An Elegant Weapon for a More Civilized Age: Evolving Computer Vision in Recognition of Lightsabers in Greyscale Images from Star Wars with Genetic Programming

Liam McDevitt
6015929
Computer Science
Brock University
St. Catharines, Canada
lm15ue@brocku.ca

Andrew Pozzuoli
6017735
Computer Science
Brock University
St. Catharines, Canada
ap15yl@brocku.ca

I. Abstract

This paper looks at applying genetic programming (GP) to evolve a way of identifying lightsabers in greyscale frames of Star Wars. One experimental variable explored was modifying the weight of fitness scores where true positive hits of lightsabers were weighted higher than true negatives which gave the GP more incentive to correctly identify lightsabers and resulted in better true positive performance overall. Varying the sizes of area filters was also explored but for this problem there was no significant difference between filter areas on performance from this GP.

II. Introduction

This problem uses genetic programming (GP) to evolve a function that uses a given set of data points relating to a pixel in an image and its surrounding pixels to identify whether or not that pixel belongs to a lightsaber in the context of that image. As an added challenge, the images are greyscaled to avoid the GP just looking for bright colours. This adds another challenge since the lightsabers could appear the same intensity as any other brightly lit pixels in the image such as a white shirt, light-coloured wall, or any light reflected off any object. Another challenge related to this particular problem is the shape of the lightsabers themselves. Since they are not confined to small areas and are shaped in a line, the area filters might not be large enough to contain more than a small segment of the lightsaber. This may mean that choosing a good set of filter data is important for taking these segments into account within the context of the whole image.

III. Experimental Setup

A. Data

The training set is a series of data points describing every pixel and surrounding pixels in a 250x250 image from Star Wars. This is a total of 62500 pixels, each pixel representing either a lightsaber or not a lightsaber based on a coloured ground truth image. The goal is for GP to use the data of each pixel to return whether or not that pixel belongs to a lightsaber. There were two testing sets as well, each a 250x250 image with the same data per pixel. All training and testing sets come from the same frame in Star Wars: Attack of the Clones but were sliced three ways into 250x250 pixel images [1].

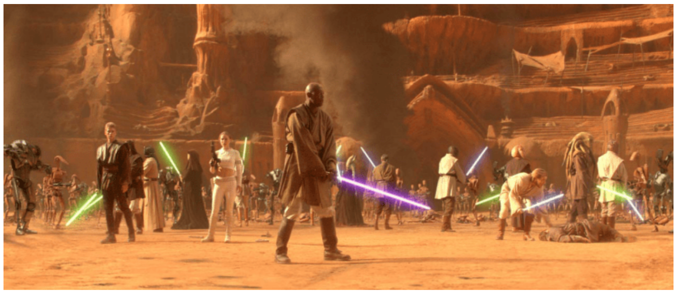


Fig. 1. Full Frame from Star Wars: Attack of the Clones

B. GP Language

The data sets were precomputed to speed up runtime of GP evolution. Rather than passing in the image and going pixel by pixel to calculate the data for the terminals during the run, this information was put into a text file with each line representing one pixel. GP then goes line by line and uses these sets of numbers as the terminals. All terminals were float values.

The filter area for terminals was variable with experimental values of 3x3, 7x7, and 11x11 centered on the pixel in question. The default filter area was 11x11 for the experiment involving weighted fitness values.

C. Parameters

The maximum tree depth and grow depth values were restricted to 7 and 4 respectively from the default 17 and 6 which reduced runtime significantly without sacrificing quality

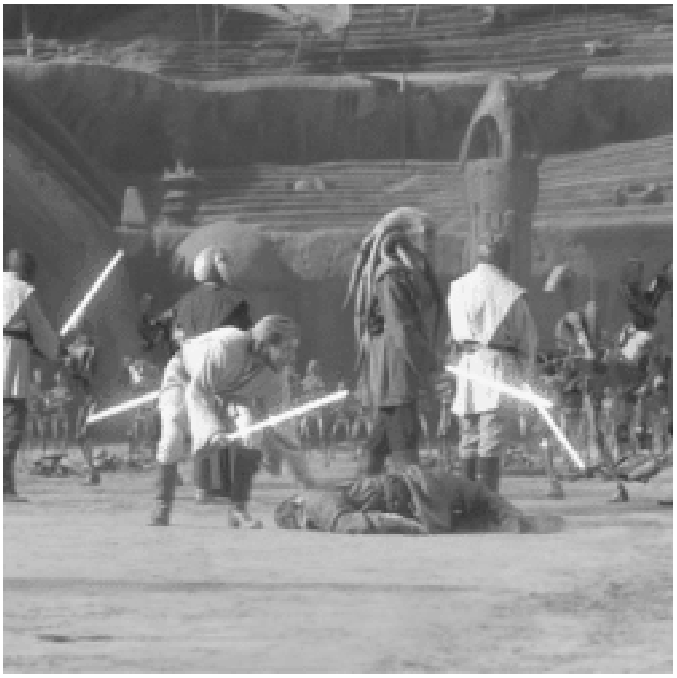


Fig. 2. Training Set Image Greyscaled

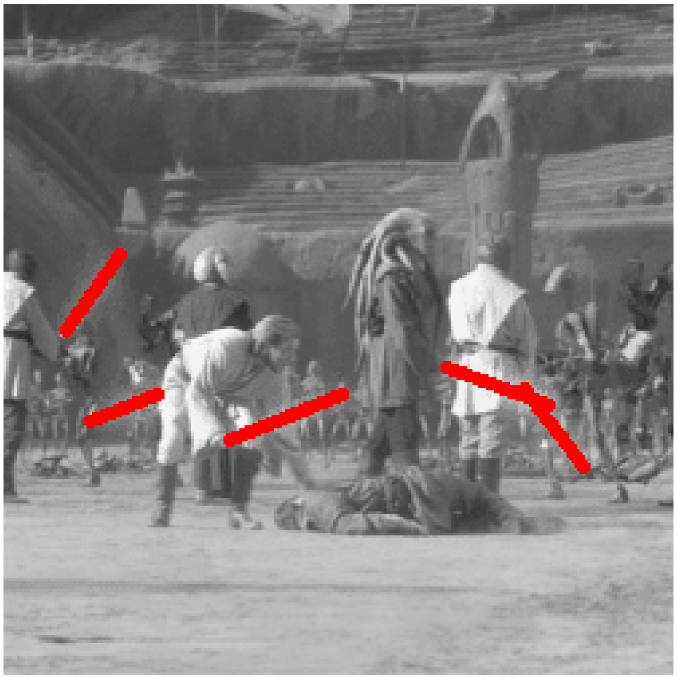


Fig. 3. Ground Truth Training Image

TABLE I
TERMINALS

Terminal	Description
Intensity	The value of the intensity of the pixel
Max Intensity	The maximum intensity value for a pixel in that area filter
Min Intensity	The minimum intensity value for a pixel in the area filter
Mean Intensity	Average intensity of all pixels in the area filter
Standard Deviation Intensity	The standard deviation of the intensity of all pixels in the area filter
Ephemeral	Random constant value initialized at start of run

of solutions during preliminary runs. Population was also reduced to 100 from the default 500 to improve runtime.

TABLE II
GP PARAMTERS

Parameter	Value
Population Size	100
Generations	50
Population Initialization	Ramped Half and Half
Minimum Grow Depth	2
Maximum Grow Depth	4
Maximum Tree Depth	7
Selection	Tournament, k=4
Crossover	Subtree Crossover, 95%
Mutation	5%
Data Points Training	62500
Data Points Testing Set 1	62500
Data Points Testing Set 2	62500
Runs Per Experiment	10

D. GP Language

TABLE III
FUNCTIONS

Function	Description
Add	Adds two terminals
Multiply	Multiplies two terminals
Subtract	Subtracts two terminals
Divide	Divides two terminals If the denominator is zero then set the denominator to 1
Negative	Makes the terminal negative
Square	Multiplies the terminal value with itself
Square Root	Takes the square root of a terminal If the terminal is negative then take the square root of the absolute value of the terminal

E. Fitness Evaluation

The program evolved a function that would ideally return a negative value if the pixel was not a lightsaber and a positive value if the pixel belonged to a lightsaber. In the data sets, this was represented as 0 for not belonging to a lightsaber and 1 for belonging to a lightsaber.

The fitness was evaluated by measuring the number of times the function correctly guessed whether a pixel was or was not part of a lightsaber (hits). Every time the evolved function produced a negative value and the input data corresponded to a non-lightsaber pixel or if the function produced a positive value and the input data corresponded to a lightsaber pixel, the sum of hits increased according to the following equation:

$$hits(i) = tp * w + tf$$

Where *i* is the individual being evaluated, *tp* is true positive, *tf* is true negative, and *w* is a weight value. The weight was added to true positive values as an experimental variable with values of 1, 100, 250, and 500.

The standardized fitness was calculated according to the following equation:

$$sfitness(i) = (N + P * w) - hits(i)$$

Where *i* is the individual being evaluated, *N* is the ground truth negative pixels in the image, *P* is the ground truth positive pixels in the image, *w* is the weight applied to true positive results of the evolved function, and *hits(i)* is the number of hits according to the aforementioned fitness evaluation. This gives the sum of errors, zero being the best and higher numbers being worse.

The adjusted fitness was calculated by subtracting the percentage of hits from one. This gives a number between 0 and 1 with 0 being the ideal and 1 being the worst.

IV. Experiment

The experiments performed analyzed the effects of modifying the weighted fitness during evaluation and filter areas during data input.

For experimenting with the weight of true positives, the experimental values were 1, 100, 250, and 500. Ten runs were performed for each weight value and the resulting fitnesses were averaged over those ten runs. The results of this experiment would indicate whether giving more weight to correct positives has an effect on the performance of the GP on identifying lightsabers.The default weight for the second experiment was 100.

For experimenting with the filter areas, the input data for each input regarding filters was modified to be either 3x3, 7x7, or 11x11 centered on the pixel in question. The results would indicate whether the filter size affected GP's ability to accurately identify a lightsaber. The default filter area for the first experiment was 11x11.

At the end of each run, the best individual was run on two testing images to see the generality of the solution evolved and a confusion matrix was generated for both the discrete number of hits and the percentage of hits on both testing images.

A coloured image was also created for each testing image visually indicating what pixels the evolved solution correctly identified and which it did not according to the following legend:

TABLE IV
Performance Image Legend

Green	True Positive
Black	True Negative
Red	False Positive
Yellow	False Negative

V. Results

A. Training Set Performance

The adjusted fitness was used to plot the average fitness over ten runs for each experiment with 1 being the worst and 0 being ideal.

1) Weight: The first fitness plot describes the performance where the true positive are weighted 1 during evaluation. The fitness plot shows a high initial average fitness that sharply improves over the next five generations and converges around generation 25 with the best fitness remaining constant throughout the run.

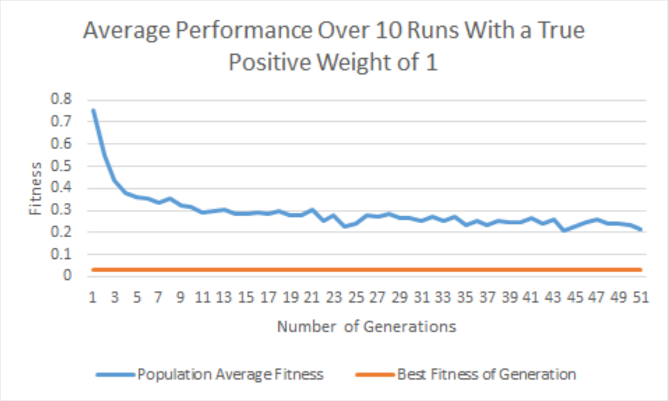


Fig. 4. Population average and average best performance over ten runs with true positives weight = 1

With true positives weighted at 100, the initial population adjusted fitness is much lower than with weight 1. There is a convergence around generation 25 and unlike with weight 1, the best individual's fitness improved over the course of the run.

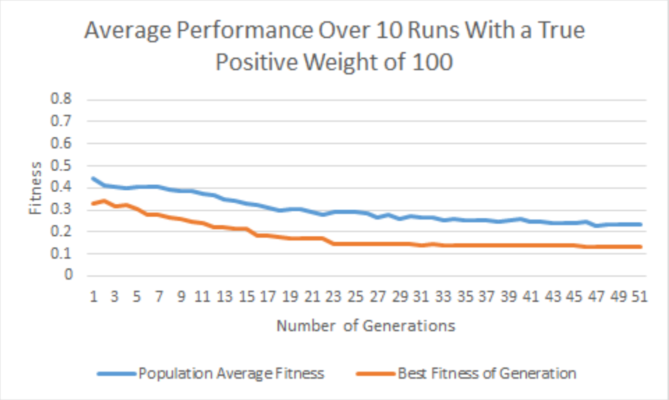


Fig. 5. Population average and average best performance over ten runs with true positives weight = 100

With true positives weighted at 250, the population average converged quickly around generation 3 and the best fitness converged slower around generation 25. Both fitnesses converged at a lower value than with true positive weigthed at 100.

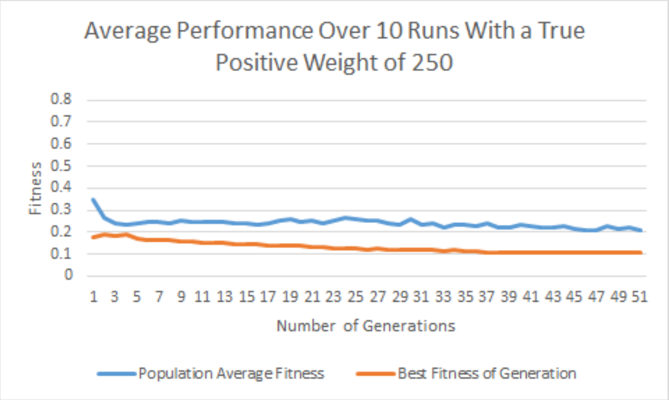


Fig. 6. Population average and average best performance over ten runs with true positives weight = 250

When true positive is weighted at 500 the population average seems to improve sharply at first and then slowly gets worse over the course of the run. The best individual per generation seems to slowly improve over the run.

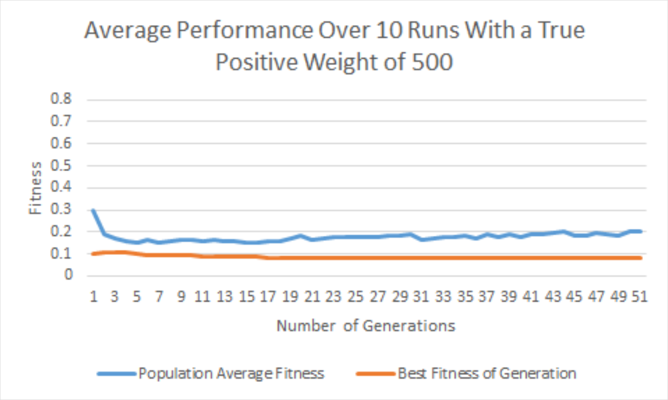


Fig. 7. Population average and average best performance over ten runs with true positives weight = 500

2) Filter Areas: All three experiments for filter areas converged around the same fitnesses, with population average around 2.5 and average best around 1.5. The only difference is how quickly the three converged. Filter area 3x3 converged the quickest around generation 25 and both area 7x7 and 11x11 converging more steadily.

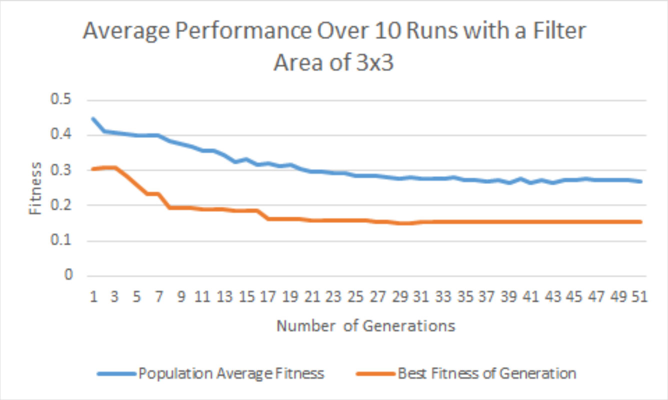


Fig. 8. Population average and average best performance over ten runs with filter area of 3x3

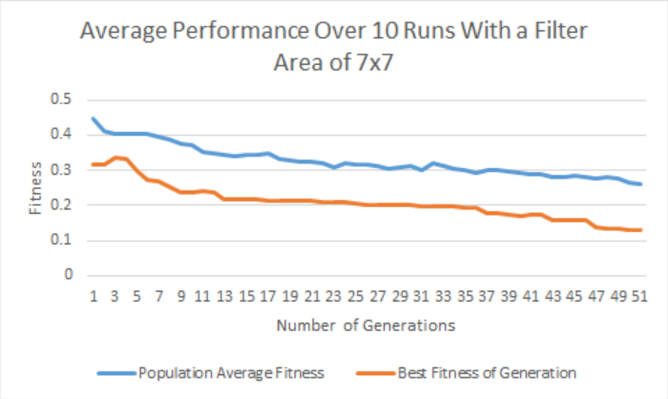


Fig. 9. Population average and average best performance over ten runs with filter area of 7x7

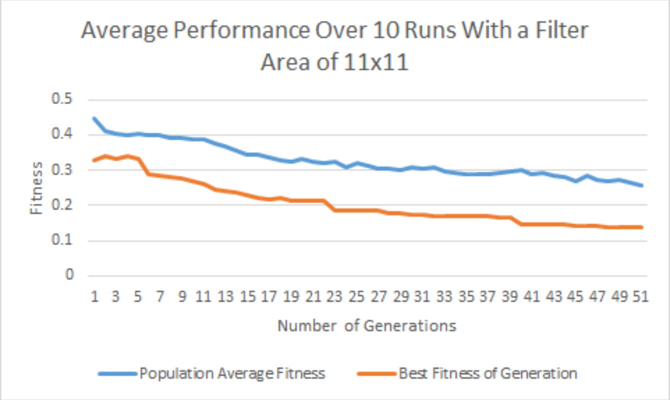


Fig. 10. Population average and average best performance over ten runs with filter area of 11x11

B. Testing Set Performance

To evaluate the GP's performance and generality, the best individual was run on two testing sets. The best of these for each experiment is reported. The following confusion matrices were taken from the best performing individual per experiment. The best run per experiment was determined by taking the best individual from every run and summing its true positive and true negative percentages from each testing set it was run on. The individual with the closest value to 4 after this calculation was determined to have performed the best on both tests (ie. the ideal solution would get a percentage of 1.0 on the true positive and true negative for both tests so we have 1.0 + 1.0 + 1.0 + 1.0 = 4.0). In the following confusion matrices, L means that the pixel belongs to a lightsaber and NL means the pixel does not belong to a lightsaber.

TABLE V
CONFUSION MATRIX LEGEND

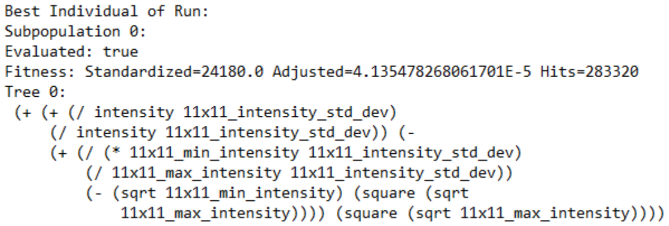
Legend		Actual Answer	
		L	NL
GP	L	True Positives	False Positives
	NL	False Negatives	True Negatives

The best individual during testing for experiment 1 involving weights was the tree in figure 11 where the true positive weight is 250.The confusion matrix shows that for the first testing set, the tree correctly identified lightsabers 93.8% of the time and correctly identified non-lightsaber pixels 91.8% of the time. On testing set 2, the tree scored 97.8% on lightsaber pixels and 91.8% on non-lightsaber pixels.

The best individual for experiment 2 involving filter areas during testing was the tree in figure 12 with a filter area of 7x7. For test image 1, this tree correctly identified lightsabers 97.99% of the time and correctly identified non-lightsaber pixels 89.9% of the time. For test image 2, the tree scored 99.1% on lightsabers and 91.2% on non-lightsabers.

VI. Statistical Analysis

We will use an ANOVA test for our various filter areas we are testing, 3x3, 7x7, and 11x11. An ANOVA test will show



```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=24180.0 Adjusted=4.135478268061701E-5 Hits=283320
Tree 0:
 (+ (+ (/ intensity 11x11_intensity_std_dev)
     (/ intensity 11x11_intensity_std_dev)) (-
     (+ (/ (* 11x11_min_intensity 11x11_intensity_std_dev)
         (/ 11x11_max_intensity 11x11_intensity_std_dev))
         (- (sqrt 11x11_min_intensity) (square (sqrt
             11x11_max_intensity)))) (square (sqrt 11x11_max_intensity))))
```

Fig. 11. Best individual across all runs in experiment 1 when run on test images. The fitness true positive weight was 250.

TABLE VI
WEIGHT 250 TESTING SET BEST PERCENTAGE HITS ON TEST IMAGE 1

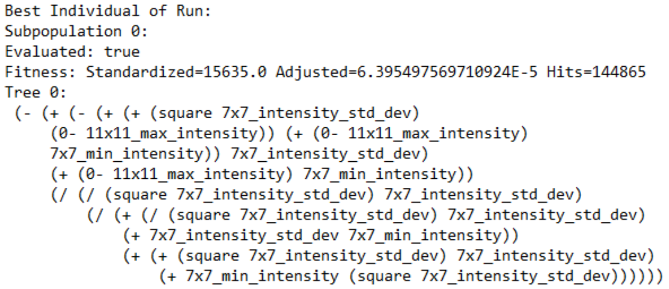
% of Hits		Actual Answer	
		L	NL
GP	L	0.93842036	0.08151831
	NL	0.061579652	0.9184817

Table VI: Refer to figure 15 in appendix for corresponding performance image

TABLE VII
WEIGHT 250 TESTING SET BEST PERCENTAGE HITS ON TEST IMAGE 2

% of Hits		Actual Answer	
		L	NL
GP	L	0.9780755	0.100990616
	NL	0.021924483	0.8990094

Table VII: Refer to figure 16 in appendix for corresponding performance image



```
Best Individual of Run:
Subpopulation 0:
Evaluated: true
Fitness: Standardized=15635.0 Adjusted=6.395497569710924E-5 Hits=144865
Tree 0:
 (- (+ (- (+ (+ (square 7x7_intensity_std_dev)
     (0- 11x11_max_intensity)) (+ (0- 11x11_max_intensity)
     7x7_min_intensity)) 7x7_intensity_std_dev)
     (+ (0- 11x11_max_intensity) 7x7_min_intensity))
     (/ (/ (square 7x7_intensity_std_dev) 7x7_intensity_std_dev)
         (/ (+ (/ (square 7x7_intensity_std_dev) 7x7_intensity_std_dev)
             (+ 7x7_intensity_std_dev 7x7_min_intensity))
             (+ (+ (square 7x7_intensity_std_dev) 7x7_intensity_std_dev)
                 (+ 7x7_min_intensity (square 7x7_intensity_std_dev))))))
```

Fig. 12. Best individual across all runs in experiment 2 when run on test images. The filter area was 7x7.

TABLE VIII
FILTER AREA 7X7 TESTING SET BEST PERCENTAGE HITS ON TEST IMAGE 1

% of Hits		Actual Answer	
		L	NL
GP	L	0.9799197	0.078797795
	NL	0.02008032	0.9212022

Table VIII: Refer to figure 17 in appendix for corresponding performance image

TABLE IX
FILTER AREA 7X7 TESTING SET BEST PERCENTAGE HITS ON TEST IMAGE 2

% of Hits		Actual Answer	
		L	NL
GP	L	0.9914738	0.0879716
	NL	0.008526187	0.91202843

Table IX: Refer to figure 18 in appendix for corresponding performance image

us if there is any significance between the means of our filters. Our null hypothesis is that the samples filter areas means are equal. Our alternate hypothesis is that at least one sample filter area mean is different from the other sample means. There is very little significant evidence (p = 0.179215) in support of the alternate hypothesis. Therefore, we are unable to deny the null hypothesis, hence there is no significant difference between the sampled means for our filter areas.

A. Filter Area

An ANOVA test was used to test for significant difference between the sampled population average fitness means for our various filter areas tested: 3x3, 7x7, and 11x11. Our null hypothesis is that the samples' filter areas means are equal. Our alternate hypothesis is that at least one sample filter area mean is different from the other sample means. There is very little significant evidence (p = 0.179215) in support of the alternate hypothesis. Therefore, we are unable to deny the null hypothesis, hence there is no significant different between the sampled means for our filter areas.

B. Weight

An ANOVA test was used to test for significant difference between the sampled population average fitness means for our tested true positive weight values. The null hypothesis is such that the weights' sampled means are all equal. The alternate hypothesis is such that at least one weight's sampled mean is different from the rest of the weight sample means. There is very strong evidence ($p = 5.029\,01 \times 10^{-25}$) in support of the alternate hypothesis. Therefore, we can conclude that there is at least one weighted sample mean different than another within the sample means.

Unpaired t-tests with unequal variance were performed on the different combinations of sample means from the previous ANOVA test. We will assume the distribution is normal. We will say that for all t-tests conducted the null hypothesis is that the sample means are equal and the alternate hypothesis is that they're not equal. The sample means are based on the population's average fitness over our 10 runs.

1) Weight=1 & Weight=100: There is very little significant evidence (p = 0.535689863) in support of the alternate hypothesis. Therefore, we are unable to reject the null hypothesis, meaning there is no significant difference between our sampled means for a weight value of 1 and a weight value of 100.

2) Weight=1 & Weight=250: There is very strong evidence (p = 0.000145053) in support of the alternate hypothesis. Therefore, we can accept the alternate hypothesis that there is a significant difference between our sampled means comparing a weight of 1 to a weight of 250.

3) Weight=1 & Weight=500: There is very strong evidence ($p = 1.899\,35 \times 10^{-12}$) in support of the alternate hypothesis. Therefore, we can accept the alternate hypothesis that there is a significant difference between our sampled means comparing a weight of 1 to a weight of 500.

4) Weight=100 & Weight=250: There is very strong evidence ($p = 9.398\,06 \times 10^{-9}$) in support of the alternate hypothesis. Therefore, we can accept the alternate hypothesis that there is a significant difference between our sampled means comparing a weight of 100 to a weight of 250.

5) Weight=100 & Weight=500: There is very strong evidence ($p = 4.997\,17 \times 10^{-20}$) in support of the alternate hypothesis. Therefore, we can accept the alternate hypothesis that there is a significant difference between our sampled means comparing a weight of 100 to a weight of 500.

6) Weight=250 & Weight=500: There is very strong evidence ($p = 3.657\,58 \times 10^{-26}$) in support of the alternate hypothesis. Therefore, we can accept the alternate hypothesis that there is a significant difference between our sampled means comparing a weight of 250 to a weight of 500.

VII. CONCLUSION

Using GP for computer vision has yielded some interesting conclusions. It is highly dependent on the types of images being used. The more complicated the image the worse GP will perform. GP does well at recognising patterns, our results are a good representation of that due to our images having a pattern based object to search through.

Since the lightsaber is a straight line, similar to images of meteors in the night sky, the GP could find it with a small filter radius. If the image did not have this pattern, it would have needed to see the entire object or it would get confused easily. This was shown in our results because from testing the different filter sizes there was no significant difference in the average performance. However, there was a notable performance difference between our results for the different weighted true positive values.

Due to lightsabers only taking up a small percentage of the pixels within the image we found that we needed to use a weighted sum for fitness. Otherwise, considering that the actual negative pixels outnumber the actual positive pixels 99 to 1, a solution that classifies every pixel as negative would be correct 99% of the time. This would account for why the best individual in the weight=1 experiment did not change over the course of the run. The lower weight we gave to the true positive the more it would avoid classifying lightsaber pixels and if the weight was too high, GP would result in over-classifying lightsabers. It was shown statistically above that adding some weight to our true positives improved performance. The convergence fitnesses for training also did not indicate how well that evolved solution performed on testing. The best individual from weight=1 for example had a better fitness in training than any of the other solutions but it performed the worst on testing since it would classify no pixels as belonging to lightsabers. These results suggest that weighting positive pixels could be successful when applied to similar computer vision problems where actual positive pixels take up a very small percentage of the image.

It also should be noted that the set of images used are relatively easy for the GP to solve since the lightsaber's pixel intensity is so bright. The main scenario in which it struggles is when the Jedi are wearing very light clothing and the lightsaber is crossing it within the image. It can be seen from the weight analysis that the percentage of your image that are positive pixels will drastically determine how to weight your fitness values in relation to confusion matrix values in testing images. A lower proportion of positive pixels should have a higher weight associated with their correct identification. Perhaps we should leave image classification to the neural nets but it is still interesting to see how GP works with computer vision problems.

References

[1] George Lucas, director. *Star Wars Episode II: Attack of the Clones*. 2002.

[2] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT PRes, 1992.

[3] Sean Luke et al. *ECJ 27: A Java-based Evolutionary Computation Research System*

Appendix

A. Testing Images

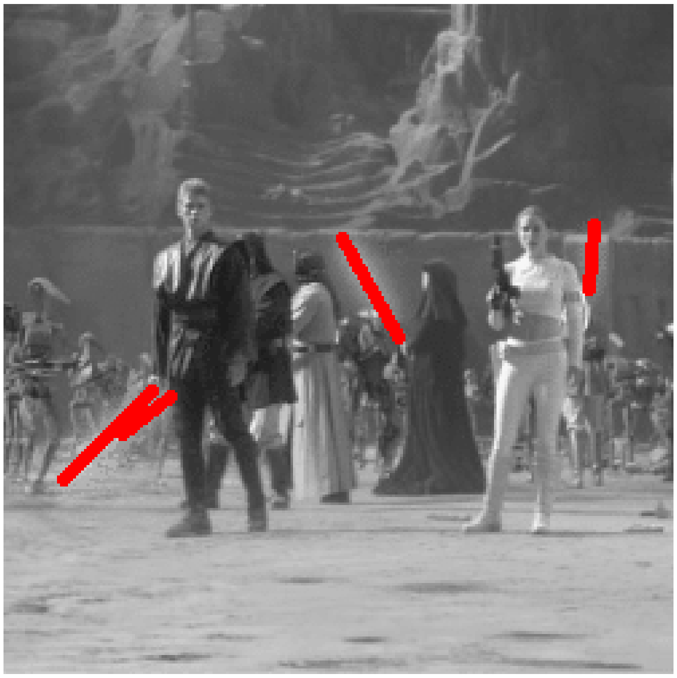


Fig. 13. Ground Truth Test Image 1

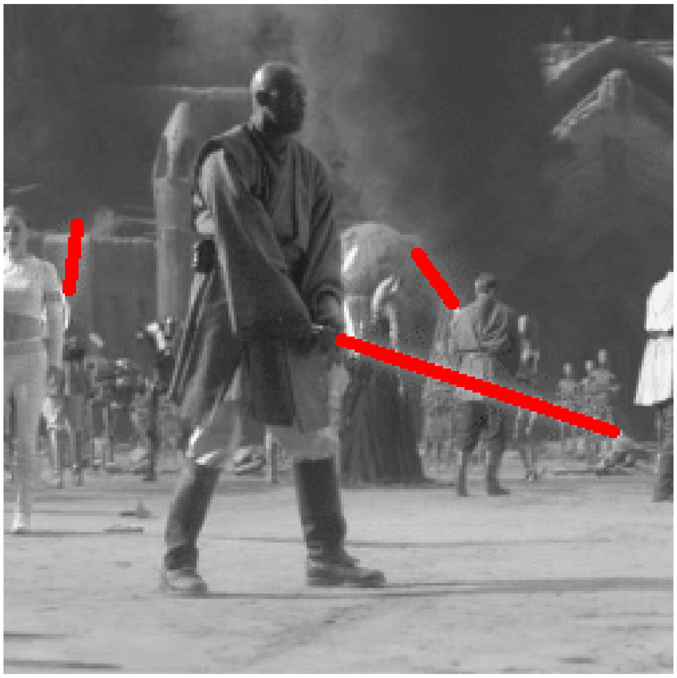


Fig. 14. Ground Truth Test Image 2

B. Performance Images for Best Runs

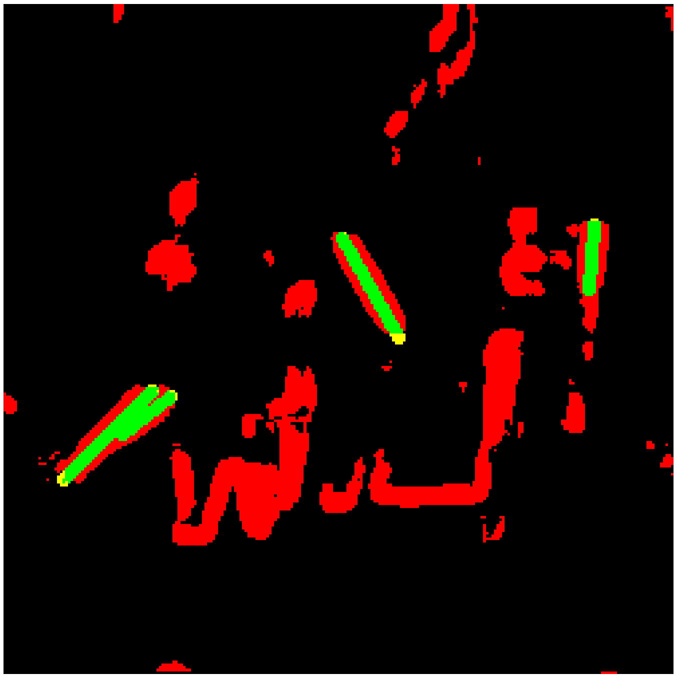


Fig. 15. Best performance for true positive weight 250 on test image 1

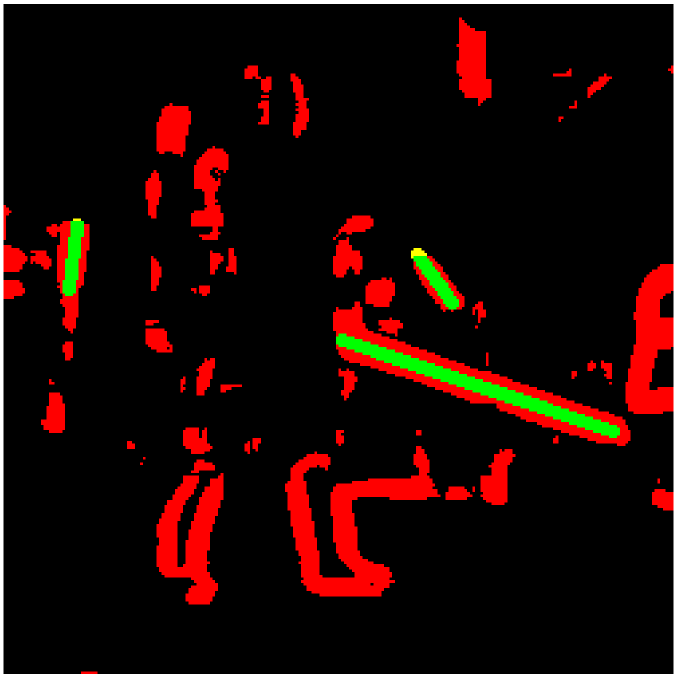


Fig. 16. Best performance for true positive weight 250 on test image 2

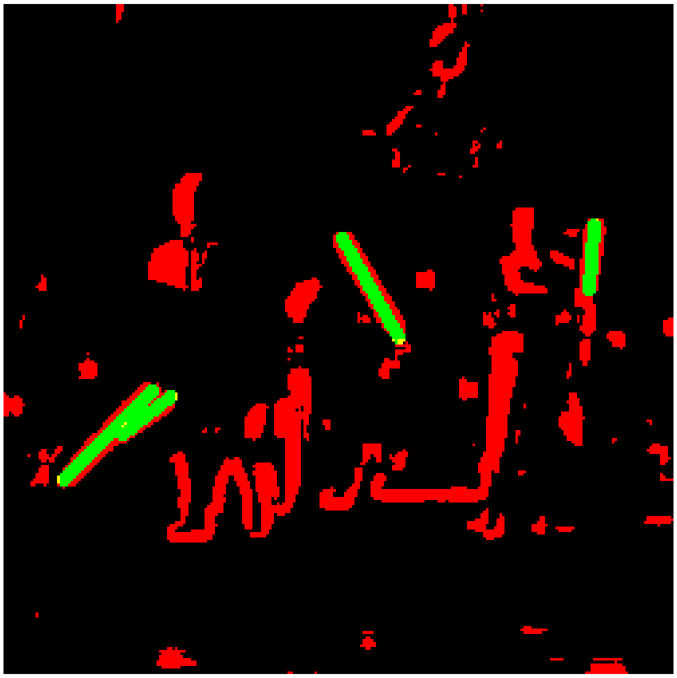


Fig. 17. Best performance for filter area 7x7 on test image 1

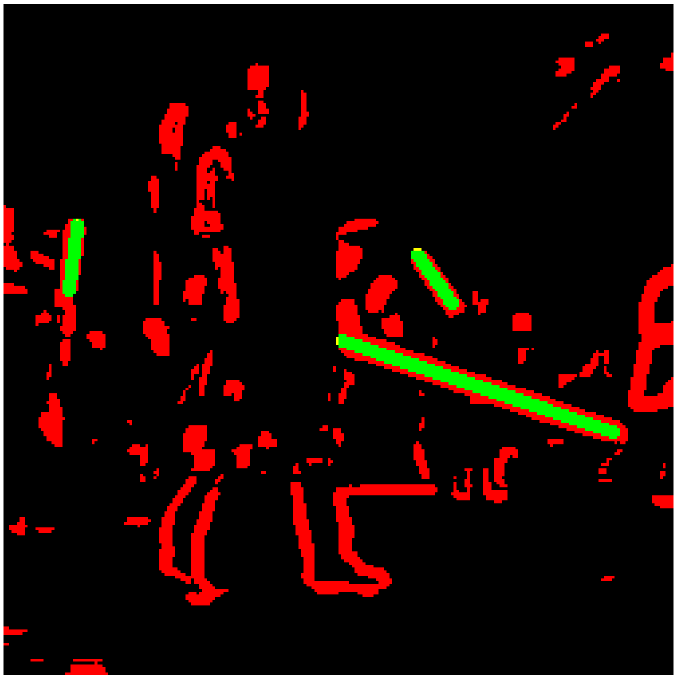


Fig. 18. Best performance for filter area 7x7 on test image 2