Towards Automated Creation of Image Interpretation Systems

Ilya Levner, Vadim Bulitko, Lihong Li, Greg Lee, and Russell Greiner

University of Alberta,
Department of Computing Science,
Edmonton, Alberta, T6G 2E8, CANADA
{ilya|bulitko|lihong|greglee|greiner}@cs.ualberta.ca

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 and first generation machine-learned systems in terms of cross-domain portability, design-cycle time, and robustness, such systems are still severely limited. This paper inspects the anatomy of the state-of-the-art Multi resolution Adaptive Object Recognition framework (MR ADORE) and presents extensions that aim at removing the last vestiges of human intervention still present in the original design of ADORE. More specifically, feature selection is still a task performed by human domain experts and represents a major stumbling block in the creation process of fully autonomous image interpretation systems. This paper focuses on minimizing such need for human engineering. After discussing experimental results, showing the performance of the framework extensions in the domain of forestry, the paper concludes by outlining autonomous feature extraction methods that may completely remove the need for human expertise in the feature selection process.

Keywords: Computer Vision, Machine learning, Reinforcement learning.

1 Introduction & Related Research

Image interpretation is an important and highly challenging problem with numerous practical applications. Unfortunately, hand engineering an image interpretation system requires a long and expensive design cycle as well as subject matter and computer vision expertise. Furthermore, hand-engineered systems are difficult to maintain, port to other domains, and tend to perform adequately only within a narrow range of operating conditions atypical of real world scenarios. In response to the aforementioned problems, various *automated* ways of constructing image interpretation systems have been explored in the last three decades [10].

Based on the notion of "goal-directed vision" [9], a promising approach for autonomous system creation lies with treating computer vision as a control problem over a space of image processing operators. Initial systems, such as the Schema System[9], had control policies consisting of ad-hoc, hand-engineered rules. While presenting a systemic way of designing image interpretation systems, the approach still required a large degree of human intervention. 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 [15] and a Markov decision process (MDP) based system [8].

Our research efforts have focused on automating the latter system, called ADaptive Object REcognition system (ADORE), which learned dynamic image interpretation strategies for finding buildings in aerial images [8]. As with many vision systems, it identified objects (in this case buildings) in a multi-step process. Raw images were the initial input data, while image regions containing identified buildings constituted the final output data; in between the data could be represented as intensity images, probability images, edges, lines, or curves. ADORE modelled image interpretation 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 image interpretation.

As a pioneering system, ADORE proved that a machine learned control policy was much more adaptive that its hand-engineered counterparts by outperforming any hand-crafted sequence of operators within its library. In addition, the system was effortlessly ported to recognize stationary (staplers, white-out, etc.) in office scenes and again was shown to outperform operator sequences designed by human domain experts [7]. However, the framework of ADORE was still limited in a number of ways and left several directions for future work and improvement. This paper discusses perhaps the biggest stumbling block to realization of autonomous image interpretation systems, namely, the need for hand-crafted features. The project that investigates approaches to fully autonomous object recognition systems is named MR ADORE for Multi-Resolution ADaptive Object REcognition [3].

The rest of the paper is organized as follows. First, we review the requirements and design of MR ADORE, in order to demonstrate the critical assumptions made and the resulting difficulties. We then present framework extensions that minimize the need for hand-crafted features and present experimental results of applying the upgraded system to the domain of forestry. The paper concludes with future research directions on how to completely replace domain experts with automated feature selection methods.

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

2 MR ADORE Design Objectives

MR ADORE 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; and (v) user-controlled trade-offs between recognition accuracy and utilized resources (e.g., run-time).

These objectives favor the use of readily available off-the-shelf image processing operator libraries (IPLs). 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 input images to final interpretation; 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 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. 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?

3 MR ADORE Operation

MR ADORE starts with a Markov decision process (MDP) [16] as the basic mathematical model by casting the IPL operators as MDP **actions** and the results of their applications (i.e., data tokens) as MDP **states**. However, in the context of image interpretation, the formulation frequently leads to the following challenges absent from typical search settings and standard MDP formulations: (i) Standard machine learning algorithms cannot learn from raw pixel level data since the individual states are on the order of several mega-bytes each. Selecting

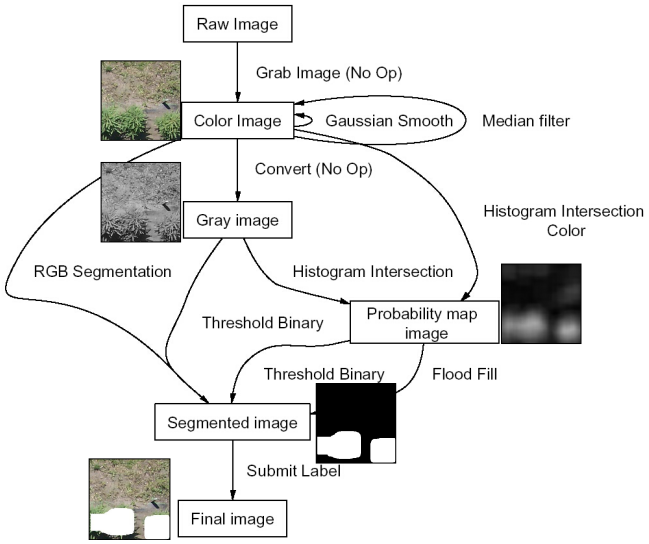


Fig. 2. Partial operator graph for the domain of forest image interpretation. The nodes and the corresponding example images depict that data processing layers, which in turn describe the *type* of MDP states present with MR ADORE. The edges represent vision routines, typically ported from the Intel OpenCV and IPL libraries, that transform one state to another (i.e., the MDP actions).

optimal features as state descriptions 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. Additionally, certain intermediate states (e.g., probability maps) have a continuous nature; (iii) In addition to the relatively high branching factor possible, due to large image processing operator libraries, some of the more complex operators may require hours of computation time each; (iv) Unlike standard search, typically used to find the shortest sequence leading to a goal state, the goal of MR ADORE is to find/produce the best image interpretation. In that respect the system solves an optimization problem rather than one of heuristic search. Therefore, since goal states are not easily recognizable as the target image interpretation is usually not known *a priori*, standard search techniques (eg IDA* [14]) are inapplicable.

In response to these challenges MR ADORE employs the following off-line and on-line machine learning techniques. First, the domain expertise is encoded in the form of training data. Each training datum consists of 2 images, the input image, and its user-annotated counterpart allowing the output of the system to be compared to the desired image labeling (typically called ground-truth). Figure 1 demonstrates a training pair for the forestry image interpretation domain. Second, during the off-line stage the state space is explored via limited depth expansions of all training image pairs. Within a single expansion all sequences of IPL operators up to a certain user-controlled length are applied to a training image. Since training images are user-annotated with the desired output, ter-

minal rewards can be computed based on the difference between the produced labeling and the desired labeling. System **rewards** are thus defined by creating a scoring metric that evaluates the quality of the final image interpretation with respect to the desired (used-provided) interpretation⋆. Then, dynamic programming methods [2] 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 (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. (See Figure 3 for an illustration of the process)

Features (f), used as **observations** by the on-line system component, represent relevant attributes extracted from the unmanageably large states (i.e., data tokens). Features make supervised machine learning methods practically feasible, which in-turn are needed to extrapolate the sampled Q-values (computed by dynamic programming on the explored fraction of the state space) onto the entire space (see Figure 4).

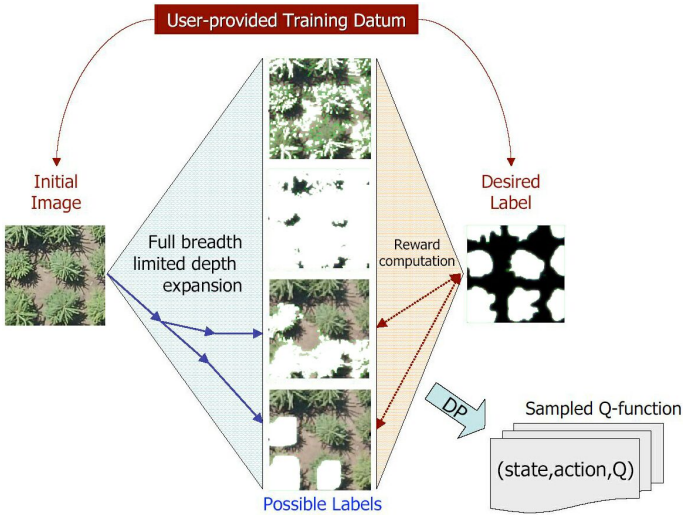


Fig. 3. Off-line operation.

Finally, when presented with a novel input image, MR ADORE exploits the machine-learned heuristic value function $Q(f(s), a)$ over the abstracted state

⋆ For all experiments presented, the intersection over union scoring metric, $\frac{A \cap B}{A \cup B}$ is used. Typically used in vision research, this pixel-based scoring metric computes the overlap between the set of hypothesis pixels produced by the system A and the set of pixels within the ground-truth image B. If set A and B are identical then their intersection is equal to their union and the score/reward is 1. As the two sets become more and more disjoint the reward decreases, indicating that the produced hypothesis corresponds poorly to the ground-truth.

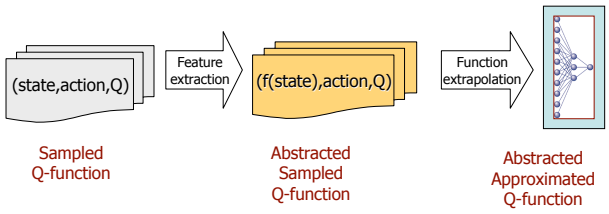


Fig. 4. Feature Extraction and Q-function approximation.

space, f(S), in order to intelligently select operators from the IPL. The process terminates when the policy executes the action `Submit(`⟨*labeling*⟩`)`, which becomes the final output of the system. (see Figure 5).

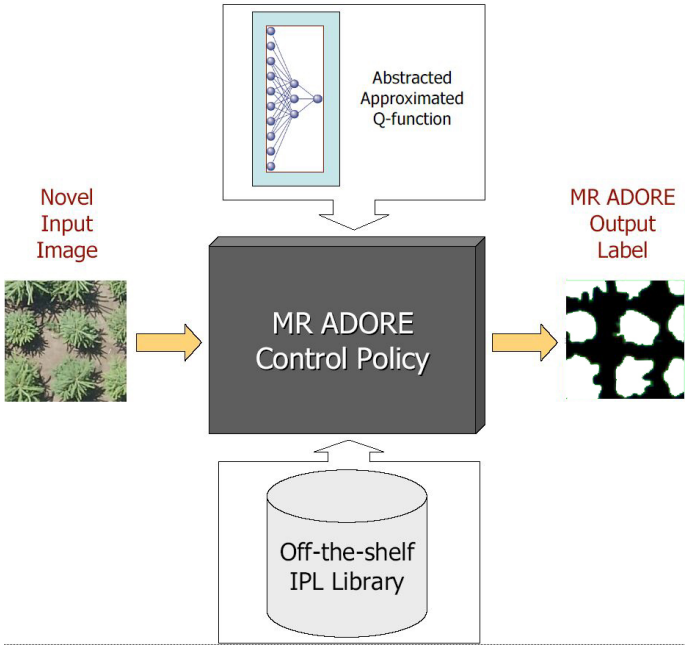


Fig. 5. On-line operation.

3.1 Learning Control Policies

The purpose of the off-line learning phase within MR ADORE is to automatically construct the on-line control policy. The policies studied to date can be conceptually represented via the following unifying framework.

First, data tokens (i.e., states) can be split into disjoint "processing levels". The initial input image is a data token of the top (first) data level and the final submitted image interpretation is a data token from the bottom (last) data level. Figure 2 illustrates six processing levels as follows: *raw image, color image, gray image, probability map image, segmented image,* and *final image.*

Consequently, the set S of data tokens can be partitioned into subsets $S_1, \ldots, S_n$ where S_i contains data tokens belonging to processing level i. Likewise, the set A of operators can be represented as a union of (disjoint) operator subsets A_i, where action $a_i \in A_i$ is applicable at level i.

Therefore, any control policy $\pi : S \rightarrow A$ can be conceptually divided into n control policies: $\pi_1, \ldots, \pi_n$ where π_i operates over S_i only, that is $\pi_i : S_i \rightarrow 2^{A_i}$. There are several different classes of π_i policies possible, including:

fixed: $\pi_i(s) = \{a_i\}$ for all states s. Such policies blindly apply the same action regardless of the data token s at hand.

full expansion: $\pi_i(s) = A_i$: all applicable operators within the set A_i are executed.

value-based: $\pi_i(s) = \{\arg\max_{a \in A_i} Q_i(s, a)\}$ where a machine-learned approximator is used for the Q_i-function. Such policies greedily select actions to be executed with the highest Q-values. This policy is specified by the machine-learned approximator and the feature selection function.

In addition to the three aforementioned types of processing level policies, two special policy types are possible within MR ADORE that do not conform to the standard policy definition presented above.

path-selector: $\pi_i(s) = a^i \in A_i, a^{i+1} \in A_{i+1}, \ldots, a^j \in A_j$: This policy selects a (sub)sequence of actions, the first of which is executed and the rest are passed on to the next processing level(s). This policy is really a derivative of the value-based policy in the sense that it too is a Q-greedy policy defined by a machine-learned function approximator and the feature selection function. The most notable difference is unlike the value-based policy, the path-selector selects a (sub)sequence of actions to execute rather than a (sub)set of actions.

follow-through: This policy $\pi_j(s)$ executes action a_j specified by a path-selector policy at a higher processing level, π_i, where $1 \leq i < j$.

Although these two policies significantly differ from the three initial polices, all five policy types can be mixed and matched to form the global policy π. (One obvious exception is that a follow-through policy must be proceeded by a path-selection policy at a higher level.)

Thus, any policy π can be described as an n-ary vector of π_i, where n is the number of processing layers. Selecting the optimal combination of π_i's type and parameters is a difficult problem. Pre-ADORE systems have solved this problem by employing domain engineering techniques in order to design π, which virtually always calls for extensive trial-and-error experiments and substantial human expertise. In [8] and [7] researchers used hand-crafted features to implement the value-based policy at all data levels (commonly known as best-first/greedy policy) within ADORE. Research in the MR ADORE project casts policy selection as a challenging open machine learning problem with the elements of meta-learning. To that end, the following three types of control policies have been explored to date.

A **static** policy, π_{static}, uses a single sequence of operators and therefore consists of only fixed π_i's. Research within the field of computer vision has

produced numerous systems using a static sequence of operators (e.g. [6]). As previously mentioned such systems require extensive trial-and-error experiments, domain expertise, etc. Since ADORE and MR ADORE perform a full-breadth search during the off-line phase, the **best** static sequence can be automatically determined from the off-line expansions of training images. Such policies, while being computationally inexpensive (on-line), generally can only achieve near-optimal performance within a narrow range of operating conditions (due to the non-adaptive nature of the policy). In fact, this is the reason why researchers initially turned to using a library of operators and adopted the "goal-directed" paradigm for object recognition.

A **"trunk-based" policy** π_{trunk} makes its only decision at the top processing level (S_1) by selecting an n-long sequence of operators based on the input image only. The policy maintains a library of *optimal*** sequences (called "trunks") collected during the off-line stage. Therefore, π_1 is a path-selector over the space of trunks while $\pi_2, \ldots, \pi_n$ are all of the follow-through type.

While best-first policies are theoretically capable of much more flexibility than static or (semi-static) trunk-based policies, they depend crucially on (i) data token features for *all* levels and (ii) adequate amounts of training data to train the Q-functions for *all* levels. Experience has shown that it is substantially harder to select features at the earlier levels where the data tokens exhibit less structure [10]. To make the problem worse, a single user-labeled training image delivers exponentially larger numbers of $\langle$ state, action, reward $\rangle$ tuples at later processing levels. Unfortunately, the first processing level gets the mere $|A_1|$ tuples per training image since there is only one data token (the input image itself). As a net result, best-first control policies have been known to backtrack frequently [8] as well as produce highly suboptimal interpretations [5], due to poor decision making at the top processing layers. In fact the trunk-based policy suffers from the same phenomenon, a lack of high-quality features and sparseness of training examples.

Rather than making control decisions at every level based on the frequently incomplete information provided by imperfect features, the **least-commitment policies** postpone their decisions until more structured and refined data tokens are derived. That is: $\pi_j = A_j$ for $1 \leq j \leq n-1$ and π_n is value-based. In other words, we apply all operator sequences up to a predefined depth and only then engage the machine-learned control policy to select the appropriate action. Doing so allows the control system to make decisions based on high-quality informative features, resulting in an overall increase in the quality increases. As a side benefit, the machine learning process is greatly simplified since feature selection and value function approximation are performed for considerably fewer processing levels while benefiting from the largest amount of training data. (Figure 6 illustrates the outlined policies.) In order to determine the merit of each policy we tested MR ADORE with static, trunk-based and least-commitment policies along with two base-line policies: random and random trunking. A random pol-

** An optimal sequence is the one yielding the greatest reward over all other sequences with respect to a given input image.

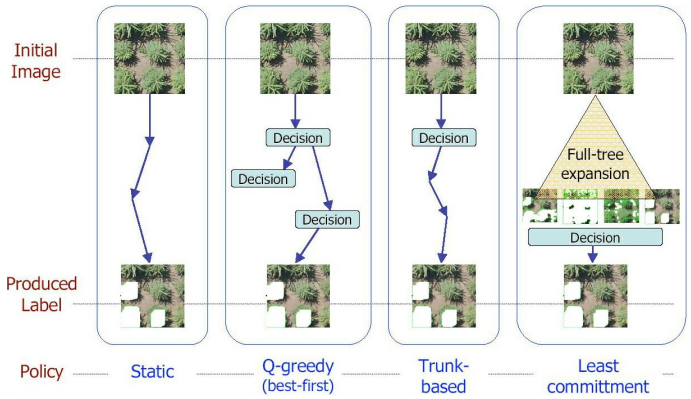
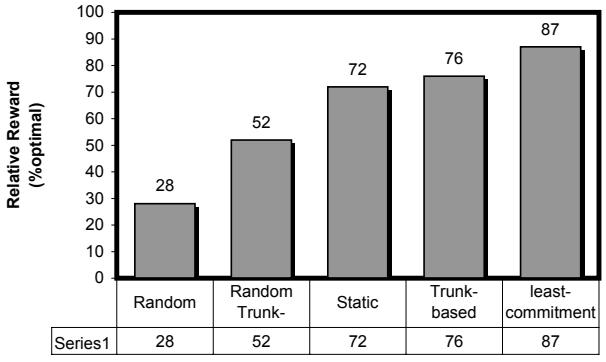


Fig. 6. Four types of on-line polices.

icy is simply a least-commitment policy that randomly assigns rewards to data tokens rather than using a machine-learned function approximator. Similarly, a random trunk-based policy is a trunk-based policy that selects an operator sequence (i.e., a trunk) at random rather than using a machine-learned function approximator to assign Q-values to each of the possible sequences. As previously mentioned, trunk-based policies suffer from lack of informative features and training examples. The performance evaluation was conducted on 35 images of forest plantations (cf. Figure1). Using a leave-one-out cross-validation strategy (i.e, 34 for training and 1 for testing), each of the five control policies was ran on the set of forestry images. Table 1 shows the experimental results.

Table 1. Performance of five different policies on the forestry data. 35 images of forest plantation scenes were used in the experiment. Results for each policy were averaged over the 35 leave-one-out cross-validation runs. **% optimal** refers to the fact that each policy is evaluated relative to the best off-line interpretation achievable given a particular IPL.



As expected, since the least-commitment policy uses a function approximator trained on data tokens at bottom processing level that contains an exponentially many more examples than preceding levels, this policy outperforms all other policies*** The trunk-based policy follows next†. As mentioned previously, the lack of information at the top processing layer clearly makes the trunk-based policy perform much worse that the least-commitment policy. The results are consistent with previous work. In [10] the researchers (informally) report that as less and less informative features are used to train the best-first policy, it converges to a static policy (i.e., the best-first policy applies the same sequence of actions regardless of input). Thus the performance of the static policy is almost as good as the best trunk-based policy. Not surprisingly the random policies perform the worst. It should be noted that the probability of performing optimally for this experiment (ie randomly choosing an optimal sequence for each image) can be easily calculated. There are 118 possible sequences for each image and 35 images, but only one optimal sequence per image. Thus $P(\pi^{optimal}) = (\frac{1}{118})^{35} \approx 3 * 10^{-72}$.

4 Future Research

One of the problems with both the trunk-based and least-commitment policies is the need for high-quality features. While both policies (trunk-based and least-commitment) reduce the need for hand-crafted features by requiring only a single set (respectively at either the top or bottom processing level), the need for a domain expert is only reduced and **not** eliminated. Principal component analysis (PCA) can be used to reduce the vast dimensionality of the raw data tokens by projecting them into an eigen space [13]. This results in an automatically generated set of features (typically one set per processing level). Lazy instance-based learning (k-nearest-neighbors) [11] can then be used to approximate the Q_i function at level i. Additionally, the features can take the state history into account insomuch as Q_i is computed on abstracted previous states: $[f_i(s_i), f_{i-1}(s_{i-1}), \ldots, f_1(s_1)]$ where $s_1, \ldots, s_i$ are the states on the path from the input image s_1 to the current image at hand, s_i. Figure 7 demonstrates the potential of using KNN with PCA-based features. An alternative to using the PCA procedure to compress the images is to simply down-sample the images to a size that can be managed by the modern machine learning methods. Unfortunately, both of these methods (PCA and down-sampling) assume that the intermediate data levels are indeed representable as images. If an intermediate processing level consists, for example, of contour points, both PCA and down-sampling cannot

*** The results presented here used an artificial neural network [12] as the function approximator and an HSV histogram for input features. It should be noted that other function approximation methods were used including k-nearest neighbors[11] and sparse networks of Winnows (SNoW) [1]. In addition a number of feature sets were tried as well.

† Here we used KNN with a number of features and distance metrics. The best results show in the table were obtained by using texture/shape features in the form of local binary patterns (lbp) together with a Euclidean distance measure.

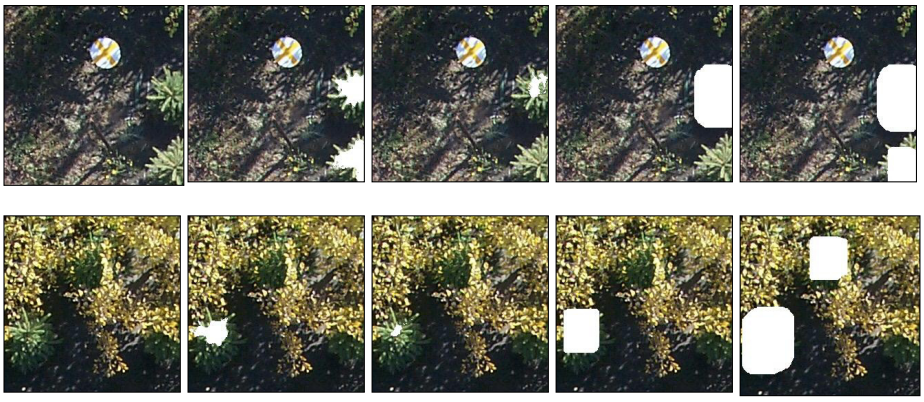


Fig. 7. Each row from left to right: the original image, desired user-provided labeling, optimal off-line labeling, best static policy of length 4 labeling, best-first policy labeling (using KNN and automatically extracted features via PCA). The adaptive policy has been observed to outperform best static policy (top row) and sometimes the human experts as well (bottom row).

be readily applied. Thus the most probable solution to the feature extraction problem will perhaps come in the form of feature extraction libraries (similar to the operator libraries). Just as MR ADORE learned efficient operator selection, next generation object recognition systems will need to *select* which features are relevant [4] for a given processing layer through off-line trail-and-error processes based, perhaps, on the very same MDP framework used today to efficiently select operator sequences.

Automated selection of optimal π_is for all processing levels is an open-ended problem. The particular difficulty lies with the tightly coupled nature of individual policies and the exponentially large space of all combinations of π_is. An initial investigation into bottom-up construction of π via filling the later levels with value-based π_is for $i = n, n-1, \ldots$ one-by-one, while leaving the earlier processing levels populated with the full expansion $\pi_j = A_j$ policies, appears promising.

5 Conclusions

Conventional ways of developing image interpretation systems often require that system developers posses significant subject matter and computer vision expertise. The resulting knowledge-engineered systems are expensive to upgrade, maintain, and port to other domains.

More recently, second-generation image interpretation systems have used machine learning methods in order to (i) reduce the need for human input in developing an image interpretation system or port it to 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 a state-of-the-art adaptive image interpretation system called MR ADORE and demonstrated several exciting machine learning

and decision making problems that need to be addressed. We then reported on the progress achieved on reducing the need for feature selection and went on to propose ideas on how to completely eliminate the need for hand-crafted features.

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 Center for Machine Learning.

References

1. S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *7th European Conference on Computer Vision*, volume 4, pages 113–130, Copenhagen, Denmark, 2002.
2. A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(1):81–138, 1995.
3. V. Bulitko, B. Draper, D. Lau, I. Levner, and G. Zhang. MR ADORE : Multiresolution adaptive object recognition (design document). Technical report, University of Alberta, 2002.
4. Vadim Bulitko, Greg Lee, and Ilya Levner. Evolutionary algorithms for operator selection in vision. In *Proceedings of the Fifth International Workshop on Frontiers in Evolutionary Algorithms*, 2003.
5. Vadim Bulitko and Ilya Levner. Improving learnability of adaptive image interpretation systems. Technical report, University of Alberta, 2003.
6. Michael C. Burl, Lars Asker, Padhraic Smyth, Usama M. Fayyad, Pietro Perona, Larry Crumpler, and Jayne Aubele. Learning to recognize volcanoes on venus. *Machine Learning*, 30(2-3):165–194, 1998.
7. B. Draper, U. Ahlrichs, and D. Paulus. Adapting object recognition across domains: A demonstration. In *Proceedings of International Conference on Vision Systems*, pages 256–267, Vancouver, B.C., 2001.
8. B. Draper, J. Bins, and K. Baek. ADORE: adaptive object recognition. *Videre*, 1(4):86–99, 2000.
9. B. Draper, A. Hanson, and E. Riseman. Knowledge-directed vision: Control, learning and integration. *Proceedings of the IEEE*, 84(11):1625–1637, 1996.
10. Bruce A. Draper. From knowledge bases to Markov models to PCA. In *Proceedings of Workshop on Computer Vision System Control Architectures*, Graz, Austria, 2003.
11. Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.
12. S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillian College Pub. Co., 1994.
13. Michael Kirby. *Geometric Data Analysis: An Emprical Approach to Dimensionality Reduction and the Study of Patterns*. John Wiley & Sons, New York, 2001.
14. Richard E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2-3):189–211, 1990.
15. R. Rimey and C. Brown. Control of selective perception using bayes nets and decision theory. *International Journal of Computer Vision*, 12:173–207, 1994.
16. R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2000.