Genetic Programming for Algae Detection in River Images

Andrew Lensen[1], Harith Al-Sahaf[1], Mengjie Zhang[1], and Brijesh Verma[2]
[1]School of Engineering and Computer Science
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
[2]School of Engineering and Technology
Central Queensland University, Rockhampton, Australia
Email: {andrew.lensen, harith.al-sahaf, mengjie.zhang}@ecs.vuw.ac.nz, b.verma@cqu.edu.au

***Abstract*—Genetic Programming (GP) has been applied to a wide range of image analysis tasks including many real-world segmentation problems. This paper introduces a new biological application of detecting Phormidium algae in rivers of New Zealand using raw images captured from the air. In this paper, we propose a GP method to the task of algae detection. The proposed method synthesises a set of image operators and adopts a simple thresholding approach to segmenting an image into algae and non-algae regions. Furthermore, the introduced method operates directly on raw pixel values with no human assistance required. The method is tested across seven different images from different rivers. The results show good success on detecting areas of algae much more efficiently than traditional manual techniques. Furthermore, the result achieved by the proposed method is comparable to the hand-crafted ground truth with a *F*-measure fitness value of 0.64 (where 0 is best, 1 is worst) on average on the test set. Issues such as illumination, reflection and waves are discussed.**

I. Introduction

Image Analysis is a broad-ranging field of computer vision which aims to produce useful information from digital images using computational techniques. Tasks include pattern recognition [1], medical imaging [2], and object detection [3], [4]. This work introduces a biological application of analysing the quantity of Phormidium algae present in images of rivers in Wellington and Nelson, New Zealand. This has the potential to provide considerable benefit, as swimming in rivers with high concentrations of algae can be harmful to life, with history of human sickness and death of domestic pets such as dogs [5]. Developing automated means of analysing images has the potential to allow for a much more accurate analysis of the current prevalence of algae in the river than the current manual techniques. This allows more appropriate warnings to people living in the area which can reduce the amount of illness as a result. The use of a Unmanned Aerial Vehicle (UAV) with a camera to collect video footage allows collection of substantially more data than manual techniques, with a reduced amount of time and cost.

Genetic Programming (GP) is an evolutionary algorithm methodology within the field of artificial intelligence (AI) based on the concepts of natural selection [6]. GP evolves a solution in the form of a computer program to attempt to solve a user defined problem. Figure 1 depicts the GP search process. It begins with a set of randomly generated initial solutions to a problem and then iteratively evolves these solutions until a stopping criteria is met, most commonly, a satisfactory solution being found, or a pre-determined number of iterations reached. For each iteration, it evaluates the performance of each solution using a *fitness function* and then applies a selection of *genetic operators* to solutions in the hope of creating solutions with a better fitness.

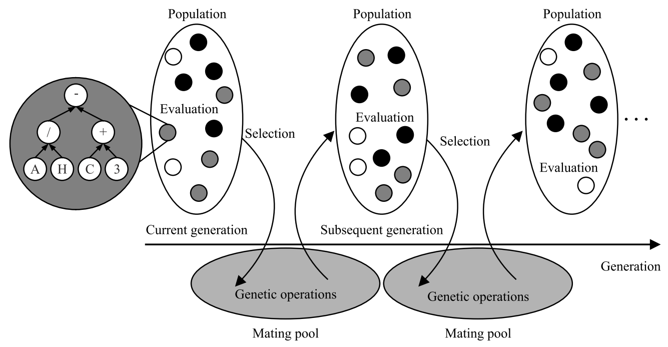


Fig. 1. The GP search process.

Programs in GP are most commonly expressed as syntax trees, which can be represented as Lisp expressions [6], [7]. The syntax tree consists of leaves which are constants or variables from a *terminal set*. The internal, i.e., non-terminal nodes are functions from a *function set*, which take one or more inputs and produce an output. The result of the program is the output of the function which is the root node of the tree. This representation is used in this work, where the root outputs the processed image.

A. Goals

A number of goals are derived from the overall aim of quantitatively analysing the presence of algae. These objectives are as follows:

- Using GP to generate a program that will convert an image into a form that is easier for a human to analyse.
- Ensuring images without algae have minimal amounts of algae falsely detected by the program. False positives give a pessimistic view of algae presence and should be avoided.

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

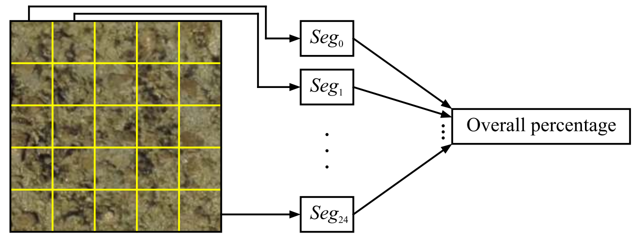


Fig. 2. Example demonstrates the steps to manually estimate the percentage of algae, where Seg_i is the percentage of algae in the i^{th} segment.

- Detecting algae where the application-generated ground truth fails to. The method used to generate ground truths is imperfect as it can be difficult to distinguish algae from certain rock types, so it is hoped our method can achieve even better results.
- Post-processing the result images to find the percentage of algae in an image, and an average across the whole dataset. These values should be similar to the values provided by the human expert.

B. Organisation

The remainder of the paper is organised as follows. Section II provides a brief description of the problem this work addresses. Section III introduces the proposed GP system and highlights the evaluation procedure. Section IV details the settings and parameters of the experiments performed. Section V then provides the results of these experiments and discusses how well they fulfil the goals. Section VI then concludes this paper with an evaluation of the overall success achieved and the potential for future work.

II. Background

The existing work on analysis of algae using GP techniques is very limited and still in its early stages. Thus, this section instead details the existing methods of algae analysis and also related GP work that has similarities to this problem.

Current analysis of Phormidium on the Hutt River uses physical techniques which involve the analysis of algae across $0.50 \times 0.50\ \mathrm{m}^2$ (square meter) quadrats of the river [5]. These quadrats are then examined in sub-segments, with a percentage of algae being estimated per segment as presented in Figure 2. The percentage of algae across the quadrat is then the average of these values.

To improve on this, an application was developed to help allow manual analysis of images so that a UAV could be used to collect images for analysis at a later time. This application is used as follows. For each input image, three colour channels are generated: *red*, *green*, and *blue*, each of which is converted into a greyscale format. Then a colour channel is manually chosen, and a simple thresholding approach is adopted to segment the image by manually selecting a threshold value that is between 0 and 255. All pixels with a hex value lower than this threshold are labelled as algae (Phormidium is a dark algae) and all higher are non-algae. On some images, additional steps such as auto-level (normalises the image) and histogram equalisation are applied to give a better result. While this application proved useful, it still requires human intervention to operate and is not fully automated so analysing many images still takes considerable time.

Image segmentation is particularly relevant to this work as the problem can be seen as partitioning an image into multiple homogeneous segments of algae and non-algae. Segmentation often attempts to simplify an image to be easier to analyse, which is what our first goal hopes to achieve. Image segmentation is used in many areas such as medical imaging and object detection. This work introduces a new application of biological analysis.

Over the past few decades, the task of image segmentation has received increasing interest and a large number of methods have been proposed. The methods of image segmentation can be categorised into at least five categories [8], [9]. *Threshold based segmentation* probably is the simplest of all other methods, which aims at performing the segmentation task by dividing the image into a number of regions based on a set of predetermined threshold values. *Edge based segmentation* assumes that object boundaries are represented by edges that can be used to identify the corresponding object. *Region based segmentation*, as opposed to edge based segmentation, has the aim of identifying objects in images by locating some points of different regions and growing them until the edges (boundaries) of the corresponding object are found. *Clustering techniques* aim at grouping similar regions of the image together that are similar based on content. *Matching* attempts to locate an object based on approximate prior knowledge. Discussing the methods of these approaches is beyond the scope of this paper.

Combining different segmentation methods to form a hybrid method has been shown to be an effective methodology for image segmentation, and better performance can be achieved compared with the use of each method in isolation [9]. Evolutionary Computation (EC) techniques such as Genetic Programming (GP), Genetic Algorithms (GAs), and Particle Swarm Optimisation (PSO) have emerged to perform a variety of tasks to combine segmentation methods. In [10], GAs were utilised to find a set of thresholding values; thus, the segmentation task was framed as an optimisation task to find these thresholds. Duraisamy and Kayalvizhi [11] proposed a PSO framework for selecting multi threshold values for image segmentation. In terms of combining EC and region based techniques, PSO was combined with the *seeded region growing* method in [12] to mitigate the problem of selecting the number of seeds. Furthermore, Al-Faris et al. [12] tested the effectiveness of this combination using a *Magnetic Resonance Imaging* (MRI) dataset, and the results showed that a better performance was achieved compared to a number of other techniques. Meanwhile, the use of EC techniques along with edge detection based methods for image segmentation was investigated in [13]. In [13], GAs were applied to calculate an edge map from a previously calculated depth and orientation gradients; however, their experiments showed that the system was not robust in handling the presence of noise. EC techniques and clustering methods have also been combined to improve the performance of those methods for image segmentation [14]–[17], and in most cases, introducing EC techniques increased the performances of these method substantially.

Genetic Programming has been applied to many different areas of image analysis, including texture classification [18], edge detection [19] and image segmentation [20].

There has been additional recent work done on image segmentation using GAs such as work done by Amelio and Pizzuti [21], which investigated using a graph-based structure for segmentation. The undirected graph consists of pixel nodes with edges connecting similar pixels. GAs are used with this representation using genes which represent an edge in the graph. Pixels that share an edge are then part of the same segmentation region. The algorithm is compared to the existing method that it is an extension of over a range of datasets with promising results. Work by Ribeiro et al. [22] investigated the use of GAs to determine the coverage of soil by crop residues. This is an example of another biological application of GAs and has similarities with this work as it also utilised each of the red, green, and blue channels of images and produced a binary image as a result. The residue is distinctive compared to the surrounding environment and so their work produced results very similar to that of an expert. Phormidium, on the other hand, is often similar in colour to the surrounding environment which makes the problem presented in this paper particularly difficult.

Genetic Programming is widely used for binary classification [23] where the problem requires classifying an instance as either belonging to the class of interest (foreground) or not (background). This is generally achieved with the root of the GP tree outputting an integer determines the class of the instance based on a threshold. For example, if the resulting value from the root node is negative then the instance is assigned a foreground label; otherwise, it considered as belonging to the background class. This idea is extended in this work by considering each individual pixel as a single binary classification and using a threshold of zero [24].

III. The Proposed Method

This section describes the design of the GP method (terminal and function sets, fitness function) as well as the pre-processing steps that prepare an image for use in the method and the post-processing steps which convert the result of the method into a useful form.

A. Pre-processing

As the method operates directly on raw pixel values, no feature extraction is performed. Instead, images are pre-processed so that the method can operate most effectively. Each image is converted to greyscale using the batch convert feature of *IrfanView 4.38* [25], to give a greyscale image called img that is a weighted average of the red, green, and blue channels. Each image is also split into three additional greyscale images, one of each of the red, green, and blue channels (named c_r, c_g, and c_b) of the image using *ImageMagick 6.90-Q16* [26]. This process produces four 8-bit greyscale images which are used as part of the terminal set.

B. Terminal Set

The terminal set comprises of the four images img, c_r, c_g, c_b as well as a random constant value from the set $\{-3, -2, -1, 0, 1, 2, 3\}$. Apart from the random constant, each terminal node is a 2D matrix with size $M \times N$ where M and N are, respectively, the width and height of the image.

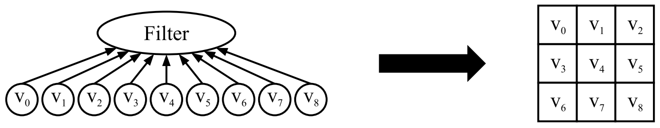


Fig. 3. Converting a *filter* node to a 3×3 convolution mask.

C. Function Set

The function set consists of eleven operators, of which eight are categorised into one group as they have the same input arguments. This group consists of the operators add, sub, div, mul, min, max, abs_{add}, and abs_{sub}, which take two images and return an image. All of these operate on a pixel-by-pixel basis, i.e., they perform the operation on each pair of pixels with the same co-ordinates in each image. The first four return the result of the normal mathematical functions $(+, -, \div, \times)$ where $\div$ is protected: if the second argument (divisor) is zero, it returns zero. min and max return the smallest and largest of the two input values, respectively. abs_{add} and abs_{sub} perform addition and subtraction respectively, and then take the absolute value of the result before returning it. The ninth operator is the *if* operator, which takes three images a, b, and c as inputs and returns a single image r as a result. This operator also works pixel-by-pixel: for each set of three pixels with co-ordinates i, if a_i is negative, set r_i to be b_i; otherwise, set r_i to be c_i. The *if* operator allows the evolved program to operate differently based on the value of a pixel, which is useful in image segmentation problems.

The final two operators work to apply a 3×3 filter to the image. The *filter* operator takes 9 constant values as inputs, and returns a 3×3 filter as an output, where each input corresponds to one entry in the 3×3 matrix as presented in Figure 3. The $conv$ operator takes a filter and an image as inputs, and then applies convolution to the image input using the filter input. It returns the result of this convolution.

As the majority of the operators take at least one image as an input and return an image as an output (except for *filter* which works in combination with $conv$), the method can evolve long chains of operators easily. There is no normalisation done by any of the function operators, so pixel values are able to fall outside of the normal 0 and 255 range at the root of the GP tree. Post-processing is used to convert the output of the root of the tree into a valid image after the GP process has completed. This allows the image produced by the GP program to be viewed easily by humans.

D. Post-processing

A simple threshold is applied to each pixel of the output image from the program, where a pixel value lower than zero is converted to a white pixel (non-algae pixel) and a non-negative pixel value is converted to a black pixel (an algae pixel). The final image is then a binary (black-and-white) representation of the original image with identical dimensions, where all black areas have been interpreted as being algae and all white areas have not. In addition, a percentage of algae in an image is calculated by summing over all black pixels as shown in Equation (1). This allows an evaluation of the algae growth in a quantitative manner, and also allows the performance of the program to be compared to that of an expert.

$$percentage\,(I) = \left(\frac{I_{\text{algae}}}{I_{\text{total}}}\right) \times 100 \tag{1}$$

Here, I is the image being evaluated, I_{algae} is the number of pixels in I that have been marked as algae, and I_{total} is the total number of pixels in the image.

E. Fitness Function

The fitness function used is the standardised F-measure function which is shown in Equation (2), where a fitness of 0 is a perfect result, and a fitness of 1 is the worst fitness. This function is used due to the imbalance between algae and non-algae: non-algae usually dominates an image and simpler measures (e.g. simple % of correct pixels) can struggle to produce useful results in these cases.

$$Fitness = 1 - \left(\frac{2 \times TP}{2 \times TP + FP + FN}\right) \tag{2}$$

Here, TP (true-positives) is the number of black pixels (algae) in the image which are black pixels in the ground truth (GT), FP (false-positives) the black pixels in the image which are white in the GT, and FN (false-negatives) the white pixels (non-algae) in the image which are black in the GT. Each of these variables are calculated from summing across all pixels in the image where pixels with a negative value are algae and others are non-algae, as in the post-processing stage. Hence, the resulting fitness is a measure of how accurately algae has been correctly identified, taking into account non-algae being incorrectly labelled as algae.

An individual program's fitness can be found as follows. The pre-processing steps are performed on a given image to generate the four greyscale images used in the terminal nodes. The GP tree is then evaluated in bottom-up order, until the root node is finally evaluated. The result of the root node is post-processed to a binary representation and then the number of TP, FP and FN are found by iterating over all pixels in the image and comparing their classification (algae or non-algae) to the ground truth. The fitness is then found using Equation 2. When training, the fitness of an individual is found by finding the average fitness across each image in the training set after a given generation. Once training is complete, the performance of the best evolved individual is found by averaging the fitness across each image in the test set (unseen images) using Equation (1).

IV. Experiment Design

This section discusses the training and test images used for the experiments, including how they were obtained and how ground truths were created. It also provides the parameter settings used.

A. Dataset

The images used in this work were taken from video footage provided by a UAV which was flown over the Hutt River with a video camera attached. This video was taken as part of previous work exploring the potential for a UAV to reduce the need for physical data collection. Video was captured from two regions: the Hutt River in Wellington and a range of rivers near Nelson. Images in each region were taken from individual frames of video where there was the minimum amount of surface reflection, waves and other noise present. This provided seven distinct images. These can be seen in the first column of Figures 4-10. The second and third columns are, respectively, the manually generated ground truth and the result of the best individual. Images are of varying dimensions with the smallest being 197×204 pixels and the largest 773×775 pixels and also of varying light-levels (illumination). This ensures a more robust program is generated which is of benefit as capturing images of a particular resolution and illumination is difficult with a UAV due to the physical factors at play.

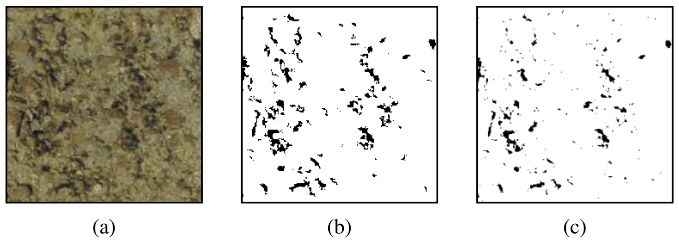


Fig. 4. Image A (a) original, (b) manually generated ground truth, and (c) best individual result.

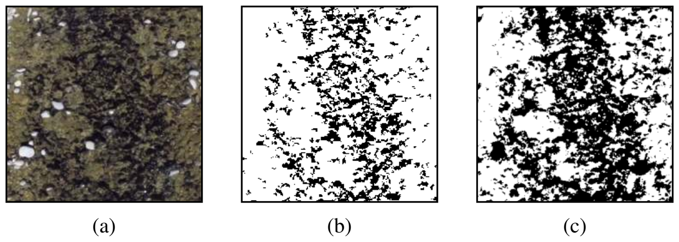


Fig. 5. Image B (a) original, (b) manually generated ground truth, and (c) best individual result.

B. Generating Ground Truths

A ground truth in image analysis is the output image which gives a perfect fitness for an input image. In other words, it is the result the GP program would give if it were perfect. Normally ground truths are created by domain experts, but in this case the expert has provided only the percentage of algae present in images. As a substitute, ground truths have been generated manually using the application discussed previously (II). The ground truth for an image, as seen in the second column of Figures 4–10, is the output of this application for that image where black pixels are algae and white are non-algae. This process takes substantial time for multiple images and requires a human to try to manually distinguish algae from dark rocks which introduces error. As such, the proposed method aims to produce better results than the application-generated ground truth and to be much more time-efficient.

C. Parameter Settings and Implementation

All the parameter settings of the GP method are provided in Table I. These parameters are based on previous work in the literature. The implementation uses *Strongly-typed GP* (STGP) [27] as the different operators vary in the number and type of their inputs and outputs. The *Java Evolutionary Computation Toolkit* (ECJ) [28] package was used to implement the GP

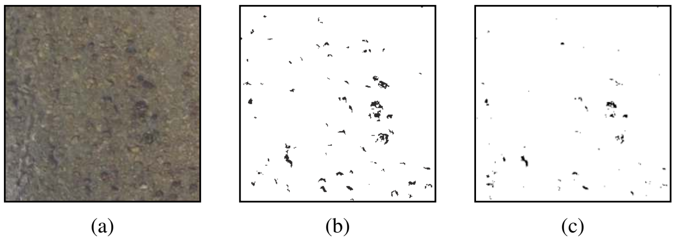


Fig. 6. Image C (a) original, (b) manually generated ground truth, and (c) best individual result.

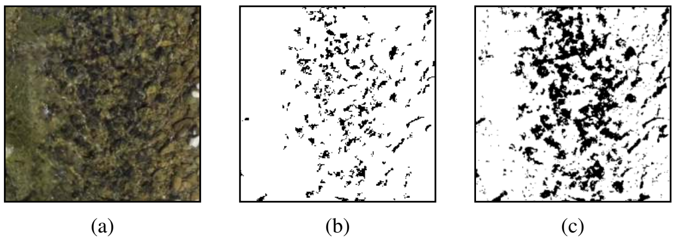


Fig. 7. Image D (a) original, (b) manually generated ground truth, and (c) best individual result.

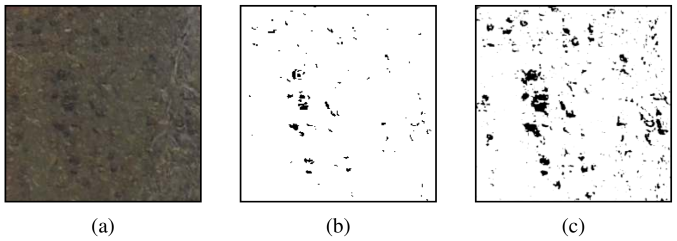


Fig. 8. Image E (a) original, (b) manually generated ground truth, and (c) best individual result.

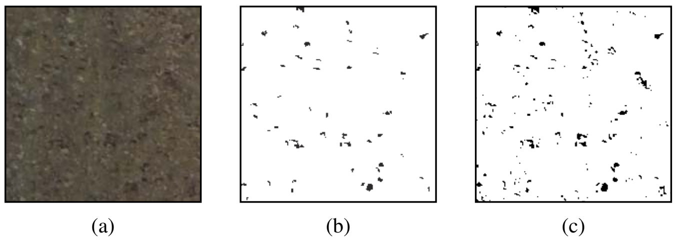


Fig. 9. Image F (a) original, (b) manually generated ground truth, and (c) best individual result.

method. GP is a non-deterministic method as it produces different results based on the initial *seed* provided to the random number generator. As a result, the experiment was run 35 times independently with different seeds provided to each run. The average performance and standard deviation on the test set across the 35 runs is reported.

The process used in each experiment run is as follows. Take each image in turn and assign it to be the test set; the remaining images are the training set , i.e., *k-fold cross-validation* [29], where k is 7. This gives seven iterations where each has one test image and six training images. For each of these iterations, run the GP method as previously mentioned. The performance of this experiment run is then the average of the performance on the test image as per Equation (2) across these seven iterations.

As this work seeks to apply GP to a real-world application-specific problem, there are no suitable benchmark solutions that the results could be fairly compared to. Instead, the performance is discussed based on the visual similarity between the produced images and the ground truths as measured by the F-measure, and the difference between the percentage of algae present in the produced images and the percentage of algae recorded by the domain expert.

V. Results and Discussions

This section presents the results of the experiment runs, and discusses the effectiveness of these in relation to the goals of this work. A F-measure fitness value of 0.640 (where 0 is best, 1 is worst) was achieved with a standard deviation of 0.066 on average on the test set. The best individual ($\#19$) had a F-measure fitness value of 0.481. From this individual, the fifth iteration (referred to as individual $19:5$) of the k-fold cross-validation was used to produce result images for all of the seven images in this section for discussion. These results are seen in the third column of Figures 4-10, and the GP tree representing this program along with the Lisp expression is shown in Figure 11(a) and (b) respectively. The subsections that follow discuss the performance of the results in terms of each of the goals of the work.

A. Qualitative Results and Analysis

Applying the best individual from the experiments to the test images produces result images (i.e. third column of Figures 4-10) that are much easier to analyse than the original images. The result images clearly show where algae is located, with a black-and-white transformation, it is very easy to see where algae is at a glance and to estimate the prevalence of it.

Specific images vary in their accuracy relative to the ground truth (GT). Results shown in Figure 4 and Figure 9 perform very well as they identify all the large regions of algae in the GT. Some areas of algae are not captured exactly as in the GT but the result is still very close considering the difficulty of the problem. Results depicted in Figure 5 and Figure 7 also perform well as they again capture nearly all areas of algae identified by the GT. Unlike the previous results, they label significantly more areas as algae compared to the GT with many areas being much denser in algae. The next subsection discusses if these results are actually more accurate than the GT in comparison to the original image. They also have some trouble with the white stones in their images, e.g., the group of stones on the left of Figure 5, are incorrectly identified as algae which is clearly wrong. Results in Figure 6 and Figure 8 are more inaccurate than the other images as they differ more noticeably from the GTs. Image 6 has some reflection in the bottom-left of the image and Figure 8 has waves and reflection in the right of the image which may have caused poorer results, as the previous images did not have much noise. The last result, presented in Figure 10, performed very poorly as it missed most regions of algae in the GT and over-emphasised the areas it did correctly identify. Figure 12(a) is a larger version of this image, i.e., Figure 10, which clearly shows a dark portion of shadow at the bottom of the image. As the Phormidium algae is very dark relative to the surrounding environment, the generated program identifies algae based on regions of

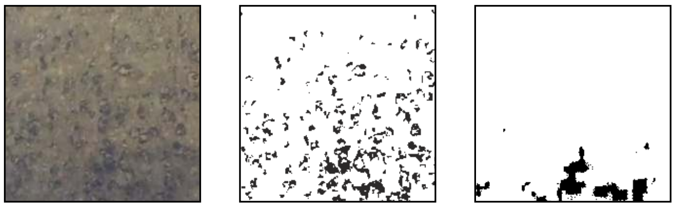


Fig. 10. Image G (a) original, (b) manually generated ground truth, and (c) best individual result.

TABLE I. GP PARAMETERS FOR EXPERIMENTS

Parameter	Value
Crossover Rate	80%
Mutation Rate	19%
Elitism Rate	1%
Tree depth	2-6
Generations	50
Population Size	200
Selection Type	Tournament
Tournament size	7
Initial Population	Ramped half-and-half

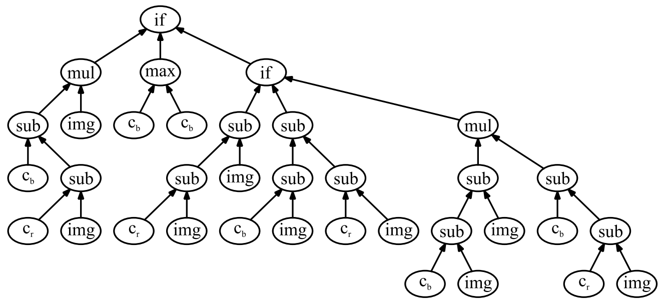
darkness. The shadows in this image are of similar darkness to algae which causes the program to perform poorly. Future work will seek to address this problem.

B. Improving on the Ground Truths

As previously mentioned, the results presented in Figure 5 and Figure 7 appear to actually better represent the algae in the original images than the GTs do. Both of these GTs leave lots of small "holes" in the main large patches of algae in the centre of the images whereas the result images give more continuous and smoother areas of algae. This highlights the difficulty involved with manual processing, as setting a higher threshold to capture these areas better causes other areas to be incorrectly marked as algae (by introducing more noise). This problem is further increased when there are rocks with shadows in the image (on the left and right of these two images) as a high threshold can very easily incorrectly capture shadows. Results depicted in Figure 5 and Figure 7 do capture shadows to a higher extent than the GTs, especially on the right of the images, but this problem is outweighed by the good results on capturing dense regions of algae more accurately. This improvement is also seen to a less extent in result Figure 8 (see enlarged version in Figure 12(b)). The GT for this image missed the areas of algae on the far left of the image whereas the result accurately captured them. The result also captured areas that appear to look like algae to the right of the image where areas of waves are; the GT only had small areas of these. These improvements again come at the cost of some false positives, but it is clear that this goal has been achieved with enough success that we can conclude this GP method can perform better than the current application as well as being much less time-consuming to run.

C. Comparisons to Expert Analysis

The images from the Hutt River (Figures 6, 8-10) were analysed by an expert in algae who then provided a percentage of that image which contained algae. These values have been compared to those generated in the post-processing step and



(a)

```
(if (mul (sub cb (sub cr img)) img) (max cb cb) (if (sub
(sub cr img) img) (sub (sub cb img) (sub cr img)) (mul (sub
(sub cb img) img)) (sub cb (sub cr img)))
```

(b)

Fig. 11. The (a) tree representation and (b) Lisp expression of program #19 (Individual 19:5) evolved by the proposed method.

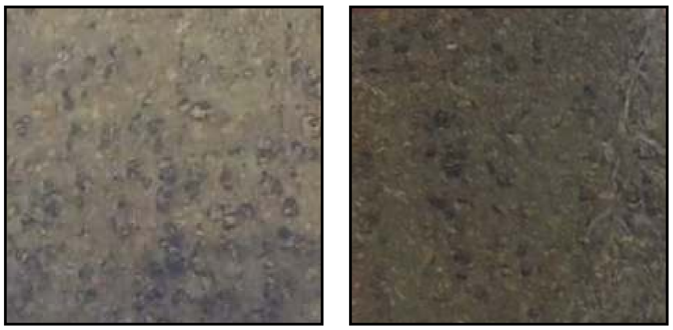


Fig. 12. Figure depicts (a) image with extensive shadowing at the lower part, and (b) enlarged image from Figure 8.

are shown in Table II. In addition, we provide the percentage of algae in the ground truths for these images for comparison. These are calculated in the same manner as for the result images.

The results in the table show mixed success. The results in row 6 and 8 perform worse than the ground truths in terms of the absolute difference compared to the percentage provided by the expert. Results 9 and 10 perform better than the GTs with 10 in particular being much nearer. As result 10 appears to actually be finding the percentage of shadow in the image (as discussed previously), it cannot be concluded that any of the result images perform significantly better than the ground truths. The other three results are relatively close to the expert analysis: all are within a few percentage points and so are accurate enough to provide an analysis of the algae present in broader terms. For example, it can be claimed that all images contain less than 10% algae with reasonable confidence. As the use of this work will be to improve warnings on the presence of algae in rivers, this sort of accuracy is accurate enough. Regardless, future work will seek to enhance the accuracy of these results so that more substantial claims can be made.

D. Further Analysis

In order to have better understanding of the program evolved by the proposed method, the program presented in

TABLE II. PROPORTION OF ALGAE IN RESULTS COMPARED TO EXPERTS EVALUATION

Figure	Ground truth	GP Result	Expert
6	1.77%	0.60%	3.52%
8	1.36%	6.05%	3.52%
9	1.36%	2.45%	2.00%
10	10.54%	4.29%	6.92%

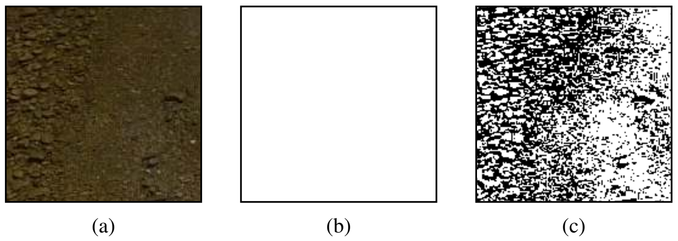


Fig. 14. First example of a control image (a) original, (b) actual ground truth, and (c) best individual result.

Figure 11 is analysed in this subsection. Each of the terminal nodes has been replaced by the corresponding image, whereas non-terminal nodes have been replaced by the resulting image after applying the corresponding operator on the inputs as depicted in Figure 13. Those images are normalised to have values between 0 and 255 for visualisation purposes. Equation (3) is used to normalise the pixel values. The final result of the program is the binary image resulting from thresholding the image at the root node.

$$x' = \left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right) \times 255 \tag{3}$$

Here, x' is the normalised value, and x is the current pixel value. The minimum and maximum pixel values of the image are, respectively, denoted as $x_{\min}$ and $x_{\max}$.

E. Control Images

The program in Figure 11 was applied to two control images (images where there are no algae present) to analyse how well the program will prevent false positives when there are no true positives. The two control images and their results are, respectively, presented in Figure 14 and Figure 15. The ground truths for these images were white images with the same dimensions as as the control images as shown in the second column of these two figures.

The result images generate a large number of false positives, particularly where there are shadows present on rocks in the river. There were no control images included as part of the (training) data set as the F-measure fitness function was used (as it produced good results on the other images) and this fitness function cannot operate correctly on images with no algae in them. As seen in Equation 2, when TP is always zero, the value of the F-measure will always be 1 regardless of false positives and false negatives. Later work will utilise other fitness functions such as *Area Under the Curve* (AUC) [30] that can support control images in training. It is hoped this will improve the performance of the result program on unseen images with no algae present.

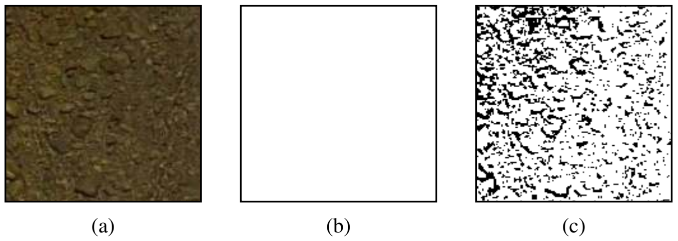


Fig. 15. Second example of a control image (a) original, (b) actual ground truth, and (c) best individual result.

VI. CONCLUSIONS

This work applied GP to a difficult domain-specific problem where manual human analysis was costly. Terminal and function sets specific to the problem and to image analysis were applied with a fitness measure and parameter settings that have given good performance in earlier work. Goals were generally well met considering the complications associated with real image data. The comparisons to expert analysis and performance on control images were not as successful as the other goals. A number of suggestions as to how to improve these results were made, and will be investigated in future work. The method struggled on the image with a large amount of shadow and also had some difficulty on images with reflections, but performed particularly well on images without these issues and even appeared to perform better than the generated ground truths in some cases.

In addition to the previously mentioned ideas, we would like to investigate the impact of removing some elements from the function and terminal sets to reduce the search space and introducing channel derivatives or standardisation of the images (to equalise illumination) on the performance, which may further improve the results on the easier images. It is also hoped these techniques can be applied to other types of algae of different colours (Green algae, Diatom) that are a problem in other river and marine environments.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[2] F. A. Faria, R. T. Calumby, and R. da Silva Torres, "RECOD at ImageCLEF 2011: Medical modality classification using genetic programming," in *Proceedings of the Conference and Labs of the Evaluation Forum*, V. Petras, P. Forner, and P. D. Clough, Eds., 2011, pp. 1–9.

[3] I. Ulusoy and C. M. Bishop, "Comparison of generative and discriminative techniques for object detection and classification," in *Toward Category-Level Object Recognition*, ser. Lecture Notes in Computer Science, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds., vol. 4170. Springer, 2006, pp. 173–195.

[4] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan, "Contextualizing object detection and classification," in *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1585–1592.

[5] M. Heath, K. Brasell, R. Young, S. Wood, and K. Ryan, "Hutt river Phormidium habitat suitability criteria and hydraulic habitat assessment," School of Biological Sciences, Victoria University of Wellington, Tech. Rep. MH01, 2012.

[6] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*. Lulu, 2008, (With contributions by J. R. Koza).

[7] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

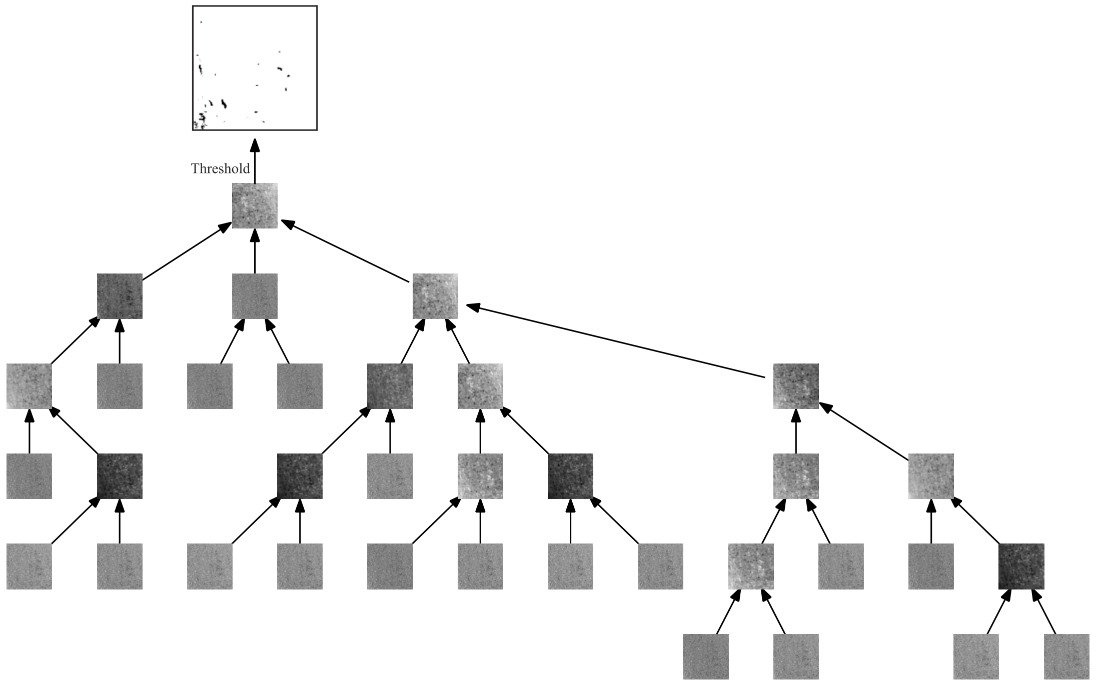


Fig. 13. The representation of the program presented in Figure 11 using the image shown in 6 as input.

[8] N. Senthilkumaran and R. Rajesh, "Edge detection techniques for image segmentation a survey of soft computing approaches," *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, pp. 250–254, 2009.

[9] W. Khan, "Image segmentation techniques: A survey," *Journal of Image and Graphics*, vol. 1, no. 4, pp. 166–170, 2013.

[10] O. Banimelhem and A. Y. Yahya, "Multi-thresholding image segmentation using genetic algorithm," in *Computer Engineering, and Applied Computing*. World Congress in Computer Science, 2011, pp. 1–6.

[11] S. Duraisamy and R. Kayalvizhi, "A new multilevel thresholding method using swarm intelligence algorithm for image segmentation," *Journal of Intelligent Learning Systems and Applications*, vol. 2, no. 3, pp. 126–138, 2010.

[12] A. Al-Faris, U. K. Ngah, N. A. M. Isa, and I. Lutfi, "Breast mri tumour segmentation using modified automatic seeded region growing based on particle swarm optimization image clustering," in *Proceedings of the 17th Online World Conference on Soft Computing in Industrial Applications*. Springer, 2012, pp. 1–11.

[13] I. Diaz, J. Branch, and P. Boulanger, "A genetic algorithm to segment range image by edge detection," in *Proceedings of the International Conference on Industrial Electronics and Control Applications*. IEEE, 2005, pp. 7–14.

[14] S. Shirakawa and T. Nagao, "Evolutionary image segmentation based on multiobjective clustering," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 2466–2473.

[15] I. Saha, U. Maulik, and S. Bandyopadhyay, "An improved multi-objective technique for fuzzy clustering with application to irs image segmentation," in *Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science, vol. 5484. Springer, 2009, pp. 426–431.

[16] U. Maulik and S. Bandyopadhyay, "Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 5, pp. 1075–1081, 2003.

[17] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1506–1511, 2007.

[18] A. Song, T. Loveard, and V. Ciesielski, "Towards genetic programming for texture classification," in *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence*, ser. Lecture Notes in Computer Science, vol. 2256. Springer, 2001, pp. 461–472.

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

[20] Y. Liang, M. Zhang, and W. Browne, "Image segmentation: A survey of methods based on evolutionary computation," in *Proceedings of the 10th International Conference on Simulated Evolution and Learning*, ser. Lecture Notes in Computer Science, vol. 8886. Springer-Verlag, 2014, pp. 847–859.

[21] A. Amelio and C. Pizzuti, "An evolutionary approach for image segmentation," *Evolutionary Computation*, vol. 22, no. 4, pp. 525–557, 2014.

[22] A. Ribeiro, J. Ranz, X. P. Burgos-Artizzu, G. Pajares, M. J. Sanchez del Arco, and L. Navarrete, "An image segmentation based on a genetic algorithm for determining soil coverage by crop residues," *Sensors*, vol. 11, no. 6, pp. 6480–6492, 2011.

[23] T. Loveard and V. Ciesielski, "Representing classification problems in genetic programming," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2. IEEE Press, 2001, pp. 1070–1077.

[24] R. Poli, "Genetic programming for feature detection and image segmentation," in *Evolutionary Computing*, ser. Lecture Notes in Computer Science. Springer, 1996, vol. 1143, pp. 110–125.

[25] I. Skiljan, "IrfanView," 2015, [Online; accessed January 2015]. [Online]. Available: http://www.irfanview.net/

[26] S. Salehi, *ImageMagick Tricks: Web Image Effects from the Command Line and PHP: Unleash the Power of ImageMagick with this Fast, Friendly Tutorial and Tips Guide*, ser. From technologies to solutions. Packt Publishing, 2006.

[27] D. J. Montana, "Strongly typed genetic programming," *Evolutionary Computation*, vol. 3, no. 2, pp. 199–230, 1995.

[28] S. Luke, *Essentials of Metaheuristics*, 2nd ed. Lulu, 2013. [Online]. Available: http://cs.gmu.edu/∼sean/book/metaheuristics/

[29] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 111–147, 1974.

[30] J. Doucette and M. Heywood, "GP classification under imbalanced data sets: Active sub-sampling and AUC approximation," in *Proceedings of the 11th European Conference on Genetic Programming*, ser. Lecture Notes in Computer Science, vol. 4971. Springer, 2008, pp. 266–277.