Genetic Programming for Automatic Target Classification and Recognition in Synthetic Aperture Radar Imagery

Stephen A. Stanhope
ERIM International
Advanced Information Systems Group
P.O. Box 134008
Ann Arbor, Michigan 48113-4008 USA
stanhope@erim-int.com

Jason M. Daida
The University of Michigan
Artificial Intelligence Laboratory & Space Physics Research Laboratory
2455 Hayward Avenue
Ann Arbor, Michigan 48109-2143 USA
daida@eecs.umich.edu

Abstract. We use the genetic programming (GP) paradigm for two tasks. The first task given a GP is the generation of rules for the target / clutter classification of a set of synthetic aperture radar (SAR) images, the second, the generation of rules for the identification of tanks in a second set of SAR images. To perform these tasks, previously defined feature sets are generated on the various images, and GP is used to select relevant features and methods of analyzing these features. GP results are then compared with previous work using the feature sets.

1. Introduction

One of the first uses of the GP paradigm for image classification appeared in Tackett [17], where GP was used to generate arithmetical expressions representing features for target / clutter classification on a set of infrared imaged military vehicles. While this work was useful from an automatic target recognition (ATR) standpoint, it was unclear whether Tackett's particular technique would work for other types of imagery or classification problems, as noted by Laird [11].

The question of GP's utility in image feature extraction problems has been, since this early work, partially tested by many other researchers. Daida [3,4], for example, used GP to generate arithmetical expressions for the purpose of extracting ridge features from SAR imaged arctic ice. Another set of examples lie in the facial feature extraction algorithms using GP implemented by Isaka [10] and Winkeler [18]. One interesting aspect of many of these works is the fact that they generally concentrated on the feature extraction aspect of image classification, rather than feature selection (although the two are closely related). In many problems (such as ATR problems), however, a large set of features is often available via prior analysis, and, rather than further feature extraction, feature selection and classification algorithm generation is necessary. In this area, although there has been much applied work done using genetic algorithms (GAs), e.g. Sims [16], little has been done using GP. In this work, we hope to further demonstrate the potential utility of the GP paradigm for feature selection and classification algorithm generation in the ATR domain.

To show the utility of the GP paradigm for feature selection and classification algorithm generation in the ATR domain, we focus on two problems. The first problem is a simple target / clutter image classification problem. The second problem is an actual target identification problem. Each problem uses a set of SAR-imaged military targets, and rather than using GP to generate features for the classification and identification of these images, we use a previously defined feature set shown to have utility for classification and identification in the SAR domain. The role of the GP is to generate logical expressions on comparisons of these features to both real-valued constants and themselves, in order to create a linear classifier of the style described in Fukunaga [7]. Resulting expressions are then compared to previously known information about the feature set in terms of both classification / identification performance and methodology, to obtain an estimate of the performance of the GP.

2. Experiment One - Target / Clutter Classification

In the first experiment, GP was used to classify a set of SAR images into those containing a man-made military target, and those containing only clutter. We first describe the SAR data set used, as well as the features generated on this data set. Second, we describe the procedure used, including how the GP was used to generate rules on the full feature set. Last, we evaluate the results of the GP, and contrast these results with previous knowledge of the characteristics of the feature set.

2.1 Data Set

The data set used in the first experiment was the DARPA ADTS (Advanced Detection Technology Sensor) standard data set [13]. This data set consisted of a large set of small SAR-imaged "chips" (128x128 pixels), each of which contained either a target with or without camouflage, or natural clutter such as trees, rocks, etc. An example of each is given below (Images 1 and 2). Other than these examples, we will not discuss the technical details of SAR here (see [2]), except to note that it is a type of imaging radar differing from infrared imagery in that it works from the microwave, as opposed to the infrared portion of the electromagnetic spectrum. This large set of images was then divided into a large training set (1447 images) and a smaller set (304 images) to be used for testing the learned classification rules. Feature vectors, as discussed in the next section, were then generated on each image, and these feature vectors were sent to the GP for processing.

Image 1: SAR-Imaged Clutter

Image 2: SAR-Imaged Target

2.2 Feature Set

A set of ten separate features was used to generate feature vectors on each individual chip mentioned above. The 10 features used were taken from a larger set of 15 features fully investigated by MIT Lincoln Laboratory (MIT/LL) [12] and are described in Table 1 below. While the use of all 15 features in the set would have been desirable, several of them were discarded due to a lack of necessary data on each image. After generating each feature i on each chip x in both the training data set and test data set, each feature vector x_i was normalized by the mean μ_{xi} and standard deviation σ_{xi} of the feature vectors generated on the training data to produce features i with $\mu_i = 0$ and $\sigma_i = 1$. These final, normalized feature vectors were then passed to the GP for processing.

Table 1: Feature set used in target / clutter classification experiment.

Feature	Description
Blob Mass	Calculates the number of pixels in a morphological blob defined on the suspected target.
Blob Diameter	Returns length of diagonal of the smallest rectangle bounding the morphological blob.
Blob Inertia	The second moment of the blob amplitudes about their center of mass.
Contrast Max	Returns the largest pixel value in the morphological blob from a clutter-normalized ($\mu_{clutter} = 0$ and $\sigma_{clutter} = 1$) image.
Contrast Mean	Returns the mean pixel value in the morphological blob from a clutter-normalized image.
Contrast Bright	Returns the number of pixels greater than one standard deviation above the mean in the morphological blob from the normalized image.
Standard Deviation	The standard deviation of the pixel amplitudes contained in the morphological blob.
Fractal Dimension	A measure of the spatial dimensionality of the pixels contained in the morphological blob.
Count	The number of pixels in the morphological blob with amplitudes greater than the 98th percentile of the surrounding clutter.
Weighted-Rank Fill Ratio	Computed by obtaining quotient of the sum of the highest-valued 50 pixels in the morphological blob, and the sum of all the pixels in the blob.

All of the above features were computed using a morphological "blob" about the target. This refers to the set of pixels in a image that represent the shape of the target, obtained through a series of image algebra operations such as defined by Crimmins [1].

2.3 Procedure

After obtaining the normalized feature vectors on each individual chip, the full set of normalized feature vectors was then passed to the GP for analysis. The GP worked in the following manner: Each individual consisted of a binary tree with a maximum depth of 7, representing a logical statement on the feature vectors. If the statement returned a value of 1 (TRUE), the image represented by the feature vector was assumed to contain a target, otherwise, the image was assumed to contain only clutter. The fitness f for a given individual i was then calculated as $f(i) = \frac{\#\ correct\ classifications}{\#\ images}$. In this case, the fitness of an individual corresponds to the probability of correct classification of images into target / clutter classes as defined by Ratches [15]; the goal of the GP is to maximize this probability.

Each node of an individual was allowed to be of four types: `Logicals`, returning a 1 or 0; `Comparators`, returning a 1 or 0; `Terminators`, referencing a feature, and `Constants`, returning a specified value sampled from a Gaussian distribution with $\mu = 0$ and $\sigma = 1$. For each type, there were multiple possible operators. The `Logical` type consisted of the operators {AND, OR}, the `Comparator` type consisted of the operators {<, >}, `Terminators` consisted of operators referencing any of the ten possible features, and `Constants` had some real value. `Logicals` and `Comparators` were allowed exactly two child nodes, while `Terminators` and `Constants` were allowed none. Further, the child nodes allowed to each node type were limited. `Logicals` were allowed to have child nodes of type `Logical` or `Comparator`, while `Comparators` were forced to have the left child node of type `Terminator`, and the right node of either type `Terminator` or type `Constant`. For a summary of this information, see Table 2 below.

Table 2: Summary of GP node type information.

Node Type	Return Value	Operators	Legal Child Types
`Logical`	{1, 0}	{AND, OR}	`Logical, Comparator`
`Comparator`	{1, 0}	{>, <}	1 `Terminator`, 1 `Terminator` or `Constant`
`Terminator`	Real	See Table 1	None
`Constant`	Real	Given a real value	None

The genetic operators worked in the following manner: The mutation operator, applied to a node, would alter that node to a random operator (or value, in the case of a node of type `Constant`) of the same type. Crossover between individuals resulted in one child, and was allowed to occur at nodes if two criteria, ensuring both consistent individuals and the tree depth restrictions, were met. First, the node types at the point of crossover had to have consistent return values (i.e. a crossover point at node types `Logical` and `Comparator` would be legal, while a crossover point at node types `Logical` and `Terminator` would not). Second, the crossover node depth of the second parent needed to be equivalent to or greater than the node depth of the crossover node of the first parent. Given these restrictions on the genetic operators used by the GP, it would be possible to classify this GP as being "strongly typed", as described by Montana [14] and Harris [8]. Additionally, this procedure is similar to other GP procedures that induce decision trees for classification purposes, such as recent work by Frietas [6].

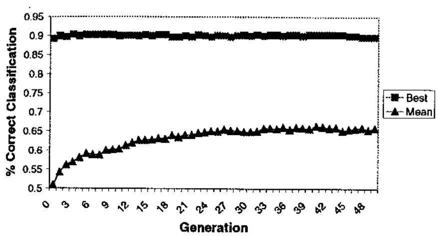
The GP used a population size of 250 over 100 generations. The probability of generating a new individual via crossover was set to $p_C = 0.60$, while the probability of mutating a specific node was set to $p_M = 0.25$. The use of these specific settings were a result of performance testing on a limited set of data, on which they worked well. No elitism was used, and a proportional selection scheme was used to select individuals for propagation into the next generation. Twenty runs were made, and for each run in this set, fitness information was gathered for each individual processed on the testing data and on the out-of-sample data. Further, information was collected over all 20 runs, and the average mean and best fitness of individuals generated by the GP over these 20 runs were computed on both the training and the out-of-sample data up to generation 50. Beyond generation 50, no significant statistical improvement in the fitness of individuals generated by the GP was seen.

2.4 Results and Discussion

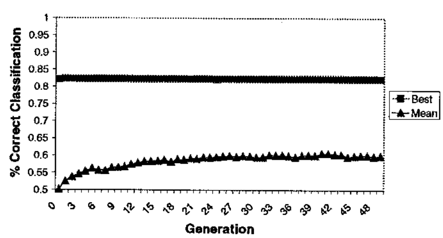
Analysis of the performance of the GP was based on two measures of performance. The first was how well the GP performed in terms of accuracy of classification. The second was how well the features which the GP found useful corresponded to prior analysis performed by MIT/LL against the full feature set, which characterized the most useful features for target / clutter classification problems [12].

In terms of accuracy of classification, the system was able to generate rules that performed reasonably well on both the training data set and the out-of-sample testing data set, as seen in Graphs 1-2. In this particular test, the best out-of-sample performance scores, giving approximately 82% correct classification, compared to results generated by MIT/LL using this feature set, in which targets could be detected correctly in approximately 75-85% of the cases evaluated. The quoted figures for the MIT/LL experiments resulted roughly 1-10 false alarms per square kilometer, as shown in ROC (Receiver Operating Characteristic) curves from that study. No such statistic exists for the GP.

Graph 1: GP Performance on Training Data - Classification



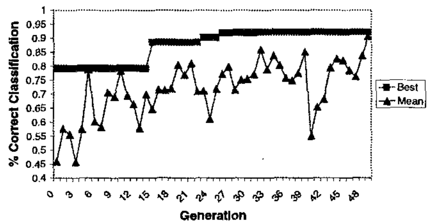
Graph 2: GP Performance on Out-of-Sample Data - Classification



One of two possible issues that could be raised is that of how well classification accuracy in the training data set implies classification accuracy in the out-of-sample data set. There was some tendency of the GP to overfit to the training data, seen most often in those rules with classification accuracy approaching 90% in the training data. However, rules that gave classification accuracy of 70-80% in the training data showed minimal degradation in moving to the out-of-sample data set.

The second issue that might be raised deals with the fact that there is seemingly a lack of improvement of the performance of the best rules found over time. There are three potential reasons for this phenomenon. The first is the relative ease of the problem, in that the problem combined a large population with a small set of features. The second is the fact that elitism was not used, and the third is that the method used in visualizing the results of the GP may have smoothed out any variance encountered in individual runs. To resolve whether or not the GP had any learning capability, a single run was made, in which the population size was lowered to 10 individuals and elitism was used to propagate the best two rules found from generation to generation explicitly. The best and mean performances of this run are given in Graph 3; in this case, significant improvements were made in the solutions initially proposed by the GP.

Graph 3: GP Performance with Small Population - Classification



Given the performance of the GP in terms of accuracy, the next measure of performance was how well the features found to be useful by the GP matched against the features revealed to be useful by the previous analysis done by MIT/LL. To answer this question, we first (Table 3) give three examples of rules found by the GP, which are characteristic of the types of rules that the GP generates. Then, in Table 4, the features noted by MIT/LL as providing best discrimination power and the five features most commonly found by GP-generated rules with high classification accuracy both in the training data set and the out-of-sample data set are given. There is some overlap in the features found to be useful by MIT/LL and the features that the GP commonly found to be useful, which might imply that the GP has performed well in this aspect.

Table 3: Sample GP-Generated Rules for Target / Clutter Classification

`(Blob_Diameter > 0.003747)`
`(Count > 0.043725)`
`((Contrast_Max > Blob_Diameter) AND (Contrast_Mean < Contrast_Max)) OR (Count > Standard_Deviation)`

Table 4: Comparison of Features Commonly Used by GP to Lincoln Laboratory Analysis

MIT/LL Best Features	GP Most Common Features
Blob Diameter	Blob Diameter
Fractal Dimension	Fractal Dimension
Contrast Mean	Contrast Mean
Contrast Bright	Contrast Max
Weighted-Rank Fill Ratio	Standard Deviation

2.5 Conclusion - Experiment One

Experiment One tested the GP defined in Section 2.3 in a target / clutter classification problem, using a previously defined feature set. The performance of the GP was compared to prior analysis done on this feature set by MIT / Lincoln Laboratory, both in terms of general classification accuracy and in terms of finding the usefulness of features that were previously shown to be useful. This comparison showed similarities in both accuracy and features found to be useful.

3. Experiment Two - Target Identification

In the second experiment, GP was used to identify those images containing a tank signature from a set of SAR images, all containing signatures of military targets. We first describe the data set used and the features generated on this data set. Second, we describe the procedure used, and last, we evaluate the results of the GP, and contrast these results with previous knowledge of the characteristics of the feature set.

3.1 Data Set

The data set using in the second experiment consisted of 15 vehicles, of which 5 were tanks, the remaining 13 being considered confuser targets, imaged with a SAR sensor at rotation increments of every 30° from 0°. Additionally, at each 30° increment, every target was imaged ± 5°, at increments of 0.5°. Specific imagery from this data set will not be shown; it is similar in appearance to the data set used in Experiment One. This resulted in 3755 individual images, of which 1258 were images of tanks. This large set was then randomly divided into two groups, a training group consisting of 3004 images (1007 of which were tanks), and an out-of-sample set consisting of 751 images (251 of which were tanks). Feature vectors, as discussed below, were generated for each image, and the resulting feature vectors were sent to the GP for processing.

3.2 Feature Set

For the second experiment, two more features were added to the feature set used in the first experiment (discussed in Section 2.2 above). The first additional feature was the result of a correlation-based template match against a tank. The second was the result of the same correlation-based template matcher against a different confuser target. The correlation template-matching algorithm used is described briefly below. These features were chosen for addition because the details of the use of this matching algorithm in this data set has been well characterized in research conducted at ERIM International (EI), thereby giving us something to compare the results of the GP against. Again, the feature vectors x_i generated were scaled by the mean μ_{xi} and standard deviation σ_{xi} of the feature vectors of the training data set to produce features i with $\mu_i = 0$ and $\sigma_i = 1$; these final scaled data were used to train the GP.

The correlation-based template matching algorithm worked in the following manner: Given a target image X, and a smaller template image Y, the matching algorithm slides the template image Y to each possible position in the target image X where it fits. At each position, the algorithm then computes a correlation of the pixel amplitudes in image Y and the pixel amplitudes in the subimage of X covered by Y. The best (closest to unity) correlation score obtained over all fits of Y in X is regarded as the correlation of X and Y.

3.3 Procedure

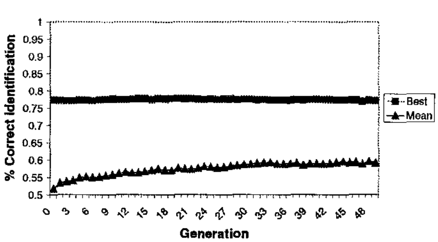
The procedure for the second experiment was the same as the procedure for the first (Section 2.3), except for the addition of the two correlation metrics to the set of possible terminators.

3.4 Results and Discussion

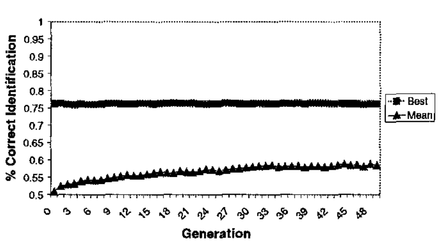
As in the analysis of Experiment One, in Experiment Two we were concerned with two measures of performance. The first measure was the performance of the GP in terms of accuracy of identification. The second was how well the features that the GP found to be useful correspond with what we know previously to be useful (the template matching score against the template of the tank).

In terms of accuracy of identification, the system was again able to generate rules that performed well on both the training data set and the out-of-sample data set, as seen in Graphs 4 and 5 below. The best out-of-sample performance of rules were typically 76% correct classification, although some rules reached approximately 80% correct classification. This is worse performance than noted by the EI study which obtained 100% correct classification with 10% misclassification in a theoretical study, and could be partially attributed to the fact that the GP did not explicitly refine real-valued thresholds used for comparisons against features. The slight tendency of the GP to overfit to the training data persisted, and like Experiment One, while a 10% improvement over time was seen in the average mean performance of the algorithm, no improvement over time was seen in the average best performance. To again test the GP in order to find out if the algorithm had any real learning capability, the same technique used in the target / clutter classification problem was again used. A single run was made, in which the population size was lowered to 10 individuals and elitism was used. The best and mean performances of this run are given in Graph 6; as in the target / clutter classification problem, significant improvements were made in the solutions initially proposed by the small-population GP.

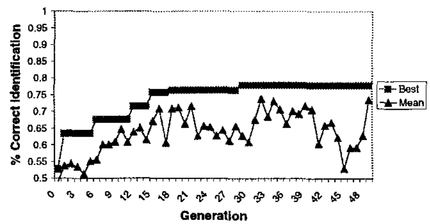
Graph 4: GP Performance on Training Data - Identification



Graph 5: GP Performance on Out-of-Sample Data - Identification



Graph 6: GP Performance with Small Population - Identification



Again, given the overall performance of the GP in terms of identification accuracy, the next measure of performance was how well the features found to be useful by the GP corresponded with the features previously known to be useful. The primary feature used by the GP was the correlation score against the tank template, as expected, which was good enough for over 75% identification accuracy. However, the GP was able to use the other features, especially the correlation score against the confuser target, to further refine its accuracy up to the aforementioned 80% accuracy. Two sample rules are given below (Table 5), which again are simple but representative of rules that the GP generated. In the first, we see a direct numerical comparison to the correlation metric against the tank, in the second we see this correlation metric used in conjunction with the correlation metric against the template of the confuser target and the contrast max feature. It should be noted, however, that after the feature normalization process, the value of the (normalized) contrast max feature was virtually zero among all targets, due to a few outliers in the data. Based on this knowledge, the second rule could be construed as identifying a target as a tank if its normalized correlation score against the tank template was both greater than zero and greater than the target's normalized correlation score against the template of the confuser target.

Table 5: Sample GP-Generated Rules for Tank / Confuser Identification

`(CORR_TANK > 2.837092)`
`((CORR_CONFUSER < CORR_TANK) AND (CORR_TANK > CONTRAST_MAX))`

Note that the above comparisons again use normalized correlations, making the comparison of a correlation match to some threshold greater than 1.0 reasonable.

3.5 Conclusion - Experiment Two

Experiment Two tested the GP defined in Section 3.3 in a target identification problem, using a previously defined feature set. In terms of identification accuracy, the GP apparently had worse performance than the best possible performance noted in a prior study. This could be partially attributed to the fact that there was no explicit tuning of numerical thresholds used for comparison. However, the GP was able to find features that were previously known to be useful, and was able to combine features in order to increase its performance.

4. Conclusion

In this work, we performed two experiments. The goal of these experiments was to gain insight into the uses of GP for feature selection and classification algorithm generation in an applied problem. The first experiment was to use GP to generate logical expressions using a set of previously defined features for the purpose of classification of a set of SAR-imaged military targets into target / clutter classes. The second was to use GP to generate logical expressions on an expanded feature set for the purpose of target identification, again on a set of SAR-imaged military targets. Both experiments yielded expressions with some level of classification utility, and further, used metrics similar to what had been previously determined to be useful. Further work in this area will be in both the areas of extending the current algorithm (especially useful would be the incorporation of an explicit hill-climbing operation, as discussed by Hooper [9] and Esparcia-Alcazar [5]) and using the algorithm for the generation of multi-class classifiers.

Acknowledgments

The authors would like to thank the following people: Eric Keydel, Ron Carpinella, and Jens Wessling for their support, and Thomas Bersano, Robert Bertram, Catherine Grasso, and John Polito for their discussion. This work was funded under an internal research and development contract awarded by ERIM International of Ann Arbor, Michigan.

Bibliography

[1] Crimmins, T. and Brown, W. 1985. Image Algebra and Automatic Shape Recognition. IEEE Transactions on Aerospace and Electronic Systems. 21(1) 60-68.

[2] Cutrona, L. J. 1970. Synthetic Aperture Radar. In Skolnik, M. I. (editor). *Radar Handbook*. New York, NY: McGraw-Hill. Chapter 23.

[3] Daida, J. M. et al. 1996. Algorithm Discovery Using the Genetic Programming Paradigm: Extracting Low-Contrast Curvilinear Features from SAR Images of Arctic Ice. In Angeline, P. and Kinnear, K. (editors). *Advances in Genetic Programming 2*. Cambridge, MA: The MIT Press. Pp. 417-442.

[4] Daida, J. M. et al. 1995. Extracting Curvilinear Features from SAR Images of Arctic Ice: Algorithm Discovery Using the Genetic Programming Paradigm. In Stein, T. (editor). *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*. New York, NY: IEEE Press. Pp. 673-675.

[5] Esparcia-Alcazar, A. and Sharman, K. 1997. Learning Schemes for Genetic Programming. In Koza, J. et al. (editors). *Genetic Programming 1997*. San Francisco, CA: Morgan-Kaufmann. Pp. 57-65.

[6] Frietas, A. 1997. A Genetic Programming Framework for Two Data Mining Tasks: Classification and Generalized Rule Induction. In Koza, J. et al. (editors). *Genetic programming 1997*. San Francisco, CA: Morgan-Kaufmann. Pp. 96-101.

[7] Fukunaga, K. 1972. *Introduction to Statistical Pattern Recognition*. New York, NY: Academic Press, Inc.

[8] Harris, C. 1997. Strongly Typed Genetic Programming To Promote Hierarchy Through Explicit Syntactic Constraints. In Koza, J. (editor). *Genetic Programming 1997*. San Francisco, CA: Morgan-Kaufmann. Pp. 72-80.

[9] Hooper, D et al. 1997. Recombinative Hill-Climbing: A Stronger Search Method for Genetic Programming. In Koza, J. (editor). *Genetic Programming 1997*. San Francisco, CA: Morgan-Kaufmann. Pp. 174-179.

[10] Isaka, S. 1997. An Empirical Study of Facial Image Feature Extraction by Genetic Programming. In Koza, J. (editor). *Genetic Programming 1997*. San Francisco, CA: Morgan-Kaufmann. Pp. 93-99.

[11] Laird, P. and Saul, R. 1994. Automated Feature Extraction for Supervised Learning. *1994 IEEE Conference on Evolutionary Computation*. Piscataway, NJ: IEEE Press. Pp. 674-679.

[12] Kreithen, D. et al. 1993. Discriminating Targets from Clutter. *MIT Lincoln Laboratory Journal*. 6(1) 25-52.

[13] MIT Lincoln Laboratory ADTS Testkit. http://www.mbvlab.wpafb.af.mil. MBVLab Data Homepage, Wright Patterson AFB, Dayton, OH.

[14] Montana, D. 1995. Strongly Typed Genetic Programming. *Evolutionary Computation*. 3(2) 199-230.

[15] Ratches, J. et al. 1997. Aided and Automatic Target Recognition Based Upon Sensory Inputs From Image Forming Systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19(9) 1004-1019.

[16] Sims, S. and Dasarathy, B. 1992. Automatic target recognition using a passive multisensor suite. *Optical Engineering*. 12(31) 2584-2593.

[17] Tackett, W. 1993. Genetic Programming for Feature Discovery and Image Discrimination. In Forrest, S. (editor). *Proceedings of the Fifth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann. Pp. 303-309.

[18] Winkeler, J. and Manjunath, B. 1997. Genetic Programming for Object Detection. In Koza, J. et al. (editor). *Genetic Programming 1997*. San Francisco, CA: Morgan-Kaufmann. Pp. 331-335.