An Effective Feature Learning Approach Using Genetic Programming with Image Descriptors for Image Classification

Ying Bi, Bing Xue, and Mengjie Zhang
School of Engineering and Computer Science, Victoria University of Wellington, Wellington, NEW ZEALAND

***Abstract*—Being able to extract effective features from different images is very important for image classification, but it is challenging due to high variations across images. By integrating existing well-developed feature descriptors into learning algorithms, it is possible to automatically extract informative high-level features for image classification. As a learning algorithm with a flexible representation and good global search ability, genetic programming can achieve this. In this paper, a new genetic programming-based feature learning approach is developed to automatically select and combine five existing well-developed descriptors to extract high-level features for image classification. The new approach can automatically learn various numbers of global and/or local features from different types of images. The results show that the new approach achieves significantly better classification performance in almost all the comparisons on eight data sets of varying difficulty. Further analysis reveals the effectiveness of the new approach to finding the most effective feature descriptors or combinations of them to extract discriminative features for different classification tasks.**

I. Introduction

Image classification is a fundamental task in computer vision with a wide range of real-world applications, including remote sensing, medical diagnosis, biologic identification, and self-driving [1–3]. Image classification is the task of assigning each image in the data set with a pre-defined class label according to the content in the image. The task is an important component of other computer vision tasks, such as object recognition. However, image classification is very challenging due to inter-class and intra-class variations in scale, illumination, rotation, and occlusion [4].

Feature extraction is important for solving image classification. The aim of feature extraction is to extract a set of discriminative features from a raw image for classification. Effective feature extraction methods can improve the performance of a classification system [4]. Existing methods, such as histogram of orientated gradients (HOG) [5], scale-invariant feature transform (SIFT) [6], and local binary patterns (LBP) [7], can extract high-level features that are invariant to certain variations such as scale and rotation. However, these methods are only effective for describing features from particular image domains and using them to solve new tasks often requires domain knowledge [8]. In the process of feature extraction, domain experts decide not only how to extract features but also what types of features should be extracted. Generally, there are two types of features, global features and local features, which are effective for different image domains [9].

Corresponding Author: Ying Bi (Email: Ying.Bi@ecs.vuw.ac.nz)

Feature learning techniques can automatically learn informative and problem-specific features from raw images for classification without domain knowledge. The learned features are often more effective for classification than manually extracted features [8]. As an evolutionary algorithm, genetic programming (GP) has been applied to feature learning for image classification and achieved promising results [8, 10, 11]. GP can automatically evolve computer programs to solve problems using principles of biological evolution and natural selection [12]. Compared with other evolutionary computation (EC) techniques, GP has a more flexible tree-based representation. Besides, GP is well known for its high interpretability of evolved solutions and good global search ability.

With a flexible tree-based representation, it is possible to integrate existing image-related operators into GP to automatically learn high-level features. For example, Gaussian filter, Sobel edge detector and Laplacian of Gaussian have been integrated into GP for feature extraction [8]. However, the features extracted by these filtering operators may not handle certain variations, such as rotation. Existing feature descriptors, such as HOG, SIFT and LBP, are well-developed and effective for dealing with certain image variations. However, these descriptors have not been systematically integrated into GP to achieve feature learning. Therefore, this paper proposes a new GP-based method to integrate these descriptors to automatically learn discriminative and robust features for image classification.

Our previous work in [13] proposed an initial method to integrate image descriptors into GP to learn global and local features (GP-GLF) for image classification. GP-GLF has achieved better performance than a large number of methods using manually-extracted features on four different binary image classification tasks. In GP-GLF, six different root nodes have been developed to combine the global features and local features extracted by descriptors, which allows GP-GLF to produce a combination of global and local features. This design may not be effective for particular tasks. For example, texture classification may only need global features and object classification may only need local features. Moreover, the performance of GP-GLF has not been examined on multi-class image classification. Therefore, this paper develops a new GP-based feature learning approach by addressing the limitations of GP-GLF. The new approach aims to automatically learn

global and/or local features in a flexible way.

The overall goal of this paper is to develop a new Feature Learning approach using GP (FLGP) to automatically select and combine existing feature descriptors to extract rich and discriminative global and/or local features for different image classification tasks. To achieve this goal, a novel program structure (individual representation), a new function set and a new terminal set will be developed in FLGP. To effectively learn discriminative features, a new feature learning process and a new fitness evaluation process will be developed. The new approach will be examined on eight different image classification data sets, including multi-class classification data sets. Further analysis of computational cost and example solutions will be conducted to provide deep insights into FLGP.

The main contributions of this paper are summarized in four aspects.

1) We develop a new function set and a new terminal set for the new FLGP approach. The new function set has three different types of functions with different functionalities. More importantly, five representative and well-known feature descriptors, i.e., uniform LBP (uLBP), HOG, SIFT, domain-independent features (DIF), and histogram (Hist), are developed as feature extraction functions for FLGP, where very few works have addressed this.
2) To better utilize the functions including the descriptors, we develop a novel program structure for FLGP to integrate the processes of global feature extraction and/or local feature extraction into a single tree. The new program structure allows FLGP to evolve solutions of variable depths to produce various numbers of features.
3) To improve the generalization ability of the learned features by FLGP, a new feature learning process and a new fitness evaluation process are developed. The min-max normalization method is employed to normalize the features from different descriptors. A linear support vector machine (SVM) for *5*-fold cross-validation is employed to evaluate the individual based on the normalized features.
4) The experimental results show that FLGP significantly outperforms almost all the benchmark methods on eight data sets. Further analysis shows that FLGP provides tree-based solutions that make it easy to understand which feature descriptors are selected, which regions of interest are detected, and whether global features or local features are extracted. This is helpful for obtaining further insight into the target problems, which has been an issue for many existing methods including convolutional neural networks (CNNs).

II. Background and Related Works

This section describes several typical feature description methods and introduces GP and strongly typed GP (STGP). Then it discusses recent work related to this study and summarizes the limitations.

A. Common Feature Extraction/Description Methods

Hist: Extracting histogram features from raw pixel values is the simplest way to obtain features [14]. The Hist features are the values of the bins in the histogram calculated using all the pixel values of the image. The number of the bins often equals 256 since the pixel values are in the range of $[0,\ 255]$.

DIF: DIF [15] extracts mean and standard deviation values from six regions and four lines of an image/region. The six regions are the original image, four small subregions of the image and the center region. The four lines are the two middle lines of the image and two middle lines of the center region.

SIFT: SIFT [6] is a widely used keypoints detection and description method. It employs the difference of Gaussian (DoG) with different scales to detect extreme points. Then the Taylor function is used to optimize and eliminate low-contrast keypoints and the Hessian matrix is used to eliminate edge responses. For each keypoint, SIFT produces 128 histogram features of gradient magnitudes and orientations. Without detecting keypoints, a dense SIFT method is developed to extract features from images with less computational complexity [16]. Note that the dense SIFT method is employed in this paper (simplified as SIFT).

HOG: HOG [5] is a well-known shape and appearance description algorithm for human detection. It contains a number of steps, including gamma and color normalization, gradient computation, weighted voting into spatial and orientation cells, and contrast normalization over overlapping spatial blocks [5]. The main idea of HOG is to extract locally normalized histogram features of gradient orientations (cell) in a densely overlapping grid (block).

LBP: LBP [7] is a simple but effective texture description method. LBP compares each central pixel with its neighbor pixels in a sliding window to generate binary codes. Then the value of the central pixel is replaced by the sum of all the products of the binary code and pre-defined weights. The histogram of the generated LBP image is used as features for image analysis. Different LBP variants have been proposed, e.g., uniform LBP [7].

Others: Other feature extraction methods include grey-level co-occurrence matrix (GLCM) [17] and Gabor features [3].

B. Genetic Programming (GP) and Strongly Typed GP

Inspired by the principles of biological evolution and natural selection, GP automatically evolves computer programs for solving particular problems [12]. GP often uses a program tree to represent the solution [18]. For each tree, the root node and the internal nodes are functions/operators chosen from a function set and the leaf nodes are terminals chosen from a terminal set. In standard GP, the functions and terminals only deal with one type of data, e.g., float-pointing number. STGP is a variant of the tree-based GP to cope with multiple data types [19]. In STGP, each terminal has an input type and each function has an input type and an output type. A function only takes particular non-terminal or terminal as its children node. In STGP, a program structure is often required to integrate functions and terminals with different types into a tree-based solution. To use GP to solve a problem, it is necessary to carefully develop a program structure, a function set and a terminal set.

C. GP for Feature Learning

1) Learning Features from Pre-extracted Features: The commonly used way of GP for feature learning is to construct high-level features using simultaneously selected subsets of original features. Nandi *et al.* [20] employed GP to classify breast masses into the malignant class and the benign class using 22 features, i.e., shape, edge-sharpness and texture, as inputs. Ain *et al.* [21] proposed a GP-based method to feature selection using 12 domain-specific features and 59 uLBP features for skin cancer image classification. Choi and Choi [22] developed a GP-based system for pulmonary nodule detection on computed tomography images, where GP was used to evolve classifiers for categorizing nodules and non-nodules using four different types of features. However, these aforementioned methods require human intervention to extract features from images in advance.

2) Learning Features from Raw Pixels: Instead of using the pre-extracted features as inputs, GP-based methods have been proposed to automatically extract image features from raw pixels. Atkins *et al.* [23] proposed a multi-tier GP algorithm to transform raw pixels into high-level features using an image filtering tier, an aggregation tier and a classification tier for image classification. Lensen *et al.* [24] integrated HOG into GP to automatically extract HOG histogram or distance features from raw pixels. However, these two methods only produce one high-level feature, which might not be effective for multi-class image classification tasks. Shao *et al.* [8] developed a multi-objective GP (MOGP) method to automatically learn features using a set of image operators. However, MOGP extracted a large number of global features, where principal component analysis was employed to reduce the dimensionality of the features. Al-Sahaf *et al.* [10] proposed a GP-criptor^{ri} method to evolve texture descriptors for texture classification using a small number of training images. The way that the GP-criptor^{ri} solution describes features is similar to that by LBP. The result showed that GP-criptor^{ri} is more effective than LBP for texture classification. In [25], a variant of GP-criptor^{ri} was developed to automatically extract a dynamic number of texture features for classification. Knowledge transferring across different domains has also been employed to improve the performance of GP-criptor^{ri} [26]. However, these methods only extract texture features, which might not be effective for other types of images, e.g., facial images.

In summary, these aforementioned methods show the superiority of GP in feature learning for image classification. However, these works have their limitations, as described above. The potential of GP in feature learning has not been extensively investigated. GP-GLF has demonstrated the effectiveness of GP with existing well-developed image descriptors for feature learning in image classification [13]. However, this method has a fixed tree depth and is only able to learn a fixed type of features. Moreover, GP-GLF has only been examined on binary image classification tasks. Motivated by these limitations, this paper significantly improves the method by developing a new feature learning process, a new program structure, a new function set, and a new terminal set. The new method can automatically find/extract various numbers of global and/or local features for different image classification tasks. This new method will be examined on binary or multi-class classification tasks with a larger number of instances.

III. The Proposed Approach

This section describes the proposed FLGP approach in detail. The overall algorithm, the new feature learning process and the new fitness evaluation process, are presented first. Then it describes the main components of FLGP, i.e., the program structure, the function set and the terminal set.

A. Overall Algorithm

The proposed FLGP approach aims to automatically evolve solutions that extract discriminative global and/or local features using existing feature descriptors from the input image. The overall feature learning process of FLGP is shown in Fig. 1, where the left part shows the general evolutionary learning process of GP and the right part shows the new fitness evaluation process. The new components and the basic configuration of the FLGP system are shown in Fig. 2.

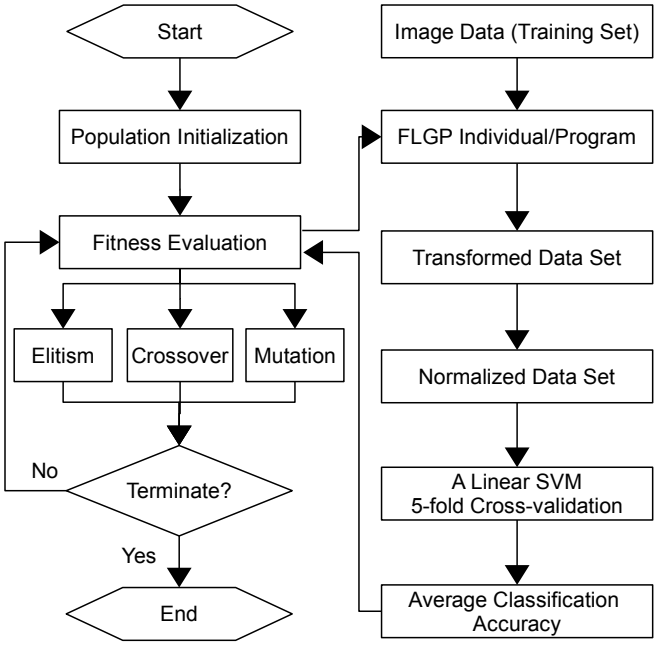

Fig. 1. The overall feature learning process of FLGP.

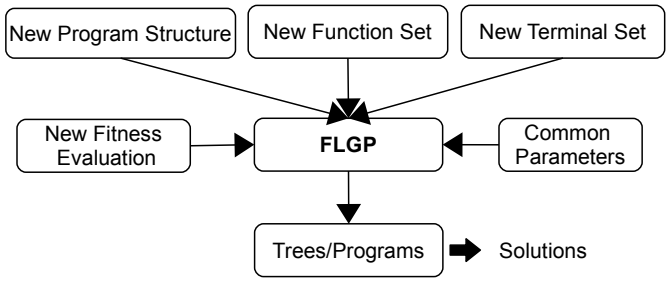

Fig. 2. The new components and the basic configuration of the FLGP system.

As shown in Fig. 1, FLGP starts with population initialization, where a number of programs/trees/individuals are randomly generated according to the new program structure, the new function and terminal sets. Then each individual is evaluated using the new fitness evaluation process. After fitness evaluation, a selection method and three genetic operators, i.e., *elitism, crossover* and *mutation*, are used to obtain a new population for the next generation. The selection method selects individuals with better fitness values for crossover and mutation. The crossover and mutation operations change the nodes or branches of the FLGP trees to search for the best one. The evolutionary learning process is terminated when a pre-defined termination criterion is satisfied. If the termination criterion is not satisfied, the process of fitness evaluation and population generation repeat again. Otherwise, the evolutionary process ends, and the best individual is returned.

During the evolutionary learning process, a fitness function is used to guide the search for the best individual. The right part of Fig. 1 shows the new fitness evaluation process in FLGP. In this process, a training set, containing N $m \times l$ images $\{I_i\}_{i=1}^{N}$ and N labels $\{Y_i\}_{i=1}^{N}$, is used. Each FLGP individual/program, as a solution to feature extraction, transforms each image I_i to a vector F^i with the size of S. S is the number of the extracted features by the FLGP program. Then $\{F^i\}_{i=1}^{N}$ are normalized and fed into a linear SVM together with $\{Y_i\}_{i=1}^{N}$ to perform classification. The linear SVM is employed because it is popular for image classification [8] and has fewer parameters compared with SVMs with other kernel functions. On image classification tasks, the classification accuracy is the most commonly used fitness function [8, 24]. To increase the generalization ability, stratified *k*-fold cross-validation (CV) is used for evaluating each individual and the mean accuracy of the *k* folds is set as the fitness value. $k = 5$ and $k = 10$ are commonly used settings for k-fold CV. To reduce the computational cost, we set k to 5 in FLGP.

1) Normalization of Extracted Features: The previous GP-GLF [13] method does not perform feature normalization. However, the produced features by GP-GLF are the combination of features with different scales. For example, the uLBP features may be in the range of $[0,\ 100]$ and the DIF features are in the range of $[0,\ 1]$. This may lead to bias towards particular types of features such as uLBP features when using the combination of these features for classification. Therefore, in FLGP, the min-max normalization method is used to rescale the output features F_i to $\overline{F}_i$, as shown in Eq. 1.

$$\overline{F}^i = \frac{F^i - min(\{F^i\}_{i=1}^{N})}{max(\{F^i\}_{i=1}^{N}) - min(\{F^i\}_{i=1}^{N})}. \quad (1)$$

In addition to the above differences, the FLGP approach has a new representation (program structure), a new function set and a terminal set, which will be described below.

B. New Program Structure

The proposed FLGP approach is based on STGP so that a new program structure is needed. The new program structure is extended from that of GP-GLF in [13] by improving its flexibility. The program structure of GP-GLF has a representation with a fixed tree depth, which limits the type of output features, i.e., the combination of global and local features. The new program structure of FLGP addresses these limitations by using a flexible structure to represent more possible ways of combining global and local features. The new program structure allows each solution to have a flexible tree depth and to produce various numbers of global and/or local features F_S.

The new program structure is shown in Fig. 3 (a). It contains the tiers of input, region detection, feature extraction, feature concatenation, and output, where different functions are used at different functional tiers. Region detection tier aims to detect small regions of interest from an input image. The region detection tier may exist or not in FLGP trees, which indicates that the solutions of FLGP can be constructed without any region detection. This allows FLGP to produce only global features. Feature extraction tier extracts global features from an input image or local features from the detected regions. The feature extraction functions are developed in the global and local scenarios. Feature concatenation tier concatenates the features from its child nodes to a feature vector. To further demonstrate the program structure, a typical example program/solution of FLGP is shown in Fig. 3 (b), where different colors indicate inputs, outputs and different functions. In this program, there are region detection functions, $Region\_S$ and $Region\_R$, feature extraction functions, G_SIFT, L_DIF, L_uLBP, and feature concatenation function, $FeaCon2$.

The new program structure allows FLGP to produce three types of features. The first type is a combination of global and local features. As shown in the example program in Fig. 3(b), the output features are a combination of global SIFT features, local DIF features and local uLBP features. The total number of the output features S equals $s_1 + s_2 + s_3$, where s_1 is the number of the SIFT features, s_2 is the number of the DIF features, and s_3 is the number of the uLBP features. These features are extracted by the G_SIFT, L_DIF and L_uLBP functions, respectively. The second type is a combination of local features. This can be achieved by building an FLGP tree where each input image (the $Image$ terminal) must connect with the region detection function, as shown in Fig. 4 (a). The third type is a combination of global features, which can be achieved by building an FLGP tree without region detection function, as shown in Fig. 4 (b).

C. Function Set

According to the new program structure, the function set of FLGP has three different types of functions: region detection functions, feature extraction functions and feature concatenation functions.

*1) **Region Detection Functions**:* The region detection functions are $Region\_S$ and $Region\_R$, which detect square and rectangle regions from an image, respectively. The $Region\_S$ function takes an image $Image$ (I_i, the size is $m \times l$), X, Y, and $Size$ as inputs and returns a square region. The coordination of the top-left point of the region in $Image$ is $(X,\ Y)$ and the size of the region is $Size$. Thus the detected region by $Region\_S$ is $Image[X : min((X+Size),\ m),\ Y :$

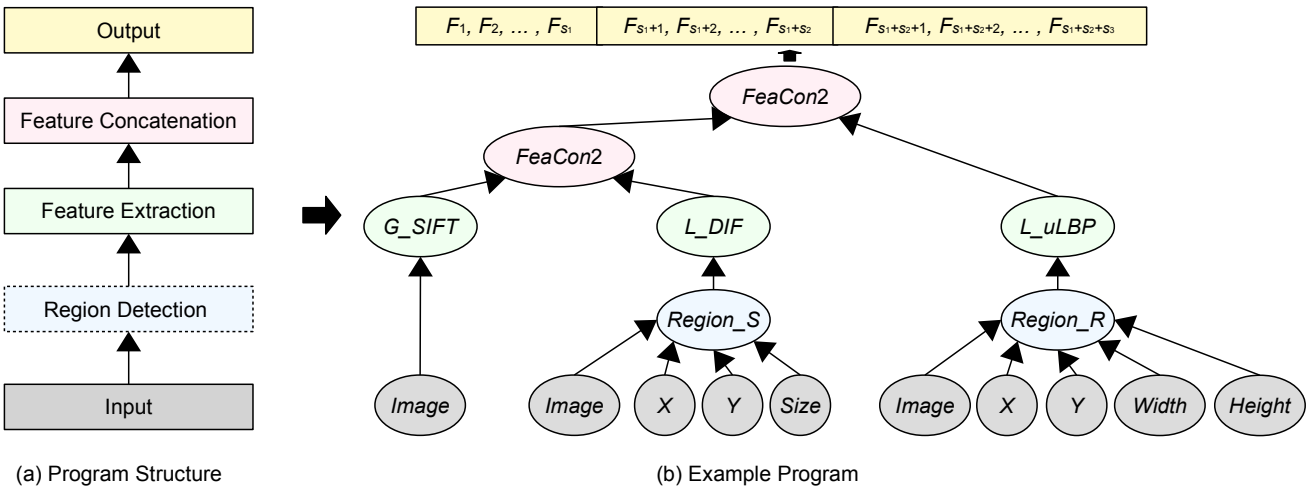

Fig. 3. The new program structure of FLGP and an example program that describes a combination of global and local features.

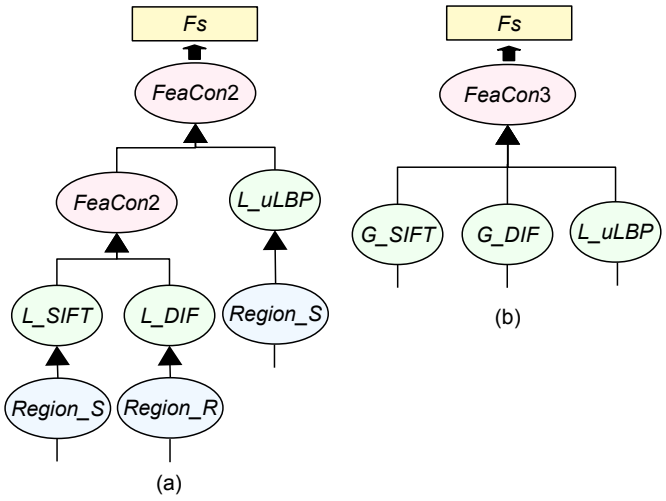

Fig. 4. Two example program structures to describe (a) a combination of local features, and (b) a combination of global features.

$min((Y + Size),\ l)]$. Similar to $Region\_S$, the $Region\_R$ function detects the $Image[X : min((X + Width),\ m),\ Y : min((Y + Height),\ l)]$ region by taking the $Image$, X, Y, $Width$, and $Height$ as inputs. In the two functions, the $Image$, X, Y, $Size$, $Width$, and $Height$ are terminals, which will be described in the next subsection. The values of these terminals are randomly generated from pre-defined ranges and can be changed by the mutation operator during the evolutionary learning process.

2) ***Feature Extraction Functions****:* GP-GLF [13] uses seven descriptors, including GLCM and Gabor features. Since GLCM, uLBP or Gabor is able to describe texture features, FLGP only uses uLBP instead of the three texture descriptors. Therefore, five representative descriptors: DIF, Hist, SIFT, HOG, and uLBP, are used in FLGP as feature extraction functions to avoid a big search space. The five descriptors have been introduced in Subsection II-A and are representative methods to describe distribution, texture, shape, and appearance information of images. In FLGP, the five methods are developed in global and local scenarios. The methods in the global scenario are G_DIF, G_Hist, G_SIFT, G_HOG, and G_uLBP, which extract features from a whole image. The methods in the local scenario are L_DIF, L_Hist, L_SIFT, L_HOG, and L_uLBP, which extract local features from a detected region. The functions in the global scenario directly use the $Image$ terminal as their children node, while the functions in the local scenario employ the region detection functions as their children nodes. Each feature extraction function transforms an image or a region into a set of features F_s, where the number of features is s. The details of these functions are listed in Table I. It is obvious that each function extracts a different number (s) of features from an image/region, as shown in the fourth column of Table I.

3) ***Feature Concatenation Functions****:* The feature concatenation functions are $FeaCon2$ and $FeaCon3$, which concatenate two feature vectors (F_{s_1} and F_{s_2}) and three feature vectors (F_{s_1}, F_{s_2} and F_{s_3}) to a feature vector F_S respectively. The children nodes of the two functions can be the feature extraction functions and/or the feature concatenation functions. This allows FLGP to produce various numbers of features.

D. Terminal Set

The new terminal set has six different terminals: $Image$, X, Y, $Size$, $Width$, and $Height$. The $Image$ terminal indicates the input gray-scale image, which is a two-dimension array ($m \times l$) with values in the range of $[0,\ 1]$ as the image is normalized by dividing 255. The other terminals are ephemeral random constants of FLGP. X and Y indicate the coordinates of the top-left point of a detected region in the image and are the parameters of the $Region\_S$ and $Region\_R$ functions. They are integers in the range of $[0,\ m-20]$ and $[0,\ l-20]$, respectively. The $Size$, $Width$ and $Height$ terminals are the size or width and height of a detected region. Their values are in the range of $[20,\ 50]$, which is smaller than that in [13] to narrow the search space.

TABLE I
FEATURE EXTRACTION FUNCTIONS.

Methods	**Input**	**Output**	**#Features** s	**Description**
G_DIF/L_DIF	1 Image/Region	1 Vector	20	Domain independent features [15].
G_Hist/L_Hist	1 Image/Region	1 Vector	32	Histogram features of the image/region [14]. The number of bins is set to 32.
G_SIFT/L_SIFT	1 Image/Region	1 Vector	128	SIFT features. The image or detected region is considered as a keypoint [16].
G_HOG/L_HOG	1 Image/Region	1 Vector	Flexible	HOG features [5]. G_HOG/L_HOG extracts the mean value of each 20×20 / 10×10 grid with a step of 10 from a HOG image.
G_uLBP/L_uLBP	1 Image/Region	1 Vector	59	Uniform LBP histogram features [7]. In G_uLBP and L_uLBP, the radius is 1.5 and the number of neighbors is 8.

IV. EXPERIMENT DESIGN

A number of experiments have been conducted to evaluate the performance of FLGP for feature learning in image classification. The experiments aim to investigate whether FLGP can achieve better performance than existing GP-based methods, CNN-based methods and traditional methods using various features. This section describes the design of the experiments.

A. Data Sets

Eight different data sets of varying difficulty are used in the experiments to examine the effectiveness of FLGP. The data sets contain five types of tasks, i.e., facial expression classification (FEI_1 [27], FEI_2 [27] and JAFFE [28]), object classification (EYALE [29] and ORL [30]), scene classification (SCENE [31]), texture classification (KTH [32]), and painting classification (VGDB [33]).

FEI_1 and FEI_2 [27] contain frontal facial images with natural or smile expression. The images in the two data sets are sampled from 200 Brazilian with different appearance, hairstyle and adorn. VGDB is to identify Vincent Van Gogh's paintings [33], which is very challenging because there are not particular objects in the images and the painting style is hard to capture. ORL [30] is to recognize faces of 40 different people from images with open or closed eyes, smiling or non-smiling, and glasses or non-glasses. JAFFE [28] has 213 images of 7 different expressions sampled from 10 Japanese females. The seven expressions are happiness, surprise, sadness, fear, anger, natural, and disgust. KTH [32] is a texture classification task of 10 classes. The images are sampled in nine scales with three poses under four illumination conditions. EYALE [29] is a face classification task, having 2424 facial images of 38 different people. The facial images are sampled under different poses and illumination conditions. SCENE [31] contains 3859 natural images in 13 groups, including the coast, forest, highway, mountain, and street. The natural images are acquired under different conditions and have high variations, which makes the task difficult.

Table II describes the details of these data sets. These data sets are split into the training and test sets according to commonly used proportions. For the FEI_1, FEI_2, VGDB, and KTH data sets, 75% images are used for training and 25% images are used for testing. For ORL, seven images per class are used for training and the remaining images are used for testing. For JAFFE, 20 images per class are used for training and the others are for testing. Since the EYALE and SCENE data sets are large, they are split into 50% and 50% to form the training set and the test set, respectively. Fig. 5 and Fig. 6 show several example images from the eight data sets.

TABLE II
DATA SET PROPERTIES.

Data set	Image size	# Classes	Training set	Test set
FEI_1	130×180	2	150	50
FEI_2	130×180	2	150	50
VGDB	200×200	2	247	83
ORL	112×92	40	280	120
JAFFE	128×128	7	140	73
KTH	100×100	10	600	210
EYALE	100×100	38	1,209	1,215
SCENE	100×100	13	1,928	1,931

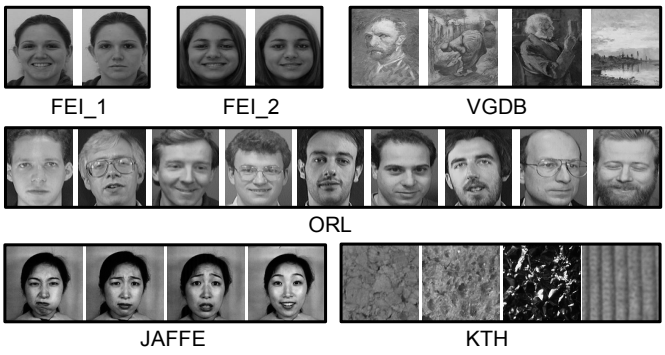

Fig. 5. Example images from the FEI_1, FEI_2, VGDB, ORL, JAFFE, and KTH data sets.

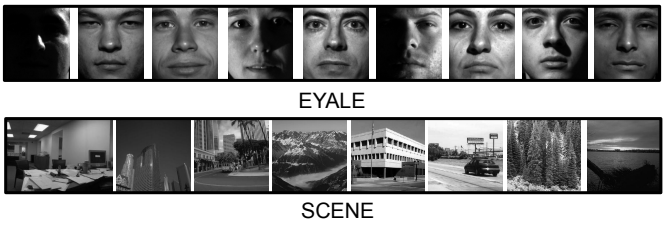

Fig. 6. Example images from the EYALE and SCENE data sets.

B. Benchmark Methods

The benchmark methods are five GP-based methods, eight traditional methods using different features, and three CNNs.

1) GP-based Methods: Five GP-based methods are GP-GLF [13], 2TGP [34], DIF+GP [15], Hist+GP, and uLBP+GP [21]. Since the new FLGP approach is an extension of GP-GLF, it is necessary to compare FLGP with GP-GLF. The 2TGP method automatically generates a high-level feature from the input image with simultaneous region detection, feature extraction and feature construction. DIF+GP, Hist+GP and uLBP+GP construct features for classification from pre-extracted features, i.e., 20 DIF features, 64 Hist features and 59 uLBP features, respectively. Since the 2TGP, DIF+GP, Hist+GP, and uLBP+GP methods are originally designed for binary classification, they are only used for comparisons on binary image classification tasks, i.e., FEI_1, FEI_2 and VGDB. Because of the high computational cost of GP-GLF, it is too expensive to run it on the multi-class classification tasks (probably needs several months to obtain all the results). Therefore, the comparisons of FLGP and GP-based methods are only on the binary image classification tasks.

2) Traditional Methods: Eight traditional methods use different well-known features are used for comparisons. The features are DIF [15], Hist [14], GLCM [17], Gabor [35], SIFT [6], HOG [5], LBP [7], and uLBP [7] features. These features are fed into a linear SVM for classification. The methods for extracting these features have been introduced in Section II. The DIF, SIFT, HOG, and ulBP features are extracted using the same functions as those employed in the function set of FLGP in the global scenario. The Hist features are 256 histogram features and the LBP features are 256 LBP histogram features. The GLCM features are the statistics of each GLCM, i.e., contrast, dissimilarity, homogeneity, energy, correlation, and ASM. Each GLCM is calculated using four different orientations ($\pi\mu/4,\ \mu \in \{0,1,2,3\}$). The Gobor features are the mean values of each 32×32 grid of the convolved images using different Gabor filters. Forty commonly used Gabor filters are used, involving eight different orientations ($\pi\mu/8,\ \mu \in \{0,\ldots,7\}$) at five scales ($\nu \in \{0,\ldots,4\}$) [35].

3) CNN-based Methods: CNNs are well-known for feature learning and image classification. Three CNN methods with different architectures are employed for comparison. They are the LeNet-5 [36], a five-layer CNN (CNN-5) [8] and an eight-layer CNN (CNN-8). The three methods use the popular rectified linear unit (ReLU) as the activation function and the softmax for classification. Dropout is added after the pooling layer and the first fully connected layer with 0.25 and 0.5 probabilities to avoid overfitting [37]. In the three methods, the loss function is cross-entropy and the adaptive subgradient method is used to train the models [38]. The number of epochs is set to 500, which allows the three methods to be fully trained on these data sets.

C. Parameter Settings

Parameter settings for the proposed FLGP approach are the commonly used settings in the community of GP, which is also the same as that in the GP-GLF method [13]. Note that we aim to develop a general method that with common settings can achieve a good performance on a variety of image classification tasks. Therefore, parameter tuning is not conducted for FLGP, although it could improve the performance. The other four GP-based benchmark methods use the same parameter settings as FLGP except for the population size. The population size for the four GP-based methods is 500, while FLGP and GP-GLF use a smaller size of 100 in order to reduce the computational cost. The crossover, mutation, and elitism rates are 0.8, 0.19, and 0.01, respectively. The selection method is Tournament selection with size 7. The tree generation method is ramped-half-and-half. The tree depth is between 2 and 6. The termination criterion for all the GP methods is reaching the maximum number of generations.

D. Test Process and Experiment Settings

The overall test process of FLGP and the eight traditional methods are the same. The FLGP solution is used as a feature extraction/description method to transform the training set and the test set. Before feeding the data sets to SVM, the min-max normalization is conducted on the transformed data set. Note that the normalization for the test set is based on the min and max values of features in the training set. A linear SVM is employed to learn a classifier using the transformed and normalized training set and the classifier is tested on the transformed and normalized test set. Finally, the classification accuracy of the test set is reported.

The implementations of all the GP-based methods are based on the *DEAP* (*Distributed Evolutionary Algorithm in Python*) [39] package and the implementation of the linear SVM is based on the *scikit-learn* [40] package with default parameter settings. In SVM, the penalty parameter (C) is 1. The experiments of GP-based methods and traditional methods run independent 30 times and the mean accuracy of the 30 runs is reported. The experiments of the three CNNs run independent 10 times due to the high computational cost.

V. Results and Discussions

This section discusses and compares the classification results of the proposed FLGP approach, the five GP-based methods, the eight traditional methods, and the three CNNs on the eight data sets. The classification results are listed in Tables III and IV. The results include maximum accuracy (Max), mean accuracy and standard deviation (Mean $\pm$ Std.). Wilcoxon rank-sum test with a 5% significance level is used to compare FLGP with a benchmark method to show the

significance of performance improvement. The symbols "+" or "–" in these tables indicate that FLGP is significantly better or significantly worse than the compared method. The symbol "=" indicates the performance of FLGP is similar to the compared method. In Tables III and IV, each small block lists the results on one data set, and the maximum classification accuracy is highlighted in bold. The final row of each block summarizes the overall results of the significance test.

A. Overall Classification Performance

As mentioned in Section IV-B, the GP-GLF and the other four GP-based benchmark methods are only used for comparisons on binary classification tasks. Thus, there are 16 benchmark methods on FEI_1, FEI_2 and VGDB and 11 benchmark methods on the remaining data sets. From the final rows of Tables III and IV, it is obvious that FLGP achieves significantly better or similar performance in almost all the comparisons. Specifically, FLGP gains 97 "+", 5 "=" and 1 "–" in the total 103 comparisons. FLGP significantly outperforms all the benchmark methods on one binary classification data set, i.e., FEI_2, and on five multi-class classification data sets, i.e., ORL, JAFFE, KTH, EYALE, and SCENE. FLGP performs significantly better than or similar to any of the 16 benchmark methods on FEI_1.

FLGP gains the maximum accuracy and the maximum mean accuracy among all the methods on seven data sets except for VGDB. Specifically, FLGP improves the maximum accuracy by 11% on JAFFE, 8.1% on KTH, 7.8% on SCENE, 6% on FEI_2, 1.7% on ORL, and 0.5% on EYALE. FLGP improves the mean accuracy by 9% on SCENE, 6.9% on KTH, 2.5% on FEI_2, 2.2% on JAFFE, and 1.3% on ORL. From the results, it is clear that FLGP is more effective than any of the benchmark methods on different types of image classification tasks.

The experimental results demonstrate the effectiveness of FLGP for feature learning in image classification. The main reasons to explain why FLGP is effective are the developments of the feature learning process, the flexible program structure, the function set and the terminal set in FLGP. With the utilization of five representative feature descriptors in global and local scenarios, respectively, FLGP can extract high-level invariant features with the potential of increasing classification performance. The flexible program structure enables FLGP to effectively search for optimal functions and terminals to form solutions that can produce various numbers of global and/or local features. The overall feature learning process enables FLGP to find optimal solutions with high generalization ability.

B. FLGP vs Five GP-based Methods

1) FLGP vs GP-GLF: Table III shows that FLGP achieves significantly better performance than GP-GLF on the three binary classification data sets. Compared with GP-GLF, FLGP improves the mean accuracy by 6.7% on FEI_1, 10.8% on FEI_2 and 9.8% on VGDB. FLGP improves the maximum accuracy by 2% on FEI_1, 8% on FEI_2 and 7.3% on VGDB. From the results, it is clear that FLGP is more effective than GP-GLF for feature learning in image classification.

As mentioned in Section I, FLGP is developed by addressing the limitations of GP-GLF. The results show that this goal was successfully achieved. FLGP is more effective than GP-GLF by producing three types of features, i.e., a combination of global and local features, a combination of global features, a combination of local features. This allows FLGP to automatically find suitable types and numbers of features to improve the classification accuracy for a given task.

2) FLGP vs The Other Four GP-based Methods: From Table III, it is noticeable that FLGP achieves significantly better results in 11 comparisons and similar results in 1 comparison out of the total 12 comparisons. Importantly, compared with the four GP-based methods (2TGP, DIF+GP, Hist+GP, and uLBP+GP), FLGP improves the mean accuracy by over 7% on FEI_1 and FEI_2. Moreover, FLGP achieves the maximum accuracy on FEI_1 and FEI_2. On VGDB, FLGP obtains a similar performance to Hist+GP. Hist+GP uses 256 features while the G_Hist/L_Hist function in FLGP only extracts 32 features, which may be the reason that FLGP cannot achieve a performance as good as Hist+GP.

With automatically extracting a set of global and/or local features from raw pixels, FLGP achieves significantly better performance than DIF+GP, Hist+GP and uLBP+GP in most comparisons. The results show the potential of GP in feature learning from raw pixels. 2TGP can learn features from raw pixels. However, FLGP is more effective than 2TGP on the three data sets. 2TGP learns only one high-level feature from an image to perform classification. In contrast, FLGP learns a set of high-level features from an image, which is more effective for classification.

C. FLGP vs Eight Traditional Methods

It can be seen from Tables III and IV that FLGP achieves significantly better results in 62 comparisons out of the total 64 comparisons. Specifically, FLGP performs significantly better than any of the eight benchmark methods on seven data sets except for VGDB. Compared with DIF, Hist, GLCM, Gabor, SIFT, HOG, LBP, and uLBP, FLGP improves the mean accuracy by 10.8% on EYALE, 9.1% on SCENE, 7.1% on JAFFE, and 6.9% on KTH. Moreover, FLGP obtains the maximum accuracy on seven data sets except for VGDB. FLGP has an increase by over 10% in terms of the maximum accuracy on FEI_2, JAFFE, EYALE, and SCENE. The VGDB data set is the only one that FLGP performs worse than one of the eight traditional methods, i.e., LBP. FLGP does not use the LBP descriptor so that it cannot achieve a performance as good as the LBP features on VGDB.

The experimental results show that the features learned by FLGP are more effective than the well-known handcrafted features in the eight methods for image classification. Using traditional methods often requires domain expertise to extract a set of effective features for classification. The new FLGP approach can automatically learn features from images without domain knowledge. The design of FLGP enables it to automatically learn various numbers of features in three types, i.e., a combination of global and local features, a combination of global features and a combination of local

TABLE III
CLASSIFICATION ACCURACY (%) OF THE PROPOSED FLGP APPROACH AND SIXTEEN BENCHMARK METHODS ON THREE BINARY DATA SETS: FEI_1, FEI_2 AND VGDB.

	FEI_1		**FEI_2**		**VGDB**	
Methods	Max	Mean±Std.	Max	Mean±Std.	Max	Mean±Std.
2TGP	96.0	88.1±6.2+	94.0	85.5±6.0+	63.9	61.6±1.5+
DIF+GP	80.0	56.7±6.9+	72.0	60.3±8.4+	68.7	61.4±3.5+
Hist+GP	70.0	48.9±7.2+	60.0	48.8±6.1+	84.3	76.0±2.6=
uLBP+GP	66.0	50.9±7.5+	72.0	48.7±7.9+	79.5	69.4±4.4+
GP-GLF	96.0	89.1±3.9+	92.0	82.5±5.8+	74.6	65.1±4.7+
DIF	74.0	61.1±4.9+	72.0	62.8±6.1+	66.3	55.6±10.3+
Hist	54.0	48.1±3.4+	54.0	50.1±2.5+	62.7	62.2±0.8+
GLCM	50.0	49.7±0.7+	54.0	50.1±0.7+	62.7	53.3±9.8+
Gabor	82.0	71.6±7.9+	74.0	65.7±5.1+	63.9	56.0±8.3+
SIFT	82.0	82.0±0.0+	78.0	78.0±0.0+	60.2	60.2±0.0+
HOG	94.0	94.0±0.0+	88.0	88.0±0.0+	57.8	57.2±0.7+
LBP	68.0	62.5±3.5+	66.0	57.6±3.6+	84.3	80.6±3.2–
uLBP	64.0	56.9±5.2+	56.0	51.9±2.3+	81.9	71.5±8.1=
LeNet-5	98.0	94.4±2.0=	94.0	90.8±1.8+	65.1	58.1±4.8+
CNN-5	98.0	95.6±1.5=	90.0	85.0±3.0+	65.1	61.5±2.1+
CNN-8	98.0	94.2±2.1=	94.0	90.0±2.3+	61.5	56.9±4.6+
FLGP	98.0	95.8±3.2	100	93.3±3.8	81.9	74.9±3.7
Overall		**13+, 3=**		**16+**		**13+, 2=, 1–**

TABLE IV
CLASSIFICATION ACCURACY (%) OF THE PROPOSED FLGP APPROACH AND ELEVEN BENCHMARK METHODS ON FIVE MULTI-CLASS DATA SETS: ORL, JAFFE, KTH, EYALE, AND SCENE.

	ORL		**JAFFE**		**KTH**		**EYALE**		**SCENE**	
Methods	Max	Mean±Std.	Max	Mean±Std	Max	Mean±Std.	Max	Mean±Std.	Max	Mean±Std.
DIF	85.0	85.0±0.0+	35.6	35.6±0.0+	56.2	56.2±0.0+	26.4	26.4±0.0+	33.5	33.5±0.0+
Hist	97.5	97.5±0.0+	19.2	19.2±0.0+	51.4	51.4±0.0+	11.0	11.0±0.0+	21.2	21.2±0.0+
GLCM	2.5	2.5±0.0+	15.1	15.1±0.0+	23.3	23.3±0.0+	5.1	5.0±0.1+	13.9	13.9±0.0+
Gabor	59.2	57.1±0.9+	46.6	43.2±1.6+	44.3	42.9±0.7+	36.7	36.3±0.2+	22.7	22.3±0.2+
SIFT	98.3	98.3±0.0+	74.0	74.0±0.0+	81.4	81.4±0.0+	88.4	88.4±0.0+	63.1	63.1±0.0+
HOG	96.7	96.7±0.0+	72.6	72.6±0.0+	51.4	51.4±0.2+	74.3	74.3±0.0+	30.8	30.8±0.0+
LBP	87.5	87.5±0.0+	21.9	21.9±0.0+	87.6	87.6±0.0+	46.4	46.4±0.0+	62.5	62.5±0.0+
uLBP	94.2	94.2±0.0+	23.3	23.3±0.0+	81.0	81.0±0.0+	56.1	56.1±0.0+	66.1	66.1±0.0+
LeNet	93.3	89.9±1.9+	79.5	68.9±7.0+	78.6	72.0±6.4+	92.4	89.4±1.6+	54.2	51.1±2.4+
CNN-5	97.5	96.3±0.8+	80.8	78.9±1.3+	84.3	81.5±1.8+	99.3	98.6±0.5+	58.9	55.4±1.4+
CNN-8	96.7	94.2±1.8+	61.6	52.5±6.2+	82.4	80.7±1.5+	90.9	88.2±1.0+	69.2	66.2±2.0+
FLGP	100	99.6±0.7	91.8	81.1±4.7	95.7	94.5±0.9	99.8	99.2±0.4	77.0	75.2±0.7
Overall		**11+**		**11+**		**11+**		**11+**		**11+**

features. The features learned by FLGP are more effective for image classification than the manually extracted features.

D. FLGP vs Three CNN-based Methods

Compared with LeNet, CNN-5 and CNN-8, FLGP achieves significantly better performance on seven data sets and similar performance on the remaining one, FEI_1. Importantly, FLGP improves the mean accuracy by over 13% on VGDB and KTH, and by over 9% on SCENE compared with the three CNNs. Surprisingly, CNN-8 performs worse than CNN-5 on six data sets except for FEI_2 and SCENE, which indicates that an increase in the depth of CNNs cannot guarantee an increase in classification accuracy. A more complex model may require more training instances/samples in order to obtain satisfactory results. Compared with the three methods with pre-defined model complexity, the flexible representation allows FLGP to evolve solutions of variable depths, which is more flexible for solving different image classification tasks.

The results indicate that the features learned by FLGP are more effective than those by the three CNNs with different architectures for image classification. Compared with the three

CNNs, FLGP uses a simpler program structure and a set of functions and terminals, but it achieves better performance on different types of data sets. This main reason is that FLGP learns various numbers and types of features, which is more flexible than the three CNNs. FLGP can learn not only global features but also local features from automatically detected regions. However, it may be difficult for the three CNNs to learn effective local features from the whole input image.

VI. Further Analysis

This section further compares FLGP with GP-GLF on the computational cost. Then it deeply analyzes the best solutions evolved by FLGP to fully understand why it achieves good performance.

A. FLGP vs GP-GLF on Computational Cost

The comparisons of FLGP and GP-GLF in training time and testing time on each of the three data sets are shown in Fig.7. It is obvious that FLGP is much faster than GP-GLF in both training and testing. FLGP uses less than 6 hours for training on each data set, while GP-GLF uses more than 8 hours on FEI_1 and FEI_2 and more than 80 hours on VGDB for training. The VGDB data set is very challenging so that GP-GLF uses over 80 hours to find an optimal solution. FLGP significantly reduces the computational cost. The testing time of FLGP and GP-GLF on FEI_1 and FEI_2 are within 0.2 minutes. The testing time of GP-GLF on VGDB is over 0.4 minutes, while FLGP needs less than 0.1 minutes for testing. The comparisons demonstrate that FLGP is faster than the previous GP-GLF method in [13]. FLGP has a more flexible program structure and a smaller function set than GP-GLF, which improve FLGP's search efficiency and reduce the complexity of the evolved solutions.

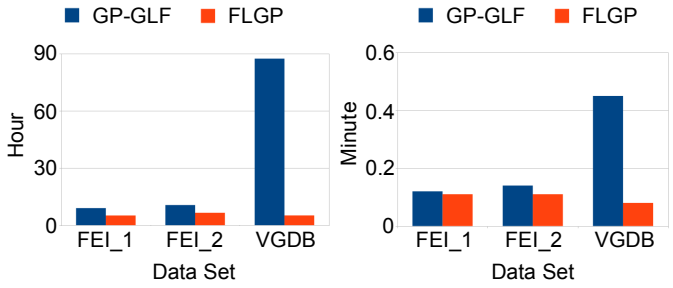

Fig. 7. The training (left) and testing (right) time of FLGP and GP-GLF on the FEI_1, FEI_2 and VGDB data sets (Note that the GP-GLF method has only been examined on binary classification data sets).

B. Analysis on Example Solutions of FLGP

1) An Example Solution on FEI_2: An example solution of FLGP on the FEI_2 data set is visualized in Fig. 8. This example solution achieves 100% classification accuracy on both the training and test sets. Two example images from the two classes (smile and natural) are used for visualization to show what and how features are extracted. This solution detects a 50×50 region using $Region\_S$ and a 38×46 rectangle region using $Region\_R$ from the input image. From each detected region, it extracts 128 SIFT features using L_SIFT. Together with the extracted 20 DIF features from the whole image by G_DIF, the solution is able to produce 276 features from an input image.

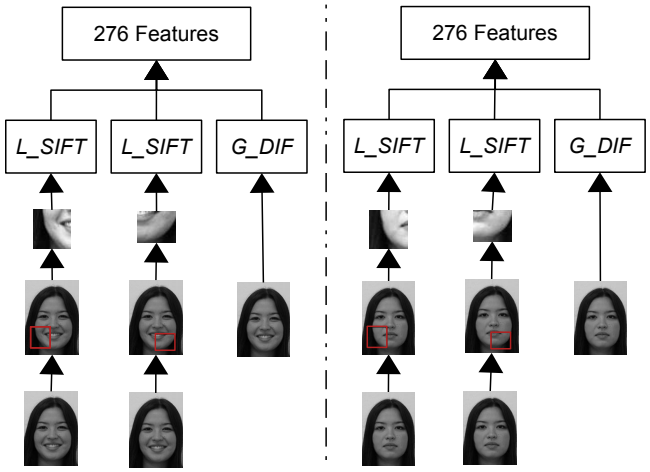

Fig. 8. An example of best-of-the-run solution evolved by the proposed FLGP approach on the FEI_2 data set and two example images are used as the inputs of the example solution to show the process of feature extraction.

From Fig. 8, it can be seen that the $Region\_S$ function detects the chin and mouth area of the left face and the $Region\_S$ function detects a similar area of the right face. It is obvious that the two regions capture the most discriminative information between the two different classes. For example, the detected area contains the teeth in the happy face image but not in the natural face image and captures the difference in mouth shapes between the two expressions.

This example solution finds a combination of local SIFT features and global DIF features for classification. The combined features are more effective for classification than the individual global DIF and SIFT features. The traditional method using DIF features only achieves a maximum accuracy of 72% and the method using SIFT features only achieves a maximum accuracy of 78% on FEI_2. The example solution improves the classification performance by detecting the regions of interest and extracting meaningful local features from the detected regions. The analysis shows that FLGP detects informative regions and extracts discriminative local and global features for classification.

2) Analysis on the Feature Extraction Functions: To further analyze FLGP, ten best FLGP solutions of each run (totally 300 solutions) on each data set are recorded. The frequency of the ten feature extraction functions in the global (G_DIF, G_Hist, G_SIFT, G_HOG and G_uLBP) and local (L_DIF, L_Hist, L_SIFT, L_HOG, and L_uLBP) scenarios in these solutions are ranked. The results of the ranking are listed in Table V, where 1 indicates the most frequently used function and 10 indicates the least frequently used function in these solutions. From Table V, it is clear that the frequency-rank of the feature extraction functions

TABLE V
RANKING OF ALL THE FEATURE EXTRACTION FUNCTIONS IN 300 BEST-OF-THE-RUN PROGRAMS OF FLGP ON EACH DATA SET.

Function	FEI_1	FEI_2	VGDB	ORL	JAFFE	KTH	EYALE	SCENE
G_DIF	5	9	7	7	7	5	8	4
G_Hist	10	10	3	5	9	6	5	6
G_SIFT	8	5	8	1	2	2	3	2
G_HOG	3	4	10	4	5	7	9	9
G_uLBP	9	7	1	2	10	1	4	1
L_DIF	6	6	5	9	3	4	7	7
L_Hist	4	8	2	6	8	3	6	8
L_SIFT	1	1	9	10	1	10	1	10
L_HOG	7	3	4	8	4	9	10	5
L_uLBP	2	2	6	3	6	8	2	3

varies with the data sets. Specifically, the L_SIFT function is the most frequently used on FEI_1, FEI_2, JAFFE, and EYALE, the G_uLBP function is the most frequently used on VGDB, KTH and SCENE, and the G_SIFT function is the most frequently used on ORL. Moreover, the frequently used functions on one data set may be less frequently used on the other data sets. For example, L_SIFT is the most frequently used on four data sets but it is the least frequently used on ORL, KTH and SCENE. The G_Hist function is the least frequently used on FEI_1, FEI_2 and JAFFE, but it is frequently used on VGDB. This confirms the difficulty of feature extraction as different data sets need various types of features. In contrast, FLGP automatically finds the best feature extraction methods or combinations of them to extract features.

It can be seen from Table V that FLGP learns more local features than global features on face data sets, i.e., FEI_1, FEI_2, JAFFE, and EYALE, which confirms that local features are more effective for object classification. In contrast, FLGP learns more global features than local features on the non-object data sets, i.e., VGDB, KTH and SCENE, as global features are more effective. The analysis shows that FLGP learns the best feature extraction functions or combinations of them to extract effective global and/or local features for image classification.

VII. CONCLUSIONS

The aim of this paper was to develop an effective GP-based approach that uses existing well-developed feature descriptors to learn global and/or local features for different image classification tasks. This goal was successfully achieved by proposing the FLGP approach with a new program structure, a new function set and a new terminal set and examining FLGP on eight different image data sets of varying difficulty. Specifically, five existing feature descriptors, i.e., Hist, DIF, SIFT, HOG, and uLBP, were employed in FLGP as functions in the global and local scenarios. FLGP was compared with a number of benchmark methods to show its effectiveness. Example solutions of FLGP were visualized and analyzed to show the good interpretability of the solutions.

Experimental results demonstrated that FLGP achieved significantly better performance in almost all the comparisons on different image classification tasks. The results confirmed the capability of FLGP on automatically learning effective global and/or local features for achieving high classification accuracy. The comparisons of FLGP and GP-GLF showed that FLGP significantly improved the classification performance by using a flexible program structure, a new function set and a new feature learning process. The comparisons of FLGP and the other four GP-based methods suggested that it is more effective to automatically learn a set of high-level features from raw pixels than to construct one high-level feature for image classification. Compared with the eight traditional methods, FLGP achieved better classification accuracy by automatically learning various types and numbers of global and/or local features. The comparisons of FLGP and the three CNN methods demonstrated that FLGP was more effective by evolving solutions of variable lengths. Furthermore, FLGP can learn discriminative local features, which may be difficult to achieve using the three CNN methods. Further analysis of computational cost demonstrated that FLGP was faster than GP-GLF in both training and testing. The analysis of the example solutions of FLGP confirmed the good interpretability of the solutions and revealed that FLGP learned discriminative features using a simple solution. The analysis also revealed that FLGP detected informative regions and found the most effective feature extraction functions to extract features from the regions/images.

This paper has shown the potential of GP in feature learning for image classification. The FLGP approach learns features without human intervention and provides solutions with high interpretability. In the future, we will develop a new GP approach with deep structures for image classification on more difficult data sets, such as ImageNet.

ACKNOWLEDGEMENT

This work was supported in part by the Marsden Fund of New Zealand Government under Contracts VUW1509 and VUW1615, and the University Research Fund at Victoria University of Wellington 209862/3580, and 213150/3662. This work of Ying Bi was supported by China Scholarship Council (CSC)/Victoria University Scholarship.

REFERENCES

[1] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *Int. J. Remote Sens.*, vol. 28, no. 5, pp. 823–870, Mar. 2007.

[2] E. Hjelmås and B. K. Low, "Face detection: A survey," *Comput. Vis. Image Und.*, vol. 83, no. 3, pp. 236–274, Sep. 2001.

[3] S. Jia, L. Shen, J. Zhu, and Q. Li, "A 3-d gabor phase-based coding and matching framework for hyperspectral imagery classification," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1176–1188, Apr. 2018.

[4] W. A. Albukhanajer, J. A. Briffa, and Y. Jin, "Evolutionary multiobjective image feature extraction in the presence of noise," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1757–1768, Sep. 2015.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, San Diego, USA, Jun., 2005, pp. 886–893.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[7] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[8] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

[9] L. Liu, L. Shao, X. Li, and K. Lu, "Learning spatio-temporal representations for action recognition: A genetic programming approach," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 158–170, Jan. 2016.

[10] H. Al-Sahaf, A. Al-Sahaf, B. Xue, M. Johnston, and M. Zhang, "Automatically evolving rotation-invariant texture image descriptors by genetic programming," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 83–101, Feb. 2017.

[11] Y. Bi, B. Xue, and M. Zhang, "An automatic feature extraction approach to image classification using genetic programming," in *Proc. Int. Conf. Appl. Eov. Comput.*, Parma, Italy, 4-6 Apr., 2018, pp. 421–438.

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

[13] Y. Bi, B. Xue, and M. Zhang, "Genetic programming for automatic global and local feature extraction to image classification," in *Proc. IEEE Congr. Evol. Comput.*, Rio de Janeiro, Brazil, 8-13 Jul., 2018, pp. 1–8.

[14] A. I. Awad and M. Hassaballah, "Image feature detectors and descriptors," *Stud. Comput. Intell.*, Springer International Publishing, Cham. 2016.

[15] M. Zhang, V. B. Ciesielski, and P. Andreae, "A domain-independent window approach to multiclass object detection using genetic programming," *EURASIP J. Adv. Sig. Pr.*, vol. 2003, no. 8, pp. 841–859, Dec. 2003.

[16] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proc. 18th ACM Int. Conf. Multimedia*, Firenze, Italy, 25-29 Oct., 2010, pp. 1469–1472.

[17] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[18] R. Poli, W. B. Langdon, and N. F. McPhee, *A field guide to genetic programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008, (With contributions by J. R. Koza).

[19] D. J. Montana, "Strongly typed genetic programming," *Evol. Comput.*, vol. 3, no. 2, pp. 199–230, 1995.

[20] R. Nandi, A. K. Nandi, R. M. Rangayyan, and D. Scutt, "Classification of breast masses in mammograms using genetic programming and feature selection," *Med. Biol. Eng. Comput.*, vol. 44, no. 8, pp. 683–694, Aug. 2006.

[21] Q. U. Ain, B. Xue, H. Al-Sahaf, and M. Zhang, "Genetic programming for skin cancer detection in dermoscopic images," in *Proc. IEEE Congr. Evol. Comput.*, San Sebastian, Spain, 5-8 Jun., 2017, pp. 2420–2427.

[22] W.-J. Choi and T.-S. Choi, "Genetic programming-based feature transform and classification for the automatic detection of pulmonary nodules on computed tomography images," *Inf. Sci.*, vol. 212, pp. 57–78, Dec. 2012.

[23] D. Atkins, K. Neshatian, and M. Zhang, "A domain independent genetic programming approach to automatic feature extraction for image classification," in *Proc. IEEE Congr. Evol. Comput.*, New Orleans, LA, USA, 5-8 Jun., 2011, pp. 238–245.

[24] 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 *Proc. Europ. Conf. Genet. Program.* Porto, Portugal: Springer, 30 Mar.-1 Apr., 2016, pp. 51–67.

[25] 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 Trans. Evol. Comput.*, vol. 21, no. 6, pp. 825 – 844, Dec. 2017.

[26] M. Iqbal, B. Xue, H. Al-Sahaf, and M. Zhang, "Cross-domain reuse of extracted knowledge in genetic programming for image classification," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 569–587, Aug. 2017.

[27] C. E. Thomaz, "Fei face database," *online: http://fei.edu.br/˜cet/facedatabase.html*, Mar. 2012.

[28] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Third IEEE Int. Conf. Auto. Face Gest. Recog.*, Nara, Japan, 14-16 Apr., 1998, pp. 200–205.

[29] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 5, pp. 684–698, May 2005.

[30] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. Sec. IEEE Workshop Appl. Comput. Vis.*, Sarasota, FL, USA, 5-7 Dec., 1994, pp. 138–142.

[31] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, San Diego, CA, USA, 20-25 Jun., 2005, pp. 524–531.

[32] 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, Stockholm, Sweden*, pp. 1–10, Jun. 2006.

[33] G. Folego, O. Gomes, and A. Rocha, "From impressionism to expressionism: Automatically identifying van gogh's paintings," in *IEEE Int. Conf. Image Process.*, Phoenix, AZ, USA, 25-28 Sep., 2016, pp. 141–145.

[34] H. Al-Sahaf, A. Song, K. Neshatian, and M. Zhang, "Two-tier genetic programming: Towards raw pixel-based image classification," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12 291–12 301, Nov. 2012.

[35] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, Apr. 2002.

[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[38] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[39] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *J. Mach. Learn. Res.*, vol. 13, pp. 2171–2175, Jul. 2012.

[40] 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," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.