COSC 4V82

Project: Airplane Detection

By Collin Price

# Introduction

The goal of assignment 3 was to use a genetic program (GP) to evolve a spatial image analyzer to detect the location of airplanes. The goal of this paper is to compare two methods of fitness measuring for finding a successful spatial image analyzer. The first method will be the same method that was used in assignment 3 and the second method will use fitness sharing. Fitness sharing is an evaluation method that is used to discourage premature convergence in a population. It allows for small differences in individuals from within the same population to become magnified and have a bigger impact.

# Experiment Setup

**Parameters for Runs**

Random seed: current time

Population size: 1000

Number of generations: 100

Training set size: 1 Image

Testing set size: 1 Image

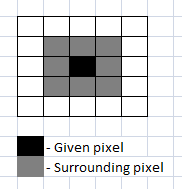
**GP Language**

Terminals

stdDev – standard deviation of nine surrounding pixels

9 pixels – is a 3x3 grid around a given pixel. See figure 1.

Figure 1:



Functions

‘-‘ – subtracts two given numbers

‘+’ – adds two given numbers

‘\*’ – multiplies two given numbers

‘/’ – protected division divides two given numbers. If second number of zero the result is zero

Max – returns the max number of two given numbers

Log – the natural logarithm of the absolute value of a given value

**Training Set**

The training image was created by taking a satellite photograph of some airplanes at an airport and masking each airplane with the colour red. See figure 2 for a sample image. The training phase of the GP was based on the number of pixels it detected as being an airplane compared to the red pixels on the masking image.

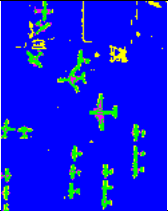
**Figure 2.**

1.  b) 
2. Is a black and white training image
3. Masked training image

**Testing Set**

In the testing set other satellite photographs are processed through the best Individual in the GP. Two images are generated from the test. The first image is a copy of the testing image except with the pixels it thinks is an airplane filled in with red pixels. The second image is a colour coded image that shows which pixels are correctly identified and which were incorrectly identified (Green = true positive, Blue = true negative, Yellow = false positive, Red = false negative). See figure 3 for an example.

**Figure 3.**

****

Sample output from the results of testing.

**Fitness Function**

Standard

The standard fitness function attempts to minimize the number of incorrectly identified pixels.

*Fitness = False Positives + False Negatives*

Fitness Sharing

Fitness sharing attempts to award those individuals who have characteristics that stand out from the other individuals. The first stage of fitness sharing is to evaluate each individual to see which pixels were correctly identified. The second stage is to look at the same pixel in each individual to see which individual correctly identified it. The individuals who correctly identified the pixel then have to share a reward. A correct pixel is given the point value of 1.0. For each individual who also identified the pixel the reward must be shared. For example, if one individual identifies a pixel correctly they are rewarded 1.0 points. If two individuals identify a pixel correctly they are each awarded 0.5 points.

# Results

Best Run (Red pixels are when the GP thinks it’s an airplane) for Standard Fitness

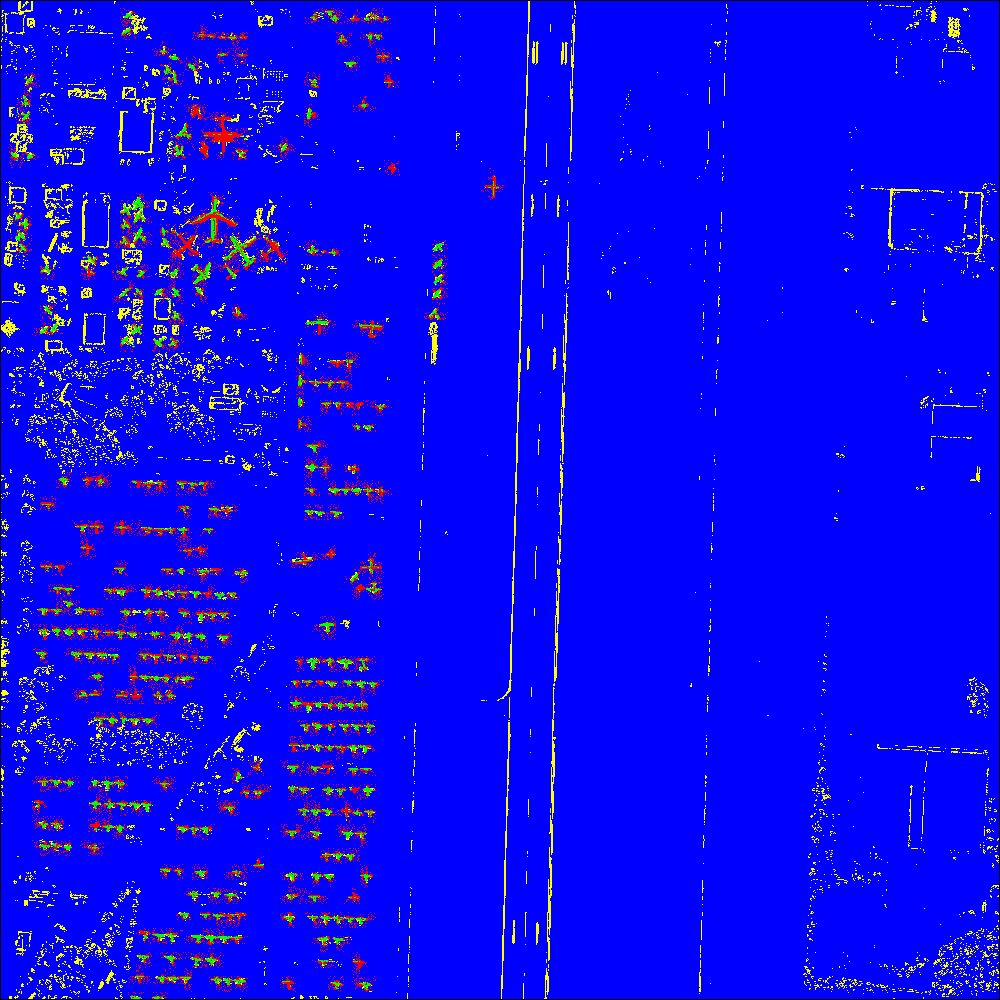


Best Run (Red pixels are when the GP thinks it’s an airplane) for Fitness Sharing



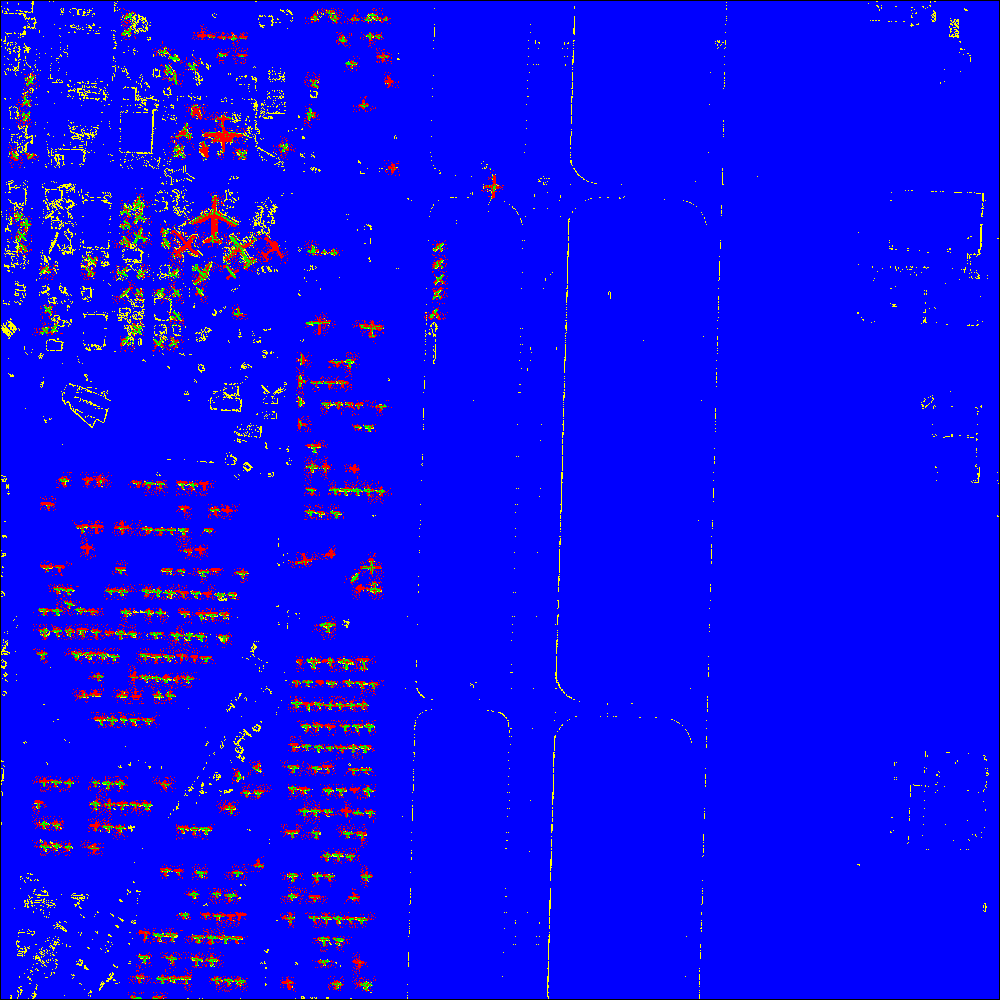
Best Run:

Colour Coded (Green = true positive, Blue = true negative, Yellow = false positive, Red = false negative) for Standard Fitness



Best Run:

Colour Coded (Green = true positive, Blue = true negative, Yellow = false positive, Red = false negative) for Fitness Sharing



Resulting Program:

Standard Fitness

(- (- (- (+ (+ (\* (Log (e (\* (Log (/ (- p9

p8) (/ p1 (Log (e p4))))) p3))) (/ p3 p8))

(/ p2 p5)) (Max (/ (e (/ p3 p8)) p3) (+ (+

(\* (Log (e (\* (Log (- (/ (Max p6 p1) (/ p2

p5)) p4)) p9))) (/ p3 p8)) (\* (Log (/ p2

(Max (Max (/ (/ p6 p4) (/ p9 (Max p6 p9)))

(/ p8 p6)) p9))) (e (Log (Max (Log (e (\*

(Log (e p4)) (/ p3 p8)))) (Max p6 p1))))))

(Max (/ (e (/ p3 p8)) p3) (\* (Log (/ (- p9

p4) p6)) (\* (Max p6 p1) p1)))))) (\* p3 p9))

(+ (- p9 (+ (+ (\* (Log (e (\* (/ p2 p3) (+

p8 p5)))) (/ (/ p2 p5) p8)) (\* (Log (/ (-

(+ p8 p5) (/ (Max p6 p1) (/ p2 p5))) (Max

StdDev p9))) (e (Log (\* (Log (e p4)) (/ p9

(/ p2 p5))))))) (/ p8 (e (\* (/ p2 p3) (e

(- p5 (+ p3 p9)))))))) (- p9 (+ (+ (/ p8

p6) (\* (- (- (e (- StdDev (+ p3 (- p9 p4))))

(e (/ p6 p4))) (Log (/ (- p9 p4) p6))) (/

(e (/ (/ p8 p6) (Max p6 p2))) (- (Max (\*

(/ p3 p8) (/ p2 p3)) (\* (Log (/ p1 (Max (/

(- p9 p4) (/ p6 p4)) p9))) (Max (/ (/ p6

p4) (/ (e (/ p8 p8)) (Max p6 p9))) (/ p8

p6)))) (Max (Log (\* p3 p6)) (/ (\* p8 (/ (/

(/ (- p9 p4) (/ p6 p4)) (Max p6 p9)) (Max

(/ (Max p6 p1) (/ p2 p5)) p9))) (/ p9 (Log

(\* (Log (e p4)) (/ p6 p4)))))))))) (- (+

(/ p2 p5) p5) (\* p1 p9)))))) (+ (- (Log (e

p4)) (+ (+ (\* (Log (e (\* (Log (/ p9 (/ (+

p8 p5) (\* (/ p2 p3) (- p9 p4))))) p3))) (/

p3 p8)) (/ p2 p5)) (Max (/ (e (/ p3 p8))

p3) (- p9 p4)))) (- (Max p6 p9) p4)))

Shared Fitness

Tree 0:

(- (Max (Log (+ (Max (+ p7 p7) p2) p6)) (Log

(Max (Max (/ p2 p6) (e (/ (\* (- (- (Max (-

p6 (- p5 p4)) (- p5 (+ (+ p9 p6) (+ p9 p5))))

(Max p6 p4)) (- p5 p4)) (/ (- p5 (+ (Max

(/ p2 p6) p4) p6)) (/ (+ p9 (+ p9 p6)) p2)))

(Log p9)))) p9))) (/ (- (- (- p5 (+ p9 p5))

(Max p6 p4)) (+ (/ (Max (Max (/ p7 p2) (Max

(+ (Log p9) (+ p5 (/ p4 p5))) (e p9))) (e

(/ p7 p3))) (- p5 (+ p5 p7))) (- (Max (/

p8 (Max (/ p8 (Max p6 p4)) (+ p9 p5))) (+

p9 p5)) (Max p7 p8)))) (Max (Log (Max (e

(/ (\* (Log (\* (- p5 p4) (/ (- p5 (+ p9 p5))

(/ (+ p9 p5) p5)))) (- p5 (Log p9))) (/ (+

(Max (/ p2 p6) p4) p6) p4))) (e (/ (\* (-

(- p5 (Max p6 p4)) (/ p2 p6)) (/ (- p5 (+

p9 p5)) (/ (+ p9 p6) p5))) (Log (+ p9 p5))))))

(Max p9 p7))))

These graphs are an average taken over 10 runs on the training set:

Training Image:

Using Standard Fitness

Number is correctly identified pixels (True Positives and True Negatives). Total Pixels = 19468

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Run #1 | Run #2 | Run #3 | Run #4 | Run #5 | Run #6 | Run #7 | Run #8 | Run #9 | Run #10 |
| Hits | 17937 | 17889 | 17661 | 17902 | 17845 | 17703 | 17811 | 17922 | 17762 | 17839 |

Using Fitness Sharing

Number is correctly identified pixels (True Positives and True Negatives). Total Pixels = 19468

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Run #1 | Run #2 | Run #3 | Run #4 | Run #5 | Run #6 | Run #7 | Run #8 | Run #9 | Run #10 |
| Hits | 17960 | 17838 | 17825 | 17548 | 17769 | 17846 | 17923 | 17786 | 17857 | 17659 |

Average Results

|  |  |  |
| --- | --- | --- |
|  | Standard Fitness | Fitness Sharing |
| Average | 17827.1 | 17801.1 |
| Standard Deviation | 93.37791 | 121.4235 |

Testing Image:

Using Standard Fitness

Number is correctly identified pixels (True Positives and True Negatives). Total Pixels = 1000000

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Run #1 | Run #2 | Run #3 | Run #4 | Run #5 | Run #6 | Run #7 | Run #8 | Run #9 | Run #10 |
| Hits | 949474 | 938432 | 945921 | 946315 | 935214 | 947583 | 949363 | 948946 | 949254 | 948616 |

Using Fitness Sharing

Number is correctly identified pixels (True Positives and True Negatives). Total Pixels = 1000000

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Run #1 | Run #2 | Run #3 | Run #4 | Run #5 | Run #6 | Run #7 | Run #8 | Run #9 | Run #10 |
| Hits | 949886 | 961456 | 934728 | 941464 | 949451 | 927795 | 939090 | 960107 | 943758 | 956391 |

Average Results

|  |  |  |
| --- | --- | --- |
|  | Standard Fitness | Fitness Sharing |
| Average | 945911.8 | 946412.6 |
| Standard Deviation | 5007.054 | 11070.51 |

Another good resulting program:

# Discussion of Results

If we look at the average case for both standard fitness and fitness sharing the results are very similar. There does not appear to be any dramatic advantage to using fitness sharing in the case of developing a spatial image analyzer.

The standard deviation of the results gives a different result. The standard deviation of the standard fitness is much lower than that of fitness sharing. Taking a look at the individual results, fitness sharing had runs that were a lot better than runs from standard fitness but it also had worse runs than standard fitness. This shows that although fitness sharing can generate better results than standard fitness, it is also less consistent. Looking at the results it could be considered twice as inconsistent because the standard deviation of fitness sharing is double that of standard fitness.

# Conclusion

The results from my testing showed that airplane identification is possible through GP. The training of this GP took and long time and I think that if I had limited my tree depth I could have trained a lot faster. To improve the results of this GP I would have trained it on a wider variety of airplane shapes and colours. Another change would have been to make a large moving window. After comparing my results with other students I should have used a 5x5 grid or a 9x9 grid for a greater range of pixels. I suspect that the reason it mostly developed an edge detector was because of the small sample of 9 pixels.