S & M 3419

Fully Automated Construction of a Deep U-Net Network Model for Medical Image Segmentation

Daoqing Gong,[1] Jiayan Yang,[1] Xinyan Gan,[1*] Xiang Gao,[1] and Yuanxia Zhang[2*]

[1]School of Public Health and Management, Guangxi University of Chinese Medicine, Nanning 530299, China
[2]School of Computer and Information Engineering, Yulin Normal University, Nanning 537000, China

(Received July 14, 2023; accepted October 4, 2023)

Keywords: fully automatic, gene expression programming, U-Net Model, medical images

In recent years, the use of deep learning technology for image processing has become mainstream, and the U-Net network has received widespread attention owing to its unique U-shaped structure, which has achieved excellent results in the field of image segmentation, especially in medical image segmentation. To enhance the performance of the U-Net network model and establish better U-Net design variables, in this paper, we propose a fuzzy-controlled multicellular gene expression programming algorithm to automatically build and optimize the U-Net. The algorithm creates an efficient variable-length gene code, generates chromosomes for the optimization of U-Net design variables, decodes the chromosomes to construct the U-Net model, dynamically calculates population fitness and fuzzy affiliation values, and achieves the optimal U-Net network through continuous evolution. The experimental results indicate that the proposed algorithm outperforms U-Net, Fully Convolutional Networks32, and VanillaUnet in image recognition segmentation, especially in the segmentation of COVID-19 CT medical images.

1. Introduction

Computer vision is a field that enables computers to understand and process visual data, such as images and videos. With the continuous development of society and the application of deep learning techniques, advancements in computer CPUs and GPUs have significantly improved the efficiency and accuracy of computer vision tasks. Deep learning techniques leverage large amounts of data to train neural networks, enabling them to better recognize and analyze media information data, including images and videos, through continuous learning.

In the field of computer vision, semantic segmentation and image classification have become integral parts of our daily lives. For instance, face recognition(1) utilizes semantic segmentation to identify specific parts of a person's face, while autonomous driving(2,3) requires the real-time monitoring of various types of semantic information on the road to avoid accidents. Medical image segmentation(4) plays a crucial role in accurately

*Corresponding author: e-mail: 18775300556@163.com

https://doi.org/10.18494/SAM4587

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

identifying disease-specific areas at the pixel level, assisting doctors in formulating treatment plans for patients.

Over the years, extensive research and development efforts have yielded impressive results in both semantic segmentation and image classification.(5) Deep learning models such as Alex Net(6) have achieved remarkable classification accuracy on ImageNet datasets by employing multilayer convolutional neural networks. The ResNet model(7) addresses the degradation problem of deep residual networks by increasing network depth. Other models, including DenseNet,(8) EfficientNet,(9) MobileNet,(10) ShuffleNet,(11) and SegNet,(12) have also demonstrated excellent performance across various tasks. These advancements in deep learning have been made possible by the availability of high-quality labeled datasets and the exponential growth of computer computing power.(13) These models have significantly contributed to the progress and success of computer vision tasks, enabling computers to better understand and interpret visual data.

Semantic segmentation methods based on deep learning have demonstrated state-of-the-art performance in tasks such as medical image classification, detection, and segmentation.(14) A survey conducted by Li *et al.*(15) revealed that segmentation is the most sought-after task in medical image analysis.(16) Since 2015, the utilization of U-Net models(17) in medical images has gained significant momentum.(18) The U-Net model has become a popular research focus in this field owing to several advantages, including high training speed, the ability to achieve good results with limited annotated data, the efficient utilization of contextual information, feature learning from local and global aspects, and high transferability. However, the application of U-Net in medical images presents challenges due to privacy concerns associated with patient data. Medical images are characterized by high noise, variability, large scale, and limited training samples, making image processing and analysis highly challenging. Among various approaches, deep-learning-based semantic segmentation methods have demonstrated superior performance in addressing medical image problems. The U-Net model, being one of the semantic segmentation methods, effectively tackles the challenges encountered in medical image tasks and serves as a powerful tool for medical image processing and segmentation.(19,20) Its key applications include the automatic multiclass segmentation of COVID-19 chest CT images,(21) the automatic segmentation of rectal cancer regions,(22) and the segmentation of liver tumors and major blood vessels.(23)

Optimizing convolutional neural networks (CNNs) is crucial for refining the overall algorithm and reducing unnecessary resource and time waste. However, network optimization requires expertise and extensive knowledge in the field.(24) It is common for network optimization to encounter challenges such as getting trapped in local optima while searching for the global optimal solution. Training time can be long, especially when dealing with large datasets. Moreover, network optimization often requires substantial computer memory and computational power. The random initialization of network parameters typically necessitates training through numerous iterations to find the best approach, often referred to as the "brute force method".(25) Considering these challenges, many experts and scholars have proposed various optimization methods.

These include grid search,(26) random search (RS),(27) Bayesian-based optimization (BGP),(28,29) Parzen estimator (TPE) for tree networks,(30) global optimization (SMBO) based on sequential models,(31) neural evolution with extended topology,(32) and evolutionary unsupervised deep learning.(33) These methods aim to overcome the limitations and improve the efficiency of network optimization by utilizing different strategies and algorithms. Overall, network optimization plays a critical role in enhancing CNN performance, but it requires specialized knowledge and expertise. Researchers have developed several optimization methods to address the challenges associated with finding the best solutions and improving efficiency in training deep neural networks.

Gene expression programming (GEP), mentioned in this paper, is an evolutionary computational method based on genetic algorithms,(34) which can be applied to a wide range of problems and has good flexibility and adaptability. Through GEP, multicellular gene expression programming can be further extended in terms of its expressive power and search space. Evolutionary algorithms have proven to be one of the most competitive methods for neural network optimization problems,(35) and there are a large number of relevant parameters(36) that require human intervention to find a good U-Net network model.(37)

In this paper, we propose a fuzzy-control-based multicellular gene expression programming algorithm to automatically build and optimize U-Net. It was also applied on medical image data of colorectal cancer images and COVID-19 CT images for validation, and compared with classical algorithms and other advanced algorithms. The results show that the algorithm proposed in this paper achieves a better fully automated segmentation effect in image segmentation. The main contributions of this paper are summarized as follows:

(1) Efficient Model Building: The proposed algorithm introduces a flexible coding strategy that automates the construction of a deep U-Net network model. This coding strategy significantly reduces the consumption of human, material, and financial resources involved in model development. By automating the process, the algorithm streamlines the model building workflow, leading to improved efficiency and cost-effectiveness.

(2) Scalability and Adaptability: The algorithm's design is not limited to specific U-Net constructs or segmentation tasks. It can be applied to various segmentation tasks, allowing for the development of models with different architectural configurations. This flexibility enables researchers and practitioners to adapt the algorithm to specific requirements and achieve optimal performance in diverse applications. Additionally, the algorithm addresses the challenges associated with framework construction, eliminating the need for manual adjustments of parameters. This eliminates the tediousness of parameter tuning and enhances accuracy and other aspects of the U-Net model.

(3) Performance Improvement: By automating the U-Net model construction and optimization process, the algorithm contributes to improved performance in image segmentation tasks. The efficient model building and adaptable nature of the algorithm allow for a better utilization of resources and enable researchers to focus on other critical aspects of the segmentation problem. Consequently, the algorithm aids in achieving higher accuracy and training speed, and improved overall results in image segmentation tasks, particularly in the medical imaging domain.

2. Preliminaries

2.1 Fuzzy control multicell gene expression programming

GEP is stochastic search and optimization adaptive evolutionary algorithm(34) that inherits the fixed length of genetic algorithms (GAs) and the flexible and versatile tree structure of genetic programming (GP). Unlike GP, GEP uses a linear chromosome rather than a tree structure to represent functions and can generate functions of arbitrary size and complexity. This gives GEP greater flexibility and scalability in solving real-world problems. GEP is unique in that it uses a gene expression approach, making the code cleaner and more readable, but owing to its iterative nature, it requires a large amount of computation especially when dealing with large-scale problems.(38) Multicellular gene expression programming is based on GEP by introducing the concept of homologous genes, which enhances the search and expression capabilities of GEP.(39) Fuzzy control is an intelligent control method based on fuzzy set theory, fuzzy linguistic variables, and fuzzy logical reasoning.(40) Fuzzy control multicellular gene expression programming is the introduction of fuzzy control based on multicellular gene expression programming, abbreviated as FMCGEP.

GEP utilizes chromosomes to represent individuals, which consist of two or more gene expressions. Gene expressions are linear strings of symbols with fixed lengths, containing both non-terminator and terminator nodes. An individual in GEP comprises both common and homologous gene types. Chromosomes are composed of multiple fixed-length, equally sized genes, and the initial population is formed by randomly generated chromosomes. Genetic operators such as mutation, string insertion, single-point recombination, two-point recombination, and genetic recombination(41) ensure the legality of gene structures and the integrity of chromosome structures. These operators play a crucial role in producing new individuals by modifying gene expressions. Tournaments are used to select individuals with the highest fitness values for reproduction, allowing for the evolution of the population in a meritocratic manner. Through this iterative process, the population gradually adapts to the environment, eventually converging towards the optimal solution.

The structure of GEP can be divided into three parts, which are the head h, tail t, and DC domain. The elements that make up each part are shown in Table 1. Different genes can be connected to each other with function characters, and the relationship between the head and the tail can be expressed as

$$t = h \times (n - 1) + 1. \tag{1}$$

Here, n represents the maximum number of arguments allowed by the operator.

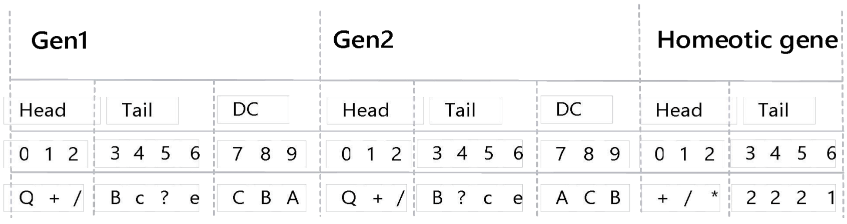
Table 1
Genetic building blocks.

Position	Constituent element
Head	+, −, *, /, %, sin, cos, tan, max, min, and, or, not,?,a,b,c,d,e,f,g,h,i
Tail	?,a,b,c,d,e,f,g,h,i
DC domain	1,2,3,4,5,6,7,8,9,10

Gen1 and Gen2 in Fig. 1 represent common gene 1 and common gene 2, respectively, where "?" in the tail of Gen1 corresponds to the C letter in the DC domain of Gen1, which maps to the third number in the Constants set (1.7601). the homeotic gene denotes a homologous gene, and 1 in the tail denotes Gen1 and 2 denotes Gen2. The decoding of MCGEP in Fig. 1 yields the expression tree network shown in Fig. 2. The formula obtained after decoding the MCGEP tree network according to Fig. 2 is shown as

$$(b+?)*(c/b)*\sqrt{(b/c+?)}\,. \tag{2}$$

The FMCGEP algorithm utilizes a fitness function to evaluate the chromosomes' fitness in the population relative to the environment.(42) This function measures the disparity between the actual and target values. After calculating the fitness value and fuzzy affiliation of the population, FMCGEP automatically adjusts the crossover rate, variance rate, and fixed set variance based on the fuzzy affiliation. This adaptive adjustment helps the population escape local optima and converge rapidly. To enhance the population's diversity, FMCGEP employs genetic operations that introduce randomness and generate new chromosomes through genetic operators. These newly generated chromosomes are added to the population, allowing for a more comprehensive exploration of the search space and the discovery of improved solutions.(43)



Gen1			Gen2			Homeotic gene	
Head	Tail	DC	Head	Tail	DC	Head	Tail
0 1 2	3 4 5 6	7 8 9	0 1 2	3 4 5 6	7 8 9	0 1 2	3 4 5 6
Q + /	B c ? e	C B A	Q + /	B ? c e	A C B	+ / *	2 2 2 1

Constants set={6.5204,15.1941,1.7601,2.9427,0.1257}

Fig. 1. MCGEP genotype network.

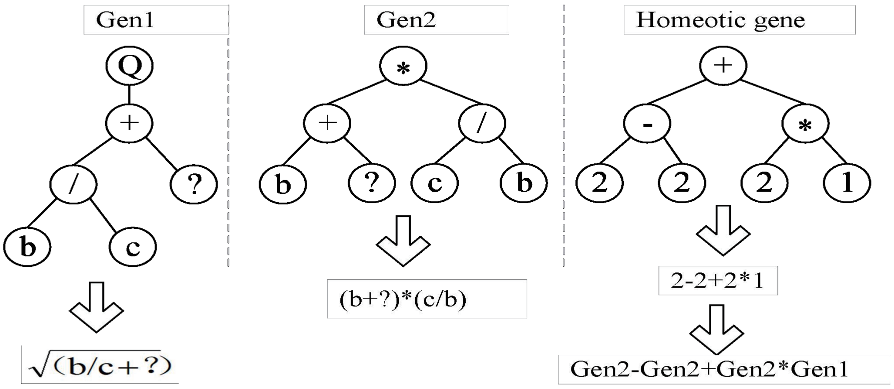


Fig. 2. MCGEP tree network.

2.2 U-Net model

The U-Net model is a pixel-level segmentation model that has been widely used in medical imaging for lesion segmentation tasks and has become an effective aid to physicians in diagnosing diseases. The structure of the U-Net is shown in Fig. 3.

The U-Net model follows a systematic process involving an encoder and a decoder to perform image segmentation. The input image is first passed through the encoder, which consists of multiple convolution and pooling operations. The convolution layers employ a 3 × 3 unfilled convolution kernel with a stride of 1 and a ReLU activation function. The pooling layers use a 2 × 2 maximum pooling technique for down-sampling. The role of the encoder is to transform the original image's low-level information into high-level, abstract feature representations with reduced dimensionality and increased channel depth. The decoder, on the other hand, performs multiple transpose convolution and up-sampling operations. The original image, which was trained by the encoder, serves as input to the decoder. The up-sampling stage employs deconvolution operations to restore the image resolution to its original size. Additionally, the feature maps from the up-sampling process are merged with the corresponding encoder layer's feature maps using skip connections. This merging process aids in reconstructing the original image by preserving important information from earlier stages of the network.

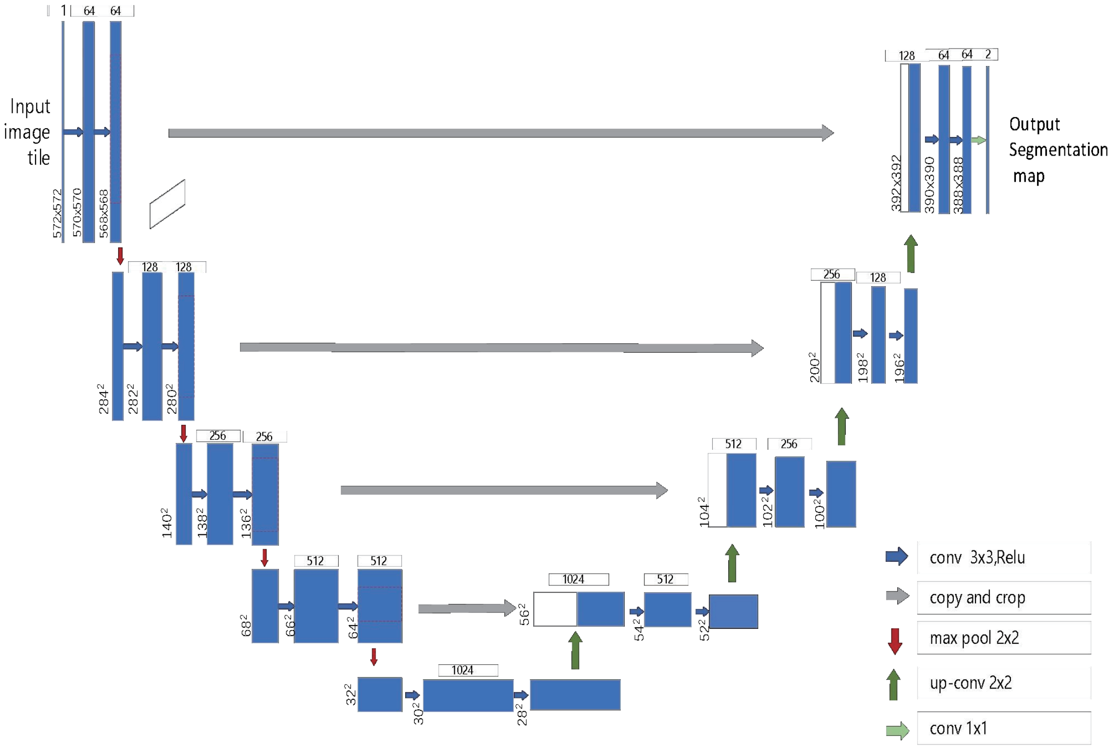


Fig. 3. (Color online) Structure of U-Net network model.

Lastly, two additional 3 × 3 convolution layers with a ReLU activation function are employed to accurately recover image details while preserving global features. One challenge in convolution is the loss of image edge information due to the absence of padding. To address this, an overlap-tile strategy that applies mirror expansion to the pixels surrounding the medical image was introduced.(17) This approach helps infer missing context, better preserves global information, and handles images of various sizes. After passing through the convolutional layers, a 1 × 1 convolutional kernel (without the ReLU activation function) is applied, and the final segmentation result is obtained. To further enhance segmentation quality, a weighted loss is utilized to assign higher importance to specific pixel points. The U-Net network consists of 18 convolutional layers. To prevent overfitting, the training dataset is augmented by incorporating transformations such as rotation, flipping, and translation.

3. Proposed Algorithm

3.1 Algorithm overview

Algorithm 1 outlines the framework of the FMCGEP-U-Net algorithm and is divided into five main sections: Inputs: population size N, maximum number of iterations G, crossover rate P_c, variation rate P_m, and tournament size P_t. Output: optimal U-Net model.

(1) The population is first initialized according to the proposed FMCGEP coding strategy, and the design variables to be optimized for the U-Net are coded into the chromosomes during the random initialization of the population (see line 1 of the code).

(2) The chromosomes are decoded to construct the U-Net model and evaluated, and the resulting evaluated values are used as individual fitness values (line 2).

Algorithm 1
FMCGEP-U-Net framework.

```
Input:   Population size $N$, Maximum number of iterations $G$, Crossover rate $P_c$, Variation rate $P_m$, Tournament
         size $P_t$.
Output:  Optimal U-Net model.
 1  $P_0$ ←Initialize a population with the size of $N$ by MCGEP coding strategy;
 2  Construct the U-Net model, evaluate the fitness of individuals in $P_0$;
 3  Sort $P_0$;
 4  Calculated membership (degree);
 5  While $t<G$ do
 6     $Q_t$ ← Ø
 7        While $Q_t < N$ do
 8           If degree < 0.6 or degree > 1
 9              $P_m = P_m$, $P_c = P_c$;
10           else: $P_m$ = Mutation_fuzzy_control, $P_c$ =Cross_fuzzy_control;
11           $q1$, $q2$ ←do mutation operation, do crossover operation;
12           $q3$ ←do the mutation first then the crossover operation;
13           $q4$ ← do the crossover first then the mutation operation;
14           $Q_t$ ←$q1$U$q2$U$q3$U$q4$;
15           End
16           Sort $Q_t$, which population size equal to $N$;
17  End
18  Select the best U-Net from $Q_t$.
```

(3) Comprehensive ranking of individual fitness values (row 3).

(4) The adaptation variance is used as a parameter for fuzzy control and the fuzzy affiliation is calculated (row 4).

(5) Finally, the optimal U-Net network model is obtained by continuous evolutionary iteration (line 18).

During the execution of the code (lines 5–17), a new population Q_t of size N needs to be created. If the affiliation of an individual is less than 0.6 or greater than 1, the mutation rate P_m and the crossover rate P_c remain unchanged; otherwise, fuzzy logic control is used to adjust the probability parameters of mutation, crossover, and constant mutation according to the fitness value of the current population. Genetic operations were carried out on the population, $q1$, $q2$, $q3$ and $q4$ represented four temporary populations, and then genetic operations of mutation, crossover, post-mutation crossover, and post-cross mutation were carried out on the temporary population. The offspring produced by all four genetic operations are collected into the parental population (Q_t) and ranked according to fitness values, and the best U-Net model is selected from Q_t at the end of each iteration.

The process of medical image identification is implemented according to our algorithm, as presented in Fig. 4. In the first step, we used the FMCGEP algorithm to train the parameters (e.g., convolution and activation function) of the U-Net model to be constructed, and we used the optimized parameters for the construction of the U-Net model. The two medical image datasets, COVID-19 and colorectal carcinomatosis, were trained using the parameter-optimized model, and then the results of this training were derived from the obtained results and the generated medical image segmentation effect map, through a fully connected layer, with the final output of the probability of image classification.

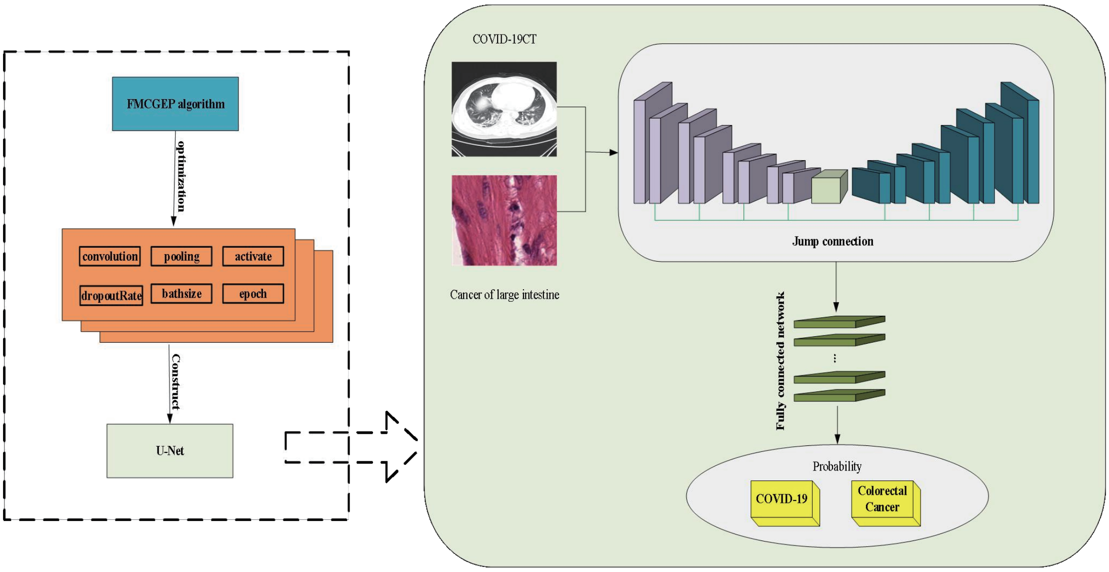


Fig. 4. (Color online) FMCGEP-U-Net algorithm implementation framework.

3.2 Genetic coding strategy

According to Sect. 2.2, the U-Net model consists of two types of operation: convolutional and pooling layers. Additionally, jump connections directly connect certain features from the encoder to the decoder. The optimal depth of the U-Net model structure is task-dependent, and the depth size impacts the model's performance. A deeper model can capture higher-level features from the image information, but it also requires more computational resources and time. Therefore, the proposed FMCGEP-U-Net genetic encoding strategy in this paper can be adapted to the specific requirements of practical problems to enhance its suitability.

In the genetic algorithm, each gene represents the parameters needed in the U-Net model. The gene length is determined by the combined lengths of the head, tail, and DC domains. Initially, a sequence of strings representing the gene is randomly generated from a set of operators and constant variables, based on the predefined parameter settings, by traversing the gene length.

3.3 FMCGEP-U-Net network encoding

There exists a significant correlation between the performance of medical image segmentation and learning parameters. Typically, the segmentation performance of a model is compared by adjusting parameters such as the activation function and pooling size.(44–46) In the FMCFGEP-U-Net strategy, gene encoding is used to represent the values of each hyperparameter in the U-Net network structure. Each hyperparameter is encoded into a single gene to express its value. The generated genome is then synthesized into the chromosome of the U-Net design variable for optimization, following the chromosome structure shown in Fig. 5. It is crucial to verify whether the generated chromosome adheres to the defined domain. This verification can be achieved by decoding each gene and calculating its value representation to perform a range check. If a gene value falls outside the defined domain, it is regenerated until it meets the defined criteria.

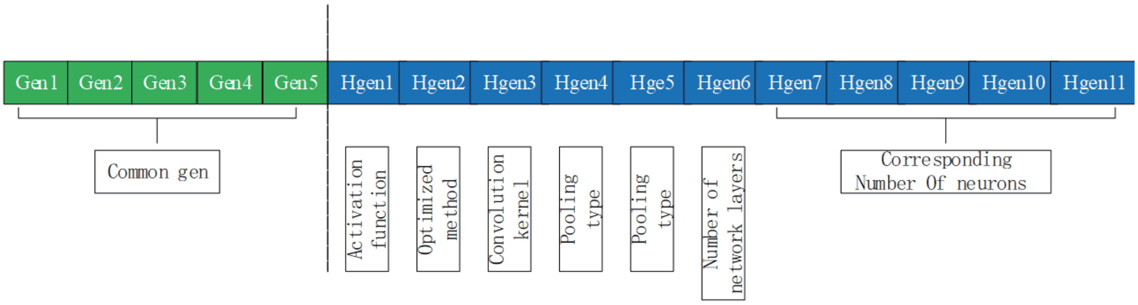


Fig. 5. (Color online) MCGEP chromosome network.

Assuming that there are M design variables to be optimized in the chromosome, they can be represented by M homologous genes, which are obtained by mapping N common genes, which are decoded into a specific column of real numbers. We extract the genetic parameters from the conforming chromosomes, which include the activation function, optimizer, convolution and pooling kernel, pooling type, dropout rate, learning rate, *L* (regularization parameter), batch size, number of epochs, number of convolution blocks, number of residual blocks, number of hidden layers, and number of neurons in each hidden layer, and use these genetic parameters to construct the U-Net network model with the specified optimizer to compile the model. To verify how good the model is, we train the model using the dataset, calculate the training data loss and metrics for the model, and save the model locally.

The activation function, optimization method, convolution kernel size, pooling kernel size, and pooling type in the homologous genes in Fig. 5 are represented by the corresponding Hgen1–Hgen6, Hgen7 indicates the number of network layers, and Hgen8–Hgen11 corresponds to the number of neurons in the hidden layer.

3.4 Population initialization

Algorithm 2 gives the main steps of population initialization. For the initialization of a population to be implemented, it can be broken down into a given minimum number of genetic units, which in turn form chromosomes from multiple genes and then individuals from chromosomes, with multiple individuals gathered to form a population. In this algorithm, three input parameters are accepted, the population size *N*, the number of chromosomal genes *Gn*, and the chromosome length *CL*, and an initialized population P_0 is output. First, P_0 and F_0 are generated giving an initial value of null; for each individual, an empty list G_0 is first initialized, a specified number of genes are generated by the statement, the genes that match are constructed into chromosomes, and the generated chromosomes are added to P_0. On the basis of the generated chromosomes, the model is evaluated for health using the specified optimization function, and the fitness value of the chromosome can be obtained and stored in F_0.

Algorithm 2
Population initialization.

```
Input:   Population size $N$, number of chromosome genes $Gn$, chromosome length $CL$.
Output:  Initialized population $P_0$.
$P_0 \leftarrow \varnothing, F_0 \leftarrow \varnothing, i \leftarrow 1$;
For $i \le N$ do
    $G_0 \leftarrow \varnothing, j \leftarrow 1$;
    For $j \le Gn$ do
        $G_0$←Generate $Gn$ gene sequences
     $a \leftarrow 1$
     If $a \le CL$ then
        $P_0$←Generate a chromosome and determine whether each gene in $G_0$ matches the defined domain. If not,
              the gene will be regenerated.
     $F_0$←Calculate chromosome fitness value
End
Return $P_0$
```

3.5 Fitness evaluation

Algorithm 3 describes the health assessment process. In the evolutionary algorithm, fitness values are assessed and used to determine which of the best individuals have the ability to reproduce offspring. An evolutionary operation is performed on each individual in the population; the parameters in the chromosome are used to construct a U-Net model; the U-Net model is trained with a training dataset and the accuracy on the test dataset is stored in the set $D_{fitness}$ as the fitness of that individual; the model evaluation value is used as the individual fitness value and the optimal fitness value of the population is obtained by continuous evolution.

Algorithm 3 describes the health assessment process. In the evolutionary algorithm, fitness values are assessed and used to determine which of the best individuals have the ability to reproduce offspring. An evolutionary operation is performed on each individual in the population; the parameters in the chromosome are used to construct a U-Net model; the U-Net model is trained with a training dataset and the accuracy on the test dataset is stored in the set $D_{fitness}$ as the fitness of that individual; the model evaluation value is used as the individual fitness value and the optimal fitness value of the population is obtained by continuous evolution.

The fitness evaluation assesses the quality of the population by measuring the fitness of each chromosome within the environment. The optimal individual is then selected as the genetic parent for the next generation of the population, and the design variables of the U-Net are further optimized to continue its evolution. The ultimate objective of this process is to achieve an optimal U-Net model. A variable-length genetic encoding strategy is employed, allowing each

Algorithm 3
Fitness evaluation.

```
Input:  Initialize population $P_t$, train data $D_{train}$, test data $D_{test}$, the training epoch number $k$ for measuring the
        accuracy tendency.
Output: Optimal fitness value $F_{train}$
For each individual in $P_t$ do
  $i \leftarrow 1$;
  while $i \leq k$ do
     Train the connection weights represented by individual $S$;
     if $i{=}{=}k$ then
        accy_list ← Ø, $D_{fitness}$ ← Ø;
        U-Net ←Build a model by using the parameter values in chromosomes as parameters for model
              training；
        Train U-Net model on $D_{train}$;
        accy_list ← Evaluate the classification accuracy of the trained U-Net on $D_{test}$
        $D_{fitness}$ ← Assign accuracy as the fitness of individual;
     End
  End
  Sort the elements in $D_{fitness}$;
  Bestvalue ← The optimal fitness value of the current population was obtained;
  Degree ←Calculate population diversity (mean/maximum);
  Dynamically adjust parameters based on degree;
  Update $S$ in $P_t$;
End
Return $P_t$
```

individual to be initialized with a small number of epochs for training, based on their architecture and weights. The fitness value of each epoch is calculated, along with metrics such as variance and fuzzy affiliation, to evolve the regulatory population. The evaluation metrics used in this study include accuracy, Dice coefficient, Jaccard index, and loss. These metrics are employed to assess the performance and effectiveness of the proposed approach.

The selection of an appropriate fitness function is crucial for enhancing the algorithm performance. It is essential to conduct a thorough analysis and choose the most suitable fitness function for specific tasks or even design a custom fitness function based on the specific situation.

3.6 Environmental selection

Algorithm 4 outlines the process of tournament selection, which involves carefully selecting individuals from the population through environmental selection. Firstly, the algorithm determines the number of individuals to be selected on the basis of the population size (line 1). Next, n individuals are randomly chosen from the entire population, and their fitness values are compared. The individuals with higher fitness values are then selected to form the next generation of the population (line 2). This process emulates natural selection, where the fittest individuals survive. The process continues until the specified requirements are met (rows 3–5). The choice of the value of n significantly affects the overall algorithm. A value that is very large increases the likelihood of selecting individuals with high fitness values, while a value that is very small may result in a less effective selection of the best individuals and have a greater impact on the subsequent generation of populations.

In this paper, we designed two sets of experiments on selected colorectal cancer images and COVID-19 CT image datasets to test the performance of the algorithm. The datasets to be used are briefly described below, and the parameter settings for the FMCGEP-U-Net method to participate in the experiments are given.

Algorithm 4
Environmental selection.

```
Input:   Current population $P_t$ U $Q_t$.
Output:  Select Population $P_{t+1}$.
1  $n$ ← Determine the optimal number of individuals selected on the basis of population size $N$;
2  $P_{t+1}$ ←From the n randomly selected individuals, compare their fitness values and select the best individual;
3  **While** | $P_{t+1}$ | < $N$ **do**
4      $S$ ←Randomly select 20% of the population size $N$ from $P_t$ U $Q_t$;
5      $P_{t+1}$ ←$P_{t+1}$ U $S$
6  End
7  Return $P_{t+1}$.
```

4. Dataset

4.1 Dataset description

The task of medical image segmentation is mainly to segment medical images into several similar or different regions by automatic or semi-automatic methods.[47] Medical image data information involves patients' personal information, and according to legal and ethical constraints and norms, this image data can only be used by specific people within a defined scope, which also leads to a more difficult access to medical image data and reflects the preciousness of publicly available medical image datasets. In this paper, we use the private COVID-19CT and colorectal cancer datasets, the source data of which is the publicly available COVID-19CT dataset, and manually select the better image composition in order to better train the model and reduce the risk of overfitting. Regarding the images and CT_NonCOVID files, the COVID-CT-Images file contains 264 CT images of newly crowned patients and the CT_NonCOVID file contains 1000 CT images of non-newly crowned patients (a total of 1264 images).

The colorectal cancer image dataset contains colorectal histology images of eight disease types. The eight different disease categories are represented by the eight folders 01_TUMOR, 02_STROMA, 03_COMPLEX, 04_LYMPHO, 05_DEBRIS, 06_MUCOSA, 07_ADIPOSE, and 08_EMPTY. Each folder contains 625 images, each with a size of 150 × 150, a pixel density of 72 dpi, and 24-bit depth, for a total of 5000 images of colorectal cancer. We randomly assigned 80% of each dataset as the training set and 20% as the validation set by breaking up each dataset, and the details of the number of images used in each dataset are shown in Table 2.

4.2 Parameter settings

The FMCGEP parameters include gene length, head and tail length, population size, the number of evolutionary iterations, and so forth. The U-Net parameters include upper and lower limits and types of convolutional kernel size, upper and lower limits and types of pooling kernel, and so forth. The detailed parameters of the algorithm are shown in Table 3.

5. Experimental Results and Analysis

We set up two sets of experiments to train and compare the FMCGEP-U-Net algorithm and the U-Net algorithm on the COVID-19 image and colorectal cancer image datasets, so we will analyze and discuss the results out of these two sets of experiments in detail.

5.1 COVID-19 image experiment

On the COVID-19 image dataset, we conducted comparative analyses mainly in terms of classification accuracy, loss function values, Dice similarity coefficients, and Jaccard similarity coefficients. To calculate these performance measures, confusion matrices were used, which

Table 2
Distribution of images in the dataset.

Dataset	Training set	Test set
COVID-19CT	1011	253
Cancer of large intestine	4000	1000

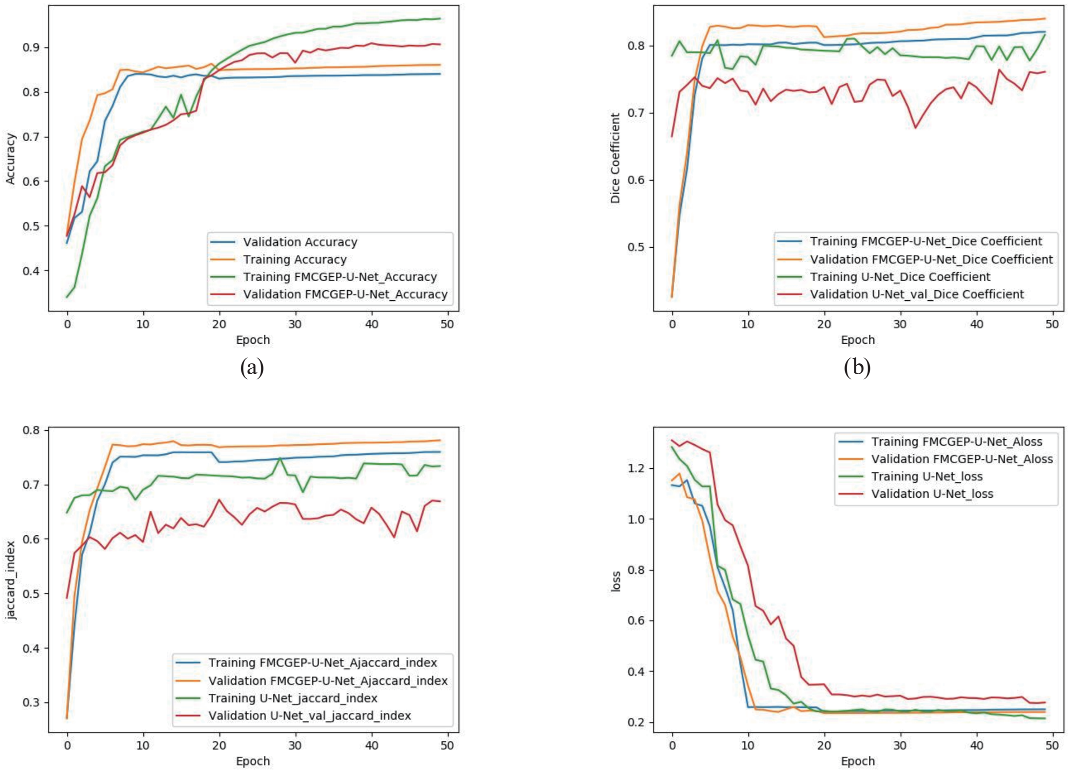
Table 3
FMCGEP-U-Net parameter settings.

Parameter	Values
FMCGEP	
Gene length	18
Head length	6
Tail length	7
DC domain	5
Population size	20
Number of evolutionary iterations	20
Tournament ratio	0.1
Crossover rate	0.3
Gene length	18
U-Net	
Lower and upper bounds of convolution kernel size	[3,5]
Lower and upper bounds of pool kernel size	[2,3]
Number of pool kernel types	2
Type of pool kernel function	5
Activation function type	5

included true positive (TP), false positive (FP), true negative (TN), and false negative (FN) variables. The results of the comparative analysis provided showed that the FMCGEP-U-Net model performed better on the COVID-19 image dataset. In terms of classification accuracy and Dice and Jaccard similarity coefficients, the FMCGEP-U-Net model obtained higher results than U-Net on both the training and validation sets. Although the FMCGEP-U-Net model was slightly lower in terms of loss function values relative to the U-Net model, its fully automated segmentation was better. Therefore, it can be concluded that the FMCGEP-U-Net model is more suitable for the image discrimination of fully automated segmentation effects in the COVID-19 image experiments.

According to the relationship between the accuracy of the training set and the verification set and the epoch shown in Fig. 6(a), it can be found that when the epoch is 10, the performance of the U-Net model starts to flatten out and keeps rising steadily. In the whole epoch range of 0 to 20, the U-Net model has a higher accuracy than the FMCGEP-U-Net model. However, at epoch 20, the accuracy of the FMCGEP-U-Net model exceeds that of U-Net and continues to rise. It shows that the FMCGEP-U-Net model has a stronger generalization ability and a faster convergence than the U-Net model. By comparing the algorithms in Table 4, the accuracy of the FMCGEP-U-Net model is as high as 90.83%, while that of the U-Net model is only 86.24%. Therefore, the FMCGEP-U-Net model has better classification performance for COVID-19 image dataset classification.

Figures 6(b)–6(d) show the variations of Dice similarity coefficient, Jaccard similarity coefficient, and loss function values in each epoch of each model. In terms of Dice coefficient



(a) (b)

(c) (d)

Fig. 6. (Color online) Experimental analysis of the COVID-19 dataset.

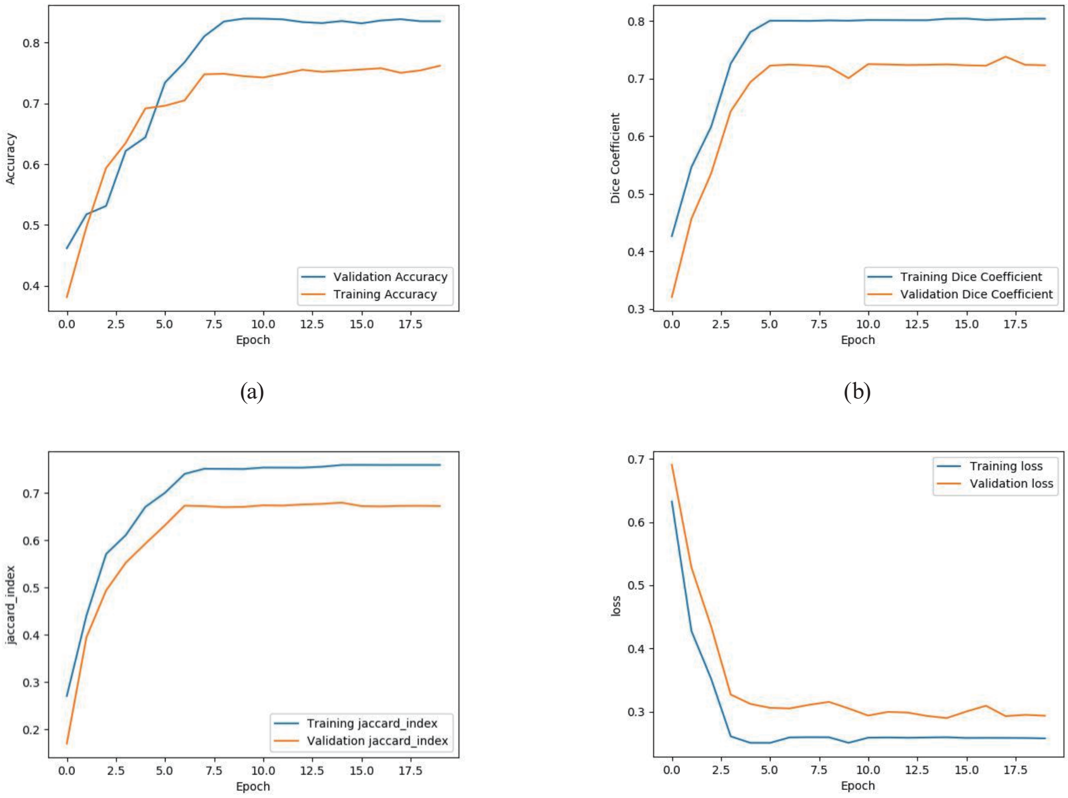
Table 4
Comparison of methods on the COVID-19 dataset.

Methods	Accuracy rate (%)	Epoch
U-Net	86.24	50
FCN32[48]	87.68	50
VanillaUnet[49]	89.62	50
FMCGEP-U-Net	**90.83**	50

and Jaccard index, the training and verification results of FMCGEP-U-Net were better than those of U-Net, indicating that FMCGP-U-Net had better target segmentation performance. However, it can be seen from the loss function diagram that the loss value of FMCGEP-U-Net is lower than that of U-Net in both training and verification sets, so the FMCGEP-U-Net model has a higher robustness than U-Net.

5.2 Colorectal cancer image experiment

According to the result of analyzing of the experimental results of FMCGEP-U-Net on the large intestine cancer dataset in Fig. 7, it can be found that the modified model shows a good trend of accuracy, Dice coefficient, Jaccard similarity coefficient, and loss function values in the



(a) (b)

(c) (d)

Fig. 7. (Color online) Experimental analysis of the colorectal carcinoma dataset.

training and test sets. Figure 7(a) shows that model accuracy gradually improves with the increase in epoch. After the fifth epoch, the accuracy of the verification set exceeded the accuracy of the training set and maintained a steady upward trend. After the eighth epoch, the accuracies of the verification and training sets remained in a reliable stable state until the end of the training. As shown in Figs. 7(b)–7(c), the Dice and Jaccard similarity coefficients of the FMCGEP-U-Net model on the training set showed good performance, with the highest value approaching 0.8, indicating that the model was capable of effectively processing the segmentation and prediction of colorectal cancer images.

In addition, according to the input image of the large intestine dataset and the predictive mask image shown in Fig. 8, the black part represents the result of image segmentation, which is marked, and the segmentation effect of this model is good. Overall, some aspects of the model can be further perfected and improved, but according to the experimental data given in this paper, its excellent prediction effect shows that the model is feasible.

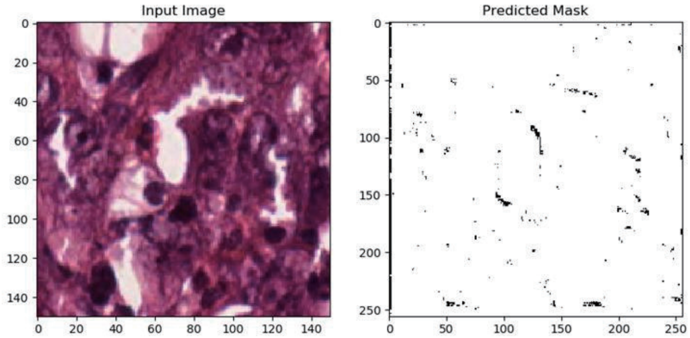


Fig. 8. (Color online) Input and predicted mask images of the large intestine cancer dataset.

6. Conclusions and Future Work

The main objective of this paper is to automatically build and optimize a U-Net model using a fuzzy-controlled multicell gene expression programming algorithm to address the medical image discrimination problem. The proposed FMCGEP-U-Net aims to automatically determine the optimal depth and parameter settings by combining fuzzy control and GEP. This approach enhances the performance of the U-Net model while reducing the human cost involved. Experimental results demonstrate that FMCGEP-U-Net achieves superior segmentation results when applied to COVID-19 CT images and colorectal cancer image data. However, it is important to further explore the training results using larger datasets, as the current study employed a small dataset.

Medical images present unique challenges, including high latitude, large data volume, and category imbalance. Overcoming these challenges would greatly benefit model training and performance evaluation. If we can effectively address the problem of medical image acquisition, we can obtain high-quality data images, enhance model accuracy, reduce data bias, and increase the overall application value. Therefore, solving the issue of medical image acquisition would have a positive impact on the medical field.

Furthermore, it is worth exploring different types of neural network structure, such as segNet and DenseNet, to address specific image-related problems in various domains. Additionally, future research should focus on using more complex neural network models to optimize network architectures. There are still numerous possibilities for the application of deep learning, such as resolving data privacy concerns, improving the protection of personal information, and potentially achieving qualitative improvements in training time. Future research efforts need to thoroughly investigate these issues and propose effective solutions to unleash the full potential of deep learning in diverse fields.

Acknowledgments

This work was supported by the university-level scientific research projects of Guangxi University of Chinese Medicine #2020MS006, Guangxi University of Chinese Medicine-

Shenzhen Kangtai Biological Products Co., Ltd. Guangxi hepatitis B prevention and control project #QT022019, and Research on Data Storage Technology for Online Shops in the Traditional Chinese Medicine Industry #2012YJZX04. Xinyan Gan and Yuanxia Zhang contributed equally to this work and should be considered corresponding authors.

References

1 A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby: arXiv (2020). https://doi.org/10.48550/arXiv.2010.11929

2 N. Singh, M. Srivastava, S. Mohan, A. Ali, V. K. Singh, and P. Singh: Commun. Comput. Inf. Sci. **1749** (2023) 549. https://doi.org/10.1007/978-3-031-25088-0_49

3 J. Li, F. Jiang, J. Yang, B. Kong, M. Gogate, K. Dashtipour, and A. Hussain: Neurocomput. **465** (2021) 15. https://doi.org/10.1016/j.neucom.2021.08.105

4 F. Ning, D. Delhomme, Y LeCun, F. Piano, L. Bottou, and P. E. Barbano: Proc. IEEE Trans. Image Processing. **14** (2005) 1360. https://doi.org/10.1109/TIP.2005.852470

5 A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis: Comput. Intell. Neurosci. **2018** (2018) 1. https://doi.org/10.1155/2018/7068349

6 A. Krizhevsky, I. Sutskever, and G. E. Hinton: Commun. ACM **6** (2017) 84. https://doi.org/10.1145/3065386

7 K. He, X. Zhang, S. Ren, and J. Sun: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2016) 90. https://doi.org/10.1109/CVPR.2016.90

8 G. Huang, Zhuang Liu, L. Van Der Maaten, and K. Q. Weinberger: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2017) 243. https://doi.org/10.1109/CVPR.2017.243

9 M. Tan and Q. V. Le: arXiv (2019). https://doi.org/10.48550/arXiv.1905.11946

10 A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam: arXiv (2017). https://doi.org/10.48550/arXiv.1704.04861

11 X. Zhang, X. Zhou, M. Lin, and J. Sun: arXiv (2017). https://doi.org/10.48550/arXiv.1707.01083

12 V. Badrinarayanan, A. Kendall, and R. Cipolla: Proc. IEEE Trans. Pattern Analysis and Machine Intelligence. **39** (2017) 2481. https://doi.org/10.1109/TPAMI.2016.2644615

13 J. Stewart, P. Sprivulis, and G. Dwivedi: Emerg. Med. Australasia. **30** (2018) 1742. https://doi.org/10.1111/1742-6723.13145

14 K.-J. Tsai, C.-C. Chang, L.-C. Chien, J.-Y. Chiang, C.-S. Chang, and Y.-J. Huang: Med. **100** (2021) e27649. https://doi.org/10.1097/MD.0000000000027649

15 Z. Li, J. Pan, H. Wu, Z. Wen, and J. Qin: Med. Image Comput. Comput. Assisted Intervention **12264** (2020) 197. https://doi.org/10.1007/978-3-030-59719-1_20

16 M. Landgren: Segmentation of Medical Images, Applications in Echocardiography and Nuclear Medicine. Licentiate Thesis, Lund University (2014).

17 O. Ronneberger, P. Fischer, and T. Brox: Med. Image Comput. Comput. Assisted Intervention **9351** (2015) 234. https://doi.org/10.1007/978-3-319-24574-4_28

18 N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni: IEEE Access **9** (2021) 234. https://doi.org/10.1109/ACCESS.2021.3086020

19 X.-X. Yin, L. Sun, Y. Fu, R. Lu, and Y. Zhang: J. Healthcare Eng. **2022** (2022). https://doi.org/10.1155/2022/4189781

20 G. Du, X. Cao, J. Liang, X. Chen, and Y. Zhan: Imaging Sci. Technol. **2** (2020) 64. https://info:doi/10.2352/J.ImagingSci.Technol.2020.64.2.020508

21 X. Chen, L. Yao, and Y. Zhang: arXiv (2020). https://doi.org/10.48550/arXiv.2004.05645

22 Y.-J. Huang, Q. Dou, Z.-X. Wang, L.-Z. Liu, Y. Jin, C.-F. Li, L. Wang, H. Chen, and R. H. Xu: IEEE Trans. Cybern. **51** (2021) 5397. http://dx.doi.org/10.1109/TCYB.2020.2980145

23 H. Huang, L. Lin, R. Tong, Hongjie Hu, Q. Zhang, Y. Iwamoto, X. Han, Yen-W. Chen, and Jian Wu: Proc. 2020 IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP, 2020) 1055. https://doi.org/10.1109/ICASSP40776.2020.9053405

24 E. Ayan: Arabian J. Sci. Eng. Computer Engineering and Computer Science. **45** (2023) 142. https://doi.org/10.1007/s13369-023-07916-4

25 T. Yu and H. Zhu: arXiv (2020). https://doi.org/10.48550/arXiv.2003.05689

26 G. B. Orr and K.-R. Müller, Eds.: Neural Networks **7** (2012) 41. https://doi.org/10.1007/3-540-49430-8

27 B. James and B. Yoshua: (2012). ICRM-2011 Int. Chemometrics Research Meeting. Chemometrics and Intelligent Laboratory Systems **1** (2012) 111. https://doi.org/10.1016/j.chemolab.2011.12.002

28 B. Letham, B. Karrer, G. Ottoni, and E. Bakshy: Bayesian Anal. **14** (2018) 495. https://doi.org/10.1214/18-BA1110
29 R.-R. Griffiths and J. M. Hernández-Lobato: Chem. Sci. **2** (2020) 11. https://doi.org/10.1039/C9SC04026A
30 H.-P. Nguyen, J. Liu, and E. Zio: Appl. Soft Compute. **89** (2020) 106. https://doi.org/10.1016/j.asoc.2020.106116
31 Y. Bo, Q. Liu, X. Huang, and Y. Pan: Tunnelling and Underground Space Technol. **124** (2022) 104448. https://doi.org/10.1016/j.tust.2022.104448
32 Y. Zhu, D. V. Vargas, and K. Sakurai: 6th Int. Symp. Computing and Networking Workshops (2018) 472. https://doi.org/10.1109/CANDARW.2018.00091
33 Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin: IEEE Trans. Cybernetics **51** (2021) 3115. https://doi.org/10.1109/TCYB.2020.2979930
34 C. Ferreira: Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. **2** (2001) 13. https://doi.org/10.1007/3-540-32849-1
35 Y. Sun, B. Xui. M. Zhang, and G. G. Yen: IEEE Trans. Neural Networks and Learning Systems **31** (2019) 41242. https://doi.org/10.1109/TNNLS.2019.2919608
36 M. Tan and Q. V. Le: arXiv (2019). https://doi.org/10.48550/arXiv.1905.11946
37 N. Siddique, S. Paheding, C. P. Elkin, and V. Vedabhaktuni: IEEE Access **9** (2021) 82031. https://doi.org/10.1109/ACCESS.2021.3086020
38 J. Zhong, Y.-S. Ong, and W. Cai: IEEE Trans. Evolutionary Computation **20** (2016) 65. https://doi.org/10.1109/TEVC.2015.2424410
39 D. Gong, Y. Peng, C. Deng, C. Yuan, A. Cao, and H. Li: Comput. Eng. Design **8** (2021) 2186. https://doi.org/10.16208/j.issn1000-7024.2021.08.013
40 L. A. Zadeh: Inf. Control **8** (1965) 338. https://doi.org/10.1016/S0019-9958(65)90241-X
41 D. Jiang, Z. Tian, Z. He, G. Tu, and R. Huang: Natural Comput. **20** (2021) 395. https://doi.org/10.1007/s11047-020-09830-2
42 L. YongShun and H. Pei: Proc. 4th Int. Conf. High Performance Compilation, Computing and Communications (2020) 150. https://doi.org/10.1145/3407947.3407966
43 J. K. Ravalico, H. R. Mainer, and G. C.Dandy: Reliability Eng. System Saf. **94** (2009) 1229. https://doi.org/10.1016/j.ress.2009.01.009
44 S. Das, M. K. Swain, G. K. Nayak, S. Saxena, and S. C. Satpathy: Multimed. Tools and Appl. **81** (2022) 34717. https://doi.org/10.1007/s11042-021-11273-5
45 J. Yang, J. Zhu, H. Wang, and X. Yang: Biomed. Signal Proc. Control **68** (2021) 102643. https://doi.org/10.1016/j.bspc.2021.102643
46 A. Farahani and H. Mohseni: Neural Comput. Appl. **33** (2021) 6307. https://doi.org/10.1007/s00521-020-05396-3
47 J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou: arXiv (2021). https://doi.org/10.48550/arXiv.2102.04306
48 J. Long, E. Shelhamer, and T. Darrell: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2015) 3431.
49 N. Wang, S. Lin, X. Li, K. Li, Y. Shen, Y. Gao, and L. Ma: IEEE Trans. Medical Imaging **3** (2023) 1649. https://arxiv.org/pdf/2206.00902.pdf

About the Authors

Daoqing Gong received his B.S. degree in software engineering from Guangxi Normal University, Nanning, China, in 2017 and his M.S. degree in computer application technology from Normal University, Nanning, China, in 2020. From 2020 to 2023, he was a teacher at Guangxi University of Chinese Medicine. He is a CCF member and an ACM member. He has published more than 10 papers in high-quality SCI journals. His research field includes medical image processing, deep learning, and evolutionary computation. (18775300556@163.com)

Jiayan Yang is an undergraduate student in the Department of Medical Information, Guangxi University of Traditional Chinese Medicine. She is engaged in the research of medical image segmentation with her supervisor Daoqing Gong. (18178052551@163.com)

Xinyan Gan received her Bachelor of Engineering degree in Industrial Management Engineering from Wuhan University of Technology, Wuhan, China, in 1995. She pursued her Master of Engineering degree in Power System and its Automation from Guangxi University, Nanning, China, from September 2009 to May 2013. Her studies in this program were focused on studying information management and information system design and development. (gan-yan_88@163.com)

Xiang Gao received his master's degree from Auburn University, U.S.A. He now works in the School of Public Health and Management, Guangxi University of Chinese Medicine. His main research area is artificial intelligence. (gaox@gxtcmu.edu.cn)

Yuanxia Zhang received his bachelor's degree in computer software from Central China Normal University in 1998 and his master's degree in computational mathematics from Dalian University of Technology in 2009. His current interests include big data processing and intelligent algorithms. (jgxyzyx@126.com)