

An Automatic Region Detection and Processing Approach in Genetic Programming for Binary Image Classification

Ying Bi, Mengjie Zhang and Bing Xue

Evolutionary Computation Research Group,

Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand

Email: {ying.bi, mengjie.zhang, bing.xue}@ecs.vuw.ac.nz

Abstract—In image classification, region detection is an effective approach to reducing the dimensionality of the image data but requires human intervention. Genetic Programming (GP) as an evolutionary computation technique can automatically identify important regions, and conduct feature extraction, feature construction and classification simultaneously. In this paper, an automatic region detection and processing approach in GP (GP-RDP) method is proposed for image classification. This approach is able to evolve important image operators to deal with detected regions for facilitating feature extraction and construction. To evaluate the performance of the proposed method, five recent GP methods and seven non-GP methods based on three types of image features are used for comparison on four image data sets. The results reveal that the proposed method can achieve comparable performance on easy data sets and significantly better performance on difficult data sets than the other comparable methods. To further demonstrate the interpretability and understandability of the proposed method, two evolved programs are analysed. The analysis shows the good interpretability of the GP-RDP method and proves that the GP-RDP method is able to identify prominent regions, evolve effective image operators to process these regions, extract and construct good features for efficient image classification.

Keywords—Image Classification; Region Detection; Region Processing; Genetic Programming; Feature Extraction; Feature Construction

I. INTRODUCTION

As an important task in computer vision and image processing, image classification has received much attention over the world in recent decades due to its wide range of applications in various fields such as face image, medical image and remote sensing image. The task can be described as categorising a set of images into different predefined groups. Although various methods have been proposed to deal with this task [1, 2], image classification is still an open issue [3], which needs further investigation.

One challenge in this field is the high dimensionality of the image data and feature space [4]. To overcome this and improve classification accuracy, finding the regions of interest (ROI) and extracting features from these regions have been proposed as an effective approach. For instance, Lazebnik

et al. [5] partition images into sub-regions and extract local histogram features from each region. The proposed method achieves significant performance improvement on recognition of natural scene categories. Bosch *et al.* [6] propose an approach to automatically detecting ROIs using the similarity measure based on a concatenating descriptor. The results show that selecting ROIs can improve about 5% classification accuracy over the state of the art on the Caltech-256 data set. Cheng *et al.* [7] apply colour and texture features in histogram to obtain homogeneous regions and the proposed method achieves good image classification performance. However, these methods [6, 7] treat ROIs identifying as region matching, which not only requires high-level domain knowledge to design good dissimilarity measures and descriptors but also demands a set of identified candidate regions.

Genetic Programming [8] as a member of Evolutionary Computation (EC) techniques, is able to achieve automatic region detection without domain knowledge and predefined candidate regions [9, 10]. GP simulates evolutionary strategies and the "survival of the fittest" principle. It is a population-based search technique where a number of computer programs consist of the population. GP starts with random initialisation of a population in the search space. Each program is evaluated by a fitness function and allocated the fitness value such as the classification accuracy. The population is updated during evolutionary process by using the selection method and genetic operators i.e. crossover, mutation and reproduction. The processes of fitness evaluation and population updating proceed until reach a predefined termination criterion. Finally, the best program/solution is founded and returned.

Recently, several GP variants are proposed to deal with region detection, feature extraction, feature construction and image classification simultaneously. In [11], a three tier GP (3TGP) approach is introduced by integrating the image filtering tier, aggregation tier and classification tier. Each tier in 3TGP can perform different tasks. The aggregation functions in the aggregation tier are employed for detecting square regions from an image and extracting domain independent features (pixel statistic) such as *mean*, *max*, *min*, *median*, and *standard deviation* from the detected regions. However,

in this approach, very simple image filters include *max*, *min*, *mean*, and *median* are evolved to deal with the input raw image and only pixel statistics are extracted from the detected regions. Later, this approach is improved as two-tier GP (2TGP), 2TGP-mix and 2TGP-line by simplifying 3TGP's representation [10] and detecting more flexible shapes of regions including *line*, *square*, *rectangle*, and *circle* [12]. Similar to the 3TGP approach, pixel statistics are extracted from the detected regions without any processing. However, sometimes the detected regions might contain noise, have poor contrast or contain much other meaningful information such as edges. Only extracting simple pixel statistics are typically not enough and accurate in these cases for tackling with difficult image classification tasks. An HOG+GP approach is proposed by Lenson *et al.* [13] for binary image classification. In this approach, region detection is embedded and the HOG descriptor [14] is utilised for extracting features from the detected region. However, the HOG descriptor is originally designed for pedestrian detection, which might be not efficient for other image classification tasks such as scene classification.

As GP has a big potential to perform region detection, feature extraction, feature construction, and image classification simultaneously, this paper will continue to investigate it in order to find better high-level features for dealing with different image classification tasks. Instead of extracting pixel statistics or HOG features from the detected region directly, a new set of operators are employed to allow GP evolve better image-related operators to deal with the detected regions before feature extraction. These operators include histogram equalisation operator, smooth filter, edge detector, shape detector, and texture descriptor. These operators are able to deal with the detected region to generate better features to facilitate feature extraction and construction.

Goals. The overall goal of this paper is to develop a GP-based approach for automatic and simultaneous region detection, region processing, feature extraction, feature construction, and image classification. In order to achieve this goal, a novel GP program representation is designed, and new operators and terminals are developed. The new method will be examined and compared with five other effective GP methods and seven non-GP methods on four image data sets with different difficulties. Specifically, the goal can be divided into the following four objectives. (1) Develop a new GP program structure which can integrate regions detection, region processing, feature extraction, feature construction and classification to a single tree/solution; (2) Develop a new function set and a new terminal set which allow the GP system to find better solutions effectively and efficiently; (3) Investigate whether the proposed GP-based method can achieve better performance than the other five GP methods and seven non-GP methods; and (4) Investigate whether some important regions in an image can be detected, what image operators can be evolved to deal with these regions, and how features from these regions can be extracted and constructed for effective classification.

II. PROPOSED METHOD

This section describes the proposed GP-RDP method in detail, including the overall process, the program structure, the function set, the terminal set, and the fitness function.

A. Overall Process and Program Structure

The overall process for image classification is shown in Fig. 1. First of all, raw images are resized and normalised in the preprocessing process. Then each image is fed to the GP system for region detection, region processing, feature extraction, and feature construction. Finally, a class label is assigned to the input image.

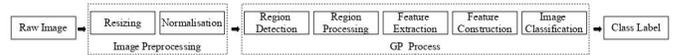


Fig. 1. Overall process for image classification

The tree-based representation is commonly used in GP, as an example tree shown in Fig. 2. The GP tree compounded by several different internal nodes and leaf nodes. These internal nodes consist of operators such as arithmetic operators (e.g. *Add (+)*) and image-related operators (e.g. *Gau1*, *Sobel_X*, *Histogram*), and the leaf nodes are made up by terminals (e.g. *Image*, *X*, *Y*, *Size*) defined according to the problem or the requirements from the operators. The proposed GP-RDP method is based on *strongly typed* GP [15] where restrictions are imposed on the input and output data types of nodes. By this means, different types of nodes can be designed for different purposes. As listed at the left side of the Fig. 2, different combinations of operators and terminals can deal with different tasks, including region detection, region processing, feature extraction, feature construction and classification.

In the proposed GP-RDP method, the overall structure is constructed in a bottom-up manner. The image is fed from the input tier. Then several important regions will be detected automatically from the input image, and image-related operators will be evolved to deal with these regions. Important features will be extracted from these processed regions and constructed for classification. The final output of the proposed method is a class label for the input image.

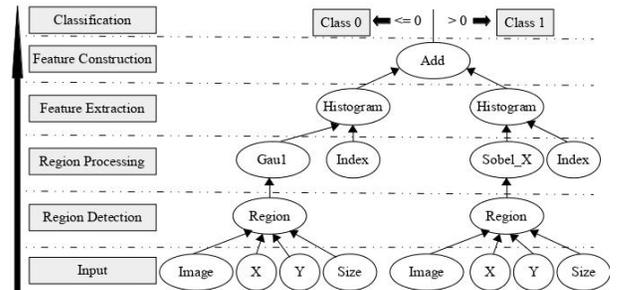


Fig. 2. An example of the program structure

B. Function Set

In our proposed method, according to the program representation, four types of operators are employed to construct the function set. The first type has one operator called **Region**, which is designed for region detection. The *Region* node takes an image, coordinates of the top left point of a region and the size of a region as input, and returns a region of interest. The shape of a detected region in the proposed GP-RDP approach is defined as square. The size of a detected region is set as 30×30 by assuming that it is enough for contain important information from an image, but not too big to increase the computational cost.

The second type of operators is employed for region processing. It has one image operator, eight image filters and two image descriptors. The description of each operator are listed in Table I. The **Hist_Eq** operator is designed to increase contrast and equalize the histogram of the input image. This operator is very important as all images must be fed to the *Histogram* node to calculate the histogram. Image filters involve conducting convolution operations, including **Gau1**, **Gau11**, **GauXY**, **Lap**, **Sobel_X**, **Sobel_Y**, **LoG1**, **LoG2** operators. The **Gau1** operator is used for reducing noise, and the other seven filters are employed for edge detection, shape detection or blob detection. The size of all these filters are set to 3×3 , which is commonly used in image processing. Two famous image descriptors **LBP** [16] and **HOG** [14] are utilised to describe important shape and texture information from the detected region. In the **LBP** operator, the number neighbours is set to 8 and the radius is set to 1.5. In the **HOG** operator, the number of orientations is 9, the block size is 3×3 and the cell size is 8×8 . All the operators take a region of an image as input and return a region. It should be pointed out that all these operators and the orders among these operators are automatically selected during evolutionary process. Hence, these operators allow GP to find better combinations of them to deal with the detected region in order to facilitate feature extraction and construction.

TABLE I
THE SECOND TYPE OF OPERATORS IN THE FUNCTION SET

Functions	Function Description
Hist_Eq	Histogram Equalisation
Gau1	Gaussian filter with $\sigma = 1$
Gau11	The first derivatives of Gaussian filter with $\sigma = 1$
GauXY	Gradient magnitude using Gaussian derivatives with $\sigma = 1$
Lap	Laplacian filter
Sobel_X	Sobel filter along X axis
Sobel_Y	Sobel filter along Y axis
LoG1	Laplacian of Gaussian filter with $\sigma = 1$
LoG2	Laplacian of Gaussian filter with $\sigma = 2$
LBP	Uniform LBP descriptor
HOG	HOG descriptor (return an image)

The third type of operator contains a designed **Histogram** operator, which is used for feature extraction. This operator takes a region and an integer (*Index*) as inputs, and returns an integer. By this *Histogram* operator, the current distribution of all pixel values in the input region is drawn and the value of

Index bin is returned. An important histogram feature can be extracted by this operator. In our paper, we set the number of total bins to 10 due to the reason that setting it too large will increase the search space while setting it too small can not effectively distinguish the distribution difference of regions from two different classes.

The final type of operators for feature construction is made up of five commonly used arithmetic functions and one logical operator. They are **Add(+)**, **Sub(-)**, **Mul(\times)**, **Neg**, **IF** and **Div(/)**. The *Div* is protected by returning zero if the divisor is zero. The *IF* operator takes three arguments as input and return the second argument if the first argument is larger than 0, otherwise return the third argument.

C. Terminal Set

Terminals can be chosen to form the leaf nodes of a tree. In the proposed GP-RDP method, there are three types of terminals in the terminal set which are child nodes of the *Region* and *Histogram* operators. The *Region* operator needs four terminals (leaf nodes), including the most important terminal called **Image** which represents the input image, the *X* and *Y* terminals which mean the top left coordinates of the detected region, and the **Size** terminal which is the width and height of the detected region. The *X* and *Y* terminals are integer randomly generated from $[0, (image\ width)-30]$ and $[0, (image\ height)-30]$. The *Histogram* operator requires a leaf node called **Index**, which is an integer generated from a range of $[1, 10]$ as the total number of bins is set as 10.

D. Fitness Function

The fitness measure of the proposed method is maximising classification accuracy. The formulation of fitness function is Equation (1) given as follow.

$$f = \frac{\text{number of correctly classified images}}{\text{total number of images}} \times 100\% \quad (1)$$

III. EXPERIMENT DESIGN

This section provides the descriptions of image data sets, baseline methods and parameter settings.

A. Data Sets

Four different types of image data sets i.e. **COIL-20** [17], **JAFFE** [18], **SCENE** [19], **UIUC** [20] are selected in experiments to test the performance of the proposed method. All the images are grey-scale images. More details about the four data sets are listed in Table II, and some example images from the corresponding data set are shown in Fig. 3. All the data sets represent different image classification tasks, i.e. the COIL-20 is object classification, the JAFFE is facial expression categorisation, the SCENE is scene classification, and the UIUC is a car detection task. The difficulties of classification on each data set are various due to noise, object orientation, illumination, and clutter background in images.

TABLE II
DATA SET PROPERTIES

Name	Size	Classes	Training Set	Validation Set	Test Set
COIL-20	128×128	<i>Ducks</i>	37	14	21
		<i>Cars</i>	37	14	21
JAFPE	128×128	<i>Happy</i>	10	10	10
		<i>Surprised</i>	10	10	10
SCENE	128×128	<i>Highway</i>	130	65	65
		<i>Streets</i>	146	73	73
UIUC	100×40	<i>Cars</i>	275	110	165
		<i>Non-cars</i>	250	100	150



Fig. 3. Example images from the corresponding data set

B. Baseline Methods

To demonstrate the performance of the proposed method, five other GP methods and seven non-GP based methods are implemented for comparison.

GP methods include *3TGP* [11], *2TGP* [10], *FeEx+GP* [21], *Hist+GP*, *LBP+GP* [22]. The first two methods use raw images as inputs. The last three methods use extracted features as inputs. In *Hist+GP*, 64 histogram features (*Hist*) of an image are extracted so that each four adjacent pixel values can be allocated to the same bin. The *Hist* features are fed to the GP method for evolving a classifier. Similar to *Hist+GP*, *LBP+GP* takes the 59 uniform LBP histogram features as the input of the GP for image classification.

non-GP methods include seven commonly used classification methods based on three types of extracted features. These features are *FeEx*, *LBP*, *Hist* features extracted in the same ways as that used in the *FeEx+GP*, *Hist+GP*, *LBP+GP* methods. Seven classification methods include Nearest Neighbour (1NN), Naïve Bayes (NB), Decision Tree (DT), Multilayer Perception (MLP), Adaptive Boosting (AdaBoost), Random Forest (RF), and Support Vector Machine (SVM). The implementation of these methods are based on the well-known *scikit-learn* [23] Python package for machine learning with the default settings. For these methods, the training instances include the training set and validation set used in the GP approaches, while the test set keeps the same.

C. Parameter Setting

In order to achieve a fair comparison, we implement all the GP methods according to the related papers by Python based on the *DEAP* (*Distributed Evolutionary Algorithm in Python*) [24] package. Parameter settings in all the GP methods are the same as listed in Table III. Each experiment runs 30 times independently with different random seeds.

TABLE III
GP RUN TIME PARAMETERS

Parameter	Value	Parameter	Value
Generations	50	Crossover rate	0.8
Population size	1024	Mutation rate	0.19
Selection type	Tournament	Elitism rate	0.01
Tournament size	7	Tree-depth	2-6

In the evolutionary process, each individual is evaluated at each generation on the training set. To avoid overfitting, the best individual on the training set is evaluated on the validation set. After 50 generations, the best individual on the validation set is tested on the test set to evaluate the performance of the method. Notice that this evaluation process is conducted in all the GP methods rather than only in the proposed method.

IV. RESULTS AND DISCUSSIONS

This section compares and discusses experimental results obtained by the proposed method, the other five GP methods and the seven non-GP methods. Further analysis on the evolved programs is also presented.

A. Classification performance

All the results in terms of maximum, average classification accuracy and standard deviation on the training and test sets obtained by the six GP methods are presented in Table IV. The classification accuracy on the test data set gained by the seven non-GP methods using three different types of features is shown in Table V. Student’s *t*-test with a 95% confidence interval is utilised to evaluate the significance of performance improvement of the proposed method compared to a GP or non-GP method. As shown in Table IV and Table V, the “+” and “-” means our proposed method is significantly better or worse than the compared method.

Compared to the other five GP methods, the proposed GP-RDP outperform these methods significantly in 10 cases out of 20 cases and is significantly worse than them in 3 cases out of 20 cases on the training sets of the four data sets. The proposed GP-STD method gains significantly better results in 13 cases and only significantly worse in 1 case out of 20 cases on all the test data sets. Compared to the other seven classification methods, the proposed GP-RDP method obtains significantly better results in 5 cases out of 21 cases on the COIL-20 data set. Note that the methods that achieve significantly better results in some cases are mainly AdaBoost and Random Forest, which are boosting and ensemble classifiers while the proposed GP-RDP method only uses a single evolved program classifier. The comparison is actually not entirely fair for the proposed method. On the remain three data sets, the proposed GP-RDP method achieve significantly better results in 19 cases on the JAFPE data set, 11 cases on the SCENE data set and 18 cases on the UIUC data set out of the total 21 cases. In summary, our proposed GP-RDP method achieves comparable performance on easy data sets and better performance on difficult data sets than that the other baseline methods.

TABLE IV
TRAINING AND TESTING RESULTS ON THE FOUR DATA SETS

Algorithms	Training (%)		Testing (%)	
	Max	Mean±St.D.	Max	Mean±St.D.
COIL-20				
GP-RDP	100.0	100.0±0.0	100.0	99.13±1.30
2TGP	100.0	100.0±0.0	100.0	98.25±1.84 ⁺
3TGP	100.0	100.0±0.0	100.0	98.41±1.77
FeEx+GP	100.0	100.0±0.0	100.0	98.41±1.55
Hist+GP	100.0	100.0±0.0	100.0	99.44±1.59
LBP+GP	100.0	100.0±0.0	100.0	98.89±1.47
JAFFE				
GP-RDP	100.0	95.33±4.27	95.00	85.83±6.34
2TGP	100.0	92.83±4.41 ⁺	100.0	83.33±8.88
3TGP	100.0	93.67±4.07	100.0	84.67±7.52
FeEx+GP	100.0	92.50±4.61 ⁺	90.00	71.17±11.16 ⁺
Hist+GP	100.0	98.50±2.63 ⁻	75.00	49.50±11.72 ⁺
LBP+GP	100.0	98.17±2.73 ⁻	65.00	48.50±9.32 ⁻
SCENE				
GP-RDP	93.84	92.32±0.79	92.03	85.19±2.68
2TGP	92.39	84.13±4.35 ⁺	84.78	75.72±3.90 ⁺
3TGP	93.12	90.99±1.36 ⁺	86.96	82.71±2.34 ⁺
FeEx+GP	89.49	86.46±2.39 ⁺	85.51	79.44±3.45 ⁺
Hist+GP	90.22	88.24±1.27 ⁺	86.23	81.50±2.30 ⁺
LBP+GP	97.46	94.57±1.22 ⁻	96.38	92.61±1.99 ⁻
UIUC				
GP-RDP	95.43	91.31±3.38	93.02	88.66±2.99
2TGP	94.29	83.82±3.86 ⁺	87.30	80.76±3.25 ⁺
3TGP	96.76	87.65±5.02 ⁺	92.38	84.26±3.77 ⁺
FeEx+GP	88.00	85.13±1.75 ⁺	85.40	81.22±2.32 ⁺
Hist+GP	69.90	67.54±1.14 ⁺	62.22	56.94±2.18 ⁺
LBP+GP	92.95	90.06±1.40	87.62	84.99±1.71 ⁺

TABLE V
ACCURACY(%) OF SEVEN CLASSIFIERS USING THREE DIFFERENT FEATURES ON THE FOUR TEST SETS

Features	1NN	NB	DT	MLP	AdaBoost	RF	SVM
COIL-20							
FeEx	100.0 ⁻	100.0 ⁻	100.0 ⁻	97.62 ⁺	100.0 ⁻	100.0 ⁻	97.62 ⁺
Hist	95.24 ⁺	100.0 ⁻	97.62 ⁺	100.0 ⁻	100.0 ⁻	100.0 ⁻	100.0 ⁻
LBP	95.24 ⁺	100.0 ⁻					
JAFFE							
FeEx	75.0 ⁺	60.0 ⁺	65.0 ⁺	50.0 ⁺	90.0 ⁻	65.0 ⁺	75.0 ⁺
Hist	85.0	75.0 ⁺	60.0 ⁺	40.0 ⁺	80.0 ⁺	60.0 ⁺	75.0 ⁺
LBP	60.0 ⁺	65.0 ⁺	55.0 ⁺	50.0 ⁺	55.0 ⁺	60.0 ⁺	65.0 ⁺
SCENE							
FeEx	89.13 ⁻	85.51	77.54 ⁺	52.90 ⁺	86.23 ⁻	84.78	52.90 ⁺
Hist	80.43 ⁺	77.54 ⁺	76.09 ⁺	82.61 ⁺	86.96 ⁻	84.06 ⁺	70.29 ⁺
LBP	83.33 ⁺	91.30 ⁻	89.86 ⁻	64.49 ⁺	94.93 ⁻	89.86 ⁻	90.58 ⁻
UIUC							
FeEx	80.63 ⁺	87.30 ⁺	81.90 ⁺	52.38 ⁺	87.62	85.08 ⁺	52.38 ⁺
Hist	67.3 ⁺	61.27 ⁺	64.44 ⁺	64.13 ⁺	67.62 ⁺	60.32 ⁺	56.83 ⁺
LBP	86.67 ⁺	90.48 ⁻	86.03 ⁺	85.08 ⁺	90.79 ⁻	86.35 ⁺	86.03 ⁺

B. Analysis on Evolved Program

To further demonstrate the good interpretability and understandability of the proposed method, two example programs are analysed. Fig. 4 and Fig. 6 show the evolved programs and the output of each node when using two example images from different classes as input. The two input images and the two images with detected regions are shown in the two figures as well. In the Fig. 4 and Fig. 6, the number above the Histogram chart represents the extracted histogram feature

value (the value of the *Index*th bin).

1) *Example on COIL-20*: Fig. 4 shows an example program evolved by the GP-RDP method on the COIL-20 data set. This program obtains 100% accuracy on the training, validation and test sets. By this program, two important regions are detected as shown in Fig. 5. The first detected region (Region 1) captures very important information as the Fig. 5 shown that in most cases the majority pixel values in this region from *Ducks* object images are very big while they are very small in the region from *Cars* object images. The difference of the Region 1 between the *Ducks* and *Cars* classes can be easily distinguished so that there are not other image operators evolved. The detected Region 2 captures margin information of the object in an image, which does not show a very obvious difference of pixel value distribution. Thus, two *HOG* descriptors, a *Sobel_X* filter and a *LoG* filter are evolved to deal with the detected Region 2 to generate better features, which can be seen from Fig. 4. The *Sub* function is evolved as the root node for feature construction in this program.

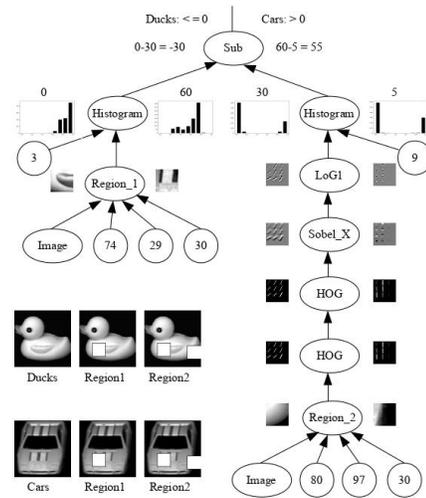


Fig. 4. An evolved program by GP-RDP method on the COIL-20 data set

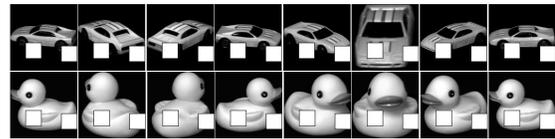


Fig. 5. Example images with two detected regions by the evolved program in Fig. 4

2) *Example on JAFFE*: An example program evolved by GP-RDP on the JAFFE data set is shown in Fig. 6. This program achieves 90% accuracy on the training set, 90% accuracy on the validation set and 95% accuracy on the test set. By this program, two important regions are detected. The detected region 1 captures prominent differences of the teeth and the nose area between two classes. A *LoG* operator is evolved to deal with this region, which further enhances the difference by finding the flat area and the area where has

