Genetic Programming for Evolving Figure-ground Segmentors from Multiple Features

Yuyu Liang, Mengjie Zhang and Will N. Browne

School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand

Abstract

Figure-ground segmentation is a crucial preprocessing step for many image processing and computer vision tasks. Since different classes of objects need specific segmentation rules, top-down approaches, which learn from object information, have been shown to be more suitable for this task than bottom-up approaches. A problem faced by most existing top-down approaches is that they require much human work/intervention, so human bias is introduced at the same time. As genetic programming (GP) does not require users to specify the structure of solutions, we apply it to evolve segmentors, aiming at conducting the figure-ground segmentation task automatically and accurately. Seven different terminal sets are exploited to evolve segmentors. Images from four datasets (bitmap, Brodatz texture, Weizmann and Pascal databases), which are increasingly diverse and difficult for segmentation tasks, are selected to test the evolved segmentors. Results show that the proposed GP based method can be successfully applied to diverse types of images. In addition, intensity based features are not sufficient for complex images, whereas features containing spectral and statistical information are necessary. Compared with four widely-used segmentation techniques, our method obtains consistently better segmentation performance.

Keywords: Figure-ground segmentation, Genetic programming, Intensity based features, Gabor features

1. Introduction

Figure-ground segmentation is a basic computer vision task, which aims to separate foreground objects or regions of interest from their backgrounds.

The results of figure-ground segmentations can be the input to many higher-level tasks, e.g. object recognition, object tracking and image editing [1].

There are two ways to conduct the figure-ground segmentation: bottom-up approach and top-down approach [2]. The former aims to segment the image into regions first, then recognize image regions that belong to objects or the background based on the image-based criteria (e.g. texture uniformity and continuity of contours). However, this approach is likely to segment an object to multiple parts, and merge the object and background parts together as they are similar based on the image-based criteria [2, 3], thus reducing the segmentation accuracy. In contrast, unlike the bottom-up approach, top-down approach does not rely on the image criteria, but learns from the prior knowledge of objects, such as object classes and shape fragments, to direct the figure-ground segmentation. As different types of images may contain different classes of objects, they require specific segmentation rules. Therefore, it is preferable to apply top-down approach that can learn rules from the object information provided by the training samples [2] rather than attempt to manually design rules.

Existing top-down figure-ground segmentation approach can be divided into two branches: model-based methods [4, 5, 6] and fragment-based methods [2, 7, 8]. Model-based methods include deformable templates, active shape models, active contour models, etc.. They normally match a deformable model to an image by minimizing the image energy, which is a function of image features [9]. Fragment-based methods extract a fragment set in the training stage, which captures shape and appearance information of the common object parts. Each fragment's figure-ground segmentation is then generated and used to match objects in the testing stage. However, model-based methods require a lot of human intervention to locate the initial models, and only when the initial model is located close to the target is it likely to obtain accurate segmentation results. Fragment-based methods require much work from researchers to collect informative fragments. Furthermore, the more involvement of human work, the higher probability of involving human bias, which should be avoided as much as possible.

As an evolutionary computation technique, genetic programming (GP) can evolve computer programs to solve problems automatically, so users are not required to specify the form or structure of solutions [10]. If GP can be introduced to evolve segmentation algorithms, the GP based methods will require less human work than the model-based and fragment-based methods. Moreover, since GP can combine input features in complex non-linear forms, GP is more flexible and more likely to evolve better methods than those defined by experts [11].

Although GP has already been introduced in the area of image segmentation since the 1990s, only a limited number of related papers have been published. Based on these works, GP is used in two ways for image segmentation — one is to apply GP directly to evolve segmentors [11, 12, 13, 14, 15, 16]; the other is to combine GP with other conventional segmentation techniques (e.g. clustering methods [17]) and utilize GP as an optimization technique [17]. The first one is pure GP approach and can evolve segmentors automatically, so our paper will also follow this approach.

The existing work, using GP to evolve segmentors, actually transforms the segmentation tasks into an extension of classification problems and evolves pixelwise classifiers. Poli [11] proposes a method using GP to evolve filters, based on which the pixelwise classification can be conducted to fulfill segmentation tasks. The terminal set contains average intensity values of small image regions, and the function set consists of simple arithmetic functions. The evolved algorithm is tested on several medical images, achieving much higher scores in sensitivity and specificity than an artificial neural network approach. The GP based segmentation method has also been utilized on texture segmentation [12, 13]. As the terminal set only includes raw pixel values, this method does not involve feature extraction, so it is efficient. The function set consists of arithmetic and logic operators (such as >=, <=, ==). The method can segment texture images accurately. A recent work [16] builds a GP based segmentation system adapted from works of Song et al. [12, 13]. It employs GP to evolve segmentors from local binary patterns (LBP) to segment the corpora lutea on 30 medical images.

Image operators can be added to the function set (such as dilation, thresholding and histogram equalization). Singh et al. [14] evolve Matlab programs from intensities and primitive image operators, which produce consistently good results on cell images. Roberts [15] employs not only arithmetic functions but also image processing operators. To tackle the problem of a high computational cost caused by image operators, a caching mechanism was introduced. The GP based method achieves 92.3% in sensitivity and 97.2% in specificity on mole images.

These works on GP based segmentation achieve promising results. However, there are still several drawbacks. Firstly, they are only tested on simple images in a limited number of domains, including texture images [12, 13], and medical images [11, 14, 15]. As the tested medical images all have clean backgrounds, they are considered as simple images. Therefore, whether GP-evolved segmentors can deal with a wide range of images, especially complex images with variations in background, is still not clear. Secondly, only intensity based features [12, 13, 14, 15], intensity statistics [11] and LBP [16] have been used as input features to evolve segmentors. As the input of GP has a great influence on its performance [18], we assume that if the terminal set contains certain kinds of features known to suit target image domains (e.g. histogram statistics and Gabor features), better performing segmentors will be evolved. Thirdly, limited comparisons with other segmentation techniques have been made. GP-evolved segmentors have been compared only with neural networks [11] and a GA-based algorithm [14]. To test the effectiveness of segmentors, comparisons with the widely-used approaches, e.g. thresholding, clustering and region-based methods, are necessary.

1.1. Goals

This paper is an extension of our former work [19], which utilizes GP to evolve segmentors from three simple feature sets, i.e. intensities, histogram statistics and “histogram statistics + spatial moments + gradient statistics”. In this paper, we exploit more feature descriptors, which can extract all the three types of general image properties (color, texture and shape) consisting of seven feature sets. It aims to determine what kind of image information is necessary for GP to evolve capable segmentors (especially for complex images, such as images with high variations in the objects of interest and/or background). The explicit objectives are as follows.

1. Investigate whether the GP evolved segmentors can have a consistent performance across diverse ranges of images, which have different difficulty levels for the segmentation task.
2. Explore the effectiveness of different image features on the performance of evolved segmentors.
3. Investigate whether the GP based segmentation method can outperform widely-used conventional techniques.

1.2. Organisation

The rest of this paper is organized as follows. In section 2, the basic knowledge of GP is provided. Section 3 describes our GP based segmentation system and the seven feature descriptors selected to evolve segmentors. Section 4 introduces experiment preparations of test images and evaluation measures. Section 5 displays results obtained on four databases. In section 6, further discussions about the evolved segmentors and comparisons with conventional segmentation techniques are made. Conclusions are drawn in section 7.

2. Background – GP

GP is one of the evolutionary computation techniques, which are inspired by the biological evolution [10]. It can evolve a population of computer programs by transforming populations of programs into new, normally better, populations. Figure 1 displays the flowchart of GP, and the basic steps of GP are described as follows.

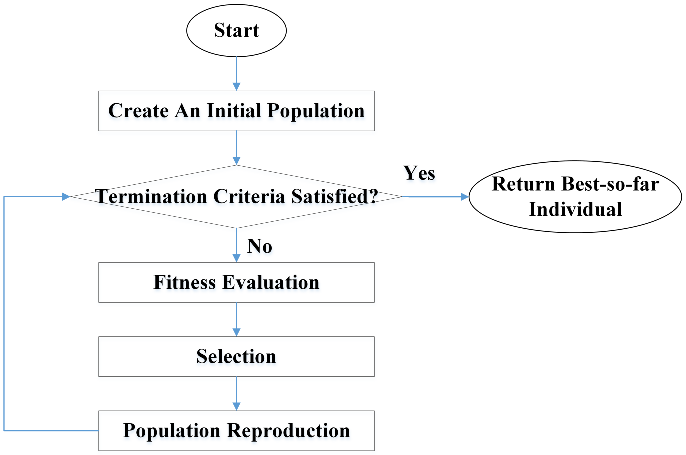

Start
Create An Initial Population
Termination Criteria Satisfied?
Yes
Return Best-so-far Individual
No
Fitness Evaluation
Selection
Population Reproduction

Figure 1: The framework of GP (adapted from Poli's work [10]).

1. Initialise a GP population.
   This is the first step to perform a GP run, in which a variety of individuals are created randomly for the later evolution. There are three common structures used in GP: tree, linear and graph structure. Among them, the tree structure is the most commonly used one, which is also employed in this paper and will be described in subsection 2.1. The initialisation methods in tree based GP include *full*, *grow* and *Ramped half-and-half* [10]. For the *Ramped half-and-half* employed in this paper, half of the initial population is constructed using the *full* method and half is constructed using the *grow* method.

2. Execute each program and evaluate its fitness.
   Execute each individual (program) on the target problem at hand. A fitness function should be designed to evaluate how well each individual has learned to solve the target problem.
3. Select individual(s) from the population based on the fitness value to participate in genetic operations.
   Individuals that have higher fitness values are more likely to be selected for the reproduction method. There are various selection methods, e.g.*fitness-proportional selection*, *truncation selection* and *tournament selection* [20]. The *tournament selection* method is the most commonly used one and is also employed in GP by our method. There are two steps in tournament selection. Firstly, a number of individuals are chosen randomly from the population. Secondly, the selected individuals are compared with each other and the best one is chosen to be the parent. For the genetic operator – crossover, which needs two parents, two selection tournaments are conducted.
4. Create new individual program(s) by applying genetic operations
   The initial population is likely to contain low fitness solutions, so the evolution process is needed to transform the initial population by genetic operators to a new population generation by generation. There are three principal genetic operators: *crossover, mutation* and *elitism.* How the operators are carried out will be presented in section 2.2.
5. Stopping criteria.
   Two common stopping criteria are that a maximal number of generation is reached or an acceptable solution is found.

2.1. Individual Representation in Tree based GP

Each program in the tree structure is built from primitives in the function set and the terminal set. The terminal set contains the inputs (variables or constants) to GP programs and the function set consists of operators, functions and statements. Both the terminal set and the function set are problem-specific and should be provided by users. Figure 2 presents a tree-structured individual. The operators in the red circle are from the function set and are internal tree nodes; while the terminal nodes in the blue circle are from the terminal set.

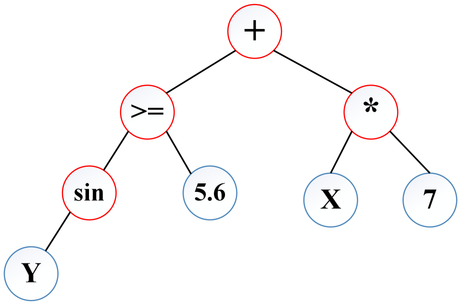


Figure 2: An individual in tree structure (sin means Sine function; X, Y are variables).

2.2. Genetic Operators in Tree based GP

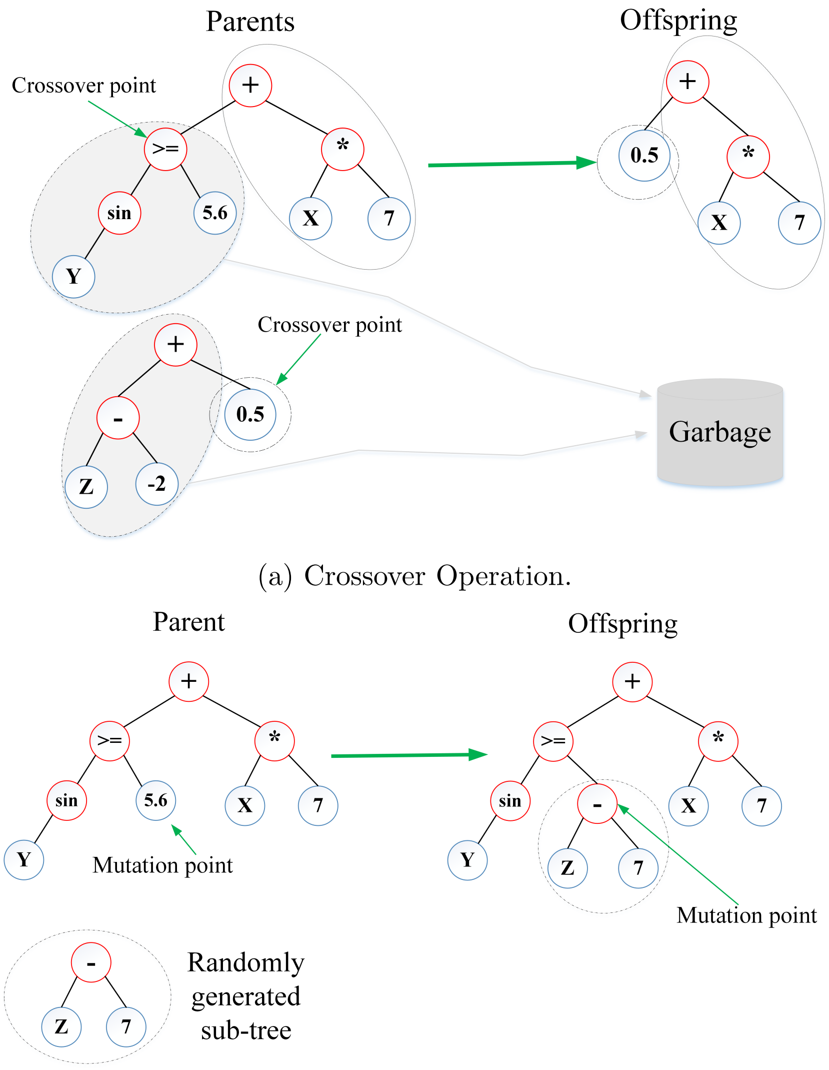
This part will discuss the three principal genetic operators: crossover, mutation and elitism. The crossover operation is shown in Figure 3a. As can be seen, two parents are needed. Firstly, a crossover point is selected randomly in each parent tree. Next, the offspring is created by replacing the subtrees rooted at the crossover point. The mutation operation is shown in Figure 3b. There are also two stages: select a mutation point and then replace the subtree rooted there with a subtree generated randomly. The elitism operator selects the best individuals based on the fitness and simply copies them to the next generation.

3. The GP based Image Segmentation System

According to the existing works [11, 12, 13, 14, 15, 16], which use GP to evolve algorithms, image segmentation is handled as a supervised classification problem at the pixel level. In Song's work [12, 13], a pixel-classification-based segmentation framework is developed, which is a common way to use GP in image-related tasks. Our paper also follows this framework. We exploit seven terminal sets to discover what kind of image information is useful to segment diverse types of images, especially complex images.

3.1. Construction of the Segmentation System

As displayed in Figure 4, there are three major steps in the GP based segmentation system. The first step is to evolve a classifier: an equal number of cutouts (sub-images) labeled as object or background are generated from training images. Features extracted from these cutouts form inputs to the system to evolve a pixel-wise classifier. In the second step, a sliding window,



(a) Crossover Operation.

(b) Mutation Operation.

Figure 3: Genetic operators (*sin* means Sine function; X, Y are variables).

whose size is the same as that of the cutouts on the first step, is used to sweep across the test images. The evolved classifier categorizes the sub-images captured by the window as object or background, and a class label is assigned to all pixels of each sub-image. Finally, as each pixel of the test images may have more than one assigned label, a majority voting process is employed to obtain the final estimated label, based on which segmentation results are produced.

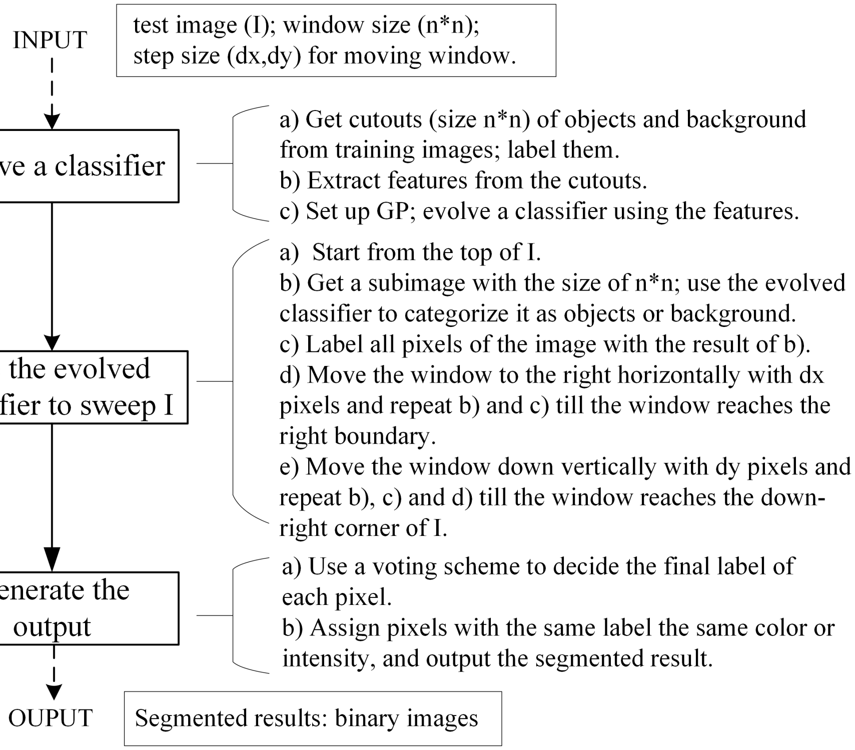

INPUT
test image (I); window size (n*n);
step size (dx,dy) for moving window.
Evolve a classifier
a) Get cutouts (size n*n) of objects and background from training images; label them.
b) Extract features from the cutouts.
c) Set up GP; evolve a classifier using the features.
Use the evolved classifier to sweep I
a) Start from the top of I.
b) Get a subimage with the size of n*n; use the evolved classifier to categorize it as objects or background.
c) Label all pixels of the image with the result of b).
d) Move the window to the right horizontally with dx pixels and repeat b) and c) till the window reaches the right boundary.
e) Move the window down vertically with dy pixels and repeat b), c) and d) till the window reaches the down-right corner of I.
Generate the output
a) Use a voting scheme to decide the final label of each pixel.
b) Assign pixels with the same label the same color or intensity, and output the segmented result.
OUPUT
Segmented results: binary images

Figure 4: The flowchart of the GP based segmentation system (adapted from [12]).

Table 1 displays the function set and Table 2 is the terminal set. From Table 1, it can be seen that only simple arithmetic and logic operators are employed. We do not include image operators since they involve more calculation, thus being costly in time. As one goal of this paper is to investigate the influence of features on evolved segmentors, we test seven terminal sets (Table 2), which contain different kinds of features. They cover all the three types of general features (brightness, texture and shape) used in the image segmentation field. We will describe them explicitly in the next subsection.

The classification accuracy is selected as the fitness function (Equation 1). In addition, GP set-up parameters are listed as follows. The number of generations is 51, and the population size is 500 (from Song [12, 13]). The reproduction operators are crossover and mutation, whose rates are 90% and

10% respectively. The other GP parameters follow Koza's settings [18].

$$f = \frac{Number.of.correctly.classified.samples}{Number.of.total.training.samples} \tag{1}$$

Table 1: Function set.

Function Name	**Definition**	**Type**
Add(a_1, a_2)	$a_1 + a_2$	Arithmetic
Sub(a_1, a_2)	$a_1 - a_2$	Arithmetic
Mul(a_1, a_2)	$a_1 * a_2$	Arithmetic
Div(a_1, a_2)	$\begin{cases} a_1/a_2 & \text{if } a_2! = 0 \\ 0 & \text{if } a_2 == 0 \end{cases}$	Arithmetic
IF(a_1, a_2, a_3)	$\begin{cases} a_2 & \text{if } a_1 \text{ is true.} \\ a_3 & \text{if } a_1 \text{ is false.} \end{cases}$	Conditional
$<= (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 <= a_2 \\ false & \text{if otherwise} \end{cases}$	Conditional
$>= (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 >= a_2 \\ false & \text{if otherwise} \end{cases}$	Conditional
$= (a_1, a_2)$	$\begin{cases} true & \text{if } a_1 == a_2 \\ false & \text{if otherwise} \end{cases}$	Conditional
Between(a_1, a_2, a_3)	$\begin{cases} true & \text{if } a_2 <= a_1 <= a_3 \\ false & \text{if otherwise} \end{cases}$	Conditional

3.2. Features

In this paper, seven commonly-used feature descriptors are investigated to evolve image segmentors. Image segmentation is based on three basic kinds of characteristics or calculated properties, i.e. color (or brightness for grayscale images), texture and shape [21, 22]. The selected features cover all the three categories, i.e. raw pixel values as the brightness feature; histogram statistics, GLCM (gray level co-occurrence matrices) statistics, LBP, Fourier power spectrum and Gabor as texture features; spatial moments and gradient statistics as shape features (shown in Table 3).

Specifically, intensities are the gray levels of pixels. The histogram statistics are first order measures [11] and six statistics are selected, i.e. mean, variance, skewness, kurtosis, energy and entropy. GLCM is a second-order

Table 2: Terminal sets (GLCM means gray level co-occurrence matrices; LBP means local binary patterns).

Name	**Description**
Terminal Set 1	Raw Pixel Values
Terminal Set 2	Histogram Statistics
Terminal Set 3	GLCM Statistics
Terminal Set 4	LBP Features
Terminal Set 5	Fourier Power Spectrum
Terminal Set 6	Gabor Features
Terminal Set 7	Moments+Gradient Statistics

statistical method that considers the spatial dependence of pixels [23]. Four types of statistics are derived from GLCM, i.e. contrast, correlation, energy and homogeneity. LBP descriptor transforms an image to an array of integer labels, which can represent the small-scale appearance of the image [24]. The Fourier power spectrum of an image or a region is the square of the magnitude of its Fourier transform [25]. Gabor filters [26] can extract image information in different scales and orientations, which is one of the most powerful local appearance descriptors. The first and second order spatial moments can describe the distribution of image regions [27]. Image gradients are directional changes in intensities or color in an image, and are able to encode edges and local contrast [28]. Three gradient statistics, the maximum, minimum and average gradient of a region, are extracted.

Equations or parameters in Table 3 are based on the following definitions. For Gabor features, $scale = 5$ means the number of filter scales is five; $orientation = 8$ represents the number of filter orientations is eight; $d_1 = 4, d_2 = 4$ means the downsampling ratio in row and column respectively. For moments, given a square region with the size of $n * n$, $f(x_i, y_i)$ is the pixel value; $d(x_i, y_i)$ is the distance of the pixel (x_i, y_i) to the top left corner of the region.

4. Experiment Preparation

We select images from four databases (section 3.1), i.e. bitmap patterns [13], Brodatz texture database [29], Weizmann horse database [30, 2] and Pascal Visual Object Classes 2012 (Pascal VOC2012) dataset [31], to test our GP based method. This paper aims to investigate whether the GP-evolved segmentors can deal with diverse types of images. These four datasets are

Table 3: Description of features (M_1 and M_2 are the first order moment and the second order moment respectively; Min, Max and Ave mean the the maximum, minimum and average gradient respectively).

Category	**Feature**		**Description or Parameter**
Brightness	Raw Pixel Values		-
Texture	Statistical Methods	Histogram Statistics	Mean, Variance, Skewness, Kurtosis, Energy, Entropy
		GLCM Statistics	Contrast,Correlation,Energy,Homogeneity
		LBP	P=8, R=1(P is the number of sampling points in a region with the radius R.)
	Structural Methods	FourierPowerSpectrum	-
		Gabor	$scale = 5, orientation = 8, d_1 = 4, d_2 = 4$
Shape	Moments + Gradient Statistics		$M_1 = \frac{\sum_{i=1}^{n} f(x_i,y_i)*d(x_i,y_i)}{n}$ $M_2 = \frac{\sum_{i=1}^{n}(f(x_i,y_i)*d(x_i,y_i)-M1)^2}{n}$ Min,Max,Ave Gradient

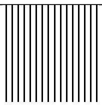
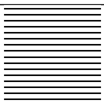
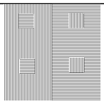
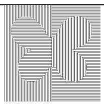
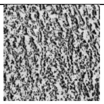
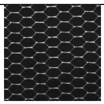
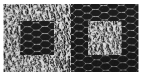
selected as they are diversed increasingly difficult for the segmentation task. Three types of evaluation methods (section 3.2), the segmentation accuracy, F_1 score [32] and negative rate metric (NRM) [33], are applied to evaluate the evolved segmentors.

4.1. Image Databases

As this preliminary work focuses on the development of a new GP based segmentation approach rather than the quantitative performance evaluation, we select a subset of images from four datasets (displayed in Table 4). In practice, grayscale, large size and real images with high variations in object/background are more complex than binary, synthetic or small size images for segmentation tasks [11].

In Table 4, there are two bitmap images, named as "Rectangular" and "Butterfly". They are synthesized from two bitmap patterns, P14 and P24. The two images are synthetic binary images, which are relatively simple. One example texture image, D24vs34, is a grayscale image synthesized from two Brodatz textures that are D24 and D34. In addition, images from the Weizmann and Pascal databases are real images and have high variations. The ten grayscale horse images from Weizmann are varied in horse positions (e.g. standing, running and eating). Each image contains one object (horse), and all the horses are located in the center of images. The eight passenger air plane images from Pascal VOC2012 are the biggest images. Certain images contain more than one object, and the object shapes and sizes vary a lot. Therefore, the segmentation of Pascal images is considered as the most difficult task.

Table 4: Four types of images used in this paper

Database	Images	Descriptions
Bitmap	 P14  P24  Rectangular  Butterfly	Size: 256*256 Synthetic, binary images
Brodatz	 D24  D34  D24vs34	Size: 320*160 Grayscale images
Weizmann	horse006 horse010 horse027 horse110 horse119	Average Size:248*211 Real images
	horse121 horse122 horse159 horse165 horse317	Varing horse positions One object
Pascal	0033 0256 0738 1288	Average Size:500*350 Real images
(Name prefix: 2007_00)	1761 2099 2266 2376	Varing object locations/sizes Multiple objects

4.2. Evaluation Methods

The first evaluation method we selected is the segmentation accuracy (Equation 2). This measure is simple and commonly used, but it is insufficient to assess segmented results. For example, when objects take up a small proportion of an image, if the whole image is incorrectly segmented as background, the segmentation accuracy would still be quite high. Therefore, another two evaluation methods (F_1 score (Equation 3) and NRM (Equation 4)) are also selected to compensate this measure.

$$accuracy = \frac{TP + TN}{Total.Pixel.Number.In.An.Image} \tag{2}$$

F_1 score combines precision and recall together, which is a measure of classification accuracies. However, it fails to consider the true negative rate

[32]. NRM takes mismatches between a prediction and the ground truth into account [33], and it is a single measure combining false negative rate and false positive rate. F_1 reaches its worst at 0 and best value at 1; while NRM is worst at 1 and best at 0.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

$$NRM = (FNR + FPR)/2 \tag{4}$$

Where $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $FNR = \frac{FN}{TP+FN}$, $FPR = \frac{FP}{FP+TN}$. TP, TN, FP and FN stand for true positive, true negative, false positive and false negative respectively. In the context of classification, TP means desired objects are correctly classified by a classifier; TN means non-objects are correctly classified as background; FP represents non-objects are incorrectly classified as objects; FN represents objects are incorrectly classified as background. FPR and FNR represent false positive rate and false negative rate respectively.

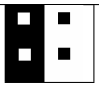
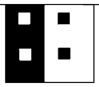
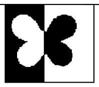
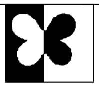
5. Experiment Results

This section tests the segmentors evolved from seven kinds of features on four datasets. For bitmap and texture images, there are two different patterns or textures in each image, so we choose one pattern as the object and the other one as background randomly. In addition to GP parameters, the size of the sliding window and its shifting step are also needed by the system, which are set based on our initial experimental tests. As the window captures sub-images, the window size must guarantee that each sub-image contains sufficient information to distinguish itself from those belonging to other classes. We set it to 4 for bitmap images and 16 for other test images. A big shifting step of the window may cause inaccurate results, yet a small step involves a high computation cost. Both shifting steps (d_x in the horizontal direction and d_y in the vertical direction) of the window are set to 2 for all the images in this paper. Each experiment runs 30 times due to the stochastic nature of GP.

5.1. Results on Bitmap Images

The training set has 2000 samples including 1000 bitmap P14 (Table 4) features labeled as background and 1000 bitmap P24 (Table 4) features labeled as objects. As bitmap images are binary images that can be described well by pixel values, we only test the intensity based segmentors on these images. From Table 5, result examples show that regions of different patterns are segmented accurately. Not only the linear boundaries of the "Rectangular" image but also the curve boundaries of the "Butterfly" image have been located precisely. In terms of statistical analysis, high segmentation accuracies, high F_1 scores and low NRM scores are obtained for both images.

Table 5: Results of intensity-based segmentors on bitmap images.

Ground Truth	**Result Examples**	**Accuracy(%)**	F_1	**NMR**
		98.81 ± 0.22	0.99	0.20
		95.96 ± 1.27	0.96	0.12

5.2. Results on Brodatz Texture Images

There are also 2000 training samples generated equally from D24 texture (4) labeled as background and D34 texture (4) labeled as object.

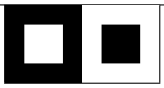
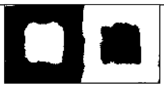
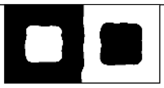
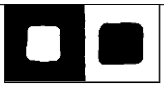
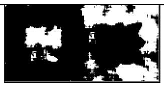
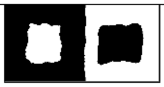
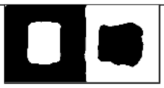
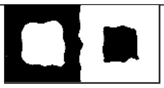
According to Table 6, the performance of evolved segmentors is generally good on the texture image with most accuracies over 90% and F_1 over 0.90. Especially, intensity and histogram statistics based segmentors perform best. Considering they are relatively simple features, the performance is impressive. Therefore, the results suggest that it may be not necessary to use complex features (like GLCM statistics and Gabor), which involve high calculation costs to evolve segmentors for texture images.

5.3. Results on Weizmann Horse

As we only use ten images from the Weizmann dataset, the leave-one-out (LOO) cross-validation [34] is employed in this part. Each training image provides 200 training samples, consisting of 100 object samples and 100 background samples.

Result examples in Table 7 suggest that segmentors evolved from intensities, histogram statistics and Gabor perform better as they locate horses accurately and generate clearer boundaries; while segmentors generated from LBP, Fourier power spectrum and "spatial moments + gradient statistics" perform unsatisfactorily. Results in Table 8 verify that histogram statistics

Table 6: Results of segmentors evolved from different features on the texture image

Feature	**Result Examples**	**Accuracy(%)**	F_1	**NMR**
Ground Truth		-	-	-
Intensity		**94.26± 2.75**	**0.94**	0.15
Histogram Statistics		93.98 ± 2.30	**0.94**	**0.07**
GLCM Statistics		92.67 ± 1.45	0.92	0.15
LBP		66.82 ± 10.06	0.53	0.35
Fourier Power Spectrum		91.16 ± 0.94	0.90	0.13
Gabor		90.91 ± 0.72	0.90	0.15
Moments + Gradient		92.02 ± 2.11	0.92	0.39

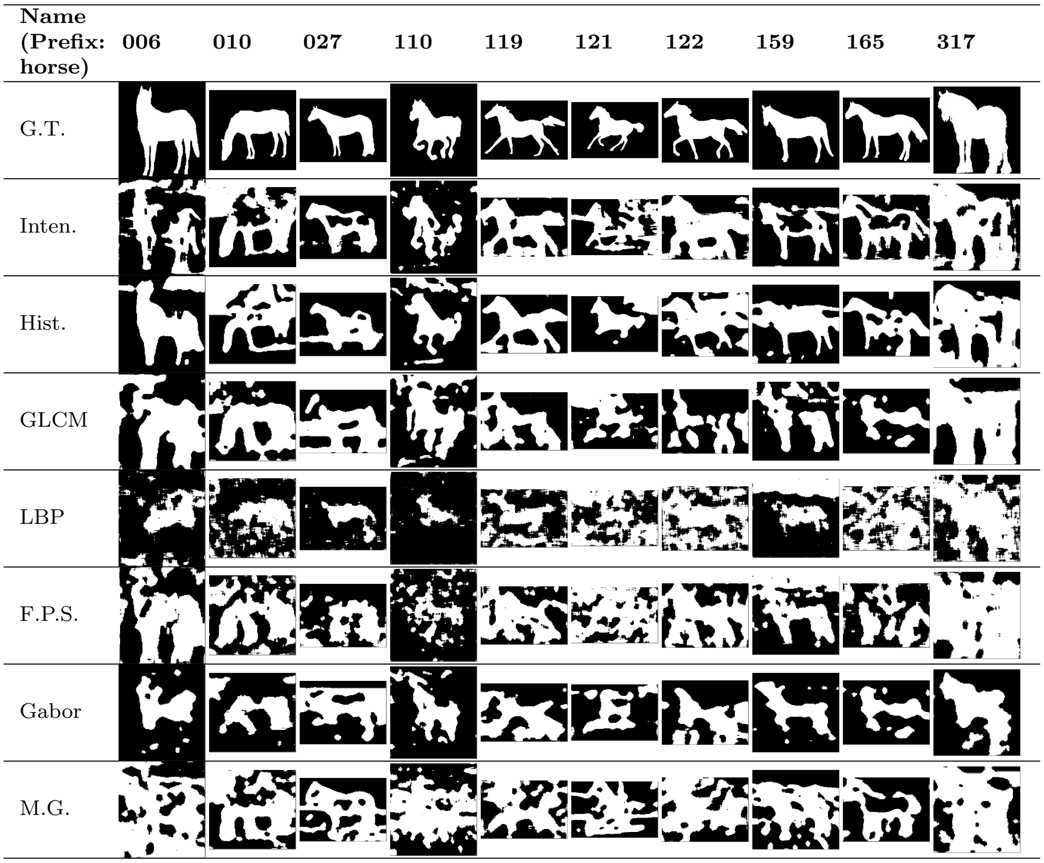
and Gabor based segmentors obtain the best scores in the three evaluation measures. It reflects that when testing on real images rather than simple bitmap or texture images, higher-level features, such as histogram statistics and Gabor, can evolve segmentors with better performance than those based on intensities.

5.4. Experiments on Pascal Database

As eight Pascal images are used in this paper, LOO cross-validation tests are employed on these images. Due to the large size of images, more samples are generated from each training images than that on the Weizmann dataset. Each image provides 500 samples equally generated from object and background.

As explained previously, Pascal images are the most complex test images. According to result examples in Table 9 and statistical results in Table 10, Gabor, Fourier, histogram statistics and "moments + gradient statistics" based segmentors obtain similar results with segmentation accuracies around

Table 7: Result examples on the Weizmann images (G.T.: ground truth; Inten.: intensity; Hist.: histogram; F.P.S.: Fourier power spectrum; M.G.: moments + gradient statistics.)



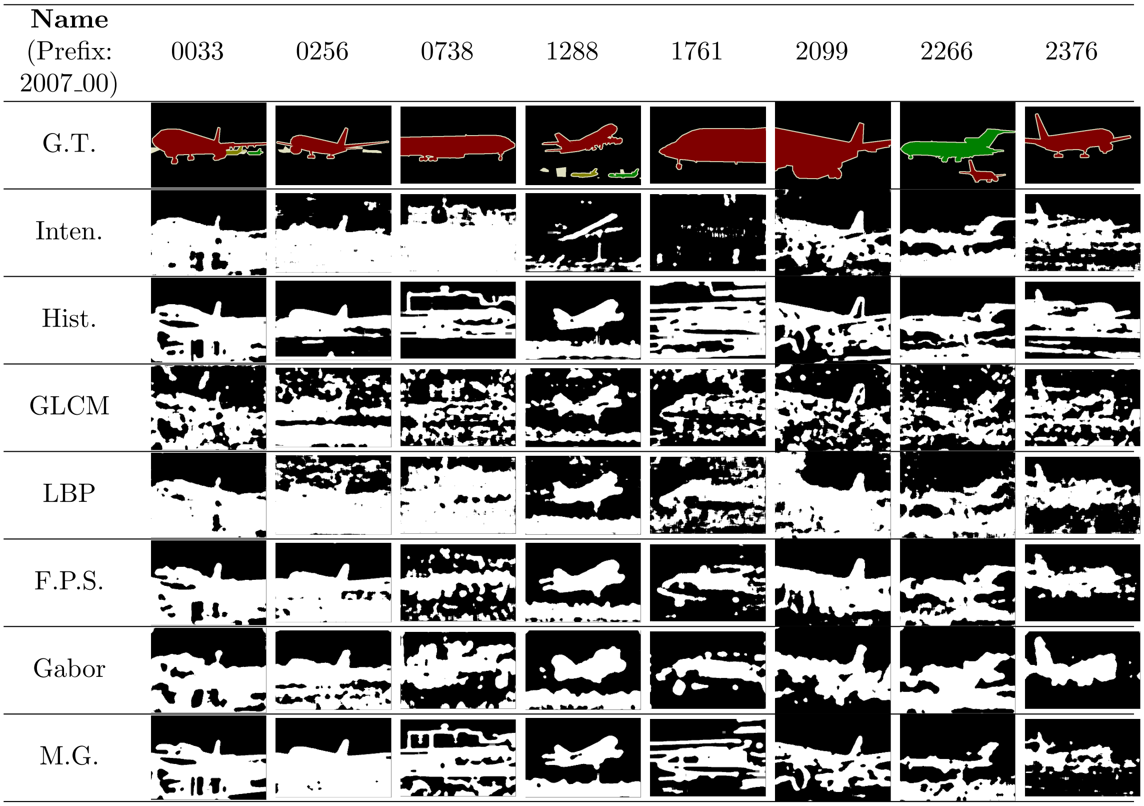
Name (Prefix: horse)
006
010
027
110
119
121
122
159
165
317
G.T.
Inten.
Hist.
GLCM
LBP
F.P.S.
Gabor
M.G.

Table 8: Statistical results on the Weizmann images (the best result is in bold)

Feature	**Accuracy (%)**	F_1	**NRM**
Intensity	74.41 ± 8.37	0.62	0.47
Histogram Statistics	77.37 ± 9.09	**0.84**	0.47
GLCM Statistics	76.74 ± 3.92	0.68	0.47
LBP	66.19 ± 10.95	0.52	0.48
Fourier	68.38 ± 7.38	0.61	0.50
Gabor	**78.29±5.40**	0.66	**0.42**
Moments + Gradient statistics	65.04 ± 10.39	0.58	0.50

75%. Among them, the Gabor based segmentation results are the best. The intensity based segmentors obtain an average segmentation accuracy over 71%, yet the F_1 score is the lowest. That is because large parts of background in some result images have been segmented as object.

Table 9: Result examples on Pascal images (G.T.: ground truth; Inten.: intensity; Hist.: histogram; F.P.S.: Fourier power spectrum; M.G.: moments + gradient statistics.)



Therefore, even though intensities may be sufficient to evolve segmentors for simple images (e.g. synthetic images, binary or texture images), higher-level features, containing statistical and spectural information, are necessary to segment complex images (e.g. images with many variations in the objects of interest and/or background).

6. Further Discussions

To find out the influence of input features on the performance of evolved segmentors, the results on Brodatz, Weizmann and Pascal images, are com-

Table 10: Statistical results on Pascal images (The best result is in bold)

Feature	**Accuracy (%)**	F_1	**NRM**
Intensity	71.39 ± 10.63	0.49	0.50
Histogram Statistics	74.56 ± 6.89	0.61	0.50
GLCM Statistics	67.39 ± 9.60	0.49	0.52
LBP	63.75 ± 14.07	0.54	0.50
Fourier	75.10 ± 7.90	0.61	**0.46**
Gabor	**75.60±8.10**	**0.62**	**0.46**
Moments + Gradient statistics	74.53 ± 7.83	0.59	0.48

pared. In addition, we also compare the Gabor-based segmentors, which perform best, with other four widely-used segmentation techniques.

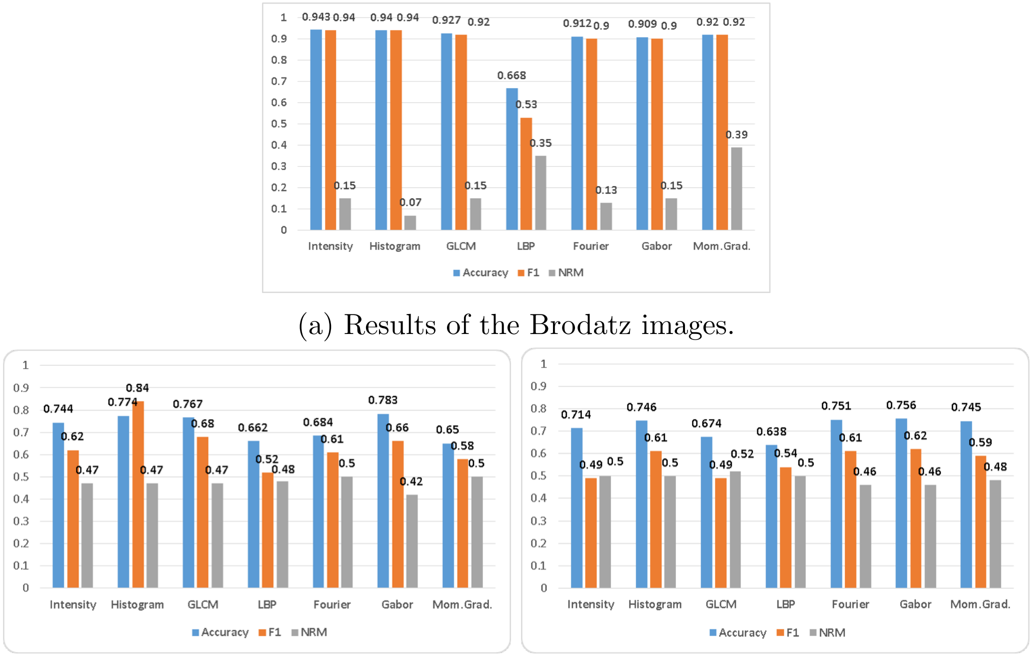
6.1. Comparison between Evolved Segmentors

Considering only a limited number of images are used, most segmentors perform satisfactorily (shown in Figure 5). Based on Figure 5a, most evolved segmentors achieve over 90% accuracy, over 0.9 F_1 score and around 0.1 NRM score on texture images, except LBP and "moments + gradient statistics" based ones. According to Figure 5b, two kinds of segmentors (histogram statistics and Gabor based ones) are the best. Figure 5c shows that four kinds of segmentors (histogram statistics, Fourier power spectrum, Gabor and "moments + gradient statistics" based ones) produce higher scores than others.

According to the analyses, from simple texture images to complicated real images, histogram statistics and Gabor features based segmentors produce consistently better results than others. In contrast, intensity based segmentors are the best on texture images. Yet, they lag far behind on the Weizmann and Pascal images. In terms of the segmentation accuracy, F_1 and NRM, intensity-based segmentors are 3.8% lower, 0.22 lower, 0.05 higher on Weizmann images; 3.7% lower, 0.12 lower, 0.04 higher on Pascal images than the best scores. The obvious changes reflect higher-level information, such as spectral and statistical information, is necessary when testing on complex real images.

6.2. Comparison with Other Methods

Four traditional segmentation techniques are selected for comparison, namely thresholding [35] and ACM [36] from the Matlab R2014b document



(a) Results of the Brodatz images.

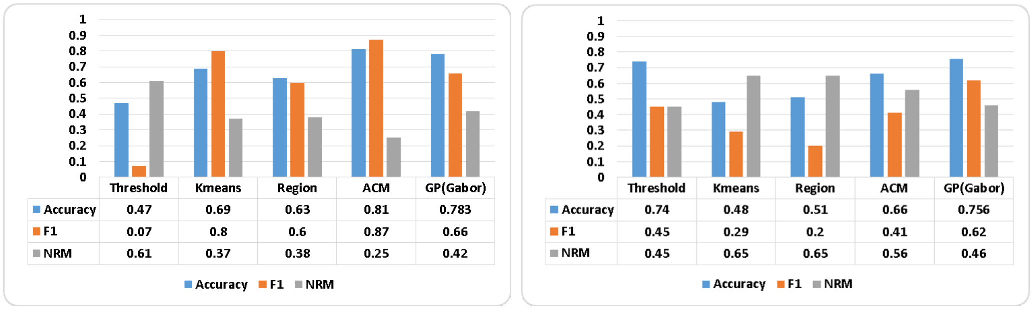
(b) Results of the Weizmann images.

(c) Results of the Pascal images.

Figure 5: Comparison between evolved segmentors on the Weizmann and Pascal images (the higher the better for the accuracy and F_1 score; vice versa for NRM).

examples, a region growing method from Kroon [37] and a K-means method from Fonseca [38]. As Gabor based segmentors achieve consistently good results on Weizmann and Pascal images, their statistical results are used for comparison with the four conventional segmentation techniques. Figure 6 displays the scores of the five methods on Weizmann and Pascal images in segmentation accuracy, F_1 and NRM.

On Weizmann images (Figure 6a), ACM performs best with the highest accuracy, highest F_1 score and lowest NRM score. It is well known that ACM relies heavily on the initial contour placement [39]. It is likely that ACM will produce accurate results when initial contours are set near objects of interest. GP(Gabor) and K-means clustering rank second with slightly worse results. According to Figure 6b, when testing on more complex Pascal images, GP(Gabor) produces the best results. K-means, region growing and ACM perform much worse than that of our GP method, and thresholding ranks second. However, the thresholding method is one of the simplest segmentation techniques as it only computes a global threshold to turn a test image into a binary result. Only when objects and background of an image have a high contrast is it likely that the thresholding method will obtain accurate segmentation results. In contrast, the evolved segmentors can produce satisfactory results consistently on both Weizmann and Pascal datasets.



	Threshold	Kmeans	Region	ACM	GP(Gabor)
■ Accuracy	0.47	0.69	0.63	0.81	0.783
■ F1	0.07	0.8	0.6	0.87	0.66
■ NRM	0.61	0.37	0.38	0.25	0.42

(a) Results of the Weizmann images.

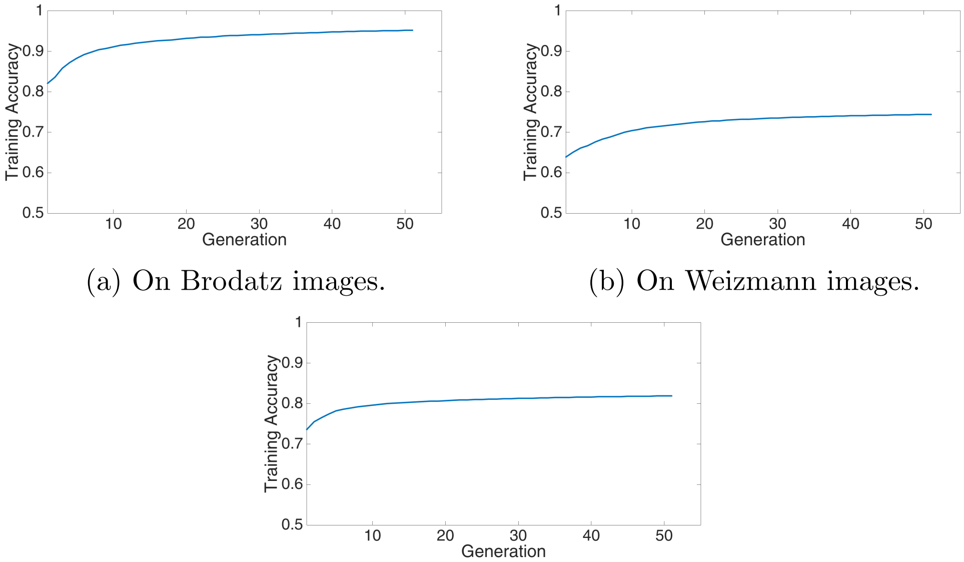
	Threshold	Kmeans	Region	ACM	GP(Gabor)
■ Accuracy	0.74	0.48	0.51	0.66	0.756
■ F1	0.45	0.29	0.2	0.41	0.62
■ NRM	0.45	0.65	0.65	0.56	0.46

(b) Results of the Pascal images.

Figure 6: Comparisons of different methods on Weizmann and Pascal images (the higher the better for the accuracy and F_1 score; vice versa for NRM).

To reveal how GP affects the segmentation performance, Figure 7 shows the training performance of the best segmentor evolved in each generation by GP. As described in section 2, the initial segmentors (segmentors in generation one) are generated randomly, and the segmentors in later generations (generation two, three and so forth) are evolved by GP based on the initial ones. In Figure 7, we can see that the training accuracy increases with the rise of the generation on all of the three datasets. Table 11 presents the statistical results of segmentors at generation one and 51. Based on the Mann-Whitney U-Test [40] at the significance level 5%, the training accuracy on all of the datasets rises significantly from generation one to generation 51. Meanwhile, the deviation of the accuracy drops. For example, on Brodatz dataset, the accuracy rises from 82.13% to 95.21% significantly, along with the decrease of the deviation from 2.46% to 0.97%. This reflects that GP can improve the initial segmentors to produce segmentors with better segmentation performance.

Based on the analyses, the GP based method can perform consistently well on the two complex datasets (the Weizmann and Pascal datasets), while some of the four traditional techniques show variable performance, and others are not effective to segment the real images. This means that the proposed method can outperform conventional commonly-used segmentation



(a) On Brodatz images.

(b) On Weizmann images.

(c) On Pascal images.

Figure 7: Training performance of the evolved segmentors using Gabor features (segmentors in generation one are created randomly).

Table 11: Statistical training performance of the evolved segmentors using Gabor features (solutions in generation one are created randomly; generation 51 is the final generation).

Dataset	Generation	Training Accuracy (%)
Brodatz	1	82.13 ± 2.46
	51	95.21 ± 0.97
Weizmann	1	63.89 ± 1.29
	51	74.41 ± 0.61
Pascal	1	73.65 ± 1.85
	51	81.90 ± 0.38

techniques. Moreover, GP has an important impact on the segmentation performance by producing better performing segmentors based on the randomly created initial segmentors.

6.3. Example Evolved Programs/Segmentors

To understand why the evolved programs can be used as good segmentors, in this part we display example programs (listed in Figure 8, Figure 9 and Figure 10) and take the one (Figure 8) evolved for the texture image

"D24vs34" (displayed in Table 4) as an example to explain why the evolved segmentors work well.

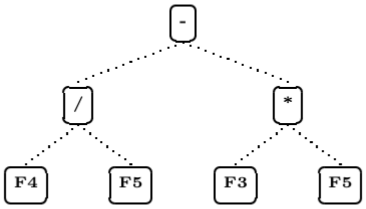


Figure 8: GP tree evolved from histogram statistics for texture image "D24vs34" ($F_i, i = 0, 1, 2, 3, 4, 5$).

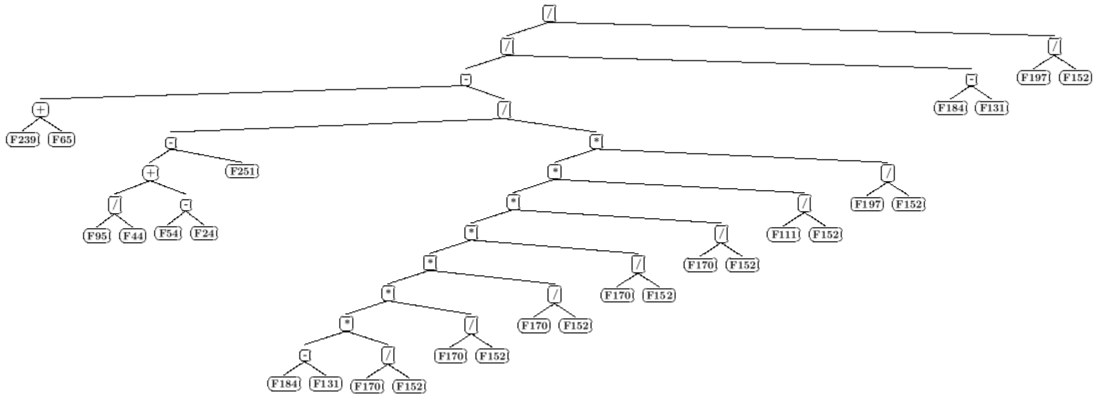


Figure 9: GP tree evolved from Fourier power spectrum for texture image "D24vs34" ($F_i, i = 0, 1, ..., 16$).

In Figure 8, the terminal set and the function set used are as follows. The terminal set is a six dimension feature set $F_i, i = 0, 1, 2, 3, 4, 5$, They correspond to six histogram statistics, i.e. mean (F_0), variance (F_1), skewness (F_2), kurtosis (F_3), energy (F_4), entropy (F_5) that are described in section 3.2. The function set is simple arithmetic and logic operators, i.e. add (+), subtract (-), multiply (*), divide (/), no bigger (<=), no smaller (>=), equal (==), if (IF) and between (Bwt), which are described in section 3.1.

The tree (shown in Figure 8) can be equivalently expressed in prefix notation, such as Lisp S-expression. The Lisp S-expression of this GP tree is shown in Equation 5, and we translate it into Equation 6. Equation 7 describes the classification strategy of GP.

$$(-(/\ F_4\ F_5)(*\ F_3\ F_5)) \tag{5}$$

$$GP\ Output = F_4/F_5 - F_3 * F_5 \tag{6}$$

$$Class = \begin{cases} A & \text{if } GP\ Output < 0 \\ B & \text{if } GP\ Output \geq 0 \end{cases} \tag{7}$$

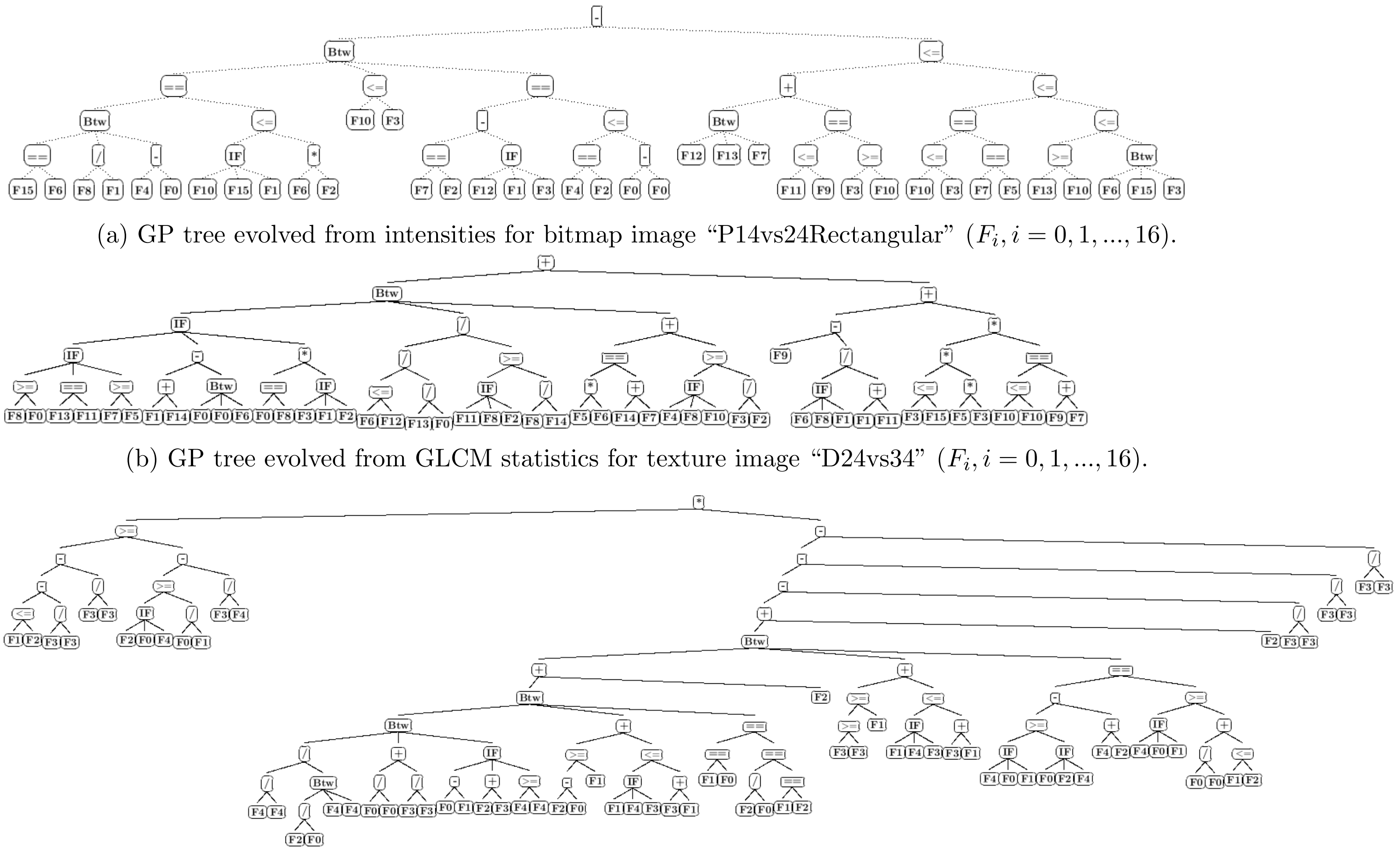
In our work, we set texture "D34" as class A and texture "D24" as class B arbitrarily (described in section 5). Based on Equation 6 and Equation 7, the texture class of a sub-region in one test image can be decided by Equation 8. From this equation, it can be seen that the evolved segmentor uses three features of a sub-region from a test image (F_3 (kurtosis), F_4 (energy) and F_5 (entropy)). Several simple arithmetic operations of these features produce the final class of this sub-region. Since texture "D34" should have lower energy, higher kurtosis and higher entropy than texture "D24", when comparing the two operations (F_4 and $F_3 * F_5^2$) in Equation 8, it is more likely for texture "D34" to obtain $F_4 < F_3 * F_5^2$; while texture "D34" to obtain $F_4 \geq F_3 * F_5^2$. In this way, "D34" and "D24" can be correctly categorized as class A and B respectively. Based on the class labels of all the sub-regions from a test image, the segmentation result of this image can be obtained.

$$Class = \begin{cases} A(D34) & \text{if}(F_4 < F_3 * F_5^2) \\ B(D24) & \text{if}(F_4 \geq F_3 * F_5^2) \end{cases} \tag{8}$$

7. Conclusions

This paper develops a supervised figure-ground segmentation system, which utilizes GP to evolve segmentors automatically. Seven feature descriptors, covering all three general features (brightness, texture and shape), are investigated as the system inputs to evolve segmentors. The evolved algorithms are tested on four images, including simple binary, synthetic texture images and complex images.

The results show that GP-evolved segmentors perform generally well on all the four datasets. This indicates that the proposed method can deal with diverse types of images. Compared with four traditional segmentation techniques, the GP-evolved segmentors achieve consistently good results. In addition, segmentors evolved from pixel intensities show promising performance on bitmap and texture images, yet they perform much worse than those evolved from histogram and Gabor features on complex images. It


-
Btw <=
== <= == + <=
Btw <= F10 F3 - <= Btw == == <=
== / - IF * == IF == - F12 F13 F7 <= >= <= == >= Btw
F15 F6 F8 F1 F4 F0 F10 F15 F1 F6 F2 F7 F2 F12 F1 F3 F4 F2 F0 F0 F11 F9 F3 F10 F10 F3 F7 F5 F13 F10 F6 F15 F3
+
Btw +
IF / + - *
IF - * / >= == >= F9 / * ==
>= == >= + Btw == IF <= / IF / * + IF / IF + <= * <= +
F8 F0 F13 F11 F7 F5 F1 F14 F0 F0 F6 F0 F8 F3 F1 F2 F6 F12 F13 F0 F11 F8 F2 F8 F14 F5 F6 F14 F7 F4 F8 F10 F3 F2 F6 F8 F1 F1 F11 F3 F15 F5 F3 F10 F10 F9 F7
*
>= -
- - - /
- / >= / - / F3 F3
<= / F3 F3 IF / F3 F4 + / F3 F3
F1 F2 F3 F3 F2 F0 F4 F0 F1 Btw F2 F3 F3
+ + ==
Btw F2 >= <= - >=
Btw + == >= F1 IF + >= + IF +
/ + IF >= <= == == F3 F3 F1 F4 F3 F3 F1 IF IF F4 F2 F4 F0 F1 / <=
/ Btw / / - + >= - F1 IF + F1 F0 / == F4 F0 F1 F0 F2 F4 F0 F0 F1 F2
F4 F4 / F4 F4 F0 F0 F3 F3 F0 F1 F2 F3 F4 F4 F2 F0 F1 F4 F3 F3 F1 F2 F0 F1 F2
F2 F0

(a) GP tree evolved from intensities for bitmap image “P14vs24Rectangular” ($F_i, i = 0, 1, ..., 16$).

(b) GP tree evolved from GLCM statistics for texture image “D24vs34” ($F_i, i = 0, 1, ..., 16$).

(c) GP tree evolved from moments&gradient statistics for texture image “D24vs34” ($F_i, i = 0, 1, 2, 3, 4$).

Figure 10: The best individual of one run.

reflects that higher-level of features, such as those containing intensity statistical and spectral information, are necessary for segmentation tasks of complex images, which are difficult to segment.

One problem is that methods using evolutionary computation techniques, especially GP, normally evolve complex solutions. This can be reflected by the solution in Figure 10c, which is evolved from 5 features, yet has a size of over 100 (the number of nodes). Therefore, the solution complexity control of GP based methods will be studied in our future work.

Reference

[1] W. Zou, C. Bai, K. Kpalma, J. Ronsin, Online glocal transfer for automatic figure-ground segmentation, Image Processing, IEEE Transactions on 23 (5) (2014) 2109–2121.

[2] E. Borenstein, S. Ullman, Learning to segment, in: Computer Vision-ECCV 2004, Springer, 2004, pp. 315–328.

[3] E. Sharon, A. Brandt, R. Basri, Segmentation and boundary detection using multiscale intensity measurements, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Vol. 1, IEEE, 2001, pp. I–469.

[4] J. Liu, J. Wang, Application of snake model in medical image segmentation, J. Convergence Inf. Technol 9 (1) (2014) 105–109.

[5] C.-Y. Liu, J. E. Iglesias, Z. Tu, A. D. N. Initiative, et al., Deformable templates guided discriminative models for robust 3d brain mri segmentation, Neuroinformatics 11 (4) (2013) 447–468.

[6] G. Gill, M. Toews, R. R. Beichel, Robust initialization of active shape models for lung segmentation in ct scans: A feature-based atlas approach, Journal of Biomedical Imaging 2014 (2014) 13.

[7] E. Borenstein, S. Ullman, Class-specific, top-down segmentation, in: Computer VisionECCV 2002, Springer, 2002, pp. 109–122.

[8] E. Borenstein, S. Ullman, Combined top-down/bottom-up segmentation, Pattern Analysis and Machine Intelligence, IEEE Transactions on 30 (12) (2008) 2109–2125.

[9] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International journal of computer vision 1 (4) (1988) 321–331.

[10] R. Poli, W. B. Langdon, N.F.M., A Field Guide to Genetic Programming, Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, UK, 2008.

[11] R. Poli, Genetic programming for image analysis, in: Proceedings of the 1st annual conference on genetic programming, MIT Press, 1996, pp. 363–368.

[12] A. Song, V. Ciesielski, Fast texture segmentation using genetic programming, in: Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, Vol. 3, IEEE, 2003, pp. 2126–2133.

[13] A. Song, V. Ciesielski, Texture segmentation by genetic programming, Evolutionary Computation 16 (4) (2008) 461–481.

[14] T. Singh, N. Kharma, M. Daoud, R. Ward, Genetic programming based image segmentation with applications to biomedical object detection, in: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM, 2009, pp. 1123–1130.

[15] M. E. Roberts, The effectiveness of cost based subtree caching mechanisms in typed genetic programming for image segmentation, in: Applications of Evolutionary Computing, Springer, 2003, pp. 444–454.

[16] M. Dong, M. G. Eramian, S. A. Ludwig, R. A. Pierson, Automatic detection and segmentation of bovine corpora lutea in ultrasonographic ovarian images using genetic programming and rotation invariant local binary patterns, Medical & biological engineering & computing 51 (4) (2013) 405–416.

[17] J. Geng, J. Liu, Image texture classification using a multiagent genetic clustering algorithm, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 504–508.

[18] J. R. Koza, Genetic programming: on the programming of computers by means of natural selection, Vol. 1, MIT press, 1992.

[19] Y. Liang, M. Zhang, W. N. Browne, A supervised figure-ground segmentation method using genetic programming, in: Applications of Evolutionary Computation, Springer, 2015, pp. 491–503.

[20] W. Banzhaf, P. Nordin, R. E. Keller, F. D. Francone, Genetic Programming: An Introduction, New York: Morgan Kaufmann, 1998.

[21] W. Khan, Image segmentation techniques: A survey, Journal of Image and Graphics 1 (4) (2013) 166–170.

[22] N. Senthilkumaran, R. Rajesh, Edge detection techniques for image segmentation–a survey of soft computing approaches, International journal of recent trends in engineering 1 (2).

[23] A. Eleyan, H. Demirel, Co-occurrence matrix and its statistical features as a new approach for face recognition, Turk J Elec Eng & Comp Sci 19 (1) (2011) 97–107.

[24] M. Pietikäinen, A. Hadid, G. Zhao, T. Ahonen, Local binary patterns for still images, in: Computer Vision Using Local Binary Patterns, Springer, 2011, pp. 13–47.

[25] J. Ilonen, T. Eerola, H. Mutikainen, L. Lensu, J. Käyhkö, H. Kälviäinen, Estimation of bubble size distribution based on power spectrum, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer, 2014, pp. 38–45.

[26] J. Kamarainen, Gabor features in image analysis, in: Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on, IEEE, 2012, pp. 13–14.

[27] H. Shu, L. Luo, J.-L. Coatrieux, Moment-based approaches in imaging. part 1, basic features, IEEE Engineering in Medicine and Biology Magazine 26 (5) (2007) 70.

[28] B. Freeman, F. Durand, Gradient image processing, Available at `http://groups.csail.mit.edu/graphics/classes/CompPhoto06/html/lecturenotes/10_Gradient.pdf`.

[29] Brodatz texture database, Available at `http://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html`.

[30] E. Borenstein, Weizmann horse database, Available at http://www.msri.org/people/members/eranb/.

[31] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2) (2010) 303–338.

[32] D. M. Powers, Evaluation: From precision, recall and f-factor to roc, Tech. rep., Australia (2007).

[33] J. Ashburner, K. J. Friston, Unified segmentation, Neuroimage 26 (3) (2005) 839–851.

[34] M. Kuhn, Futility analysis in the cross-validation of machine learning models, arXiv preprint arXiv:1405.6974.

[35] Thresholding segmentation, Available at http://au.mathworks.com/help/images/examples/correcting-nonuniform-illumination.html.

[36] Active contour based segmentation, Available at http://au.mathworks.com/help/images/ref/activecontour.html#btuep4x-7.

[37] D. Kroon, Region growing, Available at http://www.mathworks.com/matlabcentral/fileexchange/19084-region-growing2008.

[38] P. Fonseca, K-means image segmentation, Available at http://www.mathworks.com/matlabcentral/fileexchange/authors/129300.

[39] T. Liu, H. Xu, W. Jin, Z. Liu, Y. Zhao, W. Tian, Medical image segmentation based on a hybrid region-based active contour model, Computational and mathematical methods in medicine 2014.

[40] P. E. McKnight, J. Najab, Mann-Whitney U test, Corsini Encyclopedia of Psychology.