Evolutionary Architecture Optimization for Retinal Vessel Segmentation

Zeki Kuş ⓘ and Berna Kiraz ⓘ

Abstract—**Retinal vessel segmentation (RVS) is crucial in medical image analysis as it helps identify and monitor retinal diseases. Deep learning approaches have shown promising results for RVS, but designing optimal neural network architecture is challenging and time-consuming. Neural architecture search (NAS) is a recent technique that automates the design of neural network architectures within a predefined search space. This study proposes a new NAS method for U-shaped networks, MedUNAS, that discovers deep neural networks with high segmentation performance and lower inference time for RVS problem. We perform opposition-based differential evolution (ODE) and genetic algorithm (GA) to search for the best network structure and compare discrete and continuous encoding strategies on the proposed search space. To the best of our knowledge, this is the first NAS study that performs ODE for RVS problems. The results show that the MedUNAS ODE and GA yield the best and second-best results regarding segmentation performance with less than 50% of the parameters of U-shaped state-of-the-art methods on most of the compared datasets. In addition, the proposed methods outperform the baseline U-Net on four datasets with networks with up to 15 times fewer parameters. Furthermore, ablation studies are performed to evaluate the generalizability of the generated networks to medical image segmentation problems that differ from the trained domain, revealing that such networks can be effectively adapted to new tasks with fine-tuning. The MedUNAS can be a valuable tool for automated and efficient RVS in clinical practice.**

Index Terms—**Neural architecture search, genetic algorithm, opposition-based differential evolution, retinal vessel segmentation.**

I. INTRODUCTION

RETINAL vessel segmentation (RVS) is an important task in medical image analysis for detecting and managing retinal vascular diseases, including diabetic retinopathy and age-related macular degeneration [1]. Accurate segmentation

of retinal blood vessels (RBS) provides clinicians with valuable information, such as vessel morphology (shape and size) and tortuosity (degree of curvature), which aid in detecting and managing these diseases [2]. However, it is challenging due to the complex structure and varying appearance of RBS [3]. Hence, developing robust and efficient algorithms for this task has become a topic of active research in medical imaging.

Computer-aided diagnosis (CAD) [4] models that utilize deep learning can effectively detect diseases such as COVID-19 and retinal vascular diseases. There are some recent studies showing improvements in the accuracy and stability of computer-aided diagnosis models for COVID-19 [5], [6]. However, automated RVS is challenging due to the complex structure of vascular trees, pathological issues, and irregular lighting [7]. U-Net and U-shaped networks, special CNN models proposed for biomedical image segmentation have become the preferred models as the baseline for RVS [8], [9], [10], [11], [12], but they have some drawbacks. These networks are hand-designed architectures that do not detect thin vessels well in dense complex structures because of are used generally fixed parameters or need to be determined manually [13]. They also have many hyper-parameters that need to be manually determined. Therefore, more flexible structures are required to extract complex vessels with fewer parameters.

Deep learning methods, especially the U-shaped model, have shown great success in RVS [14]. Some recent NAS studies for the task addressed in this study have been presented in the literature. Researchers have proposed various techniques to enhance the segmentation performance of U-Net models for RVS. Zhou et al. [15] introduced Study Group Learning (SGL), significantly improving segmentation performance in datasets with noisy training labels. Kamran et al. [16] presented RV-GAN, a generative adversarial network using two generators and two multi-scale autoencoding discriminators to extract vessel information, achieving better results than other architectures. FR-UNet [8] utilizes multi-resolution convolution, feature aggregation, and modified residual blocks to obtain accurate pixel-level representations in medical images. TP-Net [9] employs a two-path network with a multi-scale feature aggregation module to outperform existing methods.

Evolutionary algorithms have also been utilized to search for optimal U-shaped network architectures for RVS. An evolutionary neural architecture search (NAS) method [17] with a specific search space yielded better segmentation performance with fewer parameters. Popat et al. [18] proposed manually and automatically designed U-Net architectures, achieving

competitive results with smaller sizes using a standard GA. Hyper Neural Architecture Search [19] improved U-Net performance with a ConvLSTM cell, residual connections, and a binary strategy for choosing batch normalization, finding competitive architectures with smaller sizes and similar accuracy to state-of-the-art methods. Wei et al. [7] proposed Genetic U-Net, which identifies higher-performing structures using an improved GA and a smaller but flexible search space, outperforming other models with significantly fewer parameters. BTLBO evolutionary model [20] optimizes U-shaped architecture automatically using a flexible search space, an attention mechanism, and a cache system to accelerate the evolutionary steps, showing superior performance to state-of-the-art models on publicly available RVS datasets.

This study presents a new NAS method for U-shaped networks, namely MedUNAS. We propose a new cell-based search space based on the search space presented in [21] and [22]. Opposition-based differential evolution (ODE) and genetic algorithm (GA) are performed on this search space to search for the best network structure. We adapt the proposed GA and discrete encoding strategy in [21] for RVS and also extend the encoding strategy for continuous search space with ODE. Unlike the discrete encoding presented in [21], we change the cell structure to reduce complexity, and local search is not included in the search. On the other hand, research in the field of NAS shows that continuous search space-based studies give better results [22], [23]. Therefore, we modify the proposed encoding scheme to work on continuous search space and apply the ODE to compare discrete and continuous encoding strategies. To the best of our knowledge, this is the first NAS study that applies the ODE algorithm for the RVS.

The proposed methods are evaluated on four publicly available RVS datasets, namely DRIVE, CHASE_DB1, DCA1, and CHUAC. Results are compared with recent state-of-the-art methods and NAS studies in terms of segmentation performance and model complexity. Based on the experimental results, the proposed methods generate the least complex models (except CHASE_DB1) than recent state-of-the-art studies and still achieve highly competitive results. We also perform ablation studies to evaluate the effect of the loss functions and examine the model's generalization performance for different domains and its performance on cross-datasets.

The main contributions of this study are listed as follows:

1) We propose a new NAS method for the U-shaped networks. To the best of our knowledge, this is the first NAS study that performs the opposition-based differential evolution algorithm for the RVS problem.

2) We compare discrete and continuous encoding strategies on the proposed search space and investigate the search performance of ODE and GA in terms of exploitation and exploration.

3) The proposed methods discover the least complex networks (less than 50% of the parameters of state-of-the-art methods) than state-of-the-art methods while achieving highly competitive results on four publicly available RVS datasets.

Section II explains the proposed methods, search space, encoding strategy and search methods. Section III introduces
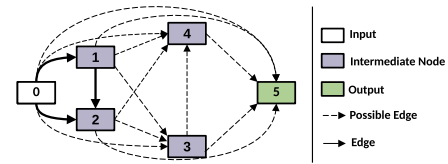

Fig. 1. Summary of cell-based micro search space and the overall architecture of encoder and decoder cells. The cell is represented as a DAG with one input node, 4 intermediate nodes, and one output node.

the experimental design, datasets, performance measures and implementation details. The results and ablation studies are presented and discussed in Section IV. Finally, the conclusion and future works are presented in Section V.

II. MEDUNAS: PROPOSED EVOLUTIONARY NAS FOR U-NET

This study proposes MedUNAS, an evolutionary approach to finding the best U-Net architecture for RVS. MedUNAS is a U-shaped architecture that employs a cell-based micro search space, which determines the structure of the cell, and a macro search space to determine the total number of cells and feature maps in the first cell. Besides, a genetic algorithm (GA) and an opposition-based differential algorithm (ODE) that shows superior performance on NAS [7], [24] are employed in MedUNAS. The following sections provide in-depth information on MedUNAS.

A. Search Space

MedUNAS consists of encoder and decoder cells which perform down-sampling and up-sampling, respectively. The encoder cells extract features through convolution operations and reduce image size using max-pooling, while the decoder cells reconstruct the image using the extracted features and increase image size through transpose convolution. Finally, a 2D convolution and the sigmoid function are applied to generate the segmented image. Fig. 1 in the Supplementary Information illustrates the overall architecture of MedUNAS.

In this study, the cell-based micro-search method is employed to define the structure of the encoder and decoder cells, which are represented as directed acyclic graphs (DAGs). We consider the cell-based micro-search based on the method presented in [21], where edges represent the operations and nodes indicate the outputs obtained from different operations. In [21], each cell contains 2 input nodes, 4 intermediate nodes, and one output node. Each intermediate node receives two information flows from its predecessor nodes, processes the information and sends it to its successor nodes and the output node. Unlike the search space in [21], in this study, each cell has only one input node and there is no restriction for all intermediate to send the information flows to the output node. The illustration of a cell is provided in Fig. 1. Each cell consists of a set of nodes (vertices) ($V = v_0, v_1, \ldots, v_5$), edges ($E$) and operations ($O$). These cells have one input, one output and 4 intermediate nodes ($|V| = 4$), which combine the outputs obtained from different operations with the sum operation. Each intermediate node ($v \in V \setminus \{v_1\}$) has 2 in-edges that can come from the previous nodes. Cell structure has two different types of edges: fixed edges, which are shown
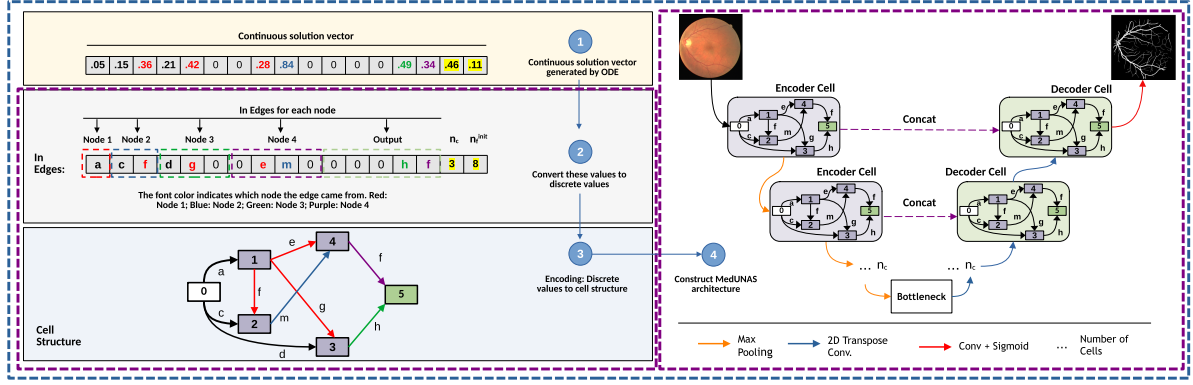
Fig. 2. Overview of the encoding steps for MedUNAS architecture. The purple dashed line shows GA steps, and the blue dashed line shows ODE steps. Step 1 (mapping continuous space to discrete space) only performs for ODE.

with solid edges in Fig. 1, and possible edges (dashed lines in Fig. 1). These types of edges form a set of edges (E), and each edge ($e \in E$) represents an operation applied to an intermediate node. The operations to be applied are selected from a set of operations (O). There are 15 possible operations to be selected for each edge (Table I in the Supplementary Information).

Also, MedUNAS architecture has two additional hyperparameters: The number of cells and the number of feature maps in the first cell that defines the macro search space. The architecture has an equal number of encoder and decoder cells, and the total number of cells (N_c) is calculated by adding the number of the encoder, decoder and bottleneck cells ($N_c = 2n_c + 1$). The number of the encoder and decoder cells ($n_c \in \{2, 3, 4, 5\}$) is a decision variable. The number of feature maps in the first cell (n_f^{init}), the second hyper-parameter in the macro search space, affects the network's performance, and the number of feature maps is doubled after each max-pooling operation. This hyper-parameter ($n_f^{init} \in \{8, 16, 32\}$) specifies how many feature maps will be generated by the end of the initial encoder cell's operations.

B. Encoding

In this study, we use the encoding proposed in [21] for the GA and modify it for the ODE. A discrete-valued vector for GA and a real-valued vector ($\in [0, 1]$) for ODE with 17 variables indicate a candidate network. The first 15 variables in this vector represent the flow between each node and the selected operations for each edge in a DAG. The last two variables indicate the number of encoder/decoder cells (n_c) and the number of feature maps in the first cell (n_f^{init}).

The encoding mechanism for a given cell structure is illustrated in Fig. 2 for both GA and ODE. First, we create a DAG with the corresponding discrete variables for the cell structure. Then, the MedUNAS architecture is constructed with this cell structure and two hyper-parameters. Besides these steps, in ODE, the continuous values in the candidate solution are converted to discrete variables.

Fig. 2 also shows an exemplary cell structure with 4 intermediate nodes. In this structure, v_1 receives information from v_0 (input node) and applies operation a (1×1 Convolution, see

Table I in the Supplementary Information). Then, the output of this operation is stored in v_1. v_2 has only two possible in-edge that can come from v_0 and v_1. Therefore, it receives information from v_0 and v_1. Then, operations c and f are applied to these nodes, respectively. After these operations, the outputs are combined with the sum operation in v_2. v_3 has three possible in-edge that can come from: v_0, v_1 and v_2. At this step, two nodes are selected randomly. v_3 receives information from v_0 and v_1, then applies operations d and g to these nodes, respectively. Similarly, the outputs of these operations are combined in v_3. The same steps are performed for v_4 and v_5.

In [21], the authors propose a solution encoding that uses both alphabet and "0" strings to represent operation and connection information, rather than the commonly used binary string ("01") encoding that only represents the neural node connections. We consider this solution encoding method for GA and modified it for the ODE. Fig. 2 shows an example cell structure formed using "a-cf-dg0-0em0-000hf" encoding. As illustrated in Fig. 2, the first bit of solution encoding indicates the in-edge information coming to v_1 and the selected operation for this in-edge. Similarly, the next two bits specify the in-edges information coming to v_2 and selected operations, and so on. Finally, the last five bits show the in-edges and selected operation information for v_5. For example, the last five bits ("000hf") denote the five possible in-edge connections (v_0, v_1, v_2, v_3, v_4) for v_5 and selected operations for these possible in-edges. According to the last five bits, there is no connection between the following nodes: v_0 - v_5; v_1 - v_5; v_2 - v_5 (see in Fig. 2). Furthermore, v_5 receives information from v_3 and applies operation h to v_3. Similarly, v_5 receives information from v_4 and performs operation f on v_4.

The solution encoding proposed in [21] cannot be used for ODE. It utilizes a real-valued vector to perform mutation and crossover operations. Therefore, we incorporated extra two-step into the solution encoding steps: (1) generate a real-valued solution vector ($\in [0, 1]$) and (2) perform mapping on this vector to a discrete solution vector (see in Fig. 2). We use the mapping scheme proposed in [22], [23]. We summarize these steps as follows: **(1)** Firstly, a solution vector containing 17 elements filled with zero values is created. The first 15 elements indicate the connections and operations for each possible edge. These elements are replaced with randomly generated real values

considering the maximum 2 in-edge constraints and the number of possible connections for each node. **(2)** The number of possible in-edges for each node is shown with different colored dashed lines in Fig. 2 (Step 2). For example, the red dashed line shows possible in-edges for v_1, while the purple dashed line shows 4 possible in-edges for v_4. v_1 has one possible in-edge. Therefore, only one real-valued number is generated randomly (0.05 in Fig. 2). As there are 15 possible operations for this in-edge, the values between 0 and 1 are partitioned into 15 equivalent bins, each corresponding to a particular operation. The first bin $[0, 0.067)$ indicates the operation $a : 1 \times 1$ conv, while the second bin $[0.067, 0.133)$ corresponds to $b : 3 \times 3$ conv, and so on. Since 0.05 corresponds to the first bin operation a selected for the first element. Similarly, elements between 7 and 10 indicate the possible in-edges for v_4. v_4 has four possible in-edge, and two of them must be selected randomly (two in-edge constraint). Therefore, two real-valued numbers are generated randomly (0.28 and 0.84 in Fig. 2). Then, corresponding operations (e and m) are selected for these two in-edges, respectively. **(3)** The last two elements specify the n_c and n_f^{init} decision variables. There are four values for n_c and three values for n_f^{init}, so the interval $[0, 1]$ is partitioned into four and three bins of equal size, respectively. Then, the corresponding values are selected for these decision variables, as shown in Fig. 2 (yellow highlighted). **(4)** As a result of the first three steps, we construct the MedUNAS architecture.

C. Search Methods

1) Genetic Algorithm: The GA is a type of optimization algorithm that is inspired by the process of natural selection. The steps involved in a GA are as follows: **(1) Initialization:** The initial population is generated randomly using the encoding method. There are 20 individuals in the initial population, each of which represents a cell structure and two hyper-parameters for an architecture. **(2) Fitness Evaluation:** Each individual in the population is evaluated based on the segmentation performance (dice score) of the model it represents. **(3) Selection:** Two parents (P_1 and P_2) are selected using the roulette wheel selection. **(4) Crossover:** We use the uniform crossover operation presented in [21]. Since the candidate solution has 5 blocks (red, blue, green, purple, and light green dashed lines in Fig. 2), each of which denotes the connections and operations for an intermediate node, the uniform crossover is performed on these blocks. Each block is swapped according to a predefined probability α to create two new children (C_1 and C_2). For example: Assume that we have two parent solution encoding: "a-cf-dg0-0em0-000hf" and "b-ed-d0a-e00h-0m0n0". Let the crossover probabilities for 5 blocks be $0.3, 0.6, 0.4, 0.2$ and 0.7. Since the crossover probabilities of blocks 2 and block 5 are greater than $\alpha = 0.5$, these blocks are swapped, and the following children are created: "a-ed-dg0-0em0-0m0n0" and "b-cf-d0a-e00h-000hf" **(5) Mutation:** C_1 and C_2 are mutated using the proposed mutation operations in [21]. As with crossover, mutation operations are applied to blocks that indicate intermediate nodes with a predefined probability ($\beta = 0.5$). Two mutation operators are used in this study, which are randomly selected to perform when

the mutation occurs. The first operator changes the operation while keeping the in-edges unchanged. The second operator fixed the selected operator but altered the in-edge connection with a different, previously unconnected intermediate node. For example, we have "a-ed-dg0-0em0-0m0n0" encoding with the following probabilities: $0.2, 0.4, 0.8, 0.3, 0.4$, and the first mutation operation is selected. Considering these probabilities, the mutation is applied only third section and the randomly selected operation changes with another operation: "a-ed-dg0-0em0-0m0n0" → "a-ed-di0-0em0-0m0n0". Similarly, if the second mutation operation is applied to the third section, as a result the following encoding is created: "a-ed-dg0-0em0-0m0n0" → "a-ed-d0g-0em0-0m0n0". The applied mutation operator type is selected randomly. **(6) Replacement:** We apply elitism of 1 and generational replacement [25].

2) Opposition-based Differential Evolution: Opposition-based Differential Evolution (ODE) is a variant of Differential Evolution (DE) algorithm that enhances its search ability by using the concept of opposition-based learning. In this study, we use DE/rand/1/bin variant for ODE [26]. The opposition-based initialization has 4 stages: The first step is to randomly create the initial population (P_0). The second step is to calculate the opposite of each solution in the initial population, which generates the opposite population (OP_0). The opposite solution for a member of the initial population is computed as:

$$OP_{j,0}^x = l^x + u^x - P_{j,0}^x,$$
$$for \, j = \{1, \ldots, NP\}; x = \{1, \ldots, D\} \quad (1)$$

where $OP_{j,0}^x$ and $P_{j,0}^x$ indicate the xth decision variable of the jth solution in opposition of the initial population and the initial population, respectively. The lower and upper limits of the xth decision variable are represented by l^x and u^x. The size of the population is denoted by NP, whereas D represents the total number of decision variables. After the first and second steps, P and OP are combined. Then, the top NP solutions are chosen from $P \cup OP$ and assigned to the P. Conversely, the opposition-based generation jumping stage is executed during the generation process with a certain probability known as the jumping probability. In this stage, the opposite population of P is calculated, and the opposite value of each decision variable in a solution is determined dynamically:

$$OP_{j,G}^x = Min_G^x + Max_G^x - P_{j,G}^x$$
$$for \, j = \{1, \ldots, NP\}; x = \{1, \ldots, D\} \quad (2)$$

where Min_G^x and Max_G^x refer to the lower and upper values, respectively, of the xth value in population P at generation G. Finally, P and OP are combined. Then, the top NP solutions are chosen from $P \cup OP$ and are assigned to the P. We use the repair function to randomly remove in-edges that exceed the nbr_in_edge > 2 constraint and randomly change duplicate in-edges (have the same operator) coming to the same node.

III. EXPERIMENTAL DESIGN

The experiments are carried out in three parts: The first and the second part are the comparisons of MedUNAS with the NAS

studies and the state-of-the-art models presented in the literature, respectively. And the last part includes the ablation studies for the investigation of MedUNAS.

A. Datasets

Four different retinal vessel segmentation datasets (exemplary images are presented in Fig. 2 in the Supplementary Information) are used for our experimental study.

The DRIVE dataset [27] contains 40 retinal images with annotated labels, 20 of which are used for training and 20 for testing. The images have a resolution of 565×584 pixels and are coloured with three channels. We use the original partition of train and test sets for our experiments, and labels that the first observer annotates are selected as ground truth [8], [28].

The CHASE_DB1 dataset [29] is a collection of 28 color retina images with a resolution of 999×960 pixels. Two independent experts manually annotate these images, and the first expert's annotations are used as the ground truth. We use the first 20 images to train the model, while the last eight are used for evaluation [8], [28].

The DCA1 dataset [30] contains 134 X-ray coronary angiograms with corresponding ground-truth images, which a specialist cardiologist has annotated. Each image is represented as a grey-scale image with a resolution of 300×300 pixels, encoded in the PGM format. The DCA1 dataset is partitioned into two distinct subsets, with one consisting of 100 images serving as the training set and the other comprised of 34 images, forming the test set.

The CHUAC dataset [29] consists of 30 coronary angiography images, each with a resolution of 189×189 pixels, sourced from the CHUAC Hemodynamics Unit. The doctors manually create 512×512 pixel "ground truth" images using an image editing program. According to prior research, the first 20 images are used for training, while the last 10 are used for testing [31].

B. Implementation Details

The performance of deep neural networks is measured by training the network on the entire training set over long training epochs and evaluating it on a test set not used in the learning process. However, in NAS research, numerous potential networks need to be evaluated on GPU. Therefore, it is not practical to train all of them with the entire training data during long periods. To overcome this problem, two different evaluation phases are used: pre-evaluation and final-evaluation. Further details are available in the Supplementary Information.

IV. Results & Discussion

The segmentation performance of the proposed methods is evaluated using accuracy (ACC), sensitivity (SEN), specificity (SPE), F1 score (F1) and intersection over union (IOU) metrics [8], [9] (See Supplementary Information for details).

A. Comparisons with NAS Studies in the Literature

In this section, we compare MedUNAS with NAS studies in the literature for DRIVE and CHASE_DB1. To the best of our

TABLE I
PERFORMANCE RESULTS OF DIFFERENT NAS STUDIES ON DRIVE DATASET

Method	Param (M)	ACC	SEN	SPE	F1
Popat et al. v1 [18]	0.18	93.56	59.67	**98.50**	–
Popat et al. v2 [18]	8.10	95.34	75.01	98.31	–
HNAS v1 [19]	157	95.42	77.07	98.10	81.08
HNAS v2 [19]	2.90	95.46	77.44	98.09	81.29
Genetic U-Net v1 [7]	0.27	95.77	83.00	97.58	**83.14**
Genetic U-Net v2 [7]	0.27	**97.07**	83.00	98.43	**83.14**
BTU-Net [20]	16.80	96.89	–	98.44	81.78
MedUNAS GA	2.32	**97.11**	**84.54**	**98.64**	82.06
MedUNAS ODE	1.27	97.06	**83.41**	98.36	**82.18**

Red and blue colors indicate the best two values for each performance measure.

TABLE II
PERFORMANCE RESULTS OF DIFFERENT NAS STUDIES ON CHASE_DB1 DATASET

Method	Param (M)	ACC	SEN	SPE	F1
BTU-Net [20]	16.80	**97.87**	–	**98.64**	**81.09**
Genetic U-Net. v1 [7]	0.27	96.67	**84.63**	98.45	**82.23**
Genetic U-Net. v2 [7]	0.27	**97.69**	**84.63**	98.57	**82.23**
MedUNAS GA	8.71	97.48	**86.52**	98.44	79.50
MedUNAS ODE	6.01	97.56	84.50	**98.60**	80.22

Red and blue colors indicate the best two values for each performance measure.

knowledge, there are no NAS studies for DCA1 and CHUAC datasets. Therefore, we do not include these datasets in the experimental study for NAS comparison. We also investigate the performance of GA and ODE in MedUNAS. There are only a few NAS studies performed for the RVS problem in the literature. In this study, we include the recent studies and use the reported results provided in the corresponding papers for comparison [7], [18], [19], [20]. Popat et al. [18] and HNAS [19] report two models with least (v1) and most (v2) parameters. Similarly, two different results are reported in the Genetic U-Net with (v1) and without (v2) the binary field of view.

Table I presents the results of different NAS approaches for the DRIVE dataset with respect to the number of parameters, ACC, SEN, SPE, and F1 measures. When the results are examined, it can be observed that MedUNAS GA outperforms other studies for ACC, SEN and SPE measures. Furthermore, MedUNAS ODE achieves the second-best performance for SEN and F1 measures. Except for Genetic U-Net models, MedUNAS ODE and GA methods are significantly better than others in terms of ACC and SEN by a large margin. Moreover, MedUNAS ODE and GA achieve better results in all metrics (except SPE), despite having fewer parameters than the methods except the Genetic U-Net. Overall, these results indicate that the discovered architectures obtained by MedUNAS are more capable of detecting vessels.

We also compare the performance of MedUNAS with other NAS studies for the CHASE_DB1 dataset. Table II presents the results. MedUNAS ODE and MedUNAS GA give competitive results with respect to all metrics. They can produce architectures with fewer parameters than BTU-Net. Genetic U-Net generates models with the least number of parameters for two datasets. This is because they use element-wise addition for feature fusion, train each network for 130 epochs, and use difference-guided crossover. In contrast, our approach uses concatenation for feature fusion, trains each network for fewer

TABLE III
COMPARISON OF PROPOSED METHODS WITH THE STATE-OF-THE-ART METHODS ON DRIVE AND CHASE_DB1

Methods	Year	Param (M)	U-shaped	DRIVE					CHASE_DB1				
				ACC	SEN	SPE	F1	IOU	ACC	SEN	SPE	F1	IOU
U-Net [10]	2015	7.76	✓	96.78	80.57	98.33	81.41	68.64	97.43	76.50	**98.84**	78.98	65.26
UNet++ [32]	2018	9.05	✓	96.79	78.91	98.50	81.14	68.27	97.39	83.57	98.32	80.15	66.88
Attention U-Net [11]	2018	8.73	✓	96.62	79.06	98.31	80.39	67.21	97.30	83.84	98.20	79.64	66.17
HRNet [33]	2019	9.64	✗	97.04	80.40	**98.64**	82.65	**70.43**	**97.58**	84.43	98.47	81.48	**68.75**
CS-Net [34]	2019	8.40	✓	96.32	81.70	98.54	80.39	70.17	97.42	84.00	98.32	80.42	67.25
AG-Net [35]	2019	-	✓	96.92	81.00	98.48	-	69.65	97.43	81.86	98.48	-	66.69
RVSeg-Net [36]	2020	5.20	✓	96.81	81.07	98.45	-	-	97.26	80.69	98.36	-	-
SCS-Net [37]	2021	-	✓	96.97	82.89	98.38	-	-	97.44	83.65	98.39	-	-
SGL [15]	2021	15.53	✓	97.05	83.80	98.34	83.16	-	**97.71**	**86.90**	98.43	82.71	-
VSSC Net [31]	2021	8.05	✗	96.27	78.27	98.21	-	-	96.33	72.33	**98.65**	-	-
RV-GAN [16]	2021	14.81	✗	**97.90**	79.27	**99.69**	**86.90**	-	96.97	81.99	98.06	**89.57**	-
FR-UNet [8]	2022	5.72	✓	97.05	83.56	98.37	83.16	**71.20**	97.48	**87.98**	98.14	81.51	**68.82**
TP-Net [9]	2023	2.67	✗	96.29	**87.49**	97.58	**85.69**	-	97.30	86.00	98.41	**85.18**	-
MedUNAS GA	2023	2.32 - 8.71	✓	**97.11**	**84.54**	**98.64**	82.06	69.59	97.48	86.52	98.44	79.50	66.03
MedUNAS ODE	2023	1.27 - 6.01	✓	97.06	83.41	98.36	82.18	69.77	97.56	84.50	98.60	80.22	67.01

Red and blue colors indicate the best two values for each performance measure.

TABLE IV
COMPARISON OF PROPOSED METHODS WITH THE STATE-OF-THE-ART METHODS ON DCA1 AND CHUAC

Methods	Year	Param (M)	U-shaped	DCA1					CHUAC				
				ACC	SEN	SPE	F1	IOU	ACC	SEN	SPE	F1	IOU
U-Net [10]	2015	7.76	✓	97.58	78.16	98.66	77.35	63.07	97.84	58.81	**99.40**	67.68	51.15
UNet++ [32]	2018	9.05	✓	97.61	79.54	98.62	77.86	63.75	**98.12**	66.87	**99.37**	73.23	57.77
Attention U-Net [11]	2018	8.73	✓	97.55	79.86	98.53	77.48	63.24	98.00	65.26	99.13	71.54	55.69
HRNet [33]	2019	9.64	✗	97.77	80.07	98.76	**79.19**	**65.54**	**98.11**	74.56	99.06	**75.26**	**60.33**
CS-Net [34]	2019	8.40	✓	97.63	78.95	98.67	77.90	63.80	97.96	67.35	99.18	71.71	55.89
FR-UNet [8]	2022	5.72	✓	97.88	**82.48**	98.75	**80.22**	**67.08**	98.03	**81.71**	98.68	**76.01**	**61.51**
VSSC Net [31]	2021	8.05	✗	97.00	77.28	98.09	-	-	97.21	**78.92**	97.97	-	-
MedUNAS GA	2023	1.01 - 1.61	✓	**98.00**	80.89	**99.05**	78.20	64.33	97.92	78.65	99.16	71.96	56.31
MedUNAS ODE	2023	0.50 - 1.07	✓	97.93	**84.12**	**98.89**	78.22	64.37	98.07	78.29	99.12	74.56	59.54

Red and blue colors indicate the best two values for each performance measure.

epochs using Early-Stopping, and does not use any problem-specific operators.

The effect of the GA and ODE in MedUNAS is examined on four publicly available RVS datasets with regard to segmentation performance and model complexity. The corresponding results are presented in Tables III and IV (see the last two rows). Our findings indicate that ODE gives better results for F1 and IOU on the DRIVE with fewer parameters (1.27 M), while GA is better in terms of remaining measures. Moreover, ODE outperforms GA for all measures (except SEN), while having less complexity on CHASE_DB1. On the DCA1, ODE yields better results than GA for SEN, F1, and IOU. Additionally, ODE generates a model with 2.02 times fewer trainable parameters for this dataset. For CHUAC, ODE outperforms GA on all measures (except SEN and SPE) with 1.50 times fewer parameters. Overall, it can be observed that ODE yields better results for F1 and IOU measures on all datasets while generating models with lower complexity.

We compared the search performances of GA and ODE using t-distributed Stochastic Neighbor Embedding (t-SNE) [38] to visualize solution vectors obtained during the search process. The results are illustrated in Fig. 3, in which higher levels correspond to more accurate solutions, and the clustering of solutions at close levels indicates better exploitation performance. GA produces clustered solutions while ODE generates evenly distributed ones in a broader range. Both methods are able to explore the search space efficiently and show strong exploitation performance. Moreover, it is seen that ODE is better at generating diverse solutions that have good fitness values.
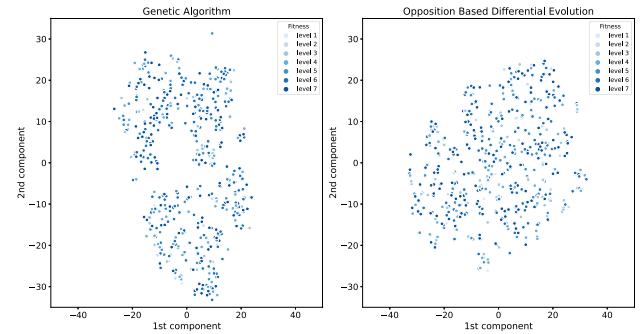

Fig. 3. Exploration-exploitation capabilities of GA (left) and ODE (right) with respect to the t-SNE projection of the generated architectures. Levels indicate the validation accuracy bins. Higher levels represent the more accurate solution.

We also observed that good solutions (level 6 and level 7) are available at distant points in the search space, indicating better exploration performance.

B. Comparisons with the State-of-The-Art Methods

In this section, the performance of MedUNAS is compared with the state-of-the-art models for DRIVE, CHASE_DB1, DCA1, and CHUAC. In the result tables, the performance of algorithms is given with respect to five performance measures and the model complexity. It should be noted that the results of the state-of-the-art models are directly taken from [8].

Table III presents the results for the proposed methods and state-of-the-art models for the DRIVE and CHASE_DB1
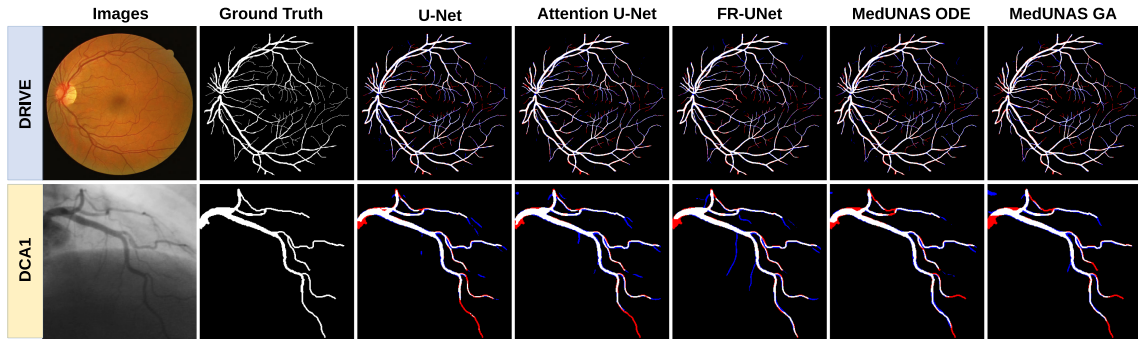
Fig. 4. Visual comparison of various U-shaped medical image segmentation models for two retinal vessel datasets. The white pixels indicate true positive pixels, the red pixels indicate false negative pixels, and the blue pixels indicate false positive pixels.

datasets. The results reveal that the proposed methods generate models with the lowest complexity for the DRIVE dataset and achieve highly competitive segmentation performance compared to state-of-the-art models. MedUNAS GA achieves the second-best ACC, SEN and SPE measures for DRIVE compared to state-of-the-art models, while yielding the best ACC, SEN and SPE results compared to U-shaped networks. SGL, RV-GAN and TP-Net give the best results for both DRIVE and CHASE_DB1. RV-GAN achieves the best ACC, SPE, and F1 on DRIVE and the best F1 value on CHASE_DB1, respectively. However, RV-GAN has up to 11.6 and 2.46 times more parameters than the proposed methods for DRIVE and CHASE_DB1, respectively. In addition, MedUNAS ODE and GA outperform RV-GAN in terms of ACC, SEN, and SPE on CHASE DB1, indicating that a model with less complexity can detect thin vessels more accurately and differentiate regions with and without vessels [37]. Despite having 12 times fewer parameters than SGL, MedUNAS ODE achieves highly competitive results on DRIVE regarding ACC, SEN, SPE, and F1 measures, while MedUNAS GA with 6 times fewer parameters yields better results than SGL on DRIVE in terms of ACC, SEN, and SPE measures. Moreover, the proposed methods provide better ACC and SPE measures than TP-Net on DRIVE and CHASE_DB1. Based on these results, the proposed methods achieve competitive results with less complexity for these datasets.

The segmentation results for two datasets, namely DCA1 and CHUAC, are presented in Table IV. The proposed methods are able to generate models with the least complexity compared to existing approaches. MedUNAS GA achieves superior results for ACC and SPE measures on DCA1 with 5.66 times fewer parameters than FR-UNet and gives second-best F1 and IOU values compared to U-shaped networks. Moreover, MedUNAS ODE shows competitive performance on DCA1 with 11.44 times fewer parameters. It achieves the best result for SEN and the second-best result compared to U-shaped networks for the other measures on this dataset. For the CHUAC, the MedUNAS ODE model obtains the second-best results with respect to ACC, F1 and IOU compared to U-shaped networks. Moreover, this model is 5.34 times less complex than FR-UNet, which performs the best with respect to SEN, F1 and IOU metrics on CHUAC. Although FR-UNet yields better performance, this is somewhat expected because it uses 160K, 492K, 193K, 124K patch images for training, while proposed methods are trained only 3K, 8K,

4K, 2K patch images for DRIVE, CHASE_DB1, DCA1 and CHUAC datasets, respectively. Compared to state-of-the-art methods, the proposed models obtained by our methods are the least complex networks for three datasets, indicating that competitive results can be obtained with less complex networks. With no special design for RVS, MedUNAS can be utilized to find an architecture superior to any manually designed network for any segmentation problem.

Fig. 4 shows the segmentation results visually for DRIVE and DCA1 (see Supplementary Information for the results on other datasets). We consider U-Net, Attention U-Net, FR-UNet [8] (pre-trained weights are available), MedUNAS ODE and MedUNAS GA for comparison. The figure illustrates False Negative (FN) and False Positive (FP) errors, where FN refers to vessels that exist in the ground truth but are not detected by the network, and FP refers to regions of the vessel that are detected by the network but do not exist in the ground truth. A good segmentation model is expected to have fewer FP and FN pixels in the images. For all datasets, the U-Net and Attention U-Net models are not able to accurately segment vessel regions, resulting in a greater FN and FP error. Therefore, the optimized architectures are more effective than U-Net and Attention U-Net in extracting vessel regions. Additionally, all segmentation models perform similarly well on the DRIVE, while MedUNAS ODE and GA give better FP errors, with fewer false segmentation regions in the generated images for DCA1. We also conduct additional evaluation only on thin vessel segmentation using FR-UNet and MedUNAS ODE models since they better detect thin vessels. We apply the same steps as in [39] to extract thin vessels in DRIVE and CHASE_DB1. The results indicate that MedUNAS ODE performs better than FR-UNet in F1, ACC, SEN, and IOU metrics by 2.0, 2.2, 2.8, and 2.7 points for DRIVE, while FR-UNet outperforms MedUNAS ODE by 6.0, 5.1, 8.6, and 8.4 points for CHASE_DB1 (See Table III in the Supplementary Information). This may be because CHASE_DB1 has fewer training samples and less thin vessel structures.

C. Ablation Studies

We examine the impact of the loss function and evaluate the model's generalization performance on different segmentation tasks, such as cell and brain vessel segmentation. Also, the robustness of the model with and without fine-tuning is tested

TABLE V
COMPARISON OF DIFFERENT LOSS FUNCTIONS FOR MEDUNAS ODE

Loss Function	ACC	SEN	SPE	F1	IOU
Dice	**97.11↑**	**83.89↑**	98.63↑	82.05	69.60
BCE	97.07	76.67	99.23↑	81.09	68.27
Focal	96.84	71.45	**99.58↑**	78.22	64.35
Jaccard	97.10↑	83.59↑	98.53↑	**82.30↑**	**69.96↑**
Dice+BCE	97.10↑	82.09	98.70↑	82.04	69.57
Dice+BCE+Jaccard	97.06	83.41	98.36	82.18	69.77

The bold values indicate the best values for each performance measure.

TABLE VI
CROSS-TRAINING EVALUATION (WITH FINE-TUNING ON TEST DATASET)

	with pre-training				
Test dataset	ACC	SEN	SPE	F1	IOU
DCA1	97.89	84.31↑	98.72	78.20	64.33
CHUAC	93.88	84.52↑	94.56	49.11	32.84
CHASE_DB1	97.38	88.08↑	98.07	79.71	66.29
	without pre-training				
Test dataset	ACC	SEN	SPE	F1	IOU
DCA1	97.87	85.49↑	98.64	78.43↑	64.65↑
CHUAC	97.95	75.57	98.97	73.77	58.56
CHASE_DB1	97.57↑	88.63↑	98.42	80.50↑	67.40↑

on different retinal blood vessel datasets. We perform the same training and evaluation steps explained in the final-evaluation phase for all ablation studies. Since MedUNAS ODE gives better results with lower parameters than MedUNAS GA, we have performed all ablation studies for this model.

1) Loss Functions: This part focuses on examining the impact of different loss functions on the segmentation result. We evaluate the performance of MedUNAS ODE trained on the DRIVE dataset, which is called ODE-DRIVE, using various loss functions to identify the best loss function [8]. Table V shows the comparison results of different loss functions for MedUNAS ODE with respect to five performance metrics. The ↑ indicates that the result of ODE-DRIVE shown in Table III is improved using the corresponding loss function compared to Dice+BCE+Jaccard loss. As shown in Table V, Dice yields the best results for ACC and SEN measures, while Jaccard produces the best results for F1 and IOU. Furthermore, Focal results in the best SPE outcomes. It can be seen that choosing the loss function correctly improves the segmentation performance.

2) Cross-training without Fine-Tuning: Cross-training evaluation in medical image segmentation is a critical process that helps assess the effectiveness of machine-learning algorithms for segmenting medical images. In addition, this approach helps evaluate the model's generalizability and ability to segment medical images from unseen datasets accurately. In this part, we use the best MedUNAS ODE model trained on the DRIVE training set. This model is evaluated on DCA1, CHUAC and CHASE_DB1 test sets without re-training. Overall, inferior results are obtained for all measures (specifically, minor decreases for the ACC and SPE measures, significant reductions for others). These results may be due to the structural differences and imaging techniques between the datasets used for training and testing. As a result, it has been seen that our model is not sufficient for cross-training without fine-tuning (see Table IV in the Supplementary Information).

3) Cross-training with Fine-Tuning: In this part, we investigate the impact of fine-tuning on cross-training. We evaluate the model searched using ODE-DRIVE on other test datasets. However, in contrast to the sub-section IV-C2, we fine-tune the model with training images of the test datasets. Additionally, the influence of pre-training on the model is investigated. The results are shown in Table VI. The ↑ indicates that the reported result in Table III has been improved. The ODE-DRIVE (without pre-training) gives better results than the reported in Table IV for the DCA1 dataset in terms of SEN, F1 and IOU measures. For the DCA1, pre-training only enhances the model's SEN

performance. Furthermore, the ODE-DRIVE model, without pre-training, achieve better results on the CHASE_DB1 across all metrics (except SPE), despite having significantly fewer parameters (4.73 times less). Additionally, this model shows an improvement of 4.13 points in the SEN measure. Similarly, pre-training only leads to an improvement in the SEN measure for CHASE_DB1. However, ODE-DRIVE does not improve the performance without pre-training for the CHUAC, whereas pre-training improves only the SEN measure. Our findings indicate that competitive results can be obtained without performing NAS for each dataset. Moreover, we observe that starting with trained weights on the DRIVE does not lead to the expected level of improvement, possibly due to structural differences between datasets and imaging techniques. Our analysis also reveals that fine-tuning could significantly improve cross-training results, particularly in terms of F1, SEN and IOU metrics.

4) Generalization Performance: Generalization performance evaluation is essential to assess the capacity of deep learning models on novel images and tasks. This is important because medical images can vary widely, and a model searched on a specific dataset may not perform well on new data. In this study, we evaluate the generalization performance of the model (ODE-DRIVE) on two additional datasets: Optofil [22] for cell segmentation and KUVESG [40] for brain vessel segmentation. The Optofil dataset contains 4914 gray-scale cell culture images, and the KUVESG dataset consists of 335 mouse cortex images. The search model is trained on the Optofil and KUVESG datasets and evaluated on the corresponding test set. It is also compared with the model presented in [22]. The results indicate that our model gives better results for almost all measures for Optofil, while it produces inferior results for KUVESG (see Table V in the Supplementary Information). This is because of the structural dissimilarities in the KUVESG dataset images, which contain more thick vessel images. Additionally, the searched model has significantly fewer parameters (4.72 times less) than the one in [24]. As a result, the searched model has a strong generalization ability for cell segmentation tasks and gives competitive results for brain vessel segmentation.

V. CONCLUSION

This study presents a novel NAS method for U-shaped networks, i.e. MedUNAS, that gives highly competitive results for recent state-of-the-art methods and NAS studies in terms of segmentation accuracy, computational resources, and training

time. The results reveal that MedUNAS ODE and GA achieve the best and second-best segmentation performance in most of the compared datasets while utilizing less than 50% of the parameters used by U-shaped state-of-the-art methods. Additionally, the proposed methods outperformed the baseline U-Net on four RVS datasets, even when using networks with up to 15 times fewer parameters because of the cell-based structure, modified encoding schemes and flexible search space. This suggests that creating a high-quality network structure with less complexity for RVS is possible. The study also highlights the effectiveness of ODE and GA on the proposed search space and compares discrete and continuous encoding strategies. The ablation studies demonstrate that the proposed method provides successful results for different medical segmentation domains, such as cell segmentation, and generated networks can be efficiently adapted to the new task with fine-tuning. The proposed approach can serve as a valuable tool for automating the RVS process in clinical practice, enabling clinicians to identify and monitor the progression of various retinal diseases more efficiently and accurately. However, MedUNAS has limitations in segmenting thin vessels and requires fine-tuning for good results on new data. In future works, MedUNAS can be trained with thin and thick vessels separately and use predictor-guided search methods to improve the search efficiency of NAS. Additionally, performing post-processing steps like erosion and dilation can enhance the segmentation results. Moreover, the proposed methods will be tested for different medical segmentation tasks and applied to the 3D vessel segmentation.

REFERENCES

[1] M. D. Abrámoff, M. K. Garvin, and M. Sonka, "Retinal imaging and image analysis," *IEEE Rev. Biomed. Eng.*, vol. 3, pp. 169–208, 2010.

[2] R.-Q. Li et al., "Real-time multi-guidewire endpoint localization in fluoroscopy images," *IEEE Trans. Med. Imag.*, vol. 40, no. 8, pp. 2002–2014, Aug. 2021.

[3] B. Wang et al., "Dual encoding U-Net for retinal vessel segmentation," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, 2019, pp. 84–92.

[4] Y.-D. Zhang et al., "Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation," *Inf. Fusion*, vol. 64, pp. 149–187, 2020.

[5] W. Wang et al., "Covid-19 diagnosis by WE-SAJ," *Syst. Sci. Control Eng.*, vol. 10, no. 1, pp. 325–335, 2022.

[6] W. Wang et al., "PSTCNN: Explainable COVID-19 diagnosis using PSO-guided self-tuning CNN," *BIOCELL*, vol. 47, no. 2, pp. 373–384, 2023.

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

[8] W. Liu et al., "Full-resolution network and dual-threshold iteration for retinal vessel and coronary angiograph segmentation," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 9, pp. 4623–4634, Sep. 2022.

[9] Z. Qu et al., "TP-Net: Two-path network for retinal vessel segmentation," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 4, pp. 1979–1990, Apr. 2023.

[10] O. Ronneberger et al., "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241.

[11] O. Oktay et al., "Attention U-Net: Learning where to look for the pancreas," 2018, *arXiv:1804.03999*.

[12] Y. Wu et al., "Vessel-Net: Retinal vessel segmentation under multi-path supervision," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, 2019, pp. 264–272.

[13] Y. Rong et al., "Segmentation of retinal vessels in fundus images based on u-net with self-calibrated convolutions and spatial attention modules," *Med. Biol. Eng. Comput.*, vol. 61, no. 7, pp. 1745–1755, 2023.

[14] C. Chen, J. H. Chuah, R. Ali, and Y. Wang, "Retinal vessel segmentation using deep learning: A review," *IEEE Access*, vol. 9, pp. 111985–112004, 2021.

[15] Y. Zhou et al., "Study group learning: Improving retinal vessel segmentation trained with noisy labels," in *Proc. Med. Image Comput. Comput. Assist. Interv. (MICCAI)*, 2021, pp. 57–67.

[16] S. A. Kamran et al., "RV-GAN: Segmenting retinal vascular structure in fundus photographs using a novel multi-scale generative adversarial network," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, 2021, pp. 34–44.

[17] Z. Fan et al., "Evolutionary neural architecture search for retinal vessel segmentation," 2020, *arXiv:2001.06678*.

[18] V. Popat et al., "Ga-based U-Net architecture optimization applied to retina blood vessel segmentation," in *Proc. 12th Int. Joint Conf. Computat. Intell.*, 2020, pp. 192–199.

[19] Y. Houreh et al., "HNAS: Hyper neural architecture search for image segmentation," in *Proc. 13th Int. Conf. Agents Artif. Intell. Sci. Technol. Pub.*, 2021, pp. 246–256.

[20] C. Rajesh et al., "An evolutionary u-shaped network for retinal vessel segmentation using a binary teaching–learning-based optimization," *Biomed. Signal Process. Control*, vol. 83, 2023, Art. no. 104669.

[21] J. Dong, B. Hou, L. Feng, H. Tang, K. C. Tan, and Y. -S. Ong, "A cell-based fast memetic algorithm for automated convolutional neural architecture design," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 17, 2022, doi: 10.1109/TNNLS.2022.3155230.

[22] Z. Kuş et al., "Neural architecture search using metaheuristics for automated cell segmentation," in *Proc. 14th Int. Conf.*, 2023, pp. 158–171.

[23] N. H. Awad et al., "Differential evolution for neural architecture search," 2020, *arXiv:2012.06400*.

[24] Z. Kuş et al., "Differential evolution-based neural architecture search for brain vessel segmentation," *Eng. Sci. Technol., Int. J.*, vol. 46, 2023, Art. no. 101502.

[25] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Berlin, Germany: Springer, 2003.

[26] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, 1997, Art. no. 341.

[27] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–509, Apr. 2004.

[28] J. Li et al., "Magf-net: A multiscale attention-guided fusion network for retinal vessel segmentation," *Measurement*, vol. 206, 2023, Art. no. 112316.

[29] A. Carballal et al., "Automatic multiscale vascular image segmentation algorithm for coronary angiography," *Biomed. Signal Process. Control*, vol. 46, pp. 1–9, 2018.

[30] F. Cervantes-Sanchez et al., "Automatic segmentation of coronary arteries in X-ray angiograms using multiscale analysis and artificial neural networks," *Appl. Sci.*, vol. 9, no. 24, 2019, Art. no. 5507.

[31] P. M. Samuel and T. Veeramalai, "VSSC Net: Vessel specific skip chain convolutional network for blood vessel segmentation," *Comput. Methods Programs Biomed.*, vol. 198, 2021, Art. no. 105769.

[32] Z. Zhou et al., "UNet: A nested U-Net architecture for medical image segmentation," in *Proc. Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support*, 2018, pp. 3–11.

[33] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2019, pp. 5686–5696.

[34] L. Mou et al., "CS-Net: Channel and spatial attention network for curvilinear structure segmentation," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, 2019, pp. 721–730.

[35] S. Zhang et al., "Attention guided network for retinal image segmentation," in *Proc. Med. Image Comput. Computer Assist. Interv.*, 2019, pp. 797–805.

[36] W. Wang et al., "Rvseg-net: An efficient feature pyramid cascade network for retinal vessel segmentation," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, 2020, pp. 796–805.

[37] H. Wu et al., "Scs-net: A. scale and context sensitive network for retinal vessel segmentation," *Med. Image Anal.*, vol. 70, 2021, Art. no. 102025.

[38] J. Siems et al., "Nas-bench-301 and the case for surrogate benchmarks for neural architecture search,", 2020, *arXiv:2008.09777*.

[39] Z. Yan, X. Yang, and K.-T. Cheng, "A three-stage deep learning model for accurate retinal vessel segmentation," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 4, pp. 1427–1436, Jul. 2019.

[40] E. Özkan et al., "Kuvesg," 2022. [Online]. Available: https://zenodo.org/record/7383294