Discovering Operators and Features for Object Detection

Yingqiang Lin and Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside, CA, 92521, USA
{yqlin, bhanu}@cris.ucr.edu

Abstract

In this paper, we learn to discover composite operators and features that are evolved from combinations of primitive image processing operations to extract regions-of-interest (ROIs) in images. Our approach is based on genetic programming (GP). The motivation for using GP is that there are a great many ways of combining these primitive operations and the human expert, limited by experience, knowledge and time, can only try a very small number of conventional ways of combination. Genetic programming, on the other hand, attempts many unconventional ways of combination that may never be imagined by human experts. In some cases, these unconventional combinations yield exceptionally good results. Our experimental results show that GP can find good composite operators to effectively extract the regions of interest in an image and the learned composite operators can be applied to extract ROIs in other similar images.

1. Introduction

Object detection is an important intermediate step to object recognition. The task of object detection is to locate and extract regions from an image that may contain potential objects. These regions are called regions-of-interest (ROIs) or object chips. The quality of object detection is dependent on the kind and quality of features extracted from an image. There are many kinds of features that can be extracted. The question is what are the appropriate features or how to synthesize features, particularly useful for detection, from the primitive features extracted from an image. The answer to these questions is largely dependent on the intuitive instinct, knowledge, previous experience and even the bias of human image experts.

In most applications, an image expert designs an approach to extract ROIs from images. The approach can often be dissected into some primitive operations on the original image or a set of related feature images obtained from the original one. It is the expert who, relying on his/her rich experience, figures out a smart way to combine these primitive operations to achieve good results. The task of finding a good approach is equivalent to finding a good point in the search space of *composite operators* formed by the combination of primitive operators. The number of ways of combining primitive operators is almost infinite. The human expert can only try a very limited number of combinations and typically only the conventional ways of combination are tried. However, a GP may try many unconventional ways of combining primitive operations that may never be imagined by human experts. The inherent parallelism of GP and the speed of computers allow the portion of the search space explored by GP to be much larger than that by human experts. The search performed by GP is guided by the goodness of composite operators in the population. As the search goes on, GP will gradually shift the population to the portion of the space containing good composite operators.

Genetic programming (GP), an extension of genetic algorithm, was first proposed by Koza [1]. In GP, the individuals can be binary trees, graphs or some other complicated structures of dynamically varying size. Poli [2] used GP to develop effective image filters to enhance and detect features of interest or to build pixel-classification-based segmentation algorithms. Stanhope and Daida [3] used GP for the generation of rules for target/clutter classification and rules for the identification of objects. Howard et al. [4] applied GP to automatic detection of ships in low-resolution SAR imagery using an approach that evolves detectors. Roberts and Howard [5] used GP to develop automatic object detectors in infrared images.

In this paper, we use GP to generate composite operators to perform object detection. The basic approach is to apply a composite operator on the original image or primitive feature images generated from the original one. Then the output image of the composite operator (also called composite feature) is segmented to obtain a binary image or mask to extract the region containing the object from the original image. The individuals in our GP based learning are composite operators represented by binary trees whose internal nodes represent the pre-specified primitive operators and the leaf nodes represent the original image or the primitive feature images. The primitive feature images are pre-determined, and they are not the output of the pre-specified primitive operators. Unlike the work of Stanhope and Daida [3] and Howard et al. [4], the

1051-4651/02 $17.00 © 2002 IEEE

input and output of each node of the tree in our system are images, not real numbers. The primitive operators and primitive features in our system are very basic and domain-independent, not specific to a kind of imagery. Thus, our system can be applied to a wide variety of images.

2. Technical approach

In our GP based approach, individuals are composite operators, which are represented by binary trees. The search space of GP is the space of all possible composite operators. The space is very large. To illustrate this, consider only a special kind of binary tree, where each tree has exactly 30 internal nodes and one leaf node and each internal node has only one child. For 17 primitive operators and only one primitive feature image, the total number of such trees is 17^{30}. It is extremely difficult to find good composite operators from this vast space unless one has a smart search strategy.

2.1. Design considerations

There are five major design considerations, which involve determining the set of terminals, the set of primitive operators, the fitness measure, the parameters for controlling the run, and the criterion for terminating a run.

- **The set of terminals.** The set of terminals used in this paper are seven primitive feature images: the first one is the original image; the others are mean and standard deviation images obtained by applying templates of sizes 3×3, 5×5 and 7×7. These images are the input to the composite operators. GP determines which operations are applied on them and how to combine the results. To get the mean image, we translate the template across the original image and use the average pixel value of the pixels covered by the template to replace the pixel value of the pixel covered by the central cell of the template. To get the standard deviation image, we compute the square root of the absolute pixel value difference between the pixel in the original image and its corresponding pixel in the mean image.

- **The set of primitive operators.** A primitive operator takes one or two images as input image(s), performs a primitive operation on them and stores the result in a resultant image. Currently, 17 primitive operators are used by GP to compose composite operators.

In the following, A and B are images of the same size, c is a constant. For operators such as ADD_OP, SUB_OP, MUL_OP, etc that take two images as input, the operations are performed on the pixel-by-pixel basis.

1. ADD_OP: A + B. Add two images.
2. SUB_OP: A – B. Subtract image B from image A.
3. MUL_OP: A * B. Multiply images A and B.
4. DIV_OP: A / B. Divide image A by image B (If the pixel in B has value 0, the corresponding pixel in the resultant image takes the maximum pixel value in A).
5. MAX2_OP: A max B. The pixel in the resultant image takes the larger value of pixels in images A and B.
6. MIN2_OP: A min B. The pixel in the resultant image takes the smaller value of pixels in images A and B.
7. ADD_CONST_OP: A + c. Increase pixel value by c.
8. SUB_CONST_OP: A - c. Decrease pixel value by c.
9. MUL_CONST_OP: A * c. Multiply pixel value by c.
10. DIV_CONST_OP: A / c. Divide pixel value by c.
11. SQRT_OP: sqrt(A). For each pixel p with value v, if v ≥ 0, change its value to $\sqrt{v}$. Otherwise, to $-\sqrt{-v}$.
12. LOG_OP: log(A). For each pixel p with value v. if v ≥ 0, change its value to log(v). Otherwise, to –log(-v).
13. MAX_OP: max(A). Replace the pixel value by the maximum pixel value in a 3×3, 5×5 or 7×7 neighborhood.
14. MIN_OP: min(A). Replace the pixel value by the minimum pixel value in a 3×3, 5×5 or 7×7 neighborhood.
15. MED_OP: med(A). Replace the pixel value by the median pixel value in a 3×3, 5×5 or 7×7 neighborhood.
16. MEAN_OP: mean(A). Obtain mean image of image A by applying a template of size 3×3, 5×5 or 7×7.
17. STDV_OP: stdv(A). Obtain standard deviation image of image A by applying a template of size 3×3, 5×5 or 7×7.

- **The fitness measure.** The fitness value of a composite operator is computed in the following way. Suppose G and G' are foregrounds in the ground truth image and the resultant image of the composite operator respectively. Let $n(X)$ denote the number of pixels within region X, then $Fitness = n(G \cap G') / n(G \cup G')$. The fitness value is between 0 and 1. If G and G' are completely separated, the value is 0; if G and G' are completely overlapped, the value is 1. It is worth noting that ground truth image is used only in training.

- **Parameters and termination.** The key parameters are the population size M, the number of generations N, the crossover rate and the mutation rate. The GP stops when it finishes the pre-specified number of generations or when the best composite operator in the population has fitness value greater than the fitness threshold.

2.2. Reproduction, crossover and mutation

The GP searches the composite operator space to generate new composite operators and gradually adapts the population of composite operators from generation to generation to improve the overall fitness of the whole population. More importantly, GP may find an exceptionally good composite operator during the search. The search is done by performing reproduction, crossover and mutation operations. The initial population is randomly generated and the fitness of each individual is evaluated.

- **Reproduction.** The reproduction operation involves selecting composite operators from the current population to form a new population. In this paper, we use tournament selection, where a number of individuals are randomly selected from the current population and the one with the highest fitness value is copied into the new population.

- **Crossover.** To perform crossover, two composite operators are selected on the basis of their fitness values. These two composite operators are called parents. One internal node in each of these two parents is randomly selected, and the two subtrees with these two nodes as root are exchanged between the parents. In this way, two new composite operators, called offspring, are created.

- **Mutation.** In order to avoid premature convergence, mutation is introduced to randomly change the structure of some individuals to help maintain the diversity of the population. The candidate composite operator for mutation is selected at random. Once a composite operator is selected to perform mutation operation, an internal node of this composite operator is randomly selected. Then the subtree rooted at this node, including the node selected, is replaced by another randomly generated binary tree. Finally, the resulting new composite operator replaces the old one in the population.

2.3. Steady-state and generational GP

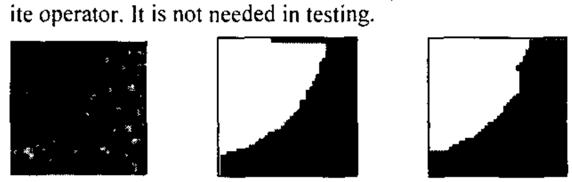
In *steady-state GP*, two parental composite operators are selected on the basis of their fitness for crossover. The children of this crossover, perhaps mutated, replace a pair of composite operators with the smallest fitness values. The two children are executed immediately and their fitness values are recorded so that they can participate in the following crossover operations if necessary. This process is repeated until crossover rate is satisfied. In *generational GP*, two composite operators are selected on the basis of their fitness values for crossover and generate two offspring. The two offspring are not put into the current population and won't participate in the following crossover operations on the current population. The above process is repeated until crossover rate is satisfied. Then, all the offspring from the crossover are evaluated and combined with the current population to form an enlarged population. Finally, reproduction operation is applied to the enlarged population to produce a new population of the same size as the old one. In addition, we adopt an elitism replacement method that copies the best composite operator from generation to generation.

3. Experiments

Various experiments were performed to test the efficacy of genetic programming in extracting regions of interest from real synthetic aperture radar (SAR) images. Unlike in our previous paper [6], here, GP is not applied to the whole training image, but only to a region carefully selected from the training image, to generate the composite operators. The generated composite operator is then applied to the whole training image and some other testing images to evaluate it. The advantage of performing training on a small selected region is that it can greatly reduce the training time, making it practical for the GP system to be used as a subsystem of other learning systems, which improve the efficiency of GP by adapting the parameters of GP system based on its performance. Our experiments show that if the training region is carefully selected and is representative of the training images, the composite operator generated by GP is effective. In the following experiments, the parameters are: population size (100), the number of generations (100), the fitness threshold value (0.9), the crossover rate (0.6), the mutation rate (0.1), the maximum size of composite operator (30), and the threshold for segmentation (0).

3.1. Extracting lake from real SAR images

Two SAR images contain lake. The training image shown in Figure 2(a) contains lake and field and the testing image shown in Figure 2(c) contains lake and grass. The size of both images is 128×128. Figure 1(a) shows the selected region used in training. The size of the region is 43×43. It is located from (85, 85) to (127, 127) in the training image. The ground truth region is shown in Figure 1(b) and it is used in training to evaluate the composite operator. It is not needed in testing.



(a) (b) (c)

Figure 1. Training region containing lake.

The steady-state GP was used to generate a composite operator to extract the lake. Figure 1(c) shows the region extracted after segmenting the output image of the best composite operator in the final population. The fitness value of the best composite operator in the initial population is 0.72 and the average fitness value of the population is 0.32. The fitness value of the best composite operator in the final population is 0.91 and the average fitness value of the population is 0.75. GP found a good composite operator to extract the lake from the real SAR image. The best composite operator has 24 internal nodes and 9 leaf nodes. Several internal nodes contain MED_OP primitive operator, which is very useful in speckle noise reduction. It is shown in Figure 3, where IM0 is original image, IM1, IM3 and IM5 are 3×3, 5×5 and 7×7 mean images respectively, and IM2 and IM4 are 3×3 and 5×5 standard deviation images respectively.

We applied the composite operator to the whole training image in Figure 2(a) and the testing image in Figure 2(c). From each image, we generated seven primitive images and used them as the input to the composite operator. Figure 2(b) shows the region extracted from Figure 2(a). The fitness value is 0.90. Figure 2(d) shows the region extracted from Figure 2(c) and the fitness value is 0.93.

(a) (b) (c) (d)

Figure 2. SAR images containing lake.

```
(MAX_OP (DIV_OP (LOG_OP (DIV_OP (STDV_OP
(DIV_CONST_OP (MAX2_OP (ADD_OP IM3 IM3)
IM4))) (MED_OP (MED_OP (SUB_OP (MED_OP
(LOG_OP (MED_OP (MED_OP (DIV_CONST_OP
(MUL_OP IM2 (DIV_OP IM2 IM1))))))) (MIN_OP
IM4))))) (MEAN_OP (MAX_OP (MEAN_OP
(MUL_OP (MED_OP IM5) IM1))))))
```

Figure 3. Learned composite operator tree in LISP notation.

3.2. Extracting target from real SAR images

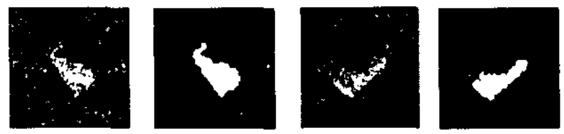
Two SAR images contain T72 tank. Both images have size 80×80. The tank in the training image shown in Figure 5(a) has azimuth angle 135° and depression angle 17°. Figure 4(a) shows the selected training region of size 50×50 and Figure 4(b) shows the ground truth. The training region is located from (19, 17) to (68, 66) in the training image. The tank in the testing image shown in Figure 5(c) has azimuth angle 225° and depression angle 20°.

(a) (b) (c)

Figure 4. Training region containing T72 tank.

The generational GP was used to generate the composite operator. The fitness value of the best composite operator in the initial population is 0.65 and the average fitness value of the population is 0.15. Figure 4(c) shows the region of interest extracted by the best composite operator in the final population. The fitness value of the best composite operator in the final population is 0.86 and the average fitness value of the population is 0.84.

We applied the composite operator to the whole training image in Figure 5(a) and the testing image in Figure 5(c). Figure 5(b) shows the region of interest extracted by the composite operator from Figure 5(a). The fitness value of the extracted region is 0.82; Figure 5(d) shows the region of interest extracted from Figure 5(c). The fitness value of the extracted region is 0.82.



(a) (b) (c) (d)

Figure 5. SAR images containing T72 tank.

4. Conclusions

Our experimental results show that GP can find good composite operators to effectively extract the regions of interest in an image and the composite operators can be applied to extract ROIs in other similar images. In the future, we plan to extend this work by designing smart crossover and mutation operators to improve the efficiency of GP and discovering new features within the regions of interest for automated object recognition.

Acknowledgement: This research is supported by the grant F33615-99-C-1440. The contents of the information do not necessarily reflect the position or policy of the U. S. Government.

References

[1] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, 1994.

[2] R. Poli, "Genetic programming for feature detection and image segmentation," in *Evolutionary Comp*, T. C. Forgarty Ed., pp. 110-125, 1996.

[3] S. A. Stanhope and J. M. Daida, "Genetic programming for automatic target classification and recognition in synthetic aperture radar imagery," *Proc. Conf. Evolutionary Programming VII*, pp. 735-744, 1998.

[4] D. Howard, S. C. Roberts, and R. Brankin, "Target detection in SAR imagery by genetic programming," *Advances in Engg. Software*, 30(5), pp. 303-311, May 1999.

[5] S. C. Roberts and D. Howard, "Evolution of vehicle detectors for infrared line scan imagery," *Proc. Workshop, Evolutionary Image Analysis, Signal Processing and Telecommunications*, pp. 110-125, 1999.

[6] B. Bhanu and Y. Lin, "Learning composite operators for object detection," *Proc. Genetic and Evolutionary Computation Conference*, July, 2002.