Sensors and Materials, Vol. 35, No. 12 (2023) 4061–4083
MYU Tokyo

S & M 3468

Automatic Construction of U-Net Network Based on Genetic Algorithm for Medical Image Segmentation

Daoqing Gong,[1] Ningwei Mo,[1] Xinyan Gan,[1*]
Yuzhong Peng,[2*] Xiang Gao,[1] and Jiayuan Pan[1]

[1]School of Public Health and Management, Guangxi University of Chinese Medicine, Nanning 530299, China
[2]School of Computer and Information Engineering, Nanning Normal University, Nanning 530100, China

(Received July 14, 2023; accepted November 30, 2023)

Keywords: genetic algorithm, adaptive U-Net framework, image identification

In recent years, U-Net has been widely utilized for the segmentation of medical biological images, demonstrating favorable outcomes. However, determining the optimal U-Net network structure for different datasets remains a challenge, often requiring an extensive architecture search or inefficient integration of various deep models for testing purposes. In this paper, we propose an automatic U-Net network design algorithm, U-Net-GA, based on the genetic algorithm. The algorithm effectively addresses the image discrimination task through the introduction of a new variable-length coding strategy, acceleration components, and genetic operators. The key advantage of the proposed algorithm lies in its "automatic" nature, enabling users to obtain the optimal U-Net network structure for a given image without requiring U-Net domain knowledge. The algorithm's effectiveness is demonstrated by its application to two different types of medical image dataset, namely, colorectal cancer and COVID-19 CT images, and a subsequent comparison with other advanced network structures. Experimental results demonstrate that the proposed algorithm exhibits superior performance compared with existing U-Net networks in terms of segmentation accuracy, Dice coefficient, Jaccard index, and loss index.

1. Introduction

As a broad field,[1] computer vision is closely related to image processing, which holds significant scientific research value and promising applications in aerospace,[2] traffic safety,[3] autonomous driving,[4] and healthcare.[5] With the emergence of the convolutional neural network (CNN),[6] neural network methods based on CNN have been rapidly applied to image segmentation tasks, including natural and medical images.[7–10] U-Net[11] demonstrates robust performance and adaptability, especially when applied to biomedical segmentation tasks.[12] However, designing advanced U-Net networks such as IterNet,[13] residual U-Net,[14] and dense U-Net[15] has traditionally relied on manual expertise from domain knowledge experts. Not every user possesses this domain knowledge, and those familiar with the data may lack

*Corresponding author: e-mail: 18775300556@163.com
https://doi.org/10.18494/SAM4588

ISSN 0914-4935

experience in designing U-Net network structures. Consequently, automatic U-Net network design has become a prominent research topic. The development of automatic tuning methods for U-Net networks enables users without specialized U-Net knowledge to explore their own network structures. Moreover, U-Net network design algorithms can enhance medical image processing technology and contribute to the advancement of medical AI.

On the basis of whether domain knowledge is needed, existing U-Net network design algorithms can be divided into two groups. The first category is the "automatic + manual tuning" U-Net network architecture design, indicating that manual adjustments are required on the basis of expertise in U-Net network design. This category includes the hierarchical evolution,[16] efficient architecture search method (EAS),[17] block design method (Block-QNN-S),[18] and advanced neural structure search method (NSAnet).[19] The other category is the "automatic" U-Net network architecture design, which eliminates the need for manual adjustments by the user. Prominent examples of this category include nnU-Net[20] and SegNAS3D.[21] Sun *et al.*[22] introduced the CNN-GA algorithm and highlighted the superiority of "automatic" CNN architecture design over the "automatic + manual adjustment" approach. Similarly, Isensee *et al.* presented nnU-Net,[20] which explores a dynamic framework for segmenting medical images utilizing the U-Net approach. Both studies emphasize that the network structure designed with an "automatic" approach does not require manual adjustments, making it an ideal choice for users lacking domain knowledge in U-Net.

Currently, there are two methods being adopted to enhance the automation of neural networks: design based on reinforcement learning[23] and design based on evolutionary algorithms.[24] Experimental findings suggest that reinforcement-learning-based designs typically demand greater computational resources when contrasted with designs grounded on evolutionary algorithms. In the context of U-Net, Montana and Davis[25] conducted an early study utilizing genetic algorithms (GAs)[26] for network weight initialization, demonstrating favorable performance. Liu *et al.*[27] introduced an innovative single-point crossover operator and a variation operator designed for WCDMA network planning. Nag and Pal[28] devised a variant operator that exploits the versatility and inflexibility of feature selection using an ensemble approach that combines feature selection and classification. GAs have found application in a wide array of medical image analysis tasks, encompassing tasks such as segmentation, registration, disease identification, and image noise reduction. For instance, Fan *et al.*[29] employed GAs to address a numerically unstable brain volume segmentation based on active models. Matsopoulos *et al.*[30] optimized affine transformation parameters using GAs for improved multimodal retinal image registration. Quellec *et al.*[31] employed GAs to fine-tune wavelet transform parameters in order to enhance the detection of microaneurysms in retinal images. Therefore, in our experiments involving two distinct categories of medical images, the use of evolutionary algorithms effectively enhances the detection performance in medical image tasks.

By combining "automatic" U-Net architecture and evolutionary algorithms, U-Net-GA revolutionizes medical image segmentation. This innovative algorithm autonomously discovers the optimal U-Net network structure, accessible even to users without specialized knowledge. U-Net-GA simplifies U-Net architecture, expediting groundbreaking results. In medical image

analysis, U-Net-GA excels as an elegant and efficient solution. It eliminates the need for extensive expertise, automating network structure discovery. By merging the "automatic" U-Net architecture with GA's capabilities, U-Net-GA unlocks the potential of medical AI, surpassing traditional methods. Efficiency prevails with U-Net-GA, navigating diverse U-Net network structures for optimal performance, accuracy, and reduced resource consumption. Users embark on a transformative journey, unveiling hidden insights and pushing healthcare boundaries. U-Net-GA is our commitment to advancing medical image analysis, integrating cutting-edge research and technology to empower users and revolutionize medical image segmentation.

The proposed algorithm fosters innovation through its provision of a completely automated method for uncovering the most suitable U-Net network, thus eradicating the necessity for human involvement throughout the evolutionary search phase. The U-Net-GA algorithm's contributions can be summarized as follows:

1) A variable-length encoding approach and its associated crossover operator have been introduced. Because the GA itself is inspired by the biological evolution algorithm, it traditionally uses a fixed length coding strategy. However, when applying it to the U-Net network, the length of encoding for each component is unknown. This can result in mis-specified quantities and add unnecessary complexity. While many researchers have developed independent variable-length coding strategies, the effectiveness and optimality of these strategies may be compromised owing to the lack of redesigned corresponding crossover operators. In contrast, our proposed algorithm addresses these challenges and provides a solution.
2) This method can be extended to the processing of most medical image data sets. The U-Net network structure has made great achievements in the segmentation task of biomedical images, but most medical images have small data sets and lack classification, so it is difficult to obtain verification. In this paper, we verify the algorithm's performance using a significant medical image dataset, which includes colorectal cancer and COVID-19 CT images. The results demonstrate that our approach can be applied effectively to various other medical image segmentation tasks, marking substantial advancements in the field.
3) This algorithm significantly accelerates the U-Net network architecture design process. To enhance U-Net design efficiency and deliver optimal networks to users, many existing algorithms consume substantial computational resources, particularly when employing data parallel strategies, which can be cost-prohibitive. In the proposed algorithm, we have implemented an asynchronous computing component that maximizes the utilization of computing resources for expediting individual fitness evaluations. Additionally, a caching mechanism is employed to further reduce the fitness evaluation time for the entire population, ultimately achieving superior results.

The rest of this article is organized as follows. First, Sect. 2 describes the work and background. The proposed algorithm is then discussed in detail in Sect. 3. Next, the experimental design is described in Sect. 4 and the experimental results and analysis are shown in Sect. 5. Finally, Sect. 6 outlines the conclusions and future work.

2. Related Work

In this section, GA and the U-Net network structure are introduced, so that readers can better understand our proposed algorithm and related work.

2.1 Genetic algorithm

GA[26] is a meta-heuristic algorithm inspired by natural selection and belongs to the category of evolutionary algorithm. This method finds extensive application in the pursuit of top-tier solutions to optimization problems by performing operations such as selection, crossover, and mutation (e.g., TSP,[32] Neuroevolutionary, and[33,34] NAS[35]). GA consists of fundamental components such as chromosome representation, fitness selection, and biological heuristic operators. Figure 1 shows the flow diagram of GA.

1) Encoding: First, we encode the architecture and parameters of U-Net as a chromosome, where each gene in the chromosome represents the parameters of a specific layer.
2) Initializing the population: GA begins by initializing a population consisting of multiple individuals (also known as Individuals). Everyone represents a potential solution to the optimization problem. The variables to be optimized are encoded into the genes of the individual, and these genomes form chromosomes that represent the genotype of the individual. Typical encoding techniques encompass binary encoding, real-value encoding, and character encoding.
3) Fitness assessment: The fitness function is typically defined to represent the objective function that needs to be solved. It is used to evaluate the performance of everyone in the population. For medical image segmentation tasks, the fitness function can be defined using segmentation performance indicators, such as the Dice coefficient. After evaluation, each individual is assigned a fitness value on the basis of its performance.[36]
4) Selection operation: The selection operation's purpose is to choose individuals with high fitness values to continue existing in the population, allowing their genes to persist and influence the population. The specific selection method should be determined on the basis of the fitness function, favoring individuals with higher fitness values for the next generation. The selection operation plays a crucial role in balancing the algorithm's convergence and maintaining population diversity. Commonly used selection methods include tournament selection[37] and roulette selection.[38]
5) Crossover and mutation: The crossover operation is a critical stage in GA. It involves selecting pairs of individuals to mate, randomly choosing crossover points within their genes,

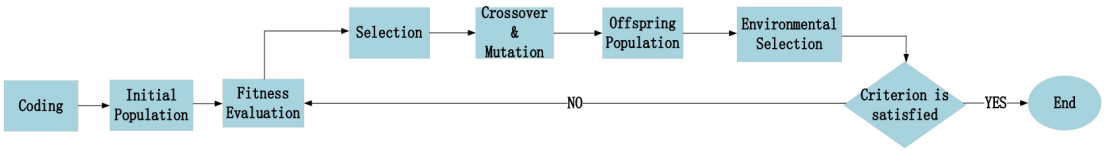

Fig. 1. (Color online) Flow diagram of GA.

and exchanging gene segments between them. Mutations are introduced to maintain population diversity and prevent premature convergence. Occasionally, in the newly generated offspring, certain genes may undergo mutation with a low random probability. This means that some bits in the gene string can change randomly, introducing variations into the population.

6) New generation population: The new generation population is a refined population gradually generated through the previous five processes. The newly generated individuals replace the old individuals to form the new generation population.
7) Environmental selection: We perform environmental selection operation following Darwin's theory of "survival of the fittest."[39] The new generation population is placed in the environment to select the best individuals.
8) Iteration: After environmental selection, we evaluate whether the individuals in the new generation population meet the requirements. If the conditions are not met, we repeat steps 3–7 until the conditions are satisfied, and then the iterative operation ends.

Through the application of GA's crossover and mutation operations, fresh individuals are created, and an iterative environmental selection process allows us to attain optimal outcomes in deep learning tasks. This is the primary motivation behind employing GA in constructing our "automation" algorithm.

2.2 U-Net

The U-Net network model primarily comprises an encoder, a decoder, and skip connections. On the left side of the U-Net model lies the encoding path, also referred to as the compression path, composed of multiple repeated compression blocks. The compression module adheres to the standard architecture commonly found in CNNs, incorporating elements such as 3 × 3 convolutional layers, ReLU activation functions, batch normalization layers, and 2 × 2 pooling layers. Figure 2 shows the U-Net network model diagram.

On the right side is the decoding path, also referred to as the expansion path. It bears a close resemblance to the compression path in terms of overall structure and comprises multiple recurring expansion modules. The expansion module differs from the compression module by employing deconvolution operations instead of convolution operations and incorporating skip connections from the compression path. With this design, U-Net exhibits a simple and elegant U-shaped structure, as specifically shown in Fig. 2.

To illustrate the U-net model in more detail, several of the key components will be introduced.

1) Convolutional layer. It comprises a collection of convolutional kernels, with adjustable height and width parameters. The adaptable convolutional kernel empowers the convolution layer to abstract feature information from the input and transform it into a fresh feature space. Furthermore, utilizing shared weights configuration reduces the computational complexity of the convolutional layer in contrast to a fully connected layer. For each convolution kernel, its output can be represented by

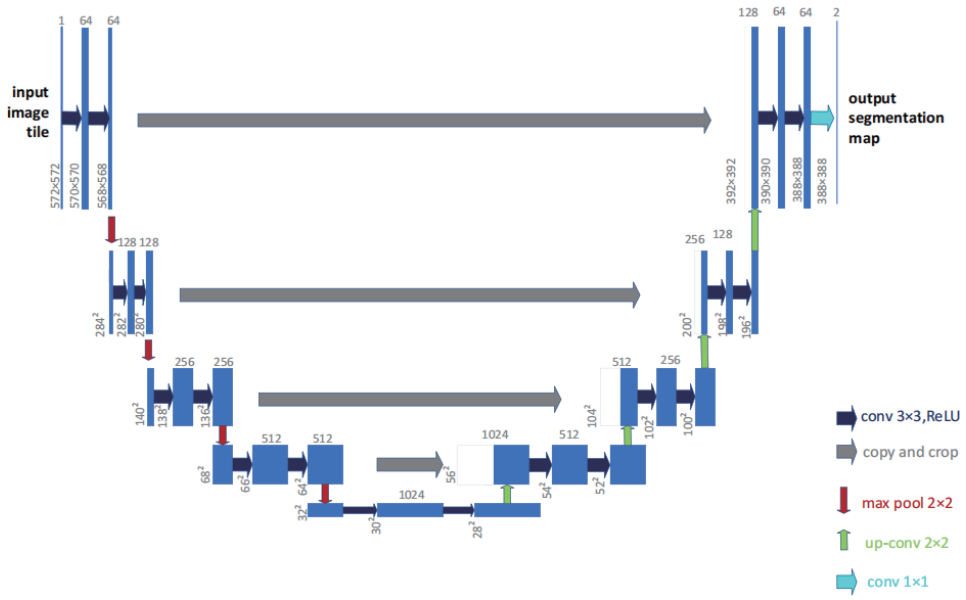

Fig. 2. (Color online) U-Net network model diagram.

$$O_i = f\left(\sum_j w_{ij} \times x_j + b_i\right), \tag{1}$$

where x_j is the j input channel, O_i is the i channel of feature mapping, f is the activation function, w_{ij} is the weight, and b_i is the bias term.[11]

2) Deconvolution (sometimes referred to as transposed convolution). In the U-net model, its role is to increase the resolution of the feature map. To obtain the expansion ability that matches the compression path, the deconvolutional layer adopts a deconvolution kernel of 2 × 2 and sets the convolution step to 2.
3) Activation functions are used to introduce nonlinearity in convolutional networks, enabling them to model complex nonlinear mappings. Common activation functions include Sigmoid functions, Tanh functions, ELU, ReLUs, and more. Among these, rectified linear units (ReLUs) are widely used as activation functions owing to their ability to mitigate the issues of gradient vanishing and explosion, while also facilitating faster model learning. In our approach, the ReLU function is employed as the activation function in both the compression and expansion modules. The ReLU function is computed as

$$ReLU(x) = \max(0, x). \tag{2}$$

4) Batch normalization is a crucial component for addressing the issue of feature distribution shifts during batch training. Significantly drifting feature distributions can compromise network training stability and exacerbate overfitting, particularly in deep neural networks. Batch normalization processes involve transforming the outputs of each neuron layer into a standard normal distribution.

Throughout the training phase, the U-Net network utilizes random elastic deformation as a data augmentation method, effectively augmenting the dataset despite having a limited number of samples. This approach mitigates the impact of training on a small sample size, improving the network's robustness and generalization. Additionally, Dropout is applied at the end of the shrinkage path to further augment the data implicitly, effectively combating overfitting. Moreover, the network utilizes the cross-entropy loss function to optimize the final feature map at the pixel level.

3. Proposed Algorithm

In this section, we provide an overview of the proposed algorithm framework discussed in Sect. 3.1. Subsequently, we delve into the specifics of each major step in Sects. 3.2 to 3.5. Our objective is to enhance readers' understanding of the proposed algorithm by meticulously documenting the details of each step and conducting a comprehensive analysis of the design choices made.

3.1 Algorithm overview

Figure 3 is the overall framework diagram, through the input of two types of image data, coding operations, initializing population operations, forming population chromosomes, encoding and decoding processes through the U-Net model, using the Dice coefficient and Jaccard index for fitness assessment, and then generating a new generation of populations through cross-crossing and mutation operations, and finally, environmental selection; that is, judging that it meets the requirements, the optimal network is output, and if it does not meet the requirements, the fitness assessment is returned to continue the operation.

Algorithm 1 outlines the framework of the proposed algorithm. The objective is to obtain the optimal network structure of U-Net for a given image dataset through evolutionary processes, while considering the population size, the maximum number of generations, and the predefined

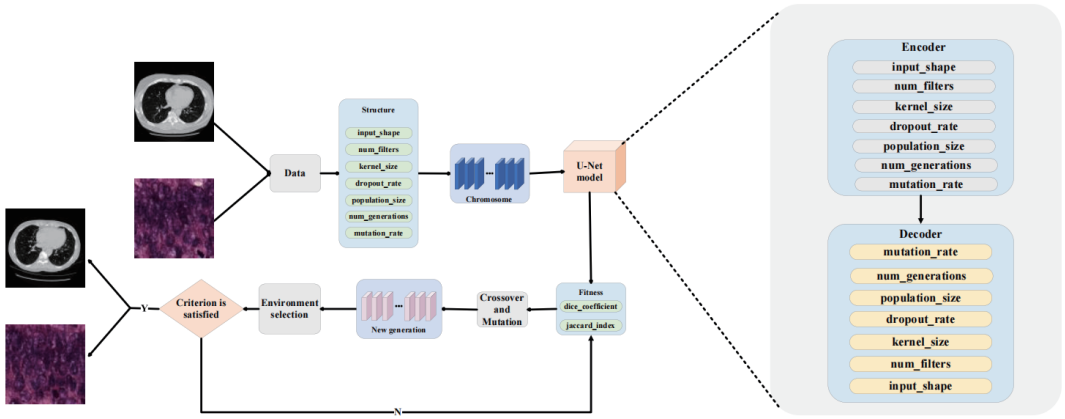

Fig. 3. (Color online) General framework of proposed method.

U-Net building blocks. To kick start the evolutionary process, an initial population is generated randomly, following the specified population size. The predefined building blocks are subsequently encoded using the proposed encoding strategy (line 1). Following this, the current generation counter is initialized to zero (line 2).

Throughout the evolution, each individual's fitness is assessed using the provided dataset, a process that encompasses encoding the U-Net's specific network structure (line 4). Subsequently, parent individuals are chosen on the basis of their fitness, and new offspring are produced through genetic operations such as crossover and mutation (line 5). The population that proceeds to the next generation is determined through environmental selection, with the current population comprising both the parent population and the newly generated offspring population (line 6). This process continues as the counter increments by 1, and the evolution persists until the counter surpasses the predefined maximum generation limit.

3.2 Population initialization

As discussed in Sect. 2, the U-Net architecture is composed of convolutional and pooling layers, and its performance is significantly affected by the network structure. To address this, we introduce a novel building block during the population initialization phase. In our proposed encoding strategy, we ensure that the feature maps have consistent sizes. Moreover, we employ a fixed convolution operation and incorporate a 1 × 1 stride to enhance the automatic design's flexibility. During individual initialization, we follow the following steps in algorithm 2. First, we randomly determine the length of the individual, which corresponds to the depth of the U-Net network and is denoted as L (line 3). Subsequently, we create a linked list consisting of L nodes (line 4). Each node is then confit *d* with specific parameters and settings (lines 5–20) before being stored in the P0 list (line 21). These steps help establish the initial population with diverse network structures, ensuring that the evolutionary process can explore different architectural configurations. By leveraging the proposed encoding strategy and individual initialization, we pave the way for the subsequent genetic operators and environmental selection to iteratively improve the network's performance and converge towards an optimal U-Net structure for a given image dataset.

Algorithm 1: Proposed Algorithm

Input: input shape, train generator, validation generator, population size, num generations, mutation rate, num filters, kernel size, dropout rate

Output: The discovered best network structure of U-Net

1 $P_0 \leftarrow$ Initialize a population with the given population size using the proposed variable-length encoding strategy;
2 $t \leftarrow 0$;
3 **while** $t <$ the maximal generation number **do**
4 Evaluate the fitness of each individual in P_t using the proposed acceleration components;
5 $Q_t \leftarrow$ Generate offspring from the selected parent individuals using the proposed mutation and the crossover operators;
6 $P_{t+1} \leftarrow$ Environmental selection from $P_t \cup Q_t$;
7 $t \leftarrow t + 1$;
8 **end**
9 **return** the individual having the best fitness in P_t

Algorithm 2: Population Initialization

Input: The population size T
Output: The initialized population P_0

$P_0 \leftarrow \varnothing$;
while $|P_0| < T$ **do**
  $L \leftarrow$ Randomly generate an integer greater than zero;
  list $\leftarrow$ Create a linked list that contains L nodes;
  for each node in the linked list **do**
    $r \leftarrow$ Uniformly generate a number from (0, 1);
    if $r < 0.5$ **then**
      node. type $\leftarrow$ 1;
      node. $F_1 \leftarrow$ Randomly generate an integer greater than zero;
      node. $F_2 \leftarrow$ Randomly generate an integer greater than zero;
    else
      node. type $\leftarrow$ 2;
      $q \leftarrow$ Uniformly generate a number from (0, 1);
      if $q < 0.5$ **then**
        node. $P_1 \leftarrow$ max;
      else
        node. $P_1 \leftarrow$ mean;
      end
    end
  end
  $P_0 \leftarrow P_0 \cup$ *list*;
end
return P_0

3.3 Adaptability assessment

Algorithm 3 presents the process of evaluating the fitness of individuals in a population (P_t) for image segmentation using U-Net. The goal is to find the optimal network structure that performs well on the given image segmentation dataset. In this algorithm, each individual in P_t is evaluated in the same manner. To accelerate the evaluation process, individuals are asynchronously placed on available GPUs. GPUs are commonly used for deep learning tasks due to their computational power.[40] The algorithm incorporates an asynchronous cache component, which is a parallel computing platform based on GPUs. Each GPU executes a small model, allowing for the efficient utilization of GPU resources and reducing the processing time for large-scale problems.

With a designated GPU, U-Net is trained on the training data using the stochastic gradient descent (SGD) algorithm.[41] The segmentation accuracy is then calculated using the fitness evaluation data. To optimize the evaluation process, a cache component is designed to store a string of U-Net identifiers combined with their corresponding fitness values. This caching mechanism helps avoid redundant fitness evaluations, enabling a faster convergence and an efficient exploration of the solution space. By employing asynchronous evaluation and caching mechanisms, Algorithm 3 enables parallel computation, reduces evaluation time, and enhances the efficiency of searching for the optimal U-Net network structure for image segmentation tasks.

Algorithm 3: Individual Fitness Evaluation

Input: The individual Individual, the available GPU, the number of training epochs, the global cache Cache, the training data D_{train}, and the fitness evaluation data $D_{fitness}$ from the given image segmentation dataset
Output: The individual Individual with its fitness

```
1  Construct a U-Net with a classifier based on the information encoded in individual and the given image
   segmentation dataset;
2  V_best ← 0;
3  for each epoch in the given training epochs do
4      Train the CNN on D_train by using the given GPU;
5      V ← Calculate the segmentation accuracy on
6      D_fitness;
7      if v > V_best then
8          V_best ← V;
9      end
10 end
11 Set V_best as the fitness of individual;
12 Put the identifier of individual and V_best into Cache;
13 return individual
```

3.4 Offspring generation

Algorithm 4 is divided into two parts: crossover (lines 1–18) and mutation (lines 19–26). In the crossover operation, two parents with the best fitness are selected using binary tournament selection (lines 3–7).[36] Once the parents are chosen, a random number is generated (line 8) to determine whether a crossover should occur. If the generated number is greater than the predefined crossover probability, both parent individuals are combined to produce offspring in Q_t (line 16). On the other hand, if the generated number is below the crossover probability, each parent individual is randomly divided into two parts, and these parts are exchanged to create two offspring (lines 10–14). The purpose of the crossover operation is to exchange genetic information between parents and generate diverse offspring that inherit desirable traits from both parents. By incorporating randomness and exploration, crossover promotes genetic diversity within the population, which can lead to improved solutions. Note that the crossover operation is followed by the mutation operation, which introduces further diversity into the population. However, the mutation operation is described in detail in lines 19–26 of the algorithm.

In the mutation operation, an initial step involves generating a random number (line 20). If this generated random number is less than the mutation probability (P_m), the current individual undergoes mutation, as indicated in lines 21–25. The mutation process entails two key steps: randomly selecting a position (represented as "i") within the individual and determining a specific mutation operation (represented as "m") from the designated mutation list. The choice of the mutation operation is made on the basis of the probabilities defined in P_l. Ultimately, the selected mutation operation is applied to the individual at the chosen position.

The reason for designing this crossover operator for mutation is to address the challenge of individuals with unequal lengths. While single-point crossover in GAs is typically designed for

Algorithm 4: Offspring Generation

Input: The population P_t containing individuals with fitness, the probity for crossover operation P_c, the probability for mutation operation P_m, the mutation operation list l_m, the probabilities of selecting different mutation operations P_l

Output: The offspring population Q_t

1 $Q_t \leftarrow \varnothing$;
2 **while** $|Q_t| < |P_t|$ **do**
3 $p_1 \leftarrow$ Randomly select two individuals from P_t and from the two then select the one with the better fitness;
4 $p_2 \leftarrow$ Repeat Line 3;
5 **while** $p_2 = p_1$ **do**
6 Repeat Line 4;
7 **end**
8 $r \leftarrow$ Randomly generate a number from (0, 1);
9 **if** $r < P_c$ **then**
10 Randomly choose a point in p and divide it into two parts;
11 Randomly choose a point in p_2 and divide it into two parts;
12 $O_1 \leftarrow$Join the first part of p_1 and the second part of p_2;
13 $O_2 \leftarrow$Join the first part of p_2 and the second part of p_1;
14 $Q_t \leftarrow Q_t \cup O_1 \cup O_2$;
15 **else**
16 $Q_t \leftarrow Q_t \cup p_1 \cup p_2$;
17 **end**
18 **end**
19 **for** each individual p in Q_t **do**
20 $r \leftarrow$ Randomly generate a number from (0,1);
21 **if** $r < P_m$ **then**
22 $i \leftarrow$ Randomly choose a point in p;
23 $m \leftarrow$ Select one operation from l_m based on the probabilities in p;
24 Do the mutation m at the point i of p;
25 **end**
26 **end**
27 **return** Q_t

individuals with equal lengths, our crossover operator is specifically designed to accommodate individuals with unequal lengths, thereby enhancing the performance of the network. The combination of crossover and mutation operators provides a foundation for searching for the optimal U-Net network structure. We have designed four different types of mutation operator, and by applying these operators, we can effectively explore and discover the best network structures, with the first three mutation operators playing crucial roles.

3.5 Environmental selection

First, by using the binary tournament selection from the current population, we determine $|P_t|$ as an individual choice, and then we put these choices of individuals in a population (expressed as P_{t+1}) (lines 2–6). Next, the best individual is selected and checked to see if it has been placed in P_{t+1}. If not, it replaces the worst individual in P_{t+1} (lines 7–10). This is the process of selecting

the best individuals that we add to the next population, a particular elitist strategy of evolutionary algorithms. This enables our algorithm to find the optimal result. Algorithm 5 constantly judges whether the new offspring adapts to the environment and carry out health assessment. A series of processes will eventually produce the optimal offspring result.

4. Experiment

To assess the effectiveness of the proposed algorithm, a series of experiments were conducted in the context of image segmentation tasks. Specifically, the proposed algorithm is introduced in Sect. 4.1 on the segmentation experiment on the medical image dataset. Then, in Sect. 4.2, a comparison of the proposed algorithm with the U-Net network is presented. In Sect. 4.3, the parameter settings of the proposed algorithm are described.

4.1 Medical image dataset

Colorectal cancer and self-collected CT images of COVID-19 cases were selected as image segmentation tasks in the experiment. These datasets were chosen owing to their diversity in medical image types, presenting challenges related to image size, classification, noise, and rotation. Furthermore, their extensive study in recent years has significantly contributed to the advancement of AI in the biomedical field.

As shown in Fig. 4, the colorectal cancer images are divided into eight categories, namely, TUMOR, STROMA, COMPLEX, LYMPHO, DEBRIS, MUCOSA, ADIPOSE, and EMPTY. Each category has 625 images, with a total of 5,000 images, and the data format is tif. We divide the data into 80% of the training set and 20% of the verification set.

Figure 5 shows a diseased CT image on the left and a normal CT image on the right. CT segmentation provides more quantitative data and visual characterization of the lesion area. However, owing to the wide spread of COVID-19, it is difficult to effectively collect disaggregated data from different countries as well as hospitals. Moreover, with the COVID-19 lesion segmentation algorithm, it is often difficult to obtain the same good segmentation performance in different medical centers. Hence, enhancing the generalization performance of

Algorithm 5: Environmental Selection

Input: The parent population P_t, the offspring population Q_t.
Output: The population for the next generation P_{t+1}

$P_{t+1} \leftarrow \varnothing$;
while $|P + 1| < |P|$ **do**
    $p_1, p_2 \leftarrow$ Randomly select two individuals from $Q_t \cup P$;
    $p \leftarrow$ Select the one who has a better fitness from $\{p_1, p_2\}$;
    $P + 1 \leftarrow P_{t+1} \cup P$;
end
$P_{best} \leftarrow$ Find the individual with the best fitness from $Q_t \cup P$;
if p_{best} is not in P_{t+1} **then**
    Replace the one who has the worst fitness in P_{t+1} by P_{best};
end
return P_{t+1}

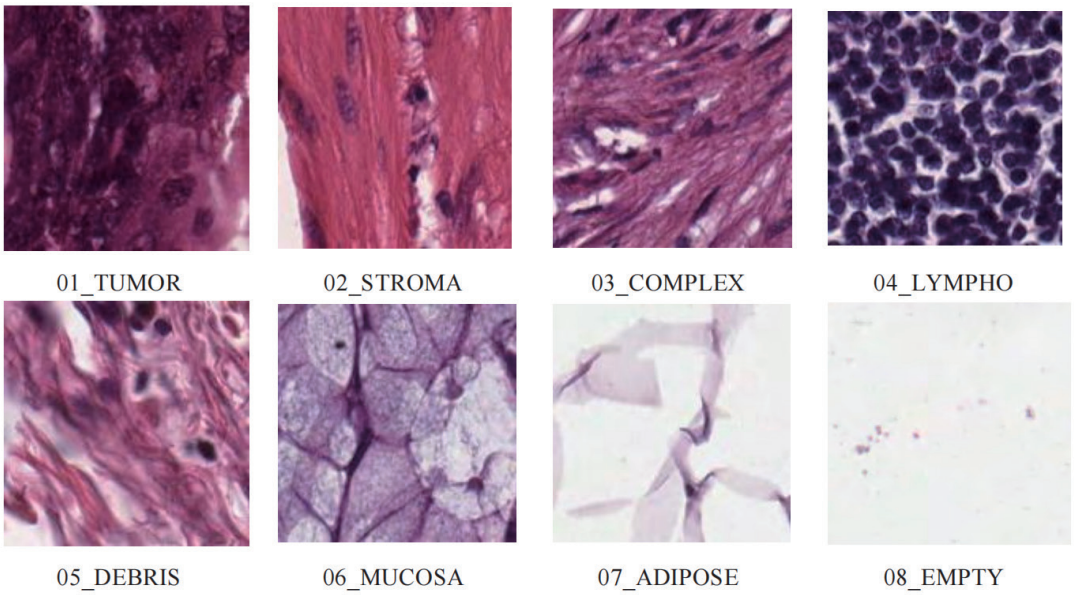

Fig. 4. (Color online) Colorectal cancer images.

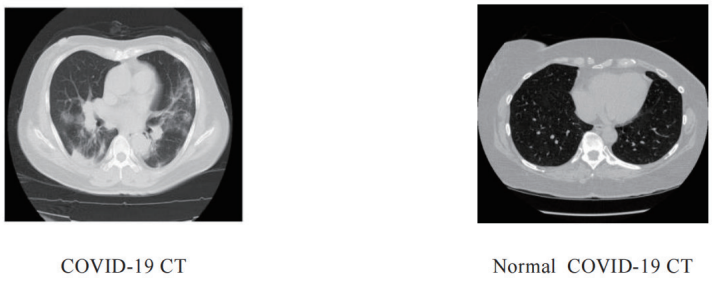

Fig. 5. CT images of COVID-19.

COVID-19 lesion segmentation algorithms on multi-center datasets has significant importance. We collected 1000 normal images and 500 diseased images, for a total of 1500 data. The two types are classified as shown in Fig. 5, and the CT chart is enhanced and the histogram is equalized. Secondly, by defining the ImageDateGenerator, the image generator, and by performing data enhancement processing such as the histogram equalization of the image, a larger sample size is obtained. Next, the image is normalized to mitigate the effect of uneven lighting in medical images. Subsequently, a CNN employs convolutional kernels to extract relevant features and combines similar features through pooling operations to implement dimensionality reduction. We divide the data into 80% of the test set and 20% of the verification set, and select the parameters and hyperparameters of the model by selecting the appropriate loss function to help the model train to obtain better training results.

4.2 Parameter settings

As previously stated, the primary objective of this study is to develop an automatic architecture discovery algorithm that is accessible to researchers without expertise in U-Net. To enhance the usability of the proposed algorithm, we have specifically designed it to eliminate the requirement for users to possess knowledge in evolutionary algorithms. Therefore, Table 1 displays the parameters of the U-Net-GA algorithm as determined by our testing configuration.

4.3 Comparison with U-Net algorithm

Currently, there are many deep learning algorithms that work on CT images of colorectal cancer and COVID-19 cases. Hamida *et al.*[42] found the optimal network for the deep learning analysis of colon cancer histopathological images, and Yildirim and Cinar[43] established a MA_ ColonNET CNN model for the classification of colon adenocarcinoma and colon benign tissue colon histopathological images. Cao *et al.*[44] and Huang *et al.*[45] used U-Net to segment pneumonia lesion areas for quantitative analysis. Shan *et al.*[46] applied VB-net for segmentation. Chaganti *et al.*[47] trained two networks to obtain lung regions and lesions, and used lung region filtering to obtain lesions for automatic segmentation and quantification. Yan *et al.*[48] proposed a CNN-based segmentation network for lesion segmentation. Fan *et al.*[49] proposed that Inf-Net uses edge information to enhance model features, and at the same time, in order to alleviate the problem of insufficient data, a semisupervised framework was proposed. Wu *et al.*[50] combined classification and segmentation models to build an interpretable COVID-19 diagnostic segmentation system. Because the U-Net algorithm is manually adjusted when it is not optimized, in order to demonstrate the effectiveness of the proposed algorithm, we chose to compare it with the U-Net network.

5. Experimental Results and Analysis

We conducted a series of experiments on the image segmentation task. Specifically, the evaluation metrics used in our experiments are described in Sect. 5.1. Then, in Sect. 5.2, the overall experimental results are presented. In Sect. 5.3, the experimental results are discussed.

Table 1
U-Net-GA parameter settings.

Parameter	Value
kernel_size	3
dropout_rate	0.2
population_size	10
num_generations	10
mutation_rate	0.05

5.1 Evaluation index

In this work, we use four metrics in the table to evaluate the performance of the model: Accuracy (ACC), Dice coefficient (Dice), Jaccard index (Jaccard), and loss function (loss). Accuracy is the ratio between the correctly classified pixels and the total pixels. The similarity measurement coefficient (Dice) is a set similarity measurement function, which is often used to calculate the similarity between two samples. The Jaccard similarity coefficient, also known as the Jaccard index, is employed to assess the similarity and dissimilarity between limited sample sets. A larger Jaccard coefficient signifies a greater similarity between samples. Conversely, the loss function quantifies the extent of disparity between the predicted value, denoted as $f(x)$, and the actual value, Y, serving as a measure of model robustness. A smaller loss function value indicates a better model performance. The specific indicators and formulas are shown in Table 2.

5.2 Experimental result

We designed two sets of experiments, one for the model comparison of the colorectal cancer dataset, and the other for the model comparison of the CT image data of COVID-19 cases.

5.2.1 Comparison of colorectal cancer dataset models

As depicted in Fig. 6, the U-Net-GA model demonstrates favorable results across four distinct evaluation metrics. Overall, with each iteration, the model exhibits a steady increase in performance, reaching stability by Epoch 8, with minimal fluctuations observed in the evaluation indices.

Figure 7 shows the predictions of U-Net and U-Net-GA trained models. The predicted results were shown in the form of a mask map, and the white spots represented the tumor distribution. As can be seen from Fig. 7, the segmentation accuracy of the optimized model is clearly higher than that of the benchmark model.

Table 2
Evaluation indicators.

Index	Calculation
ACC (accuracy)	$ACC = \frac{(TP+TN)}{(TP+TN+FP+FN)}$
Dice (Dice coefficient)	$Dice = \frac{2\lvert A \cap B \rvert}{\lvert A \rvert + \lvert B \rvert}$
Jaccard (Jaccard index)	$J(A,B) = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$
Loss (loss function)	$loss = -\frac{1}{n}\sum_i y_i \ln a_i$

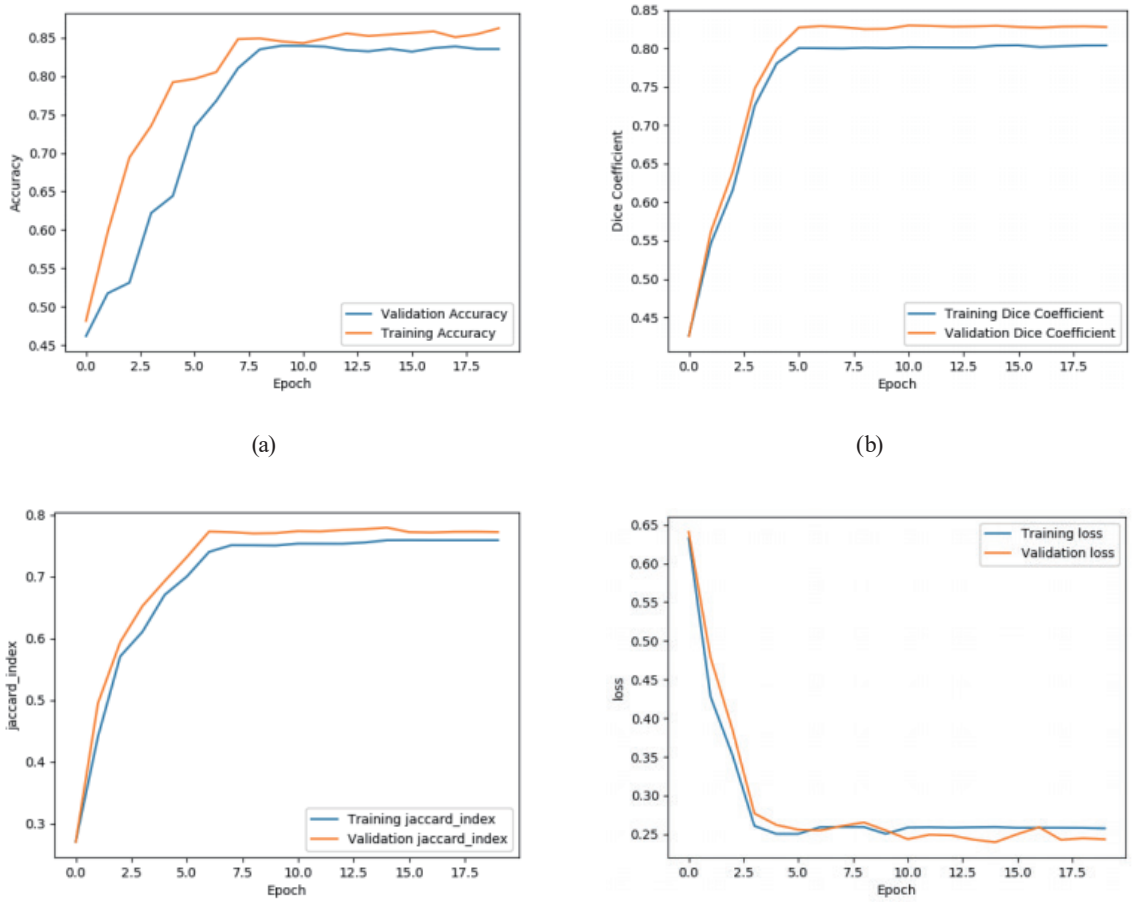

Fig. 6. (Color online) Experimental results of colorectal cancer images on U-Net-GA. (a) Accuracy and epoch. (b) Dice coefficient and epoch. (c) Jaccard index and epoch. (d) Loss function and epoch.

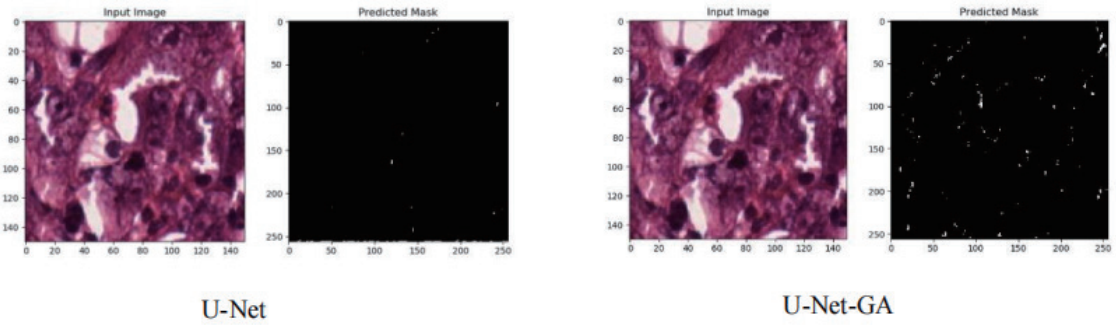

Fig. 7. (Color online) Predicted Mask comparison on U-Net-GA.

Table 3 shows that after training for 20 epochs, U-Net-GA achieves impressive results on the colorectal cancer dataset. U-Net-GA performs well in three out of four different evaluation metrics, but it slightly lags the algorithm presented in Ref. 52 in terms of the 'Dice' metric.

Table 4 shows the change in the number of neurons randomly generated by the U-Net model during each generation of our evolution. We will choose the last case 16,32,64,64,128,256 as the result of our evolution.

5.2.2 Model comparison of CT image dataset for COVID-19 cases

The comparison of the prediction of the mask map in Fig. 8 shows that the prediction result of the U-Net-GA algorithm is much better than that of the U-Net algorithm.

The U-Net-GA algorithm proposed in this paper has an accuracy rate of 95.17% in the colorectal cancer dataset, which is 9.93% higher than that of the U-Net algorithm. The Dice coefficient of the U-Net-GA algorithm is 87.91% and that of the U-Net algorithm is 83.86%, which is generally 4.05% higher. The Jaccard coefficient of the U-Net-GA algorithm is 84.74% and that of the U-Net algorithm is 83.05%, an increase of 1.69%. For the loss function, our U-Net-GA algorithm is comparable to the U-Net algorithm, which has been improved by 0.02%. Compared with the algorithms presented in Refs. 51 and 52, the method proposed in this study achieves better results in all four metrics. The detailed comparison of various algorithms is shown in Table 5.

Table 6 shows the changes in the number of neurons randomly generated by the U-Net model during each generation of evolution, and we will choose the last cases 16, 64, 128, 256 as the result of evolution.

Table 3
Comparison with other state-of-the-art algorithms.

Model	Epoch	*ACC* (%)	*Dice* (%)	*Jaccard* (%)	*Loss*
FCN32[51]	20	81.86	82.50	74.45	0.36
VanillaUnet[52]	20	79.62	89.68	75.36	0.38
U-Net	20	84.36	82.23	74.96	0.31
U-Net-GA	20	85.24	83.86	77.94	0.25

Table 4
Changes in the number of U-Net neurons–colorectal cancer dataset.

Population algebra	Value
1	16, 16, 64, 128
2	16, 16, 32, 128
3	64, 64, 128, 256, 256
4	16, 16, 32, 64, 64, 128, 128, 256
5	16, 32, 64, 128, 256, 256
6	32, 32, 128, 128, 256
7	16, 32, 64, 128, 256
8	32, 32, 64, 64
9	16, 16, 32, 64, 128, 128, 256
10	16, 32, 64, 64, 128, 256
Best result	16, 32, 64, 64, 128, 256

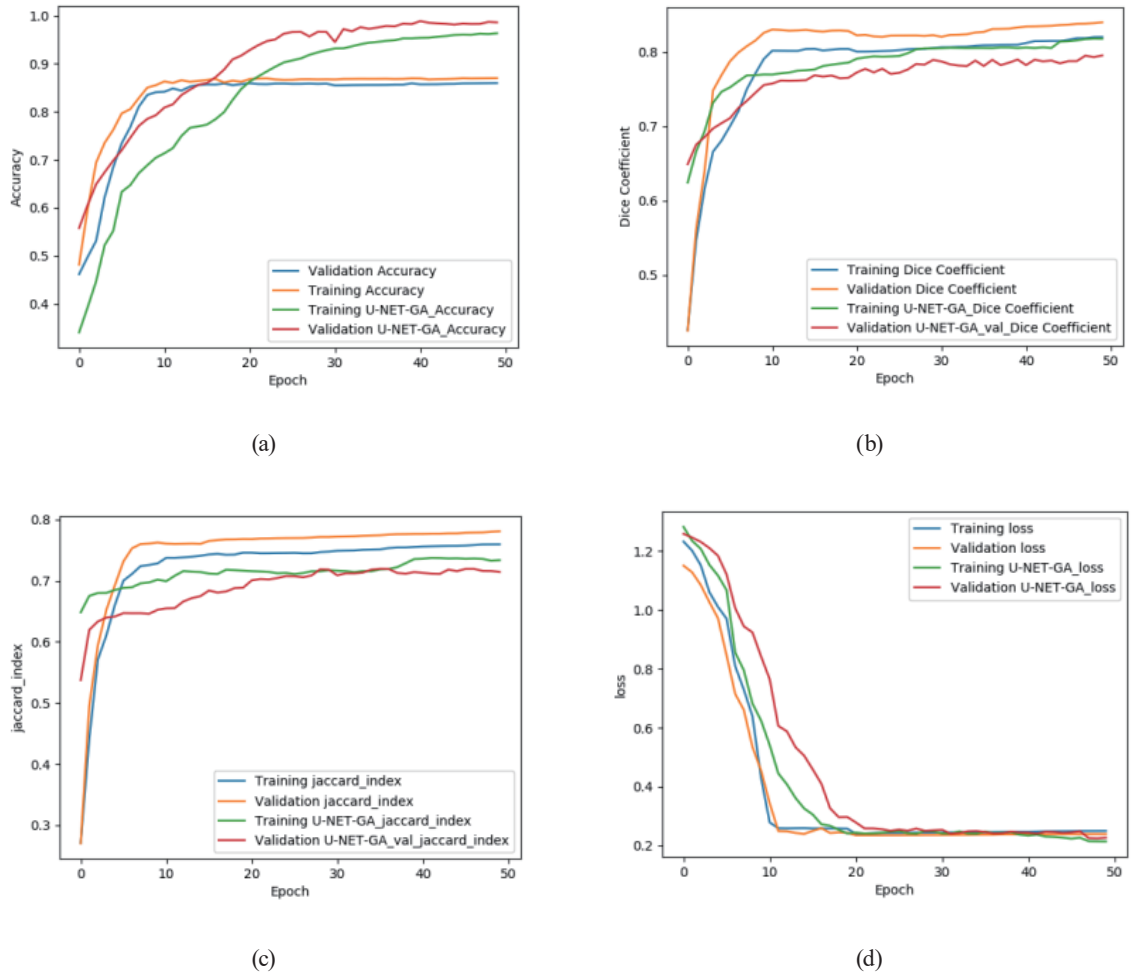

Fig. 8. (Color online) Experimental results of CT images of COVID-19 cases on U-Net-GA and U-Net. (a) Accuracy and epoch. (b) Dice coefficient and epoch. (c) Jaccard index and epoch. (d) Loss function and epoch.

Table 5
Changes in the number of U-Net Neurons–CT image data of COVID-19 cases.

Model	Epoch	*ACC* (%)	*Dice* (%)	*Jaccard* (%)	*Loss*
FCN32[51]	50	87.68	82.50	84.45	0.31
VanillaUnet[52]	50	89.62	86.54	78.36	0.30
U-Net	50	95.04	83.86	83.05	0.25
U-Net-GA	50	95.17	87.91	84.74	0.23

Table 6
Changes in the number of U-Net neurons–CT image data of COVID-19 cases.

Population algebra	Value
1	16, 16, 32, 256
2	16, 32, 128
3	16, 16, 32, 64, 256, 256
4	64, 128
5	16, 64, 128, 256
Best result	16, 64, 128, 256

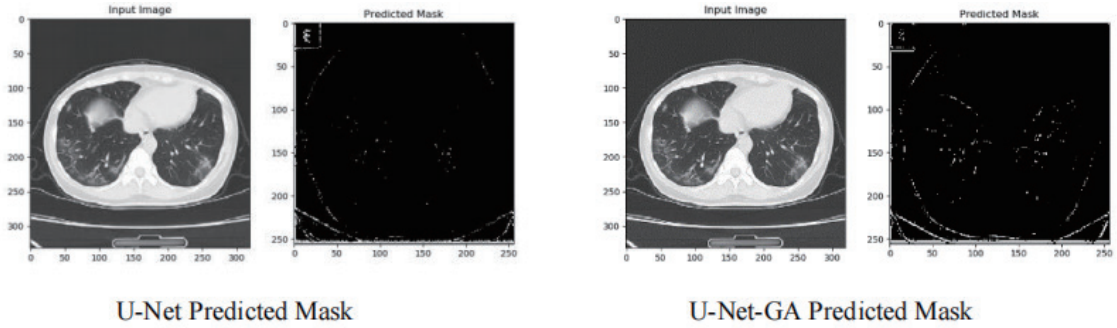

Fig. 9. Predicted Mask comparison.

Figure 9 clearly shows that the segmentation accuracy of the mask image predicted with the U-Net-GA trained model is higher than that predicted with the benchmark model in the prediction process.

5.3 Discussion of experimental results

In the experiment, we use a GA to optimize the architecture of the U-Net model and segmented the medical image using the GA-U-Net algorithm. We define the U-Net model, Evaluator class, Dice_coefficient, and Jaccard_index evaluation metrics, and implement these parts using the Keras library. We then use ImageDataGenerator to read training, validation, and test data from a folder and train a GA-U-Net model using ga_unet functions.

Finally, we evaluate the performance of the model and use the model to make predictions on the test data. In addition, note that our overall model is just an example, and there are many aspects that can be further optimized, such as adding more optimization algorithms and hyperparameter tuning strategies, using larger datasets and more complex models, and so on.

Note also that GAs are not one-size-fits-all optimization algorithms and may have limitations in practical applications. Therefore, when using GAs to optimize U-Net models, we need to consider multiple optimization algorithms and choose the algorithm that is most suitable for specific tasks and datasets. In addition, we also need to pay attention to the interpretability and generalizability of the model so that we can better understand and apply the model.

6. Conclusions and Future Work

The purpose of this study is to design an automatic search algorithm U-Net-GA for the U-Net network structure by using GAs. This algorithm can find the U-Net network structure to solve the image segmentation problem for users without U-Net expertise. A new coding strategy is designed using the GA for convolution, and parallel and cache components are designed to accelerate fitness assessment and save computing resources. Experiments were conducted on colorectal cancer images that have become popular in recent years and CT images of new coronary pneumonia cases. The experimental results show that U-Net-GA is superior to U-Net/

CN32/VanillaUnet in image segmentation, which achieves our original goal of allowing people without U-Net domain knowledge to apply U-Net networks to solve image processing problems of interest. In the future, we will strive to develop effective evolutionary algorithms and apply them to deep learning neural networks to advance the development of AI.

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), National Natural Science Foundation of China (#62262044), and Natural Science Foundation of Guangxi Province (#2023GXNSFAA026027). Yuzhong Peng and Xinyan Gan contributed equally to this work and should be both considered corresponding authors.

References

1 A. Parvaiz, M. A. Khalid, R. Zafar, H. Ameer, M. Ali, and M. M. Fraz: Eng. Appl. Artif. Intell. **122** (2023) 106. https://doi.org/10.1016/j.engappai.2023.106126
2 X. Tian, X. Wang, Y. Song, G. Wei, and R. Zeng: IEEE Trans. Aerosp. Electron. Syst. **59** (2023) 5646. https://doi.org/10.1109/TAES.2023.3262504
3 C. Zhu, J. Gao, M. Lu, Y. Zhang, Z. Wang, Q. Huang, Z. Wu, Y. Gao, Y. Wang, and W. Y. William: Opt. Express **31** (2023) 13028. https://doi.org/10.1364/OE.482241
4 A. Jain, M. Harper, M. Jayabalan, J. Mustafina, W. Khan, P. Liatsis, N. L. Ismail, and D. Al-Jumeily OBE: Int. Conf. Data Science and Emerging Technologies **165** (2022) 117. https://doi.org/10.1007/978-981-99-0741-0_9
5 K. Anuradha, E. Yuvasri, P. Mageshwaran, N. Divakar, and S. Kamalesh: J. Surv. Fish. Sci. **10** (2023) 1580. https://doi.org/10.17762/sfs.v10i4S.1295
6 Y. LeCun, Y. Bengio, and G. Hinton: Nature **521** (2015) 436. https://doi.org/10.1038/nature14539
7 A. Krizhevsky, I. Sutskever, and G. E. Hinton: Commun. ACM **60** (2017) 84. https://doi.org/10.1145/3065386
8 C. Clark and A. Storkey: Int. Conf. Machine Learning **37** (2015) 1766. http://proceedings.mlr.press/v37/clark15.pdf
9 I. Sutskever, O. Vinyals, and Q. V. Le: Adv. Neural Inf. Process. Syst. **27** (2014). https://doi.org/10.48550/arXiv.1409.3215
10 T. N. Sainath, B. Kingsbury, A.-r. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (2013) 315. https://doi.org/10.1109/ASRU.2013.6707749
11 O. Ronneberger, P. Fischer, and T. Brox: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th Int. Conf., Munich, Germany, October 5–9, 2015, Proc. Part III **18** (2015) 234. https://doi.org/10.1007/978-3-319-24574-4_28
12 N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni: IEEE Access **9** (2021) 82031. https://doi.org/10.1109/ACCESS.2021.3086020
13 L. Li, M. Verma, Y. Nakashima, H. Nagahara, and R. Kawasaki: Proc. IEEE/CVF Winter Conf. Applications of Computer Vision (2020) 3656. https://openaccess.thecvf.com/content_WACV_2020/papers/Li_IterNet_Retinal_Image_Segmentation_Utilizing_Structural_Redundancy_in_Vessel_Networks_WACV_2020_paper.pdf
14 D. Li, D. A. Dharmawan, B. P. Ng, and S. Rahardja: 2019 IEEE Int. Conf. Image Processing (ICIP) (2019) 1425. https://doi.org/10.1109/ICIP.2019.8803101
15 F. Jia, W. H. Wong, and T. Zeng: Proc. IEEE/CVF Int. Conf. Computer Vision (2021) 354. https://openaccess.thecvf.com/content/ICCV2021W/NeurArch/papers/Jia_DDUNet_Dense_Dense_U-Net_With_Applications_in_Image_Denoising_ICCVW_2021_paper.pdf
16 H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu: arXiv preprint arXiv:1711.00436 (2017). https://doi.org/10.48550/arXiv.1711.00436
17 H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang: Proc. AAAI Conf. Artificial Intelligence **32** (2018). https://doi.org/10.1609/aaai.v32i1.11709

18 Z. Zhong, J. Yan, W. Wu, J. Shao, and C.-L. Liu: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2018) 2423.https://openaccess.thecvf.com/content_cvpr_2018/papers/Zhong_Practical_Block-Wise_Neural_CVPR_2018_paper.pdf
19 B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2018) 8697. https://openaccess.thecvf.com/content_cvpr_2018/papers/Zoph_Learning_Transferable_Architectures_CVPR_2018_paper.pdf
20 F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, and S. Wirkert: arXiv preprint arXiv:1809.10486 (2018). https://doi.org/10.48550/arXiv.1809.10486
21 K. C. Wong and M. Moradi: Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proc., Part III **22** (2019) 393. https://doi.org/10.1007/978-3-030-32248-9_44
22 Y. Sun, B. Xue, M. Zhang, and G. G. Yen: IEEE Trans. Evol. Comput. **24** (2019) 394. https://doi.org/10.1109/TEVC.2019.2916183
23 R. S. Sutton and A. G. Barto: IEEE Cambridge, MA **22447** (2018). https://books.google.co.jp/books?hl=zh-CN&lr=&id=uWV0DwAAQBAJ&oi=fnd&pg=PR7&dq=R.+S.+Sutton&ots=mjnEt7Z4n3&sig=AkAUAhtCaZZJwyCdvkum9bmecLQ&redir_esc=y#v=onepage&q=R.%20S.%20Sutton&f=false
24 J. Galletly: Kybernetes **27** (1998) 979. https://doi.org/10.1108/k.1998.27.8.979.4
25 D. J. Montana and L. Davis: IJCAI **89** (1989) 762. https://www.ijcai.org/Proceedings/89-1/Papers/122.pdf
26 J. H. Holland: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence (MIT press, 1992). https://www.jstor.org/stable/24939139
27 H.-L. Liu, F. Gu, Y.-M. Cheung, S. Xie, and J. Zhang: IEEE Comput. Intell. Mag. **9** (2014) 44. https://doi.org/10.1109/MCI.2013.2291690
28 K. Nag and N. R. Pal: IEEE Trans. Cybern. **46** (2015) 499. https://doi.org/10.1109/TCYB.2015.2404806
29 Y. Fan, T. Jiang, and D. J. Evans: IEEE Trans. Med. Imaging **21** (2002) 904. https://doi.org/10.1109/TMI.2002.803126
30 G. K. Matsopoulos, P. A. Asvestas, N. A. Mouravliansky, and K. K. Delibasis: IEEE Trans. Med. Imaging **23** (2004) 1557. https://doi.org/10.1109/TMI.2004.836547
31 G. Quellec, M. Lamard, P. M. Josselin, G. Cazuguel, B. Cochener, and C. Roux: IEEE Trans. Med. Imaging **27** (2008) 1230. https://doi.org/10.1109/TMI.2008.920619
32 J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht: Proc. First Int. Conf. Genetic Algorithms and their Applications (2014) 160. https://bmcpublichealth.biomedcentral.com/articles/10.1186/1471-2458-14-1144
33 D. J. Stein and C. Bouwer: J. Anxiety Disord. **11** (1997) 409. https://doi.org/10.1162/106365602320169811
34 K. O. Stanley and R. Miikkulainen: Evol. Comput. **10** (2002) 99. https://doi.org/10.1016/S0887-6185(97)00019-4
35 P. Liashchynskyi and P. Liashchynskyi: arXiv preprint arXiv:1912.06059 (2019). https://doi.org/10.48550/arXiv.1912.06059
36 B. L. Miller and D. E. Goldberg: Complex Syst. **9** (1995) 193. https://wpmedia.wolfram.com/uploads/sites/13/2018/02/09-3-2.pdf
37 A. Lipowski and D. Lipowska: Physica A: Stat. Mech. Appl. **391** (2012) 2193. https://doi.org/10.1016/j.physa.2011.12.004
38 G. J. Romanes: Darwin and after Darwin: The Darwinian Theory. 1892 (Open court publishing Company, 1892) Vol. 1. https://books.google.co.jp/books?hl=zh-CN&lr=&id=WmEta3hDhKoC&oi=fnd&pg=PA1&dq=G.+J.+Romanes+1892&ots=y-4eRiIGV_&sig=2sB--1nMaq_hgFqbwGrtf4wY08E&redir_esc=y#v=onepage&q=G.%20J.%20Romanes%201892&f=false
39 J. M. Smith: The Theory of Evolution (Cambridge University Press, 1993). https://books.google.co.jp/books?hl=zh-CN&lr=&id=aU3XUTt9WXgC&oi=fnd&pg=PR9&dq=J.+M.+Smith+1993&ots=YxD4GGJwQE&sig=OsZK1_Jbn4OoRpLDbVZlj5FQS8E&redir_esc=y#v=onepage&q=J.%20M.%20Smith%201993&f=false
40 X. Yao: Proc. IEEE **87** (1999) 1423. https://doi.org/10.1109/5.784219
41 L. Bottou: Neural Networks: Tricks of the Trade: Second Edition (2012) 421. https://doi.org/10.1007/978-3-642-35289-8_25
42 A. B. Hamida, M. Devanne, J. Weber, C. Truntzer, V. Derangère, F. Ghiringhelli, G. Forestier, and C. Wemmert: Comput. Biol. Med. **136** (2021) 104730. https://doi.org/10.1016/j.compbiomed.2021.104730
43 M. Yildirim and A. Cinar: Int. J. Imaging Syst. Technol. **32** (2022) 155. https://doi.org/10.1002/ima.22623
44 Y. Cao, Z. Xu, J. Feng, C. Jin, X. Han, H. Wu, and H. Shi: Radiology: Cardiothoracic Imaging **2** (2020) e200082. https://doi.org/10.1148/ryct.2020200082

45 L. Huang, R. Han, T. Ai, P. Yu, H. Kang, Q. Tao, and L. Xia: Radiology: Cardiothoracic Imaging **2** (2020) e200075. https://doi.org/10.1148/ryct.2020200075
46 F. Shan, Y. Gao, J. Wang, W. Shi, N. Shi, M. Han, Z. Xue, D. Shen, and Y. Shi: Med. Phys. **48** (2021) 1633. https://doi.org/10.1002/mp.14609
47 S. Chaganti, P. Grenier, A. Balachandran, G. Chabin, S. Cohen, T. Flohr, B. Georgescu, S. Grbic, S. Liu, and F. Mellot: Radiology: Artif. Intell. **2** (2020) e200048. https://doi.org/10.1148/ryai.2020200048
48 Q. Yan, B. Wang, D. Gong, C. Luo, W. Zhao, J. Shen, Q. Shi, S. Jin, L. Zhang, and Z. You: arXiv preprint arXiv:2004.10987 (2020). https://doi.org/10.48550/arXiv.2004.10987
49 D.-P. Fan, T. Zhou, G.-P. Ji, Y. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao: IEEE Trans. Med. Imaging **39** (2020) 2626. https://doi.org/10.1109/TMI.2020.2996645
50 Y.-H. Wu, S.-H. Gao, J. Mei, J. Xu, D.-P. Fan, R.-G. Zhang, and M.-M. Cheng: IEEE Trans. Image Proc. **30** (2021) 3113. https://doi.org/10.1109/TIP.2021.3058783
51 J Long, E Shelhamer, T Darrell. Proceedings of the IEEE Conf. Comput. Vision Pattern Recognit. (2015) 3431–3440. https://openaccess.thecvf.com/content_cvpr_2015/papers/Long_Fully_Convolutional_Networks_2015_CVPR_paper.pdf
52 N. Wang, S. Lin, X. Li, K. Li, Y. Shen, Y. Gao, and L. Ma: IEEE Trans. Med. Imaging **42** (2023) 2740. https://doi.org/10.1109/TMI.2023.3264433

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)

Ningwei Mo is an undergraduate student in the Department of Medical Information, Guangxi University of Traditional Chinese Medicine. She is engaged in the study of medical image segmentation with her supervisor Daoqing Gong. (mnwemail@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 information management and information system design and development. (ganyan_88@163.com)

Yuzhong Peng received his B.S. degree in computer science and technology from Guangxi Teachers Education University, Nanning, China, in 2003, his M.S. degree in computer application technology from Guangxi Teachers Education University, Nanning, China, in 2009, and his Ph.D. degree in computer software and theory from Fudan University, Shanghai, China, in 2016. From 2009 to 2018, he was a teacher at Guangxi Teachers Education University and is currently a professor at Nanning Normal University. He is a CCF member and an ACM member. He is the author of a book and more than 30 articles, and holds 3 patents. His research interests include machine learning, data modeling/mining, and biological/medical informatics. (jedison@163.com)

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

Jiayuan Pan is an undergraduate student in the Department of Medical Information, Guangxi University of Traditional Chinese Medicine. He is engaged in the study of medical image segmentation with his supervisor Daoqing Gong. (m13976499048@163.com)