IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 16, 2019, accepted March 30, 2019, date of publication April 4, 2019, date of current version April 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2908991

NAS-Unet: Neural Architecture Search for Medical Image Segmentation

YU WENG[1], TIANBAO ZHOU[1], YUJIE LI[2], AND XIAOYU QIU[3]

[1]College of Information Engineering, Minzu University of China, Beijing 100081, China
[2]School of Information Engineering, Yangzhou University, Yangzhou 225009, China
[3]Library of Shandong University of Traditional Chinese Medicine, Jinan 250355, China

Corresponding author: Tianbao Zhou (tianbaochou@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772575, and in part by the National Key R&D Program of China under Grant 2017YFB1402101.

ABSTRACT Neural architecture search (NAS) has significant progress in improving the accuracy of image classification. Recently, some works attempt to extend NAS to image segmentation which shows preliminary feasibility. However, all of them focus on searching architecture for semantic segmentation in natural scenes. In this paper, we design three types of primitive operation set on search space to automatically find two cell architecture DownSC and UpSC for semantic image segmentation especially medical image segmentation. Inspired by the U-net architecture and its variants successfully applied to various medical image segmentation, we propose NAS-Unet which is stacked by the same number of DownSC and UpSC on a U-like backbone network. The architectures of DownSC and UpSC updated simultaneously by a differential architecture strategy during the search stage. We demonstrate the good segmentation results of the proposed method on Promise12, Chaos, and ultrasound nerve datasets, which collected by magnetic resonance imaging, computed tomography, and ultrasound, respectively. Without any pretraining, our architecture searched on PASCAL VOC2012, attains better performances and much fewer parameters (about 0.8M) than U-net and one of its variants when evaluated on the above three types of medical image datasets.

INDEX TERMS Medical image segmentation, convolutional neural architecture search, deep learning.

I. INTRODUCTION

With the development and popularization of medical imaging analysis equipment, including Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and ultrasound, have become indispensable devices for medical institutions to carry out disease diagnosis, surgical planning and prognosis evaluation. MRI is the most widely used technique in the field of radio imaging. One of the outstanding features of MRI imaging is the wide variety of imaging sequences. In MRI, the contrast of an image depends on the phase contrast pulse sequence. The most common pulse sequences are T1 (spin-lattice; that is, magnetization in the same direction as the static magnetic field) weighted and T2 weighted spin (spin-spin; transverse to the static magnetic field). An MRI may yield different information compared with CT. There may be risks and discomfort associated with MRI scans. Compared with CT scans, MRI scans typically take longer and are louder, and they usually need the subject to enter a narrow, confining tube. Ultrasound imaging (ultrasonography) uses high-frequency sound waves to view inside the body. Unlike CT and MRI, the resolution of ultrasound images is relatively low.

Medical image analysis is the first step in analyzing medical images, which helps to make images more intuitive and improves diagnostic efficiency. Medical image segmentation is a critical step in the field of medical image analysis. In order to provide a reliable basis for clinical diagnosis and pathology research, and assist doctor to make a more accurate diagnosis, it need to segment the parts of medical images we focus and extract relevant features. Initially, medical image analysis was done with sequential application of low-level pixel processing (e.g. region based method [1] or threshold based method [2]) and mathematical modeling to construct compound rule-based systems that solved particular tasks [3]. The segmentation results of this period are generally not semantically labeled. In the era of deep learning, image segmentation generally denotes semantic segmentation, which refers to the

The associate editor coordinating the review of this manuscript and approving it for publication was Huimin Lu.

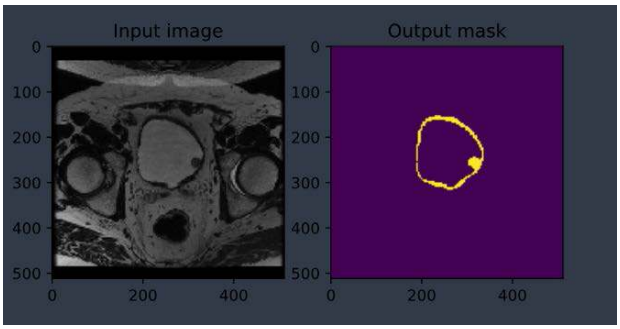


FIGURE 1. An example of bladder segmentation results by deep learning method. (a) is the original bladder image, (b) is the segmentation result.

recognition of images at the pixel level (the object category to each pixel in the image belongs is marked) [4], [5]. For example, in Figure 1, the medical image in the left image consists of the bladder wall and the other tissues, and the right image is the result of its semantic segmentation, which divides the pixel semantic object, i.e. the yellow-labeled bladder wall, similarly, other organizations are considered as background and to be marked purple. The most successful type of deep learning models for image analysis to date are convolutional neural networks (CNNs).

The mutual promotion of deep learning and big data and cloud computing has brought much development to computer vision [6]. CNN is the most commonly used neural network in the field of computer vision which is proposed to solve image classification problems. Image segmentation is a common task in both natural and medical image analysis. To tackle this, CNN can simply be used to classify each pixel in the image individually, by presenting it with patches extracted around the particular pixel, and produce a multi-channels likelihood map of the same size as input image. However, a huge memory will cost when keep the dimension of feature maps all the time. More usually, a down-sampling layer (such as max pooling and average pooling) is added after several convolutional layers to reduce the dimension of feature map and refine the high-level context. Unfortunately, this may result in output with a far lower resolution than the input. FCNs (Fully Convolutional Networks) [7] is one of several methods proposed to prevent this decrease in resolution. It was the first work to train end-to-end for pixel-wise prediction by replace fully connected layers as a series of up-sampling layers after convolutional layers. Classical CNNs generally use a fully connected layer to obtain fixed-length feature vectors after last convolutional layer, and put them into a classifier (e.g. softmax layer). In contrast, FCNs can accept any size of input image — the up-sampling layer after the last convolutional layer can restore the dimension of its inputs to the same as the input image, so that a prediction can be generated for each pixel while preserving the spatial information in the original input image, and finally pixel-by-pixel classification on the up-sampling feature maps to the expected image segmentation. Similar to FCNs, U-net [8] consists of convolutional layers, down-sampling layers, and up-sampling layers. Different from FCNs, the number of down-sampling layers and up-sampling layers and convolution layers between them in the U-net is the same. In addition, U-net uses the skip connection operation to connect each pair of down-sampling layer and the up-sampling layer, which makes the spatial information directly applied to much deeper layers and a more accurate segmentation result.

From the earliest LeNet [9] to AlexNet [10], VggNet [11], GoogleNet [12], ResNet [13] and the recent DenseNet [14], the performance of CNN models getting stronger and more mature. Many works have designed network structures for specific tasks [15], [16].These popular network architectures are currently designed by industry experts and scholars for months or even years. This is because designing a network architecture with excellent performance often requires a large amount of knowledge in the field. The average researcher does not have this capability, and the design process is time-consuming and labor-intensive. Based on this, the focus of current convolutional neural networks has move to neural architecture search (NAS) [17]. NAS can be seen as subfield of AutoML (auto machine learning) and has significant overlap with hyper-parameter optimization and meta-learning. The present research on NAS focuses on three aspects: search space, search strategy, and performance estimation strategy. The search space defines which architectures can be represented in principle. Incorporating prior knowledge about properties well-suited for task can reduce the size of the search space and simplify the search task. For example, in image classification, the search space includes the selection of primitive operations at each searching step and the prior backbone architecture used to define outer network. The search strategy details how to explore the search space. The objective of NAS is typically to find architectures that have high evaluated performance on unseen data (e.g. split training datasets into training and validation, and search architecture on training but evaluated by validation) [17]. Much work has been done on the NAS, most of which focus on image classification task [18]–[23].

Although NAS has great potential in the field of computer vision, the real promise depends on that can be extended to deal with visual tasks other than image classification, in particular, computer vision core issues such as image semantic segmentation, instance segmentation and object detection which rely on high-resolution image input and multi-scale image representation. Introducing NAS directly from image classification to image semantic segmentation is not feasible: firstly, the search space of classification task differ notably from segmentation; secondly, the idea of transfer learning from low to high image resolutions was unexpected [24]. A logical idea for solving the above two problems is build specific search space for image segmentation and search the architecture with high-resolution image. Some work has been trying to solve the above two problems and achieved some success — exactly the idea that recent work has followed [24], [25].

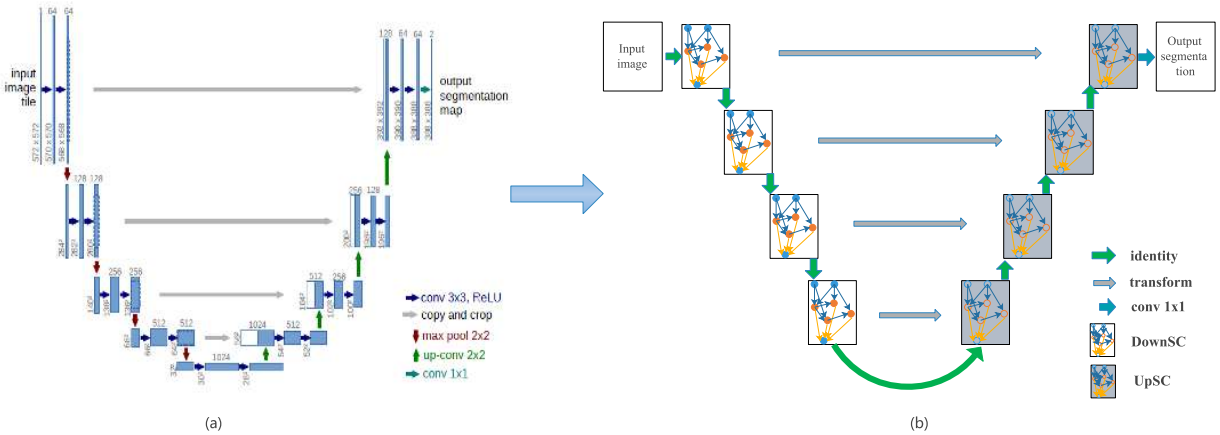


FIGURE 2. (a) U-Net architecture (b) The U-like backbone of Nas-Unet architecture, the rectangle represents cell architectures need to search. Notice that both of DownSC and UpSC contain their down or up operation in NasUnet, and the green arrow merely represents the flow of feature map (input image). The gray arrow is a transform operation belongs to UpSC and is also automatically searched.

In this paper, we propose new primitive operation sets for medical image segmentation. Inspired by the success of U-net and its variants in medical image segmentation, we use a U-like architecture as our backbone network (which means outlier network) and parallel search two cell-based architectures (down-sampling cell and up-sampling cell) on PASCAL VOC 2012 [26], denoted as DownSC and UpSC respectively. Finally, we get our architecture denoted as NAS-Unet which stackd by the same number of DownSC and UpSC. Our work reveals that NasUnets are more efficient in the parameters usage and get a much better performance than U-Net and FC-DenseNet [27] (a variant U-Net) in all types of medical image dataset we mentioned before. In summary, our contributions are as follows:

1) This article is the first attempt to apply NAS to medical image segmentation.
2) We propose different primitive operation sets for DownSC and UpSC respectively on U-Like backbone network for searching. When done with search, we empirically found that the standard skip connection is replaced with cweight operation (see V-A) in our UpSC architecture.
3) We show that the performance of NAS-Unet outperforms U-Net and one of its variants (FC-Densenet) in all types of medical image segmentation datasets we evaluated without using any pre-trained backbone. The training time of NAS-Unet closes to U-Net, but parameters amount is only 6%. FC-Densenet has twice memory cost than ours.

We have released our code at https://github.com/tianbaochou/NasUnet.

II. RELATED WORK

A. MEDICAL IMAGE SEGMENTATION BASED ON MODERN CNNS

To our knowledge, Ciresan *et al.* [28]. first use Deep Neural Network to medical image segmentation. A segmentation stacks of electron microscopy images is token on using a convolutional neural network. 'patch' is the key idea for finishing segmentation — to segment an entire stack, the classifier is applied to each pixel of every slice in a sliding window by extracting a patch around the pixel. A drawback of this naïve sliding-window approach is that input patches from neighboring pixels have plenty of overlap and redundant computation. It is also noted in [28] that segmenting an entire stack with that approach is time-inefficiency, which takes at least 10 minutes for an stack on four GPUs. Ronneberger *et al.* [8] rewrite the fully connected layers as convolutions and attempt the same task with better results. As show in Figure 2 (a), the authors took the idea of FCNs [7] one step further and propose the U-Net architecture, their design is based on encoder-decoder network framework: a input images put into a encoder architecture to extract the high-level context and then that context flow to a decoder architecture to restore the spatial information and pixel classification results. Although this is not the first work to use encoder-decoder (e.g. Shelhamer *et al.* [7] use a pre-trained modern CNN networks as encoders and a 'up' convolutional layer as decoders (FCNs-32)) in convolutional neural network, the authors combined it with horizontal skip-connections to directly connect opposing contracting and expanding convolutional layers.

After the U-Net network was proposed, it performs well in the field of medical image segmentation. Many researchers have made various improvements on the. Çiçek *et al.* [29] first proposed a 3D U-Net network architecture that implements 3D image segmentation by inputting a continuous 2D slice sequence of 3D images. Milletari *et al.* [30] proposed a 3D deformation architecture called V-net based on U-Net. The V-Net architecture uses the Dice coefficient loss function instead of the cross-entropy loss function, that directly minimizes this commonly used segmentation error measure. The authors have further introduced residual blocks to the original U-shaped design. Both of above two methods extended U-shaped architecture with 3D convolutional kernels.

Drozdzal *et al.* [31] make a different between long skip connections (which means a skip connection between two feature maps that are far away) and short skip connections (which usually called single residual block) and find that both of them to be beneficial in creating deep architectures for medical image segmentation. Simon *et al.* [27] combine Densely Connected Convolutional Networks (DenseNets) with U-shaped architecture by replacing the convolutional layer as Dense Block in that backbone and extend to nature image segmentation, and achieve a well-performance.

B. NEURAL ARCHITECTURE SEARCH

Designing a good neural network architecture is time-consuming and laborious, in order to reduce the efforts and resources cost on manually designing network architecture, some scholars put their attention on neural architecture search (NAS). Recently, most of works focus on search CNN architecture for image classification and few on RNN for language task. As we mentioned before, NAS includes three components: search space, search strategy, and performance estimation. The search algorithms mainly includes heuristic algorithm [19], [21], [32]–[34], reinforcement learning [35]–[38], [38], [39], the Bayesian optimization method [40], [41] and the gradient-based method [20], [42], [43]. Performance estimation can be understood from two aspects. Firstly, the performance of the candidate architecture is evaluated to determine whether to be kept (or expanded) for the next update. Secondly, we need a deeper network stacked by the cells (when use cell-based search space) or current candidate architecture and put training dataset into it for training and to evaluate the final performance.

The network search space includes the topology of nodes and operations between each connected node. The former works try to directly build the entire network architecture [36], [44]. However, since NASNet [37] successfully stacks the cell together on ImageNet, more recent works [20], [22], [23], [45], [46] prefer to search the repeatable cell structure, but keep the backbone network fixed at first. The latter can improve search efficiency. Recently, lots of work on NAS have proposed many efficient algorithms to generate the topology of nodes, while are predicated on powerful but tractable architecture search spaces. In fact, if we have a rich and not overly expansive search space, even use a random search may achieve strong results [20], [40]. Therefore, in this paper, we focus our effect on constructing the cell-based level search space for medical image segmentation. In addition, we use current differential architecture search methods [20], [22] as our search algorithm to accelerate our search process.

C. NAS APPLICATION ON IMAGE SEGMENTATION

NAS has mainly solved image classification tasks since it was proposed. A few work recently applied NAS to image segmentation. Chen *et al.* [24] first introduce NAS to solve image segmentation. The authors show that even with random search on constructing a recursive search space, the architecture search outperforms human-invented architectures and achieves better performance on many segmentation datasets. However, this work does not use one-shot searching, which focused on search a small Atrous Spatial Pyramid Pooling (ASPP) module called DPC (similar as decoder) and fix the pre-trained backbone (modified Xception) as encoder. Liu *et al.* [25]. propose Auto-DeepLab: a general-purpose network level search space, and jointly search across two-level hierarchy (network level and cell level architecture). The authors indicate that search space includes various existing designs such as DeepLabv3, ConvDeconv and Stacked Hourglass. However, the search space of Auto-DeepLab does not include U-Net architecture, which is the most famous architecture in the field of medical image segmentation.

The most similar work to ours is [27], which use dense blocks to replace convolutional layers both in contracting stage and expanding stage. However, we replace all predesigned block with cells searched by NAS method.

III. CELL-BASED ARCHITECTURE SEARCH SPACE

In this section, we begin by describing the common representation of the CNN architecture we used. We will show how to represent a cell architecture as a DAG. After that, we will introduce our search space for medical image segmentation. Finally, we will introduce two types of cell architecture in detail.

A. CNN ARCHITECTURE REPRESENTATION

A directed acyclic graph (DAG) is used to represent the network topology architecture, in which each node h_i represents input image or a feature map and each edge e_{ij} is associated with an operation between (e.g. convolution operation, a pooling operation and a skip connection) node h_i and node h_j. When the generation method of the DAG is unrestricted, its network architecture space will be very large, which will bring great challenges to the present search algorithm. Therefore, we use cell-based architecture. When determining the best cell architecture, we can stack the cells into a deeper network on the backbone network. In other words, the architecture of cell is shared by entire network.

B. SEARCH SPACE FOR MEDICAL IMAGE SEGMENTATION

In this section, we will introduce our selection of primitive operation set for DownSC and UpSC architecture. After that, we will describe how to construct them.

1) THE SELECTION OF PRIMITIVE OPERATIONS

How to choose suitable primitive operations? We have investigated the popular CNN architecture and the former NAS that has great success on image classification, and summary the important standard for selecting primitive operations in our work:

1) **No redundancy:** It means that each primitive operation should has some unique properties that cannot be replaced by the others. Although some works [25], [34]

show that 5 × 5 convolution may be considered during the searching. Large receptive fields such as 5 × 5 size convolution and 7 × 7 size convolution can be replaced by stacking enough 3 × 3 size convolutions. Therefore, all convolution operation will limit to 3 × 3 size and pooling operation is 2 × 2.

2) **Less parameters:** It means consuming less memory resources during the search process; The original U-Net need about 31 million parameters, it was huge for mobile device. In our work, depthwise-separable convolution operation will be introduced since it will dramatically reduce network parameters without sacrificing network performance.

When set the sliding step (stride value) as 2, the convolution operation can halve the dimension of feature map or double the dimension, the later called 'Up'-convolution. This indicates that the down operation and up operation can be derived from the same base operation. In contrast, different from the primitive operations in image classification, the 'up'-version of some operations make no sense (e.g. the identity operation) and the 'up'-version of pooling operations (e.g. the average pooling and max pooling) do not exist. For the sake of convenience, we build 3 different types of primitive operation set.

TABLE 1. Three types of primitive operation set used for searching cells. Note that all the size of convolutional operations include cweight operation is 3 × 3 and pooling operation is 2 × 2.

POs type	Down POs	Up POs	Normal POs
1	average pooling	up cweight	identity
2	max pooling	up depth conv	cweight
3	down cweight	up conv	dilation conv
4	down dilation conv	up dilation conv	depth conv
5	down depth conv	-	conv
6	down conv	-	-

As the Table 1 show, the depth conv indicates depthwise-separable operation and other operation, except dilation conv [47] and cweight [48], are prevalent in current NAS methods. Cweight operation indicates squeeze-and-excitation [48] operation. In earlier CNN architecture, the features we generate for all channels are directly combined evenly. A natural next step is to automatically learn the weights of every channel. It is exactly what the squeeze-and-excitation operation does. The squeeze-and-excitation operation suppresses some redundant features and enhances useful features by assigning weights to feature channels. Down cweight operation and up cweight operation would halve or double the dimension of feature map before channels re-weighted. It's worth to notice that when the former NAS articles show their good architecture have searched on image classification tasks, dilation (atrous) convolution operation almost not appear. However, the original intention of this operation is to solve the problem of image segmentation. As we mentioned before, different from image classification tasks, search the architecture with image segmentation need high-resolution input. A huge memory consumption is undoubtedly obvious. For example, an 512 × 512 image cost about 3 GB GPU memory to predict result use original U-Net architecture, batch size no more than 4 when load model on 12GB Titan pascal GPU.

We use the Conv-ReLU-GN order for all convolutional operations. GN indicates group normalization [49], as the Wu *et al.* show that this normalization is better than batch normalization, especially when the batch size is much smaller. Since the batch size of segmentation task is much smaller than image classification, we use group normalization instead of batch normalization.

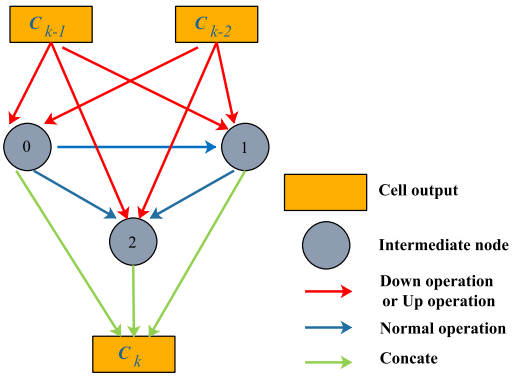


FIGURE 3. An example of cell architecture. The red arrow indicates an down operation (such as max pooling), the blue arrow indicates the normal operation (e.g. identity operation, convolution operation which no reduce the dimension of feature map) and the green arrow represents a concatenate operation.

C. TWO TYPES OF CELL ARCHITECTURE

As show in Figure 2 (b), we design two types of cell architectures called DownSC and UpSC based on U-like backbone. Inside both of two cells, the input nodes are defined as the cell outputs in the previous two layers [20], [37]. As shown in Figure 3, all operations adjacent to the input nodes are either Down POs or Up POs. Let $H = h_i$ be the set of M intermediate nodes (or called feature map layers). Same to DARTs [20], the total number edges between all intermediate nodes and input nodes is $E = 2M + M(M-1)/2$.

On the contracting step, we link L_1 cells to learn the different level of semantic context information, and produce a much smaller probability map denoted DC_{out}. Similarly, on the expanding step, we use the same number of cells to restore the spatial information of each probability value in DC_{out} and expand them consistent with input image. The total number of cells in final network denoted Nas-Unet is $L = 2L_1$. Unlike FC-densenet architecturec [27], we not only replace the convolution layers as these cells, but also move up-sampling operation and down-sample operation into the cells. In other word, both normal operation (such as identity operation) and up/down-sample operations are considered into the cells. As shown in Figure 2 (b), the transform is also a operation from Norm POs in UpSC. Our search space covers many popular U-like architecture, such as U-Net [8] and

FC-DenseNet [27]. It's worth to notice that original U-Net architecture has an additional convolution layer at the middle of network. However, in our article, we do not follow this experience, since we have a strictly symmetric architecture stacked by the serval number of two pair of cells.

IV. SEARCH STRATEGY

We first describe how to construct an over-parameterized network with all candidate path using recent works [20], [22], [50]. We then introduce a more efficient architecture parameters update strategy for saving GPU (since the CPUs is much slower for searching process, we have to use GPUs) memory [50].

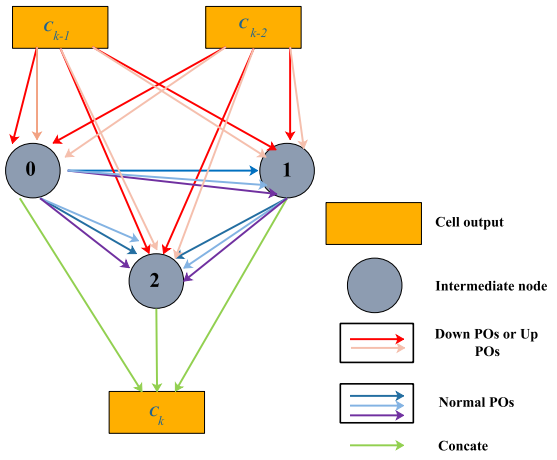


FIGURE 4. An example of Over-Parameterized Cell Architecture. each edge assosicate with *N* candidate operations from different primitive operation sets.

A. OVER-PARAMETERIZED CELL ARCHITECTURE

Given a cell architecture $C(e_1, \cdots, e_E)$ where e_i represents a certain edge in the DAG. Let $O = o_i$ be one of three types of primitive operation set in the above with N candidate operations. Instead of setting each edge associates with definite operation, we set each edge to be a mixed operation that has N parallel paths (As shown in Figure 4), denoted as MixO. Therefore, the over-parameterized cell architecture can be expressed as $C(e_1 = MixO_1, \cdots, e_E = MixO_E)$. The output of a mixed operation MixO is defined based on the output of its N paths:

$$MixO(x) = \sum_{(i=1)}^{N} w_i o_i(x). \tag{1}$$

As shown in Eq 1, w_i indicates the weight of o_i, in One-Shot [50] is constant 1, but in DARTS [20] is calculated by applying softmax to N real-valued architecture parameters $\{\alpha_i\} : e^{\alpha_i} / \sum_j e^{\alpha_j}$. The initial value of α_i is $1/N$.

B. A GPU MEMORY SAVING UPDATE STRATEGY

In the above, an output of each edge is a mixed operation for N candidate primitive operations, which means the output feature maps of all N paths can only be calculated when all operations are loaded into the GPU memory. However, training a compact model only use one path. As such, [20] and [50] roughly need N times GPU memory compared to training a compact model. In this paper, we use a binary gate proposed by Cai *et al.* for learning binarized path instead of N path [22]. The difference between DARTS and binary gate method (denote ProxylessNAS) is that the former update all of the architecture parameters by gradient descent at each step, but the latter only update one of them. As show in Figure 5, When training network weight parameters, we first freeze the architecture parameters and stochastically sample binary gates for each batch of input data. Then the weights parameters of active paths are update via standard gradients descent on the training dataset. When training architecture parameters, the weight parameters are frozen, then we reset the binary gates and update the architecture parameters on the validation set (Figure 5 (a)). These two update steps are performed in an alternative manner. Once the training of architecture parameters is finished, we can then derive the compact architecture by pruning redundant paths. In this work, we simply choose the path with the k ($k = 2$, for our works) highest path weight as inputs. In summary, in this way, regardless of the value of N, only two paths are involved in each update step of the architecture parameters, and thereby the memory requirement is reduced to the same level of training a compact model. It's worth to notice that ProxylessNAS method only considers two paths for updating at each update step, which will result in the trained degree of operation not on the two paths being much lower than the operation on. Therefore, we need more iteration for updating. In addition, a extra time costs at moving feature map not in GPU memory to GPU.

V. EXPERIMENTAL RESULTS

Herein, we report the details of how to implement NAS-Unet. After that we will report the medical image segmentation results on benchmark dataset with our network stacked by best-found cells.

A. NASUNET IMPLEMENTATION DETAILS

We consider the number of intermediate $M = 4$ in both DownSC and UpSC, and the total number of cells $L = 2L_1 = 8$. The DownSC search space is approximate $6^6 + 5^8 = 437281$, and the UpSC search space about $6^6 + 4^8 = 112192$. So the total size of the search space is in the order of 10^{10}, which is much smaller than [25]. Unlike DARTS, we do not follow the practice of doubling the number of filters when halving the height and width of feature map.

We conduct architecture search on the PASCAL VOC 2012 dataset [26] for medical image segmentation. More specifically, we use 480×480 random image crops. We randomly select half of the images in training set as validation set. When we use DARTS search strategy, the batch size is 2 and the architecture search optimization is conducted for a total of 120 epochs. A batch size can be 8 when we use

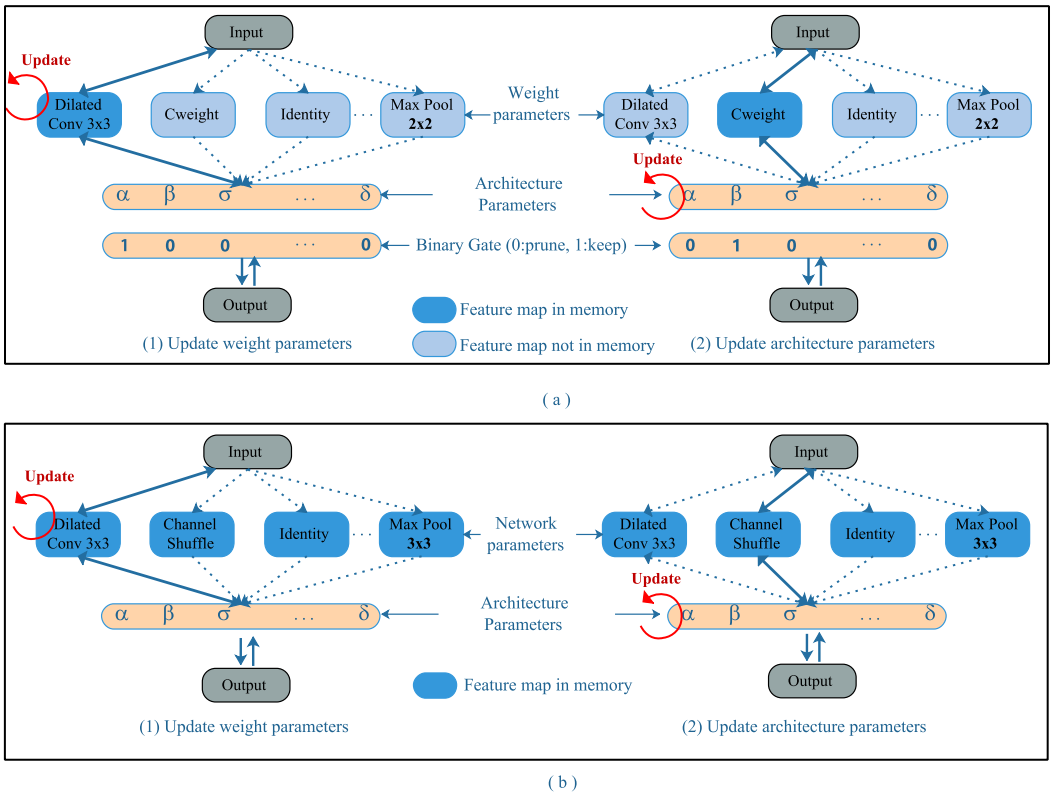


FIGURE 5. The different update strategy between DARTS and ProxylessNAS. (a) only update one path when updating weight parameters, but (b) update all path.

binary gate update strategy, but a much 200 epochs is needed (see section IV-B).

Because this article focuses constructing an efficient cell search space for medical image segmentation, any method of differential search strategy will be work [20], [22], [23], [50]. We want to search proxy cell architectures on a much complicated image dataset (PASCAL VOC 2012 dataset [26]) and transfer it to medical image datasets. So in our experiments, we use DARTS update stratgy. It is also feasible to use the ProxylessNAS, but need search these datasets respectively.

When learning network weight w, we use SGD optimizer with momentum 0.95, cosine learning rate that decays from 0.025 to 0.01, and weight decay 0.0003 [20]. When learning the architecture α, we use Adam optimizer [51] with learning rate 0.0003 and weight decay 0.0001. We empirically found that the Mean Intersection over Union (mIoU) and the Pixels Accuracy (pixAcc) is increase slowly (Figure 6), when we both optimize α from the beginning or follow [25] — starting optimizing after a constant epoch (such as 50). Therefore, we optimize α at the beginning. The entire architecture search optimization takes about 1.5 days on one Titan Pascal GPU. The DownSC and UpSC we searched are shown as Figure 7(a) and (b).

From the Figure 7, We can see that the search processing in our search space more inclined to choose the 'cweight' version operation (include down cweight operation, up cweight operation and cweight operation), since the 'cweight' version operation accounted for a large proportion in both DownSC and UpSC architecture. It's worth noting that the cweight operation replaces the standard skip connection to pass high resolution information (include more accuracy spatial information and high-level semantic information) between the down-sampling and the up-sampling paths (it is exactly the gray arrow indicate transform in Figure 2 (b)). It means that passing high resolution information is not a simple concatenation, but a weighted-concatenation.

B. MEDICAL IMAGE SEGMENTATION RESULTS

To evaluate the performance of NAS-Unet, we use three types of medical image datasets (Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and ultrasound): Promise12 [52], Chaos [53] and NERVE [54] datasets. The weights of all the model were updated by minimizing negative the Dice Similarity Coefficient (DSC) function denoted Dice Loss. We use both the DSC and the Mean Intersection over Union (mIOU) to evaluate model performance. The baseline methods are U-Net [8], FC-Densenet [27]. For the sake of fairness, we re-implement them by Pytorch [55] and use the same data augmentation(we also try to improve the quality of some noisy images [56], [57]). In addition, We use the same Adam optimizer with initial learning rate 3.0e-4 and weight decay 5.0e-5 and train for 200 epochs.

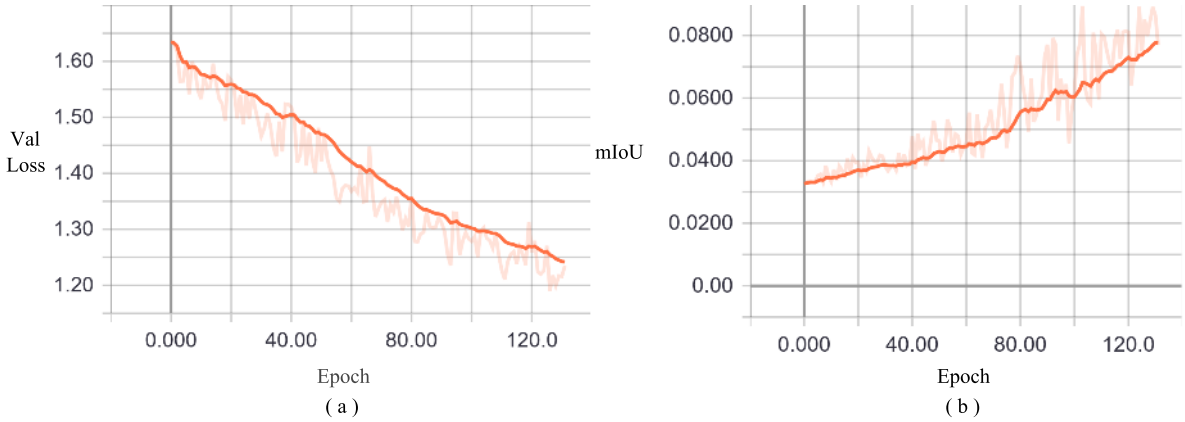


FIGURE 6. The validation loss and mIoU of candidate best architecture network stacked by DownSC and UpSC on PASCAL VOC2012 dataset.

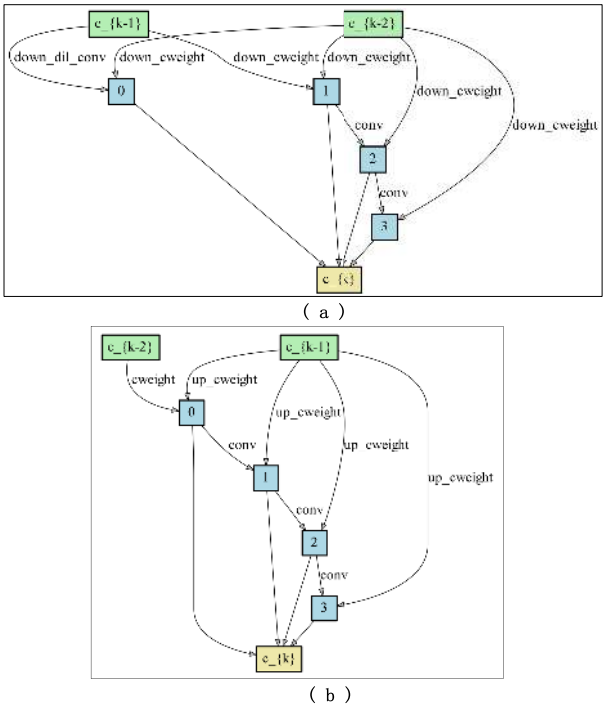


FIGURE 7. The cell architecture searched when the intermediate number is 4 and search on PASCAL VOC2012 dataset. (a) is DownSC architecture and (b) is UpSC architecture. The c_{k-1} and c_{k-2} indicate the previous two cells in DownSC, but the previous cell and horizontal pre-correspondence cell in UpSC.

1) PROMISE12

Promise12 [52] contains 50 training cases and these cases include a transversal T2-weighed MR image of the prostate. Training set has about 1250 images with corresponding labels (only the voxel values 0 and 1). Each 2d MRI slice is resized to dimension 256 × 256 and histogram equalized using contrast limited adaptive histogram equalization (CLAHE). The training dataset is divided into 40 training cases and 10 validation cases. As shown in the Table 2, our models outperform all the baseline methods and without any pre-training. The Train Time and GM represent the train time cost (total days and hours) when batch size is 2 and the GPU memory costs when batch size is 2 repsectively (so as to the following tables).

TABLE 2. Promise12 validation set results.

Model Type	mIoU	DSC	Train Time	GM
U-Net	0.978	0.938	**6h**	**1.3**
FC-Densenet	0.982	0.956	23h	2.7
NasUnet	**0.983**	**0.9737**	8h	1.5

2) CHAOS

Chaos [53] challenge will be held in The IEEE International Symposium on Biomedical Imaging (ISBI) on April 8-11, 2019 Venice, ITALY. The challenge will start at the ISBI conference. Two databases (Abdominal CT and MRI) will be used in five competition and we choose two: Liver Segmentation (CT only) and Segmentation of abdominal organs (MRI only). The first challenge is segmenting liver from computed tomography (CT) data sets, and the second is segmenting four abdominal organs (i.e. liver, spleen, right and left kidneys) from magnetic resonance imaging (MRI) data sets. Each data set in these two databases corresponds to a series of DICOM images belonging to a single patient. The first database contains CT images of 40 different patients. In total, 2874 slices (each slice is 512 × 512) will be provided for training and 1408 slices will be used for tests. The second database includes 120 DICOM data sets from two different MRI sequences (T1-DUAL in phase (40 data sets), out phase (40 data sets) and T2-SPIR (40 data sets)). The data sets have a resolution 256 × 256 and the number of slices is between 26 and 50 (average 36). In total, 1594 slices (532 slice per sequence) will be provided for training and 1537 slices will be used for the tests.

As for now, we use CT images for 2874 slices and MR images for 940 slices for evaluate our model. As show in Table 3, our models achieve a better performance than all baseline methods on both CT images and MR images

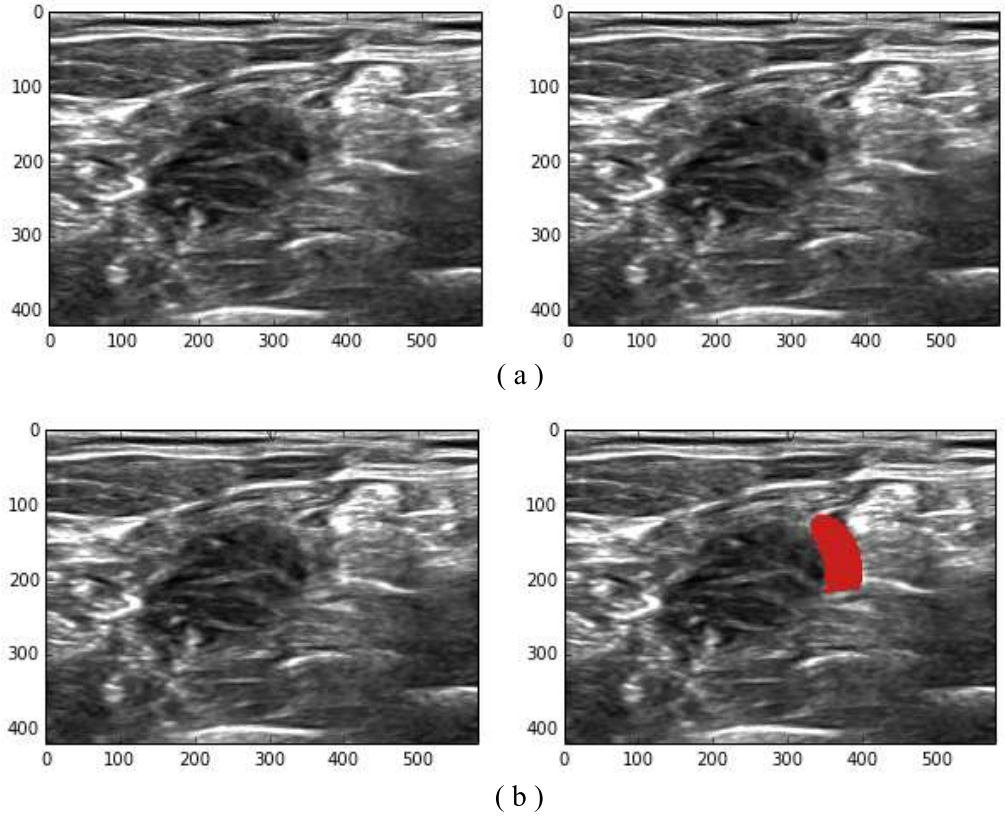


FIGURE 8. An example of two contradictory images, (a) is similar to (b), but without no-empty mask. As with all human-labeled data, potential mistakes in the ground truth is easy to happen.

TABLE 3. Chaos validation set results on CT images and MR images.

Model Type	mIoU	DSC	Train Time	GM
U-Net (CT)	0.982	0.937	**1d-6h**	**2.4**
FC-Densenet (CT)	0.983	0.965	3d-4h	6.84
NasUnet (CT)	**0.985**	**0.974**	1d-15h	3.5
U-Net (MR)	0.46	0.682	**9h**	**1.3**
FC-Densenet (MR)	0.51	0.734	21h	2.7
NasUnet (MR)	**0.54**	**0.76**	11h	1.5

without any pre-training. It is worth to mentioning that in MR images dataset, we have improved the im-balance categories problems (the frequency ratios of these five categories is 1066 : 40 : 3.7 : 4.1) by re-weighting five categories into dice loss function. The batch size also set to 400.

3) ULTRASOUND NERVE

Ultrasound nerve segmentation is a Kaggle challenge in 2016. The task in this competition is to segment a collection of nerves called the Brachial Plexus (BP) in ultrasound images. Some images (about 60% of the training set) do not contain the Brachial Plexus area. The images have a size of 580×420 pixels. There are 5635 training images and 5508 test images (of which 20% were used for public ranking and 80% for the final ranking). The training dataset contains many contradictory images which means that two images that are very similar but with one image having a non-empty mask and the other one has an empty mask (as show in Figure 8). Therefore, we follow the Julian's methods [58] to remove contradictory images by computing a signature for each image. After that the disimilary is the the cosine distance between the two signature vectors of two images. This results in a distance matrix for all the training set images, which is then thresholded to decide which images should be removed. In the end, we kept 4456 training images (out of the 5635). Finally, we random split 0.2 of training images as validation set. As show in Table 4, our models achieve a better performance than all baseline methods without any pre-training.

TABLE 4. Nerve dataset validation set results.

Model Type	mIoU	DSC	Train Time	GM
U-Net	0.989	0.74	**18h**	**2.3**
FC-Densenet	0.989	0.844	2d-3h	6.85
NasUnet	**0.992**	**0.881**	1d-3h	3.4

VI. CONCLUSION

In this paper, we attempt to extend Neural Architecture Search to medical image segmentation. We design three types

of primitive operation set for our search space and search cell-based architecture stacked by DownSC and UpSC. We choose U-like backbone (our search space includes U-net and lots of its variants) to search on and introduce a memory-saving search algorithm (Binary gate) [22] to accelerate the search process. The results of the search, NAS-Unet, is evaludated by training on medical image segmentation datasets from scratch. On Promise12, NAS-Unet significantly outperforms the baseline methods. On Chaos and Ultrasound Nerve NAS-Unet also outperforms than these baseline methods.

REFERENCES

[1] O. C. Eidheim, L. Aurdal, T. Omholt-Jensen, T. Mala, and B. Edwin, "Segmentation of liver vessels as seen in MR and CT images," *Int. Congr. Ser.*, vol. 1268, pp. 201–206, Jun. 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0531513104006132. doi: 10.1016/j.ics.2004.03.184.

[2] A. Bert *et al.*, "An automatic method for colon segmentation in CT colonography," *Computerized Med. Imag. Graph.*, vol. 33, no. 4, pp. 325–331, Jun. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0895611109000226

[3] T. Klinder, J. Ostermann, M. Ehm, A. Franz, R. Kneser, and C. Lorenz, "Automated model-based vertebra detection, identification, and segmentation in CT images," *Med. Image Anal.*, vol. 13, no. 3, pp. 471–482, Jun. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841509000085

[4] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain intelligence: Go beyond artificial intelligence," *Mobile Netw. Appl.*, vol. 23, no. 2, pp. 368–375, Apr. 2018. doi: 10.1007/s11036-017-0932-8.

[5] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep features learning for medical image analysis with convolutional autoencoder neural network," *IEEE Trans. Big Data*, to be published.

[6] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data," *IEEE Syst. J.*, vol. 11, no. 1, pp. 88–95, Mar. 2017.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, Nov. 2015, pp. 234–241.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2012.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2015, pp. 1–14.

[12] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[14] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[15] H. Lu *et al.* (2018). "CONet: A cognitive ocean network." [Online]. Available: https://arxiv.org/abs/1901.06253

[16] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2315–2322, Aug. 2018.

[17] T. Elsken, J. H. Metzen, and F. Hutter. (2018). "Neural architecture search: A survey." [Online]. Available: https://arxiv.org/abs/1808.05377

[18] C. Liu *et al.*, "Progressive neural architecture search," in *Proc. ECCV*, Sep. 2018, pp. 19–34.

[19] L. Xie and A. Yuille, "Genetic CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1388–1397.

[20] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Jun. 2018, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=S1eYHoC5FX

[21] T. Elsken, J. H. Metzen, and F. Hutter "Simple and efficient architecture search for convolutional neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Jan. 2018, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=SySaJ0xCZ

[22] H. Cai, L. Zhu, and S. Han, "Proxylessnas: Direct neural architecture search on target task and hardware," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–13.

[23] R. Luo, F. Tian, T. Qin, and T.-Y. Liu, "Neural architecture optimization," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–12.

[24] L.-C. Chen *et al.*, "Searching for efficient multi-scale architectures for dense image prediction," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–12.

[25] C. Liu *et al.* (2019). "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation." [Online]. Available: https://arxiv.org/abs/1901.02985

[26] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015. doi: 10.1007/s11263-014-0733-5.

[27] J. Simon, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop (CVPR)*, Jul. 2017, pp. 11–19.

[28] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proc. NIPS*, 2012, pp. 2843–2851.

[29] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, Oct. 2016, pp. 424–432.

[30] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.

[31] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*, G. Carneiro *et al.*, Eds. Cham, Switzerland: Springer, 2016, pp. 179–187.

[32] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, 2002.

[33] K. O. Stanley and D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artif. Life* , vol. 15, no. 2, pp. 185–212, Apr. 2009. doi: 10.1162/artl.2009.15.2.15202.

[34] R. Miikkulainen *et al.*, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, R. Kozma, C. Alippi, Y. Choe, and F. C. Morabito, Eds. Amsterdam, The Netherlands: Elsevier, 2018. [Online]. Available: http://nn.cs.utexas.edu/?miikkulainen:chapter18

[35] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Mar. 2017, pp. 1–18. doi: 1611.02167.

[36] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Feb. 2017, pp. 1–16.

[37] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8697–8710.

[38] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Jan. 2018, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=BJQRKzbA-

[39] H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang, "Efficient architecture search by network transformation," in *Proc. AAAI*, Apr. 2018, pp. 1–8. [Online]. Available: http://arxiv.org/abs/1707.04873

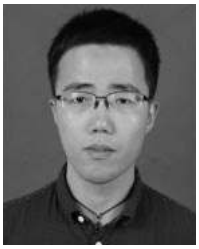
[40] H. Jin, Q. Song, and X. Hu, "Efficient neural architecture search with network morphism," *CoRR*, vol. abs/1806.10282, 2018. [Online]. Available: http://arxiv.org/abs/1806.10282

[41] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing, "Neural architecture search with Bayesian optimisation and optimal transport," in *Proc. NeurIPS*, 2018, pp. 2020–2029.

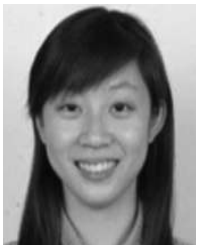
[42] R. Shin, C. Packer, and D. Song, "Differentiable neural network architecture search," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–4.

[43] W. Grathwohl, E. Creager, S. Kamyar, S. Ghasemipour, and R. Zemel, "Gradient-based optimization of neural network architecture," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–6.
[44] E. Real *et al.*, "Large-scale evolution of image classifiers," in *Proc. ICML*, 2017, pp. 2902–2911. [Online]. Available: http://arxiv.org/abs/1703.01041
[45] Z. Zhong, J. Yan, W. Wu, J. Shao, and C.-L. Liu, "Practical block-wise neural network architecture generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2423–2432.
[46] M. Tan, B. Chen, R. Pang, V. Vasudevan, and Q. V. Le. (2018). "Mnasnet: Platform-aware neural architecture search for mobile." [Online]. Available: https://arxiv.org/abs/1807.11626
[47] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[48] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[49] Y. Wu and K. He, "Group normalization," in *Proc. ECCV*, Sep. 2018, pp. 3–19.
[50] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80, 2018, pp. 549–558. [Online]. Available: http://proceedings.mlr.press/v80/bender18a.html
[51] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980
[52] G. Litjens *et al.*, "Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge," *Med. Image Anal.*, vol. 18, no. 2, pp. 359–373, Feb. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841513001734
[53] (2019). *CHAOS—Combined (CT-MR) Healthy Abdominal Organ Segmentation*. [Online]. Available: https://chaos.grand-challenge.org/Combined_Healthy_Abdominal_Organ_Segmentation/
[54] (2016). *Ultrasound Nerve Segmentation Kaggle*. [Online]. Available: https://www.kaggle.com/c/ultrasound-nerve-segmentation
[55] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. NIPS Autodiff Workshop, Future Gradient-Based Mach. Learn. Softw. Techn.*, Long Beach, CA, USA, Dec. 2017.
[56] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 41–50, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790613002644
[57] H. Lu, Y. Li, T. Uemura, H. Kim, and S. Serikawa, "Low illumination underwater light field images reconstruction using deep convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 142–148, May 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17326845
[58] J. Rebetez. (2016). *FHTAGN.Net: Kaggle Ultrasound Nerve Segmentation (25th with 0.70679 Final Ranking)*. [Online]. Available: http://fhtagn.net/prog/2016/08/19/kaggle-uns.html#data

YU WENG received the Ph.D. degree in computer science from the University of Science and Technology Beijing, China, in 2010. He is currently an Associate Professor of computer science with the Information Engineering Department, Minzu University of China. He has published more than 20 conference and journal papers. His current research interests include machine learning, cloud computing, and distributed computing.



TIANBAO ZHOU is currently pursuing the master's degree with the Information Engineering Department, Minzu University of China. His current research interests include machine learning and computer vision.



YUJIE LI received the B.S. degree in computer science and technology from Yangzhou University, Yangzhou, China, in 2009, the M.S. degrees in electrical engineering from the Kyushu Institute of Technology, Kitakyushu, Japan, and Yangzhou University, in 2012, and the Ph.D. degree from the Kyushu Institute of Technology, in 2015.

XIAOYU QIU received the M.S. degree in computer sciences from Shandong Normal University, in 2008. He is currently a Librarian with the Library of Shandong University of Traditional Chinese Medicine. His current research interests include different aspects of pattern recognition, artificial intelligence, and distributed systems.