

# Genetic Programming for Automatic Global and Local Feature Extraction to 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**—Feature extraction is an essential process to image classification. Existing feature extraction methods can extract important and discriminative image features but often require domain expert and human intervention. Genetic Programming (GP) can automatically extract features which are more adaptive to different image classification tasks. However, the majority GP-based methods only extract relatively simple features of one type i.e. local or global, which are not effective and efficient for complex image classification. In this paper, a new GP method (GP-GLF) is proposed to achieve automatically and simultaneously global and local feature extraction to image classification. To extract discriminative image features, several effective and well-known feature extraction methods, such as HOG, SIFT and LBP, are employed as GP functions in global and local scenarios. A novel program structure is developed to allow GP-GLF to evolve descriptors that can synthesise feature vectors from the input image and the automatically detected regions using these functions. The performance of the proposed method is evaluated on four different image classification data sets of varying difficulty and compared with seven GP based methods and a set of non-GP methods. Experimental results show that the proposed method achieves significantly better or similar performance than almost all the peer methods. Further analysis on the evolved programs shows the good interpretability of the GP-GLF method.

## I. INTRODUCTION

Image classification as a fundamental task in computer vision and pattern recognition, has obtained much attention over the world in recent decades. It has been applied to a wide range of applications in various fields such as biology images [1], face images [2], remote sense images [3], and medical images [4]. Image classification is to categorise images into predefined groups based on the content in the images. It is a very challenging task due to the image variations and the high dimensionality of the image data.

Feature extraction is one of the most important steps in an image classification system [5, 6]. Image feature extraction is to extract important and representative information from an image that is helpful for describing the image data and reducing the dimensionality of the image data. Generally, two types of features, i.e. global features and local features can be extracted to describe an image [5]. Global features describe a whole image based on all the pixel values, which can be considered as a particular property of the image, such as shape, texture and colour [5]. Local features aim at detecting and describing regions of interest/keypoints from an image [5].

A number of well-known and effective methods have been developed to extract global or local features from an image or keypoints. These methods include Domain Independent Feature (DIF) [7], Histogram (Hist) [5], Grey-Level Co-occurrence Matrix (GLCM) [8, 9], Histogram of Orientated Gradients (HOG) [10], Scale Invariant Feature Transform (SIFT) [11], Local Binary Patterns (LBP) [12, 13], and Speeded Up Robust Features (SURF) [14].

Most existing feature extraction methods are originally designed for specific image tasks [5], such as GLCM and LBP for texture description, SURF and SIFT are for keypoints detection and description. Generally, domain experts are required to determine which method should be used and how to set parameters for the selected method when dealing with a new task [15]. However, these experts are not always at hand which needs time to find and cost to employ [15].

Genetic Programming (GP) [16], as an Evolutionary Computation (EC) technique, aims at automatically evolving computer programs that are able to solve problems without human intervention and domain knowledge. Among all the EC techniques, GP is the widely used technique in image analysis [17]. Compared with other EC techniques, the representation of tree-based GP is more flexible, which results in great potential on image-related tasks [18–21].

GP has been applied to automatic feature extraction and image classification based on raw images [6, 18, 22–29]. In [18, 25–27], GP is employed to region detection, feature extraction, feature construction, and image classification. These methods capture general local information/features, i.e. pixel statistics, from images. The results show that the local features perform especially well on images that have objects or faces with relatively fixed positions. However, only one simple feature extracted from each detected region is not effective for complex image classification. In [6, 22–24], feature extraction is based on a whole image, where a feature vector is evolved for classification. However, the methods in [6, 23] are originally designed for texture description, which might not be suitable for other types of image classification tasks.

Existing work shows that both global and local features are important for image classification [30, 31]. Global features are effective for finding similar images in the same class while local features are more important for capturing distinctive information from objects or partial objects in an image.

However, the work of GP on feature extraction only focus on extracting one type of image features, i.e. local features [18, 25–29] or global features [6, 22–24], which are often not effective and efficient for complex image classification. In addition, existing image descriptors such as HOG, SIFT and LBP are the well-known effective methods for feature extraction, but they are seldom cooperated with GP to automatically extract more adaptive and informative features for image classification.

### A. Goals

The goal of this paper is to design a new GP method (GP-GLF) that is able to automatically and simultaneously extract informative and adaptive local features and global features using a set of effective feature extraction operators. To achieve this goal, a set of well-known feature extraction methods will be used as GP functions in the function set to cope with global and local scenarios. A novel program structure will be designed to integrate these functions into a single solution/tree and to allow GP to extract rich global and local features. To evaluate the performance of the extracted features by the proposed GP-GLF method, four image classification data sets of varying difficulty will be used, and the state-of-the-art GP methods and non-GP methods will be employed as peer competitors. To achieve this, the overall goal leads to the following objectives to be investigated in this paper.

- 1) Design a new program structure that allows GP to extract global and local features and to synthesise these features as a feature vector with flexible length;
- 2) Develop a new function set and a new terminal set, where region detectors and the state-of-the-art feature extraction operators can be employed for extracting informative features with potential of achieving good classification performance;
- 3) Evaluate and compare the performance of the proposed method with the existing GP methods and a set of non-GP methods; and
- 4) Analyse example programs/trees with the best performance to understand how the proposed method extracts informative global and local features for effective image classification.

## II. RELATED WORK

Several GP variants have been proposed to extract image features from raw images. These features are typically global features which are synthesised from a whole input image. A multi-objective GP approach was proposed by Shao *et al.* [22] to learn features for image classification. This method was based on the idea of Convolutional Neural Networks (CNNs), where four layers i.e. a data input layer, a filtering layer, a max-pooling layer, and a concatenate layer were designed. This method used an image as input and returned a feature vector concatenated from the image after a number of operations such as filtering and pooling. A linear Support Vector Machine (SVM) was employed to perform classification based on the learnt features. However, the comparisons of the learnt features

with the other extracted features by HOG, LBP, SIFT and others were not entirely fair due to the reason that this method used more instances to learn features at first.

Al-Sahaf *et al.* [6] proposed a GP approach to automatically evolving texture descriptors for texture image classification with a small number of training instances. A feature vector is generated by the evolved GP descriptor scans an image via a sliding window. The evolved descriptors take the pixel statistics, i.e. mean, max, standard deviation, and min, of the sliding window as inputs. Thus the extracted features are invariant to rotation. A number of conventional classification methods, such as  $k$ -Nearest Neighbour (1NN), were employed to perform classification based on the described feature vector. Compared to the features extracted by LBP, GLCM and other LBP variants, the features described by this method were more informative and effective for texture image classification.

Later on, Al-Sahaf *et al.* [23] developed a dynamic GP method where a flexible length of feature vector was synthesised for texture classification. The experimental results showed that this method could extract features with a flexible length and outperform the method proposed in [6]. However, these two GP descriptors inspired by the LBP descriptor were originally proposed for describing texture features, which might not perform well on other image data.

Price and Anderson [24] designed a GP method for image feature descriptor learning and applied multiple kernel learning and support vector machine (MKL-SVM) for classification. This method employed a set of image operators, such as Canny, Hough Circle and Harris Corner Detector, as functions to learn image descriptors. Experiments showed promising results in image classification. However, more data sets should be used to examine the performance of this method.

GP has been successfully applied to region detection, feature extraction and image classification simultaneously. The extracted features are based on the automatically detected regions, which are local features. Atkins *et al.* [25] developed a multi-tier GP approach (simplified as 3TGP in [26]) to achieving automatic feature extraction and image classification. In this method, three tiers including an image filtering tier, an aggregation tier and a classification tier, were designed where each tier targeted a subtask. The aggregation tier was employed for extracting domain independent features, i.e. pixel statistics, from the automatically detected regions. These extracted features were further constructed in the classification tier to a single high-level feature, which was used for classification based on a predefined threshold.

Later, the 3TGP approach was changed by Al-Sahaf *et al.* [26] who proposed a two-tier GP (2TGP) approach to image feature extraction and classification using raw images as input. The representation of 3TGP was simplified in 2TGP where only the aggregation tier and the classification tier were employed. Two variants of 2TGP were proposed in [27] to automatically detect more flexible shapes and size of regions and to extract features from these detected regions. However, the performance of the 3TGP and 2TGP methods was evaluated on a few data sets with a small size of images.

A GP-HoG method was proposed by Lenson *et al.* [18] in 2016, where GP was employed for automatic region detection, feature construction, feature extraction, and image classification using raw pixel values as input. This approach used the advanced feature descriptor HOG as a function in GP to extract high-level HOG histogram features from the automatically detected regions. The GP-HoG method demonstrated a good example to integrate the advanced HOG descriptor in GP to achieve high-level feature extraction from the automatically detected regions and showed promising results in image classification. Compared to the 2TGP method, the GP-HoG method has obtained better results on all the three date sets.

### III. PROPOSED METHOD

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

#### A. Overall Program Structure

To achieve automatic global and local feature extraction, a new program structure is designed on the basis of the Strongly Typed GP [32] with a tree-based representation. An example program is shown in Fig. 1. This example program takes an image as input and produces a concatenated feature vector as output, representing a set of global and local features.

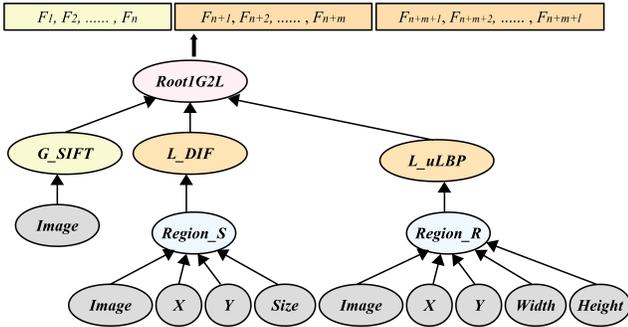


Fig. 1. An example of the program structure.

The example program can be divided into two parts, where conventional feature extraction methods such as SIFT are used to perform feature extraction in the global or local scenario. As shown in different colours, the left branch of the example program is for extracting global features from an input image using feature extraction methods ( $G\_SIFT$ ). The right two branches of the example program are for local feature extraction, where two different region detectors ( $Region\_S$ ,  $Region\_R$ ) are employed before using feature extraction operators ( $L\_DIF$ ,  $L\_uLBP$ ).

To concatenate all the extracted features, different root nodes such as  $RootIG2L$  are designed to synthesise all the extracted features in different scenarios as a feature vector. Compared to the majority of existing GP methods where the length of the output of the evolved GP program is fixed, in the proposed GP-GLF method, the output length (namely the length of the feature vector synthesised) of a GP program is variable. For each evolved program, the length of the

output feature vector is determined by the detected regions, the evolved root nodes, and the evolved feature extraction methods. The variable length of the synthesised feature vector is more flexible and adaptive for representing image features for different image classification tasks.

To avoid generating a feature vector with too high dimension, six different root nodes are developed. They are  $RootIG1L$ ,  $RootIG2L$ ,  $Root2G2L$ ,  $Root1G3L$ ,  $Root2G3L$ , and  $Root1G4L$ , where  $L$  indicates local feature extraction and  $G$  indicates global feature extraction. The six designed root nodes represent six different program structures of the proposed GP-GLF method, as shown in Fig. 2. These program structures allow the proposed GP-GLF method to extract different types of global features, detect variable types of regions of interest, and extract different types of local features. In the simplest case, one type of global features and local features is extracted. In the most complex cases, two types of global features and three types of local features, or one type of global features and four types of local features are extracted.

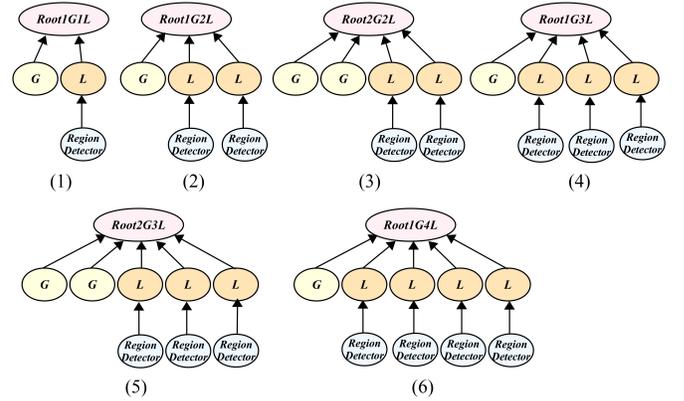


Fig. 2. Six program structures (only root nodes and internal nodes showing).

#### B. Function Set and Terminal Set

In the proposed GP-GLF method, a set of well-known feature extraction/description methods are employed as functions in the function set. These methods are DIF [7], Hist, GLCM [8, 9], Gabor bank filters (GB) [33], SIFT [11], HOG [10], uniform Local Binary Patterns histogram (uLBP) [12, 13].

As the aim of the proposed GP-GLF method is to extract both local and global image features, these feature extraction methods are employed in global scenario and local scenarios. In global scenario, these methods are used to extract a feature vector from a whole image. These methods are marked as  $G\_DIF$ ,  $G\_Hist$ ,  $G\_GLCM$ ,  $G\_GB$ ,  $G\_SIFT$ ,  $G\_HOG$ ,  $G\_LBP$ . In local scenario, these methods are to describe a feature vector from a detected region. These functions are  $L\_DIF$ ,  $L\_Hist$ ,  $L\_GLCM$ ,  $L\_GB$ ,  $L\_SIFT$ ,  $L\_HOG$ ,  $L\_uLBP$ . These functions employ region detectors as their children nodes.

These functions take an image or a region as input and return a feature vector with a variable length. The parameter settings for several methods in different scenarios are different. More details about these functions are shown in Table I. The fourth column and the fifth column of Table I list the number

TABLE I  
FEATURE EXTRACTION FUNCTIONS

Methods	Input	Output	No. (G)	No. (L)	Description
DIF	1 Image/Region	1 Vector	20	20	Domain independent features [7].
Hist	1 Image/Region	1 Vector	32	32	Histogram features using 32 bins based on the pixel intensities of the image/region.
GLCM	1 Image/Region	1 Vector	24	24	GLCM features [8, 9]. Four different orientations are used and the contrast, dissimilarity, homogeneity, energy, correlation and ASM are extracted from each GLCM.
GB	1 Image/Region	1 Vector	384	Flexible	Gabor bank features [33]. 24 Gabor filters with four different orientations at six scales are used. In global scenario, the mean value of each $32 \times 32$ grid is extracted from the image after filtering. In local scenario, the size of the grid is $16 \times 16$ .
SIFT	1 Image/Region	1 Vector	128	128	SIFT features [11]. The whole image/region is considered as a keypoints.
HOG	1 Image/Region	1 Vector	144	Flexible	HOG features [10]. Using the HOG method to extract features. In global scenario of HOG, the orientation is set to 9, the pixels in each cell is set to $32 \times 32$ , and the cells in each block is set to $3 \times 3$ . The mean value of a $20 \times 20$ grid is extracted from the image after filtering. In local scenario of HOG, the orientation is set to 9, the pixels in each cell is set to $8 \times 8$ , and the size of the grid is $10 \times 10$ .
uLBP	1 Image/Region	1 Vector	59	59	uniform LBP histogram features [12, 13].

TABLE II  
TWO REGION DETECTORS

Name	Input	Output
Region_S	1 Image, X, Y, Size	1 Square Region
Region_R	1 Image, X, Y, Width, Height	1 Rectangle Region

of features extracted by the corresponding method in global (G) or local (L) scenario.

To extract local features, two region detectors are designed and employed, i.e. *Region\_S* and *Region\_R*, as shown in Table II. These two detectors can detect a square/rectangle region at an appropriate position in an image with a suitable size by taking several arguments as inputs. The *Region\_S* operator detects a square region, which requires four arguments, i.e. *Image*, *X*, *Y*, and *Size*. The *Region\_R* operator detects a rectangle region, which needs five arguments, i.e. *X*, *Y*, *Size*, *Width*, and *Height*.

Four different types of terminals are used for representing the inputs of the designed GP system. They are *Image*, *X*, *Y*, *Size*, *Width*, and *Height*. The first terminal is the *Image*, which represents the input grey-scale image. It is a 2-D array with image pixel values in range  $[0, 1]$  (the raw image is normalised by dividing 255). The *X* and *Y* terminals are the coordinates of the top left point of a detected region in the input image. They are integers in range  $[0, ImageWidth - 20]$  or  $[0, ImageHeight - 20]$ . The terminals, *Size*, *Width* and *Height*, mean the size, the width and height of a square/rectangle region. They are between  $[20, 70]$  as the image sizes of our data sets are  $128 \times 128$ . The values of *X*, *Y*, *Size*, *Width*, and *Height* are randomly generated initially and evolved during the evolutionary process.

### C. Evaluation Process and Fitness Function

The overall evaluation process of the GP-GLF method contains two phases, i.e. training and testing. In the training process, the GP-GLF method is used to evolve image descriptors to extract global and local features from the input image and a simplest classifier, 1NN, is employed to perform classification based on the learnt image features. Due to 1NN needs a training set and a test set to evaluate the performance, two collections of images are used for training in this process, i.e. a tra-train set and a tra-test set. The tra-train set is used as

a training set and the tra-test set is used as a test set for 1NN in the training process. The fitness function of the GP-GLF method is the classification accuracy of the tra-test set.

In the testing phase, the best individual in the training phase is set as the final image descriptor. This image descriptor is used to extract image features and 1NN is used to perform classification. An unseen data to the training phase is set as a test set and the tra-train set is used as a training set in 1NN. The classification accuracy of the test set is reported. Therefore, to evaluate the performance of the proposed method, a data set must be spilt into three subgroups, i.e. a tra-train set, a tra-test set and a test set.

## IV. EXPERIMENTAL DESIGN

### A. Data sets

To evaluate the performance of the extracted features by the proposed GP-GLF method, four different binary image classification data sets of varying difficulty are employed. They are **JAFFE** [34], **YALE** [35], **FLOWER** [36], and **TEXTURE** [37]. JAFFE and YALE are two facial expression image data sets. The JAFFE data set is formed by the *Happy* and *Surprised* classes from the Japanese Female Facial Expression (JAFFE) Database [34]. The YALE is built by the centred and rotated images with *Happy* and *Sad* expressions from the Yale Face Database [35]. The FLOWER data set contains *Lotus* and *Sunflower* classes from the Caltech 101 Dataset [36]. The TEXTURE data set consists of the *Cork* and *Brown bread* classes from the KTH-TIPS2 Database [37]. All the images are grey-scale images. More details about these four data sets are shown in Table III. Two example images from each class are shown in Fig. 3.

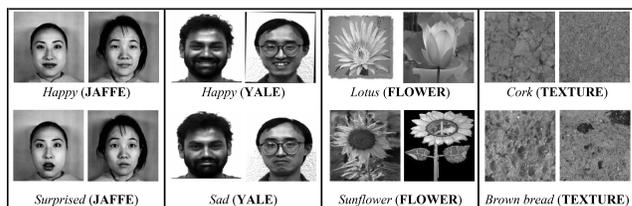


Fig. 3. Example images from each class of the four data sets.

TABLE III  
DATA SET PROPERTIES

Name	Size	Classes	Tra-train Set Size	Tra-test Set Size	Test Set Size
JAFFE	128×128	<i>Happy</i>	10	10	10
		<i>Surprised</i>	10	10	10
YALE	128×128	<i>Happy</i>	15	7	8
		<i>Sad</i>	15	7	8
FLOWER	128×128	<i>Lotus</i>	34	16	16
		<i>Sunflower</i>	43	21	21
TEXTURE	128×128	<i>Cork</i>	216	103	113
		<i>Brown bread</i>	216	108	108

### B. Baseline methods

To examine the effectiveness of the proposed method, seven GP-based methods and a set of non-GP methods are employed for comparisons.

1) *GP methods*: The seven GP methods are MLGP [28], GP-RDP [29], 2TGP [26], 3TGP [25], DIF+GP [7], Hist+GP, and uLBP+GP. The MLGP, GP-RDP, 2TGP, and 3TGP methods take an image as input and return a high-level image feature for binary image classification. The region detection, feature extraction, feature construction, and image classification are simultaneously achieved in these methods. The DIF+GP, Hist+GP and uLBP+GP methods use extracted image features as input, where GP is employed as a classification method. In Hist+GP, 64 histogram features are extracted and used as input of GP. In uLBP+GP, 59 uniform LBP features are used as input of GP. In contrast to the proposed method, these methods do not need a tra-test set during the training phase. Thus the tra-test set is used as a validation set to select the best individual for testing.

In these competitors, only one image feature is constructed for classification, while the GP-GLF method extracts a set of features for classification. The comparisons may not be entirely fair. But the comparisons will show whether the GP-GLF method can achieve better performance than the existing GP methods.

2) *Non-GP methods*: As the GP-GLF method employs a set of feature extraction methods as functions, it should be compared with these methods and the combinations of them to show its effectiveness. The non-GP methods are based on seven effective feature extraction methods and seven conventional classification algorithms including ensemble methods. The seven feature extraction methods have been introduced in Section III-B. They are DIF, Hist, GLCM, GB, SIFT, HOG, and uLBP. The parameter settings of these methods are the same as that used in the global scenario of the GP-GLF method. Specifically, in Hist, there are 256 histogram features are extracted. The seven classification methods are 1NN, naive Bayes (NB), decision tree (DT), multilayer perception (MLP), adaptive boosting (AdaBoost), random forest (RF), and SVM. The implementation of these methods are based on the well-known *scikit-learn* [38] Python package.

Compared with each feature extraction method, the GP-GLF method extracts more features. To achieve fair comparison, the combinations of two features by the seven feature extraction methods, such as SIFT+GB, DIF+SIFT, SIFT+HOG, and

GB+HOG, are employed for comparison. The combination of all the features (All) is also employed for comparison. In all these non-GP methods, the tra-train set and the tra-test set are used for training and the test set is used for testing.

### C. Parameter Settings

The implementation of all the GP methods including the proposed method are based on the *DEAP (Distributed Evolutionary Algorithm in Python)* [39] package. The parameter settings of the proposed GP-GLF method are listed in Table IV. In the other GP methods, the parameter settings are the same as that in GP-GLF except for the population size. In GP-GLF, a smaller population size is used in order to reduce computation cost, while the population size of the other GP methods is 1024.

For all the GP methods, the experiment on each data set has been run 30 times independently and the average results of the 30 runs are reported. For all the non-GP methods, where the feature extraction methods are deterministic methods, the experiment on each data set has been run once.

TABLE IV  
GP RUN TIME PARAMETERS

Parameter	Value	Parameter	Value
Generations	50	Population size	100
Crossover rate	0.8	Mutation rate	0.19
Elitism rate	0.01	Tree-depth	2-6
Selection type	Tournament	Tournament size	7

## V. RESULTS AND DISCUSSIONS

This section discusses and compares the classification results obtained by the proposed GP-GLF method, the other GP methods and the non-GP methods.

The results obtained by the proposed method, the other GP methods and the non-GP methods are shown in Tables V and VI. To show the significance of classification performance improvement, the Student's *t*-test with a 5% significance level is employed to compare the GP-GLF method with a baseline method. In Tables V and VI, "+" indicates the GP-GLF method significantly outperforms the compared method, "-" means the GP-GLF method performs significantly worse than the compared method, and "=" indicates the GP-GLF method performs similar to the compared method.

### A. Compared with GP Methods

Table V summarises the overall classification results i.e. maximum, mean and standard deviation accuracy obtained by the GP-GLF method and the other seven GP methods on the four data sets. The best average accuracy on each data set is showed with bold font. Compared with all the other GP methods, the GP-GLF method achieves the best maximum classification accuracy and mean classification accuracy on all the four data sets. In the total 28 cases of comparisons, the GP-GLF method achieves significantly better results in 27 cases. Specifically, on the YALE data set, the GP-GLF method has a more than 14% increase in the mean accuracy compared with the other GP methods.

From Table V, it is obvious that the GP-GLF achieves significantly better results than all the GP methods. The main reason is that the GP-GLF method is able to extract a set of features for classification while the other GP methods produce one single constructed high-level feature for image classification. The comparisons show that it is more effective to use GP to extract many features for image classification.

TABLE V  
CLASSIFICATION ACCURACY(%) OF ALL THE GP METHODS ON THE FOUR DATA SETS

Methods	Max	Mean±St.D.	Max	Mean±St.D.
Data Set	<b>JAFFE</b>		<b>YALE</b>	
GP-GLF	100.0	<b>95.00+6.06</b>	100.0	<b>95.62+10.35</b>
MLGP	100.0	<b>91.67+6.50+</b>	87.5	73.96+12.51+
GP-RDP	95.0	82.83+8.03+	93.75	76.25+13.73+
2TGP	95.0	82.83+8.53+	100.0	<b>80.83+13.78+</b>
3TGP	100.0	82.67+9.20+	93.75	77.71+16.98+
DIF+GP	90.0	70.67+13.59+	81.25	61.04+8.49+
Histogram+GP	75.0	52.17+9.28+	81.25	49.38+14.01+
uLBP+GP	65.0	53.83+6.01+	81.25	55.83+13.40+
Data Set	<b>FLOWER</b>		<b>TEXTURE</b>	
GP-GLF	97.3	<b>86.4+4.60</b>	99.1	<b>97.84+0.75</b>
MLGP	86.49	67.57+9.39+	97.74	90.23+3.48+
GP-RDP	89.19	80.81+6.60+	88.24	80.81+4.51+
2TGP	94.59	<b>84.95+6.53+</b>	81.9	75.60+3.87+
3TGP	97.3	83.69+4.81+	88.24	82.68+4.18+
DIF+GP	89.19	80.18+5.94+	88.69	83.65+2.36+
Histogram+GP	62.16	53.33+5.67+	94.57	87.36+3.86+
uLBP+GP	89.19	64.77+12.57+	97.29	<b>92.37+2.77+</b>

### B. Compared with Non-GP Methods

The overall classification results obtained by the proposed GP-GLF method and the non-GP methods on the four data sets are presented in Table VI. The mean, standard deviation and maximum classification accuracy achieved by the GP-GLF method are showed with bold font in the second row of each block in Table VI. Except for using one type of features, two types of features are also used for comparison. The total number of combinations of two types of features is 21. Due to the page limitation, only better results achieved by the two types of features than the GP-GLF method, such as DIF+GB, are reported. The last row of each block in Table VI lists the results obtained on all the features (All).

On the JAFFE data set, compared with the non-GP methods with one type of image features, the GP-GLF method obtains significantly better or similar results in 46 cases out of the 49 ( $7 \times 7$ ) cases. Compared with all the methods with the two types of image features, the GP-GLF method achieves significantly better or similar performance in 139 cases out of the 147 ( $7 \times 21$ ) cases. On the YALE data set, the GP-GLF method achieves significantly better or similar results in all the cases of comparisons. The maximum accuracy obtained by the non-GP methods is 93.75%, which is lower than the mean accuracy (95.62%) achieved by the GP-GLF method.

On the FLOWRE data set, the GP-GLF method achieves similar or significantly better performance in 42 out of the 49 cases compared with the non-GP methods with one type of features. Compared with the non-GP methods with two types of features, the GP-GLF method gains significantly better or similar performance in 124 cases out of the 147 cases. On this data set, most of the better results are obtained by the GB and SIFT features and their combinations with other types of

TABLE VI  
CLASSIFICATION ACCURACY(%) OBTAINED BY THE PROPOSED METHOD AND THE NON-GP METHODS

Methods Data sets	1NN	NB	DT	MLP	AdaBoost	RF	SVM
<b>JAFFE</b>							
GP-GLF	<b>95.0±6.06 (100.0)</b>						
DIF	90.00+	55.00+	80.00+	50.00+	85.00+	75.00+	50.00+
Hist	90.00+	55.00+	60.00+	90.00+	60.00+	45.00+	90.00+
GLCM	65.00+	60.00+	60.00+	80.00+	50.00+	70.00+	50.00+
GB	100.0-	90.00+	75.00+	65.00+	95.00=	95.00=	50.00+
SIFT	100.0-	95.00=	90.00+	85.00+	80.00+	95.00=	85.00+
HOG	100.0-	85.00+	90.00+	50.00+	95.00=	95.00=	80.00+
uLBP	75.00+	65.00+	35.00+	45.00+	65.00+	85.00+	75.00+
DIF+GB	100.0-	90.00+	75.00+	95.00=	95.00=	90.00+	50.00+
DIF+HOG	90.00+	90.00+	90.00+	50.00+	90.00+	100.0-	50.00+
GB+HOG	100.0-	90.00+	75.00+	50.00+	95.00=	90.00+	50.00+
SIFT+Hist	90.00+	70.00+	95.00=	75.00+	90.00+	95.00=	90.00+
SIFT+GLCM	65.00+	60.00+	60.00+	50.00+	80.00+	100.0-	50.00+
SIFT+GB	100.0-	95.00=	90.00+	100.0-	80.00+	90.00+	50.00+
SIFT+HOG	100.0-	95.00=	90.00+	85.00+	80.00+	100.0-	85.00+
All	65.00+	65.00+	85.00+	50.00+	80.00+	85.00+	50.00+
<b>YALE</b>							
GP-GLF	<b>95.62±10.35 (100.0)</b>						
DIF	68.75+	68.75+	87.50+	50.00+	81.25+	75.00+	68.75+
Hist	56.25+	56.25+	68.75+	56.25+	68.75+	50.00+	56.25+
GLCM	50.00+	37.50+	81.25+	43.75+	68.75+	62.50+	56.25+
GB	56.25+	93.75=	81.25+	50.00+	81.25+	87.50+	56.25+
SIFT	62.50+	75.00+	62.50+	81.25+	62.50+	62.50+	75.00+
HOG	31.25+	50.00+	75.00+	43.75+	81.25+	81.25+	50.00+
uLBP	50.00+	62.50+	56.25+	43.75+	75.00+	75.00+	56.25+
DIF+GB	56.25+	93.75=	93.75=	68.75+	87.50+	87.50+	56.25+
GB+Hist	43.75+	87.50+	87.50+	43.75+	81.25+	93.75=	56.25+
GB+uLBP	50.00+	87.50+	75.00+	50.00+	93.75=	75.00+	56.25+
All	50.00+	43.75+	93.75=	31.25+	87.50+	75.00+	56.25+
<b>FLOWER</b>							
GP-GLF	<b>86.4±4.6 (97.3)</b>						
DIF	72.97+	83.78+	78.38+	59.46+	81.08+	83.78+	59.46+
Hist	67.57+	70.27+	48.65+	62.16+	64.86+	62.16+	56.76+
GLCM	56.76+	62.16+	56.76+	51.35+	67.57+	70.27+	56.76+
GB	86.49=	81.08+	75.68+	94.59-	86.49=	86.49=	56.76+
SIFT	91.89-	97.30-	89.19-	97.30-	97.30-	97.30-	56.76+
HOG	64.86+	51.35+	78.38+	51.35+	70.27+	81.08+	56.76+
uLBP	59.46+	78.38+	59.46+	70.27+	62.16+	62.16+	56.76+
DIF+GB	86.49=	81.08+	70.27+	94.59-	91.89-	94.59-	56.76+
SIFT+DIF	81.08+	97.30-	89.19-	97.30-	94.59-	94.59-	56.76+
GB+GLCM	56.76+	62.16+	64.86+	43.24+	86.49=	94.59-	56.76+
GB+Hist	72.97+	72.97+	72.97+	75.68+	89.19-	86.49=	56.76+
GB+HOG	72.97+	64.86+	62.16+	97.30-	91.89-	78.38+	56.76+
SIFT+GB	86.49=	89.19-	83.78+	94.59-	94.59-	94.59-	56.76+
SIFT+GLCM	56.76+	62.16+	89.19-	56.76+	94.59-	91.89-	56.76+
SIFT+Hist	67.57+	72.97+	86.49=	70.27+	97.30-	86.49=	56.76+
SIFT+HOG	81.08+	54.05+	91.89-	83.78+	97.30-	94.59-	56.76+
SIFT+uLBP	59.46+	86.49=	86.49=	78.38+	97.30-	86.49=	56.76+
All	56.76+	67.57+	86.49=	54.05+	97.30-	89.19-	56.76+
<b>TEXTURE</b>							
GP-GLF	<b>97.84±0.75 (99.10)</b>						
DIF	90.50+	84.16+	87.33+	51.13+	90.50+	90.50+	78.28+
Hist	93.21+	85.97+	91.86+	90.95+	95.93+	94.12+	48.87+
GLCM	83.71+	72.40+	94.57+	47.06+	96.38+	92.31+	48.87+
GB	98.19-	79.64+	82.81+	96.38+	93.21+	91.86+	48.87+
SIFT	95.93+	92.31+	86.88+	84.62+	86.88+	88.69+	48.87+
HOG	71.49+	69.23+	67.42+	71.04+	71.95+	73.76+	62.90+
uLBP	96.83+	86.88+	85.07+	85.52+	90.05+	90.05+	48.87+
DIF+GB	98.19-	81.00+	87.78+	91.86+	95.02+	91.40+	48.87+
GB+Hist	98.64-	85.07+	92.76+	93.21+	99.10-	95.48+	48.87+
GB+HOG	99.10-	79.19+	87.33+	93.21+	97.74=	91.40+	48.87+
GB+SIFT	98.19-	84.16+	88.69+	96.83+	91.86+	93.21+	48.87+
GB+uLBP	99.55-	83.26+	90.05+	92.76+	97.29+	93.67+	48.87+
GLCM+uLBP	83.71+	78.73+	95.48+	48.42+	98.19-	96.83+	48.87+
Hist+uLBP	98.64-	87.78+	92.31+	93.21+	98.64-	95.02+	48.87+
All	83.71+	77.83+	94.57+	56.11+	97.74=	97.29+	48.87+

features. The main reason is that images in this data set have objects with scale, deformation and rotation variations, where can be well handled by the SIFT and GB features. However, it should be noticed that the majority of these better results of the non-GP methods are obtained by ensemble methods, while in GP-GLF, only 1NN is employed. On the TEXTURE data

set, the GP-GLF method gains significantly better or similar results in 48 cases out of the 49 cases compared with the non-GP methods with one type of features. Compared with the non-GP methods with two types of features, the GP-GLF method gains significantly better or similar performance in 138 cases out of the 147 cases. In this data set, INN on the combination of features GB+uLBP obtains the best classification accuracy i.e. 99.55% among all these methods.

Compared with the non-GP method with all the seven types of features (All), the GP-GLF method achieves significantly better or similar results on the JAFFE, YALE and TEXTURE data sets. On the FLOWER data set, the GP-GLF method gains significantly worse results in 2 cases out of the 7 cases compared with the non-GP method with all the features.

In summary, compared with most of the non-GP methods and the GP methods, the GP-GLF method achieves significantly better or similar results. There are two main reasons that can explain why the proposed method achieved better results than the original feature extraction methods. Firstly, local features are extracted by the GP-GLF method, which are more discriminative than the global features especially on face image classification and object image classification. Secondly, instead of selecting feature extraction methods manually given a image classification task, the GP-GLF method can automatically evolve suitable feature extractors to extract discriminative image features for classification. One drawback is the GP-GLF method extracts more features than the original feature extraction method. But the comparisons with classification results on two types of features and all the features show that the GP-GLF method is able to extract more informative image features for effective image classification.

## VI. FURTHER ANALYSIS

This section analyses example programs evolved by the proposed GP-GLF method to better understand why it can extract informative image features with the potential of achieving good classification performance.

### A. An Example Program on The JAFFE Data Set

Fig. 4 shows an example program obtained on the JAFFE data set with 100% classification accuracy in training and testing. This example tree identifies two rectangle regions with different sizes at different positions in an input image. The right part of Fig. 4 gives examples of the original image and two detected regions. The smaller detected region captures the lower right side of a face, which contains partial shape of the face and shows difference under two different expressions. The bigger detected regions detects the middle left part of a face where the left eye, partial nose and left eyebrow are included. The major difference of eye showings in the *Happy* and *Surprised* expressions are captured by this detected region.

In this example tree, a  $G\_SIFT$  operator is evolved to describe a 128D feature vector from an input image, which represents the global features. Two types of local features are extracted by the evolved  $L\_GB$  and  $L\_SIFT$  operators. The  $L\_GB$  operator extracts 24 features from the smaller detected

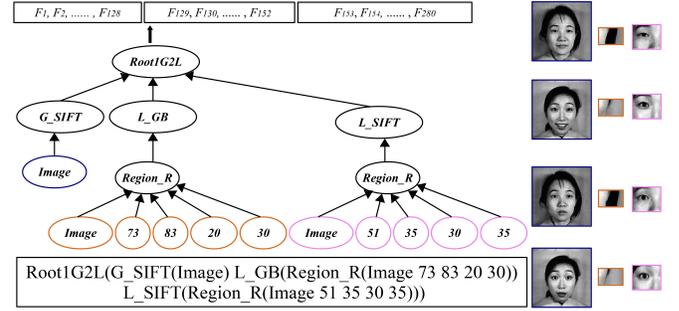


Fig. 4. An example program evolved by the proposed GP-GLF method on the JAFFE data set. The right part shows example images and the detected regions.

region. The  $L\_SIFT$  describes 128 features from the bigger detected region. In total, this example tree extracts 280 features including both global and local features from an input image.

### B. Example Programs on The Other Three Data Sets

Table VII lists the best programs evolved from the YALE, FLOWER and TEXTURE data sets by the proposed GP-GLF method. All these selected programs achieve the best classification performance in training and testing. On the YALE data set, one type of global feature and two types of local feature are extracted by the evolved feature extraction methods, which are HOG, uLBP and Hist. On the difficult object classification data set FLOWER, the SIFT feature extraction method is evolved to extract both global and local features from an input image as it is invariant to translation, scaling, and rotation variations [11]. On the TEXTURE data set, the best evolved program contains GLCM, uLBP and Hist descriptors, which are good at extracting texture and distribution information from an image/region. As these three data sets represent different image classification tasks, the analysis on the evolved programs reveals that the proposed GP-GLF method can automatically select effective feature extractors for a given image classification task.

TABLE VII  
THE BEST EVOLVED PROGRAMS ON THE OTHER THREE DATA SETS

Data Sets	Evolved Programs	Performance
YALE	Root2G2L(G_HOG(Image) G_HOG(Image)	100% (Training),
	L_uLBP(Region_S(Image 53 92 21))	100% (Testing)
	L_Hist(Region_S(Image 78 52 32))	
FLOWER	Root1G2L(G_SIFT(Image) L_SIFT(Region_S	100% (Training),
	(Image 33 41 52)) L_SIFT(Region_S(Image 31 41 52))	97.30% (Testing)
TEXTURE	Root2G3L(G_GLCM(Image) G_uLBP(Image)	100% (Training),
	L_Hist(Region_S(Image 9 59 28))	99.10% (Testing)
	L_Hist(Region_S(Image 7 20 36))	
	L_Hist(Region_R(Image 58 62 25 28))	

## VII. CONCLUSIONS

This paper proposed a new GP-GLF method which can automatically extract global and local features from an input image for image classification. In the GP-GLF method, a new function set, a new terminal set and a novel program structure were designed. A set of feature extraction methods were employed as GP functions in global and local scenarios to extract informative and discriminative image features. Six new root nodes were developed to concatenate extracted global

and local features to a feature vector. Four different image data sets of varying difficulty were employed to test the performance of the GP-GLF method on image classification. Compared with the seven GP methods and a set of non-GP methods, the GP-GLF method achieved significantly better or similar performance on the four data sets in almost all the comparisons. Further analysis on evolved programs demonstrated the good interpretability and understandability of the GP-GLF method. The analysis also revealed that the GP-GLF method can evolve effective and adaptive image descriptors to achieve automatic global and local feature extraction for effective image classification.

In this paper, the GP-GLF method has shown promising results in different image classification tasks. In the near future, we will focus on investigating a new transfer learning approach in it to speed up the learning.

#### REFERENCES

- [1] M. Ohi, Y. Li, Y. Cheng, and T. Walz, "Negative staining and image classification powerful tools in modern electron microscopy," *Biological Procedures Online*, vol. 6, no. 1, pp. 23–34, 2004.
- [2] E. Hjeltnäs and B. K. Low, "Face detection: A survey," *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236–274, 2001.
- [3] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, 2007.
- [4] M.-L. Antonie, O. R. Zaiane, and A. Coman, "Application of data mining techniques for medical image classification," in *Proceedings of the Second International Conference on Multimedia Data Mining*. Springer-Verlag, 2001, pp. 94–101.
- [5] M. Hassaballah, A. A. Abdelmegeid, and H. A. Alshazly, "Image features detection, description and matching," in *Image Feature Detectors and Descriptors*. Springer, 2016, pp. 11–45.
- [6] H. Al-Sahaf, A. Al-Sahaf, B. Xue, M. Johnston, and M. Zhang, "Automatically evolving rotation-invariant texture image descriptors by genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 83–101, 2017.
- [7] M. Zhang and V. Ciesielski, "Genetic programming for multiple class object detection," in *Australian Joint Conference on Artificial Intelligence*, vol. 1747. Springer, Heidelberg, 1999, pp. 180–192.
- [8] R. M. Haralick, K. Shanmugam *et al.*, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 6, pp. 610–621, 1973.
- [9] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
- [10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893.
- [11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, vol. 1. IEEE, 1994, pp. 582–585.
- [13] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [14] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision—ECCV 2006*, pp. 404–417, 2006.
- [15] H. Al-Sahaf, "Genetic programming for automatically synthesising robust image descriptors with a small number of instances," *PhD thesis, Victoria University of Wellington, New Zealand*, 2017.
- [16] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.
- [17] B. Xue and M. Zhang, "Evolutionary feature manipulation in data mining/big data," *ACM SIGEVOlution.*, vol. 10, no. 1, pp. 4–11, 2017.
- [18] A. Lensen, H. Al-Sahaf, M. Zhang, and B. Xue, "Genetic programming for region detection, feature extraction, feature construction and classification in image data," in *European Conference on Genetic Programming*. Springer, Heidelberg, 2016, pp. 51–67.
- [19] W. Fu, M. Johnston, and M. Zhang, "Low-level feature extraction for edge detection using genetic programming," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1459–1472, 2014.
- [20] Y. Liang, M. Zhang, and W. N. Browne, "Genetic programming for evolving figure-ground segmentors from multiple features," *Applied Soft Computing*, vol. 51, pp. 83–95, 2017.
- [21] R. d. S. Torres, A. X. Falcão, M. A. Gonçalves, J. P. Papa, B. Zhang, W. Fan, and E. A. Fox, "A genetic programming framework for content-based image retrieval," *Pattern Recognition*, vol. 42, no. 2, pp. 283–292, 2009.
- [22] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1359–1371, 2014.
- [23] H. Al-Sahaf, M. Zhang, A. Al-Sahaf, and M. Johnston, "Keypoints detection and feature extraction: A dynamic genetic programming approach for evolving rotation-invariant texture image descriptors," *IEEE Transactions on Evolutionary Computation*, pp. 825–844, 2017.
- [24] S. R. Price and D. T. Anderson, "Genetic programming for image feature descriptor learning," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 854–860.
- [25] D. Atkins, K. Neshatian, and M. Zhang, "A domain independent genetic programming approach to automatic feature extraction for image classification," in *2011 IEEE Congress on Evolutionary Computation*. IEEE, 2011, pp. 238–245.
- [26] H. Al-Sahaf, A. Song, K. Neshatian, and M. Zhang, "Extracting image features for classification by two-tier genetic programming," in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012. [Online]. Available: DOI:10.1109/CEC.2012.6256412
- [27] —, "Two-tier genetic programming: Towards raw pixel-based image classification," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12 291–12 301, 2012.
- [28] Y. Bi, B. Xue, and M. Zhang, "An automatic feature extraction approach to image classification using genetic programming," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2018, pp. 421–438.
- [29] Y. Bi, M. Zhang, and B. Xue, "An automatic region detection and processing approach in genetic programming for binary image classification," in *The 32st International Conference Image and Vision Computing New Zealand (IVCNZ 2017)*. IEEE, 2017, pp. 1–6.
- [30] T. W. Chow and M. Rahman, "A new image classification technique using tree-structured regional features," *Neurocomputing*, vol. 70, no. 4, pp. 1040–1050, 2007.
- [31] D. ping Tian *et al.*, "A review on image feature extraction and representation techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 4, pp. 385–396, 2013.
- [32] D. J. Montana, "Strongly typed genetic programming," *Evolutionary computation*, vol. 3, no. 2, pp. 199–230, 1995.
- [33] B. S. Manjunath and W.-Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.
- [34] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *The Third IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 1998, pp. 200–205.
- [35] A. Georghiadis, P. Belhumeur, and D. Kriegman, "Yale face database," *Center for Computational Vision and Control at Yale University*, <http://cvc.yale.edu/projects/yalefaces/yalefa>, vol. 2, 1997.
- [36] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.
- [37] P. Mallikarjuna, A. T. Targhi, M. Fritz, E. Hayman, B. Caputo, and J.-O. Eklundh, "The kth-tips2 database," *Computational Vision and Active Perception Laboratory (CVAP), Stockholm, Sweden*, 2006.
- [38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [39] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.