Mathematical Problems of Computer Science 32,101--106, 2009.

# Detection and Classification of Objects by Applying Genetic Programming

Nerses A. Safaryan

Institute for Informatics and Automation Problems of NAS RA
nerses.s@gmail.com

**Abstract**

This article is devoted to discussion of the problem of detection and classification of objects in digital images by using genetic programming (GP).

## 1. Introduction

Automatic systems of object detection in digital images have many applications in target recognition, video surveillance. The major task of object detection is to locate and extract regions of an image that may contain potential objects [1]. The quality of object detection is dependent on the type and quality of features extracted from an image that may contain potential objects [2]. In this work we tried synthesze composite operators by means of GP, which classify many different types of objects [3], [4]. A composite operator consists of primitive operators and primitive feature images [5].

GP uses five major considerations in the task of object detection. The set of terminals is sixteen primitive feature images generated from the original image, the set of primitive operators (see Table 1), the fitness measure, parameters and termination. Brief discussion about these considerations is followed.

The set of terminals include following images: $F_0, F_1, \ldots, F_{15}$, where $F_0$ is the original image, the $F_1, F_2, F_3$ are $3\times3$, $5\times5$, $7\times7$ mean images, the $F_4, F_5, F_6$ are $3\times3$, $5\times5$, $7\times7$ deviation images, the $F_7, F_8, F_9$ are $3\times3$, $5\times5$, $7\times7$ maximum images, the $F_{10}, F_{11}, F_{12}$ are $3\times3$, $5\times5$, $7\times7$ minimum images and the $F_{13}, F_{14}, F_{15}$ are $3\times3$, $5\times5$, $7\times7$ median images. The primitive operators are given below in the Table 1.

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 $Fitness = n(G \cap G') / n(G \cup G')$.

Table 1.

| No. | Operator | Description |
|---|---|---|
| 1 | *ADD(A,B)* | Add images A and B. |
| 2 | *SUB(A,B)* | Subtract image B from A. |
| 3 | *MUL(A,B)* | Multiply images A and B. |
| 4 | *DIV(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(A,B)* | The pixel in the resultant image takes the larger pixel value of images A and B. |
| 6 | *MIN2(A,B)* | The pixel in the resultant image takes the smaller pixel value of images A and B. |
| 7 | *ADDC(A)* | Increase each pixel value by c. |
| 8 | *SUBC(A)* | Decrease each pixel value by c. |
| 9 | *MULC(A)* | Multiply each pixel value by c. |
| 10 | *DIVC(A)* | Divide each pixel value by c. |
| 11 | *SQRT(A)* | For each pixel with value v, if $v \geq 0$, change its value to $v$. Otherwise, to $-\sqrt{-v}$.□ |
| 12 | *LOG(A)* | For each pixel with value v, if $v \geq 0$, change its value to ln(v). Otherwise, to –ln(-v). |
| 13 | *MAX(A)* | Replace the pixel value by the maximum pixel value in a 3×3, 5×5 or 7×7 neighborhood. |
| 14 | *MIN(A)* | Replace the pixel value by the minimum pixel value in a 3×3, 5×5 or 7×7 neighborhood. |
| 15 | *MED(A)* | Replace the pixel value by the median pixel value in a 3×3, 5×5 or 7×7 neighborhood. |
| 16 | *MEAN(A)* | Replace the pixel value by the average pixel value of a 3×3, 5×5 or 7×7 neighborhood. |
| 17 | *STDV(A)* | Replace the pixel value by the standard deviation of pixels in a 3×3, 5×5 or 7×7 neighborhood. |

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

## 2. Crossover and Mutation

By searching through the composite operator space, GP adapts the population of composite operators from generation to generation and improves the overall fitness of the whole population. The search is done by performing selection, crossover and mutation operations.

The selection operation involves selecting composite operators from the current population. In this work, tournament selection is used, 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.

To perform crossover, two composite operators are selected on the basis of their fitness values. The higher the fitness value, the more likely the composite operator is selected for crossover. These two composite operators are called parents. One internal node in each of these two parents is randomly selected, and the two subtrees rooted at these two nodes are exchanged between the parents to generate two new composite operators, called offspring.

In order to avoid premature convergence, mutation is introduced to randomly change the structure of some individuals to maintain the diversity of the population. Composite operators are randomly selected for mutation. There are three types of mutation invoked with equal probability:

1. Randomly select a node of the binary tree representing a composite operator and replace the subtree rooted at this node, including the node selected, by another randomly generated binary tree.

2. Randomly select a node of the binary tree representing a composite operator and replace the primitive operator stored in the node with another primitive operator of the same number of inputs as the replaced one. The replacing primitive operator is selected at random from all the primitive operators with the same number of input as the replaced one.

3. Randomly select two subtrees within a composite operator and swap them. Of course, neither of the two sub-trees can be a sub-tree of the other.

## 3. The Synthesis of Composite Operators

To synthesze composite operators for object detection in digital images is used following algorithm: we may have a population of composite operators after one time applying this algorithm, the best composite operator from the population identies one type of objects.

- for gen = 1 to N do // N is the number of generation,
- keep the best composite operator in P,
- repeat,
- select 2 composite operators from P based on their fitness values for crossover,
- select 2 composite operators with the lowest fitness values in P for replacement,
- perform crossover operation and let the 2 offspring replace the 2 composite operators selected for replacement,
- execute the 2 offspring and evaluate their fitness values,
- until crossover rate is met,
- perform mutation on each composite operator with probability of mutation rate and evaluate mutated composite operators,
- After crossover and mutation, a new population P' is generated,
- let the best composite operator from population P replace the worst composite operator in P' and let P = P',
- if the fitness value of the best composite operator in P is above fitness threshold value then,
- stop.

We have set of composite operators $O=\{O_0, O_1, \dots, O_{16}\}$ and set of terminals. Randomly generate population P of size M from O and P. Let $O_n, O_k, O_m$ be selected from O and $F_1, F_p$ from F. We obtain combination of them, for example $O_k(O_m(F_1))$ (composite operator). Then evaluate each composite operator in P and perform crossover operation. If we show composite operators as binary tree, they will be like on figure 1. We can have two offspring (composite operators) after crossover operation, which is shown in figure 1.1. These two offsprings replace the two worst composite operators in P. This is continued until crossover rate is met. Then mutation is performed, which can replace the primitive operator stored in the node with another primitive operator.

For example, if we observe the tree on figure 2.1, after mutation we can have the tree on figure 2.2. We get the tree, which is the best.

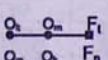


Figure. 1

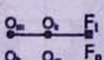


Figure. 1.1

Figure. 2.1

Figure. 2.2

## Experiments

Various experiments are performed on real synthetic aperture radar (SAR) images (128×128), to detect some type of objects. Figure 5(a) is used to synthesze a composite operator. The objects in images are roads. To synthesze the composite operator the set of terminals (Figure 3) and set of primitive operators are used. Following composite operator is found [2]:
(MAX (MAX (MIN (DIVC (DIV(ADDC (ADD (ADDC (ADD (SUBC(ADDC (ADD (SUBC (STDV (MAX(SUBC PFIM15)))) (MAX (SUBCPFIM14))))) (MAX (SUBCPFIM14)))) (MAX (SUBC PFIM14))))PFIM15))))).
The composite operator has 27 nodes, five leaf nodes, three contain 5×5 median image and the other two contain 7×7 median image. Output of composite operator is shown in Figure 5(b) and after segmentation: 5(c).

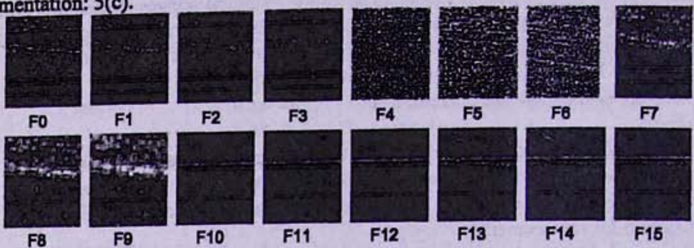
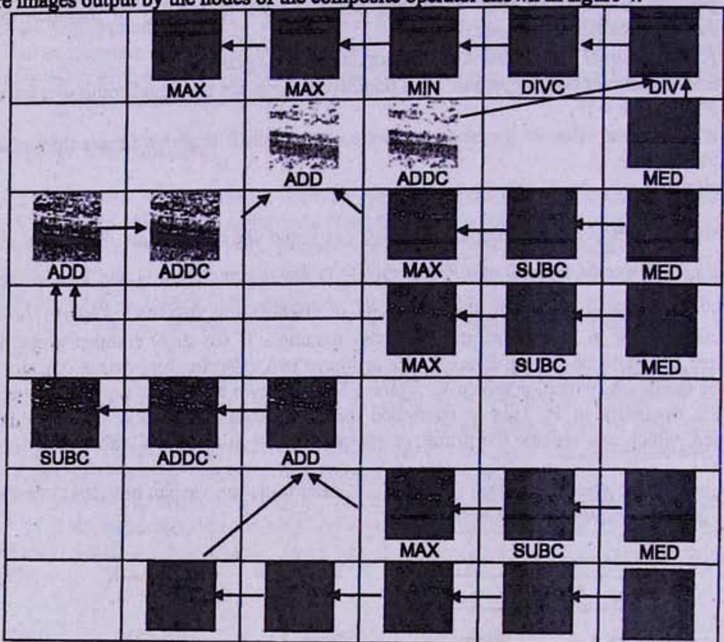



Figure. 3. Set of terminals

Feature images output by the nodes of the composite operator shown in figure 4.



| | SUBC | STDV | MAX | SUBC | MED |
|---|---|---|---|---|---|
| | | | | | |

Figure. 4. Feature images output by the nodes of the best composite operator.

a b c

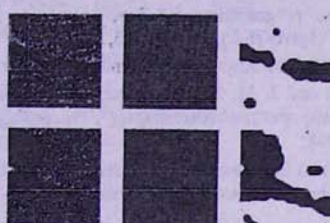
Figure. 5. a- original image, b- composite feature image, c- after segmentation

The same composite operator is used for other experiment (it was applied on image in figure 7(a), in which object also road), the result of which is shown in Figure 6.

a b c

Figure. 6. a- original image, b- composite feature image, c- after segmentation

The result of other experiment, the composite operator is the following (MULC (MED (MED (MED (MED (MED (MED (MED (MED (**MAX** (MED (**MAX** (MED (MIN (ADDC (LOG (ADD (MAX (MIN (MULCPFIM2))) (DIV (MIN MULC (MED (MIN (MAX (SUB PFIM2 (MULC PFIM2))))))) PFIM1)))))))))))))))))) [2], is shown on Figure 7. This composite operator has 30 nodes and its depth is 23. It has four leaf nodes, three contain 5×5 mean image and the other one contains 3×3 mean image. The composite operator is applied on images, which contain river and field. If we present the composite operator as binary tree, it will be so as shown in Figure. 9.



a b c

Figure. 7. a- original images, b- composite feature images, c- after segmentation

We made some changes in the composite operator (we added to composite operator two MAX primitive operators, which are signed by bold style). After changes the composite operator has 32 nodes and its depth is 25. The result of experiment after changes is given in Figure 8.

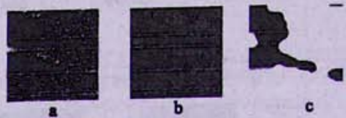



Figure. 8. a- original images, b- composite feature images, c- after segmentation

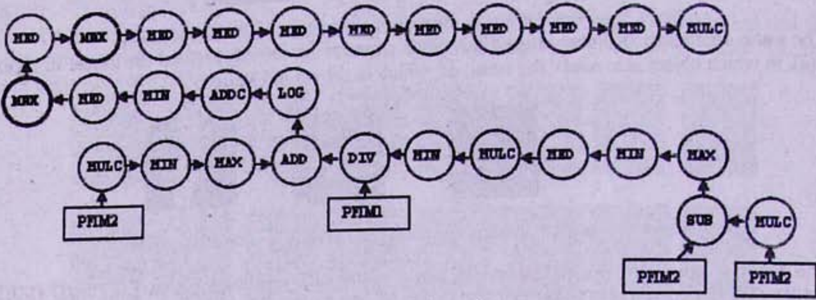



Figure. 9. Binary tree of second composite operator

## References

[1]. C. Harris and B. Buxton, "Evolving edge detectors with genetic programming," *Proceedings of the Genetic Programming, 1st Annual Conference*, Cambridge, MA, USA, MIT Press, pp. 309-314, 1996.

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

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

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

[5]. B. Bhanu and Y. Lin, "Learning composite operators for object detection", *Proceedings of the Genetic and Evolutionary Computation Conference*, pp.1003-1010, 2002.

## Օբյեկտների հայտնաբերումը և դասակարգումը գենետիկ ծրագրավորման կիրառմամբ

Ն. Սաֆարյան

### Ամփոփում

Աշխատանքը նվիրված է գենետիկ ծրագրավորման միջոցով թվային պատկերներում օբյեկտների հայտնաբերման և դասակարգման խնդրի քննարկմանը։