A new GP apporach for image segmentation

Yuan Gao

January 2022

1 Introduction

Image segmentation is one of the important and useful techniques in image processing. It can result robust and high degree of accuracy.[1] In methodology, Image segmentation is the art of partitioning an image into multiple smaller segments or groups of pixels, such that each pixel in the digital image has a specific label assigned to it. Pixels with the same label have similarity in characteristics. Image segmentation can detect the edges, boundaries and outlines within an image. [2]

Figure-ground image segmentation is a special case of image segmentation, which only separates foreground objects or regions of interest from the background. It is a crucial preprocessing step in the fields of computer vision and image processing, as its results can be input to the higher- level tasks, e.g. object tracking and image editing [3].

Genetic programming (GP) is a main branch of evolutionary computation techniques, which are inspired by biological evolution [4]. GP can automatically create a computer program/algorithm to solve the problem based on a high-level statement of the task. Considering the unsatisfactory performance of existing segmentation methods, GP has been introduced to evolve segmentors by existing works [5, 6, 7, 8, 9], which show that the segmentors produced by GP can produce promising results in certain image domains, e.g. texture images [5, 6], biomedical images [8] and object images [9].

1.1 Goals

The overall goal of this paper is to develop a multi-layer GP (MLGP) approach to achieving simultaneous region detection, region construction, and image segmentation. The approach can integrate a set of image-related operators/functions in GP to detect informative high-level image region. To achieve this, a multi-layer program structure, a function set and a terminal set are developed. This method will be examined and compared with six other common image segmentation methods on three image segmentation datasets of varying difficulty. Specifically, this paper will investigate the following objectives.

1. How the MLGP approach can achieve automatic region detection,region construction, and segmentation using a single solution/tree by developing a program structure, a function set and a terminal set;

2. Whether the MLGP approach can achieve better segmentation performance than the other six common segmentation methods.

2 Multi-layer GP

The multi-layer GP (MLGP) approach has a new program structure of multi-layer, a new function set, including a number of image-related operators, and a new terminal set. This section will introduce these components of MLGP.

2.1 Multi-layer Program Structure

To achieve automatic region detection, region construction, and segmentation simultaneously, a multi-layer program structure is designed in the MLGP approach. The program structure has four layers, i.e., an input layer, a region detection layer, a region construction layer, and a segmentation layer. The program structure is shown in Fig.1. An example program is also shown in Fig. 1 to show how the multiple layers are constructed in a single program tree. In this figure, each layer is shown in a different colour.

Input Layer: The input layer takes the image being segmented and constant parameters, known as ephemeral random constants, as inputs. The input layer often indicates the terminals employed in the MLGP approach.

Region Detection Layer: The region detection layer aims to detect object from the large input image and transfer input image to ground image.

Region Construction Layer: The region construction layer aims to construct ground image using the features extracted from the feature extraction layer. The region construction layer can use the commonly used arithmetic functions, i.e., $+$, , $\times$ and protected $\div$, which are the commonly used functions in the classic GP method. With these functions, the region can be constructed into a high-level region.

The program structure of the proposed MLGP method is constructed according to the six layers in a bottom-up manner, as shown in Fig.1. It is a tree-based representation, where operators are used to form the internal nodes and terminals are used to form the leaf nodes. To deal with different tasks at each layer, a set of operators and terminals are employed. As the example program tree shown in Fig. 1, it has operators i.e. Thresholdling, Cannyedge, chanVese, Snakes, walk and terminals i.e. Image, X, Y, Z.

More details about the operators and terminals can be seen in Sect.2.3.

With the multi-layer program structure, MLGP can integrate different types of functions into a single tree and construct solutions for simultaneous region detection, region construction, and segmentation.

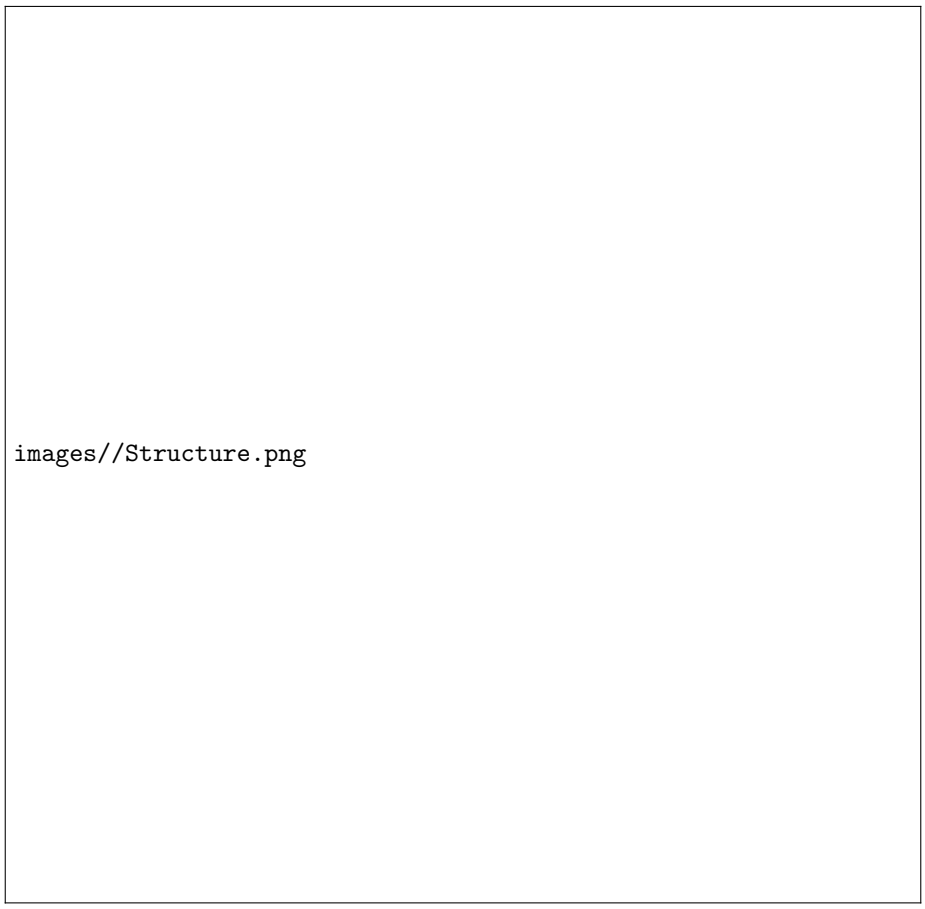

Figure 1: Multi-layer program structure and an example program that can be evolved by MLGP

Terminal	Output type	Description
Image	Img	The input image. It is an array with values in the range of [0, 255]
X	Int 1	The parameter of Thresholding, it is in the range of range [0, 1]
Y	Int 2	The parameter of walk, it is in the range of range [0, 1]
Z	Int 3	The parameter of walk, it is in the range of range [0, 1]

Table 1: Terminal set of MLGP

Function	Input type	Output type	Description
Region detection Functions			
Thresholding	Img	region	Detect a region by from a input image
Canny_edge	Img	region	Detect a region by from a input image
chanVese	Img	region	Detect a region by from a input image
waterShed	Img	region	Detect a region by from a input image
Snakes	Img	region	Detect a region by from a input image
walk	Img	region	Detect a region by from a input image
Region construction functions			
ADD	region, region	region	addition of two regions, if an entry is bigger than 1, return 1, otherwise 1
ADD2	region, region	region	addition of two regions, if an entry is bigger than 1, return 0,otherwise 1
weighted_sum	region, region, Double	region	region1 * float + region2 * (1-float), then, if an entry is smaller than 0.5, return 0, otherwise 1

Table 2: Function set of MLGP

2.2 Terminal Set

The terminal set has four terminals, indicating the inputs of the MLGP approach. The terminals are Image, X, Y, Z. The data types and detailed information of these terminals are described in Table1. The image terminal represents the input RGB-scale image, which is a 3-D array with image pixel values in range [0, 255]. The X, Y and Z terminals are the parameters within a range from 0 to 1. In the MLGP approach, the values of X, Y and Z are randomly selected from their ranges at the initialisation step and changed by mutation during the evolutionary process.

2.3 Function Set

The function set of MLGP has region detection functions, and region construction functions for each layer. The input type, output type and descriptions of these functions are listed in Table2. This section describes these functions.

Region Detection Functions: The region detection layer has six functions. Each function is a common method to do image segmentation and is also

the baseline function. Each function takes image as input and returns a figure-ground image. In the function Thresholding, the X is terminal. In the function walk, the Y and Z are terminal. The values of these terminals are randomly generated from their pre-defined ranges.

Region Construction Functions : The region construction layer has three functions, i.e., ADD, ADD2 and weightedsum.

2.4 Fitness Function

In the MLGP approach, the fitness function(F) is the f1 score, which is good and commonly used for figure-ground image segmentation. The fitness function is defined as follows.

$$F = \frac{TP}{TP + \frac{1}{2}(FP + FN)} * 100\% \tag{1}$$

where TP, FP, and FN indicate the total numbers of true positives, false positives, and false negatives, respectively. TP indicates the positive samples are correctly classified into the positive class. FP indicates the positive samples are classified into the negative class and FN indicates the negative samples are classified into the positive class. The fitness function is to be maximised and the value range is in [0, 100].

2.5 Overview of MLGP for Image Segmentation

3 Experiment Design

This section designs the experiments, including benchmark datasets, baseline methods and parameter settings.

3.1 Benchmark Datasets

To evaluate the performance of the proposed MLGP approach, three different datasets are used as benchmark datasets to conduct the experiments. These datasets are **Sprat**, **Hourse**, **Gilt** [10]. These datasets represent 3 typical image segmentation tasks. Segment the specific fish from background. The difficulties of image classification varies with the datasets due to different variations such as scale, illumination, rotation in images. The images are RGB-scale images. In the experiments, each dataset is spilt into the training set, the validation set and the test set, having 50%, 25%, 25% images respectively. The original

Name	Size	Training set	Validation set	Test set
Sprat	100*100	1203	403	403
Gilt	100*100	1203	403	403
Hourse	100*100	1203	403	403

Table 3: Dataset properties

images datasets are resized to 100 × 100 with high quality in order to maintain the image size consistent of each dataset and to reduce the computational cost. Details of the datasets are listed in Table 3. Several example images of these datasets are shown in Fig 2.

Figure 2: Example images of three datasets

3.2 Baseline methods

To show the effectiveness of the proposed MLGP approach, six methods are used as baseline methods for comparisons. These methods are based on commonly used image segmentation methods.

3.3 Parameter Settings

All the GP-based algorithm are implemented in Python based on the DEAP (Dis- tributed Evolutionary Algorithm in Python) [8] package. Parameter settings for all the GP-based algorithms are the same as listed in Table 4.5. In these GP-based algorithms, the tree generation method is ramped half-and-half. On each dataset, each algorithm has been run for 30 times independently with different random seeds.

Method	Description
Thresholding	An image processing method that creates a bitonal (aka binary) image based on setting a threshold value on the pixel intensity of the original image.[11]
Canny_edge	Canny Edge Detection is used to detect the edges in an image.
ChanVese	The Chan-Vese segmentation algorithm is designed to segment objects without clearly defined boundaries.[12]
WaterShed	Watersheds separate basins from each other. The watershed transform decomposes an image completely and thus assigns each pixel either to a region or a watershed.[13]
Snake	A snake is an energy minimizing, deformable spline influenced by constraint and image forces that pull it towards object contours and internal forces that resist deformation. Snakes may be understood as a special case of the general technique of matching a deformable model to an image by means of energy minimization.[14]
Walk	The random walker algorithm was initially motivated by labeling a pixel as object/background based on the probability that a random walker dropped at the pixel would first reach an object (foreground) seed or a background seed. However, there are several other interpretations of this same algorithm that have appeared.[15]

Table 4: Baseline methods

4 Results and Discussions

The experimental results of the MLGP approach and the six baseline methods are listed in Table 5. The Wilcoxon rank-sum test with a 5% significance level is employed to compare the proposed approach with baseline methods. In Table 5, the "+" symbol indicates that the proposed approach is significantly better than the compared method, the "" symbol indicates the proposed approach performs significantly worse than the compared method, and the "=" symbol indicates that the proposed approach achieves similar results to the compared method.

4.1 Compared with baseline methods

Table 5 lists all the test results of MLGP and the total 6 baseline methods on the three datasets. On the Sprat dataset,the results of the proposed method is shown as mean and its standard deviation.The proposed method achieves significantly better result in precision than any of the 6 baseline methods in 4 performance metrics. The performace of MLGP is better than other methods in 4 performance metrics but is worse than Walk in f1-score, Canny_edge in accuracy, Chanvese and walk in recall.

On the Gilt dataset, the proposed method achieves significantly better result in f1-score and precision than any of the 6 baseline methods in 4 performance metrics. The performace of MLGP is better than other methods in 4 perfor-

Methods	F1-score	Accuracy	Recall	Precision
Sprat				
MLGP	41.61 ±0.5	83.73 ±0.52	54.7 ±2.52	38.19 ±1.13
Thresholding	9.32 +	63.72 +	18.26 +	6.49 +
Canny_edge	31.02 +	85.48 -	34.15 +	29.42 +
ChanVese	22.67 +	40.0 +	87.41 -	13.29 +
WaterShed	35.8 +	77.85 +	52.77 +	30.95 +
Snakes	13.4 +	47.49 +	50.75 +	8.15 +
Walk	46.92 -	82.16 +	56.08 -	35.55 +
Gilt				
MLGP	60.01 ±0.21	80 ±0.65	61.5 ±2.53	61.01 ±3.65
Thresholding	8.57 +	54.59 +	15.06 +	6.16 +
Canny_edge	30.61 +	81.89 -	29.06 +	32.97 +
ChanVese	31.38 +	46.02 +	87.36 -	19.43 +
WaterShed	40.25 +	75.37 +	53.09 +	35.51 +
Snakes	15.62 +	49.31 +	41.58 +	10.06 +
Walk	46.92 +	80.46 -	56.83 +	43.66 +
Hourse				
MLGP	47.87 ±0.69	81 ±1.31	57.59 ±2.76	44.95 ±2.16
Thresholding	16.52 +	40.34 +	24.06 +	13.15 +
Canny_edge	34.74 +	75.88 +	26.76 +	51.16 -
ChanVese	51.85 -	64.26 +	77.43 -	40.46 +
WaterShed	52.09 -	78.76 +	46.0 +	66.32 -
Snakes	26.34 +	41.92 +	42.68 +	20.0 +
Walk	57.02 -	81.99 -	48.6 +	74.84 -

Table 5: Four performance metrics of MLGP and the 6 baseline methods on the three datasets

mance metrics but is worse than Canny_edge and walk in accuracy and Chanvese in recall.

On the Hourse dataset, the performace of MLGP is better than any other methods in 4 performance metrics but is worse than ChanVese, WaterShed and Walk in f1-score, Walk in accuracy, ChanVese in recall and Canny_edge, WaterShed and Walk in precision.

In summary, the comparisons and discussions show that the proposed MLGP approach is a promising approach for image segmentation. With a single-tree representation, the proposed MLGP approach can perform region detection, region construction, and segmentation simultaneously and automatically.

5 Further Analysis

6 Summary

This paper presented the MLGP approach where the tree/program structure includes an input layer, a region detection layer, a region construction layer, and a segmentation layer for image segmentation. With the new program structure and the new function set, the MLGP approach can achieve automatic region detection, region construction, and image segmentation, simultaneously. The performance of the MLGP approach was examined on three image datasets of varying difficulty and compared with six methods. The experimental results showed that the MLGP approach achieved significantly better or comparable results than the six baseline methods. The MLGP approach can evolve very simple programs that can achieve high or perfect accuracy.

This paper showed how a single GP tree can be effectively used to automatically and simultaneously perform region detection, region construction, and image segmentation. With a different program structure or a different function set, the way of GP for image segmentation can be different. There is still a big research space in this direction.