important machine learning and decision making problems that need to be addressed. We then reported on the progress achieved in each of the directions with supervised and unsupervised machine learning methods.

While early in its development, MR ADORE has already demonstrated a typical image interpretation accuracy of 70-90% in the challenging domain of forest image interpretation. As Figure 11 illustrates, it can outperform the best static policies as well as human interpreters.

Acknowledgements

Bruce Draper participated in the initial MR ADORE design stage. Lisheng Sun, Yang Wang, Omid Madani, Guanwen Zhang, Dorothy Lau, Li Cheng, Joan Fang, Terry Caelli, David H. McNabb, Rongzhou Man, and Ken Greenway have contributed in various ways. We are grateful for the funding from the University of Alberta, NSERC, and the Alberta Ingenuity Centre for Machine Learning.

References

Abney, S. (2002). Bootstrapping. *Neural Information Processing Systems: Natural and Synthetic (NIPS-02)*.

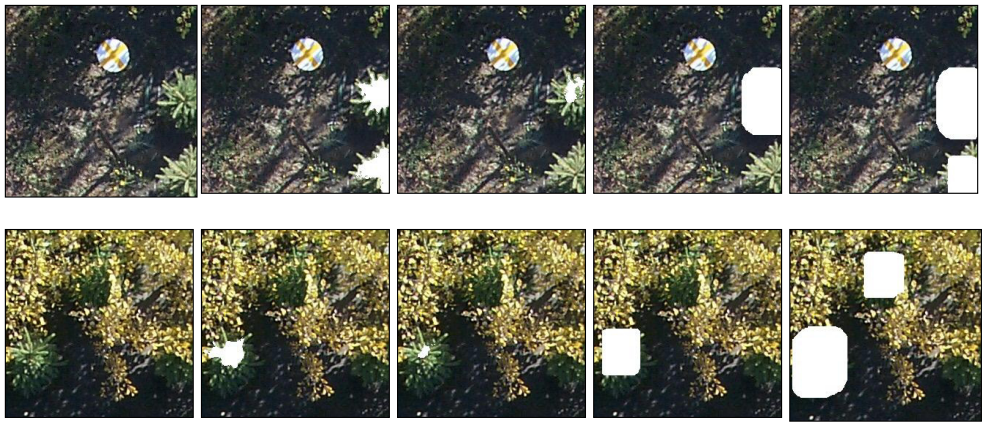

Figure 11. Adaptive kNN-guided policies outperform best static policies (top row) and sometimes outperform human labelers (bottom row). Each row from left to right: the original image, desired user-provided labeling, optimal labeling, best static policy of length 4 labeling, 1-NN policy labeling.

Arbib, M. (1972). *The metaphorical brain: An introduction to cybernetics as artificial intelligence and brain theory*. New York: Wiley-Interscience.

Arbib, M. (1978). Segmentation, schemas, and cooperative computation. In S. Levin (Ed.), *MAA studies in Mathematics*, 118–155.

Barto, A. G., Bradtke, S. J., & Singh, S. P. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence, 72*, 81–138.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory*. Morgan Kaufmann Publishers.

Bulitko, V., Draper, B., Lau, D., Levner, I., & Zhang, G. (2002). *MR ADORE : Multi-resolution adaptive object recognition (design document)* (Technical Report). University of Alberta.

Bulitko, V., & Lee, G. (2003). *Automated operator selection for adaptive image interpretation* (Technical Report). University of Alberta.

Cover, T. M., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, IT-13*, 21–27.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, 39*, 1–38.

Draper, B., Bins, J., & Baek, K. (2000). ADORE: adaptive object recognition. *Videre, 1*, 86–99.

Draper, B., Hanson, A., & Riseman, E. (1996). Knowledge-directed vision: Control, learning and integration. *Proceedings of the IEEE, 84*, 1625–1637.

Draper, B. A. (2003). From knowledge bases to Markov models to PCA. *Proceedings of Workshop on Computer Vision System Control Architectures*. Graz, Austria.

Gougeon, F. (1995). A system of individual tree crown classification on conifer stands at high spatial resolutions. *Proceedings of the 17th Canadian Symposium on Remote Sensing* (pp. 635–642).

Hagan, M. T., Demuth, H. B., & Beale, M. (1996). *Neural network design*. PWS Publishing Company.

Korf, R. (1985). Depth-first iterative deepening : An optimal admissible tree search. *Artificial Intelligence, 27*.

Korf, R. E. (1990). Real-time heuristic search. *Artificial Intelligence, 42*, 189–211.

Nigam, K., Mccallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*, 103–134.

Reinefeld, A. (1993). Complete solution of the eight-puzzle and the benefit of node ordering in IDA. *Proceedings of IJCAI-93* (pp. 248–253).

Rimey, R., & Brown, C. (1994). Control of selective perception using bayes nets and decision theory. *International Journal of Computer Vision, 12*, 173–207.

Szummer, M., & Jaakkola, T. (2001). Kernel expansions with unlabeled data. *Advances in Neural Information Processing Systems* (pp. 626–632).

Szummer, M., & Jaakkola, T. (2002). Partially labeled classification with Markov random walks. *Advances in Neural Information Processing Systems*.