Image Segmentation of Boats using Genetic Programming

Nick Aksamit
Department of Computer Science
Brock University
St. Catharines, Canada
na16dg@brocku.ca

I. Introduction

Image segmentation is a method of extracting needed regions from an image, in a process that is usually automatic or semi-automatic. There are many applications for such a problem, with a main source emanating from medical care, or more specifically, medical imaging [1]. Many modern image segmentation methods employ various intuitions obtained from convolutional neural networks, autoencoders, and long-short term memory [2]. Examples of such systems are SegNet [3], U-Net [4], and OCNet [5]. These systems have proven themselves beneficial in segmenting images.

Unlike the previously mentioned deep learning techniques, Genetic Programs (GPs) has also been used for image segmentation [6] with sufficient results. GP is an evolutionary algorithm whereby solutions are represented as a tree of functions and terminals. Genetic operators are exercised to slightly modify trees throughout many iterations which are referred to as generations. When compiled, a solution takes the form of a program that can be executed to obtain a solution. With supplemented image filters, GP can be used to develop a program that segments an image into previously defined regions of interest. In this work, a comparative study is performed on various alterations to the GP function set, and parameters. Two images of boats are applied to the GP system. One is notably more challenging given the presence of a dock. Experiments are done in a sequential matter, whereby a found enhancement in outcome is employed in future trials.

The following work is organized as follows: Section II contains information pertaining to how experiments were conducted. Afterwards, all results and discussion can be found in Section III. Finally, generalized examination is delivered in the concluding Section IV.

II. Experimental Setup

This section contains information that would be necessary for replication of results. However, due to the stochastic nature of GP, findings may not be perfectly reproduced. A Python GP system named DEAP [7] was used for all experiments in this study.

This section is segmented into the following sections: II-A explains the images used along with their respective coordinates for training and testing. Subsequently, Section II-B describes the GP parameters used throughout experimentation. Following is Section II-C where the function set and terminal set are illustrated. Afterwards, Section II-D defines the fitness function, which is how solution quality is defined. Finally, all additional information needed to interpret the results discussed can be found in Section II-E.

Image	Type	(xMin, yMin, xMax, yMax)
1	Training	(28, 55, 276, 233)
1	Testing	(996, 210, 1211, 442)
2	Training	(1098, 610, 1792, 981)
2	Testing	(289, 306, 810, 726)

TABLE I
Coordinates of Testing and Training Regions

A. Image Data

Two images were obtained from different locations for the purpose of use in image segmentation. The goal of the GP system is to properly classify the pixel of a boat. A classification image was created from the two original images with hand-drawn positive classification regions. In addition, training and testing regions are allocated for their respective purpose in creating a GP model for image segmentation.

Figure 12 in Appendix A depicts the first image used for segmentation, and was acquired from [8]. Subsequently, Figure 13, attained from [9], portrays the second image. The bright purple regions express positive classification, with the rest denoting negative. Likewise, a green box represents the training region, while a red box indicates the testing region. It should be noted that while the images are coloured, not all experiments use RGB images. Rather, this only occurs in experiments 3 and 4. In addition, image two was not included for experimentation until sets 3 and 4 as well. All experiments that do not use RGB strictly employ a grayscale clone of the attached images. The coordinates of the expressed green box and red box are found in Table I, and are useful in calculating the fitness and generating a performance image.

B. GP Parameters

The main objective of this work is to determine the affect a change of GP parameters and function set has on the outcome of a given model. It has been shown that different parameter configurations can affect the GP system in a multitude of

Parameter	Value
Generations	100
Crossover Rate	90%
Mutation Rate	10%
Number of Executions	15
Number of Elites	1
Min Tree Size (Init.)	2
Max Tree Size (Init.)	4
Min Tree Size (Mut.)	1
Max Tree Size (Mut.)	2
Tournament Size	3

TABLE II
DEFAULT GP PARAMETERS

ways, with some producing better results than others [10]. That being said, not all parameters have been altered. In Table II, the parameters that remain static throughout all experiments are shown. Modified parameters within trials include the number of samples taken from the image, and population size. An increase to both of these criterion values can lead to better quality solutions, but at the drawback of an increased computational cost.

There are other parameters that remain static but are not located in the aforementioned table. Selection of individuals for the next population was done using the tournament process, where a size of 3 is selected. One-point crossover is performed, where a random node is selected on two individuals, and the subtree at that location is swapped. Likewise, for mutation a random node is again selected, but instead an entirely new subtree is generated. This subtree is created to be within a certain size range, as can be seen in Table II. In order to initialize the GP system, an initial population must first be generated. The half-and-half approach is used for this purpose, where there is a 50% change of either a full or grow tree generation. The population size is not listed in Table II as the last experiment has its value changed to 750. However, for all other experiments a value of 300 is used.

C. GP Language

Similarly to the GP parameters, utilizing a different language may make the process of finding the optimal solution easier. For example, in symbolic regression of x^3, it is much easier to include a power function into the language, rather than solely relying on multiplication, or even summation. For the purposes of this work, the language is also altered to view how this may influence results.

Each individual is interpreted as a tree, with functions for nodes and terminals for leaves. Each function has a certain arity, which determines the number of child nodes it must possess. In contrast, terminals are a singular value that either remain static, or change if it takes the form of an argument. This work does not utilize any arguments at the tree level, and so only static terminal values are employed. Many of the exercised functions are image filters, most of which were generated from the Pillow library [11]. Only a single filter is generated by Adobe Photoshop, and it is denoted as EdgesF.

In experiments, the primitive set, which encompasses both function and terminals, have three revisions. All of these may be viewed in Appendix B, where bolded entries illustrate a change from the previously completed trial. Table VIII and IX express the function and terminal set for experiment 1, respectively. Afterwards, Table X shows the function set for test 2, while Table XI does the same but with the third. The terminal set does not change between experiments 2 and 3, and their respective entries can be seen in Table XII.

In Table X, the arities of each filter function is increased to allow for x and y offsets as arguments. Additionally, the EmF and EdgesF filters are added as testing showed increased performance with their addition. This was likely due to their ability to highlight edges, but reduce complexity in all other areas. Table XI exhibits four new functions that allow for the use of RGB coloured images rather than grayscale. It is notable that Table XII includes terminals which ensure a fixed filter size. MeanF, MaxF, and MinF all require a size argument, but only accept integer values 1, 3, 5, 7, 9, and 11. With the included terminals it ensures that an invalid integer is not provided for the aforementioned filters, while still allowing the use of ephemeral constants that work alongside the needed offset arguments.

D. Fitness Function

Given the task of image segmentation, the primary objective is to classify pixels into various predefined categories. In the case of this work, these categories are either a boat or not a boat. To save a large amount of computation when training, the entire image is not used for training every individual. Rather, true and false classifications are segregated into separate sets. Then, an already decided set number of samples are randomly gathered from each set and used for calculation of the fitness. This was done to ensure true classifications have a fair chance at being chosen, as in Figures 12 and 13, there is a vast amount of false classification pixels in comparison with true.

Algorithm 1 Fitness

```
Require: x be a compiled program
  hits ← 0
  for i from xMin to xMax do
      xPos ← i
      for j from yMin to yMax do
          yPos ← j
          if x ≥ 0 and classification = True then
              hits ← hits + 1
          else if x < 0 and classification = False then
              hits ← hits + 1
          end if
      end for
  end for
  return hits
```

Algorithm 1 demonstrates the pseudocode used for calculation of fitness during training. yPos and xPos are both global variables that are altered to obtain the correct value from the

Classification	Colour
True Positive	Green
True Negative	Black
False Positive	Red
False Negative	yellow

TABLE III
PERFORMANCE IMAGE COLOURATION

various filters that may be used by the individual. xMin, xMax, yMin, and yMax are the containing box of the training image. This allows for only a portion of an image to be used, as is seen on the aforementioned figures. It would also allow for future work to use multiple training containers on the same image, rather than one. For testing the same process is used, but the container values (xMin, xMax, yMin, yMax) are modified. In addition, when testing all pixels within the container are used, unlike the random sampling done during training. Coordinates used throughout the experimentation process for both images can be found in Table I.

E. Objective and Additional Information

The main objective of this work is to segment an image of boats into separate classifications. In this case, it is a binary segmentation of two categories: boat and no boat. Two images are used, but as mentioned in Section II-A, only the first is used for experiments 1 and 2, and then both for the remaining. It should be noted that the two images are not the same, as one contains boats stationed at a dock, and the other boats in open sea. With the dock appearing similar in colour to boats, it may be difficult to differentiate between both. Thus, image one is slightly more difficult for segmentation than image two.

In addition to a confusion matrix, a performance image is used as output of testing. The performance image visually illustrates a true positive and negative, along with false positive and negative. At most four colours are shown on such an image, each representing a different classification. Table III differentiates each colour from their corresponding meaning. Statistical testing is completed to compare between results in each experiment. First a Shapiro-Wilk test is completed as a normality check, followed by a Student's t-test or Mann-Whitney U test, depending on the normality of the returned samples. All statistical tests are completed at a 95% confidence level, with 15 independent executions completed on each configuration.

A total of four trials are performed. The first consists of testing different number of samples taken during training. When considering known positive to known negative, the amounts used are: 10-50, 50-10, 100-10, 10-100, 100-100. Afterwards, offsets are added into the function set and tested for their efficacy. In experiment three, additional modifications are made to the function set and terminal set to allow for the use of RGB images. The purpose is to determine if segmentation on RGB imagery is better than grayscale. Lastly, the population size is altered to determine if additional individuals enhance the result.

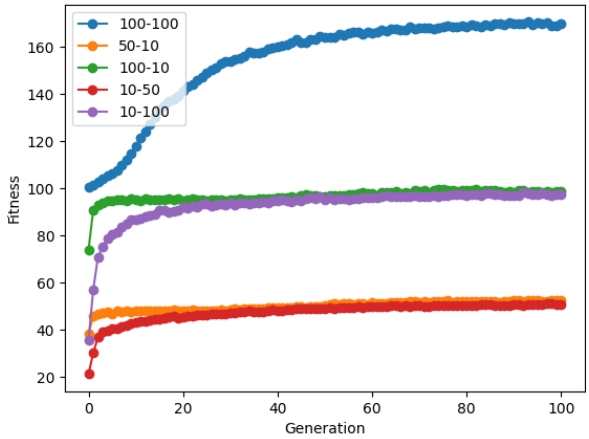

Fig. 1. Average Generational Fitness by Sampling Sizes (15 executions)

III. RESULTS

In this section the results of all experimentations are shown and discussed. Summary tables of statistical results are provided for concise observation of outcome quality. Sections are divided into their corresponding experimentation set, starting with sampling sizes, followed by offsets, image colouration, and lastly, population size.

A. Sampling Sizes

Experiments were conducted to inspect how the GP system reacts to a different number of image samples. The values used for sampling between known positive and negative are: 10-50, 50-10, 100-10, 10-100, 100-100. Figure 1 illustrates the average generational performance during training, on all the aforementioned sampling sizes. It is clear that using more samples in general increases the overall average fitness of executions, as is seen with 200 samples at the top (blue), 110 in the center (green, purple), and 60 at the bottom (red, orange). Another noticeable factor is the rate of improved fitness.

When sampling is done with greater priority on the positive classifications, fitness rises at a much greater pace, as compared with a larger priority on negative. This may not be the case with all image segmentation tasks, as it likely correlates with how the image is known to be segmented, along with its complexity. In the case of Figure 12, it is a binary classification, there are more negative classifications than positive, and most negative classifications are similar in features. Thus, sampling from the positive in this case may lead to understanding its features better, and interpreting those that do not follow as negative.

The Shapiro-Wilk normality test is applied to the outputs of each result. It was found that only 100-100 is statistically normal, and so the Mann-Whitney U test is applied to examine any deviance in medians. After application, the best sampling size amongst those used is 100-100. Likewise, 100-10 and 10-100 are statistically better than 10-50 and 50-10, but not amidst

Ranking	Sampling Size
1	100-100
2	100-10
2	10-100
3	50-10
3	10-50

TABLE IV
RANKING OF SAMPLING SIZES

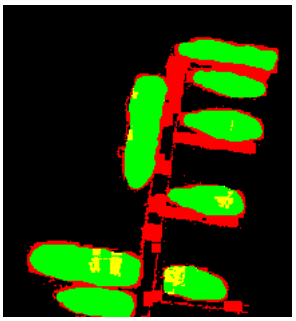

Fig. 2. 100-100 Sampling Size Performance Image

each other. Lastly, 10-50 and 50-10 are not better than any of the other sampling sizes used. These statistical results do not differ from what is observed in Figure 1. Using more samples in general leads to a better outcome, while the variation in positive to negative samples may just coincide with the rate of achieving it. The statistical ranking of sampling size and performance of GP is summarized in Table IV.

Figure 2 presents the performance image of the best individual from using 100-100 samples. Many of the boats and surrounding water are correctly classified, however some portions of the dock remain incorrectly segmented. Not only this, but the edges of the boat are incorrectly classified as well. However, due to the true classifications being produced by hand, the ring of red pixels surrounding boats can be ignored if not too thick, as it might be human error. On the other hand, misclassification of the dock is a true mistake that GP was not able to overcome in this set of experiments.

B. Addition of Offsets

Further experiments were completed to analyze the effect offsets have on the outcome of this image segmentation task. Offsets mean that rather than take the pixel value of the current position, it might be an advantage to know the value of a pixel above, below, diagonally, etc. Before the addition of offsets, unshown testing was completed to determine if two additional filters would provide better results. They were deemed beneficial statistically, and so they are included, along with offsets, in Table X.

Figure 3 portrays the average, along with average best generational fitness by usage of offsets. Although it may intuitively be thought of as beneficial, the plot illustrates that there is little difference in this modification. Due to little diversity between the average generational fitness of the two plots, the best generational fitness for both is also provided.

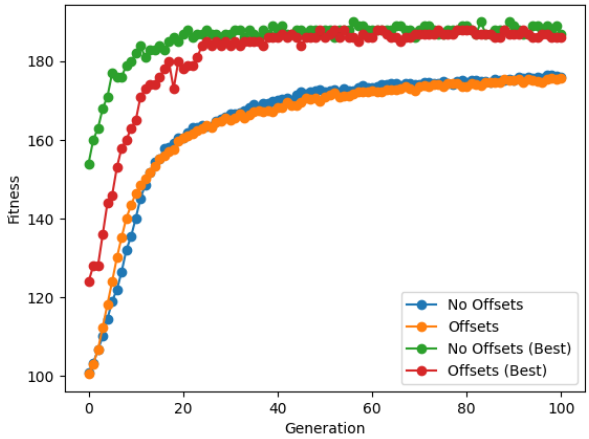

Fig. 3. Average Generational Fitness by Offset Usage (15 executions)

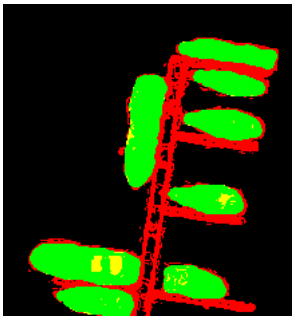

Fig. 4. Offsets Performance Image

When viewing the best, it appears that using no offsets rises much quicker than including it. However, they both reach similar values nearing the final generations.

The Shapiro-Wilk test is used to test for normality, and neither the results from using offsets, nor without are statistically normal. The Mann-Whitney U test is used for ensuring any differences in median values. Between the two experiments performed, there is no statistical difference in result. Thus, using offsets did not provide the benefit intuitively thought. It should be noted that without using offsets, the p-value returned from the Mann-Whitney U test is 0.08, close to the 0.05 threshold. However, with future alterations to the function set to allow for coloured images, using offsets may add benefits. For this reason, the use of offsets is continued for the rest of the experiments in this work. Statistical results of the Mann-Whitney U test are outlined in Table V.

Test	p-value
Offsets > No Offsets	0.08
No Offsets > Offsets	0.92

TABLE V
OFFSETS COMPARISON (MANN-WHITNEY U)

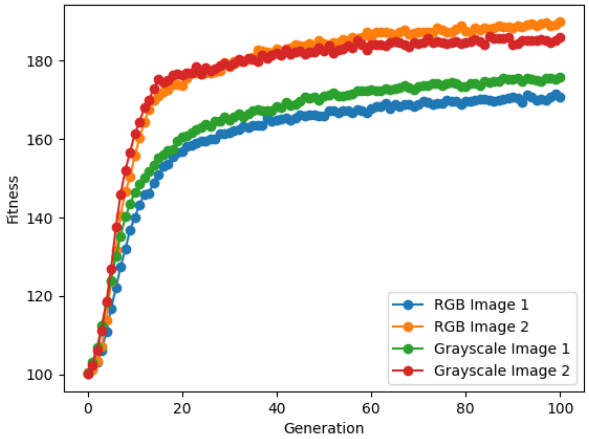

Fig. 5. Average Generational Fitness by Image Colouration (15 executions)

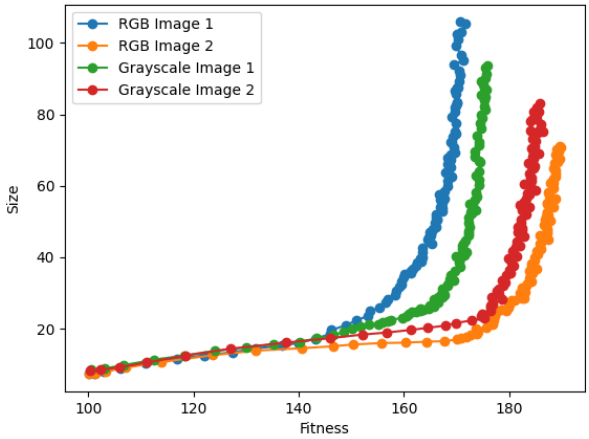

Fig. 6. Average Size and Fitness by Image Colouration (15 executions)

Figure 4 illustrates the corresponding performance image of the best result from using offsets. Upon comparing with Figure 2, it is apparent that some improvements are made, along with some points of incorrect segmentation. The dock is noticeably grainy with red colouration, indicating false positives. Even with the inclusion of offsets, the dock still remains a challenging factor in segmenting this selected image. Perhaps this will be reduced in the following experiment where RGB imagery is used.

C. Grayscale and RGB

In this experimentation set, additional filters are included to allow the use of RGB imagery. All additional entries to the function set can be viewed in Table XI. Moreover, a second image (Figure 13) is used for the first time in this work. As previously mentioned in Section II-E, it may be easier to segment the second image as its boats are stationed in open water rather than near a dock that is similar in colour. As such, fitness scores during training and testing are expected to be higher than for the first image that has been used previously.

The aforementioned notion seems to be correct for training, as in Figure 5, image two is seen distinctly above both plots for image one. At the same time, the results when comparing RGB and grayscale are swapped between both images. For image one, it appears that on average, using grayscale provides better results. For image two, it is RGB that gives better results on average, over that of grayscale.

Bloat is a factor that may punish GP with unnecessary additions to an individual. Some combinations of functions in the function set may give no result, and so are not needed. A simple case of this is an event such as multiplication of any number by zero, which from the implementation of this work would make two nodes redundant, namely the multiplication, and the other number. A simple check to view any potential bloat is a plot of average size and average fitness. If fitness is on the x-axis, and size on the y-axis, and some set of values are found to be vertically linear, then bloat is likely present.

Image	Test	p-value
1	RGB > Grayscale	0.72
1	Grayscale > RGB	0.29
2	RGB > Grayscale	**0.001**
2	Grayscale > RGB	0.99

TABLE VI
IMAGE COLOURATION COMPARISON (MANN-WHITNEY U)

If the fitness are size are on swapped axes, then horizontal linearity expresses the same.

Figure 6 illustrates such a plot with all the performed trials in this experiment. Clearly visible are vertically linear sets of points on all four executed trials. However, verticality is stressed on the plots for image one. Even with the seemingly good performance on image two from Figure 5, it appears that it is still susceptible to bloat. Future work should focus on reducing this bloat, as it may improve the efficiency of GP for image segmentation, and perhaps may improve the misclassification of the dock in image one.

Statistical testing is completed to determine superiority of the two variants, with results summarized in Table VI. Using the Shapiro-Wilk test, only the use of RGB on image one is normal, and so the Mann-Whitney U test is used for difference of medians. Despite their discernible contrast in Figure 5, neither use of RGB nor grayscale on image one is statistically better than one another. In training the performance of grayscale was on average superior, but overall there is no statistical difference with the use of RGB. Although grayscale was on average superior, RGB is used in the final set of experiments. Similarly to the offsets experimentation, RGB has the ability to differentiate between three colour channels, whereas in grayscale there is only one channel. This introduces a complexity that did not drastically reduce performance, but rather maintained one similar to when a single channel is employed. However, usage of three channels can allow for more complex solutions, which may allow for

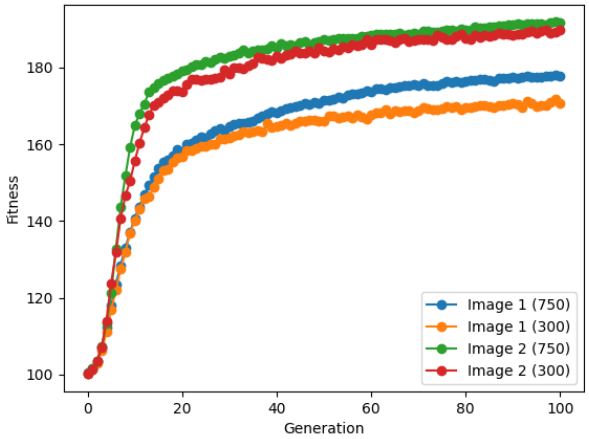

Fig. 7. Average Generational Fitness by Population Size (15 executions)

Image	Test	p-value
1	750 > 300	**0.002**
1	300 > 750	0.99
2	750 > 300	0.26
2	300 > 750	0.73

TABLE VII
POPULATION SIZE COMPARISON (MANN-WHITNEY U)

better segmentation results. On image two, it is RGB that is statistically better than grayscale. RGB will also be used in the final set of experiments on population size.

D. Population Size

The final set of experiments in image segmentation using GP comes from modification of the population size. With an increased population, more individuals compete to find an optimal solution, but at the cost of additional computational time. Thus, it is envisaged that increasing the parameter in this experimentation will lead to increased fitness during training, and better a better quality of outcome.

Figure 7 outlines the average generational fitness of the four trials. All executions operated with RGB imagery, offsets, and 100-100 sampling sizes. Similarly to the previous experiments, fitness on image two is much higher than that of image one, even with the change in population size. Peculiarly, however, there is clearer variation between the two trials on image one rather than two. This leads to the belief that in training, increased computation for image two is not needed, as results found before were already adequate. The same cannot be said for image one.

Following the procedure of all experiments, the Shapiro-Wilk test found none of the gathered outcome sets to be normal, and so the Mann-Whitney U test is exercised. On image two, using a population size of 300 is not significantly better than using 750. This result persists from the interpretations communicated before from Figure 7. Another notion that persists is the discrepancy between average fitness between

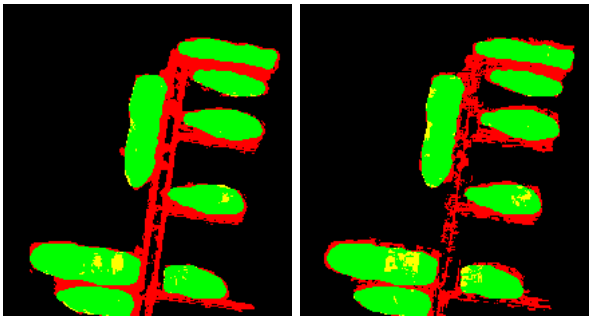

Fig. 8. Performance Image of Image One on Population Size (300 left; 750 right)

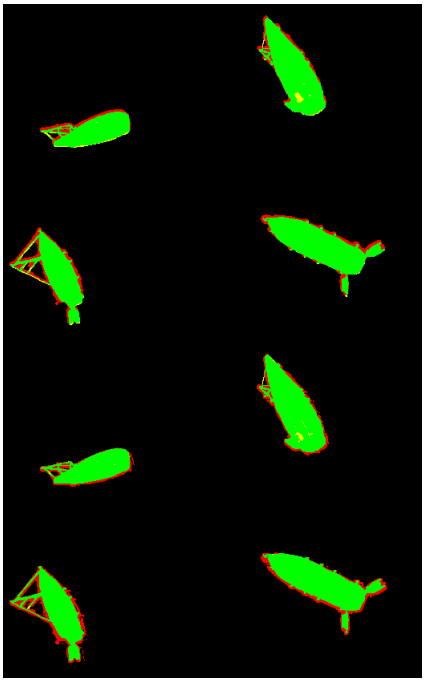

Fig. 9. Performance Image of Image Two on Population Size (300 top; 750 bottom)

population sizes on image one. Conclusions from the statistical test indicate that having the larger population size of 750 performed much better than 300. Unlike how image two did not require additional computation, segmentation on image one was enhanced by more. This may be attributed to the presence of the dock that is lacking in the second image, which from previous experimentations was shown to cause difficulties in classification of pixels. The discussed results are generalized in Table VII.

Performance images for both image one (Figure 8) and two (Figure 9) are shown. For image one, the boost in population noticeably increases the number of correct classifications with respect to the dock. The performance image corresponding to 750 population has many more grainy red pixels. While an improvement over previous experiments, it also means that more work must be done to correctly classify boats when located near a dock. Since boats are regularly spotted attached

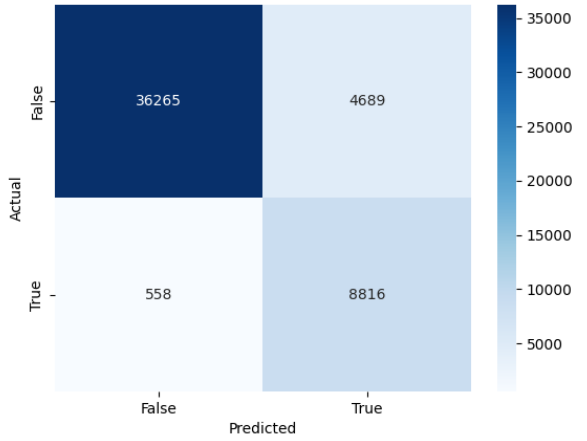

Fig. 10. Confusion Matrix of Best Individual (Image One)

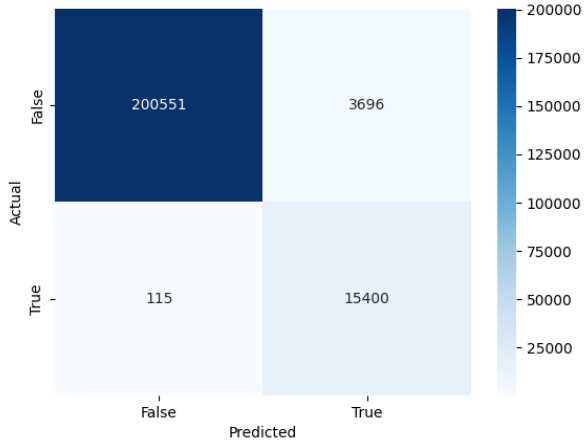

Fig. 11. Confusion Matrix of Best Individual (Image Two)

to docks when not in use, this may be an important and challenging task. On the contrary, there are minor divergences between the performance images for image two (Figure 9). When using a higher population, there are less false positive classifications surrounding the edge of the boats. However, these classifications may be correct, as the ground-truth images were filled in by hand, and are subject to human error in what is considered a boat. It can be said that the GP model managed to suitably segment boats from image two.

In addition to performance images, the confusion matrix of the best individuals for images one and two are illustrated in Figures 10 and 11, respectively. Initially it can be seen that image two has many more overall pixels for classification, in addition to more in both positive and negative, given that it is a larger image than one. It is of note that for image two, there are considerably less incorrect categorizations made. 3696 false negatives are achieved, and 115 for false positives. This is compared with image one where there are 4689 false negatives and 558 false positives. Thus, even with image one being smaller, more incorrect classifications are made during testing. This follows the notions found during training, as the fitness for image two was consistently found above one. This also further points towards the obstacle faced during training of image one, namely the dock. Given the much larger amount of false positives, there is still work to be done in reducing this quantity.

Both of the best individuals used for the performance images and confusion matrices are seen in Appendix C as Figures 14 and 15. Comparing the two individuals, sizing becomes a discernible contrast. The best solution to image one is much larger than that of image two, and contains some bloat. Meanwhile, image two has a concise solution that is able to correctly classify a large majority of pixels. As has been mentioned ad nauseam, the solution for image one may be of such size due to searching for trees that improve upon misclassifications of the dock. Although improvements have been made with various experiments in this work, it still remains an aspect that challenges good performance.

IV. Conclusion

In this work, genetic programming is attempted for the image segmentation problem. A total of two images were used for classifying boats, with one noticeably more challenging than the other, due to the inclusion of a dock with similar colours to the attached boats. Classification areas of both ground truth images were created by hand, and both images were obtained from different sources (Image 1: [8], Image 2: [9]).

A total of four sets of experimentations are completed in attempt to sequentially improve upon solutions. The first study encompassed how different sampling sizes of images during training affect the testing result. As is intuitively thought, using a larger sampling size in general improves upon solutions better than lower. Sampling is completed from a set number of known positive and negative classifications. A higher amount of positive samples led to a quicker rise in fitness, but the fitness score plateaued to a similar value as when there are the same amount of samples, but more are negative. Thus, whether using more positive or negative did not statistically affect the outcome, but a larger overall sampling size did.

The second set of trials focused on if using offsets could improve solutions. A statistical test of medians did not express any difference in conclusions, and so it cannot be said that offsets improved segmentation, nor worsened. However, with their added complexity, offsets could also lead to having better solutions with more modifications to the function set, and so they were kept during the later trials.

Subsequent experiments are concerned with the usage of RGB images in comparison with grayscale. For the first time in this work, another image was introduced at this stage to give further insight into results. For the newly introduced image two, there was a statistical improvement in using RGB over grayscale, however the opposite is true for image one. RGB did not offer any vast improvement in solution quality. Due to

RGB being useful for image two, it was utilized again in the final stage of experimentation.

Throughout a large portion of previous trials, the function set was changed to determine if this could lead to better conclusions. In the final set of research findings, the population size of GP was increased, as more individuals may strengthen the probability of finding good solutions. This notion occurred for image one, but not image two. As was mentioned earlier, image one is more challenging because of a similar-coloured dock. This problem was slightly mitigated with more individuals, but still not fully solved as from Figure 8, the dock still appears to be falsely classified as positive.

Overall, it was seen that sampling and population size both negatively affect GP if their values are too low. However, for some problems that are already solved with less individuals, such as the case on image two, a higher population will not improve the quality of solutions by much. Offsets, and use of RGB imagery may be more image-dependent. Their added complexity might be useful in future work that extends their utilization with additional features.

It is recommended that in future studies a greater extent of effort is put into classification of boats that are near some objects of similar colour, such as a dock. While the GP system implemented in this work managed to successfully classify boats in open water, it suffered gravely when attempting a problem where a dock is present.

Acronyms

GP Genetic Program
GA Genetic Algorithm

References

[1] A. Norouzi, M. S. M. Rahim, A. Altameem, T. Saba, A. E. Rad, A. Rehman, and M. Uddin, "Medical image segmentation methods, algorithms, and applications," *IETE Technical Review*, vol. 31, no. 3, pp. 199–213, 2014.

[2] S. S. Chouhan, A. Kaul, and U. P. Singh, "Soft computing approaches for image segmentation: a survey," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28483–28537, 2018.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[5] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.

[6] A. Khan, A. S. Qureshi, N. Wahab, M. Hussain, and M. Y. Hamza, "A recent survey on the applications of genetic programming in image processing," *Computational Intelligence*, vol. 37, no. 4, pp. 1745–1778, 2021.

[7] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "Deap: Evolutionary algorithms made easy," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2171–2175, 2012.

[8] Google, "Port dalhousie," Accessed March 2022. [Online]. https://earth.google.com/web/@43.20809705,-79.26093972,76.76673998a,743.05918723d,35y,0.66273159h,1.75803608t,0r.

[9] SeaShare, "Cover photo," Accessed March 2022. [Online]. https://m.facebook.com/sharethesea/photos/a.111104947270638/312550730459391/?type=3source=44ref=page_internal.

[10] E. B. De Lima, G. L. Pappa, J. M. de Almeida, M. A. Gonçalves, and W. Meira, "Tuning genetic programming parameters with factorial designs," in *IEEE congress on evolutionary computation*, pp. 1–8, IEEE, 2010.

[11] A. Clark and Contributors, "Pillow library documentation," Accessed March 2022. [Online]. https://pillow.readthedocs.io/en/stable/index.html.

Appendix A
Images

Fig. 12. Standard Image

Fig. 13. Standard Image (2)

Appendix B
Function and Terminal Sets

Function	Arity	Description
ADD	2	x + y
SUB	2	x - y
MUL	2	x * y
DIV	2	x / y
NEG	1	-x
MAX	2	max(x, y)
MIN	2	min(x, y)
MeanF	1	MeanFilter(x)
MaxF	1	MaxFilter(x)
MinF	1	MinFilter(x)
EdgeF	0	EdgeFilter()
EdgePF	0	EdgePlusFilter()

TABLE VIII
Function Set (Experiment Set 1)

Terminal	Value
Ephemeral Constant	$x = \text{rand}(-5, 5), x \in \mathbb{R}$
Integer	1
Integer	3
Integer	5
Integer	7
Integer	9
Integer	11

TABLE IX
TERMINAL SET (EXPERIMENT SET 1)

Function	Arity	Description
ADD	2	x + y
SUB	2	x - y
MUL	2	x * y
DIV	2	x / y
NEG	1	-x
MAX	2	max(x, y)
MIN	2	min(x, y)
MeanF	**3**	**MeanFilter(i, x, y)**
MaxF	**3**	**MaxFilter(i, x, y)**
MinF	**3**	**MinFilter(i, x, y)**
EdgeF	**2**	**EdgeFilter(x, y)**
EdgePF	**2**	**EdgePlusFilter(x, y)**
EmF	**2**	**Emboss(x, y)**
EdgesF	**2**	**EdgesFilter(x, y)**

TABLE X
FUNCTION SET (EXPERIMENT SET 2)

Function	Arity	Description
ADD	2	x + y
SUB	2	x - y
MUL	2	x * y
DIV	2	x / y
NEG	1	-x
MAX	2	max(x, y)
MIN	2	min(x, y)
MeanF	3	MeanFilter(i, x, y)
MaxF	3	MaxFilter(i, x, y)
MinF	3	MinFilter(i, x, y)
EdgeF	2	EdgeFilter(x, y)
EdgePF	2	EdgePlusFilter(x, y)
EmF	2	Emboss(x, y)
EdgesF	2	EdgesFilter(x, y)
Red	**1**	**RedChannel(x)**
Blue	**1**	**BlueChannel(x)**
Green	**1**	**GreenChannel(x)**
avgRGB	**1**	**(Red + Green + Blue) / 3**

TABLE XI
FUNCTION SET (EXPERIMENT SET 3)

Terminal	Value
Ephemeral Constant	$x = \text{rand}(-5, 5), x \in \mathbb{R}$
Ephemeral Constant	$x = \text{rand}(-10, 10), x \in \mathbb{Z}$
FilterSizeOne	**1**
FilterSizeThree	**3**
FilterSizeFive	**5**
FilterSizeSeven	**7**
FilterSizeNine	**9**
FilterSizeEleven	**11**

TABLE XII
TERMINAL SET (EXPERIMENT SET 2 & 3)

APPENDIX C
OVERALL BEST SOLUTIONS

ADD(DIV(MIN(DIV(-1.473828938318551, SUB(Red(MaxF(FilterSizeSeven, 0, 1)), Red(MaxF(FilterSizeSeven, 0, 1)))), SUB(8.796123982485923, ADD(2.2091736245435385, Green(MaxF(FilterSizeSeven, NEG(NEG(3)), -2))))), -8.75205968278296), MIN(MAX(SUB(8.796123982485923, Red(MaxF(FilterSizeSeven, 0, 1))), SUB(Red(MaxF(FilterSizeSeven, 0, 1)), Red(EdgesF(NEG(5), 2)))), MIN(MAX(MIN(-8.75205968278296, SUB(8.796123982485923, Red(EdgesF(NEG(NEG(0)), NEG(1))))), SUB(Red(MaxF(FilterSizeSeven, 1, 1)), Red(EdgesF(1, 8)))), MAX(NEG(ADD(Red(MaxF(FilterSizeSeven, 2, 1)), Green(MaxF(FilterSizeSeven, 0, 1)))), SUB(Red(MaxF(FilterSizeSeven, 0, 0)), Red(EdgesF(-7, 2)))))))

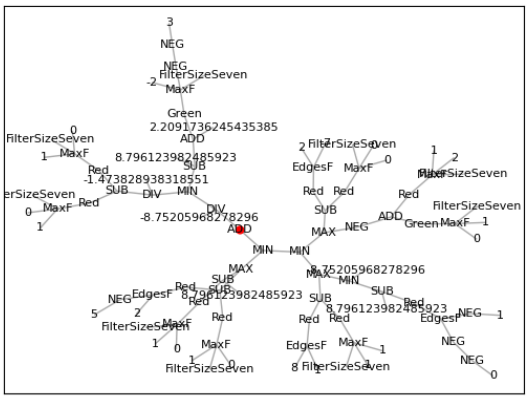

Fig. 14. RGB 750 Population Im 1

SUB(MUL(MUL(Red(MaxF(FilterSizeThree, 0, -1)), -5.538400326022039), DIV(1.4843381604411547, -2.9071439426537493)), Green(EmF(-5, -4)))

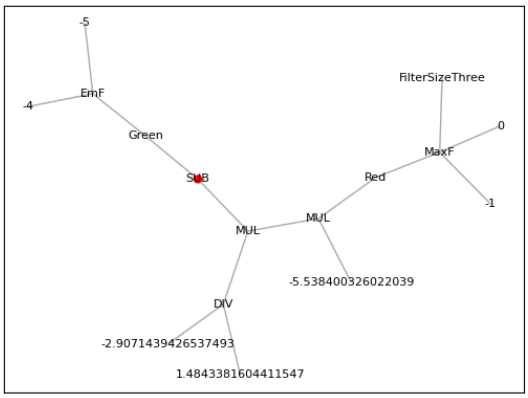

Fig. 15. RGB 750 Population Im 2