Engineering Science and Technology, an International Journal 46 (2023) 101502

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full length article

Differential evolution-based neural architecture search for brain vessel segmentation

Zeki Kuş [a], Berna Kiraz [a,b,*], Tuğçe Koçak Göksu [a], Musa Aydın [a], Esra Özkan [c], Atay Vural [c,d], Alper Kiraz [c,e,f], Burhanettin Can [a,b]

[a] Fatih Sultan Mehmet Vakif University, Department of Computer Engineering, Beyoğlu, Istanbul, 34445, Turkey
[b] Fatih Sultan Mehmet Vakif University, Data Science Research and Application Center (VEBIM), Beyoğlu, Istanbul, 34445, Turkey
[c] Koç University, Research Center for Translational Medicine (KUTTAM), Sarıyer, Istanbul, 34450, Turkey
[d] Koç University, School of Medicine Department of Neurology, Sarıyer, Istanbul, 34450, Turkey
[e] Koç University, Department of Physics, Sarıyer, Istanbul, 34450, Turkey
[f] Koç University, Department of Electrical and Electronics Engineering, Sarıyer, Istanbul, 34450, Turkey

ARTICLE INFO

Keywords:
Attention U-Net
Brain vessel segmentation
Differential evolution
Neural architecture search
U-Net

ABSTRACT

Brain vasculature analysis is critical in developing novel treatment targets for neurodegenerative diseases. Such an accurate analysis cannot be performed manually but requires a semi-automated or fully-automated approach. Deep learning methods have recently proven indispensable for the automated segmentation and analysis of medical images. However, optimizing a deep learning network architecture is another challenge. Manually selecting deep learning network architectures and tuning their hyper-parameters requires a lot of expertise and effort. To solve this problem, neural architecture search (NAS) approaches that explore more efficient network architectures with high segmentation performance have been proposed in the literature. This study introduces differential evolution-based NAS approaches in which a novel search space is proposed for brain vessel segmentation. We select two architectures that are frequently used for medical image segmentation, i.e. U-Net and Attention U-Net, as baselines for NAS optimizations. The conventional differential evolution and the opposition-based differential evolution with novel search space are employed as search methods in NAS. Furthermore, we perform ablation studies and evaluate the effects of specific loss functions, model pruning, threshold selection and generalization performance on the proposed models. The experiments are conducted on two datasets providing 335 single-channel 8-bit gray-scale images. These datasets are a public volumetric cerebrovascular system dataset (*vesseINN*) and our own dataset called *KUVESG*. The proposed NAS approaches, namely UNAS-Net and Attention UNAS-Net architectures, yield better segmentation performance in terms of different segmentation metrics. More specifically, UNAS-Net with differential evolution reveals high dice score/sensitivity values of 79.57/81.48, respectively. Moreover, they provide shorter inference times by a factor of 9.15 than the baseline methods.

1. Introduction

Brain is a fragile organ that consumes a high amount of energy. Protection of the brain is maintained by the blood–brain barrier and resources for energy are provided through the cerebral vasculature. Disruption of either of these structures is accepted as an important contributor of disease mechanism in many neurodegenerative disorders including Alzheimer's disease, multiple sclerosis, cerebral small-vessel disease, vascular dementia [1]. Imaging and subsequent image analysis of the cerebral vasculature are frequently used in biomedical research to uncover novel mechanisms that lead to the disturbance of cerebrovasculature in order to find novel treatment targets.

Vasculature is a web-like tubular structure and it is difficult to reliably quantify the vessel-related parameters such as length and branching in this complex structure by manual methods. High and irregular background signal is a frequently encountered problem in microscopy that can hinder correct identification of blood vessels, leading to increased error ratio in such analysis. Therefore, proper segmentation is a key to accurate vessel analysis. Vessel segmentation has been first performed in medical imaging in order to effectively

* Corresponding author at: Fatih Sultan Mehmet Vakif University, Department of Computer Engineering, Beyoğlu, Istanbul, 34445, Turkey.
E-mail addresses: zkus@fsm.edu.tr (Z. Kuş), bkiraz@fsm.edu.tr (B. Kiraz), tkocak@fsm.edu.tr (T.K. Göksu), maydin@fsm.edu.tr (M. Aydın), eozkan19@ku.edu.tr (E. Özkan), atayvural@ku.edu.tr (A. Vural), akiraz@ku.edu.tr (A. Kiraz), bcan@fsm.edu.tr (B. Can).

https://doi.org/10.1016/j.jestch.2023.101502
Received 6 February 2023; Received in revised form 16 July 2023; Accepted 31 July 2023
Available online 17 August 2023

monitor pathology samples [2]. Manual vessel segmentation is not a feasible process in terms of time, labor, and observer reporting variability because it corresponds to a time-consuming and tedious operation that requires professional expertise. Therefore, semi-automated or completely-automated methods are required for faster and more accurate vessel segmentation that will lead to critical improvements in treatment planning. The following factors make the vessel segmentation process difficult: (1) The forms of vessels do not adhere to a simple pattern; (2) The existence of additional structures (lesions and the optic disc) [3]; (3) Different imaging circumstances (low image contrast, noise, and pathological issues).

Vessel segmentation has become a hot topic in biomedical image analysis [4,5]. Several "handcrafted" and "supervised" approaches have been presented for vessel segmentation. Handcrafted approaches depend on the inherent features of vascular systems, whereas supervised methods are based on labeled ground truth images. Deep neural network architectures, which are among the supervised methods, have shown remarkable success in vessel segmentation [6–11]. These deep architectures are also used in brain segmentation studies [12,13]. The Computer Aided Disease Diagnosis system that uses CNN-assisted segmentation and SVM classification to identify brain tumors in 2D MRI slices is proposed in [14]. Ramya et al. [15] have proposed a novel framework consisting of filtering, enhancement, segmentation, feature extraction, and classification step for recognizing Alzheimer's disease (AD) using image processing techniques. In [16], a new method is proposed for detecting and classifying brain tumors using a deep saliency map, pre-trained EfficientNet-B0, and improved dragonfly optimization algorithms, and it outperforms previous accuracy results on brain tumor datasets. HHOCNN [17], which optimize hyperparameters and classifies segmented tumor regions, is proposed for brain tumor classification. Our work differs from these studies in terms of deep learning architecture, brain images and evaluation metrics.

Hyper-parameter Optimization (HPO) and Neural Architecture Search (NAS) are some of the most common problems for automated machine learning (AutoML) [18]. Every machine learning system has many hyper-parameters that need to be tuned, and the selection of these hyper-parameters directly affects the overall performance. The approaches proposed for NAS have aimed at finding the optimal neural network for a given machine learning problem. Therefore, NAS studies can be considered as a subset of HPO studies. Neural networks are generated in the pre-defined search space using the chosen method (usually meta-heuristics, and their performance is evaluated on a given dataset. The generated neural networks are expected to provide a good performance as well as low computational complexity.

In this study, differential evolution-based NAS approaches are presented for the segmentation of 2D brain vessel images. The conventional differential evolution (DE) and the opposition-based differential evolution (ODE) are used to search the optimal network architectures. Since U-Net based models are shown to be successful in bioimage segmentation, U-Net and Attention-UNet architectures are employed for NAS as baselines in this study. We introduce a novel search space combining the advantages of Nas-Bench-101 [19] and DARTS [20] which are frequently used search spaces for NAS studies. Accordingly, four differential evolution based NAS approaches, namely *UNAS-Net + DE, UNAS-Net + ODE, Attention UNAS-Net + DE*, and *Attention UNAS-Net + ODE* are presented. Experiments are conducted on a combination of two datasets consisting of images taken using two-photon microscopy. A total of 335 single-channel 8-bit grayscale images are provided by these datasets. 179 images of the *vesseINN* [21] dataset available in the literature were used together with 156 images obtained by the authors in the Cellular and Molecular Imaging Center at Koç University Translational Medicine Research Center (KUTTAM). This dataset is called *KUVESG*. Our proposed approaches are compared with the U-Net, Attention-UNet and ResNet + U-Net network in terms of twenty-one segmentation performance metrics and model complexity. Based on the experimental results, UNAS-Net improves the segmentation performance with respect to different performance measures and reduces the model complexity of the baseline U-Net and Attention U-Net. Moreover, although a single objective, i.e. minimizing the loss function, is considered in the DE-based search algorithms, these methods produce the models with both high segmentation performance and low inference times due to the proposed cell-based micro search space and macro search space.

The main contributions of this paper are listed as follows:

- A new search space combining the benefits of Nas-Bench-101 [19] and DARTS [20] is presented. We optimize encoder/decoder networks within UNAS-Net and Attention UNAS-Net and control the depth of the networks by using these search spaces, which consist of cell-based micro search space and macro search space. Thus, we generate efficient networks regarding segmentation performance and model complexity.
- DE and ODE algorithms are employed to search the optimal network architectures. To the best of our knowledge, this study is the first attempt to use the ODE in NAS. We demonstrate that DE-based NAS with the proposed search space gives better results. Furthermore, experimental results reveal that UNAS-Net and Attention UNAS-Net, which are best architectures generated by DE and ODE, outperform the U-Net and Attention U-Net in terms of segmentation performance and computational complexity.
- We publicly share the *KUVESG* dataset, which consists of mouse cortex images taken from a multiphoton microscope to fill the gap in annotated dataset for the vasculature segmentation. We invite researchers to use this dataset and share their findings with the scientific community.
- We perform ablation studies in order to show the effects of different loss functions, model pruning, generalization performance and threshold selection on the proposed models. We analyze the proposed method in detail and discuss the outcomes.

The rest of the paper is organized as follows: Section 2 provides a brief literature survey on brain vessel segmentation and NAS. Section 3 presents the detail of proposed DE-based NAS approaches. The experimental design including the dataset, performance measures and implementation details are presented in Section 4. The results of the experiments are discussed in Section 5. Finally, Section 6 presents the conclusion and future work.

2. Related work

Deep convolutional neural network (CNN) has emerged as an effective model for several vessel segmentation problems [22]. Numerous CNN architectures have been proposed for the retinal blood vessel segmentation [6,7,11]. A deep learning architecture with the structured prediction is presented in [23]. [24] proposes a multiple deep convolutional neural network (MDCNN). A cross-connected convolutional neural network (CcNet) that combines some extracted features at intermediate layers is presented in [25]. A fully convolutional deep learning architecture is also used for the retinal blood vessel segmentation [26, 27].

On the other hand, there are various CNN-based deep learning models presented for the brain vasculature analysis [28–30]. Fully Convolutional Dense Convolutional Network (FC-DenseNets) is shown to perform an automatic 3D vascular segmentation for two-photon microscopy images in [31]. Besides, a fully automated 3D CNN model, called BRAVE-NET (BRAinVEssel-NETwork), is applied to brain vessel segmentation on the dataset for the cerebrovascular diseases. A CNN-based framework, namely Vessel Segmentation & Analysis Pipeline (VesSAP), is presented to perform vascular analysis of the whole mouse brain. VesSAP consists of three stages: *(1) Pre-processing:* Cleaning is performed on the images obtained by light-sheet microscopy; *(2) Deep Learning:* Segmentation is performed with CNN-based network and 3D reconstruction of vasculature; *(3) Analysis:* 3D Vascular analysis of whole mouse brain and statistical evaluation.

U-Net is a specialized convolutional network model proposed for biomedical image segmentation [32]. Several U-Net based models such as deep residual U-Net, Attention-UNet, etc. have been successfully employed for various vessel segmentation tasks [8,9]. In [33], a U-Net architecture is developed for arterial brain vessel segmentation using the magnetic resonance (MR) images. U-Net with residual connection is presented to detect the vessels as well as the tiny one [34]. A U-Net based model with inception module and dense connections is employed for the retina blood vessel segmentation, the lung segmentation, and the brain tumor segmentation [35]. A hybrid method combining U-Net and random forest provides a good accurate results for vessel segmentation [36]. [37] presents an empirical comparison of three deep neural networks including auto-encoder, U-Net, and ResNet+U-Net for the brain vessel segmentation. A weighted attention and the skip connection scheme are involved in a U-Net architecture for the retina blood vessel segmentation [38]. Attention U-Net based models, a U-Net model with attention mechanism, have been proposed and evaluated on retinal vessel segmentation problems [39–41]. [42] introduces the Attention ResU-Net model, which combines the Attention U-Net and residual block structures for blood vessel segmentation. In [43], vertical and horizontal pooling operations are applied to create attention maps with the dual-direction attention block module in Fully Attention-based U-Net Network.

2.1. Neural architecture search

Recently, various approaches have been developed in the literature; reinforcement learning [44,45], gradient-based [46,47], evolutionary based [20,48,49]. Reinforcement learning based NAS approach is proposed in [45]. Successful results are obtained with the proposed structure on the CIFAR-10 dataset, which is one of the image classification tasks. Zoph et al. [50] are proposed a new NAS approach, namely NASNet. In the proposed approach, the network structure is divided into small blocks called cells and then the search is performed on these cells instead of the entire network architecture (cell-based search space). Each cell is represented as a graph without any edge or node limitation (there are some limitations in literature [19]. In this network structure, different cell structures are generated, since there is no repetitive pattern rule between operations in the cell. Liu et al. [20] introduce the differentiable architecture search (DARTS), which uses cell-based search space and provides successful results for image classification and natural language processing problems. NAS benchmarks such as NAS-Bench-101 [19], NAS-Bench-201 [51] and NAS-Bench-301 [52] have been published for a fair comparison of the proposed NAS methods and to reduce the computation time caused during the training of networks.

Evolutionary NAS approaches have achieved successful results for different problems [53,54]. Regularized Evolution approach is presented in [53], which gives competitive results as well as less computational complexity for NAS-Bench-101. In [55], a deep belief network-based NAS approach is proposed. Particle Swarm Optimization is used as a search algorithm in this study. Moreover, Differential Evolution (DE) yields better performance than five different approaches on different NAS benchmarks [56].

Although NAS studies on image classification problems have been increasing rapidly and giving successful results, there are limited NAS studies in the biomedical image segmentation tasks. NAS studies on the medical image segmentation can be examined in two different groups: 2D and 3D segmentation networks [48,57,58]. A NAS approach based on policy gradient reinforcement learning is proposed [59]. The authors generate networks with lower computational complexity than state-of-the-art approaches for cardiac MR images using the novel 2D segmentation network design, which is called Densely connected encoder–decoder CNN. In [60], NAS-Unet is presented for 2D medical image segmentation based on U-Net. The approach achieve higher segmentation performance than baseline U-Net for Promise12, Chaos, and ultrasound nerve datasets. An Automated Neural Network, namely AutoSegNet, is introduced in [61]. AutoSegNet searches the component of the 2D segmentation network via Recurrent Neural Network controller. It gives better segmentation performance for two different industrial medical image datasets than state-of-the-art methods such as U-Net and pyramid scene parsing network (PSPNet). Evolutionary NAS methods have also developed for the 2D retinal vessel segmentation problem [62,63]. Moreover, genetic algorithm based NAS approaches are used as search algorithms to find the optimal U-Net architectures [63,64].

3. DE-based NAS for U-Net architectures

Due to its proven success in brain vessel segmentation [33,37,65] U-Net model is employed as baseline in NAS approach followed in this study. We also considered the improved version of U-Net architecture, namely Attention U-Net [66]. We explored two DE-based NAS approaches for searching the optimal U-Net and Attention U-Net architectures. These are the conventional DE and opposition-based differential evolution (ODE) search algorithms. Conventional DE [67], categorized as a population-based meta-heuristic algorithm, has achieved successful results in NAS studies [56]. Moreover, it is stated that ODE, in which opposition-based learning is utilized, accelerates the convergence speed of DE [68]. This section presents the details of the proposed NAS approaches including DE, ODE, UNAS-Net, and Attention UNAS-Net.

3.1. Differential evolution

DE, presented in [67], is a well-known evolutionary algorithm especially for continuous optimization problems. In DE, each candidate solution is represented by a real-valued vector. The first population is generated at random and is evaluated. Following the initialization step, DE uses mutation, crossover, and selection operators to create a new population.

In this study, each candidate solution ($X_{i,G}$, where $i = 1, \ldots, NP$, NP is the population size, and G is the generation number) is a vector of real-valued numbers in the range $[0, 1]$. The fitness function is defined as the dice score obtained by the candidate network (see Section 4.4) (used in [48,57]) and is given as:

$$Dice\ Score = \frac{2|y \cap \hat{y}|}{|y| + |\hat{y}|}, \tag{1}$$

where y is the ground truth image; $\hat{y}$ denotes the predicted image. We consider the classical variant of DE (denoted as *DE/rand/1/bin*) in which the random mutation (*DE/rand/1*) and binomial crossover are employed. A mutant vector ($V_{i,G}$) at generation G is generated for each solution in the current population (target vector) as:

$$DE/rand/1: \quad \mathbf{V}_{i,G} = \mathbf{X}_{r1,G} + F \times (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}), \tag{2}$$

where $r1, r2, r3 \in \{1, 2, \ldots, NP\}$ are random integers and $F \in [0, 2]$ is the scaling factor. After generating mutant vectors, the corresponding trial vectors ($U_{i,G}$) are created using binomial crossover given as:

$$U^k_{i,G} = \begin{cases} V^k_{i,G}, & \text{if } rand() \le CR \\ X^k_{i,G}, & \text{otherwise} \end{cases} \quad \text{for } k = [1, \ldots, D], \tag{3}$$

where CR is the crossover probability, $rand()$ is the random number drawn from the uniform distribution, D is number of decision variables. In case of boundary constraint violation, the values exceeding the boundaries are replaced with a random value sampled from uniform distribution ($[0, 1]$) [69]. Finally, the selection process is applied. In the selection stage, each target vector is compared with its corresponding trial vector in terms of objective function value. If the target vector is superior to its corresponding trial vector, the target vector is selected for the next generation [70,71]. Since the problem addressed in this

study is a maximization problem, the vector for the next generation is selected as:

$$X_{i,G+1} = \begin{cases} U_{i,G}, \text{if } f(U_{i,G}) > f(X_{i,G}) \\ X_{i,G}, \text{otherwise} \end{cases} \quad (4)$$

where $f(.)$ denotes the fitness value of the corresponding vector. The pseudo-code of DE is given in Algorithm 1.

Algorithm 1: Differential Evolution Pseudo-code

input: NP: Population size; $f(\cdot)$: Fitness value
$G \leftarrow 0$
$P_G \leftarrow$ Generate initial population
for *i to* NP **do**
  $f(X_{i,G}) \leftarrow$ **Fitness_Eval(**$X_{i,G}$**)**
$G = G + 1$
while *Termination condition is not satisfied* **do**
  for *i to* NP **do**
    $V_{i,G}$ = **Mutation(**$X_{i,G-1}$**)**
    $U_{i,G}$ = **Crossover(**$X_{i,G-1}$, $V_{i,G}$, CR**)**
    $U_{i,G}$ = **Boundary_check(**$U_{i,G}$**)**
    $f(U_{i,G}) \leftarrow$ **Fitness_Eval(**$U_{i,G}$**)**
    if $f(U_{i,G}) > f(X_{i,G-1})$ **then**
      $P_{G,i} \leftarrow U_{i,G}$
    else
      $P_{G,i} \leftarrow X_{i,G}$
  $G = G + 1$

3.2. Opposition-based differential evolution

In ODE [68], the opposition-based learning (OBL) is employed in the population initialization and generation steps of DE. In this study, we consider the classical variant of DE (*DE/rand/1/bin*) in ODE.

The opposition-based initialization consists of the following four steps. (1) The initial population (P_0) is randomly generated. (2) The opposite of each solution in the initial population is calculated in order to generate the opposite population (OP_0). The opposite of a solution in the initial population is calculated as:

$$OP^k_{i,0} = a^k + b^k - X^k_{i,0}, \quad for\ i : \{1, \ldots, NP\}; k : \{1, \ldots, D\}, \quad (5)$$

where $OP^k_{i,0}$ is kth value of the ith individual in opposition of the initial population, respectively; a^k and b^k are lower and upper bounds of the kth decision variable; D is the number of decision variables; NP is the population size. (3) The initial and opposite populations are merged. (4) The best NP solutions are selected from the merged population as the initial population ($P_G \leftarrow select(P_G \cup OP_G)$).

On the other hand, the opposition-based generation jumping step is performed with a jumping probability (JR) during the generation process. In the jumping step, the opposite population of the current population is computed. Instead of fixed values, the opposite value of each variable in the solution is calculated dynamically using:

$$OP^k_{i,G} = Min^k_G + Max^k_G - P^k_{i,G} \quad for\ i : \{1, \ldots, NP\}; k : \{1, \ldots, D\}, \quad (6)$$

where Min^k_G and Max^k_G denote the minimum and maximum values of kth value in population P at generation G (P_G), respectively. Finally, the current population and its opposite population are merged and NP best solutions are selected from the merged population ($P_G \leftarrow select(P_G \cup OP_G)$). The pseudo-code of ODE is given in Algorithm 2.

3.3. UNAS-Net

In this study, we consider the UNAS-Net [72] to optimize the U-Net architecture. UNAS-Net consists of cell-based micro search space and macro search space as in Efficient Multi-objective NAS framework [48].

Algorithm 2: Opposition-Based Differential Evolution Pseudo-code

$G \leftarrow 0$
$P_G \leftarrow$ Generate initial population
/* **Generate Opposition of Initial Population** */
for $i = 0; i < NP; i$++ **do**
  for $k = 0; k < D; k$++ **do**
    $OP^k_{i,G} = a^k + b^k - X^k_{i,G}$
for *i to* NP **do**
  $f(X_{i,G}) \leftarrow$ **Fitness_Eval(**$X_{i,G}$**)**
  $f(OP_{i,G}) \leftarrow$ **Fitness_Eval(**$OP_{i,G}$**)**
/* **Generate Opposition of Initial Population** */
$P_G \leftarrow \{P_G, OP_G\}$; `// Select best NP solution`
$G = G + 1$
while *Termination condition is satisfied* **do**
  for *i to* NP **do**
    $V_{i,G}$ = **Mutation(**$X_{i,G-1}$**)**
    $U_{i,G}$ = **Crossover(**$X_{i,G-1}$, $V_{i,G}$, CR**)**
    $U_{i,G}$ = **Boundary_check(**$U_{i,G}$**)**
    $f(U_{i,G}) \leftarrow$ **Fitness_Eval(**$U_{i,G}$**)**
    if $f(U_{i,G}) > f(X_{i,G-1})$ **then**
      $P_{G,i} \leftarrow U_{i,G}$
    else
      $P_{G,i} \leftarrow X_{i,G}$
  /* **Opposition Based Generation Jumping** */
  if *rand(0, 1)* $< JR$ **then**
    for $i = 0; i < NP; i$++ **do**
      for $k = 0; k < D; k$++ **do**
        $OP^k_{i,G} = Min^k_G + Max^k_G - P^k_{i,G}$
      $f(OP_{i,G}) \leftarrow$ **Fitness_Eval(**$OP_{i,G}$**)**
    $P_G \leftarrow \{P_G, OP_G\}$; `// Select best NP solution`
  /* **Opposition Based Generation Jumping** */ $G = G + 1$

The cell-based micro search space contains the hyper-parameters that determine the structure of a cell, while the macro search space includes two hyper-parameters: The total number of cells in the architecture and the number of feature maps in first cell. This subsection presents the details of UNAS-Net including the search space and encoding scheme.

3.3.1. Search space

The UNAS-Net is composed of encoder and decoder cells for down-sampling and upsampling, respectively. Encoder cells perform feature extraction by applying a series of convolution operations. The max-pooling operation with stride 2 which is used to reduce the image size, follows the encoder cells. On the other hand, decoder cells reconstruct the image using the extracted features in the encoder cells and increase the size of input using transpose convolution. At the final step, $2D$ convolution with a filter size of 1×1 is applied to the output from the last decoder cell, which is followed by the *sigmoid function* to obtain the segmented image. The general structure of UNAS-Net is illustrated in Fig. 1.

Cell-based micro search method [19,20], which defines the structure of encoder and decoder cells, is used in this study. Each cell is represented as a directed acyclic graph ($G = (V, E)$ where V and E denote the set of vertices and edges, respectively). A cell structure is illustrated in Fig. 2. The graph consists of an input (v_1), an output (v_7) and up to 5 intermediate vertices (v_2 to v_6). Each intermediate vertex ($v \in V \setminus \{v_1, v_7\}$) has a label $l \in L$ that represents the operation, where L is the set of labels. Each directed edge ($e \in E$) represents the connections between the vertices. In this study, we restrict the number of vertices ($|V| \leq 7$), the number of edges ($E \leq 9$), and the number of possible operations ($|L| \leq 14$) to reduce the search space.

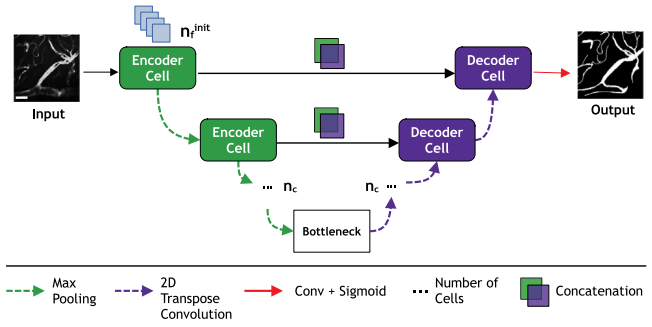

Fig. 1. The general structure of UNAS-Net. Scale bar indicates 100 μm.

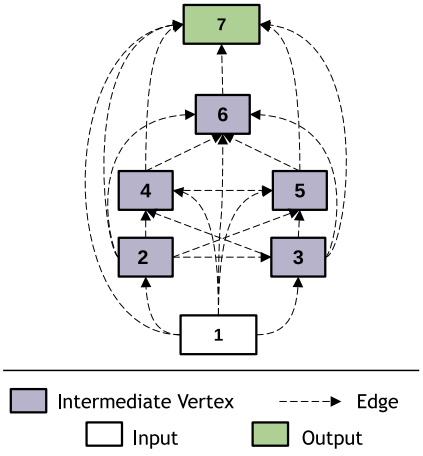

Fig. 2. Overview of cell-based micro search space and the general structure of encoder and decoder cells.

Table 1
Possible $2D$ convolution operations in intermediate vertices based on the DARTS [20].

Operation	Description
1×1 conv	1×1 Convolution
3×3 conv	3×3 Convolution
5×5 conv	5×5 Convolution
7×7 conv	7×7 Convolution
3×3 depconv	3×3 Depthwise Separable Convolution
5×5 depconv	5×5 Depthwise Separable Convolution
7×7 depconv	7×7 Depthwise Separable Convolution
3×3 dilconv	3×3 Dilated Convolution
5×5 dilconv	5×5 Dilated Convolution
7×7 dilconv	7×7 Dilated Convolution
3×3 asymconv	3×3 Asymmetric Convolution
5×5 asymconv	5×5 Asymmetric Convolution
7×7 asymconv	7×7 Asymmetric Convolution
skip	Skip-Connection

Table 1 provides the fourteen possible $2D$ convolution operations based on DARTS [20].

There are two hyper-parameters in the macro search space: Number of cells and number of feature maps in the first cell (See Fig. 1). UNAS-Net architecture has equal number of encoder and decoder cells. The total number of cells in the architecture (N_c) is calculated as the sum of the number of encoder cell (n_c), the number of decoder cells (n_c), and the bottleneck cell ($N_c = 2n_c + 1$). $n_c \in \{2, 3, 4, 5\}$ is a decision variable to be optimized by search algorithms. On the other hand, the number of features extracted from the given input directly affects the performance of the network. $n_f^{init} \in \{8, 16, 32\}$ indicates the number of feature maps to be generated as a result of operations on the first encoder cell. The number of feature maps is doubled after each max-pooling operation.

3.3.2. Encoding

As mentioned before, we consider two DE-based algorithms as the search method. In these algorithms, a real-valued vector is used to represent a candidate solution that has 28 variables ($\in [0, 1]$) used for the structure of an encoder/decoder cell (21 variables), the convolution operations (5 variables), and the hyper-parameters in the macro search space (2 variables).

Since a cell is represented as a directed acyclic graph (DAG), we consider the adjacency matrix to describe the DAG. Cell structure with 7 vertices imposes the adjacency matrix to be a 7×7 ($|V| \leq 7$) upper triangular matrix with entries 0 or 1 whose elements show whether two vertices are connected or not. The first 21 variables in the candidate solution indicate the adjacency matrix that represents the structure of the encoder/decoder cell. Besides, in the cell structure (see Fig. 2), there are five intermediate vertices each of which has a label. Therefore, the next 5 elements (22nd to 26th) indicate the selected labels among the fourteen possible ones given in Table 1. Finally, the last two elements in the solution indicate the number of cells (n_c) and the number of feature maps of the first cell (n_f^{init}).

Since a candidate solution is represented by a real-valued vector, we use a mapping scheme to construct an UNAS-Net architecture [56] from the real-valued vector. Fig. 3 illustrates this scheme step by step with an exemplary real-valued vector. These steps are given as follows:

(1) The first 21 elements in the solution are used for the adjacency matrix. Each value in the solution indicates whether there is a connection between vertices or not: if the value is greater than 0.5, that means there is an edge between the corresponding pair of vertices; therefore, the corresponding entry in the adjacency matrix should be 1. As seen in Fig. 3, the first element indicates whether or not there is a connection between *vertex 1* and *vertex 2*. Since the value is 0.61, the entry in the first row and the second column of the adjacency matrix becomes 1. In addition, the second element is used for the connection between *vertex 1* and *vertex 3*. The corresponding value is 0.49; therefore, the corresponding entry becomes 0. For the remaining elements, we convert the real value to binary one in the same manner.

(2) The next 5 elements (22nd to 26th) in the solution indicate the selected operations. Since there are fourteen possible operations (labels), the range $[0, 1]$ is divided into fourteen equal-sized bins: The range $[0, 0.07)$ corresponds to the first possible operation, namely 1×1 conv given in Table 1; the range $[0.07, 0.14)$ corresponds to 3×3 conv; the range $[0.14, 0.21)$ corresponds to 5×5 conv; and so on. The 22nd element indicates the label for *vertex 2* in the cell structure. Since the corresponding value is 0.94, we use the skip operation in *vertex 2*.

(3) The last two elements indicate the number of cells (n_c) and the number of feature maps of the first cell (n_f^{init}), respectively. n_c can takes four different values, therefore the range $[0, 1]$ is divided into four equal-sized bins: The range $[0, 0.25)$ corresponds to the value of 2, the range $[0.25, 0.50)$ corresponds to the value of 3, and so on. On the other hand, n_f^{init} can be 8, 16, or 32, so the range $[0, 1]$ is separated into three equal-sized bins. In Fig. 3, the value of second last element is 0.75, which means that n_c is set to 5. Similarly, the last element is 0.96, which indicates that the value of n_f^{init} is 32.

(4) As a result of the first two steps, we create the adjacency matrix with its labels.

(5) We generate the cell structure for the encoder/decoder cell using adjacency matrix and labels.

(6) We construct the UNAS-Net architecture using the cell structure, n_c, and n_f^{init}.

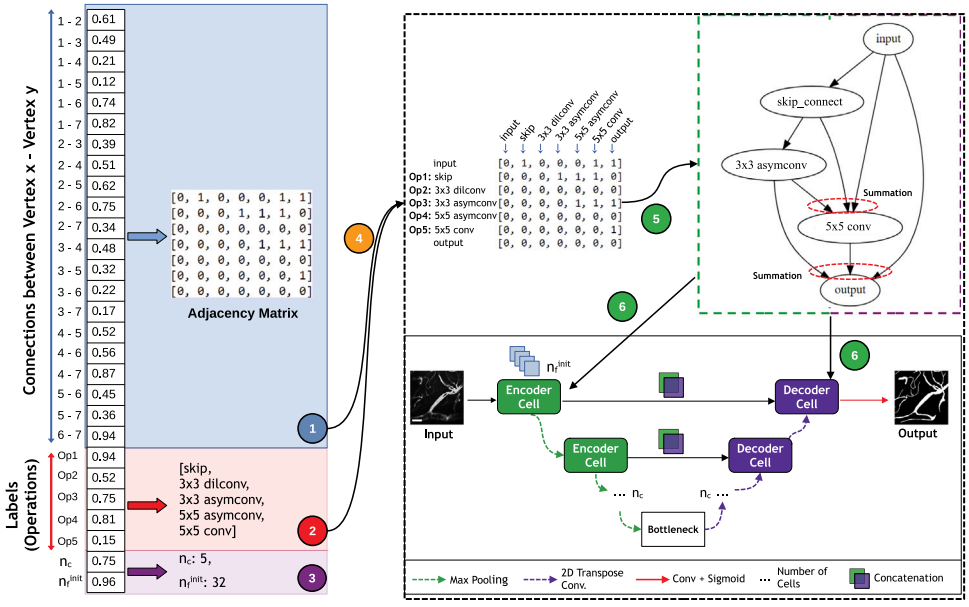

Fig. 3. The encoding for UNAS-Net Architecture (left part) and the steps for mapping: **Step 1:** Create the adjacency matrix; **Step 2:** Select the labels; **Step 3:** Select the values for two hyper-parameters; **Step 4:** Combine the adjacency matrix with its labels. **Step 5:** Create a DAG for the cells. **Step 6:** Construct UNAS-Net using DAG and two hyper-parameters. Scale bar indicates 100 μm.

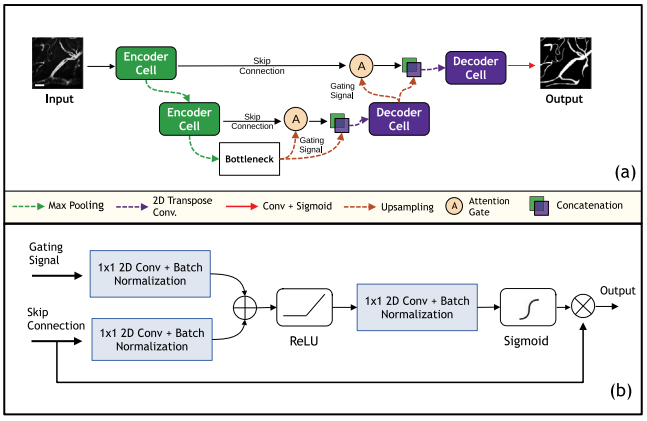

Fig. 4. The overview of Attention-UNAS-Net. (a) Attention-UNAS-Net Architecture, (b) Attention Gate. Scale bar indicates 100 μm.

3.4. Attention UNAS-Net

An improved version of U-Net, namely Attention U-Net, is proposed for the segmentation of medical images in [66]. Attention gates are utilized into the architecture. The features learned at the encoding step in the U-Net structure must be concatenated with other features before the decoding step. However, the low-level features learned in the first stages and the high-level features learned in the later stages do not have the same significance for segmentation. Therefore, these features need to be weighted in the concatenation stage. The attention gate is used for weighting different features.

The Attention UNAS-Net structure based on Attention U-Net is proposed in this study. This structure uses the same search space and encoding as in UNAS-Net. Besides, attention gates are utilized before decoding steps. The Attention UNAS-Net structure is shown in Fig. 4.

4. Experimental design

In this section, we present the URes-Net model proposed in [37] for comparison. Then, we provide the dataset used in the experimental study, which is followed by the performance measures and the implementation details for the approaches.

4.1. URes-Net

ResNet + U-Net network [37] used for comparison purposes consists of two parts: the first part is called the narrowing path, and the second part is called the expansion path. A narrowing path consists of iterating two 3 × 3 convolutions, reducing the size of the input data across the layers. The number of features obtained in the input layer is 64, and this number is doubled at each step of the narrowing path. In this path, downsampling is done with a maximum pooling operation of 2 × 2 (stride = 2). At each step of the expansion path, 2 × 2 upsampling is performed and the resulting feature channels are concatenated with the feature channels in the narrowing path in the same step. As the number of features is halved after each upsampling in the growth path, expansion and narrowing paths are symmetrical.

4.2. Dataset

Experiments were performed by combining two similar datasets. The first dataset, *vesseINN* [21], is an open source volumetric cerebrovascular system dataset obtained by two-photon microscopy. This dataset consists of a total of 12 volumetric stacks containing images of mouse cortex and human squamous cell carcinoma tumors. In this study, we employed 9 stacks containing a total of 179 images with 512 × 512 px dimensions from *vesseINN*. The second dataset, *KUVESG*, consists of mouse cortex images taken from cleared mouse brain samples with the 3DISCO procedure. The procedure is explained elsewhere [73]. Briefly, this organic solvent-based clearing protocol uses tetrahydrofuran for dehydration and dibenzyl ether (DBE) for matching the refractive index. After completing the protocol, 1 mm thick cleared brain slides in DBE solution were imaged with the confocal setup of the Leica multiphoton system TCS SP8 microscopy. The brain vasculature was visualized with premortem transcardial perfusion of gelatin-albumin-FITC solution (Sigma, A9771). This dataset consists of 39 images of 1024 × 1024 px dimensions. We obtained 156 images of 512 × 512 px dimensions from *KUVESG* by dividing each image into 4 equal parts. With the combination of these two datasets, a total of 335 single-channel 8-bit grayscale images was obtained. Among these

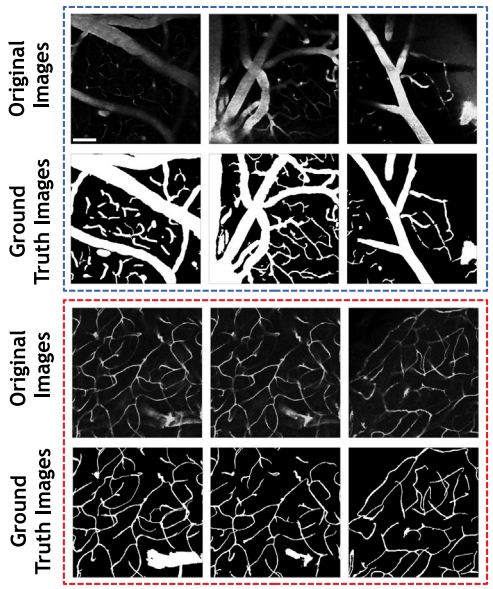

Fig. 5. Original and ground truth image pairs from the *vesseINN* (blue dashed line border) and *KUVESG* datasets (red dashed line border). Scale bar indicates 100 μm.

images, 301 were used for model training and 30 for tests. Fig. 5 shows the image pairs selected from *vesseINN* and *KUVESG* datasets [74].

4.3. Performance measures

Performance evaluation of a deep-learning based segmentation method requires multiple performance measures which are typically divided into six categories [75,76]. Overlap-based measures like the Dice Coefficient and the Intersection Over Union are frequently used because they are easily interpretable, implementable, and comparable [77–79]. Other measure categories such as volume, pair counting, probabilistic and spatial distance-based are used less frequently. The boundary and alignment of vessels are more important for our problem. Therefore, for precise performance evaluations of our methods we consider twenty-one performance measures belonging to five categories (excluded the volume based) as recommended in [75]. These performance measures are listed in Table 2.

4.4. Implementation details

In order to measure the actual performance of deep learning networks, the network must be trained with the entire training set during long training epochs and evaluated on the test set that is not included in the learning process. However, there are many candidate networks in NAS studies whose performance should be evaluated on GPU. It is infeasible to train all of these candidate networks with whole training data over long training periods. Therefore, the study is divided into two steps: Short-term evaluation and long-term evaluation.

In the short-term evaluation step, the candidate networks generated during the running of the heuristic method are evaluated (lines 4 and 6 in Algorithm 1, and lines 8, 9, 18 and 27 in Algorithm 2). At this step, a subset consisting of half of the training dataset is used to train the generated candidate networks, and the quality of a network is evaluated on the validation set created using 10% of this subset. When creating these sets, the images are randomly selected and the same images are used for a fair evaluation of the generated networks. We have implemented the *EarlyStopping* process in order to use the total training time more efficiently. In the short-term evaluation step, each candidate network is trained for a maximum of 36 training epochs [19]. However, if the validation loss value obtained by a network on the validation set does not improve over three consecutive training epochs, the training is interrupted at that step. This way, we were able to evaluate more candidate networks in less time. The best dice score obtained at the end of the training determines the quality of this candidate network and is used as the fitness value of the solution. The selected heuristics are terminated after training 500 network architectures on the GPU.

In the long-term evaluation step, the actual performance of the top five networks obtained as a result of running heuristics is evaluated. At this step, the best networks are initialized with random weights (the weights learned in the short-term evaluation step are not used to make a fair evaluation) and trained with the whole training dataset over 200 training epochs. These trained networks are evaluated on the test set, which has never been included in the training before (including the short-term evaluation step), and the actual performance of the network is obtained. Data augmentation is not applied at any stage of the study.

We consider the following parameter settings of DE and ODE as recommended in [56,69]: the population size (NP) is set to 20; the scaling factor F is 0.5; the crossover probability (CR) is set to 0.5. Besides, the jumping probability (JR) in ODE is 0.3.

The Pytorch library was used for the networks generated during the NAS study. The URes-Net network is implemented with the Tensorflow 2 library. The following parameters are used in common for UNAS-Net and URes-Net: Optimizer: Adam, Loss Function: Dice loss, Learning Rate: 1e-3 and Batch Size: 2 (This value has chosen due to memory problem) [48]. The following hardware has been used for the experimental studies: Ryzen 5600X processor, 12 GB RTX 3060 GPU, 16 GB RAM. The source code and dataset are available at Github.

Table 2
A summary of the performance measures selected in this study. The category column indicates the performance measure group to which each performance measure belongs.

Category	Performance measure
Overlap Based	Dice (DICE) Intersection over Union (IoU) Jaccard Index (JAC) Sensitivity (SNS) Specificity (SP) Sensibility (SB) Global consistency error (GCE) Conformity (CNF) Accuracy (ACC) Precision (PRC)
Pair Counting Based	Rand Index (RI) Adjusted Rand Index (ARI)
Information Theoretic Based	Mutual information (MI) Variation of information (VOI)
Probabilistic Based	Interclass correlation (ICC) Probabilistic distance (PBD) Cohen's kappa (KAP) Area under ROC Curve (AUC)
Spatial Distance Based	Hausdorff distance (HD95) Average Hausdorff distance (AHD) Mahalanobis Distance (MHD)

5. Results and discussion

5.1. Comparison of U-Net models

Both high segmentation performance and short inference time were achievement goals for the generated models. For this reason, numerical comparisons are performed in two different ways: Segmentation performance and model complexity. While 21 different measures described in Section 4.3 are used to evaluate the segmentation performance, the floating point operations (FLOPs) measure is used for accurate and reliable evaluation of model complexity [80].

The proposed methods are compared with the U-Net and Attention U-Net architectures which are considered as state-of-the-art baseline

Table 3
Performance results obtained by different approaches based on different measures. Each values obtained for five different seed values. In each row, the best approach is marked in bold.

Performance measure	U-Net	UNAS-Net		Attention U-Net	Attention-UNAS-Net		URes-Net
		DE	ODE		DE	ODE	
DICE	76.93	78.80↑	**79.57**↑	76.36	78.86↑	79.36↑	79.38
IoU	63.84	65.72↑	66.74↑	63.01	65.78↑	66.49↑	**66.81**
JAC	64.54	66.36↑	67.42↑	63.65	65.74↑	66.27↑	**67.60**
SNS	74.55	81.14↑	**81.48**↑	76.49	76.93↑	78.56↑	80.32
SP	**98.39**	97.52↓	97.81↓	97.71	98.02↑	97.90↑	97.85
SB	**83.67**	75.71↓	77.81↓	77.84	81.94↑	79.79↑	79.77
GCE	7.07	7.46↓	6.97↑	7.74	7.02↑	7.15↑	**6.71**
CNF	27.92	44.87↑	**47.70**↑	31.01	43.05↑	44.58↑	47.04
ACC	95.96	95.84↓	96.19↑	95.56	96.12↑	96.04↑	**96.34**
PRC	**83.45**	79.09↓	79.85↓	79.90	82.32↑	81.36↑	80.83
RI	92.39	92.15↓	92.75↑	91.69	92.63↑	92.50↑	**93.01**
ARI	71.14	72.38↑	73.66↑	69.88	72.34↑	72.62↑	**73.97**
MI	25.85	27.07↑	27.65↑	25.47	26.62↑	26.83↑	**27.78**
VOI	38.41	40.32↑	38.65↑	**41.30**	38.83↓	39.11↓	37.84
ICC	−3.44	−3.64↓	−3.36↑	−3.78	−3.40↑	−3.47↑	**−3.21**
PBD	36.04	27.57↑	**26.15**↑	34.49	28.47↑	27.71↑	26.48
KAP	75.15	76.79↑	77.78↑	74.27	76.46↑	76.82↑	**77.89**
AUC	86.47	89.33↑	**89.64**↑	87.10	87.48↑	88.23↑	89.09
HD95	26.63	14.72↑	15.93↑	30.81	13.79↑	11.33↑	**10.56**
AHD	9.73	9.38↑	**8.94**↑	10.22	9.40↑	9.52↑	9.02
MHD	13.64	13.54↑	12.05↑	14.48	12.00↑	12.48↑	**10.93**

Table 4
Comparison of different approaches in terms of model complexity, inference time in CPU and size. FLOPs in gigabytes; Time in seconds; #Params and Model Size on Disk in megabytes.

Method	FLOPs (G)	# Params (M)	Model size (M)	Time
U-Net	436	31	121	32.72
UNAS-Net + DE	220	3	14	27.10
UNAS-Net + ODE	84	6	25	20.72
Attention U-Net	531	34	136	39.56
Attention-UNAS-Net + DE	58	5	21	13.86
Attention-UNAS-Net + ODE	175	10	40	27.55
URes-Net	436	34	407	32.46

methods for medical image segmentation. The principal goal of this study is to use DE and ODE heuristics in order to obtain models better that the baseline methods in terms of segmentation performance and model complexity. Segmentation performance results are shown in Table 3. The results shown in this table are the averages of the results from five independent runs of each method. The ↑ and ↓ signs shown in Table 3 indicate whether the proposed method provides improvement for the corresponding performance criterion on the baseline. As a baseline, we used the U-Net for the UNAS-Net and the Attention U-Net for the Attention-UNAS-Net.

Table 3 shows that the best UNAS-Net network generated using DE and ODE heuristics outperforms the segmentation performance of U-Net in most of the segmentation measures. UNAS-Net + DE and UNAS-Net + ODE outperform U-Net in 14 of 21 and 18 of 21 measures, respectively. Comparing the UNAS-Net + DE and UNAS-Net + ODE results reveals that UNAS-Net + ODE has better segmentation performance in 19 of the 21 measures. The model complexities of the methods are compared in Table 4 which shows that UNAS-Net + DE and UNAS-Net + ODE perform 1.98 and 5.19 times less FLOPs than U-Net, respectively. In terms of model complexity, it is clear that the UNAS-Net + ODE technique uses 2.62 times less floating point operations than UNAS-Net + DE. As a result, it has been observed that the ODE heuristic for UNAS-Net gives better results than U-Net and UNAS-Net + DE in terms of both segmentation performance and model complexity. Based on this, we can say that the ODE heuristic for UNAS-Net investigates the search space better than DE.

Another proposed NAS-based method, Attention-UNAS-Net, is compared with Attention U-Net and the segmentation performance of the Attention U-Net network is improved. Table 3 shows that Attention-UNAS-Net + DE and Attention-UNAS-Net + ODE achieved better segmentation performance than baseline Attention-UNet for all measures except the VOI. There is no clear difference between the DE and ODE heuristics for Attention-UNAS-Net, as in UNAS-Net. The best network generated with the ODE heuristic outperformed the DE heuristic in terms of 12 measure. The Attention-UNAS-Net + DE and Attention-UNAS-Net + ODE methods produced 9.15 and 3.03 times less complex models compared to Attention-UNAS-Net, respectively. DE generates a less complex model than ODE for Attention-UNAS-Net, whereas ODE gives better segmentation results.

The NAS methods proposed in this study are also compared with the URes-Net architecture proposed by [37]. The segmentation performance of U-Net and Attention-UNet has been enhanced by URes-Net for all measures (except VOI for U-Net and Attention-UNet; SP, SB, PRC for U-Net). It performs the same number of floating-point operations as U-Net but 1.21 times less than Attention-UNet. Although URes-Net shows better segmentation performance overall than UNAS-Net and Attention-UNAS-Net-based methods, it is up to 7 times more complex in terms of computational complexity.

All methods are also compared visually in Fig. 6. The number of overlapping pixels on the ground truth and output, which is called true positive pixels, is not sufficient to evaluate the performance of medical segmentation studies. In addition, the network model should not mark areas that are not vessels on the ground truth. This type of error is called a False Positive (FP) Error. Finally, the areas that vessels on the ground truth may not be marked on the output. This is called a False Negative (FN) Error. According to these definitions, TP pixels are expected to be more on the output obtained by the network, and FP and FN pixels are expected to be as few as possible. Fig. 6 confirms that UNAS-Net models perform more accurate segmentation than baseline U-Net network in terms of TP, FP and FN pixels. However, it seems that UNAS-Net is not as successful in segmentation of thin vessels as thick vessels. Attention-UNAS-Net models are compared with the baseline Attention-UNet in Fig. 6. Images segmented by the Attention-UNAS-Net models have more TP pixels and fewer FP and FN pixels than Attention-UNet.

As a result, the UNAS-Net + DE and UNAS-Net + ODE methods outperformed the baseline U-Net in 14 of 21 and 18 of 21 measures, respectively. Additionally, these NAS methods require 1.98 and 5.19 times fewer FLOPs than U-Net. The Attention-UNAS-Net + DE and Attention-UNAS-Net + ODE methods also achieved better segmentation

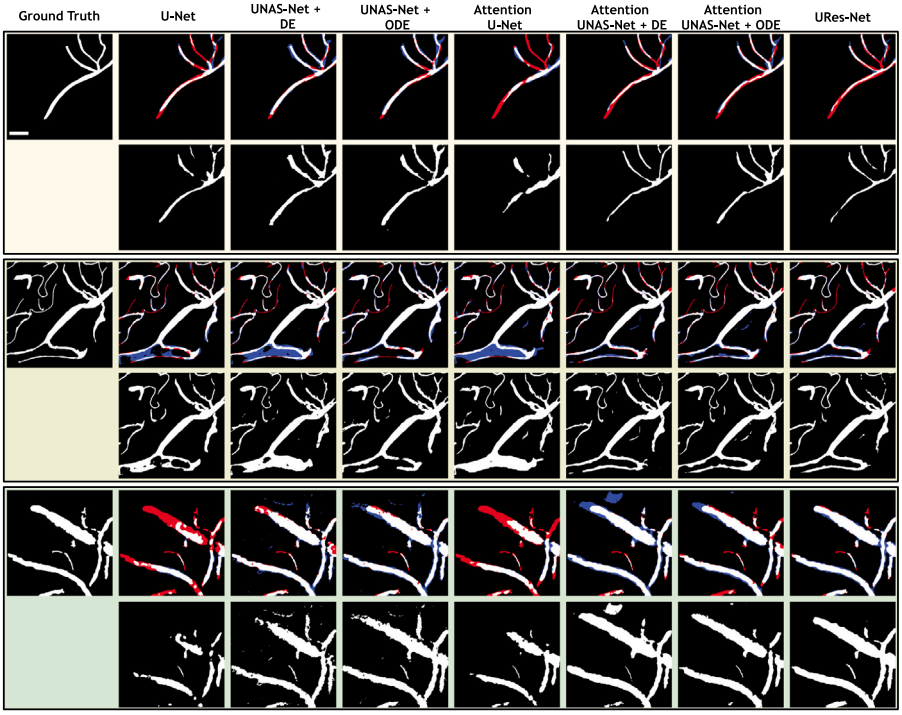

Fig. 6. Comparison of different medical image segmentation models visually. The first row shows the error maps (White: True positive pixels; Red: False negative pixels; Blue: False positive pixels) for each method, and the second row indicates the segmentation result of these methods. Scale bar indicates 100 μm.

performance with up to 9.15 less complex models than the baseline Attention-UNet, except for the VOI measure. The NAS methods are compared to the state-of-the-art URes-Net architecture proposed by [37], which shows better segmentation performance overall but is up to 7 times more complex in terms of computational complexity. In summary, the proposed NAS methods outperform the baseline models in terms of segmentation performance and model complexity, achieving highly competitive results with up to 7 times less complex networks than state-of-the-art URes-Net.

5.2. Ablation studies

In this section, ablation studies are performed for the best UNAS-Net ODE (U-ODE) and Attention-UNAS-Net (Att-ODE) networks (results are shown in Table 3). The impact of the loss function, n_c (number of cell) parameter, and threshold value on segmentation performance is examined. We train the models with all training images during 200 epochs, except for the threshold comparison. Then, we test it with all images in the test set as in the long-term evaluation for all ablation study experiments. The results tables show the average values of five performance metrics obtained as a result of training the best models (see Table 3) with five different seed values and testing them on all test images.

5.2.1. Loss function

In this part, we explore the effect of loss function since it has a direct impact on the segmentation performance. To determine the best loss function, the performances of UNAS-Net ODE and Attention-UNAS-Net ODE are evaluated on test sets with different loss functions: Dice, Jaccard, Binary Cross Entropy (BCE), Dice + BCE and Dice + BCE + Jaccard [75,81]. Table 5 presents the results of different loss functions for both UNAS-Net ODE and Attention-UNAS-Net ODE. For UNAS-Net ODE, Dice + BCE + Jaccard gives the best results in terms of DICE, IoU and ACC, while BCE gives the best results in terms of SP. Besides, Dice gives the best results in terms of SNS. All loss functions except BCE improve the performance of the baseline loss functions (Dice loss) in terms of DICE, IoU, SP and ACC values. The selection of the loss function provides a minimum of 0.35 and a maximum of 0.94 points of improvement.

Table 5
Performance results of different loss functions for UNAS-Net ODE and Attention-UNAS-Net ODE.

	Performance Metrics				
	UNAS-Net ODE				
Loss function	DICE	IoU	SNS	SP	ACC
Dice	79.57	66.74	**81.48**	97.81	96.19
Jaccard	79.87	67.31	78.19	98.46	96.49
BCE	77.94	64.82	76.14	**98.54**	96.52
Dice + BCE	79.66	67.03	77.92	98.35	96.40
Dice + BCE + Jaccard	**80.13**	**67.68**	78.60	98.38	**96.54**
	Attention-UNAS-Net ODE				
Dice	79.36	66.49	**78.56**	97.90	96.04
Jaccard	78.79	66.13	76.90	98.36	96.41
BCE	78.13	65.14	76.20	**98.52**	**96.60**
Dice + BCE	78.45	65.63	75.75	98.48	96.35
Dice + BCE + Jaccard	**79.53**	**66.91**	77.05	98.47	96.48

On the other hand, for Attention-UNAS-Net ODE, Dice + BCE + Jaccard gives the best results in terms of DICE and IoU, while BCE gives the best results in terms of SP and ACC. Selecting the appropriate loss function can enhance the segmentation performance between 0.17 and 0.62 points. As a result, the selection of loss function positively impacts segmentation performance. Using a combined loss instead of a single loss function gives optimal results for this study.

5.2.2. Model pruning

Network parameters and number of floating point operations (FLOPs) play a crucial role in determining model performance and inference time. Table 4 demonstrates that networks generated by the proposed search space have less complex structures compared to baseline networks. However, the impact of the n_c hyper-parameter, which

Table 6
The performance comparison for the different number of cells.

	Performance metrics					
	UNAS-Net ODE					
#Cells	FLOPs (G)	DICE	IoU	SNS	SP	ACC
2	46	65.89	51.67	60.10	**98.26**	94.01
3	65	73.03	60.42	70.36	97.94	95.22
4	84	**79.57**	**66.74**	**81.48**	97.81	**96.19**
5	103	72.68	58.94	70.16	98.12	95.31
	Attention-UNAS-Net ODE					
2	91	74.67	61.11	75.70	97.29	95.23
3	133	73.67	60.01	72.97	97.48	95.02
4	175	**79.36**	**66.49**	**78.56**	**97.90**	**96.04**
5	217	74.59	61.16	76.15	97.24	95.12

Table 7
Threshold comparison results.

	Performance metrics				
	UNAS-Net ODE				
Threshold	DICE	IoU	SNS	SP	ACC
0.3	**80.01**	**67.43**	81.29	**97.84**	**96.20**
0.5	79.57	66.74	**81.48**	97.81	96.19
Otsu	**80.01**	67.42	**81.48**	97.81	96.19
	Attention-UNAS-Net ODE				
0.3	79.16	66.23	78.40	**97.92**	**96.04**
0.5	**79.36**	**66.49**	**78.56**	97.90	**96.04**
Otsu	79.19	66.27	78.56	97.90	**96.04**

Table 8
Performance results of proposed methods on Optofil dataset. Red and blue colors indicate the best two values for each measure. U-ODE: UNAS-Net ODE; Att-ODE: Attention-UNAS-Net ODE.

Method	FLOPs (G)	ACC	SNS	SP	F1	IoU
Kus et. al [72]	**4.18**	98.99	78.06	**99.74**	**82.97**	**72.67**
U-ODE	**2.64**	98.78	**83.04**	99.40	79.93	70.15
Att-ODE	5.48	**99.10**	**80.90**	**99.75**	**84.60**	**71.13**

directly affects the complexity of the generated networks, on segmentation performance and complexity should be examined separately. We have gradually increased the complexity of the network by changing the number of cell hyper-parameter. Results presented in Table 6 show that the number of floating point operations changes between 41% and 124% from $n_c = 2$ to $n_c = 5$ for UNAS-Net ODE and change between 46% and 138% for Attention-UNAS-Net ODE. However, it is observed that the segmentation performance also decreases. For complex networks such as Attention-UNAS-Net, n_c selection has a smaller impact on results, whereas for less complex networks such as UNAS-Net, n_c selection has a greater effect. Also, DICE, IoU, SNS metrics are more affected by n_c than SP and ACC metrics. The results indicate that while the number of cell (n_c) has a direct relationship with model complexity, there is no clear correlation between n_c and segmentation performance. In fact, increasing the number of cell value ($n_c = 5$) leads to a degradation of results. Therefore, more than just optimizing the network structure inside the encoder and decoder cell is required, as seen in Table 6. At the same time, the simultaneous optimization of the n_c (micro and macro search spaces) values also positively affects the performance. Tables 3 and 6 also reveal that the optimal result is achieved by selecting $n_c = 4$ using the proposed DE-based heuristics. This shows the effectiveness of the proposed methods.

5.2.3. Threshold comparison

Research indicates that accurately selecting the threshold value has a substantial impact on the detection of both thin and thick vessels [81]. In order to create segmented binary images, a certain threshold value must be applied to the probability maps ($\in (0, 1)$) that are given as the network's output. However, choosing high threshold values causes loss of thin vessel information in these images. Therefore, the threshold value should be chosen carefully so that neither information about thin vessels nor information about thick vessels is lost.

In the ablation study, we employ two fixed threshold values referred to as high and low thresholds, as well as the Otsu threshold method which dynamically calculates the threshold value based on the input image. The high threshold value is set to be 0.5 in accordance with the comparison study shown in Section 5.1. On the other hand, the low threshold value is set as 0.3 in accordance with the recommendation in the literature [81]. The results for performance comparisons with different threshold settings are provided in Table 7.

The results show that choosing the low threshold value for UNAS-Net ODE led to an improvement in all metrics except SNS. For the Attention-UNAS-Net ODE, it has been found that the results are not better than those obtained with the default threshold value of 0.5. Additionally, we visualized the effects of the different threshold values, as depicted in Fig. 7. The results indicate that utilizing a low threshold value yields an output similar to the ground truth image. Conversely, a high threshold value results in the loss of delicate vessel information. However, using a low threshold value can lead to incorrect separation of vessels, as demonstrated by the green rectangle in Fig. 7. As a result, it can be concluded that the threshold value effects the segmentation performance, and utilizing dual-threshold values, as proposed in [81], could enhance the accuracy of the segmented results.

5.2.4. Generalization performance

It is crucial to evaluate the performance of deep learning models on new images and tasks to determine their capability. This is especially important for medical images as they can vary greatly, and a model that works well on one dataset may not perform as well on new data. Our study evaluates the generalization performance of the best models that are searched using ODE (UNAS-Net ODE and Attention-UNAS-Net ODE) on the KUVESG dataset. We have tested these models on two extra datasets: DRIVE [82] and Optofil [72]. The NAS steps are not performed again for these two extra datasets. We train the searched models (best models that are searched using ODE) on these two datasets and evaluate these on the test set. Also, this procedure is named fine-tuning. The Optofil dataset includes a total of 4914 images of cell cultures, which are represented in gray-scale format, and the DRIVE dataset consists of 40 colored retinal blood vessel images.

First, we train the searched models on the Optofil training set and evaluate these on the test set, following the same process described in [72]. However, we recalculate all measures using the best model from [72] to ensure a fair comparison. Average results are presented in Table 8. Attention-UNAS-Net + ODE performs best for ACC, SPE and F1 measures, giving second-best results for SNS and IoU with a minor (1.31 times) complexity increase. Also, UNAS-Net outperforms other methods with 1.58 times fewer FLOPs in terms of SNS and gives highly competitive results for other measures except for F1.

We also compares the searched models with NAS studies on DRIVE dataset in terms of model complexity and segmentation performance. We train the searched models on the DRIVE training set and evaluate these on the test set. The same pre-processing steps are performed like in [81]. The literature only contains a few number of NAS research that were conducted for DRIVE. In their research, Popat et al. [83] and HNAS [64] used various models with different parameter sizes. We compare two of their models, v1 and v2, which have the fewest and most parameters, respectively. Additionally, the Genetic U-Net [84] present two different results, one with the binary field of view (v1) and one without (v2). We compare our results to those of recent studies [64, 83–85] by using the results that are reported in the corresponding papers. Results are shown in Table 9. Attention-UNAS-Net ODE gives best results in all metrics (except F1). Furthermore, it outperforms HNAS

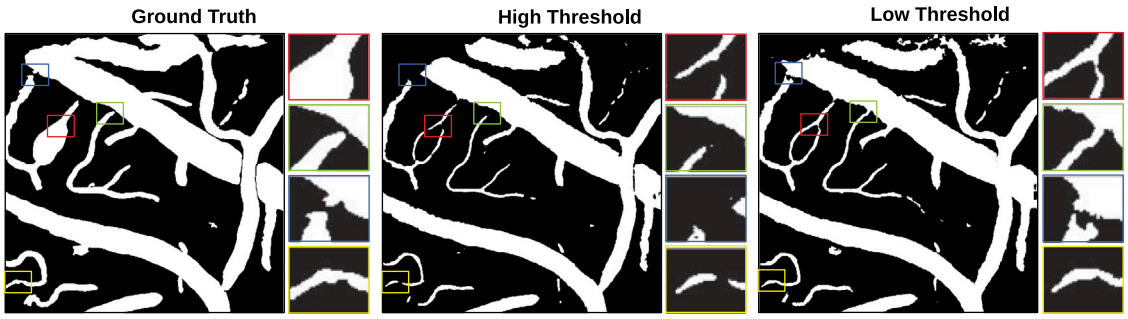

Fig. 7. Graphical representation of the outcomes of low and high threshold analysis on the vesselNN dataset. We magnify the four highlighted rectangle in the images to display the details.

Table 9
Performance results of different NAS studies on DRIVE dataset. Red and blue colors indicate the best two values for each performance measure. U-ODE: UNAS-Net ODE; Att-ODE: Attention-UNAS-Net ODE.

Method	Param (M)	ACC	SNS	SP	F1
Popat et al. v1 [83]	0.18	93.56	59.67	**98.50**	–
Popat et al. v2 [83]	8.10	95.34	75.01	98.31	–
HNAS v1 [64]	157	95.42	77.07	98.10	81.08
HNAS v2 [64]	2.90	95.46	77.44	98.09	81.29
Genetic U-Net v1 [84]	0.27	95.77	83.00	97.58	**83.14**
Genetic U-Net v2 [84]	0.27	**97.07**	83.00	98.43	**83.14**
BTU-Net [85]	16.80	96.89	–	98.44	81.78
U-ODE	6.60	97.00	**83.74**	98.31	**82.04**
Att-ODE	10.32	**97.08**	**89.71**	**98.95**	82.03

v1 and BTU-Net for all metrics despite having 1.62 and 15.21 times fewer parameters, respectively. The UNAS-Net ODE model performs the second-best results for ACC and SP Moreover, it achieves better results with 2.54 times fewer parameters for ACC and F1 than BTU-Net. It outperforms all models in terms of ACC, SNS and F1 (except Genetic U-Net) with fewer parameters. As a result, the searched models can generalize well to new cell segmentation and blood vessel segmentation tasks, even without performing NAS.

6. Conclusion

In this study, we propose two DE-based NAS approaches to optimize U-Net and Attention U-Net architectures. These approaches called as UNAS-Net and Attention UNAS-Net architectures consist of cell-based micro search space and macro search space that combine the advantages of widely used search spaces [19,20]. Proposed approaches find networks that give better segmentation performance for 2D vessel segmentation problem and search micro and macro search spaces simultaneously. We performed the experiments on two datasets including a public dataset *vesseINN* and our own dataset *KUVESG*. The architectures generated outperform the U-Net and Attention U-Net architectures in terms of segmentation performance and computational complexity. UNAS-Net network generated using DE and ODE heuristics outperforms the segmentation performance of U-Net in most of the segmentation measures and has up to 5.19 times fewer FLOPs. Similarly, Attention UNAS-Net gives better results than Attention U-net and generates up to 9.15 times less complex models. Such significant improvements in segmentation performance and computational complexity are important in wide usage of our algorithms by lay users. Additionally, it is known that results of ablation studies can provide valuable insights into the design of improved models and can guide future research in the field. Therefore, we employed ablations studies investigating the effects of loss function, model pruning, and threshold values on the segmentation performance. The ablation results demonstrate that the combination of loss functions improves the segmentation performance. Furthermore, we observe that a low threshold value improves the preservation of thin vessel information and enhances the segmentation performance. Additionally, it is concluded that the number of cells plays a crucial role in determining the model complexity and must be optimized simultaneously with the cell structure. Moreover, the results of generalization experiments show that the proposed methods perform well on various datasets, which suggests that the results are generalizable. However, results also showed that the proposed models are less successful in segmenting thin vessels than thick vessels. Also, the best models searched using ODE on KUVESG only achieve good segmentation results with fine-tuning on unseen test datasets (domains) that are different from train examples. These models need fine-tuning steps to get good generalization performance. In future works, discrete search space can be used in order to optimize UNAS-Net structure. The pre-processing techniques such as contrast enhancement and post-processing techniques like Dilation and Erosion can substantially improve medical image segmentation. These essential steps can augment both accuracy and appearance for the segmented regions. Moreover, the performance of DE can be sensitive to the values of its control parameters. These parameters can affect the algorithm's ability to find good solutions and the time it takes to converge. Therefore, these parameters can be tuned in future works. This study can also be easily expanded to medical image segmentation problems different than brain vasculature analysis.

CRediT authorship contribution statement

Zeki Kuş: Conceptualization of this study, Methodology, Software. **Berna Kiraz:** Conceptualization of this study, Methodology, Software. **Tuğçe Koçak Göksu:** Conceptualization of this study, Methodology, Software. **Musa Aydın:** Conceptualization of this study, Methodology, Software. **Esra Özkan:** Conceptualization of this study, Methodology, Software. **Atay Vural:** Conceptualization of this study, Methodology, Software. **Alper Kiraz:** Conceptualization of this study, Methodology, Software. **Burhanettin Can:** Modification for the final layout.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

A. Kiraz acknowledges partial support from the Turkish Academy of Sciences (TÜBA).

References

[1] M.D. Sweeney, K. Kisler, A. Montagne, A.W. Toga, B.V. Zlokovic, The role of brain vasculature in neurodegenerative disorders, Nature Neurosci. 21 (2018) 1318–1331, http://dx.doi.org/10.1038/s41593-018-0234-x.

[2] S. Akbar, M. Sharif, M.U. Akram, T. Saba, T. Mahmood, M. Kolivand, Automated techniques for blood vessels segmentation through fundus retinal images: A review, Microsc. Res. Tech. 82 (2) (2019) 153–170.

[3] R. Rigamonti, V. Lepetit, Accurate and efficient linear structure segmentation by leveraging ad hoc features with learned filters, in: N. Ayache, H. Delingette, P. Golland, K. Mori (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 189–197.

[4] D. Lesage, E.D. Angelini, I. Bloch, G. Funka-Lea, A review of 3D vessel Lumen segmentation techniques: Models, features and extraction schemes, Med. Image Anal. 13 (6) (2009) 819–845, http://dx.doi.org/10.1016/j.media.2009.07.011, Includes Special Section on Computational Biomechanics for Medicine.

[5] D. Jia, X. Zhuang, Learning-based algorithms for vessel tracking: A review, Comput. Med. Imaging Graph. 89 (2021) 101840, http://dx.doi.org/10.1016/j.compmedimag.2020.101840.

[6] T.A. Soomro, A.J. Afifi, L. Zheng, S. Soomro, J. Gao, O. Hellwich, M. Paul, Deep learning models for retinal blood vessels segmentation: A review, IEEE Access 7 (2019) 71696–71717, http://dx.doi.org/10.1109/ACCESS.2019.2920616.

[7] C. Chen, J.H. Chuah, R. Ali, Y. Wang, Retinal vessel segmentation using deep learning: A review, IEEE Access 9 (2021) 111985–112004, http://dx.doi.org/10.1109/ACCESS.2021.3102176.

[8] X. Liu, L. Song, S. Liu, Y. Zhang, A review of deep-learning-based medical image segmentation methods, Sustainability 13 (3) (2021) http://dx.doi.org/10.3390/su13031224.

[9] N. Siddique, S. Paheding, C.P. Elkin, V. Devabhaktuni, U-net and its variants for medical image segmentation: A review of theory and applications, IEEE Access 9 (2021) 82031–82057, http://dx.doi.org/10.1109/ACCESS.2021.3086020.

[10] L. Yang, H. Wang, Q. Zeng, Y. Liu, G. Bian, A hybrid deep segmentation network for fundus vessels via deep-learning framework, Neurocomputing 448 (2021) 168–178, http://dx.doi.org/10.1016/j.neucom.2021.03.085.

[11] A. Babu, V. Jegathesan, D. David, K.S. Suriya, Retinal blood vessels segmentation using deep learning model-a review, in: 2022 6th International Conference on Devices, Circuits and Systems, ICDCS, 2022, pp. 375–379, http://dx.doi.org/10.1109/ICDCS54290.2022.9780680.

[12] N. Wijethilake, D. Meedeniya, C. Chitraranjan, I. Perera, M. Islam, H. Ren, Glioma survival analysis empowered with data engineering–a survey, IEEE Access 9 (2021) 43168–43191, http://dx.doi.org/10.1109/access.2021.3065965.

[13] I. Rubasinghe, D. Meedeniya, Ultrasound nerve segmentation using deep probabilistic programming, J. ICT Res. Appl. 13 (3) (2019) 241, http://dx.doi.org/10.5614/itbj.ict.res.appl.2019.13.3.5.

[14] V. Rajinikanth, S. Kadry, Y. Nam, Convolutional-neural-network assisted segmentation and SVM classification of brain tumor in clinical MRI slices, Inform. Technol. Control 50 (2) (2021) 342–356.

[15] J. Ramya, B.U. Maheswari, M.P. Rajakumar, R. Sonia, Alzheimer's disease segmentation and classification on MRI brain images using enhanced expectation maximization adaptive histogram (EEM-AH) and machine learning, Inform. Technol. Control 51 (4) (2022) 786–800.

[16] M.A. Khan, A. Khan, M. Alhaisoni, A. Alqahtani, S. Alsubai, M. Alharbi, N.A. Malik, R. Damaševičius, Multimodal brain tumor detection and classification using deep saliency map and improved dragonfly optimization algorithm, Int. J. Imaging Syst. Technol. 33 (2) (2022) 572–587.

[17] S.Z. Kurdi, M.H. Ali, M.M. Jaber, T. Saba, A. Rehman, R. Damaševičius, Brain tumor classification using meta-heuristic optimized convolutional neural networks, J. Pers. Med. 13 (2) (2023) 181.

[18] F. Hutter, L. Kotthoff, J. Vanschoren, Automated Machine Learning: Methods, Systems, Challenges, Springer Nature, 2019.

[19] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, F. Hutter, NAS-bench-101: Towards reproducible neural architecture search, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol.97, PMLR, 2019, pp. 7105–7114.

[20] H. Liu, K. Simonyan, Y. Yang, DARTS: Differentiable architecture search, 2018, http://dx.doi.org/10.48550/ARXIV.1806.09055, arXiv.

[21] P. Teikari, M.A. Santos, C. Poon, K. Hynynen, Deep learning convolutional networks for multiphoton microscopy vasculature segmentation, 2016, ArXiv, arXiv:1606.02382.

[22] S. Moccia, E. De Momi, S. El Hadji, L.S. Mattos, Blood vessel segmentation algorithms — Review of methods, datasets and evaluation metrics, Comput. Methods Programs Biomed. 158 (2018) 71–91, http://dx.doi.org/10.1016/j.cmpb.2018.02.001.

[23] P. Liskowski, K. Krawiec, Segmenting retinal blood vessels with deep neural networks, IEEE Trans. Med. Imaging 35 (11) (2016) 2369–2380, http://dx.doi.org/10.1109/TMI.2016.2546227.

[24] Y. Guo, Ü. Budak, A. Şengür, A novel retinal vessel detection approach based on multiple deep convolution neural networks, Comput. Methods Programs Biomed. 167 (2018) 43–48, http://dx.doi.org/10.1016/j.cmpb.2018.10.021.

[25] S. Feng, Z. Zhuo, D. Pan, Q. Tian, CcNet: A cross-connected convolutional network for segmenting retinal vessels using multi-scale features, Neurocomputing 392 (2020) 268–276, http://dx.doi.org/10.1016/j.neucom.2018.10.098.

[26] I. Atli, O.S. Gedik, Sine-net: A fully convolutional deep learning architecture for retinal blood vessel segmentation, Eng. Sci. Technol. Int. J. 24 (2) (2021) 271–283, http://dx.doi.org/10.1016/j.jestch.2020.07.008.

[27] S. Sethuraman, V.P. Gopi, Staircase-net: A deep learning based architecture for retinal blood vessel segmentation, Sādhanā 47 (2022) 191, http://dx.doi.org/10.1007/s12046-022-01936-w.

[28] M. Haft-Javaherian, L. Fang, V. Muse, C.B. Schaffer, N. Nishimura, M.R. Sabuncu, Deep convolutional neural networks for segmenting 3D in vivo multiphoton images of vasculature in alzheimer disease mouse models, PLoS One 14 (3) (2019) e0213539, http://dx.doi.org/10.1371/journal.pone.0213539.

[29] W. Tahir, S. Kura, J. Zhu, X. Cheng, R. Damseh, F. Tadesse, A. Seibel, B.S. Lee, F. Lesage, S. Sakadžic, et al., Anatomical modeling of brain vasculature in two-photon microscopy by generalizable deep learning, BME Front. 2020 (2020).

[30] C. Kirst, S. Skriabine, A. Vieites-Prado, T. Topilko, P. Bertin, G. Gerschenfeld, F. Verny, P. Topilko, N. Michalski, M. Tessier-Lavigne, N. Renier, Mapping the fine-scale organization and plasticity of the brain vasculature, Cell 180 (4) (2020) 780–795.e25, http://dx.doi.org/10.1016/j.cell.2020.01.028.

[31] R. Damseh, P. Pouliot, L. Gagnon, S. Sakadzic, D. Boas, F. Cheriet, F. Lesage, Automatic graph-based modeling of brain microvessels captured with two-photon microscopy, IEEE J. Biomed. Health Inf. 23 (6) (2019) 2551–2562, http://dx.doi.org/10.1109/JBHI.2018.2884678.

[32] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[33] M. Livne, J. Rieger, O.U. Aydin, A.A. Taha, E.M. Akay, T. Kossen, J. Sobesky, J.D. Kelleher, K. Hildebrand, D. Frey, et al., A U-net deep learning framework for high performance vessel segmentation in patients with cerebrovascular disease, Front. Neurosci. 13 (2019) 97.

[34] Y. Zhang, A.C.S. Chung, Deep supervision with additional labels for retinal vessel segmentation task, in: A.F. Frangi, J.A. Schnabel, C. Davatzikos, C. Alberola-López, G. Fichtinger (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, Springer International Publishing, Cham, 2018, pp. 83–91.

[35] Z. Zhang, C. Wu, S. Coleman, D. Kerr, DENSE-INception U-net for medical image segmentation, Comput. Methods Programs Biomed. 192 (2020) 105395, http://dx.doi.org/10.1016/j.cmpb.2020.105395.

[36] Y.M. Kassim, O.V. Glinskii, V.V. Glinsky, V.H. Huxley, G. Guidoboni, K. Palaniappan, Deep U-net regression and hand-crafted feature fusion for accurate blood vessel segmentation, in: 2019 IEEE International Conference on Image Processing, ICIP, 2019, pp. 1445–1449, http://dx.doi.org/10.1109/ICIP.2019.8803084.

[37] T. Koçak, M. Aydın, B. Kiraz, Empirical comparison of deep neural networks for brain vessel segmentation, in: 2021 6th International Conference on Computer Science and Engineering, UBMK, 2021, pp. 751–756, http://dx.doi.org/10.1109/UBMK52708.2021.9559015.

[38] X. Xiao, S. Lian, Z. Luo, S. Li, Weighted res-UNet for high-quality retina vessel segmentation, in: 2018 9th International Conference on Information Technology in Medicine and Education, ITME, 2018, pp. 327–331, http://dx.doi.org/10.1109/ITME.2018.00080.

[39] C. Wu, Y. Zou, J. Zhan, DA-U-Net: Densely connected convolutional networks and decoder with attention gate for retinal vessel segmentation, IOP Conf. Ser.: Mater. Sci. Eng. 533 (1) (2019) 012053, http://dx.doi.org/10.1088/1757-899x/533/1/012053.

[40] Y. Lv, H. Ma, J. Li, S. Liu, Attention guided U-net with atrous convolution for accurate retinal vessels segmentation, IEEE Access 8 (2020) 32826–32839, http://dx.doi.org/10.1109/ACCESS.2020.2974027.

[41] C. Guo, M. Szemenyei, Y. Yi, W. Wang, B. Chen, C. Fan, SA-UNet: Spatial attention U-net for retinal vessel segmentation, in: 2020 25th International Conference on Pattern Recognition, ICPR, 2021, pp. 1236–1242, http://dx.doi.org/10.1109/ICPR48806.2021.9413346.

[42] S. Zhao, T. Liu, B. Liu, K. Ruan, Attention residual convolution neural network based on U-net (AttentionResU-net) for retina vessel segmentation, IOP Conf. Ser.: Earth Environ. Sci. 440 (3) (2020) 032138, http://dx.doi.org/10.1088/1755-1315/440/3/032138.

[43] K. Li, X. Qi, Y. Luo, Z. Yao, X. Zhou, M. Sun, Accurate retinal vessel segmentation in color fundus images via fully attention-based networks, IEEE J. Biomed. Health Inf. 25 (6) (2021) 2071–2081, http://dx.doi.org/10.1109/JBHI.2020.3028180.

[44] B. Baker, O. Gupta, N. Naik, R. Raskar, Designing neural network architectures using reinforcement learning, 2016, CoRR abs/1611.02167, arXiv:1611.02167.

[45] B. Zoph, Q.V. Le, Neural architecture search with reinforcement learning, 2016, CoRR abs/1611.01578, arXiv:1611.01578.

[46] W. Grathwohl, E. Creager, S.K.S. Ghasemipour, R. Zemel, Gradient-based optimization of neural network architecture, 2018.

[47] R. Shin, C. Packer, D. Song, Differentiable neural network architecture search, 2018.
[48] M. Baldeon Calisto, S.K. Lai-Yuen, EMONAS-net: Efficient multiobjective neural architecture search using surrogate-assisted evolutionary algorithm for 3D medical image segmentation, Artif. Intell. Med. 119 (2021) 102154.
[49] Y. Liu, Y. Sun, B. Xue, M. Zhang, G.G. Yen, K.C. Tan, A survey on evolutionary neural architecture search, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–21, http://dx.doi.org/10.1109/tnnls.2021.3100554.
[50] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8697–8710, http://dx.doi.org/10.1109/CVPR.2018.00907.
[51] X. Dong, Y. Yang, NAS-bench-201: Extending the scope of reproducible neural architecture search, 2020, CoRR abs/2001.00326, arXiv:2001.00326.
[52] J. Siems, L. Zimmer, A. Zela, J. Lukasik, M. Keuper, F. Hutter, NAS-bench-301 and the case for surrogate benchmarks for neural architecture search, 2020, CoRR abs/2008.09777, arXiv:2008.09777.
[53] E. Real, A. Aggarwal, Y. Huang, Q.V. Le, Regularized evolution for image classifier architecture search, in: Proceedings of the Aaai Conference on Artificial Intelligence, Vol. 33, 2019, pp. 4780–4789.
[54] C. Wei, C. Niu, Y. Tang, Y. Wang, H. Hu, J. Liang, NPENAS: Neural predictor guided evolution for neural architecture search, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–15, http://dx.doi.org/10.1109/TNNLS.2022.3151160.
[55] N. Qiang, B. Ge, Q. Dong, F. Ge, T. Liu, Neural architecture search for optimizing deep belief network models of fMRI data, in: Q. Li, R. Leahy, B. Dong, X. Li (Eds.), Multiscale Multimodal Medical Imaging, Springer International Publishing, Cham, 2020, pp. 26–34.
[56] N.H. Awad, N. Mallik, F. Hutter, Differential evolution for neural architecture search, 2020, CoRR abs/2012.06400, arXiv:2012.06400.
[57] Z. Zhu, C. Liu, D. Yang, A. Yuille, D. Xu, V-NAS: Neural architecture search for volumetric medical image segmentation, in: 2019 International Conference on 3D Vision, 3DV, 2019, pp. 240–248, http://dx.doi.org/10.1109/3DV.2019.00035.
[58] W. Bae, S. Lee, Y. Lee, B. Park, M. Chung, K.-H. Jung, Resource optimized neural architecture search for 3D medical image segmentation, in: D. Shen, T. Liu, T.M. Peters, L.H. Staib, C. Essert, S. Zhou, P.-T. Yap, A. Khan (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2019, Springer International Publishing, Cham, 2019, pp. 228–236.
[59] A. Mortazi, U. Bagci, Automatically designing CNN architectures for medical image segmentation, in: Y. Shi, H.-I. Suk, M. Liu (Eds.), Machine Learning in Medical Imaging, Springer International Publishing, Cham, 2018, pp. 98–106.
[60] Y. Weng, T. Zhou, Y. Li, X. Qiu, NAS-Unet: Neural architecture search for medical image segmentation, IEEE Access 7 (2019) 44247–44257, http://dx.doi.org/10.1109/ACCESS.2019.2908991.
[61] Z. Xu, S. Zuo, E.Y. Lam, B. Lee, N. Chen, AutoSegNet: An automated neural network for image segmentation, IEEE Access 8 (2020) 92452–92461, http://dx.doi.org/10.1109/ACCESS.2020.2995367.
[62] Z. Fan, J. Wei, G. Zhu, J. Mo, W. Li, Evolutionary neural architecture search for retinal vessel segmentation, 2020, http://dx.doi.org/10.48550/ARXIV.2001.06678, arXiv.
[63] J.-D. Sun, C. Yao, J. Liu, W. Liu, Z.-K. Yu, GNAS-U2Net: A new optic cup and optic disc segmentation architecture with genetic neural architecture search, IEEE Signal Process. Lett. 29 (2022) 697–701, http://dx.doi.org/10.1109/LSP.2022.3151549.
[64] Y. Houreh, M. Mahdinejad, E. Naredo, D. Dias, C. Ryan, HNAS: Hyper neural architecture search for image segmentation, in: Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, SciTePress, INSTICC, 2021, pp. 246–256, http://dx.doi.org/10.5220/0010260902460256.
[65] T.R. Patel, N. Paliwal, P. Jaiswal, M. Waqas, M. Mokin, A.H. Siddiqui, H. Meng, R. Rai, V. Tutino, Multi-resolution CNN for brain vessel segmentation from cerebrovascular images of intracranial aneurysm: a comparison of U-Net and DeepMedic, in: H.K. Hahn, M.A. Mazurowski (Eds.), Medical Imaging 2020: Computer-Aided Diagnosis, Vol. 11314, SPIE, International Society for Optics and Photonics, 2020, p. 113142W, http://dx.doi.org/10.1117/12.2549761.
[66] O. Oktay, J. Schlemper, L.L. Folgoc, M.C.H. Lee, M.P. Heinrich, K. Misawa, K. Mori, S.G. McDonagh, N.Y. Hammerla, B. Kainz, B. Glocker, D. Rueckert, Attention U-net: Learning where to look for the pancreas, 2018, CoRR abs/1804.03999, arXiv:1804.03999.
[67] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[68] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (1) (2008) 64–79, http://dx.doi.org/10.1109/TEVC.2007.894200.
[69] S.-J. de-la Cruz-Martínez, E. Mezura-Montes, Boundary constraint-handling methods in differential evolution for mechanical design optimization, in: 2020 IEEE Congress on Evolutionary Computation, CEC, 2020, pp. 1–8, http://dx.doi.org/10.1109/CEC48606.2020.9185495.
[70] R. Mallipeddi, P. Suganthan, Q. Pan, M. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2) (2011) 1679–1696, http://dx.doi.org/10.1016/j.asoc.2010.04.024, The Impact of Soft Computing for the Progress of Artificial Intelligence.
[71] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417, http://dx.doi.org/10.1109/TEVC.2008.927706.
[72] Z. Kuş, M. Aydın, B. Kiraz, B. Can, Neural architecture search using metaheuristics for automated cell segmentation, in: L. Di Gaspero, P. Festa, A. Nakib, M. Pavone (Eds.), Metaheuristics, Springer International Publishing, Cham, 2023, pp. 158–171.
[73] E. Özkan, Y. Çetin-Taş, E. Şekerdağ, B. Yiğit, N. Shomalizadeh, S. Sapancı, C. Ozler, J. Kesibi, A.B. Kızılırmak, M. Pekmez, H. Yapıcı-Eser, M. Zeybel, S. Karahüseyinoğlu, Y. Gürsoy-Özdemir, Hyperglycemia with or without insulin resistance triggers different structural changes in brain microcirculation and perivascular matrix, Metab. Brain Dis. (2022) http://dx.doi.org/10.1007/s11011-022-01100-7.
[74] E. Özkan, A. Vural, A. Kiraz, B. Kiraz, M. Aydı n, B. Can, Z. Kuş, T. Göksu, KUVESG, Zenodo, 2022, http://dx.doi.org/10.5281/ZENODO.7383294, URL https://zenodo.org/record/7383294.
[75] A.A. Taha, A. Hanbury, Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool, BMC Med. Imaging 15 (1) (2015) http://dx.doi.org/10.1186/s12880-015-0068-x.
[76] O.U. Aydin, A.A. Taha, A. Hilbert, A.A. Khalil, I. Galinovic, J.B. Fiebach, D. Frey, V.I. Madai, An evaluation of performance measures for arterial brain vessel segmentation, BMC Med. Imaging 21 (1) (2021) http://dx.doi.org/10.1186/s12880-021-00644-x.
[77] L.R. Dice, Measures of the amount of ecologic association between species, Ecology 26 (3) (1945) 297–302.
[78] K.H. Zou, S.K. Warfield, A. Bharatha, C.M. Tempany, M.R. Kaus, S.J. Haker, W.M. Wells, F.A. Jolesz, R. Kikinis, Statistical validation of image segmentation quality based on a spatial overlap index1: Scientific reports, Acad. Radiol. 11 (2) (2004) 178–189, http://dx.doi.org/10.1016/S1076-6332(03)00671-8.
[79] V. Yeghiazaryan, I. Voiculescu, Family of boundary overlap metrics for the evaluation of medical image segmentation, J. Med. Imaging 5 (01) (2018) 1, http://dx.doi.org/10.1117/1.jmi.5.1.015006.
[80] Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, W. Banzhaf, NSGA-net: Neural architecture search using multi-objective genetic algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 419–427, http://dx.doi.org/10.1145/3321707.3321729.
[81] W. Liu, H. Yang, T. Tian, Z. Cao, X. Pan, W. Xu, Y. Jin, F. Gao, Full-resolution network and dual-threshold iteration for retinal vessel and coronary angiograph segmentation, IEEE J. Biomed. Health Inf. 26 (9) (2022) 4623–4634, http://dx.doi.org/10.1109/JBHI.2022.3188710.
[82] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, B. van Ginneken, Ridge-based vessel segmentation in color images of the retina, IEEE Trans. Med. Imaging 23 (4) (2004) 501–509, http://dx.doi.org/10.1109/TMI.2004.825627.
[83] V. Popat, M. Mahdinejad, O.D. Cedeño, E. Naredo, C. Ryan, GA-based U-net architecture optimization applied to retina blood vessel segmentation, in: IJCCI, 2020, pp. 192–199.
[84] J. Wei, G. Zhu, Z. Fan, J. Liu, Y. Rong, J. Mo, W. Li, X. Chen, Genetic U-net: Automatically designed deep networks for retinal vessel segmentation using a genetic algorithm, IEEE Trans. Med. Imaging 41 (2) (2022) 292–307, http://dx.doi.org/10.1109/TMI.2021.3111679.
[85] C. Rajesh, R. Sadam, S. Kumar, An evolutionary U-shaped network for retinal vessel segmentation using binary teaching–learning-based optimization, Biomed. Signal Process. Control 83 (2023) 104669, http://dx.doi.org/10.1016/j.bspc.2023.104669, URL https://www.sciencedirect.com/science/article/pii/S1746809423001027.