A recent survey on the applications of genetic programming in image processing

Asifullah Khan\textsuperscript{1,2} | Aqsa Saeed Qureshi\textsuperscript{1} | Noorul Wahab\textsuperscript{1} | Mutawarra Hussain\textsuperscript{1} | Muhammad Yousaf Hamza\textsuperscript{3}

\textsuperscript{1}Pattern Recognition Lab, Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Nilore, Pakistan
\textsuperscript{2}Deep Learning Lab, Centre for Mathematical Sciences, Pakistan Institute of Engineering and Applied Sciences, Nilore, Pakistan
\textsuperscript{3}Department of Physics and Applied Mathematics, Pakistan Institute of Engineering and Applied Sciences, Nilore, Pakistan

Correspondence
Asifullah Khan, Department of Computer Science, Pakistan Institute of Engineering and Applied Sciences, Nilore, Islamabad 45650, Pakistan.
Email: asif@pieas.edu.pk

Abstract
Genetic programming (GP) has been primarily used to tackle optimization, classification, and feature selection related tasks. The widespread use of GP is due to its flexible and comprehensible tree-type structure. Similarly, research is also gaining momentum in the field of image processing, because of its promising results over vast areas of applications ranging from medical image processing to multispectral imaging. Image processing is mainly involved in applications such as computer vision, pattern recognition, image compression, storage, and medical diagnostics. This universal nature of images and their associated algorithm, that is, complexities, gave an impetus to the exploration of GP. GP has thus been used in different ways for image processing since its inception. Many interesting GP techniques have been developed and employed in the field of image processing, and consequently, we aim to provide the research community an extensive view of these techniques. This survey thus presents the diverse applications of GP in image processing and provides useful resources for further research. In addition, the comparison of different parameters used in different applications of image processing is summarized in tabular form. Moreover, analysis of the different parameters used in image processing...
related tasks is carried-out to save the time needed in the future for evaluating the parameters of GP. As more advancement is made in GP methodologies, its success in solving complex tasks, not only in image processing but also in other fields, may increase. In addition, guidelines are provided for applying GP in image processing related tasks, the pros and cons of GP techniques are discussed, and some future directions are also set.

**KEYWORDS**
artificial intelligence, computational intelligence, genetic programming, image processing

### 1 | INTRODUCTION

The sense of vision plays an essential role in the process of human perception. Human vision is restricted only to the visual band of the electromagnetic spectrum, but machine vision covers nearly the whole electromagnetic spectrum, ranging from gamma rays to radio waves.\(^1\) Image processing tries to emulate the capabilities of the human eye and brain in extracting features or segmenting regions. Therefore, image processing is a challenging task in the sense that these algorithms have to be accurate, fast, reliable, as well as robust. Development in image processing has considerably increased with the decline in the prices of computers. Due to its diverse applications, image processing cannot be wholly distinguished from its closely related fields; computer vision and image analysis. This overlapping is because image processing is also involved in both computer vision and image analysis at different levels. In the somewhat restricted definition of image processing, it is a process whose inputs and outputs are images and can be extended to encompass processes that involve techniques of feature extraction from images to identify the individual objects.\(^2\)

Different intelligent techniques such as an artificial immune system, genetic algorithm (GA), artificial neural network (ANN), ant colony optimization, evolutionary computation (EC), and genetic programming (GP) have been exploited in the field of image processing. The term “Computational Intelligence” (CI) is a broad term encompassing the different intelligent techniques mentioned above. CI can find optimum/near-optimum solutions to computationally hard problems in a variety of domains.\(^3\) Thus to split the various image processing techniques, we have two broad categories; conventional image processing and CI-based image processing techniques.

This survey focuses on the applications of GP in image processing. GP is one of the promising CI technique that comes under the subcategory of EC techniques based on the Darwinian theory of evolution. GP evolves output in the form of a tree or a computer program. Different programs are generated depending on the terminal and function sets used. Other existing CI paradigms do not produce solutions in the form of computer programs, but instead involve specialized structures such as weight vectors for neural networks, coefficients for polynomials, chromosome strings in the conventional GA, and so forth.\(^4\) In literature EC techniques are used to solve the image and vision related problems\(^5\)\(^–\)\(^9\). GP comes under the umbrella of EC along with GA, evolutionary programming, differential evolution, and evolutionary strategies.\(^10\) GP is a particular
form of the GA, which uses a fixed (though variants now exist) length representation in the form of string of bits or real numbers to represent individuals called chromosomes.

In contrast to GA, GP represents individuals as trees that can be evaluated to obtain results. Initially, a population of individuals is randomly generated using a terminal set (which contains constants, argument-less functions, variables) and a function set (e.g., +, −, /, if-else, and so forth). Based on their fitness, the individuals are given chances for reproduction and allowed to change via crossover and mutation. Crossover is used to search for an optimal solution, whereas mutation introduces rapid changes in the population and thus helps to avoid trapping in local optima.

GP has gained popularity in applications such as data modeling, symbolic regression, image and signal processing, medicine, bioinformatics, financial trading, and industrial process control. This popularity of GP is mainly through its flexible nature, generality, almost no requirement of preprocessing, its ability to provide the mathematical expression of the solution, and parallelization. There are different variants of GP, and one of them is brain programming, which combines the features in such a way that important visual representations are enhanced for the given problem.11–15

This survey addresses GP’s applicability in image processing and is organized as follows. The background of GP and image processing is described in Section 2. The importance of the review is presented in Section 3. The similarities of the GP approaches in different categories of IP are given in Section 4 and further reviewed in Section 5. Section 6 is about the advantages and disadvantages of using GP in image processing. Section 7 presents guidelines for applying GP in image processing. The comparison and discussions are provided in Section 8, while Section 9 concludes the article.

2 RELATED CONCEPTS

This section briefly describes the very basics of the two subjects of this review; image processing and GP. We also discuss the scope of GP in image processing.

2.1 Image processing

Image is a visual representation of an object produced on a surface and considered as a graph of function having points \((x, y, \text{func}(x, y))\), in which \((x, y)\) and \(\text{func}(x, y)\) are location and value of the point, respectively.16 According to Olague,5 image can be represented as a graph of function. Therefore, it can be represented as an image itself, the second way is to represent it as a matrix of positive integers, and third way is to represent it as three-dimensional surface. Before the invention of paper, images were produced on stones and other materials. In the case of computers, a visual representation of an image is displayed on a monitor, a liquid crystal display, or a multimedia projector. However, for computer storage, these images are defined as two-dimensional matrices of pixel (picture-element) values. These pixel values are the intensity or gray level of the image and can be represented in the form of a function \(F(x, y)\), where \(x\) and \(y\) are spatial coordinates. If the intensity values within an image are finite discrete quantities, then, such an image is a digital image. A pixel of size one byte (8 bits) can represent 256 intensity values from 0 (black) to 255 (white). The values in between this range give different shades, as shown in Figure 1. When values of such a representation are processed/modified in some way, we call it image processing. For example, enhancing the image quality, removing noise, segmenting specific parts, making a
comparison with other images, and so forth, all include processing the image in some way. For the image in Figure 1, if we want to change the center pixel to black, then we change its value from 78 to 0.

2.2 Genetic programming

GP is one of the promising EC techniques and is viewed as a specialization of GA. GP and GA mainly differ in the representation scheme. GA uses strings of bits, integers, or real numbers to represent individuals. By contrast, GP mainly represents individuals as trees and is well suited for mapping functions, model development, nonlinear regression, and other related problems. Fonlupt and Koza\textsuperscript{17,18} has pointed out various exciting problems, where GP produced human-competitive results. GP is a domain-independent method and can solve complex problems automatically.\textsuperscript{19} Moreover, pioneering works of Koza, Langdon, Poli, and Banzhaf has boosted research in the field of GP.\textsuperscript{19–23}

Figure 2 depicts the genetic search cycle of EC techniques, where an initial population is generated, and then, the fittest individuals are selected as parents based on some evaluation criterion. In the next step, genetic operators (e.g., crossover, mutation, and reproduction) are applied to produce offsprings. In the last step, fittest individuals are selected as a population for the next generation. The whole search cycle continues after each generation until a termination criterion fulfills, and the best candidate becomes the fittest individual.

Similarly, Figure 3 depicts the basic flow of GP, in which an initial population is generated randomly. Then parents are selected randomly from this initial population, and different genetic operators are applied. After the application of genetic operators, the selected individuals become part of the next generation. This process is repeated until a termination criterion meets, and finally, the best-evolved GP tree is saved.

2.3 Scope of GP in image processing

Koza in 1992 introduced GP as a problem-solving paradigm, and since then, it has been applied in many image-related problems. Its expressive power has been utilized in various image processing tasks such as image preprocessing, region analysis, segmentation, object detection, classification, and postprocessing. Notably, in the field of medical imaging, it is often applied for classifying cancerous and noncancerous cells. Moreover, GP has been used for developing accurate classifiers for object detection, classification of medical images, and optical character recognition.
Multiobjective GP (MOGP)\textsuperscript{24–27} is also widely used for image processing related problems in which the optimization of more than one objective function is required. Most of the time, the objectives that are needed to optimize are conflicting. For example, in image watermarking techniques, the objective usually is to increase both the imperceptibility and payload of the watermarked image. However, there is always a tradeoff between the two objectives. For image recognition and vision related tasks interest point detection (IPD) plays a significant role. There are different well know techniques which combine image processing and computer vision-based learning methodologies with searching capability of GP.\textsuperscript{28–35} All of these techniques raise the IPD as an optimization problem and act as a predecessors for image retrieval techniques.\textsuperscript{36} Similarly, GP is used in military-related applications such as detecting objects, and analysis of Satellite and infrared images. Besides medical and military applications, GP is also employed in other exciting fields such as environmental studies, exploration, crop production, and image indexing.\textsuperscript{19}

![Genetic search cycle of evolutionary computation techniques](image)

**FIGURE 2** Genetic search cycle of evolutionary computation techniques

3 | IMPORTANCE OF THE REVIEW

Due to the rapid increase in the availability of images and videos over the last few years, GP has been successfully applied to many image processing applications. In this regard, assessing the prospects of GP in the field of image processing will be a useful guide for researchers. Generally, the performance of algorithms related to segmentation, edge detection, enhancement, and classification related problems suffer if the images are blurred. In this situation, GP can evolve suitable filters so that images are filtered before applying any image processing task. Owing to the importance of GP in image processing, GP-based methodologies have been consistently evolving, and new ideas and techniques have been proposed. This survey can help to explore GP related approaches in different areas of image processing. The pros and cons of GP are discussed
for practical purposes and further research. In addition, the techniques presented in this article highlights many aspects of GP.

4 TERMINOLOGIES USED IN GP

In this section, the various terminologies associated with GP are discussed. The GP techniques applied in different fields of image processing (presented in this article) are different in terms of their domains. Still, they do share some similarities in solving the problems.

**Representation:** In most of the approaches, GP individuals are represented as a tree structure. Moreover, the linear representation for GP, which is also constructed using functions and terminals, is employed in a few works.37

**Function set:** The function set is chosen according to the problem at hand. For example, for regression related problems, the function set might comprise of arithmetic operations (*, %, +, −). Similarly, for image processing applications, a specialized function set, according to the nature of the problem domain, may be used.

**Terminal set:** The terminals like functions, also do not have any specific predefined set. The GP terminal set is comprised of variables (also called program input), constants, or random inputs. In the case of image processing applications, mostly raw pixel values are used as terminals.

**Fitness function:** Function and terminal sets, which are used to express a GP tree, also define the search space that GP explores during the search process. Fitness function measures how good
or bad is a specific region within the search space. Different fitness functions, depending on the nature of the problem, have been used as an evaluation measure during the search process, such as root-mean-square-error (RMSE), peak-signal-to-noise ratio (PSNR), accuracy, area-under-the receiver operating characteristic (AUC-ROC) curve, and so forth.

**Initial population:** If prior knowledge about the properties of the desired solutions is not known, then the initial individuals are generated randomly. Moreover, other methods initialize the population with the help of a seed.² Olague and Chan-Ley³⁸ applied the best set of solutions of previous runs to initialize a new set of experiments with improvements that reach up to 5%.

**Selection method:** In GP evolution cycles, mainly two types of selection methods are used, that is, parent selection and survivor selection. There are different types of selection methods, but tournament selection is the widely used selection mechanism. In parent selection, individuals having higher fitness are selected as parents for the next generation. Whereas, survivor selection is performed on individuals who are produced from selected parents.

**Genetic operators:** Different genetic operators (crossover, mutation, reproduction, and so forth) are used for the generation of offspring to introduce diversity among the individuals of the population.

5 | CATEGORY-WISE IMAGE PROCESSING APPLICATIONS OF GP

This section presents the different GP techniques applied in various fields of image processing, such as image enhancement, compression, segmentation, retrieval, classification, and registration.

5.1 | GP in image enhancement

Images can be enhanced to improve their visual appearance. Enhanced images can further improve the image processing related tasks such as image segmentation, object detection, and recognition. However, image enhancement for one application may not be the right candidate for another application, and this means that image enhancement has different semantics for different applications.

Different image enhancement techniques can be carried out either in the original (spatial) or transformed (frequency) domain³⁹–⁴² In the original domain, the operations are carried out directly on the pixels. By contrast, in the case of the frequency domain, first, the image is transformed in the frequency domain, and then the enhancement is carried out. Sometimes the desired objects that need to be detected (called the region of interest) are emphasized during the enhancement step to help perceive them. For example, to make the process of object extraction easy, an image can be enhanced by decreasing the similarities between the specified object and the background.

In an exciting work, Poli et al.⁴⁰ used a pseudo color transformation that utilized GP and developed a program for image enhancement. Similarly, Wang and Tan⁴¹ used GP algorithms to evolve morphological operations that converted a binary image into the desired image, which contained only the required features. In Wang’s approach, the automatic evaluation mechanism enabled the GP algorithm to generate practical morphological algorithms. On the other hand, Khan et al.⁴² proposed a GP-based hybrid filter that assisted in reducing the region noise. Their method helped
to preserve the details related to edges and structure of the region. Block diagram of Khan’s technique is shown in Figure 4. It comprises of two phases. In the first phase, features were extracted from the noisy magnetic-resonance-imaging (MRI) images using three different types of filters. The extracted features were concatenated to form a feature vector. This feature vector was then used in the second phase to train a GP module. After the training phase, the best evolved GP expression was utilized to check the effectiveness of the proposed technique on new images.

5.2 | GP in image restoration

During the process of acquiring and transmitting or capturing digital images, the image quality might degrade. Different techniques can be applied to restore the original image. In the case of image restoration, the cause of degradation is either known or unknown. Figure 5 shows the case when the cause of degradation is known. In this case, the original image can be restored using prior knowledge. In a case where there is no such information, then the degraded function can be estimated by image observation, experimentation, or modeling. In literature, blind deconvolution and image denoising-based methods are reported for restoring the original image. Restoration by an estimated degradation function is sometimes called blind deconvolution. Whereas in the case of image denoising, spatial or frequency domain filters are used for the restoration of the original image.

5.2.1 | GP in image deconvolution

In literature, GP is rarely used for image deconvolution. Among the few reported methods, GP-based blind image deconvolution filter was proposed by Mahmood et al. In Majeed’s
technique, for a small neighborhood of each pixel of a degraded image, a set of feature vectors was formed. An estimator was then trained by exploiting GP-based automatic feature selection ability, to select and combine useful features. The proposed technique was compared with Richardson–Lucy deconvolution and Wiener filtering approaches, and comparatively good results were reported in terms of $\text{RMSE}$ and $\text{PSNR}$.

5.2.2 | GP in image denoising

Many researchers used GP as an effective strategy to remove noise from an image.\textsuperscript{45–51} Chaudhry et al.\textsuperscript{46} proposed GP for restoring degraded images by evolving an optimal function that estimated pixel intensity. Their technique was a hybrid of GP and fuzzy logic, which denoises gray level Gaussian noise images in the spatial domain. First, for deciding if a pixel needed to be rigged, a mapping function based on fuzzy logic was used. Then, GP was applied to evolve an optimal pixel intensity-estimation function.

Another denoising method based on local-adaptive learning (for Gaussian and salt & pepper noise) method was reported by Yan et al.\textsuperscript{47} In the training stage of their process, clustering was used to classify the image based on similar local structures. Then GP was applied to determine optimal filters (which themselves were tree-like individuals) for each cluster. The function set was composed of Gaussian and bilateral filters, as well as arithmetic operators. An increased PSNR was reported in comparison to other local learning-based methods such as K-clustering with singular-value-decomposition.

On the other hand, to remove Rician noise from MRI, an optimal composite morphological filter was generated via GP.\textsuperscript{48} In their method, a GP individual performs morphological operations on the corrupted image to obtain an observed copy. RMSE of the feature sets for the degraded image and the observed image was used to calculate the fitness of each individual. For evaluation, a noisy image was filtered by the developed filter to obtain an estimated copy. Moreover, their method (in terms of RMSE and PSNR) was also compared with other techniques.

Another work for removing mixed/Gaussian noise using GP was reported by Petrovic and Crnojevic.\textsuperscript{49} GP-based two-step filter (each having its estimator), was used to remove the noisy pixels missed by the first detector.

Harding\textsuperscript{50} used Cartesian GP to evolve image filters and evaluated their fitness functions on a graphics processing unit (GPU). The average error on each pixel was used as the fitness score.
Majid et al.\textsuperscript{51} employed GP to estimate optimal values of noisy pixels for impulse noise removal. Noisy pixels were detected first using the directional derivative; then, their costs were estimated using GP estimator by incorporating noise-free pixels. Feature vectors were constructed using noisy pixels with at least three neighboring noise-free pixels. Recently, Hernandez-Beltran et al.\textsuperscript{52} used a GP-based restoration technique to remove haze from images. During training, GP-based estimators were evolved based on mean-absolute-error (MAE). Beltran’s method showed significant performance improvement when compared with the latest techniques. However, the quality of restored images was only evaluated against MAE and PSNR metrics.

5.3 GP in image registration

Image registration involves matching different images of the same scene, which are captured at various intervals, from different directions or by different sensors. One objective of image registration is to bring into line the images in such a way so that high-level processing can be executed.

Only a few researchers have employed GP for image registration. Chicotay et al.\textsuperscript{53} presented GP-based approach for massive size image registration, in which transformation $T$ on an image mapped every pixel of the input image to a different pixel in the coordinate system of the referenced image. Mutual information was used as a measure to search for a function that generated the highest value when there existed a maximum overlap between the referenced and the transformed image. RMSE was used to evaluate each individual. A comparison was made with the scale-invariant feature transform (SIFT)\textsuperscript{54} based image registration. Though the results were not as good as a SIFT-based technique, they were still comparable, keeping in view that unlike the SIFT-based method, their technique did not make any assumptions about the transformation model to initiate or bound the registration process. The function set included transformation functions such as sine, cosine, power, rotation, and radial basis function.

Langdon et al.\textsuperscript{55} employed GP optimization to improve GPU-based implementation of Nifty Reg Software. Whereby the Nifty Reg is open-source software for medical image registration, and the optimization was performed for six different graphics cards. The implementation was carried-out using compute unified device architecture (CUDA). GP with linear variable-length genome specified changes to the CUDA kernel. Two parameters (compute level and size of the block) for CUDA were also tuned along with postevolution bloat removal. Each genome was saved as a text line. Crossover and mutation were prohibited from including code lines to those parts of the kernel, where the containing variables might go beyond the scope. For each generation, a new image was created randomly, and each GPU kernel was run on it. GPU kernel generated an answer which was checked against that of the CPU. And the runtime was compared with that of the original Kernel.

Outliers within data significantly degrade the performance of a classifier. To overcome such degradation in the performance of an image registration classifier, Hyun and Tariq\textsuperscript{56} reported a novel GP-based method. In their approach, first, feature extraction was performed using SIFT.\textsuperscript{54} The features were then classified into three categories, that is, inliers, outliers, and nonclassified features. Inliers and outliers extracted from the first phase were provided as training data to GP. GP then categorized the nonclassified features into two groups, that is, inliers and outliers. All the outliers were removed from the dataset. And the image registration was performed on the preprocessed data (after outlier removal). The Block diagram of Lee’s technique is shown in Figure 6.
5.4 | GP in image compression

The increasing use of images and their storage requirements initiated the need to compress them. The basic idea behind image compression is to remove redundant bits, and thus encode the information contained in the image so that while restoring, the encoded image is obtained without considerable loss. Restoring the exact image is vital in case of medical diagnosis or other security forensics. Transmitting images over the internet also requires compression to consume less bandwidth.

Fukunaga and Stechert\textsuperscript{57} described a GP system for lossless image compression, which learned a nonlinear model for pixel prediction based on a pixels. Four adjacent pixels were used as terminals for the GP. For each image, a unique model was generated and was represented as \textit{s-expression}. The high computational cost of evaluating the \textit{s-expression} for each pixel was overcome by removing function call overhead by employing the Genome Compiler. This compiler translates \textit{s-expressions} into efficient SPARC machine code before execution. The proposed method was compared with other compression techniques, including CALIC, LOCO-I, gzip, and was reported to be superior in the compression achieved, though it was slow. Figure 7 depicts the steps of Fukunage’s method.

In another technique, Parent et al.\textsuperscript{37} proposed lossless compression of medical images. They used a linear GP driven by compressed GA (cGA) and found a transformation, represented as \textit{T(d)}, which improved the compression ratio of data \textit{d}. Moreover, this transformation could remove certain types of redundancy. In Parent’s technique cGA identifies collection of genes in the population which are important. The terminal set comprised of constants, and the function set included four transformation functions. These transformations acted as preprocessing
before real compression and yielded enhanced compression as compared with standard GA-based techniques.

5.5 GP in image segmentation

The primary purpose of image segmentation is to segment out different gray levels of an image. If the pixels belonging to regions are homogeneous, then they are assigned the same label. Otherwise, various labels are attached. In other words, a good segmentation criterion is required to look for homogeneity within-region and heterogeneity between regions.58

Developing a comprehensive way to check the accuracy of image segmentation algorithms is a significant problem. In the field of image processing, GP has been widely used to segment region of interest from images.59–68 In 1996, Poli developed filter for detecting enhanced features for image segmentation.69 Vojodi et al.62 used GP to combine different and unrelated evaluation measures. They selected three evaluation measures, which are based on the layout of entropy, similarity within the region, and disparity between the areas for the creation of composite evaluation measures.

In another technique, Song and Ciesielski63 used GP to evolve automatic texture classifiers, which were then used for texture segmentation. As opposed to conventional methods, their method does not require the manual construction of models to extract texture features because the classifier’s input is raw pixels instead of features. In addition, the conventional methods are not universally applicable as they rely on the knowledge of the nature of texture, which may differ from region to region and image to image.

GP can capture variation within images; that is why GP is popular in evolving a suitable image segmentation technique. However, GP-based techniques mainly developed vast and expansive segmentation algorithms. In this regard, Liang et al.66 proposed a MOGP-based segmentation technique, in which classification accuracy and program complexity are included within the fitness function. Liang’s method evolved a suitable solution with an optimal tradeoff between accuracy and program complexity. In another approach, the GP-based segmentation technique developed an accurate and reliable figure-ground segmentation.67 Their segmentation approach was evaluated against four different datasets.

Similarly, another segmentation technique was reported that used strongly type GP and used two-stage during the GP evolution cycle.68 Dong et al.64 attempted to categorize the texture within an image to be either corpora lutea (CL) (i.e., an endocrine gland that is generated from the follicular tissue after ovulation) or non-CL, based on local neighborhoods. A 16-bit invariant uniform local binary patterns (LBP) histogram of pixels in the neighborhood was formed to represent texture descriptions. Feature vector created by the histogram bin values was fed as input to GP. GP was used to train a classifier for distinguishing between CL texture and other textures. For segmentation, a sliding window was used to scan the image in raster order. The GP classifier then assigned each image pixel in the window, a class label. Majority voting was used in the case of multiple tags. For CL detection, properties related to the set of the region were computed for each image’s output region. Then a GP classifier was trained using these properties. Finally, the classifier was used to detect whether the segmented part of an image is a CL or not.

To address the tradeoff between localization accuracy (requiring a small window) and noise rejection (requiring large window) posed by selecting the window size, Fu et al.70 used GP to automatically search discriminating pixels and their neighbors for constructing edge detectors. Rather than using a set of pixels from a moving window, GP used a full image. The selected pixels were
then used to form linear and nonlinear filters for detecting edges. The parameters of these filters were estimated via a hybrid of particle-swarm-optimization (PSO) and differential evolution. A shifting function, representing four directional shifting functions, was included in the function set. A comparison was made with other detectors showing good results for GP-based detectors. They employed $F$-measure to evaluate the accuracy of the detectors. Similarly, another GP-based image segmentation technique for extracting regions of interest from the background was proposed by Liang et al. Feature selection using GP was used to find out the useful features that helped to segment out the desired region of interest. Three different types of GP-based feature selection methods were proposed. In all of the three ways, fitness function within GP was either based on a single or multi-objective method. Their experimental results showed that the GP-based feature selection, which used multiobjective fitness function, improved the performance of the classifier, and also reduced the computational complexity. The block diagram of Liang’s technique is shown in Figure 8.

5.6 | GP in image retrieval

Due to the decline in the prices of image acquisition devices and the development of efficient image processing algorithms, the image databases are increasing in number. Consequently, it has become inevitable to design effective and fast methods for retrieving desired images from such significant collections. There are different techniques for image retrieval, such as associating some metadata (tags, keywords) with the images, or using content-based retrieval, which is based on similarities of the contents of the given image (or feature) and the desired image. Different shapes, textures, colors, and so forth, can be used as features for Image retrieval related tasks.

In an exciting work by Torres et al., GP was applied for creating a merged similarity function for content-based image retrieval. They showed that features could be combined from multiple feature vectors, or weights can be assigned based on image similarities. In the case where combining images gets more complicated, then the GP is used for combining nonlinear image similarities. The resulting composite descriptor was simply a combination of predefined descriptors. This GP-based composite descriptor combined the similarity values obtained from each descriptor and then produced a more effective similarity function.

Ciesielski et al. used a segmentation algorithm based on texture-versus-all-else classifiers. These classifiers were evolved by GP to retrieve from an extensive heterogeneous collection of images.

Calumby et al. used GP to iteratively combine multimodal similarity measures, such as those extracted from text and content, to generate new similarity functions that would fit the user preferences. For each discovered function, the evaluation returned a measure of quality that was based on how well that function ranked the training set objects. The proposed method showed higher
efficiency when compared with image CLEF photographic retrieval task. A somewhat similar framework was also reported by Ferreira et al. on the other hand, used GP to combine multiple textual sources of evidence, such as image file name, the content of HTML, page title, alt tag, keywords, description, and text passages around the image, to rank web-based image retrievals.

### 5.7 GP in image classification

Image classification is the process of classifying images based on the visual contents. various artificial intelligence-based technologies, such as ANNs and fuzzy systems, have been applied to develop autonomous classification algorithms and have shown promising results. Two broad families of machine learning approaches used in image classification are parametric (that requires learning phase) and nonparametric methods (that does not require learning phase). Some examples of parametric classifiers are support vector machine (SVM), decision trees, and GA. Whereas, nearest-neighbor image classifier is an example of nonparametric classifiers. When GP is used for classification, the inputs are features, and the output is a mathematical expression that returns different values for different classes.

In literature, GP is one of the bioinspired model for performing object classification. Using GP for classification requires a threshold to be set for the program output to specify different classes. In the case of static range selection, boundaries of program output space are fixed and predefined. However, in dynamic range selection, the boundaries are searched automatically. In centered dynamic range selection, the class boundaries are dynamically determined by calculating the center of the program output values for each class. In the slotted dynamic class boundary determination method, the output value of a program is split into many slots. Each slot is assigned to a value for each class. It then dynamically determines the class by only taking the class with the highest value at the slot. Several techniques have used GP for classification. Nandi et al. used GP for feature selection to classify breast masses in mammograms into benign and malignant groups. To narrow down the pool of features, they used a few procedures such as sequential forward selection, student’s t-test, and so forth. Once important features were selected, these were divided into two groups. Either union or intersection operation was performed over these groups to create a new set of data points for the GP classifier.

Similarly, Kobashigawa et al. showed that with the increase in problem difficulty level, GP achieves better results than ANN methods. Kobashigawa’s work also revealed the robustness of GP to unseen examples along with an inherent capability of optimal global searching, which could minimize efforts that are required during training processes. On the other hand, Smart and Zhang employed the evolutionary process of GP to dynamically determine the boundaries between images of coins having different denominations. Pixel level domain-independent statistical features such as average intensity, variance, and so forth were given as input to GP to automatically select features that were relevant to this multiclass image classification problem. As compared with a static range selection, reasonably good results were reported using the dynamic methods, centered dynamic range selection, and slotted dynamic range selection.

Similarly, Atkins et al. proposed a GP-based domain-independent technique for extracting features and image classification. The Block diagram of Atkins’s approach is shown in Figure 9. First raw images were preprocessed by the filtering layer whose outputs (the filtered images) were fed to the second layer, called the aggregation layer. The aggregation layer then performed feature aggregation and produced a real value. Finally, the output of the aggregation layer was passed on
to the classification layer to perform classification. For this layer, a threshold of zero was used, so a negative output would mean class A and nonnegative would classify the image as belonging to class B. The proposed procedure was tested on four different datasets, and the reported results suggested that it outperformed the basic GP methodology with increasing problem difficulty.

In another approach, Al-Sahaf et al.\textsuperscript{95} presented a GP-based approach that extended the work of Atkin’s et al.\textsuperscript{94} and introduced aggregation functions that read in different shapes such as lines, circles, and rectangles to enable sampling windows that were not square. They did not use the filtering layer and still achieved better results as compared with a canonical GP that used extracted features and performed classification by the three-tier GP. Guo and Nandi\textsuperscript{96} used a modified fisher criterion-based GP (MF-GP) for generating features. The generated features were evaluated for their discriminating ability by the minimum distance classifier (MDC). Improved results were reported for MF-GP compared with multilayer perceptron, SVM, and alternative fisher criterion-based GP with MDC.

A semiautomatic approach for classifying remote sensing images (RSI) was proposed by dos Santos et al.\textsuperscript{97} GP was used to learn user preferences via user indicated relevant as well as nonrelevant regions. The image region descriptors were combined that encoded color and texture properties. The reported results showed that the method outperformed maximum-likelihood-classification when used for RSI classification. In the same way, dos Santos et al.\textsuperscript{98} improved the results of the previous work by combining optimum-path-forest (OPF) with composite descriptors obtained using the GP framework. OPF classifier represents each class of objects by one or numerous optimal-path trees rooted at essential samples, called prototypes. The OPF-based classification system took into account user interaction.

Choi and Choi\textsuperscript{99} proposed a system for automatic detection of pulmonary nodules, which first segmented the lung volume using thresholding, then detected and segmented nodule candidates using multiple thresholding and rule-based pruning. From these nodules, geometrical and statistical features were extracted, and a GP-based classifier was trained. The fitness function was constructed by combining the AUC-ROC curve, the true-positive-rate (TPR), and specificity. They reported that compared with the previously proposed methods for this application, this GP-based classifier showed high sensitivity and a reduced false-positive rate. Zhang and Smart\textsuperscript{100} developed fitness function for classification based on probabilities (derived from Gaussian distribution) that are associated with different classes. Two fitness functions (using the overlapped region and weighted distribution distance) were developed, assuming the outputs from different classifiers

---

**FIGURE 9** Three tier genetic programming for image classification\textsuperscript{94}
as random independent variables. Zhang’s approach exploited many top GP programs for classification, and the class with the highest probability was used as the class of the object pattern. In comparison to a primary GP classification, which also used multiple best programs and voting, the proposed technique was reported to have good results in terms of classification accuracy and execution time. Recently, La Cava et al. proposed GP-based multiclass classification technique in which transformation is performed against multidimensional feature space. La Cava’s technique was compared with different techniques related to different domains. Results showed that the proposed solution could scale well to those problems that have high feature dimensions. Similarly, Burks and Punch proposed a GP-based classification technique for detecting tuberculosis from X-ray images. Unlike traditional classification methods, no preprocessing and segmentation steps are needed before the training of the GP-based classification approach. Moreover, Burks’s technique needs less training time in comparison to other traditional proposed methods.

5.8 GP in image watermarking

The consistently broader use of information technology demands the protection of information. Therefore, in this regard, digital watermarking is used as a promising technique to overcome the issues related to the protection of information, especially for the authentication of medically related information. One of the measures to evaluate the quality of the watermarked image is to evaluate its imperceptibility. In watermarking techniques, imperceptibility shows that the visual appearance of the watermarked image should be close to the original image. However, when more information (payload) is embedded in the image, it causes distortion in the original image. That is why there is a tradeoff between imperceptibility and payload. In the past, many GP-based watermarking techniques have been proposed for the development of efficient and reliable watermarking systems. To increase the robustness and imperceptibility in digital image watermarking, GP was employed by Golshan and Mohamadi. Instead of setting the Perceptual Shaping Function (PSF) to a constant function, GP was utilized to develop an intelligent PSF. A fitness function based on both robustness and imperceptibility was used to evaluate the performance of each PSF individual. Similarly, Golshan and Mohammadi used a hybrid approach of GP and PSO for the same purpose. In technique, developed by Gilani et al., GP was used to estimate the distortion within the distorted watermarked signals. Both the watermarked and the distorted watermarked signals were fed to a GP module. The best-estimated distortion function returned by GP was then applied to the original watermarked signal. Varying strengths of Gaussian and JPEG compression attacks were tested for the proposed technique.

Similarly, Usman and Khan proposed evolving application specific visual tuning function (VTF), in which GP optimizes the balance between imperceptibility and robustness while processing an 8 × 8 block of discrete cosine transform (DCT) image. The watermark was structured according to the human visual system (HVS) and a cascade of attacks. VTF is given as:

\[ a_G(k_1, k_2) = f \left( X_{0,0}, X(i,j), \alpha(i,j) \right), \]

where \( X_{0,0} \) is the discrete cosine coefficient and signifies dependency of VTF on luminance sensitivity, \( X(i,j) \) is AC coefficient and symbolizes dependency of VTF on contrast masking, and \( \alpha(i,j) \) shows frequency sensitivity. The current value of Watson’s VTF, DC, and AC (DCT) coefficients of
8 × 8 blocks were provided as variable terminals. Each potential VTF was evaluated for imperceptibility related fitness, whereas for robustness, bit correct ratio represented an objective measure. Test images were then watermarked with the evolved VTF.

To select the watermarking level, Jan et al.\textsuperscript{108} proposed GP-based approach. Coefficients were selected using a 32 × 32 block, whose discrete wavelet transform was obtained. Luminance, contrast, and noise-visibility-function (NVF) were used as terminals for GP trees. Watermarking level was given by:

\[
\alpha = f(lum(i, j), \text{cont}(i, j), \text{co}(i, j), \text{NVF}(i, j)),
\]

where co is a selected coefficient, cont is contrast, and lum is luminance. Robustness against different attacks was reported, whereas to check the imperceptibility of the watermark, mean square error (MSE) and PSNR were used. Similarly, Abbasi and Seng\textsuperscript{109} used a similar approach but used a block size of 4 × 4. Khan et al.\textsuperscript{110} presented a DCT-based watermarking system that employed GP for finding optimal perceptual shaping function according to HVS. Each GP tree represented a perceptual shaping function, which was evolved to embed high strength watermark in areas of high variance and low strength watermark in areas of low variance. Change in a local variance of the watermarked image with respect to the original image was used as a fitness function. This technique was tested for JPEG compression and Gaussian noise. Recently another interesting reversible watermarking technique based on GP for the protection of medical-related information was proposed by Arsalan et al.\textsuperscript{103} The Block diagram of Arsalan’s technique is shown in Figure 10. First, the histogram modified image was formed after the preprocessing of the original image. Integer wavelet transform (IWT) was then applied to the histogram modified image. After applying IWT, GP was used to find out the coefficients within the wavelet domain for the purpose of embedding watermark. The aim of the proposed GP-based intelligent watermarking scheme was to produce a watermarked image having low distortion and high payload.
5.9 | GP in object detection

Object detection is the task of finding different types of objects belonging to different categories and is a challenging task especially, in the field of image processing and computer vision. In the field of image processing, GP has been used by many researchers for accurate and efficient prediction of objects from cluttered and noisy scenes or images. In a review article, Krawiec et al. analyzed the applications of GP in object detection related applications.

Howard et al. utilized GP to evolve detectors to detect ships in synthetic aperture radar imagery. Terminal nodes were real numerical values derived from random constants or pixel statistics. A value greater than zero was decided to be a target detection, while a value of zero or less was for ocean pixel. In Lin and Bhanu approach, GP was used to synthesize composite operators and features from primitive operations and features for object detection. A composite operator was applied to primitive feature images; the output was segmented to obtain a binary image and was used to extract the target object from the original image. The size of a composite operator, as well as misclassified pixels, were taken into consideration, while the fitness function used in Lin’s technique was based on the minimum description length (MDL) principle. In another work, Bhanu’s and Lin have used a similar approach of composite operators, but instead of MDL-based fitness function, they used the following fitness measure:

$$\text{Fitness} = \frac{n(G \cap G')} {n(G \cup G')}$$

(3)

where, $G$ and $G'$ are foregrounds in the ground-truth and in the detected image, respectively, $n$ being the number of pixels in a given region.

Martin used GP to create algorithms for obstacle detection, which analyzes a domain to find its constraints. Lowest nonground pixels were manually marked, and these images were fed to GP, whose output was then compared with the ground truth images. A robot was then controlled by the best-evolved program.

Edges are detected traditionally by using local window filters; however, in Fu et al. work, GP was used for domain-independent global edge detection using the whole raw image as input. Different shifting functions were used along with other commonly used operators. $F$-measure was used in constructing the fitness function:

$$F_m = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{2r_ip_i}{r_i + p_i}\right),$$

(4)

where $i$ represents image, $N$ the number of images, $r_i$ and $p_i$ are the recall and precision of a given image. Better results were reported as compared with Laplacian and Sobel edge detectors.

In another work, Fu et al. used GP to evolve edge detectors. Instead of distributing a fixed size window into small areas based on different directions, it searched for features based on flexible blocks, and the fitness function was based on $F$-measure. Similarly, GP was also used for improving the performance of an edge detection system, where the fitness function was based on the accuracy of the training data. In another work by Fu et al., composite features were constructed for edge detection by estimating the observations of the programs evolved by GP as triangular distributions. Gaussian filter gradient, histogram gradient, and normalized standard deviation were used as a terminal set. In order to detect edges, an unsupervised GP system was
proposed in Fu et al.\textsuperscript{126} However, fitness function was based on the energy functions in the active contours. In comparison with a Sobel edge detector, the evolved GP edge detectors were reported to have better performance.

Similarly, Liddle et al.\textsuperscript{115} used a MOGP for object detection. MOGP evolves a set of classifiers rather than a single classifier, as in the case of single-objective GP. The proposed technique used the NSGA-II algorithm. A two-phase training process applied the MOGP algorithm twice using different objectives, for example, maximizing both TPR and true negative rate (TNR); or maximizing detection rate (DR) while at the same time minimizing false alarm rate (FAR). In the interesting work of Nag and Pal,\textsuperscript{89} GP was used for object detection, but instead of using raw pixels and terminals, they used pixel statistics such as mean, standard deviation, and moments. A new fitness measure termed as “false alarm area” was used along with a combination of DR and FAR.

On the other hand, Zhang et al.\textsuperscript{113} presented domain-independent features such as mean and standard deviation as terminals for GP to detect multiple objects. They used three different ways (rectilinear: based on different rectangles; circular: using circles of different radii, and using an average of pixels) for obtaining pixel statistics. Evaluation of programs was performed with a fitness function based on DR and FAR as such:

\[
\text{fitness} = K_1 \cdot (1 - \text{DR}) + K_2 \cdot \text{FAR}, \tag{5}
\]

where, \(K_1\) and \(K_2\) are constants. Zhang\textsuperscript{117} introduced a two-phase GP approach for object detection. In the first phase, cutouts from the training images were used with classification accuracy as the fitness function. The second phase was initialized with the population from the first phase, and a window was moved over the whole image. For the second phase, the following fitness function was used:

\[
\text{fitness} = K_1 \cdot (1 - \text{DR}) + K_2 \cdot \text{FAR} + K_3 \cdot \text{FAA} + K_4 \cdot \text{size}, \tag{6}
\]

where FAR is the false alarm rate, DR stand for detection rate, FAA is the false alarm area (positive classifications – objects in the image), \textit{size} is the program size, while \(K_1\), \(K_2\), \(K_3\), and \(K_4\) are constants.

Hunt et al.\textsuperscript{118} followed the previous two-phase approach,\textsuperscript{117} augmented with validation and sampling methods in order to avoid overfitting. Validation was performed after every two generations. Generalization ability is usually evaluated by calculating the hyperarea (area covered by the best Pareto-front), and distance (the difference between the performance of the classifier on training and validation set). Puente et al.\textsuperscript{128} used GP to generate vegetation indices for the detection of dry, dead, and dry vegetation. Similarly in another approach GP is used for designing vegetation indices for estimating soil erosion model.\textsuperscript{129}

Nguyen et al.\textsuperscript{130} used GP for the detection of rice leaf. In Nguyen’s work, the dataset was created by taking images from the top of the rice field, and a total of 600 images of size 640 \(\times\) 840 were captured from the camera. Out of the total 600, 300 images were used for the training of the classifier. After capturing images, the next step was the conversion of color images into grayscale. In order to deduce the positive and negative samples from the set of gray images, a window size of 20 \(\times\) 20 pixels was used to extract subregions within the images. If each subimage contained a portion of rice leaf then, it was labeled as a positive example; otherwise, the negative label was assigned to that subpart. After preprocessing of original images, a total of 9000 images of size 20 \(\times\) 20 pixels was generated in which half belonged to a positive class,
and half belonged to the negative class. For training of GP program, pixels were considered as a terminal set, whereas the function set was comprised of four different arithmetic operators and a square-root function. The weighted sum of TPR and TNR was used as a fitness criterion. In order to Reference 130 ensure that the value of fitness was between 0% and 100%, the following constraint was followed $w_1 + w_2 = 1$. The block diagram of Nguyen’s technique is shown in Figure 11.

### 5.10 GP in motion detection

In the past, many modeling and background subtraction related techniques have been designed for motion detection. Moreover, to avoid manually coded motion detection systems, different researchers used GP-based automatically evolved systems. It was observed that generally, the GP-based evolved programs outperformed manually coded programs. To tackle the unstable background (such as rainy background, moving background due to a moving camera) in motion detection, GP was employed in Reference 137, where classification accuracy based on motion and nonmotion was used as a fitness measure.

Another difficult task in the case of motion detection is to detect motion from a noisy scene when there is no information about the noise. Pinto and Song tackled this problem by using GP-based approach in which motion detectors were generated during the testing phase on the basis of the fitness function. In this approach, Gaussian noise was added in the video and showed better results for detecting motion in different environments. In another work, the GP program was used for analyzing the various type of motion detection techniques such as detecting simple motion, detection of fast-moving objects, motion detection from a noisy background. Another advantage of using GP for motion detection is that the evolved detectors can also tolerate noise; that is why GP may be considered as one of the best approaches for the detection of motion.

Similarly, Xie and Shang used GP for anomaly detection from crowded scenes. In Xie’s approach, multiframe LBP difference based on LBP was used for extracting features from video frames. Training of GP was performed on extracted features. The proposed scheme detected abnormalities in real-time videos. Similarly, Song and Zhang proposed GP-based target motion detection approach that automatically evolved GP program and separated target motion from other irrelevant motions such as the noisy background. The technique proposed by Song et al.’s was comprised of two phases. In the first phase (evolution phase), the data used during training was divided into training and test parts. Parameter optimization during training was performed.
on the basis of the performance of GP-based evolved programs on test data. After the evolution of the GP program, the next phase was the application phase, in which the best-evolved GP program from the evolution phase was used to check the performance on unseen data samples. The block diagram of Song’s technique is shown in Figure 12. This technique was used for detecting motion from the video, so the first two-dimensional array of size $20 \times 20$ was captured as video frames from different locations of videos. The image was assigned to the positive class if the majority of pixels within the frame were labeled as sampled by a human expert. During the GP training, program accuracy was used as a fitness function, whereas detection accuracy versus the number of generations was used as an evaluation measure. In literature, some other interesting techniques related to brain programming for object tracking have also been reported.\textsuperscript{140,141} As brain programming applies concepts of neuroscience like the two-stream hypothesis. Therefore, the models develop dorsal and ventral streams to create an artificial visual cortex, optimized with GP.

## 6 | CATEGORY WISE APPLICATIONS OF GP

This section presents different GP-based techniques that are applied to different categories of image processing. Table 1 lists the references as well as the GP parameter settings for each category. An overall analysis of Table 1 shows that in all of the reported image processing related applications, a large population is used in comparison to the number of generations. The large population within each generation helps to increase diversity and hence increases the chance to obtain better individual in less number of generations. Moreover, most of the GP related image processing applications used tournament selection. The advantage of using the tournament selection method is that it helps to maintain constant selection pressure, and even programs with average fitness have chances to reproduce a child in the coming generation. Table 1 shows that a higher crossover probability is used in comparison to mutation probability because higher values of mutation probability increase the search area within the search space, and the algorithm may get stuck in local minima. In addition, in image processing related applications, ramped half and half is the commonly used population initialization method. This method produces the initial tree of variable length and thus help to increase the diversity of the initial population. The last column of Table 1 shows that several runs are carried out in most of the reported works to show the effectiveness of proposed methods.
| Category       | Reference | Number of generations | Population size | Selection method | Mutation rate | Crossover rate | Population initialization methods | Runs |
|----------------|-----------|-----------------------|-----------------|------------------|---------------|----------------|-----------------------------------|------|
| Enhancement    | 41        | 100                   | 4096            | –                | 0.25          | 0.5            | Random growth                     | –    |
| Restoration    | 47        | 50                    | 500             | Tournament       | 0.05          | –              | Ramped half-and-half              | –    |
|                | 48        | 200                   | 200             | Lexicographic    | Variable      | –              | Ramped half-and-half              | –    |
|                | 49        | 300                   | 100             | –                | –             | –              | –                                 | –    |
|                | 50        | –                     | 50              | Tournament       | 0.05          | Not used        | –                                 | –    |
|                | 51        | 500                   | 50              | Tournament       | Variable      | Variable       | Ramped half-and-half              | –    |
| Registration   | 53        | –                     | 150             | Elitism          | 0.3           | 0.9            | –                                 | –    |
|                | 55        | 500                   | 300             | 50% truncation   | 0.5           | 0.5            | Random single mutants             | –    |
|                | 56        | 50                    | 300             | Tournament       | Variable      | Variable       | Ramped half-and-half              | –    |
| Compression    | 57        | 30                    | 500             | Tournament       | Not used      | 0.9            | –                                 | –    |
|                | 37        | 50                    | 500             | Fitness proportional | 0.05 | 0.8            | Random                           | 100  |
| Segmentation   | 70        | 200                   | 500             | –                | 0.15          | 0.8            | –                                 | –    |
|                | 62        | 25                    | 100             | Tournament       | 0.2           | –              | Ramped half-and-half              | –    |
|                | 64        | 500                   | 600             | Tournament       | 0.1           | 0.8            | Random                           | Threefold |
|                | 65        | –                     | 500             | –                | 0.1           | 0.9            | –                                 | –    |
| Retrieval      | 72        | 150                   | 200             | Fitness proportional | 0.0  | 0.9            | –                                 | –    |
|                | 71        | 25                    | 600             | Tournament       | 0.25          | –              | Ramped half-and-half              | –    |
|                | 73        | 20                    | 60              | Tournament       | 0.2           | 0.8            | –                                 | –    |
|                | 75        | 10                    | 60              | Tournament       | 0.2           | 0.8            | Ramped half-and-half              | 10   |
|                | 76        | 30                    | 300             | Tournament       | 0.05          | 0.9            | Ramped half-and-half              | –    |

(Continues)
| Category          | Reference | Number of generations | Population size | Selection method       | Mutation rate | Crossover rate | Population initialization methods | Runs |
|-------------------|-----------|-----------------------|-----------------|------------------------|---------------|---------------|-----------------------------------|------|
| Classification    | 93        | 500                   | –               | –                      | –             | –             | –                                 | 100  |
|                   | 77        | 30                    | 500             | –                      | –             | –             | Random                            | –    |
|                   | 82        | 50                    | 300             | Fitness proportional   | 0.3           | 0.5           | Ramped half-and-half              | 10   |
|                   | 88        | 50                    | –               | Tournament             | 0.2           | 0.8           | Ramped half-and-half              | 30   |
|                   | 89        | –                     | –               | Roulette wheel         | 0.3           | 0.8           | Ramped half-and-half              | 10   |
|                   | 90        | –                     | 100             | Tournament             | 0.19          | 0.80          | Ramped half-and-half              | 30   |
|                   | 94        | 50                    | 1024            | Tournament             | 0.29          | 0.8           | –                                 | 40   |
|                   | 95        | 50                    | 1024            | Tournament             | 0.19          | 0.8           | Ramped half-and-half              | 30   |
|                   | 96        | –                     | –               | –                      | –             | –             | Random                            | –    |
|                   | 97        | 10                    | 60              | Tournament             | 0.2           | 0.8           | Ramped half-and-half              | –    |
|                   | 99        | 80                    | 300             | Generational           | Variable      | Variable      | Ramped half-and-half              | –    |
|                   | 100       | 51                    | 500             | Fitness proportional   | 0.3           | 0.6           | Ramped half-and-half              | 50   |
| Watermarking      | 103       | 50                    | 25              | Roulette               | Variable      | Variable      | Ramped half-and-half              | –    |
|                   | 104       | 100                   | 10              | Keep best              | 0.1           | 0.9           | –                                 | –    |
|                   | 105       | 100                   | 10              | –                      | 0.1           | 0.9           | –                                 | –    |
|                   | 106       | 32                    | 120             | Tournament             | Variable      | Variable      | Ramped half-and-half              | –    |
|                   | 107       | 40                    | 160             | Tournament             | Variable      | Variable      | Ramped half-and-half              | –    |
|                   | 108       | 10                    | 25              | –                      | –             | –             | –                                 | –    |
|                   | 110       | 30                    | 300             | –                      | –             | –             | Ramped half-and-half              | –    |

(Continues)
| Category          | Reference | Number of generations | Population size | Selection method  | Mutation rate | Crossover rate | Population initialization methods | Runs |
|-------------------|-----------|-----------------------|-----------------|-------------------|---------------|----------------|-----------------------------------|------|
| Object detection  | 114       | 40                    | 1000            | Tournament        | Not used      | Unknown        | Ramped half-and-half              | –    |
|                   | 119       | 70                    | 100             | Tournament        | 0.05          | 0.6            | –                                 | –    |
|                   | 121       | 51                    | 4000            | Tournament        | –             | –              | Ramped half-and-half              | –    |
|                   | 122       | 250                   | 200             | –                 | 0.15          | 0.8            | –                                 | 30   |
|                   | 123       | 200                   | 600             | –                 | 0.15          | 0.8            | –                                 | 30   |
|                   | 124       | 200                   | 500             | Elitism           | 0.15          | 0.80           | –                                 | 30   |
|                   | 125       | 200                   | 200             | –                 | 0.15          | 0.8            | –                                 | 30   |
|                   | 126       | 100                   | 30              | –                 | 0.3           | 0.65           | –                                 | 30   |
|                   | 115       | 60                    | 500             | Tournament        | 0.3           | 0.7            | –                                 | 40   |
|                   | 120       | 70                    | 100             | Tournament        | 0.05          | 0.6            | –                                 | 10   |
|                   | 117       | –                     | –               | –                 | –             | –              | –                                 | –    |
|                   | 113       | 100, 150, 150         | 100, 500, 700   | Fitness proportional | 0.25       | 0.70           | Ramped half-and-half              | 10   |
|                   | 118       | 20, 40                | 500             | Tournament        | 0.30          | 0.70           | Ramped half-and-half              | 40   |
| Motion detection  | 132       | 300                   | 30              | Elitism           | 0.05          | 0.85           | –                                 | –    |
|                   | 133       | 300                   | 30              | –                 | 0.05          | 0.85           | –                                 | –    |
|                   | 134       | 70                    | 200             | Lexicographic parsimony pressure | 0.1 | 0.9 | Ramped half-and-half | – |
|                   | 135       | 200                   | 200             | Elitism           | 0.1           | 0.85           | –                                 | –    |
|                   | 136       | 100                   | 50              | Tournament        | Adaptive       | Adaptive       | Ramped half-and-half              | 25   |
|                   | 137       | 300                   | 30              | –                 | 0.05          | 0.85           | –                                 | –    |
|                   | 138       | 30                    | 200             | Generational      | 0.05          | 0.85           | –                                 | –    |
|                   | 139       | 300, 150              | 30              | Elitism           | 0.1           | 0.85           | –                                 | 10   |
|                   | 140       | 30                    | 400             | Tournament        | 0.1           | 0.4            | Ramped half-and-half              | –    |
|                   | 141       | 15                    | –               | –                 | –             | –              | –                                 | 6    |

(Continues)
7 | ADVANTAGES AND DISADVANTAGES OF USING GP FOR IMAGE PROCESSING

GP is a relatively new technique among all the evolutionary computing algorithms and has been widely applied in various image processing related techniques. In literature, GP has shown excellent performance for optimization and classification related problems; however, advantages and disadvantages are also associated with GP-based optimization techniques. Some of which are discussed below.

7.1 | Advantages

**Understandability:** GP outputs a program or a collection of programs in the form of mathematical expressions, which are easy to comprehend if simplified and converted to normal notation.

**GP versus GA:** Being the prominent types of evolutionary algorithms, both the paradigms share some characteristics but differ in others. They mainly differ in the way individuals are represented. GP uses a tree representation, whereas GA uses a string representation. In the case of GA, individuals are generally raw data, whereas, in GP, the individuals are mathematical expressions. The tree-based representation gives GP an edge over GA because of its flexibility; however, GA is faster compared with GP.

**Diverse search space:** Genetic operators (crossover and mutation) used in GP introduce diversity and thus increases the span of search space. Larger search space helps in finding the most optimal solution for the problem at hand.

**Small testing or execution time:** The GP needs considerable time for training an optimal GP-based classifiers, but the finally selected GP tree needs very short execution time during the test phase. As GP-based classification requires the small duration of test time, therefore, such systems are suitable for those applications in which appropriate time is available during training, and a short time is required during testing.

**Flexibility of GP fitness function:** Another advantage of using GP is that its fitness function is flexible and can be adjusted or designed according to the problem at hand. Moreover, multiobjective fitness functions are mostly used in image processing related tasks.

7.2 | Disadvantages

**Computational cost:** Fitness of each individual/program in the population is evaluated after every generation; therefore, the training process usually takes a long time. This shortcoming is considerably mitigated by recent advancement in CPU speed and number of cores, especially by using GPU.

**Needs large training data:** A large dataset is needed for the training process in order to reach an optimal solution.

**No guaranteed solution:** Due to the stochastic nature, GP does not guarantee an optimal solution; therefore it could not guarantee the best solution to the problem in hand.
8 | CONCLUSION

This work presented a detailed study of the various image processing applications of GP. The automatic problem-solving capability of GP and increasing demand for image processing in a variety of fields has prompted researchers to look for efficient, robust, and cost-effective intelligent techniques. Moreover, due to the different nature of the image processing tasks, no hard and fast rules can be set. In addition, the terminal and function sets need to be problem tailored, and different fitness measures have to be developed. In addition, by incorporating the domain knowledge related to the image processing field, GP is able to handle complex image processing tasks. In this article, the application of GP in image processing related applications, different features of GP such as terminal and function set, fitness function, and other related parameters are discussed. In addition, the advantages and disadvantages of applying GP in image processing are discussed. Below are our observations related to applications of GP in image processing:

- In most of the applications of GP in image processing, large population size, and crossover probability are used in comparison to the number of generations and mutation probability, respectively.
- In image processing related applications, the terminal set of GP is mostly set according to some statistical features related to the image.
- Tournament and ramped half-and-half methods are used as a selection and population initialization method in most of the reported works.
- Selection of fitness function for a particular image processing application is the most important part and should be set in consultation with the expert of that image processing application.
- As parameter setting is also an important step in applying GP in any of the image processing related tasks. Before setting the GP parameters, a researcher must study and analyze the GP parameter settings in related image processing applications. This can help save time, whenever parameters of GP are needed to be set for any image processing related application.
- In literature, most of the reported work related to GP is oriented towards classification and object detection tasks.
- Relatively less work has been reported for image enhancement, registration, and compression, so more interesting techniques related to these fields can be exploited.
- Due to the heavy processing involved in image processing tasks, the algorithms require large training time. Training time can be considerably reduced by harnessing GPUs for enhanced algorithms.
- A GP-based ensemble is likely to better exploit the decision spaces of the individual classifiers.
- Recently, deep neural networks have shown remarkable performance in many image processing applications.\textsuperscript{145,146} In this regard, the meta classification/regression of individual learners, and specifically that of deep neural networks through GP, has good potential in learning complex problems.\textsuperscript{147}

ACKNOWLEDGMENT

This work is supported by the Higher Education Commission of Pakistan under Indigenous PhD Fellowship Program (PIN # 213-54573-2EG2-097).
REFERENCES

1. Gonzalez RC, Richard EW. Digital Image Processing. USA: Prentice-Hall, Inc; 2006.
2. Jain AK. Fundamentals of Digital Image Processing. Hoboken, NJ: Prentice-Hall, Inc; 1989.
3. Dimopoulos C, Zalzala AMS. Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. IEEE Trans Evol Comput. 2000;4:93-113. https://doi.org/10.1109/4235.850651.
4. Angeline PJ. Genetic programming: on the programming of computers by means of natural selection. Biosystems. 1994;33:69-73. https://doi.org/10.1016/0303-2647(94)90062-0.
5. Olague G. Evolutionary Computer Vision. Berlin/Heidelberg, Germany: Springer; 2016. https://doi.org/10.1007/978-3-662-43693-6.
6. Cagnoni S, Lutton E, Olague G. Genetic and Evolutionary Computation for Image Processing and Analysis. London, UK: Hindawi Publishing Corporation; 2007. https://doi.org/10.1155/9789774540011.
7. Olague G, Cagnoni S, Lutton E. Introduction to the special issue on evolutionary computer vision and image understanding. Pattern Recogint Lett. 2006;27:1161-1163. https://doi.org/10.1016/j.patrec.2005.07.013.
8. Cagnoni S, Lutton E, Olague G. Editorial introduction to the special issue on evolutionary computer vision. Evol Comput. 2008;16:437-438. https://doi.org/10.1162/evco.2008.16.4.437.
9. Nakane T, Bold N, Sun H, Lu X, Akashi T, Zhang C. Application of evolutionary and swarm optimization in computer vision: a literature survey. IPSJ Trans Comput Vis Appl. 2020;12:3. https://doi.org/10.1186/s41074-020-00665-9.
10. Woodward JR. GA or GP? that is not the question. Paper presented at: Proceedings of the 2003 Congress on Evolutionary Computation 2003 CEC’03; vol. 2; 2003:1056-1063. https://doi.org/10.1109/CEC.2003.1299785.
11. Dozal L, Olague G, Clemente E, Hernández DE. Brain programming for the evolution of an artificial dorsal stream. Cognit Comput. 2014;6:528-557. https://doi.org/10.1007/s12559-014-9251-6.
12. Olague G, Clemente E, Hernandez DE, Barrera A, Chan-Ley M, Bakshi S. Artificial visual cortex and random search for object categorization. IEEE Access. 2019;7:54054-54072. https://doi.org/10.1109/ACCESS.2019.2912792.
13. Olague G, Dozal L, Clemente E, Ocampo A. Optimizing an artificial dorsal stream on purpose for visual attention. In: Schuetze O, Coello CA., Tantar A-A, Tantar E, Bouvry P, Moral PD, Legrand P, EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III. New York, NY: Springer; 2014:141-166. https://doi.org/10.1007/978-3-319-01460-9_7.
14. Hernández DE, Olague G, Clemente E, Dozal L. Optimizing a conspicuous point detector for camera trajectory estimation with brain programming EVOLVE-A Bridg. between Probab. Set Oriented Numer. Evol Comput. 2014;500:121-140. https://doi.org/10.1007/978-3-319-01460-9_6.
15. Olague G, Clemente E, Dozal L, Hernández DE. Evolving an artificial visual cortex for object recognition with brain programming. EVOLVE-A Bridg. between probab. set oriented numer. Evol Comput III. 2014;500:97-119. https://doi.org/10.1007/978-3-319-01460-9_5.
16. Zhang M, Cagnoni S, Olague G. Evolutionary computer vision. Proceedings of the 11th Annual Conference on companion Genetic and Evolutionary Computation Conference GECCO '09; 2009:3355; New York, ACM Press. https://doi.org/10.1145/1570256.1570423.
17. Fonlupt C. Book review: genetic programming IV: routine human-competitive machine intelligence. Genet Program Evolvable Mach. 2005;6:231-233. https://doi.org/10.1007/s10710-005-7579-0.
18. Koza JR. Human-competitive results produced by genetic programming. Genet Program Evolvable Mach. 2010;11:251-284. https://doi.org/10.1007/s10710-010-9112-3.
19. Poli R, Langdon WB, McPhee NF, Koza JR. A field guide to genetic programming. Lulu. com; 2008.
20. Langdon WB, Poli R. Foundations of Genetic Programming. Berlin, Heidelberg: Springer; 2002 https://doi.org/10.1007/978-3-662-04726-2.
21. Koza JR. Genetic programming II: automatic discovery of reusable programs. Comput Math Appl 1995;29:115. https://doi.org/10.1016/0898-1221(95)90099-3.
22. Koza JR. Genetic programming as a means for programming computers by natural selection. *Stat Comput.* 1994;4:87-112.

23. Banzhaf W, Nordin P, Keller RE, Francone FD. *Genetic Programming: An Introduction.* Vol 1. San Francisco: Morgan Kaufmann; 1998.

24. Bleuler S, Brack M, Thiele L, Zitzler E. Multiobjective genetic programming: reducing bloat using SPEA2. Paper presented at: Proceedings of the 2001 Congress on Evolutionary Computation. (IEEE Cat. No.01TH8546), Seoul, Korea (South); vol. 1, 2001:536-43; IEEE. https://doi.org/10.1109/CEC.2001.934438.

25. Shao L, Member S, Liu L, Member S, Li X. Feature learning for image classification via multiobjective genetic programming. *IEEE Trans Neural Networks Learn Syst.* 2014;25:1359-1371. https://doi.org/10.1109/TNNLS.2013.2293418.

26. Rodriguez-Vazquez K, Fonseca CM, Fleming PJ. Identifying the structure of NonLinear dynamic systems using multiobjective genetic programming. *IEEE Trans Syst Man Cybern A Syst Humans.* 2004;34:531-545. https://doi.org/10.1109/TSMCA.2004.826299.

27. Tay JC, Ho NB. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Comput Ind Eng.* 2008;54:453-473. https://doi.org/10.1016/j.cie.2007.08.008.

28. Trujillo L, Olague G. Using evolution to learn how to perform interest point detection. Paper presented at: Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, China; vol. 1, IEEE; 2006:211-214; IEEE. https://doi.org/10.1109/ICPR.2006.1153.

29. Trujillo L, Olague G. Synthesis of interest point detectors through genetic programming. Paper presented at: Proceedings of the 8th Annual Conference Companion on Genetic and Evolutionary Computation- GECCO '06; 2006:887; ACM Press, New York. https://doi.org/10.1145/1143997.1144151.

30. Olague G, Trujillo L. Evolutionary-computer-assisted design of image operators that detect interest points. *Evol Comput.* 2008;16:483-507. https://doi.org/10.1162/evco.2008.16.4.483.

31. Olague G, Trujillo L. Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming. *Image Vis Comput.* 2011;29:484-498. https://doi.org/10.1016/j.imavis.2011.03.004.

32. Olague G, Trujillo L. Interest point detection through multiobjective genetic programming. *Appl Soft Comput.* 2012;12:2566-2582. https://doi.org/10.1016/j.asoc.2012.03.058.

33. Perez CB, Olague G. Learning invariant region descriptor operators with genetic programming and the F-measure. Paper presented at: Proceedings of the 2008 19th International Conference Pattern Recognit, Tampa, FL, USA; 2008:1-4; IEEE. https://doi.org/10.1109/ICPR.2008.4761178.

34. Perez CB, Olague G. Evolutionary learning of local descriptor operators for object recognition. Paper presented at: Proceedings of the 11th Annual Conference Conference on Genetic and Evolutionary Computation- GECCO ’09; 2009:1051; ACM Press, New York, NY. https://doi.org/10.1145/1569901.1570043.

35. Perez CB, Olague G. Genetic programming as strategy for learning image descriptor operators. *Intell Data Anal.* 2013;17:561-583. https://doi.org/10.3233/IDA-130594.

36. Trujillo L, Legrand P, Olague G, Lévy-Véhel J. Evolving estimators of the pointwise Hölder exponent with genetic programming. *Inf Sci (NY).* 2012;209:61-79. https://doi.org/10.1016/j.ins.2012.04.043.

37. Parent J, Nowe A, Steenhaut K, Defaweux A. Linear Genetic Programming using a compressed genotype representation. 2005 IEEE Congr. Evol. Comput., vol. 2, IEEE; 2005:1164-1171. https://doi.org/10.1109/CEC.2005.1554822.

38. Olague G, Chan-Ley M. Hands-on artificial evolution through brain programming. *Genetic Programming Theory and Practice.* Vol XVII. New York, NY: Springer; 2020;227-253 https://doi.org/10.1007/978-3-030-39958-0_12.

39. Poli R, Cagnoni S. Evolution of pseudo-colouring algorithms for image enhancement with interactive genetic programming. Cognitive Science Research Papers-University of Birmingham CSR P 1997.

40. Poli R, Cagnoni S, Marta VS. Genetic programming with user-driven selection: experiments on the evolution of algorithms for image enhancement. *Genet Program.* 1997;1:269-277.

41. Wang J, Tan Y. Morphological image enhancement procedure design by using genetic programming. Paper presented at: Proceedings of the 13th Annual Conference on Evolutionary Computing - GECCO '11; 2011:1435; ACM Press, New York, NY. https://doi.org/10.1145/2001576.2001769.

42. Khan SU, Ullah N, Ahmed I, Chai WY, Khan A. MRI images enhancement using genetic programming based hybrid noise removal filter approach. *Curr Med Imaging.* 2018;14:867-873.
43. Suganuma M, Shirakawa S, Nagao T. Deep Neural Evolution. Singapore, Asia: Springer; 2020. https://doi.org/10.1007/978-981-15-3685-4.

44. Mahmood MT, Majid A, Han J, Choi YK. Genetic programming based blind image deconvolution for surveillance systems. Eng Appl Artif Intell. 2013;26:1115-1123. https://doi.org/10.1016/j.engappai.2012.08.001.

45. Jakobovic D, Manzoni L, Mariot L, Picek S. CoInGP: convolutional inpainting with genetic programming; 2020. ArXiv Prepr ArXiv200411300.

46. Chaudhry A, Khan A, Ali A, Mirza AM. A hybrid image restoration approach: using fuzzy punctual kriging and genetic programming. Int J Imaging Syst Technol. 2007;17:224-231. https://doi.org/10.1002/ima.20105.

47. Yan R, Shao L, Liu L, Liu Y. Natural image denoising using evolved local adaptive filters. Signal Process. 2014;103:36-44. https://doi.org/10.1016/j.sigpro.2013.11.019.

48. Sharif M, Arfan Jaffar M, Tariq MM. Optimal composite morphological supervised filter for image denoising using genetic programming: application to magnetic resonance images. Eng Appl Artif Intell. 2014;31:78-89. https://doi.org/10.1016/j.engappai.2013.11.011.

49. Petrovic NI, Crnojevic V. Universal impulse noise filter based on genetic programming. IEEE Trans Image Process. 2008;17:1109-1120. https://doi.org/10.1109/TIP.2008.924388.

50. Harding S. Evolution of image filters on graphics processor units using Cartesian genetic programming. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence); 2008:1921-1928; IEEE. https://doi.org/10.1109/CEC.2008.4631051.

51. Majid A, Lee C-H, Mahmood MT, Choi T-S. Impulse noise filtering based on noise-free pixels using genetic programming. Knowl Inf Syst. 2012;32:505-526. https://doi.org/10.1007/s10115-011-0456-7.

52. Hernandez-beltran JE, Diaz-ramirez VH, Trujillo L, Legrand P. Design of estimators for restoration of images degraded by haze using genetic programming. Swarm Evol Comput. 2019;44:49-63. https://doi.org/10.1016/j.swevo.2018.11.008.

53. Chicotay S, David OE, Netanyahu NS. Image registration of very large images via genetic programming. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA; 2014:329-334. https://doi.org/10.1109/CVPRW.2014.56.

54. Lowe DG. Distinctive image features from scale-invariant Keypoints. Int J Comput Vis. 2004;60:91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

55. Langdon WB, Modat M, Petke J, Harman M. Improving 3D medical image registration CUDA software with genetic programming. Paper presented at: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, Vancouver BC Canada; July 12, 2014:951-958. https://doi.org/10.1145/2576768.2598244.

56. Hyun I, Tariq M. Adaptive outlier elimination in image registration using genetic programming R. Inf Sci (NY). 2017;421:204-217. https://doi.org/10.1016/j.ins.2017.08.098.

57. Fukunaga A, Stechert A. Evolving nonlinear predictive models for lossless image compression with genetic programming. Koza JR, Banzhaf W, Chellapilla K, Deb K, Dorigo M, Fogel DB, Garzon MH, Goldberg DE, Iba H, Riolo R. Genet Program. San Francisco, CA, USA: Morgan Kaufmann; 1998;95-102.

58. Poli R. Genetic programming for image analysis. Paper presented at: Proceedings of the 1st Annual Genetic and Evolutionary Computation Conference; 1996:363-368; New York, NY.

59. Liang J, Wen J, Wang Z, Wang J. Evolving semantic object segmentation methods automatically by genetic programming from images and image processing operators. Soft Comput. 2020;24:12887-12900. https://doi.org/10.1007/s00500-020-04713-1.

60. Liang Y, Zhang M, Browne WN. Feature construction using genetic programming for figure-ground image segmentation BT - intelligent and evolutionary systems. In: Leu G, Singh HK, Elsayed S, eds. Proceedings of the 20th Asia Pacific Symposium, IES 2016, Canberra, Australia. Cham: Springer International Publishing; 2017:237-250 https://doi.org/10.1007/978-3-319-49049-6_17.

61. Liang Y, Zhang M, Browne WN. Wrapper feature construction for figure-ground image segmentation using genetic programming BT - artificial life and computational intelligence. In: Wagner M, Li X, Hendtlass T, eds. Proceedings of the 3rd Australasian conference. ACALCI 2017, Geelong, VIC, Australia, January 31 – February 2, 2017. Cham: Springer International Publishing; 2017:111-123 https://doi.org/10.1007/978-3-319-51691-2_10.
62. Vojodi H, Fakhari A, Eftekhari Moghadam AM. A new evaluation measure for color image segmentation based on genetic programming approach. *Image Vis Comput*. 2013;31:877-886. https://doi.org/10.1016/j.imavis.2013.08.002.

63. Song A, Ciesielski V. Texture segmentation by genetic programming. *Evol Comput*. 2008;16:461-481. https://doi.org/10.1162/evco.2008.16.4.461.

64. Dong M, Eramian MG, Ludwig SA, Pierson RA. Automatic detection and segmentation of bovine corpora lutea in ultrasonographic ovarian images using genetic programming and rotation invariant local binary patterns. *Med Biol Eng Comput*. 2013;51:405-416. https://doi.org/10.1007/s11517-012-1009-2.

65. Liang Y, Zhang M, Browne WN. Engineering applications of Artifcial intelligence image feature selection using genetic programming for fi gure-ground segmentation. *Eng Appl Artif Intel*. 2017;62:96-108. https://doi.org/10.1016/j.engappai.2017.03.009.

66. Liang Y, Zhang M, Browne WN. Figure-ground image segmentation using feature-based multi-objective genetic programming techniques. *Neural Comput Appl*. 2017;51:3075-3094. https://doi.org/10.1007/s00521-017-3253-8.

67. Liang Y, Zhang M, Browne WN. Genetic programming for evolving figure-ground segmentors from multiple features. *Appl Soft Comput J*. 2017;51:83-95. https://doi.org/10.1016/j.asoc.2016.07.055.

68. Liang Y, Zhang M, Browne WN. Learning figure-ground image segmentors by genetic programming. Paper presented at: Proceedings of the Genetic and Evolutionary Computation Conference Companion; 2017:239-240; ACM, New York, NY. https://doi.org/10.1145/3067965.3075989.

69. Poli R. Genetic Programming for feature detection and image segmentation. *AISB Workshop Evolutionary Computing*. New York, NY: Springer; 1996:110-125 https://doi.org/10.1007/BFb0032777.

70. Fu W, Johnston M, Zhang M. Low-level feature extraction for edge detection using genetic programming. *IEEE Trans Cybern*. 2014;44:1459-1472. https://doi.org/10.1109/TCYB.2013.2286611.

71. Torres RD, Falcão AX, Gonçalves MA, et al. A genetic programming framework for content-based image retrieval. *Pattern Recognit*. 2009;42:283-292. https://doi.org/10.1016/j.patcog.2008.04.010.

72. Ciesielski V, Kurniawan D, Song A. Towards image retrieval by texture segmentation with genetic programming. Paper presented at: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing; 2007:281-286; IEEE. https://doi.org/10.1109/CIISP.2007.369182.

73. Calumby RT, da Silva TR, Gonçalves MA. Multimodal retrieval with relevance feedback based on genetic programming. *Multimed Tools Appl*. 2014;69:991-1019. https://doi.org/10.1007/s11042-012-1152-7.

74. Arni T, Clough P, Sanderson M, Grubinger M. Overview of the ImageCLEFphoto. In: Peters C, Deselaers T, Ferro N, et al., eds. *Photographic Retrieval Task*. Vol 2009. Berlin, Heidelberg: Springer; 2008:500-511. https://doi.org/10.1007/978-3-642-04447-2_62.

75. Ferreira CD, Santos JA, Torres RDS, Gonçalves MA, Rezende RC, Fan W. Relevance feedback based on genetic programming for image retrieval. *Pattern Recognit Lett*. 2011;32:27-37. https://doi.org/10.1016/j.patrec.2010.05.015.

76. Saraiva PC, Cavalcanti JMB, Gonçalves MA, dos Santos KCL, de Moura ES, Torres RDS. Evaluation of parameters for combining multiple textual sources of evidence for Web image retrieval using genetic programming. *J Brazillian Comput Soc*. 2013;19:147-160. https://doi.org/10.1007/s13173-012-0087-1.

77. Kobashigawa JS, Youn H-s, Iskander MF, Yun Z. Classification of buried targets using ground penetrating radar: comparison between genetic programming and neural networks. *IEEE Antennas Wirel Propag Lett*. 2011;10:971-974. https://doi.org/10.1109/LAWP.2011.2167120.

78. Hernández DE, Olague G, Hernández B, Clemente E. CUDA-based parallelization of a bio-inspired model for fast object classification. *Neural Comput Appl*. 2018;30:3007-3018. https://doi.org/10.1007/s00521-017-2873-3.

79. Hernández DE, Clemente E, Olague G, Briseño JL. Evolutionary multi-objective visual cortex for object classification in natural images. *J Comput Sci*. 2016;17:216-233. https://doi.org/10.1016/j.jocs.2015.10.011.

80. Chan-Ley M, Olague G. Categorization of digitized artworks by media with brain programming. *Appl Optics*. 2020;59:4437-4447. https://doi.org/10.1364/AO.385552.

81. Loveard T, Ciesielski V. Representing classification problems in genetic programming. Paper presented at: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546); May 27, 2001:1070-1077. https://doi.org/10.1109/CEC.2001.934310.
82. Smart W, Zhang M. Classification strategies for image classification in genetic programming. Paper presented at: Proceeding of Image and Vision Computing Conference; 2003:402-407; Palmerston North, New Zealand.

83. Tackett WA, Char KG. Genetic Programming for Feature Discovery and Image Discrimination. Proceedings of the Fifth International Conference on Genetic Algorithms. University of Illinois at Urbana-Champaign: Morgan Kaufmann; 1993:303-311.

84. Stanhope SA, Daida JM. Genetic programming for automatic target classification and recognition in synthetic aperture radar imagery. Paper presented at: International Conference on Evolutionary Programming, San Diego, CA, USA; 1998:735-744; Springer. https://doi.org/10.1007/BFb0040824.

85. Daida JM, Bersano-Begey TF, Ross SJ, Vesecky JF. Computer-assisted design of image classification algorithms: dynamic and static fitness evaluations in a Scaffolded genetic programming environment. Genet Program. 1996;1996:279-284. https://doi.org/10.7551/mitpress/3242.003.0039.

86. Devarriya D, Gulati C, Mansharamani V, Sakalle A, Bhardwaj A. Unbalanced breast cancer data classification using novel fitness functions in genetic programming. Expert Syst Appl. 2020;140:112866. https://doi.org/10.1016/j.eswa.2019.112866.

87. Elola A, Del J, Nekane M, Perfecto C, Alexandre E, Salcedo-sanz S. Hybridizing Cartesian genetic programming and harmony search for adaptive feature construction in supervised learning problems. Appl Soft Comput J. 2017;52:760-770. https://doi.org/10.1016/j.asoc.2016.09.049.

88. Tran B, Zhang M, Xue B. Multiple feature construction in classification on high-dimensional data using GP. Paper presented at: Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece; December 6, 2016;1-8; IEEE. https://doi.org/10.1109/SSCI.2016.7850130.

89. Nag K, Pal NR. A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification. IEEE Trans Cybern. 2016;46:499-510. https://doi.org/10.1109/TCYB.2015.2404806.

90. Iqbal M, Xue B, Al-Sahaf H, Zhang M. Cross-domain reuse of extracted knowledge in genetic programming for image classification. IEEE Trans Evol Comput. 2017;21:569-587. https://doi.org/10.1109/TEVC.2017.2657556.

91. Al-Sahaf H, Zhang M, Al-Sahaf A, Johnston M. Keypoint detection and feature extraction: a dynamic genetic programming approach for evolving rotation-invariant texture image descriptors. IEEE Trans Evol Comput. 2017;21:825-844. https://doi.org/10.1109/TEVC.2017.2685639.

92. 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. 2016;21:1-1. https://doi.org/10.1109/TEVC.2016.2577548.

93. Nandi RJ, Nandi AK, Rangayyan RM, Scutt D. Classification of breast masses in mammograms using genetic programming and feature selection. Med Biol Eng Comput. 2006;44:683-694. https://doi.org/10.1007/s11517-006-0077-6.

94. Atkins D, Neshatian K, Zhang M. A domain independent genetic programming approach to automatic feature extraction for image classification. Paper presented at: Proceedings of the 2011 IEEE Congress on Evolutionary Computation, New Orleans, LA, USA; 2011:238-245; IEEE. https://doi.org/10.1109/CEC.2011.5949624.

95. Al-Sahaf H, Song A, Neshatian K, Zhang M. Two-tier genetic programming: towards raw pixel-based image classification. Expert Syst Appl. 2012;39:12291-12301. https://doi.org/10.1016/j.eswa.2012.02.123.

96. Guo H, Nandi AK. Breast cancer diagnosis using genetic programming generated feature. Pattern Recognit. 2006;39:980-987. https://doi.org/10.1016/j.patcog.2005.10.001.

97. dos Santos JA, Ferreira CD, Torres RDS, Gonçalves MA, Lamparelli RAC. A relevance feedback method based on genetic programming for classification of remote sensing images. Inf Sci (Ny). 2011;181:2671-2684. https://doi.org/10.1016/j.ins.2010.02.003.

98. dos Santos JA, da Silva AT, da Silva TR, Falcão AX, Magalhães LP, Lamparelli RAC. Interactive classification of remote sensing images by using optimum-path Forest and genetic programming. Paper presented at: Proceedings of the International Conference on Computer Analysis of Images and Patterns; 2011:300-307; Springer, New York, NY. https://doi.org/10.1007/978-3-642-23678-5_35.

99. Choi W-J, Choi T-S. Genetic programming-based feature transform and classification for the automatic detection of pulmonary nodules on computed tomography images. Inf Sci (NY). 2012;212:57-78. https://doi.org/10.1016/j.ins.2012.05.008.
100. Zhang M, Smart W. Using Gaussian distribution to construct fitness functions in genetic programming for multiclass object classification. Pattern Recognit Lett. 2006;27:1266-1274. https://doi.org/10.1016/j.patrec.2005.07.024.

101. La W, Silva S, Danai K, Spector L, Vanneschi L, Moore JH. Multidimensional genetic programming for multiclass classification. Swarm Evol Comput. 2019;44:260-272. https://doi.org/10.1016/j.swevo.2018.03.015.

102. Burks AR, Punch WF. Genetic programming for tuberculosis screening from raw X-ray images. Proceedings of the Genetic and Evolutionary Computation Conference, 2018:1214-1221; ACM, New York, NY. https://doi.org/10.1145/3205455.3205461.

103. Arsalan M, Saeed A, Khan A, Rajarajan M. Protection of medical images and patient related information in healthcare: using an intelligent and reversible watermarking technique. Appl Soft Comput J. 2017;51:168-179. https://doi.org/10.1016/j.asoc.2016.11.044.

104. Golshan F, Mohamadi K. An intelligent watermarking algorithm based on genetic programming. Paper presented at: Proceedings of the 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), Kuala Lumpur, Malaysia; 2010:97-100; IEEE. https://doi.org/10.1109/ISSPA.2010.5605497.

105. Golshan F, Mohammadi K. A hybrid intelligent SVD-based digital image watermarking. Paper presented at: Proceedings of the 21st International Conference on Systems Engineering, Las Vegas, NV, USA; 2011:137-141; IEEE. https://doi.org/10.1109/ICSEng.2011.32.

106. Gilani L, Khan A, Mirza AM. Distortion estimation in digital image watermarking using genetic programming. Int J Appl Math Comput Sci. 2007;2:167-172. https://doi.org/10.5281/zenodo.1057077.

107. Usman I, Khan A. BCH coding and intelligent watermark embedding: employing both frequency and strength selection. Appl Soft Comput. 2010;10:332-343. https://doi.org/10.1016/j.asoc.2009.08.004.

108. Jan Z, Jabeen F, Jaffar A, Rauf A. Watermarking scheme based on wavelet transform, genetic programming and Watson perceptual distortion control model for JPEG2000. Paper presented at: Proceedings of the 2010 6th International Conference on Emerging Technologies, Islamabad, Pakistan; 2010:128-133. https://doi.org/10.1109/ICET.2010.5638368.

109. Abbasi A, Seng WC. Robust image watermarking using genetic programming. J Softw Syst Dev. 2012;2012:1. https://doi.org/10.5171/2012.881411.

110. Khan A, Mirza AM, Majid A. Optimizing perceptual shaping of a digital watermark using genetic programming. Iran J Electr Comput Eng. 2004;3:144-150.

111. Winkeler JF, Manjunath BS. Genetic programming for object detection. Genet Program. 1997;1:330-335.

112. Zhang M, Ciesielski V. Genetic programming for multiple class object detection. Paper presented at: Proceedings of the Australasian Joint Conference on Artificial Intelligence 1999:180-192; Springer, New York, NY. https://doi.org/10.1007/3-540-46695-9_16.

113. Zhang M, Ciesielski VB, Andreade P. A domain-independent window approach to multiclass object detection using genetic programming. EURASIP J Adv Signal Process. 2003;2003:206791. https://doi.org/10.1155/S1110865703303063.

114. Howard D, Roberts S, Brankin R. Target detection in SAR imagery by genetic programming. Adv Eng Softw 1999;30:303-11. https://doi.org/10.1016/S0965-9978(98)00093-3.

115. Liddle T, Johnston M, Zhang M. Multi-objective genetic programming for object detection. Paper presented at: Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain; 2010:1-8. https://doi.org/10.1109/CEC.2010.5586072.

116. Zhang M, Andreade P, Pritchard M. Pixel statistics and false alarm area in genetic programming for object detection. Paper presented at: Proceedings of the Applications of Evolutionary Computation EvoWorkshops2003 Evo[BIOL], Evo[COP], Evo[IASP], Evo[MUSART], Evo[ROB], Evo[STIM], Essex, UK; vol. 2611, 2003:455-66. https://doi.org/10.1007/3-540-36605-9_42.

117. Zhang M. Improving object detection performance with genetic programming. Int J Artif Intell Tools. 2007;16:849-873. https://doi.org/10.1142/S0218488507003576.

118. Hunt R, Johnston M, Zhang M. Improving Robustness of Multiple-Objective Genetic Programming for Object Detection. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7106 LNAI, 2011:311-320. https://doi.org/10.1007/978-3-642-25832-9_32.
119. Lin Y, Bhanu B. Object detection via feature synthesis using MDL-based genetic programming. *IEEE Trans Syst Man Cybern Part B*. 2005;35:538-547. https://doi.org/10.1109/TSMCB.2005.846656.

120. Bhanu B, Lin Y. Object detection in multi-modal images using genetic programming. *Appl Soft Comput*. 2004;4:175-201. https://doi.org/10.1016/j.asoc.2004.01.004.

121. Martin MC. Evolving visual sonar: depth from monocular images. *Pattern Recognit Lett*. 2006;27:1174-1180. https://doi.org/10.1016/j.patrec.2005.07.015.

122. Fu W, Johnston M, Zhang M. Genetic programming for edge detection: a global approach. Paper presented at: Proceedings of the 2011 IEEE Congress on Evolutionary Computation, New Orleans, LA, USA; 2011:254-261. https://doi.org/10.1109/CEC.2011.5949626.

123. Fu W, Johnston M, Zhang M. Genetic programming for edge detection using blocks to extract features. Paper presented at: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation; 2012:855; ACM Press, New York, NY. https://doi.org/10.1145/2330163.2330282.

124. Fu W, Johnston M, Zhang M. Genetic programming for edge detection via balancing individual training images. Paper presented at: Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia; 2012:1-8; IEEE. https://doi.org/10.1109/CEC.2012.6252879.

125. Fu W, Johnston M, Zhang M. Distribution-based invariant feature construction using genetic programming for edge detection. *Soft Comput*. 2013;19:2371-2389. https://doi.org/10.1007/s00500-014-1432-4.

126. Fu W, Johnston M, Zhang M. Unsupervised learning for edge detection using Genetic Programming. Paper presented at: Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China; 2014:117-124; IEEE. https://doi.org/10.1109/CEC.2014.6900444.

127. Krawiec K, Howard D, Box PO. Overview of object detection and image analysis by means of genetic programming techniques. *Proceedings of the 2007 Frontiers in the Convergence of Bioscience and Information Technologies*; Jeju, Korea (South): IEEE; 2007:785-790. https://doi.org/10.1109/FBIT.2007.148.

128. Puente C, Olague G, Smith SV, Bullock SH, Hinojosa-Corona A, González-Botello MA. A genetic programming approach to estimate vegetation cover in the context of soil erosion assessment. *Photogramm Eng Remote Sens*. 2011;77:363-376. https://doi.org/10.14358/PERS.77.4.363.

129. Puente C, Olague G, Trabucchi M, Arjona-Villicaña P, Soubervielle-Montalvo C. Synthesis of vegetation indices using genetic programming for soil erosion estimation. *Remote Sens (Basel)*. 2019;11:156. https://doi.org/10.3390/rs11020156.

130. Nguyen ML, Ciesielski V, Song A. Rice leaf detection with genetic programming. Paper presented at: Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico; 2013:1146-1153; IEEE. https://doi.org/10.1109/CEC.2013.6557695.

131. Xie C, Shang L. Anomaly detection in crowded scenes using genetic programming. Paper presented at: Proceedings of the 2014 IEEE Congress on Evolutionary Computation, Beijing, China; 2014:1832-1839. https://doi.org/10.1109/CEC.2014.6900396.

132. Shi Q, Song A. Selective motion detection by genetic programming. Paper presented at: Proceedings of the 2011 IEEE Congress on Evolutionary Computation (CEC), New Orleans, LA, USA; 2011:496-503. https://doi.org/10.1109/CEC.2011.5949659.

133. Pinto B, Song A. Detecting motion from noisy scenes using genetic programming. Paper presented at: Proceedings of the 2009 24th International Conference Image and Vision Computing; 2009:322-327; IEEE, New Zealand. https://doi.org/10.1109/IVCNZ.2009.5378389.

134. Liu L, Shao L, Li X, Lu K. Learning Spatio-temporal representations for action recognition: a genetic programming approach. *IEEE Trans Cybern*. 2016;46:158-170. https://doi.org/10.1109/TCYB.2015.2399172.

135. Song A, Shi Q, Yin W. Understanding of GP-evolved motion detectors. *IEEE Comput Intell Mag*. 2013;8:46-55. https://doi.org/10.1109/MCI.2012.2228594.

136. Bianco S, Ciocca G, Schettini R. Combination of video change detection algorithms by genetic programming. *IEEE Trans Evol Comput*. 2017;21:914-928. https://doi.org/10.1109/TEVC.2017.2694160.

137. Song A, Pinto B. Study of GP representations for motion detection with unstable background. Paper presented at: Proceedings of the 2015 IEEE Congress on Evolutionary Computation, Barcelona, Spain; 2015:1-8; IEEE. https://doi.org/10.1109/CEC.2015.7256334.

138. Shi Q, Wei Y, Song A. Analysis of motion detectors evolved by genetic programming. Paper presented at: Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia; 2012:1-8; IEEE. https://doi.org/10.1109/CEC.2012.6256355.
139. Song A, Zhang M. Genetic programming for detecting target motions; 2017:0091 https://doi.org/10.1080/09540091.2012.744873.

140. Olague G, Hernandez DE, Clemente E, Chan-Ley M. Evolving head tracking routines with brain programming. IEEE Access. 2018;6:26254-26270. https://doi.org/10.1109/ACCESS.2018.2831633.

141. Olague G, Hernández DE, Llamas P, Clemente E, Briseño JL. Brain programming as a new strategy to create visual routines for object tracking. Multimed Tools Appl. 2019;78:5881-5918. https://doi.org/10.1007/s11042-018-6634-9.

142. Engelbrecht AP. Computational Intelligence. Chichester, UK: John Wiley & Sons, Ltd; 2007. https://doi.org/10.1002/9780470512517.

143. Williams PJ, Molteno TCA. A comparison of genetic programming with genetic algorithms for wire antenna design. Int J Antennas Propag. 2008;2008:1-6. https://doi.org/10.1155/2008/197849.

144. Robilliard D, Marion-Poty V, Fonlupt C. Population parallel GP on the G80 GPU. Lecture Notes in Computer Science (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4971 LNCS; 2008:98-109; Springer, New York, NY. https://doi.org/10.1007/978-3-540-78671-9_9.

145. Olague G, Ibarra-Vázquez G, Chan-Ley M, Puente C, Soubervielle-Montalvo C, Martinez A. A Deep Genetic Programming Based Methodology for Art Media Classification Robust to Adversarial Perturbations. In International Symposium on Visual Computing; 2020:68-79; Springer, New York, NY https://doi.org/10.1007/978-3-030-64556-4_6.

146. Khan A, Sohail A, Zahoor U, Qureshi AS. A survey of the recent architectures of deep convolutional neural networks; 2019. ArXiv Prepr ArXiv190106032.

147. Ahmed U, Khan A, Khan SH, Basit A, Haq IU, Lee YS. Transfer learning and meta classification based deep churn prediction system for telecom industry; 2019. ArXiv Prepr ArXiv190106091.

How to cite this article: Khan A, Qureshi AS, Wahab N, Hussain M, Hamza MY. A recent survey on the applications of genetic programming in image processing. Computational Intelligence. 2021;37:1745–1778. https://doi.org/10.1111/coin.12459