



An Automatic Feature Extraction Approach to Image Classification Using Genetic Programming

Ying Bi^(✉), Bing Xue, and Mengjie Zhang

Evolutionary Computation Research Group,
School of Engineering and Computer Science,
Victoria University of Wellington, Wellington 6140, New Zealand
{Ying.Bi,Bing.Xue,Mengjie.Zhang}@ecs.vuw.ac.nz

Abstract. Feature extraction is an essential process for image data dimensionality reduction and classification. However, feature extraction is very difficult and often requires human intervention. Genetic Programming (GP) can achieve automatic feature extraction and image classification but the majority of existing methods extract low-level features from raw images without any image-related operations. Furthermore, the work on the combination of image-related operators/descriptors in GP for feature extraction and image classification is limited. This paper proposes a multi-layer GP approach (MLGP) to performing automatic high-level feature extraction and classification. A new program structure, a new function set including a number of image operators/descriptors and two region detectors, and a new terminal set are designed in this approach. The performance of the proposed method is examined on six different data sets of varying difficulty and compared with five GP based methods and 42 traditional image classification methods. Experimental results show that the proposed method achieves better or comparable performance than these baseline methods. Further analysis on the example programs evolved by the proposed MLGP method reveals the good interpretability of MLGP and gives insight into how this method can effectively extract high-level features for image classification.

Keywords: Genetic Programming · Image classification
Feature extraction · Image analysis

1 Introduction

Image classification is an important task in computer vision and pattern recognition with a wide range of applications such as image database annotation, image retrieval and video annotation [1]. Image classification can be defined as categorising an image into different predefined groups based on the content of the image. Although a number of techniques have been proposed to find solutions to this task [1], image classification is still an open issue due to the large variations in images, which needs further investigation.

Feature extraction is a key component of image classification. It can reduce the high dimensionality of the image data. The presences of stable and representative image features have positive effects on the performance of the classification system. A number of approaches such as the Grey-Level Co-occurrence Matrix (GLCM) [2], Local Binary Patterns (LBP) [3], Histogram of Orientated Gradients (HOG) [4], and Scale Invariant Feature Transform (SIFT) [5] are designed to extract features from the whole image or keypoints. However, it is very difficult to design an effective feature extraction method among these methods when dealing with a specific tasks. Generally, image domain experts and human intervention are required for image feature extraction, which needs cost and time to find [6].

Evolutionary computation (EC) techniques have a big potential to find the best solution from a set of solutions through a number of iterations/generations for a particular problem without domain knowledge and human intervention. Among the EC techniques, Genetic Programming (GP) is the most widely used technique on image analysis [7]. In the past decade, GP has been successfully applied to image classification, feature extraction, image segmentation, object detection, image registration and so on [8,9]. The flexible representation and the good ability of handling different data types allow GP to easily perform a particular image task by using image-related operators in its function set. For example, GP can evolve effective edge detectors based on Gaussian-based filters [10]. However, the existing work on using image-related operators/descriptors in GP for feature extraction and image classification is limited. Therefore, this work attempts to develop a GP approach to feature extraction and image classification which can benefit from the prior designed image-related operators/descriptors.

In [11–14], GP has been successfully applied to achieve automatic region detection, feature extraction, feature construction, and image classification simultaneously. However, there are a variety of image operators such as Gaussian filter, Histogram Equalisation, Sobel edge detector, Laplacian, which are more advanced for facilitating feature extraction than the simple pixel statistic feature extraction approaches in [11,12]. These operators can reduce noise, increase contrast or detect edges of an image, which are helpful for improving the quality of image data or finding more distinctive features such as edges from an image. The existing HOG and LBP image descriptors also have good ability for describing specific image features including shape and texture. Rather than only using HOG in [14], employing a set of image operators/descriptors in GP allows it to evolve high-level image features according to the data set it is trained on.

1.1 Goals

The overall goal of this paper is to develop a GP approach to achieving automatic region detection and feature extraction for effective image classification. This approach aims at integrating a set of image-related operators/descriptors in GP to detect more informative high-level image features for image classification. To achieve this, we propose a new MLGP approach, where a new program representation is designed, a set of operators including image operators and region detectors, and a new terminal set are developed. The new method will

be examined and compared with five other GP-based methods and 42 non-GP methods on six different image data sets of varying difficulty. Specifically, the overall goal can be divided into the following four objectives.

- (1) Develop a new program representation in GP which can integrate region detection, feature extraction, feature construction, and classification to a single solution/tree;
- (2) Develop a new function set and a new terminal set which allow GP to benefit from the combination of image-related operators/descriptors and produce high-level features with the potential of achieving good classification performance;
- (3) Investigate whether the proposed method can outperform the other five GP methods and 42 non-GP methods; and
- (4) Analyse the example trees with high performance to understand how the high-level features are extracted from the detected regions and further be constructed for effective classification.

2 Related Work

Zhang and Ciesielski [15] proposed a domain independent image feature extraction method (simplified as FeEx in this paper) and employed GP to evolve classifiers based on these features for object detection. The evolved classifier was used for classifying pixels into object or non-object groups. Compared to neural networks, the GP method achieved better detection rate and false alarm rate. Nandi *et al.* [16] introduced GP for classifying breast mass into the benign and malignant categories based on the selected texture features. The GP approach has shown promising results in classification. However, this approach requires domain experts to identify regions of interest and to extract image features. The other similar work on medical image classification can be found in [17, 18]. Human intervention and domain experts are required to extract image features when using GP to evolve classifiers. The performance of the GP method on image classification highly relies on these extracted features.

Al-Sahaf *et al.* [6] proposed a GP approach to automatically evolving texture descriptor for texture image classification with a small number of training instances. A number of conventional classification methods such as Nearest Neighbour (1NN) were employed to perform classification based on the extracted features. A dynamic GP method was proposed in [8], where a flexible length of feature vector is synthesised for texture classification. Experimental results have shown that this method outperformed the previous method where a fixed length of feature vector is extracted. However, these two GP descriptors were inspired by the LBP descriptor and are originally proposed for describing the texture feature, which might not perform well on the other image data.

Atkins *et al.* [13] proposed a multi-tier GP approach (simplified as 3TGP in this paper) to achieving automatic image feature extraction and classification. There are three tiers, i.e. an image filtering tier, an aggregation tier and a classification tier designed in this method, where each tier targets a subtask. The image filtering tier

is used for evolving several general filters such as max, mean and min to perform convolution operations to the input image. The aggregation tier is employed for detecting square regions and extracting domain independent features namely pixel statistics from these regions. However, the method performs image filtering before region detection, which might not be efficient.

Later, Al-Sahaf *et al.* [11] proposed a two-tier GP (2TGP) approach for 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 are employed. Two variants of 2TGP are proposed in [12] to detect more flexible regions and to extract features from these regions.

The features extracted by the 3TGP and 2TGP methods from detected regions are pixel statistics, which are relatively simple. To address this problem, Lenson *et al.* [14] designed a GP-HoG method based on the framework of 2TGP. This method designed the advanced feature descriptor HOG as a function in GP to extract high-level HOG histogram features from the detected regions. The GP-HoG method demonstrated a good example to integrate the HOG descriptor in GP to achieve high-level feature extraction and showed promising results in image classification. However, only using the HOG descriptor might not be efficient for GP to deal with complex image classification tasks such as texture image or scene image classification.

3 The Proposed Method

This section presents the proposed MLGP method in detail, including the GP program structure, the function set, the terminal set, and the fitness function.

3.1 Program Structure

To achieve automatic region detection, feature extraction, feature construction, and classification simultaneously, a multi-layer program structure is designed

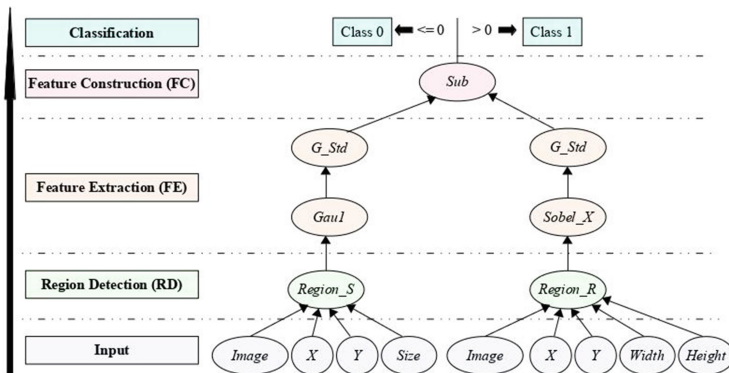


Fig. 1. An example of the program structure.

in the MLGP method. Figure 1 gives an example program to show how the multiple layers are constructed in a single program tree. There are five layers in the example program, i.e. Input, Region Detection, Feature Extraction, Feature Construction, and Classification, where each layer is shown in a different colour in Fig. 1.

The first (bottom) layer is **Input**, where images and constant parameters are feed from this layer to the GP method. The second layer is **Region Detection (RD)**, where prominent regions in an image are identified. The third layer is **Feature Extraction (FE)**, where several image operators are applied to deal with the detected regions such as using Gaussian smooth filter to reduce noise or using Sobel to detect edge features. Benefiting from these image operators, important and good features are expected to be detected and extracted. The fourth layer is **Feature Construction (FC)**, where extracted features are further constructed to a new high-level feature. The final (top) layer is **Classification**, in which a class label is assigned to the input image according to the value of the new feature and the predetermined threshold.

The program structure of the proposed MLGP method is constructed according to the five layers in a bottom-up manner, as shown in Fig. 1. It is a tree-based representation, where operators consist of the internal nodes and terminals consist of the leaf nodes. To deal with different tasks at each layer, a set of operators and terminals are employed. As the example program tree shown in Fig. 1, there are operators i.e. *Region_S*, *Gau1*, *G_Std*, *Sub*, *Region_R*, *Sobel_X* and terminals i.e. *Image*, *X*, *Y*, *Size*, *Width*, *Height*. More details about the operators and terminals can be seen in Sect. 3.2.

3.2 Functions and Terminals

(1) *Terminals for the Input layer*: There are four types of terminals for this layer, which represent the input image and the constant parameters of the proposed GP method. They are **Image**, **X**, **Y**, **Size**, **Width**, and **Height**. The *Image* terminal represents the input grey-scale image, which 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 $[0, \text{Image Width}]$ or $[0, \text{Image Height}]$. The terminals *Size*, *Width* and *Height* mean the width and height of a square/rectangle region. They are between $[20, 70]$ as the image sizes of our data sets are 128×128 or 40×100 . In the MLGP method, the values of *X*, *Y*, *Size*, *Width*, and *Height* are randomly generated initially and evolved during evolutionary process.

(2) *Operators for the RD layer*: There are two operators **Region_S** and **Region_R** are used for this layer. These two operators can detect a square/rectangle region at an appropriate position in an image with a suitable size by taking arguments from the Input layer as inputs. The *Region_S* operator detects a square region, which requires four arguments, including *Image*, *X*, *Y*, and *Size*. The *Region_R* operator detects a rectangle region, which needs five arguments, including *X*, *Y*, *Size*, *Width*, and *Height*. Notice that if there is an area in the detected region beyond the input image, only the area inside the input image is used as the detected region.

Table 1. Operators for the FE layer

Operator	Input	Output	Description
G_Std	A region	A number	The standard deviation of a region
Hist_Eq	A region	A region	Histogram Equalisation
Gau1	A region	A region	Gaussian smooth filter with $\sigma = 1$
Gau11	A region	A region	The first derivatives of Gaussian filter with $\sigma = 1$
GauXY	A region	A region	Gradient magnitude using Gaussian derivatives with $\sigma = 1$
Lap	A region	A region	Laplacian filter
Sobel_X	A region	A region	Sobel filter along X axis
Sobel_Y	A region	A region	Sobel filter along Y axis
LoG1	A region	A region	Laplacian of Gaussian filter with $\sigma = 1$
LoG2	A region	A region	Laplacian of Gaussian filter with $\sigma = 2$
LBP	A region	A region	Uniform LBP descriptor
HOG	A region	A region	HOG descriptor

(3) *Operators for the FE layer:* There are one designed operator and 11 image-related operators used for this layer, as listed in Table 1. The 11 image-related operators are used for dealing with regions detected by the RD layer. These operators include one histogram equalisation operator, eight image filters and two image descriptors. The **Hist_Eq** operator is designed to increase contrast and equalize the histogram of an image. The **Gau1**, **Gau11**, **GauXY**, **Lap**, **Sobel_X**, **Sobel_Y**, **LoG1**, and **LoG2** filters perform convolution operations on an image. The **Gau1** operator is used for reducing noise, and the remain filters are used for edge detection, flat detection or shape detection. In all these filters, the size are set to 3×3 as it is the commonly used. Two well-known image descriptors **LBP** [19] and **HOG** [4] are used in the function set for describing important shape and texture information of an image. In the **LBP** operator, the number neighbours is set to 8 and the radius is set to 1.5. In the **HOG** operator, the number of orientations is 9, the block size is 3×3 and the cell size is 8×8 .

Another important operator for this layer is **G_Std**, which calculates the standard deviation of an image/region. This operator must be selected in each program tree, which means only the standard deviation value is finally extracted from the detected/processed region. The standard deviation is a good measure for quantifying the variation of pixel values in an image/region. It is invariant to the pixel location changes. It should be pointed out that all the image operators and the orders among these operators are automatically evolved during GP evolutionary process. Hence, these operators allow GP to find good combinations of them to identify the difference of the standard deviation values of the images/regions from different classes and to extract good features.

(4) *Operators for the FC layer:* There is one arithmetic function used for this layer. It is **Sub(-)**, which takes two floating-point numbers as input and returns a floating-point number. One of its child nodes is the **G_Std** operator. Due to the standard deviation is always positive, only the **Sub** operator is employed in this

layer in order to reduce the search space and allow the final program output to be possible or negative.

(5) *Operators for the Classification layer*: The operation for this layer is that if the output from the FC layer is positive, the class label for the input image is 1 (class 1), otherwise the class label is 0 (class 2).

3.3 The Fitness Function

In the MLGP method, the fitness function (F) is the classification accuracy, which is straightforward and commonly used for binary image classification. The formula is shown by Eq. (1).

$$F = \text{Classification Accuracy} = \frac{TP + TN}{TOTAL} \times 100\% \quad (1)$$

where TP is the total number of True Positives, TN is the total number of True Negatives, and $TOTAL$ is the total number of classified images in the data set. TP represents the positive samples correctly classified into the positive class, and TN means the negative samples correctly categorised into the negative class. The proposed MLGP method is employed to evolve programs which can maximize the fitness function, i.e. classification accuracy.

4 Experiment Design

4.1 Datasets

To evaluate the performance of the proposed method, six different data sets are used in the experiments. These selected data sets represent six typical image classification tasks, including **COIL-20** [20] as object classification, **UIUC** [21] as car detection, **JAFFE** [22] as facial expression classification, **SCENE** [23] as scene classification, **TEXTURE** [24] as texture classification, and **BIRDS** [25] as fine-grained image classification. As the proposed method aims at dealing with binary image classification, each data set (except for UIUC) is formed by selecting two classes from the original data set. The difficulties of these data sets are various due to different variations such as scale, illumination, rotation in images. The majority

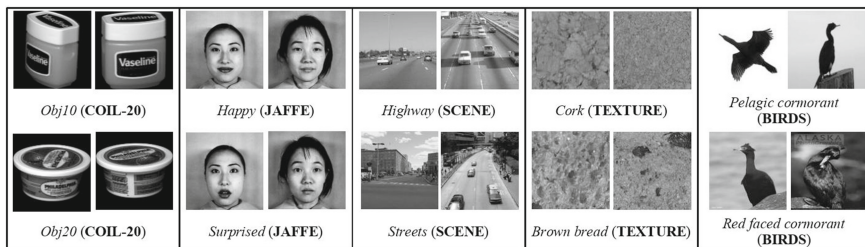


Fig. 2. Example images from COIL-20, JAFFE, SCENE, TEXTURE, and BIRDS.

data sets (except for BIRDS) are original gray-scale image data sets as this paper focuses on gray-scale image. The colour images in BIRDS are converted to gray-scale images.

In the experiments, each image data set is spilt into the training set, the validation set and the test set, having 50%, 25%, 25% images respectively. In JAFFE, the number of images in each set is the same due to the total number of images is small. The original images of the JAFFE, SCENE, TEXTURE, and BIRD data sets are resized to 128×128 with high quality in order to maintain the image size consistent of each data set and to reduce the dimensionality of image data. Details of the data sets are listed in Table 2. Several resized and transformed example images from each data set are shown in Figs. 2 and 3.

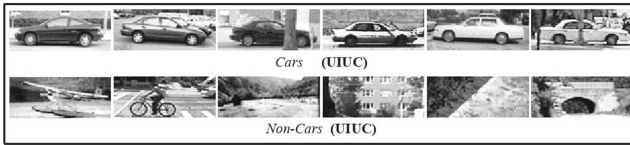


Fig. 3. Example images from UIUC.

Table 2. Data set properties

Name	Size	Classes	Training set	Validation set	Test set
COIL-20	128×128	<i>Obj20</i>	36	18	18
		<i>Obj10</i>	36	18	18
UIUC	100×40	<i>Cars</i>	275	110	165
		<i>Non-cars</i>	250	100	150
JAFFE	128×128	<i>Happy</i>	10	10	10
		<i>Surprised</i>	10	10	10
SCENE	128×128	<i>Highway</i>	130	65	65
		<i>Streets</i>	146	73	73
TEXTURE	128×128	<i>Cork</i>	216	103	113
		<i>Brown bread</i>	216	108	108
BIRDS	128×128	<i>Pelagic Cormorant</i>	30	15	15
		<i>Red Faced Cormorant</i>	26	13	13

4.2 Baseline Methods

In order to examine the performance of the proposed method, five GP-based methods and 42 non-GP methods are implemented as baseline methods in the experiments. The five GP-based methods are 3TGP [13], 2TGP [11], FeEx+GP [15], Hist+GP, and uLBP+GP. The Hist+GP method uses 64 histogram features as input and the uLBP+GP method utilises 59 uniform LBP histogram

features as input, where GP is used as a classification method. The 42 non-GP methods are based on seven commonly used machine learning classification methods and six existing image feature extraction methods. In each method, image features are extracted by a commonly used feature extraction method and then feed to a classification method to learn a classifier/classifiers. The six image feature extraction methods are FeEx [15], Histogram, GLCM [26], HOG [4], LBP [3], and uLBP [19], and details are shown in Table 3. The seven classification methods include 1NN, Gaussian Naive Bayes (NB), Decision Tree (DT), Multilayer Perception (MLP), Adaptive Boosting (AdaBoost), Random Forest (RF), Support Vector Machine (SVM with Radial Basis Function (RBF)). The implementation of these methods are based on the well-known *scikit-learn* [27] Python package. In the MLP method, the number of neurons in the hidden layer is set to 50 [28], the activation function is the logistic sigmoid function, and the learning rate is adaptive. The non-linear kernel RBF is employed in SVM as it is commonly used. For all these non-GP methods, the training instances include the training set and the validation set used in the GP approaches, while the test set keeps the same.

Table 3. Image feature extraction methods

Methods	Description
FeEx	20 domain independent features [15]
Histogram	256 histogram features based on the pixel intensities of the gray-scale image
GLCM	24 GLCM features. Four different orientations are used and the contrast, dissimilarity, homogeneity, energy, correlation and ASM are extracted from each GLCM
HOG	HOG features. Using the HOG method to extract features. In HOG, the orientation is set to 9, the pixels in each cell is set to 32×32 (12×12 for UIUC), and the cells in each block is set to 3×3
LBP	256 LBP histogram features. In LBP, the number of neighbour is set to 8 and the radius is set to 1.5
uLBP	59 uniform LBP histogram features

4.3 Parameter Settings

All the GP methods are implemented in Python based on the *DEAP (Distributed Evolutionary Algorithm in Python)* [29] package. Parameter settings in all the GP methods are the same as listed in Table 4. On each data set, each algorithm has been run 30 times independently with different random seeds.

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

Table 4. GP run time parameters

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

5 Results and Discussions

All the experimental results of GP methods are shown in Table 5 and that of non-GP methods are listed in Table 6. The Student’s *t*-test with a 5% significance level is employed to compare the proposed method with a GP or non-GP method. In Tables 5 and 6, the “+” indicates that the proposed method is significantly better than the corresponding method, the “-” indicates the proposed method performs significantly worse than the corresponding method, and the “=” indicates that the proposed method performs similar to the corresponding method.

5.1 Compared with GP Methods

Table 5 shows the test results in terms of maximum, mean and standard deviation of classification accuracies obtained by the MLGP method and the other five GP methods on the six data sets in 30 runs. The first COIL-20 data set is easy so that all the GP methods obtain 100% maximum accuracy. There is no significant improvement over the other GP methods on this data set. On UIUC and JAFFE, the MLGP method obtains significantly better or similar performance compared to the other GP methods. On JAFFE, the MLGP method obtains 100% maximum classification accuracy and 91.67% mean classification accuracy, which achieves nearly 9% increase to the 82.83% maximum mean classification accuracy obtained by the 2TGP method. On SCENE and TEXTURE, the proposed MLGP method achieves significantly better performance than the 2TGP, 3TGP, FeEx+GP and Hist+GP methods. Compared to these four GP methods, the MLGP method has a 7% and 3% increase in the mean classification accuracy on SCENE and TEXTURE. However, the uLBP+GP method outperforms the MLGP method significantly on these two data sets. Images in these two data sets contain a large amount of texture information, which can be captured well by the uniform LBP histogram features in the uLBP+GP method. The proposed MLGP method extracts features from the detected regions while the uLBP+GP method uses uniform LBP histogram features from the overall image, which might further improve its performance. On the final BIRDS data set, the proposed MLGP method achieves significantly better or comparable performances than the other GP methods.

Table 5. Classification accuracy (%) of all the GP methods on the six data sets

	Max	Mean \pm St.D.	Max	Mean \pm St.D.	Max	Mean \pm St.D.
	COIL-20		UIUC		JAFFE	
MLGP	100.0	99.91 \pm 0.5	92.38	89.47 \pm 2.06	100.0	91.67 \pm 6.50
2TGP	100.0	100.0 \pm 0.0=	90.48	86.55 \pm 2.89+	95.00	82.83 \pm 8.53+
3TGP	100.0	100.0 \pm 0.0=	93.02	88.42 \pm 2.42=	100.0	82.67 \pm 9.20+
FeEx+GP	100.0	100.0 \pm 0.0=	88.25	81.76 \pm 2.56+	90.00	70.67 \pm 13.59+
Hist+GP	100.0	99.91 \pm 0.50=	65.71	60.81 \pm 2.08+	75.00	52.17 \pm 9.28+
uLBP+GP	100.0	99.81 \pm 0.69=	85.71	81.51 \pm 2.22+	65.00	53.83 \pm 6.01+
	SCENE		TEXTURE		BIRDS	
MLGP	92.75	90.97 \pm 1.40	97.74	90.23 \pm 3.48	71.43	61.67 \pm 6.45
2TGP	86.23	81.33 \pm 2.12+	81.90	75.60 \pm 3.87+	67.86	51.79 \pm 7.70+
3TGP	88.41	82.56 \pm 2.19+	88.24	82.68 \pm 4.18+	71.43	56.19 \pm 5.75+
FeEx+GP	86.96	83.16 \pm 2.37+	88.69	83.65 \pm 2.36+	64.29	54.64 \pm 5.77+
Hist+GP	86.96	83.29 \pm 1.65+	94.57	87.36 \pm 3.86+	78.57	51.67 \pm 9.53+
uLBP+GP	96.38	92.85 \pm 1.92-	97.29	92.37 \pm 2.77-	71.43	60.36 \pm 7.57=

In total, the proposed MLGP method achieves significantly better performance in 21 cases and comparable performance in 7 cases of out of the total 30 cases. In summary, the proposed MLGP method achieves significantly better or comparable performance on these six different image classification tasks compared to the other GP methods.

5.2 Compared with Non-GP Methods

Table 6 lists all the test results of the total 42 non-GP methods on the six data sets. On COIL-20, the proposed method achieves similarly or significantly better results than the non-GP methods. On UIUC, the MLGP method gains significantly better results than all the classification methods with the FeEx, Histogram, GLCM, LBP, uLBP features. In total, the MLGP method obtains 39“+” and 3“-” on this data set. On JAFFE, the MLGP method obtains 100% maximum accuracy and 91.67% mean accuracy, which achieves better or comparable performance in 40 cases out of the 42 cases. On SCENE, the MLGP method significantly outperforms all the classification methods with the FeEx and Histogram features. But this method is significantly worse than the classification methods with the other four features in some cases. The results on TEXTURE also show a similar pattern. In total, the proposed MLGP method outperforms the non-GP methods in 25 cases out of 42 cases on both SCENE and TEXTURE, and is significant worse than these methods in 13 cases on the SCENE data set and in 10 cases on the TEXTURE data set. On the most difficult BIRDS data set, the proposed MLGP method achieves comparable or significantly better results than the classification methods with the FeEx, Histogram, GLCM, and HOG features. But the classification methods with

Table 6. Classification accuracy (%) of 42 non-GP methods on the six data sets

	1NN	NB	DT	MLP	AdaBoost	RF	SVM-RDF
	COIL-20		(MLGP 100/99.91 ± 0.5)				
FeEX	100.0=	100.0=	97.22+	100.0=	97.22+	100.0=	97.22+
Histogram	100.0=	100.0=	100.0=	100.0=	100.0=	100.0=	50.00+
GLCM	100.0=	100.0=	100.0=	80.56+	100.0=	100.0=	50.00+
HOG	100.0=	100.0=	100.0=	100.0=	100.0=	100.0=	100.0=
LBP	100.0=	100.0=	100.0=	100.0=	100.0=	100.0=	52.78+
uLBP	100.0=	100.0=	97.22+	100.0=	97.22+	100.0=	50.00+
	UIUC		(MLGP 92.38/89.47 ± 2.06)				
FeEX	83.49+	87.62+	83.49+	77.78+	86.67+	86.67+	77.14+
Histogram	55.87+	66.03+	62.22+	60.95+	65.08+	67.94+	52.38+
GLCM	84.13+	79.68+	85.40+	61.90+	86.98+	86.67+	52.38+
HOG	92.06-	64.76+	86.98+	68.89+	97.14-	92.38-	66.98+
LBP	85.71+	85.40+	79.37+	86.35+	88.25+	84.76+	52.38+
uLBP	86.67+	85.08+	82.22+	83.17+	86.98+	87.62+	52.38+
	JAFFE		(MLGP 100.0/91.67 ± 6.5)				
FeEX	90.00=	55.00+	80.00+	50.00+	85.00+	75.00+	50.00+
Histogram	90.00=	55.00+	60.00+	90.00=	60.00+	45.00+	90.00=
GLCM	65.00+	60.00+	80.00+	50.00+	75.00+	70.00+	50.00+
HOG	100.0-	100.0-	90.00=	50.00+	90.00=	90.00=	90.00=
LBP	75.00+	65.00+	70.00+	80.00+	55.00+	60.00+	50.00+
uLBP	75.00+	65.00+	35.00+	45.00+	65.00+	85.00+	75.00+
	SCENE		(MLGP 92.75/90.97 ± 1.4)				
FeEX	88.41+	85.51+	80.43+	80.43+	86.23+	86.23+	79.71+
Histogram	79.71+	81.16+	81.88+	82.61+	85.51+	87.68+	52.90+
GLCM	92.03-	88.41+	89.86+	91.30=	91.30=	93.48-	52.90+
HOG	94.93-	87.68+	89.13+	89.86+	92.75-	92.03-	90.58=
LBP	89.86+	94.20-	94.20-	87.68+	96.38-	94.93-	52.90+
uLBP	89.86+	94.20-	95.65-	90.58=	96.38-	94.93-	52.90+
	TEXTURE		(MLGP 97.74/90.23 ± 3.48)				
FeEX	90.50=	84.16+	87.33+	51.13+	90.50=	90.50=	78.28+
Histogram	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+
HOG	81.90+	52.04+	74.21+	52.04+	76.02+	78.73+	52.04+
LBP	98.19-	83.26+	87.78+	93.67-	91.40=	88.69+	48.87+
uLBP	96.83-	86.88+	85.07+	85.52+	90.05=	90.05=	48.87+
	BIRDS		(MLGP 71.43/61.67 ± 6.45)				
FeEX	57.14+	53.57+	46.43+	53.57+	64.29-	46.43+	53.57+
Histogram	53.57+	50.00+	53.57+	50.00+	53.57+	53.57+	53.57+
GLCM	53.57+	53.57+	60.71=	60.71=	53.57+	53.57+	53.57+
HOG	57.14+	60.71=	57.14+	57.14+	64.29-	53.57+	57.14+
LBP	71.43-	71.43-	57.14+	64.29-	75.00-	78.57-	53.57+
uLBP	78.57-	75.00-	57.14+	75.00-	71.43-	78.57-	53.57+

the uLBP and LBP features achieves significantly better results than the MLGP method in the majority cases, which indicates that images in the BIRDS data set also contain much texture information and the global features is very important for the difficult fine-grained classification task.

The methods that achieve significantly better results than the proposed MLGP method in some cases are mainly AdaBoost and Random Forest, which are boosting and ensemble classifiers, while the proposed MLGP method only uses a single evolved program. In addition, compared with the non-GP methods that learn classifiers from a set of image features which are global features extracted from the whole image, the proposed method only use a single constructed feature from smaller regions for classification based on the predefined threshold. The comparison to the non-GP methods is actually not entirely fair for the proposed method. Even in these cases, the proposed method still obtains better or comparable performance compared to the non-GP methods.

The results also confirm that the performance of these non-GP methods highly rely on the feature extraction method and the classification method for dealing with different image classification tasks. For example, the HOG features with 1NN achieves better results on the JAFFE and TEXTURE data sets, but obtains worse results on the SCENE and BIRDS data sets. Even the boosting and ensemble classification methods perform better than the others in most cases, the simplest 1NN with particular features such as LBP or uLBP performs much better than using the other classification methods with the same features on the TEXTURE data set. These results reveal that the feature extraction method and the classification method must be carefully selected and suited when dealing with image classification tasks. Oppositely, the proposed MLGP method can obtain promising results in image classification without such considerations.

6 Further Analysis

This section analyses two example programs evolved by the MLGP method to show the good interpretability and understandability of the proposed method. These two example programs evolved on COIL-20 and JAFFE will give more insight into how they achieve good classification performance.

6.1 Example Program on the COIL-20 Data Set

As the COIL-20 data set is very easy, the majority programs evolved by MLGP in 30 runs can achieve perfect classification performance. To show the good interpretability and understandability, a simplest program is selected for analysis, as shown in Fig. 4. This program achieves 100% classification accuracy on training, validation and test sets. Figure 4 gives the example program, the example image from different classes, and the outputs of each nodes of the example program. In the figure, the red colour represents the outputs/regions of the *Obj10* class, and the green colour indicates that of the *Obj20* class.

This program identifies two rectangle regions with different sizes at different positions in an input image. The size of the left identified region (the left side of program in Fig. 4) is actually 60×57 as the size of the image is 128×128 . This region captures the differences of the partial objects from different classes. The size of the right detected region (the right side of program in Fig. 4) is 43×48 . This region finds the distinctive difference among two classes by capturing an area from the top right side of an image. In this region, the *Obj10* class shows more white colour of the lid of the Vaseline product, while the *Obj20* class contains more black colour. The *G_Std* operator calculates the standard deviation value of each detected region in this program. In terms of the standard deviation value, the *Obj10* is smaller than the *Obj20* in the left region, but it is bigger than the *Obj20* in the right region. Hence, the difference is constructed by the *Sub* operator for classification.

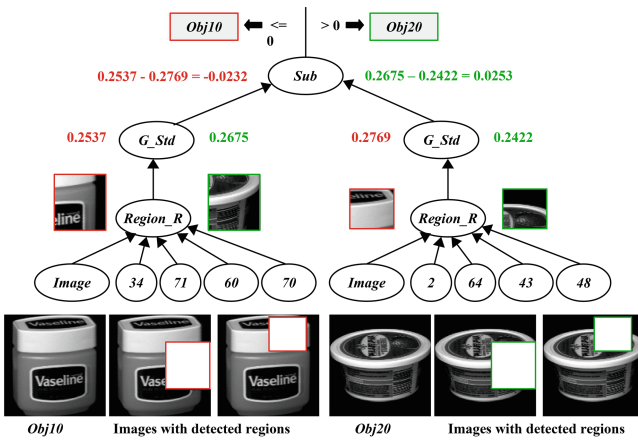


Fig. 4. An example program evolved by the MLGP method on the COIL-20 data set. (Color figure online)

6.2 Example Program on the JAFFE Data Set

Figure 5 demonstrates an example program evolved by the MLGP method on the JAFFE data set. This program achieves 95% classification accuracy on the training set, 80% accuracy on the validation set, and 100% accuracy on the test set. This program detects two different rectangle regions of an image. The left detected region with a size of 22×28 is smaller than the right region with a size of 46×31 . The left region captures an area between the two eyes/eyebrows in a face image. The *Happy* and *Surprised* classes do not show significant difference in the left region. The *Happy* and *Surprised* classes do not show significant difference in the left region. The *Hist.Eq* and *Lap* operators are evolved to deal with the left region, where the first one increases the contrast of the region, and the second operator detects the flat area and the area with edges. The difference of the two classes in the left region is enhanced by the two operators as we can see from the

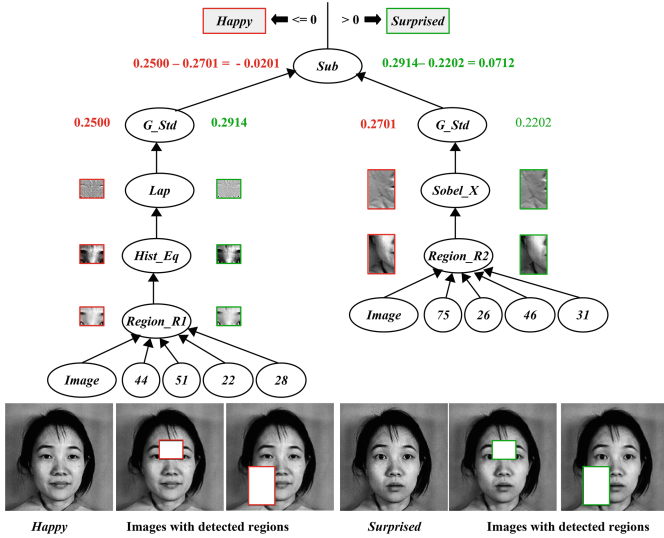


Fig. 5. An example program evolved by the MLGP method on the JAFFE data set.

left side of Fig. 5. The right detected region detects the lower left side of an face where the partial shape of the face is included. The *Sobel_X* operator is evolved to detect the edges along the horizontal direction. By these evolved operators, more features such as edges are detected and the differences of the standard deviation values between two classes are enhanced. The two standard deviation values of two detected regions are constructed by the *Sub* operator further for classification. This example program describes a relationship between two regions by using a set of operators, which achieves perfect classification performance on the test set of the JAFFE data set.

7 Conclusions

This paper proposed an MLGP approach to achieving automatic region detection, high-level feature extraction, feature construction, and image classification simultaneously. A novel program structure and a new function set were designed in the MLGP method, which well combined the prior image domain common sense knowledge and the flexible representation of GP. The performance of MLGP was examined on six image data sets of varying difficulty and was compared to other five GP methods and 42 non-GP methods. The experimental results shown that the MLGP method achieved significantly better or comparable results than the other five GP methods. Compared with the 42 non-GP methods where traditional feature extraction and classification methods are used, the MLGP method achieved more stable and better performance. The proposed MLGP method could automatically adjust to different image tasks

and find good solutions due to the utilisation of prior general domain knowledge. Besides the good performance on classification, the further analysis on the evolved programs illustrated the good interpretability and understandability of the MLGP method. The MLGP method evolved very simple programs but with very high or perfect classification accuracy. The analysis of evolved programs shown that important and prominent high-level image features could be extracted and constructed by MLGP from the automatically detected regions for classification.

In this paper, only one high-level feature is constructed by MLGP for image classification, which might not be efficient. To address this problem, the features extracted by the FC layer of MLGP will be further investigated. A conventional classification method will be used for image classification based on these features to investigate whether the classification performance can be further improved on these data sets.

References

1. Pinz, A.: Object categorization. *Found. Trends Comput. Graph. Vis.* **1**(4), 255–353 (2005)
2. Haralick, R.M., Shanmugam, K., et al.: Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* **6**, 610–621 (1973)
3. Ojala, T., Pietikainen, M., Harwood, D.: 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, pp. 582–585. IEEE (1994)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893. IEEE (2005)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
6. Al-Sahaf, H., Al-Sahaf, A., Xue, B., Johnston, M., Zhang, M.: Automatically evolving rotation-invariant texture image descriptors by genetic programming. *IEEE Trans. Evol. Comput.* **21**(1), 83–101 (2017)
7. Xue, B., Zhang, M.: Evolutionary feature manipulation in data mining/big data. *ACM SIGEVOlution* **10**(1), 4–11 (2017)
8. Al-Sahaf, H., Zhang, M., Al-Sahaf, A., Johnston, M.: Keypoints detection and feature extraction: a dynamic genetic programming approach for evolving rotation-invariant texture image descriptors. *IEEE Trans. Evol. Comput.* (2017). <https://doi.org/10.1109/TEVC.2017.2685639>
9. Bi, Y., Zhang, M., Xue, B.: An automatic region detection and processing approach in genetic programming for binary image classification. In: *The 32nd International Conference Image and Vision Computing New Zealand (IVCNZ 2017)*, pp. 1–6. IEEE (2017)
10. Fu, W., Johnston, M., Zhang, M.: Genetic programming for edge detection: a Gaussian-based approach. *Soft Comput.* **20**(3), 1231–1248 (2016)
11. Al-Sahaf, H., Song, A., Neshatian, K., Zhang, M.: Extracting image features for classification by two-tier genetic programming. In: *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8. IEEE (2012). <https://doi.org/10.1109/CEC.2012.6256412>

12. Al-Sahaf, H., Song, A., Neshatian, K., Zhang, M.: Two-tier genetic programming: towards raw pixel-based image classification. *Expert Syst. Appl.* **39**(16), 12291–12301 (2012)
13. Atkins, D., Neshatian, K., Zhang, M.: A domain independent genetic programming approach to automatic feature extraction for image classification. In: 2011 IEEE Congress on Evolutionary Computation, pp. 238–245. IEEE (2011)
14. Lensen, A., Al-Sahaf, H., Zhang, M., Xue, B.: Genetic programming for region detection, feature extraction, feature construction and classification in image data. In: Heywood, M.I., McDermott, J., Castelli, M., Costa, E., Sim, K. (eds.) EuroGP 2016. LNCS, vol. 9594, pp. 51–67. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30668-1_4
15. Zhang, M., Ciesielski, V.: Genetic programming for multiple class object detection. In: Foo, N. (ed.) AI 1999. LNCS (LNAI), vol. 1747, pp. 180–192. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-46695-9_16
16. Nandi, R., Nandi, A.K., Rangayyan, R.M., Scutt, D.: Classification of breast masses in mammograms using genetic programming and feature selection. *Med. Biol. Eng. Compu.* **44**(8), 683–694 (2006)
17. Ain, Q.U., Xue, B., Al-Sahaf, H., Zhang, M.: Genetic programming for skin cancer detection in dermoscopic images. In: 2017 IEEE Congress on Evolutionary Computation, pp. 2420–2427. IEEE (2017)
18. Ryan, C., Fitzgerald, J., Krawiec, K., Medernach, D.: Image classification with genetic programming: building a stage 1 computer aided detector for breast cancer. In: Gandomi, A.H., Alavi, A.H., Ryan, C. (eds.) Handbook of Genetic Programming Applications, pp. 245–287. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20883-1_10
19. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002)
20. Nene, S.A., Nayar, S.K., Murase, H., et al.: Columbia object image library (coil-20). Technical report, Columbia University (1996)
21. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern. Anal. Mach. Intell.* **26**(11), 1475–1490 (2004)
22. Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J.: Coding facial expressions with Gabor wavelets. In: The Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200–205. IEEE (1998)
23. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 524–531. IEEE (2005)
24. Mallikarjuna, P., Targhi, A.T., Fritz, M., Hayman, E., Caputo, B., Eklundh, J.O.: The KTH-TIPS2 database. Computational Vision and Active Perception Laboratory (CVAP), Stockholm, Sweden (2006). <http://www.nada.kth.se/cvap/databases/kth-tips>
25. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Technical report CNS-TR-2010-001, California Institute of Technology (2010)
26. Haralick, R.M.: Statistical and structural approaches to texture. *Proc. IEEE* **67**(5), 786–804 (1979)

27. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
28. Khotanzad, A., Lu, J.H.: Classification of invariant image representations using a neural network. *IEEE Trans. Acoust., Speech, Signal Process.* **38**(6), 1028–1038 (1990)
29. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: evolutionary algorithms made easy. *J. Mach. Learn. Res.* **13**, 2171–2175 (2012)