Automatically Designing U-Nets Using A Genetic Algorithm for Tree Image Segmentation

Binke Xu[1], Ying Bi[1], Bing Xue[1], Jan Schindler[2], Brent Martin[2], and Mengjie Zhang[1]
[1]*School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand*
[2]*Manaaki Whenua - Landcare Research, Informatics, Wellington, New Zealand*
Email: xubink@myvuw.ac.nz; ying.bi@ecs.vuw.ac.nz; bing.xue@ecs.vuw.ac.nz; SchindlerJ@landcareresearch.co.nz; MartinB@landcareresearch.co.nz; mengjie.zhang@ecs.vuw.ac.nz

***Abstract*—Tree image segmentation is an important task of analysing the forest cover in ecosystems. Convolutional neural networks (CNNs) particularly U-Nets are popular methods for image segmentation, but they often require rich expertise from both the problem domain and the neural network domain to design effective model architectures. Recently, automatically designing U-Nets has shown promise, but none of them has been proposed for tree image segmentation. When designing U-Nets, the encoding and decoding schemes are the keys to the success of the search, which have not been fully investigated yet. Therefore, this paper investigates a new approach based on genetic algorithms (GAs) to automatically design/learn U-Nets for tree segmentation from remote-sensing images. The new encoding scheme enables chromosomes of a GA to represent potential U-Net models of variable length based on blocks. The decoding scheme decodes chromosomes to potential U-Net models with the characteristics of U-Nets. More importantly, the new designs allow the new approach to effectively and efficiently search for U-Net models of variable lengths by only optimising several key parameters. The results show that the proposed approach achieves better performance than a well-known manually designed U-Net by automatically learning the U-Net model with a significantly smaller number of parameters on one tree image segmentation dataset.**

***Index Terms*—Image segmentation; Evolving neural networks; Evolutionary deep learning; Genetic algorithm; Computer vision**

I. Introduction

As an important part of ecosystems, forests stabilize soils, provide suitable habitats for insects and potential economic value to humans. Managing a forest typically requires people to obtain the statistics and distributions of tree species, statistics of forest coverage, etc. However, it is very difficult to manually collect these data over large areas and in complex terrain. Therefore, it is necessary to develop a more efficient and safe method to automatically collect forest data. A promising approach is to analyse remote sensing images, which can be obtained using satellites and aircrafts. Image segmentation could help analyse the coverage of trees in the forests by segmenting the trees from other objects. Existing work has shown the promise of image segmentation in analysing forest data by achieving good results [1, 2, 3]. However, image segmentation is a challenging task due to many factors, including high variations across images, complex background, and a limited number of training images.

Different types of methods have been developed for image segmentation. Traditional methods include threshold-based methods, region-based methods, edge-detection-based methods, and classification-based methods [4]. These methods typically require rich domain expertise and human intervention, thus their flexibility and performance are often limited. In recent decades, artificial neural networks (ANNs), particularly convolutional neural networks (CNNs), have become a popular approach to image segmentation [5]. CNNs typically consist of a large number of layers including convolutional layers and pooling layers to automatically extract informative features from images and perform classification. Due to the powerful ability of feature/representation learning, CNN-based methods have been successfully applied for many computer vision tasks, such as image classification, object detection, and image segmentation [6]. In the last decade, many CNN variants have achieved promising results in image segmentation [5]. Well-known examples include fully convolutional network (FCN), U-Net, SegNet, stacked deconvolutional network (SDN), feature pyramid network (FPN), regional CNN (R-CNN), mask R-CNN, and path aggregation network (PANet) [5].

Among these different CNN variants, U-Net [7] is one of the most popular variants for image segmentation, particularly in the case where the training data is small. Unlike traditional CNNs, the architecture of U-Net consists of a contracting path and an expansive path, which are connected by a bottleneck block. The shape of U-Net is similar to the letter "U". A number of convolutional layers and pooling layers are used in the contracting and expansive paths. Several variants of U-Net have been developed for different types of image segmentation tasks, such as medical image segmentation [8], water segmentation [9], biomedical volumetric image segmentation [10], and nuclei segmentation [11].

However, most existing U-Net methods require rich expertise from both the problem domain and the ANN domain to design an effective network architecture to solve a specific task [12, 13]. In addition, very few of these methods have been

2022 IEEE Symposium Series on Computational Intelligence (SSCI) | 978-1-6654-8768-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SSCI51031.2022.10022182

developed for tree segmentation from remote-sensing images [14]. It is possible to use existing U-Net methods for tree segmentation, but their network architectures may not be the most effective and efficient ones. This will lead to a waste of training time and computing resources for such a real-world application. In addition, existing manually designed U-Net models often contain a huge number of parameters, which make the model complex and hard to explain. Therefore, it is desirable to develop an intelligent method to automatically search for promising U-Net models with a smaller number of parameters for tree image segmentation.

However, it is a challenging task to automatically design the architectures of ANNs due to the large and complex search space. Existing works have explored different neural architecture search (NAS) methods, including reinforcement learning-based methods, gradient-based methods, and evolutionary-based methods [15]. Evolutionary-based NAS methods typically employ evolutionary computation (EC) methods with a population-based search mechanism to search for optimal NN architectures by optimising an objective/fitness function. Compared with the other two types of methods, evolutionary-based NAS methods have good global search ability and do not require functions/objectives to be differentiable. Therefore, evolutionary-based NAS methods have been successfully applied for different tasks, including image classification, natural language processing, and object detection [15]. However, very few evolutionary-based NAS methods have been explored for real-world image segmentation. Unlike other computer vision and image analysis tasks, solving image segmentation often uses a different CNN architecture such as U-Net. Therefore, it is necessary to develop a new EC-based NAS method for tree image segmentation.

Genetic algorithms (GAs) [16] are an EC-based method that has been widely applied for optimisation problems including NAS. Specifically, GAs use a string-based encoding to represent a solution and search for (near)-optimal solutions based on natural selection and genetic operators. GAs have been used to automatically search for different ANN architectures, such as autoencoders, CNNs and long short-term memory (LSTM) [15]. In recent years, GA-based methods have also been developed to evolve U-Nets and achieved better performance than manually designed ANNs [12, 13, 17]. But these methods are for medical image segmentation, not tree image segmentation. In addition, the solution encoding, search mechanism and network evaluation have not been fully explored in GA-based NAS for image segmentation. Motivated by these, this paper investigates a new GA-based method to evolve U-Nets for tree segmentation from remote-sensing images.

The overall goal of this paper is to develop a new GA-based approach to automatically evolving U-Nets for tree segmentation from remote-sensing images. The proposed approach is termed as GA-Unet-B, denoting automatically designing U-Nets based on Block. To achieve this goal, new encoding and decoding schemes, new crossover operator, and a fitness measure are developed in the new approach to effectively and efficiently find optimal U-Net models for tree image segmentation. Specifically, the new designs allows the proposed approach to evolve U-Net modes of variable-length based on blocks and to be more efficient by only optimising several key parameters. The proposed GA-Unet-B approach will be examined on a real-world benchmark dataset of tree image segmentation. Empirical analysis will be conducted by comparing the best U-Net model evolved by GA-Unet-B with an existing manually designed U-Net. Further analysis of the segmentation results and the evolved models will provide more insights into the new approach.

II. Background and Relate Work

This section provides basic concepts of image segmentation. The related works on evolving architectures of ANNs are then reviewed and discussed. The limitations are summarised, which shows the motivations of our work.

A. *Image Segmentation*

Image segmentation is an important task in computer vision and image processing. The task aims to segment/group images into small homogeneous regions. A typical example is to segment a sea scene image into the water and non-water parts [9]. There are two types of image segmentation tasks, i.e., semantic segmentation and instance segmentation [5]. Semantic segmentation groups pixels of the same objects into the same category, i.e., the objects with the same semantic meaning are in the same category, while instance segmentation groups pixels of each object into one category no matter whether it has the same semantic meaning to the others. Figure 2 shows examples of semantic segmentation and instance segmentation.

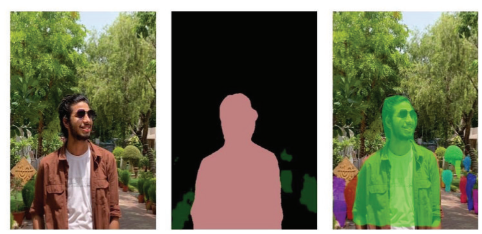

Fig. 1: Semantic segmentation and instance selection [18].

To address image segmentation, different algorithms have been proposed. A detailed review of traditional methods can be found in [4, 19]. Deep learning methods, including deep ANNs, have been widely applied for image segmentation. Minaee *et al* [5] provide a comprehensive review of existing ANNs for image segmentation, including FCN, U-Net, SegNet, SDN, FPN, R-CNN, mask R-CNN, and PANet.

FCN [20] is a milestone of image segmentation algorithms that uses convolutional layers to extract features and generate masks of the same size as the input images. U-Net [7] is based on FCN for medical image segmentation. The key difference is that the structure of U-Net is symmetric, i.e., both the contracting path and the expansive path have the same number of blocks, and the block in the contracting path will use a transpose layer to connect and copy the feature to the corresponding block in the expansive path. Figure 2 shows an example of U-Net.

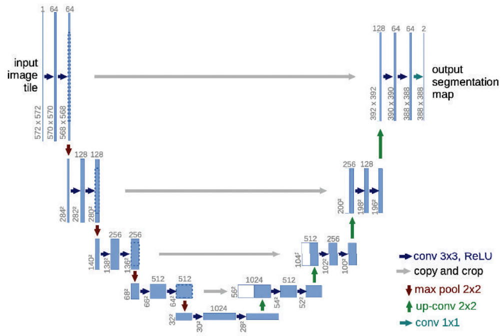

Fig. 2: U-Net architecture[7].

U-Net and its variants have been applied for many medical image segmentation tasks. In [10], 3D U-Net was developed for volumetric segmentation tasks. In [21], U-Net has been used for pulmonary nodules segmentation and achieved 0.736 dice coefficient. In [22], U-Net achieved 0.74 dice coefficient on rectal tumour segmentation and 3D U-Net achieved 0.863 intersections over union (IoU) on Xenopus kidney segmentation. In general, U-Net is a promising approach achieving great results on medical image segmentation, particularly when the training data is limited. However, U-Net has not frequently applied for other image segmentation tasks of small training data, such as tree segmentation.

B. Related Work

GAs have been widely applied for evolving architectures of CNNs for image classification. Sun *et al.* [23] proposed a GA-based method to automatically design CNNs for image classification based on the idea of skip connection. Specifically, a skip layer is used to replace the original convolutional layer and the fully connected layers to facilitate the search. This method achieved better performance and evolved models with fewer parameters over 18 effective CNNs, including manually and automatically designed CNNs. In [24], a GA-based method (i.e. EvoCNN) with a crossover operator based on unit alignment was proposed to automatically evolve variable-length CNNs for image classification. This method achieved better performance than a large number of benchmark methods on eight image classification datasets. In [25], the EvoCNN method was further extended by developing different mutation strategies and encoding the activation functions for each layer. This method achieved 93.76% accuracy on the Fashion MNIST dataset. More related works on using GAs or other EC methods to evolve ANNs can be found in [15].

Very few works on using GAs to evolve CNNs particularly U-Nets for image segmentation were proposed. Hassanzadeh *et al.* [12] applied GA to evolve U-Nets for medical image segmentation based on Dense and Residual blocks. This method achieved better results than several manually or automatically designed U-Nets on six image datasets. Hassanzadeh *et al.* [13] proposed the EvoU-Net method for medical image segmentation. This method evolved small networks based on U-Net and achieved better performance than existing U-Net and its variants. In [17], the Genetic U-Net method was proposed for retinal vessel segmentation. In this method, GA was applied to design U-Net based on several building blocks including DenseNet and ResNet blocks. This method achieved better performance than many manually designed U-Nets on two datasets of retinal vessel segmentation.

To sum up, existing works showed that it is promising to automatically evolve/design NNs for image-related tasks using GAs. However, most of existing works focus on solving image classification or medical image segmentation tasks. None of them has been extended to solve other real-world problems such as tree segmentation from remote-sensing images. Due to the differences of the images and tasks, it is necessary to develop a new approach to automatically design ANNs or U-Nets for a specific task. Therefore, to address this, this paper will develop a new GA-based approach to automatically evolving U-Nets for tree image segmentation.

III. The Proposed Approach

This section introduces the proposed GA-Unet-B approach, including the encoding and decoding schemes, the fitness function, and the genetic operators.

A. Overall Algorithm

To perform tree image segmentation, the GA-Unet-B approach is developed to automatically evolve U-Nets in this paper. U-Net has a block-based architecture with a contracting path and an expansive path. To enable a GA to search for effective U-Net architectures for tree image segmentation, a new encoding and decoding scheme based on blocks, a new fitness function, and a new crossover operator are developed in the proposed GA-Unet-B approach. The new designs allow GA-Unet-B to evolve variable-length U-Nets based on blocks for tree image segmentation.

The overall algorithm of GA-Unet-B is described in Algorithm 1. The inputs of the system are two sets of images and corresponding masks (labels), i.e., the original training set is split into two subsets, i.e., D_{Evobp} for model training and D_{Evofit} for model testing during the evolutionary process. An initial population of chromosomes are generated according to the encoding scheme. Each chromosome is then decoded to a U-Net model, which is trained based on back-propagation and a gradient descent algorithm on D_{Evobp}. The fitness value (goodness) of the chromosome is calculated on D_{Evofit} using the fitness function. At each iteration/generation, a new population is generated based on a selection method and genetic operators. The evolutionary process terminates when a predefined termination criterion, i.e., reaching the maximal number of generations, is satisfied. After the evolutionary process, the U-Net model represented by the best chromosome is returned and applied to unseen data (i.e. the test set).

Algorithm 1: GA-Unet-B

Input : D_{Evobp}: images and masks for model training; D_{Evofit}: image and masks for model evaluation (fitness evaluation).
Output : $Unet^{best}$: the best U-Net model.

1 $P_0 \leftarrow$ randomly initialise a population of chromosomes based on the encoding scheme;
2 **for** *each p in P_0* **do**
3 Decode p to a U-Net model;
4 Train the U-Net model using D_{Evobp} and back-propagation;
5 Test the U-Net model on D_{Evofit};
6 Calculate the dice coefficients, the dice loss, and the number of parameters of U-Net and set them as the fitness values for p;
7 **end**
8 $Unet^{best} \leftarrow$ Update the best U-Net model;
9 $i = 1$;
10 **while** *Termination condition is not satisfied* **do**
11 $P_i \leftarrow$ Copy the best chromosome of P_0;
12 $S_i \leftarrow$ Select a set of potential chromosomes as parents;
13 $P_i \leftarrow$ Apply the new crossover operator to generate new chromosomes from parents;
14 $P_i \leftarrow$ Apply the mutation operator to generate new chromosomes from parents;
15 Repeat steps 2-7 to evaluate P_i;
16 $Unet^{best} \leftarrow$ Update the best U-Net model;
17 $i = i + 1$;
18 **end**
19 Return $Unet^{best}$ of P_{i-1} and apply it to unseen data.

B. Encoding and Decoding Schemes

U-Net is constructed by a contracting path, an expansive path, and a bottleneck block. The contracting path and the expansive path are typically comprised of a number of convolutional layers and pooling layers. Importantly, the contracting and expansive paths are symmetrical, e.g., the number of layers and types of layers are the same or highly related. In other words, if the key architectures of the contracting path is set, these parameters will be used to construct models for the expansive path. Based on this, GA can be used to search for only the architectural parameters for the contracting path and the bottleneck block of U-Net.

The architecture of U-Net [7] is a FCN-based architecture [20], which outputs images (masks) from the input images. The output images are often segmentation masks, where each pixel has a label. In other words, there are no fully connected layers in U-Net, which makes it different from existing CNN-based methods for image classification. The same as CNNs, U-Net is a build-up of multiple blocks and each block has several convolutional layers with an equal number of filters and a pooling layer. Therefore, in this paper, an encoding scheme based on blocks is proposed to encode a potential U-Net in GA. Specifically, the proposed GA-Unet-B approach only needs to optimise a small set of parameters, i.e., numbers of blocks, numbers of the convolutional layer in a block, numbers of filters in a convolutional layer, the filter size, pooling type, dropout rate, and batch normalization.

In GA-Unet-B, a chromosome is used to represent a potential U-Net for tree representation based on blocks. Specifically, each chromosome is represented by a list and each element in the list represents a block. A block is represented by several bits/elements, representing the number of convolutional

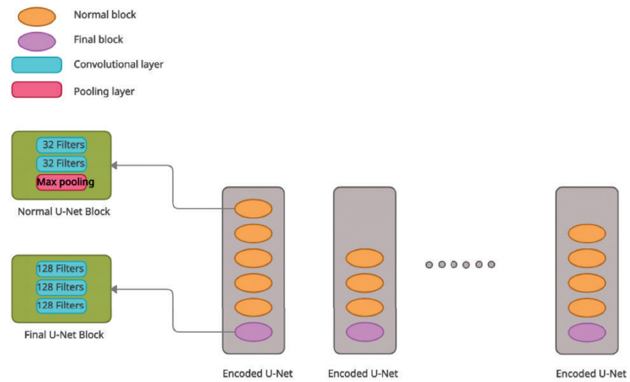

Fig. 3: Illustration of chromosome encoding.

#convolutional layers	#filters	Filter size	Pooling type	Dropout rate	Batch normalisation
2	32	3	0	0.4	1

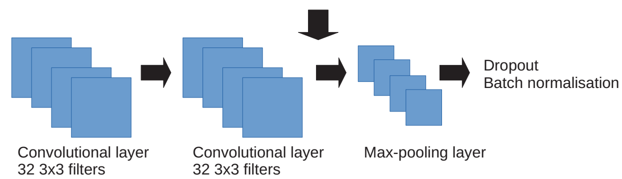

Fig. 4: Encoded block.

layers, the number of filters, the size of filters, the type of pooling, the dropout rate, and the type of batch normalisation. Figure 3 shows the encoding of variable-length chromosomes, representing U-Net comprised of multiple blocks. It is worth mentioning that the encoding allows for generating variable-length U-Net models, which is more efficient and effective for search. An example of the encoding of each block is shown in Fig. 4. By using the example encoding, a block consists of three convolutional layers, a pooling layer, and a batch normalisation layer. The number of filters is 32 and the size of filters is 3×3 in the convolutional layers. The pooling type is max-pooling, the rate of dropout is 0.4, and the batch normalisation layer is active.

Encoded parameters and corresponding value ranges: In GA-Unet-B, parameters for constructing a U-Net model are summarised. The parameters with value ranges are searched by using the proposed GA-Unet-B approach, while the values of the other parameters are commonly used settings. The parameters and their value ranges are described as follows:

- Numbers of blocks:
  - 3, 4, 5, 6;
- Numbers of convolutional layers in a block:
  - 2, 3, 4, 5;
- Convolutional layers:
  - Stride : 2×2;
  - Numbers of filters: 8, 16, 32, 64, 128;
  - Size of filter: 3×3, 5×5, 7×7;
  - Active function: ReLU;
- Pooling layer:

- 0: max-pooling; 1: average-pooling;

- Dropout rate:
  - 0, 0.1, 0.2, 0.3, 0.4 , 0.5;
- Batch normalisation layer:
  - 1: Active; 0: Inactive.

At the population initialisation process, a chromosome is randomly generated to represent U-Nets according to the above value ranges. During the fitness evaluation process and the testing process, a chromosome is decoded to a feasible U-net model for image segmentation.

Decoding and network construction: In this process, a U-Net model is constructed from a chromosome. Specifically, a chromosome represents the parameters for the contracting path and the bottleneck path. For the bottleneck path, the number of blocks is 1 and each block only contain convolutional layers. For the contracting path, the number of blocks is determined by the length of the list in the chromosome and the settings of each block are based on to the value of each item/element in the corresponding list. For the expansive path, the same settings as the contracting path are used. In addition, the output feature maps of each block in the contracting path are also fed into the corresponding block of the expansive path, which is the main characteristics of U-Nets. With the contracting path, the bottleneck block, and the expansive path, a U-net mode is constructed and used for image segmentation. The input of U-Net is a colour image with RGB channels and the output of U-Net is a segmentation mask with scores or labels. Therefore, a 1×1 convolutional layer is added to the final layer of the expansive path and a Sigmoid activation function is used for the convolutional layer to generate the segmentation scores in the range of [0, 1]. The scores are transformed into labels (i.e., 0 or 1) according to a 0.5 threshold value.

C. Network Evaluation and Fitness Function

The performance (goodness) of the constructed U-Net model (i.e., the chromosome) is evaluated using fitness/objective functions. The performance of U-Net can be used to show the goodness of the corresponding chromosome. Since the task is tree image segmentation, the commonly used measure for this task is used as the objective function. In GA-Unet-B, two different measures are used to evaluate and select chromosome, i.e., the dice loss [26] and the number of parameters of the network. The number of parameters indicates the complexity of the U-Net model, which is expected to be minimised. When selecting the chromosome, the one with the fewer dice loss will be chosen, if the loss is equal, then select the one with fewer parameters.

Dice coefficient: The dice loss is based on the dice coefficient (DC), which evaluates the similarity of two sets. In image segmentation, this measure typically evaluates the similarity of the real segmentation mask (i.e., the ground truth) and the predicted segmentation mask.

The DC is defined as

$$DC = \frac{2(A \cap B)}{|A| + |B|} \tag{1}$$

where A represents the ground truth and B represents the predicted mask. $A \cap B$ denotes the true positives, which means the number of correctly predicted pixels in the image. $|A|$ or $|B|$ denotes the total number of pixels in the image.

Dice loss: Based on the dice coefficient, the dice loss is defined as

$$DL = 1 - DC \tag{2}$$

The dice loss is expected to be minimised during the training and evolutionary process. During the fitness evaluation, the U-Net model is trained using gradient descent via back-propagation on D_{Evobp}. In this process, the dice loss is used as the loss function for network training. After the training process, the dice loss is calculated by applying the trained U-Net model on D_{Evofit}. Since training the U-Net model is very time-consuming and computationally expensive, early stopping is used, i.e., the training will stop if the loss is not improved/decreased in five epochs. Based on the dice loss and the number of parameters, the selection method can be used to select promising chromosomes for population updating, which will be introduced in the next subsection.

D. Population Generation/Updating and Genetic Operators

In this process, promising chromosomes are selected as parents and genetic operators are applied to generate new chromosomes to replace the current population. First, the best chromosome will be directly copied to the new generation. Second, a selection method is applied to select a set of chromosomes as parents to generate new chromosomes. The selection method is tournament selection since it is easy to compare multiple chromosomes when different criteria are used. In each selection process, the dice loss values of randomly selected chromosomes are compared and the best chromosome with the smallest dice loss is selected as a parent. If there are multiple best chromosomes, the second measure, i.e., the number of parameters of the network, is used to select the chromosome representing the smallest network. The selection method allows to select better-performing chromosomes representing less complex U-Net models.

Third, the crossover operator is applied to each paired chromosomes/parents to generate two new ones by swapping some blocks/bits of the two parents, which is shown in Fig. 5. Specifically, it aligns the same type of element in two chromosomes and swaps the randomly selected elements to generate new chromosomes/offspring. Fourth, the mutation operator is applied to each chromosome by replacing some bits/elements with randomly generated ones, which can improve the diversity of the population.

IV. Experiment Design

This section describes the experiment design, including the dataset, the comparison methods, parameter settings, and the implementation.

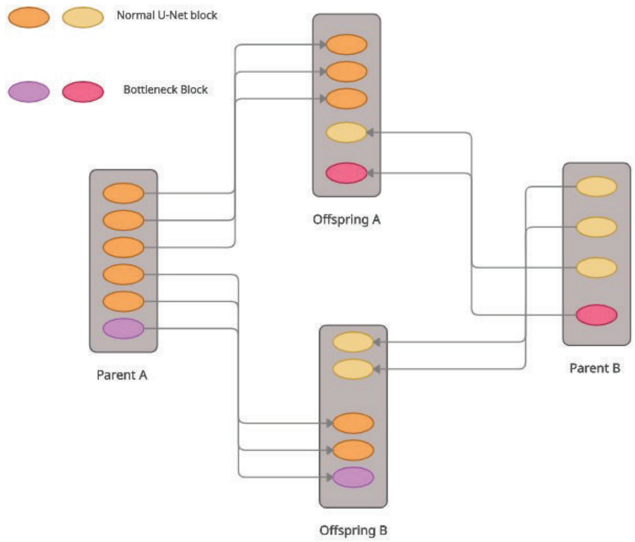

Fig. 5: Illustration of the crossover operator.

A. Dataset and Preprocessing

This paper focuses on solving tree segmentation from remote-sensing images. The dataset used in the experiment contains 1671 256×256 RGB colour images along with corresponding, hand-annotated, pixel-wise labels of tree canopy/other land cover. The images are aerial photographs with 10 cm pixel resolution, containing visual contents of the city, trees, forest, houses, roads, etc. The tree segmentation task aims to segment the regions containing tree crowns from the other content. Example images of the dataset and the corresponding masks are shown in Fig. 6. In the mask, the pixels in white represent the regions with trees, while the pixels in black represent the other regions.

In the experiments, the whole dataset is split into the training set and the test set, i.e., 80% images are used for training and the remaining 20% images are for testing. The training set is used in the evolutionary process and the final model training process. Specifically, 90% images of the training set (i.e. D_{Evobp}) are used to train the model decoded by a chromosome and the remaining 10% images (i.e. D_{Evofit}) are used to calculate the fitness value in the fitness evaluation process. After the evolutionary process, the full training set is used to train the best model, which is applied to the test set. The results of the test set are then reported.

B. Benchmark Method

The proposed GA-Unet-B approach is compared with a manually designed U-Net to show its effectiveness. The U-Net method refers to [7], which is a famous manually designed U-Net for medical image segmentation with limited training data. In this paper, it is termed Standard U-Net. The parameter settings for Standard U-Net are kept the same as [7] to maintain its superiority. Note that this paper solves a new problem/application and there is no existing method for this task. Therefore, only the Standard U-Net is used for comparisons at the current stage, which is sufficient to show that

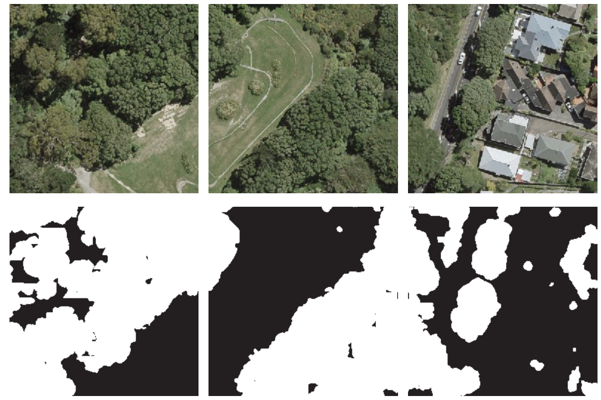

Fig. 6: Example images and the corresponding ground truth (masks) of the dataset.

the effectiveness of the automatically designed U-Nets. Also, the Standard U-Net will be trained on the tree segmentation dataset 10 time and use the mean dice coefficient to compare with the automatically designed U-Net. In our future study, we will investigate more different types of methods to solve this task.

C. Parameter Settings

The parameter settings for the proposed approach are summarised in Table I. Specifically, the size of the population is 10 and the maximal number of generations is 10. Since evolving U-Net is time-consuming and computationally extensive, a small population size and a small number of generations are used. Thus, the total number of network evaluations is 100. These may not be the optimal parameters for the proposed approach, but it is more affordable in real-world applications. In the network evaluation, the maximal number of epochs is 100 and the model will stop training if its performance is not changed in five consecutive epochs. In the network training, the batch size is 4 due to the small training set and the learning rate is 0.0001. Adam is used to optimise the parameters of U-Nets owing to its efficiency in tuning the learning rate for rapid convergence.

All the experiments have been implemented in Python 3.8, TensorFlow 2.4.1, and Keras 2.4.3. In addition, an NVIDIA RTX A5000 GPU with CUDA 11.3 was used to run the experiments and speed up the network training. All the methods have been executed 30 independent times using different random seeds. The best and average results are reported.

TABLE I: Parameter setting of GA-Unet-B

Parameter	Value
Population size	10
Number of generations	10
Epochs	100
Batch size	4
Early stopping	10 epochs
Optimizer	Adam
Learning rate	0.0001
Mutation rate	5%
Crossover rate	100%

V. Results and Discussions

This section compares the performance of the proposed GA-Unet-B approach and the Standard U-Net for tree image segmentation in terms of the segmentation results and the model complexity (i.e., the number of parameters). Then it analyses the segmentation masks obtained by GA-Unet-B and the best evolved U-Net model.

A. Comparison Between the Proposed GA-Unet-B Approach and the Standard U-Net

The segmentation results of GA-Unet-B and Standard U-Net are presented in Table II. The first block compares the dice coefficient values (%) of GA-Unet-B and Standard U-Net and the second block compares the number of parameters of the U-Net models. The results are from 10 independent runs with different random seeds. Table II shows that GA-Unet-B achieves the best dice coefficient of 86.13% and the mean dice coefficient of 85.56%, which are much higher than that by Standard U-Net achieving the best dice coefficient of 74.31%. The results show that the U-Nets evolved by the proposed GA-UNet-B method are more accurate than the manually designed U-Net for tree image segmentation. Comparing the number of parameters of the U-Net models evolved by GA-Unet-B with that of Standard U-Net, it is clear that the evolved U-Net models are less complex by using a significantly smaller number of parameters. As a result, the proposed approach can achieve better segmentation results and reduce the number of parameters of the U-Net model than Standard U-Net for tree segmentation from remote-sensing images.

TABLE II: Segmentation results of GA-Unet-B and Standard U-Net

Measure		GA-Unet-B	Standard U-Net
Dice coefficient (%)	Max	86.13	74.31
	Min	85.45	64.03
	Mean±Std.	85.56±0.21	71.40±3.39
Number of parameters	Max	7,837,473	–
	Min	164,737	–
	Mean	1,548,665	31,055,297

B. Analysis of the Best U-Net Model Evolved by GA-Unet-B

The parameters of the best U-Net model evolved by GA-Unet-B are summarised in Table III. Specifically, the best U-Net model consists of five blocks, four blocks for the contracting path and the expansive path, and one block for the bottleneck to connect these two paths. The first block in the contracting path includes five convolutional layers of 64 3×3 filters. The max-pooling layer is connected with the final convolutional layer in this block. In addition, this block uses a dropout rate of 0.1 and batch normalisation. In the other four blocks, different numbers and types of layers and dropout rates are evolved. Compared with Standard U-Net, this evolved U-Net is very different in the model architecture. For example, in Standard U-Net (i.e. Fig. 2), the bottleneck uses a large number of convolutional filters and a small number of layers, while the evolved U-Net model uses a larger number of layers and a smaller number of filters. This indicates the GA-Unet-B approach can evolve effective and efficient U-Net models that may not be designed by human/experts.

C. Visualisation of the Segmentation Results

For real-world applications, it is also necessary to analyse the segmentation results. Figure 7 shows the segmentation masks predicted by the U-Net model evolved by GA-Unet-B. It shows that the predicted masks captured most of the tree regions and sometimes did better than the original mask (ground truth). For example, Fig. 7 (b) the original mask marks the top right corner of the image that contains trees as background (black), the mask/prediction generated by the evolved U-Net model correctly marks this area as trees (white). In Fig. 7 (c), the original mask (ground truth) marks some of the top left corner of the image as background (black), but the predicted mask correctly marks this region as trees (white). This pattern can also be found in other predicted masks. However, from the segmentation results, it can also be found that the evolved U-Net model missed some of the details. For example, in Fig. 7 (a), the top left corner of the image has some shadows, which are not the tree regions, but the predicted mask marks those regions as tree regions. This shows there is still space to further improve the performance of tree image segmentation. Overall, the prediction made by the evolved U-Net model is impressive and the performance of the evolved U-Net model could be further improved with higher quality ground truth masks, i.e. finer delineation of tree canopies including shadow areas.

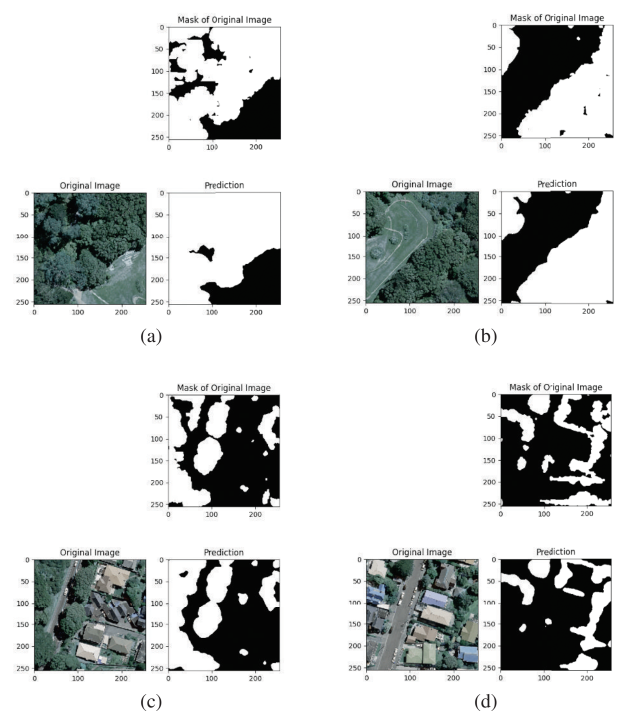

Fig. 7: The segmentation mask predicted by the U-Net model evolved by GA-Unet-B.

TABLE III: Parameters of the best U-Net model evolved by GA-Unet-B

#Block number	#Convolutional layer	Filter size	#Number of filters	Dropout rate	Batch normalisation	Pooling type
1	5	3	64	0.1	Active	Max-pooling
2	4	5	32	0.4	Inactive	Average-pooling
3	2	7	16	0.2	Inactive	Average-pooling
4	2	3	16	0.4	Active	Max-pooling
Bottleneck	5	5	8	0	Inactive	None

VI. Conclusions

This paper applied GA to automatically evolve U-Nets for tree segmentation from remote-sensing images. The proposed approach can automatically evolve variable-length U-Net models effectively by using the encoding and decoding schemes. The evolved U-Net model achieved much better performance than the compared U-Net model on the tree image segmentation dataset. More importantly, the U-Net model evolved by the proposed approach had a significantly smaller number of parameters than the compared U-Net model. This work showed the potential of using GAs to automatically design U-Nets for real-world image segmentation tasks.

This paper is a starting point of applying evolutionary-based neural architecture search methods for tree image segmentation. However, the performance of the proposed method can be further improved by designing more effective encoding strategies and efficient network evaluation strategies, and comparing its performance with other state-of-the-art methods including automatically designed U-Net models. In the future, we will address these limitations in order to develop an automatic, efficient and effective tree image segmentation approach.

Acknowledgements

This work was funded by New Zealand Ministry of Business, Innovation and Employment under contract C09X1923 (Catalyst: Strategic Fund). The Imagery was captured for the "Wellington City Council" by AAM NZ Limited and is available online at https://data.linz.govt.nz/layer/95524-wellington-010m-urban-aerial-photos-2017/.

References

[1] H. Huang, X. Li, and C. Chen, "Individual tree crown detection and delineation from very-high-resolution uav images based on bias field and marker-controlled watershed segmentation algorithms," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 7, pp. 2253–2262, 2018.

[2] L. Qiu, L. Jing, B. Hu, H. Li, and Y. Tang, "A new individual tree crown delineation method for high resolution multispectral imagery," *Remote Sensing*, vol. 12, no. 3, p. 585, 2020.

[3] J. T. Walton, D. J. Nowak, and E. J. Greenfield, "Assessing urban forest canopy cover using airborne or satellite imagery," *Arboriculture and Urban Forestry.*, 2008, DOI: 10.48044/jauf.2008.046.

[4] Y. Liang, M. Zhang, and W. N. Browne, "Image segmentation: a survey of methods based on evolutionary computation," in *Proceedings of Asia-pacific Conference on Simulated Evolution and Learning*. Springer, 2014, pp. 847–859.

[5] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[6] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015, pp. 234–241.

[8] C. Affonso, A. L. D. Rossi, F. H. A. Vieira, A. C. P. de Leon Ferreira *et al.*, "Deep learning for biological image classification," *Expert Systems with Applications*, vol. 85, pp. 114–122, 2017.

[9] A. J. McLeay, A. McGhie, D. Briscoe, Y. Bi, B. Xue, R. Vennell, and M. Zhang, "Deep convolutional neural networks with transfer learning for waterline detection in mussel farms," in *Proc. IEEE SSCI*, 2021, pp. 1–8.

[10] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Proceedings of International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2016, pp. 424–432.

[11] Z. Zeng, W. Xie, Y. Zhang, and Y. Lu, "Ric-unet: An improved neural network based on unet for nuclei segmentation in histology images," *IEEE Access*, vol. 7, pp. 21 420–21 428, 2019.

[12] T. Hassanzadeh, D. Essam, and R. Sarker, "An evolutionary DenseRes deep convolutional neural network for medical image segmentation," *IEEE Access*, vol. 8, pp. 212 298–212 314, 2020.

[13] ——, "EvoU-Net: An evolutionary deep fully convolutional neural network for medical image segmentation," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 181–189.

[14] J. A. C. Martins, K. Nogueira, L. P. Osco, F. D. G. Gomes, D. E. G. Furuya, W. N. Gonçalves, D. A. Sant'Ana, A. P. M. Ramos, V. Liesenberg, J. A. dos Santos *et al.*, "Semantic segmentation of tree-canopy in urban environment with pixel-wise deep learning," *Remote Sensing*, vol. 13, no. 16, p. 3054, 2021.

[15] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A survey on evolutionary neural architecture search," *IEEE Transactions on Neural Networks and Learning Systems*, 2021, DOI: 10.1109/TNNLS.2021.3100554.

[16] D. Whitley, "A genetic algorithm tutorial," *Statistics and Cp=omputing*, vol. 4, no. 2, pp. 65–85, 1994.

[17] J. Wei *et al.*, "Genetic U-Net: Automatically designed deep networks for retinal vessel segmentation using a genetic algorithm," *IEEE Transactions Medical Imaging*, vol. 41, no. 2, pp. 292–307, 2022.

[18] "Image segmentation: Part 1," https://towardsdatascience.com/image-segmentation-part-1-9f3db1ac1c50, accessed: 2022-02-05.

[19] K.-S. Fu and J. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[21] G. Tong, Y. Li, H. Chen, Q. Zhang, and H. Jiang, "Improved u-net network for pulmonary nodules segmentation," *Optik*, vol. 174, pp. 460–469, 2018.

[22] J. Wang, J. Lu, G. Qin, L. Shen, Y. Sun, H. Ying, Z. Zhang, and W. Hu, "A deep learning-based autosegmentation of rectal tumors in mr images," *Medical Physics*, vol. 45, no. 6, pp. 2560–2564, 2018.

[23] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing cnn architectures using the genetic algorithm for image classification," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3840–3854, 2020.

[24] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394–407, 2019.

[25] R. de Lima Mendes, A. H. da Silva Alves, M. de Souza Gomes, P. L. L. Bertarini, and L. R. do Amaral, "gaCNN: Composing cnns and gas to build an optimized hybrid classification architecture," in *Proc. IEEE CEC*, 2021, pp. 79–86.

[26] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 240–248.