V-NAS: Neural Architecture Search for Volumetric Medical Image Segmentation

Zhuotun Zhu\(^1\), Chenxi Liu\(^1\), Dong Yang\(^2\), Alan Yuille\(^1\) and Daguang Xu\(^2\)
\(^1\)Johns Hopkins University, \(^2\)NVIDIA Corporation
{zhuotun, alan.l.yuille}@gmail.com cxliu@jhu.edu {dongy, daguangx}@nvidia.com

Abstract

Deep learning algorithms, in particular 2D and 3D fully convolutional neural networks (FCNs), have rapidly become the mainstream methodology for volumetric medical image segmentation. However, 2D convolutions cannot fully leverage the rich spatial information along the third axis, while 3D convolutions suffer from the demanding computation and high GPU memory consumption. In this paper, we propose to automatically search the network architecture tailoring to volumetric medical image segmentation problem. Concretely, we formulate the structure learning as differentiable neural architecture search, and let the network itself choose between 2D, 3D or Pseudo-3D (P3D) convolutions at each layer. We evaluate our method on 3 public datasets, i.e., the NIH Pancreas dataset, the Lung and Pancreas dataset from the Medical Segmentation Decathlon (MSD) Challenge. Our method, named V-NAS, consistently outperforms other state-of-the-arts on the segmentation tasks of both normal organ (NIH Pancreas) and abnormal organs (MSD Lung tumors and MSD Pancreas tumors), which shows the power of chosen architecture. Moreover, the searched architecture on one dataset can be well generalized to other datasets, which demonstrates the robustness and practical use of our proposed method.

1. Introduction

Over the past few decades, medical imaging techniques, e.g., magnetic resonance imaging (MRI), computed tomography (CT), have been widely used to improve the state of preventative and precision medicine. With the emerging of deep learning, great advancement has been made for medical image analysis in various applications, e.g., image classification, object detection, segmentation and other tasks. Among these tasks, organ segmentation is the most common area of applying deep learning to medical imaging [9].

In this work, we focus on the volumetric medical image segmentation. Taking the pancreas and lung tumors segmentation from CT scans as an example as shown in Fig. 1, the main challenges lie in several aspects: 1) the small size of organs with respect to the whole volume; 2) the large variations in location, shape and appearance across different cases; 3) the abnormalities, i.e., the pancreas and lung tumors, can change the texture of surrounding tissues a lot; 4) the anisotropic property along z-axis, which make the automatic segmentation even harder.

To tackle these challenges, many segmentation methods have been proposed in the literature. Starting from handcrafted features, there are methods proposed to use intensity thresholding [1], region growing [16], and deformable models [4], which often suffer from the limited feature representation ability and are less invariant to the large organ variations. With a huge influx of deep learning related methods, fully convolutional neural networks (FCNs), e.g., 2D and 3D FCNs, have become the mainstream methodology in the segmentation area by delivering powerful representation ability and good invariant properties. The 2D FCNs based methods [3, 20, 21, 22, 31] perform the segmentation slice-by-slice from different views, then fuse 2D segmentation output to obtain a 3D result, which is a remedy...
against the ignorance of the rich spatial information. To make full use of the 3D context, 3D FCNs based methods [5, 6, 14, 32] directly perform the volumetric prediction. However, the demanding computation and high GPU consumption of 3D convolutions limit the depth of neural networks and input volume size, which impedes the massive application of 3D convolutions. Recently, the Pseudo-3D (P3D) [17] was introduced to replace 3D convolution $k \times k \times k$ with two convolutions, i.e., $k \times k \times 1$ followed by $1 \times 1 \times k$, which can reduce the number of parameters and show good learning ability in [13, 25] on anisotropic medical images. However, all the aforementioned existing works choose the network structure empirically, which often impose explicit constraints, i.e., either 2D, 3D or P3D convolutions only, or 2D and 3D convolutions are separate from each other. These hand-designed segmentation networks with architecture constraints might not be the optimal solution considering either the ignorance of the rich spatial information for 2D or the demanding computations for 3D.

Drawing inspiration from recent success of Neural Architecture Search (NAS), we take one step further to let the segmentation network automatically choose between 2D, 3D, or P3D convolutions at each layer by formulating the structure learning as differentiable neural architecture search [10, 12]. To the best of our knowledge, we are one of the first to explore the idea of NAS/AutoML in medical imaging fields. Previous work [15] used reinforcement learning and the search restricts to 2D based methods, whereas we use differentiable NAS and search between 2D, 3D and P3D, which is more effective and efficient. Without pretraining, our searched architecture, named V-NAS, outperforms other state-of-the-arts on segmentation of normal Pancreas, the abnormal Lung tumors and Pancreas tumors. In addition, the searched architecture on one dataset can be well generalized to others, which shows the robustness and potential clinical use of our approach.

2. Related Work

2.1. Medical Image Segmentation

The volumetric medical image segmentation has been dominated by deep convolutional neural networks based methods in recent years. [20] proposed the UNet architecture tailored to tackle medical image analysis problems in 2D, which is based on an encoder-decoder framework: the encoder is designed to learn higher and higher level representations while the decoder decompresses compact features into finer and finer resolution to obtain dense predictions. Then, a similar approach was presented by [5] to extend UNet to 3D input. Later on, VNet [14] proposed to incorporate residual blocks penalized by the Dice loss rather than the cross-entropy loss on 3D data, which directly minimizes the used segmentation error measurement. Meanwhile, a few recent works have been proposed to combine 2D and 3D FCNs as a compromise to leverage the advantages of both sides. [26] adopted a 3D FCN by feeding the segmentation predictions of 2D FCNs as input together with 3D images. H-DenseUNet [8] hybridized a 2D DenseUNet for extracting intra-slice features and a 3D counterpart for aggregating inter-slice contexts. Similarly, 2D FCNs and 3D FCNs are not optimized at the same time in [8, 26].

2.2. Neural Architecture Search

Neural Architecture Search (NAS) is the process of automatically discovering better neural architectures than human designs. We summarize the progress in along two dimensions: search algorithm and dataset/task.

Many NAS algorithms belong to either reinforcement learning or evolutionary algorithm. In the reinforcement learning formulation [34], the actions generated by an agent define the network architecture, and the reward is the accuracy on the validation set. In the evolutionary formulation [18], architectures are mutated to produce better offsprings, again measured by validation accuracy. Although these algorithms are general, they are usually computationally costly. To address this problem, [11] progressively expand the search space in order to achieve better sample efficiency. Differentiable NAS approaches [10, 12, 23] utilize sharing among candidate architectures, and are arguably the most efficient family of algorithms to date.

At the same time, we also notice that the earlier papers [19, 27, 33] focused solely on MNIST or CIFAR10 dataset. Later, [11, 18, 34] searched for “transferable architectures” from the smaller CIFAR10 to the much larger ImageNet dataset. More recently, [2, 29] demonstrated the possibility to directly search for architectures on the ImageNet dataset. Finally, [10] extended NAS beyond image classification to semantic segmentation.

This paper sits at the frontier of both dimensions discussed above. We follow the differentiable NAS formulation for its efficiency. In terms of application domain, we directly search on volumetric image segmentation data, which is more demanding and challenging than 2D image labeling.

3. Method

We define a cell to be a fully convolutional module, typically composed of several convolutional (Conv+BN+ReLU) layers, which is then repeated multiple times to construct the entire neural network. Our segmentation network follows the encoder-decoder [14, 20] structure while the architecture for each cell, i.e., 2D, 3D, or P3D, is learned in a differentiable way [10, 12]. The whole network structure is illustrated in Fig. 2, where green Encoder and blue Decoder are in the search space. We start with depicting the detailed network structure in Sec. 3.1, and then describing the search space of green Encoder and blue...
Decoder in Sec. 3.2 and Sec. 3.3, respectively, followed by the optimization and search process in Sec. 3.4.

3.1. Basic Network Architecture

As shown in the upper part of Fig. 2, our task is to train a convolutional neural network model to predict the voxel labels of a CT scan as input. Similar to the state-of-the-art segmentation networks U-Net [20], V-Net [14], 3D U-Net [5] and ResDSN [32], our overall network structure consists of a high-to-low resolution process as a feature extractor, and then recovers the resolution through a low-to-high process to obtain dense predictions. To downsample 3D feature maps from a high resolution to a low resolution, the “Conv-Max Pool Down” in the encoder path is implemented by a conv kernel of 1×1×1 with a stride of [2, 2, 1] followed by a MaxPool 1×1×2 with a stride of [1, 1, 2]. The counterpart along the decoder path is realized by the “Up” module to upsample 3D feature maps from a low resolution to a high resolution. More specifically, the “Up” layer first projects the input feature map to match the number of feature channels of the higher Encoder feature by a 1×1×1 conv, followed by the 3D tri-linear interpolation and element-wise sum with the Encoder feature at a higher resolution. The residual connections from the lower-level encoder to the higher-level decoder aggregate more detailed information to semantic meaningful feature maps to give more accurate dense predictions. A pyramid volumetric pooling module [30] is stacked at the end of the decoder path before the final output layer for fusing multiscale features.

3.2. Encoder Search Space

The set of possible Encoder architecture is denoted as $\mathcal{E}$, which includes the following 3 choices (c.f., Fig. 2 for Encoder $\begin{bmatrix} X \\ Y \end{bmatrix}$): 

$$ \{ \text{Encoder } \begin{bmatrix} 3 \times 3 \times 1 \\ 1 \times 1 \times 1 \end{bmatrix}, \text{Encoder } \begin{bmatrix} 3 \times 3 \times 3 \\ 1 \times 1 \times 1 \end{bmatrix}, \text{Encoder } \begin{bmatrix} 3 \times 3 \times 1 \\ 1 \times 1 \times 3 \end{bmatrix} \} $$

As shown in Eq. 1, we define 3 Encoder cells, consisting of the 2D Encoder $E_0$, 3D Encoder $E_1$, and P3D Encoder $E_2$. 3×3×1 is considered as 2D kernel. The input of the $l$-th cell is denoted as $x^l$ while the output as $x^{l+1}$, which is the input of the $(l+1)$-th cell. Conventionally, the encoder operation $O^l_e \in \mathcal{E}$ in the $l$-th cell is chosen from one of the 3 cells, i.e., either $E_0$, $E_1$, or $E_2$. To make the search space continuous, we relax the categorical choice of a particular Encoder cell operation $O^l_e$ as a softmax over all 3 Encoder convolution cells. By Eq. 2, the relaxed weight choice is parameterized by the encoder architecture parameter $\alpha$, where $\alpha^l_i$ determines the probability of encoder $E_i$ in the $l$-th cell,

$$ x^{l+1} = O^l_e(x^l) \approx \hat{O}^l_e(x^l) $$

$$ \hat{O}^l_e(x^l) = \sum_{i=0}^{2} \frac{\exp(\alpha^l_i)}{\sum_{j=0}^{2} \exp(\alpha^l_j)} E_i(x^l), $$

where $l = 1, \ldots, L$.

3.3. Decoder Search Space

Similarly, the set of possible Decoder architectures is denoted as $\mathcal{D}$, consisting of the following 3 choices (c.f., Fig. 2 for Decoder $\begin{bmatrix} X \\ Y \end{bmatrix}$): 

$$ \{ \text{Decoder } \begin{bmatrix} 3 \times 3 \times 1 \\ 3 \times 3 \times 1 \end{bmatrix}, \text{Decoder } \begin{bmatrix} 3 \times 3 \times 3 \\ 3 \times 3 \times 3 \end{bmatrix}, \text{Decoder } \begin{bmatrix} 3 \times 3 \times 1 \\ 1 \times 1 \times 3 \end{bmatrix} \} $$

As given in Eq. 3, we define 3 Decoder cells, composed of the 2D Decoder $D_0$, 3D Decoder $D_1$, and P3D Decoder $D_2$. The Decoder cell is defined as dense blocks, which shows powerful representation ability in [8, 13]. The input of the $b$-th Decoder cell is denoted as $x^b$ while the output as $x^{b+1}$, which is the input of the $(b+1)$-th Decoder cell. The decoder operation $O^b_d$ of the $b$-th block is chosen from either $D_0$, $D_1$, or $D_2$. As shown in Eq. 4, we also relax the categorical choice of a particular decoder operation $O^b_d$ as a
**Algorithm 1: V-NAS**

Partition the whole labeled dataset $S$ into the disjoint $S_{\text{train}}$, $S_{\text{val}}$ and $S_{\text{test}}$.

Create the mixed operations $\bar{O}_d^i$ and $\bar{O}_d^b$ parametrized by $\alpha_d^i$ and $\beta_d^b$, respectively.

**while training not converged do**

1. Update weights $w$ by descending $\nabla_w L_{\text{train}}(w, \alpha, \beta)$
2. Update $\alpha$ and $\beta$ by descending $\nabla_{\alpha, \beta} L_{\text{val}}(w, \alpha, \beta)$

Replace the relaxed operation $\bar{O}_d^i$ with $O_d^i = E_i, i = \arg\max_k \exp(\alpha_d^i k) / \sum_{j=0}^2 \exp(\alpha_d^j)$;

Replace the relaxed operation $\bar{O}_d^b$ with $O_d^b = D_i, i = \arg\max_k \exp(\beta_d^b k) / \sum_{j=0}^2 \exp(\beta_d^j)$;

Retrain the discretized architecture on the $S_{\text{trainval}}$.

softmax over all $3$ Decoder convolution cells, parameterized by the decoder architecture parameter $\beta$, where $\beta_d^b$ is the choice probability of decoder $D_i$ in the $b$-th dense block,

$$x^{b+1} = O_d^b(x^b) \approx O_d^b(x^b)$$

$$O_d^b(x^b) = \sum_{i=0}^2 \frac{\exp(\beta_d^b k)}{\sum_{j=0}^2 \exp(\beta_d^j)} D_i(x^b), \quad (4)$$

where $b = 1, \ldots, B$.

3.4. Optimization

After relaxation, our goal is to jointly learn the architecture parameters $\alpha$, $\beta$ and the network weights $w$ by the mixed operations. The introduced relaxations in Eq. 2 and Eq. 4 make it possible to design a differentiable learning process optimized by the first-order approximation as in [12]. The algorithm for searching the network architecture parameters is given in Alg. 1. After obtaining optimal encoder and decoder operations $O_d^i$ and $O_d^b$ by discretizing the mixed relaxations $\bar{O}_d^i$ and $\bar{O}_d^b$ through $\arg\max$, we retrain the searched optimal network architectures on the $S_{\text{trainval}} = \{S_{\text{train}}, S_{\text{val}}\}$ and then test it on $S_{\text{test}}$.

4. Experiments

4.1. NAS Implementation Details

In this work, we consider a network architecture with $L=3+4+6+3=16$ and $B=5$, shown as color blocks in Fig. 2. The search space contains $3^L+|B|=3^{21} \approx 10^{10}$ different architectures, which is huge and challenging. The architecture search optimization is conducted for a total of 40,000 iterations. When learning network weights $w$, we adopt the SGD optimizer with a base learning rate of 0.005 with polynomial decay (the power is 0.9), a 0.9 momentum and weight decay of 0.0005. When learning the architecture parameters $\alpha$ and $\beta$, we use Adam optimizer with a learning rate of 0.0003 and weight decay 0.001. Instead of optimizing $\alpha$ and $\beta$ from the beginning when weights $w$ are not well-trained, we start updating them after 20 epochs. After the architecture search is done, we retrain weights $w$ of the optimal architecture from scratch for a total of 40,000 iterations. The searching process takes around 1.2 V100 GPU days for one partition of train, val and test. All our models are trained on one V100 GPU with a customized batch size tuned to take full usage of the GPU memory due to different size input, which is computationally efficient in terms of neural architecture search task brought by the patch input.

In order to evaluate our method in the 4-fold cross-validation manner to fairly compare with existing works, we randomly divide a dataset into 4 folds, where each fold is evaluated once as the $S_{\text{test}}$ while the remaining 3 folds as the $S_{\text{train}}$ and $S_{\text{val}}$ with a train v.s. val ratio as 2 : 1. Therefore, there are in total 4 architecture search processes considering the 4 different $\{S_{\text{train}}, S_{\text{val}}\}$. The searched architecture might be different for each fold due to different $\{S_{\text{train}}, S_{\text{val}}\}$. In this situation, the ultimate architecture is obtained by summing the choice probabilities $(\alpha, \beta)$ across the 4 search processes and then discretize the aggregated probabilities. Finally, we retrain the optimal architecture on each $S_{\text{trainval}}$ and evaluate on the corresponding $S_{\text{test}}$. All our implemented experiments use the same split of cross-validation and adopt Cross-Entropy loss, evaluated by the Dice-Sørensen Coefficient (DSC) formulated as DSC($P$, $Y$) = $\frac{2|P \cap Y|}{|P| + |Y|}$, where $P$ and $Y$ denote for the prediction and ground-truth voxels set for a foreground class, respectively. This evaluation measurement ranges in [0, 1] where 1 means a perfect prediction. We conduct experiments on 3 public datasets, i.e., the NIH Pancreas dataset, the Pancreas and Lung dataset from the Medical Segmentation Decathlon (MSD) Challenge. And ablation studies are done on the NIH Pancreas dataset.

4.2. NIH Pancreas Dataset

We conduct experiments on the NIH pancreas segmentation dataset [21], which contains 82 normal abdominal CT volumes. The size of CT volumes is $512 \times 512 \times D$, where the number of slices $D$ is different for different cases, ranging in [181, 466]. The physical spatial resolution for one voxel is $w \times h \times d$, where $d = 1.0$mm and $w = h$ that ranges from 0.5mm to 1.0mm. For the data pre-processing, we simply truncate the raw Hounsfield Unit (HU) values to be in $[-100, 240]$ and then normalize each raw CT case to have zero mean and unit variance to decrease the data variance caused by the physical processes [7] of medical images. As for the data augmentation in the training phase, we adopt simple yet effective augmentations on all training patches, i.e., rotation (90°, 180°, and 270°) and flip in all three axes (axial, sagittal and coronal), to increase the
number of 3D training examples which can alleviate the scarce of CT scans with expensive human annotations. Our training and testing procedure take patches as input to make more memory for the architecture design, where the training patch size is $96 \times 96 \times 64$ and the testing patch size is $64 \times 64 \times 64$ for the fine scale testing.

As shown in Table 1, our searched optimal architecture outperforms recent state-of-the-arts [26, 28, 32] segmentation algorithms. It is well worth noting that state-of-the-arts [26, 32] adopt a two-stage coarse-to-fine framework to have an extra segmentation network to refine the initial segmentation maps whereas our method outperforms them by only one stage segmentation, which is more efficient and effective. We also obtain the smallest standard deviation and the highest Min DSC, which demonstrates the robustness of our segmentation across all CT cases. Furthermore, we implement the “Mix” baseline that equally initializes all architecture parameters $\alpha$ and $\beta$ and keep them frozen during the training and testing procedures, which basically means the output takes exactly equal weight from 2D, 3D, and P3D in the encoder and decoder paths. Quantitatively, the search mechanism outperforms the “Mix” baseline by 3.17% and 0.79% in terms of the Min and Mean DSC, respectively, which verifies the effectiveness of the searching framework.

In details, we code the searched optimal architecture on the NIH Pancreas dataset by [0 0 0, 0 0 0 1, 2 0 2 0 2 2, 0 0 0] for the 16 Encoder cells, and [0 0 1 0 1] for the 5 Decoder blocks, where “0”, “1” and “2” individually denote for the 2D, 3D and P3D, which are derived from definitions given in the Eq. 1 and Eq. 3. We observe that 2D convolutions are mostly picked up in the beginning for encoders while P3D appears in the intermediate encoders, and 3D convolutions are mostly chosen in the ending decoders. We hypothesize that 2D layer is efficient to extract the within-slice information coupled with the P3D to fuse learned feature maps in the intermediate stage while 3D kernels are effective in the semantic meaningful layers close to the output prediction.

We visualize two slices randomly chosen from three NIH pancreas cases as shown in Fig. 3. For the Case “#72” with a DSC of 90.96%, the pancreas appearance and boundary are well-captured and distinguished from its surroundings. For the Case “#81” with a DSC close to the “Mean DSC”, the pancreas regions are generally predicted well though with some minor under-estimations near the head. As for the Case “#42” with the min DSC, the “VNBS” makes mistakes in the condition where the surrounding tissues are very complicate and the boundaries are ambiguous.

<table>
<thead>
<tr>
<th>Method</th>
<th>Categorization</th>
<th>Mean DSC</th>
<th>Max DSC</th>
<th>Min DSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>V-NAS (Ours)</td>
<td>Search</td>
<td>85.15 ± 4.55%</td>
<td>91.18%</td>
<td>70.37%</td>
</tr>
<tr>
<td>Baseline (Ours)</td>
<td>Mix</td>
<td>84.36 ± 5.25%</td>
<td>91.29%</td>
<td>67.20%</td>
</tr>
<tr>
<td>Xiu et al. [26]</td>
<td>2D/3D</td>
<td>84.63 ± 5.07%</td>
<td>91.57%</td>
<td>61.58%</td>
</tr>
<tr>
<td>Zhu et al. [32]</td>
<td>3D</td>
<td>84.59 ± 4.86%</td>
<td>91.45%</td>
<td>69.62%</td>
</tr>
<tr>
<td>Yu et al. [28]</td>
<td>2D</td>
<td>84.50 ± 4.97%</td>
<td>91.02%</td>
<td>62.81%</td>
</tr>
<tr>
<td>Cai et al. [3]</td>
<td>2D</td>
<td>82.40 ± 6.70%</td>
<td>90.10%</td>
<td>60.00%</td>
</tr>
<tr>
<td>Zhou et al. [31]</td>
<td>2D</td>
<td>82.37 ± 5.68%</td>
<td>90.85%</td>
<td>62.43%</td>
</tr>
<tr>
<td>Dou et al. [6]</td>
<td>3D</td>
<td>82.25 ± 5.91%</td>
<td>90.32%</td>
<td>62.53%</td>
</tr>
<tr>
<td>Roth et al. [22]</td>
<td>2D</td>
<td>78.01 ± 8.20%</td>
<td>88.65%</td>
<td>34.11%</td>
</tr>
<tr>
<td>Roth et al. [21]</td>
<td>2D</td>
<td>71.42 ± 10.11%</td>
<td>86.29%</td>
<td>23.99%</td>
</tr>
</tbody>
</table>

Table 1. Comparison with other state-of-the-arts on the NIH Pancreas dataset evaluated by the 4-fold cross validation. Our one-stage segmentation network outperforms two-stage coarse-to-fine state-of-the-arts [26, 32]. The “Categorization” column categorizes each method by whether the segmentation method is based on 2D, 3D, or by the dynamic searching in our proposed method. The architecture searched on the NIH Pancreas dataset is coded as [0 0 0, 0 0 0 1, 2 0 2 0 2 2, 0 0 0] for the 16 Encoder cells, and [0 0 1 0 1] for the 5 Decoder blocks.

4.3. MSD Lung Tumors

We also evaluate our framework on the Lung Tumors dataset from the Medical Segmentation Decathlon Challenge (MSD) [24], which contains 64 training and 32 testing CT scans. It is aimed for the segmentation of a small target (lung tumors) in a large image, where only the lung cancers are labelled and to be segmented. Since the testing label is not available and the challenge panel is currently closed, we report and compare results of 4-fold cross-validation on the available 64 training set. The truncation range is set to be $[-1000, 1000]$ to cover almost all the lung HU values in the data pre-processing while the data augmentation is the same as mentioned in Sec. 4.2. More specifically, the patch size is set to be $64 \times 64 \times 64$ for the training and testing on MSD Lung Tumors dataset.

As given in Table 2, our method (V-NAS-Lung) beats 3D UNet [5] and VNet [14] by a large margin, at least 2.33% in terms of the “Mean DSC”. The search process consistently outperforms the “Mix” version which takes equally the 2D, 3D and P3D as a fixed configuration. It is worth noting that the “Max DSC” of ours falls behind 3D UNet and VNet. We conjecture that since the overall network architecture is configured by the average choice probabilities of parameters $\alpha$ and $\beta$ on 4 splits, our method tends to stably achieve the
best overall segmentation performance, which is consistent with the much higher “Median DSC”. More specifically, the searched architecture on Lung tumors is coded as $[0 \ 0 \ 0, \ 1 \ 2 \ 0 \ 1, \ 2 \ 1 \ 2 \ 0 \ 0 \ 0, \ 0 \ 0 \ 0]$ and $[0 \ 0 \ 2 \ 1 \ 1]$. 

To take one step further, we report results of directly training the searched optimal architecture from the NIH Pancreas dataset (V-NAS-NIH) on the MSD Lung tumors dataset from scratch. The searched architecture generalizes well and achieves better performance than “Mix”, 3D UNet and VNet. By comparing the two searched architectures from NIH Pancreas and MSD Lung Tumors datasets, we find that the two optimal architectures V-NAS-Lung and V-NAS-NIH share $68\%$ ($11$ out of $16$ Encoder cells) for the encoder path and $60\%$ ($3$ out of $5$ Decoder blocks) for the decoder path. The good property of transferring the network architecture searched on one dataset to another makes it possible for us to train the network architecture searched on a fairly big dataset with rich annotations to a small dataset with scarce annotations. We have not shown the “Min DSC” in the table since all approaches miss some lung tumors considering the lowest DSC to be $0$, which shows that small lung tumors segmentation is a quite challenging task.

4.4. MSD Pancreas Tumors

Different from the NIH normal pancreas dataset, the MSD Pancreas Tumors dataset is labeled with both pancreatic tumors and normal pancreas regions. The original training set contains 282 portal venous phase CT cases, which are randomly split into 4 folds in our experiment, where each fold has its own training, validation and testing set and the final segmentation performance is reported on the average of 4 folds. Since the resolution along $z$-axis of this dataset is very low and number of slices can be as small as 37, the resolution of all cases on MSD Pancreas Tumors dataset are first re-sampled to an isotropic volume resolution of $d = 1.0\text{mm}$ for each axis. Then the pre-processing and data augmentation is the same as Sec. 4.2. The patch size is set to be $64 \times 64 \times 64$ for both training and testing phases. Due to variant shapes and locations of tumors, the tumor segmentation is much more challenging and clinically important than the normal pancreas segmentation task since the early detection of pancreatic tumors can save lives.

As shown in Table 3, our searched architecture consistently outperforms 3D UNet and VNet, especially the pancreas tumors DSC delivers an improvement of at least $1.79\%$, which is regarded as a fairly good advantage. The $7.68\%$ improvement over the manual “Mix” setting on the pancreas tumors consistently proves the advantage of the architecture search in the volumetric image segmentation domain. In details, the searched architecture on this dataset is coded as $[0 \ 2 \ 2, \ 2 \ 0 \ 0 \ 0, \ 2 \ 2 \ 1 \ 2 \ 1 \ 1, \ 0 \ 1 \ 1]$ and $[1 \ 0 \ 2 \ 0 \ 1]$, by which we observe there are more P3D and 3D convolutions selected compared with the searched optimal architecture from the NIH normal Pancreas dataset. We hypothesize that the between-slice information is very important to detect abnormalities since radiologists need to scroll up and down when reading CT scans to help the diagnosis.

We illustrate the visualization results of different methods.
Table 2. Performance of different methods on the MSD Lung tumors dataset evaluated by the same 4-fold cross validation. The searched architecture on Lung tumors is coded as [0 0 0, 1 0 1, 2 1 0 0 0, 0 0 0] and [0 0 2 1 1]. It is worth noting that the searched architecture on the NIH dataset is well generalized to the Lung tumors dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Categorization</th>
<th>Mean DSC</th>
<th>Max DSC</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>V-NAS-Lung (Ours)</td>
<td>Search</td>
<td>55.27 ± 31.18%</td>
<td>90.32%</td>
<td>66.95%</td>
</tr>
<tr>
<td>V-NAS-NIH (Ours)</td>
<td>Search</td>
<td>54.01 ± 31.39%</td>
<td>92.17%</td>
<td>68.93%</td>
</tr>
<tr>
<td>Baseline (Ours)</td>
<td>Mix</td>
<td>52.27 ± 31.40%</td>
<td>89.57%</td>
<td>61.71%</td>
</tr>
<tr>
<td>3D UNet</td>
<td>3D</td>
<td>52.94 ± 31.28%</td>
<td>93.58%</td>
<td>61.08%</td>
</tr>
<tr>
<td>VNet</td>
<td>3D</td>
<td>50.47 ± 31.37%</td>
<td><strong>93.85</strong>%</td>
<td>57.82%</td>
</tr>
</tbody>
</table>

Table 3. Performance of different methods on the MSD Pancreas tumors dataset evaluated by the same 4-fold cross validation. The results are given on the normal pancreas regions and pancreatic tumors, respectively. The searched architecture on Pancreas tumors dataset is coded as [0 2 2, 2 0 0, 2 2 1 2 1, 0 1 1] and [1 0 2 0 1].

<table>
<thead>
<tr>
<th>Method</th>
<th>Categor.</th>
<th>Pancreas Tumors DSC</th>
<th>Pancreas DSC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>Max</td>
</tr>
<tr>
<td>V-NAS (Ours)</td>
<td>Search</td>
<td>37.78 ± 32.12%</td>
<td>92.49%</td>
</tr>
<tr>
<td>Baseline (Ours)</td>
<td>Mix</td>
<td>30.10 ± 31.40%</td>
<td>92.95%</td>
</tr>
<tr>
<td>3D UNet</td>
<td>3D</td>
<td>35.61 ± 32.20%</td>
<td><strong>93.66</strong>%</td>
</tr>
<tr>
<td>VNet</td>
<td>3D</td>
<td>35.99 ± 31.27%</td>
<td>92.95%</td>
</tr>
</tbody>
</table>

Table 4. Performance (“Mean DSC”) of different encoder and decoder configurations on NIH dataset evaluated by the same 4-fold cross validation. The architecture is manually set with different choices from 2D, 3D and P3D. Ours obtains 85.15% in Table 1.

<table>
<thead>
<tr>
<th>Encoder\Decoder</th>
<th>3D</th>
<th>2D</th>
<th>P3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D</td>
<td>84.09%</td>
<td>83.77%</td>
<td>84.20%</td>
</tr>
<tr>
<td>2D</td>
<td>83.66%</td>
<td>83.29%</td>
<td>84.08%</td>
</tr>
<tr>
<td>P3D</td>
<td>84.32%</td>
<td>84.69%</td>
<td><strong>84.75</strong>%</td>
</tr>
</tbody>
</table>

5. Discussions

To further verify the advantage of automatically selecting among 2D, 3D and P3D convolution layers via the neural architecture search, we conduct ablation studies on manually choosing types in encoder and decoder paths to be purely either 2D, 3D or P3D on NIH Pancreas and MSD Lung Tumors datasets in this section.

5.1. Manual Setting on NIH Pancreas Dataset

As shown in Table 4, we manually configure the architecture of Encoder and Decoder, where we train and test all configurations on the same 4-fold cross validation. More specifically, all Encoders are set to be one type (2D, 3D, or P3D), and the same strategy is applied to the Decoders. Each row denotes the pure categorical choice for the Encoder cells while the column for the Decoder. We can find that 2D, 3D, and P3D kernels contribute differently to the segmentation by the experimental results. The P3D as Encoder and the P3D as Decoder achieve a mean DSC of 84.75% to outperform all other manual configurations. It is conjectured that the pure P3D takes most advantage of the anisotropic data annotation of the NIH dataset, where the annotation was done slice-by-slice along the z-axis. The different capability of learning semantic features between 2D, 3D and P3D for the dense volumetric image segmentation problem drives us to naturally formulate it to be a neural architecture search task. As it turns out, the automatic selection among the 2D, 3D and P3D delivers the best performance with a mean DSC of 85.15% in Table 1.

5.2. Manual Setting on MSD Lung Tumors Dataset

On the MSD Lung Tumors dataset, we also report the manual architecture settings of 3D/3D, 2D/2D and P3D/P3D, e.g., “3D/3D” stands for the configuration of only choosing 3D in both Encoder and Decoder cells.
Figure 4. The visualization illustration of predicted segmentation for “VNAS”, “Mix”, “3D UNet” and “VNet” on the MSD Pancreas Tumors dataset, which is the most challenging task among our segmentation tasks. Each row denotes a slice visualization from one case, and the specific cases numbers are “309”, “021”, “069” and “329” from top to bottom rows. The masked blue and red regions denote for the normal pancreas regions and tumor regions, respectively. Best viewed in color.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean DSC</th>
<th>Max DSC</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D/3D</td>
<td>53.74 ± 30.66%</td>
<td>91.44%</td>
<td>60.55%</td>
</tr>
<tr>
<td>2D/2D</td>
<td>52.01 ± 31.50%</td>
<td>92.58%</td>
<td>63.27%</td>
</tr>
<tr>
<td>P3D/P3D</td>
<td>51.48 ± 32.46%</td>
<td>92.40%</td>
<td>63.89%</td>
</tr>
</tbody>
</table>

Table 5. Performance of different encoder and decoder configurations on MSD Lung Tumors evaluated by the same 4-fold cross validation. The architecture is manually configured with different choices of 2D, 3D and P3D. Ours obtains 55.27% in Table 2.

As given in Table 5, the “3D/3D” manual configuration achieves the best “Mean DSC” of 53.74 ± 30.66%. We suspect that the lung cancers are located inside the lung organs, which needs the rich spatial context to predict the abnormality. Consistent with what we observe in Sec. 5.1, the neural architecture search idea outperforms all manual configurations to obtain a best mean DSC of 55.27 ± 31.18% with an advantage of 1.53% over the “3D/3D” in Table 2.

6. Conclusion

We propose to integrate neural architecture search into volumetric segmentation networks to automatically find optimal network architectures between 2D, 3D, and Pseudo-3D convolutions. The search process is computationally efficient and effective. By searching in the relaxed continuous space, our method outperforms state-of-the-arts on both normal and abnormal organ segmentation tasks. Moreover, the searched architecture on one dataset can be well generalized to another one. In the future, we would like to expand the search space to hopefully find even better segmentation networks and reduce the computations.

Acknowledgements We thank Huiyu Wang for his insightful discussions and suggestions.
References


