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

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 task 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.

Keywords: Volumetric Image Segmentation · Neural Architecture Search

1 Introduction

Over the past few decades, medical imaging techniques, *e.g.*, magnetic resonance imaging (MRI), computed tomography (CT), and X-ray, have been widely used to improve the state of preventative and precision medicine. Coupled with the emerging of deep learning, great advancement has been witnessed 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 [4].

In this work, we focus on the volumetric medical image segmentation. Taking the pancreas and lung tumors segmentation from CT scans as an example, 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 lung and pancreas 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, handcrafted features based methods often suffer from the limited feature representation ability. 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 [1,11,12,13,16] 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 [2,8,17] 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. Meanwhile, a few recent works have been proposed to combine 2D and 3D FCNs as a compromise to leverage the advantages of both sides. [15] adopted a 3D FCN by feeding the segmentation predictions of 2D FCNs as input together with 3D images. H-DenseUNet [3] hybridized a 2D DenseUNet for extracting intra-slice features and a 3D counterpart for aggregating inter-slice contexts. However, 2D FCNs and 3D FCNs are not optimized at the same time in [3,15]. Recently, the Pseudo-3D (P3D) [10] 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 [7,14] 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** [5,6]. To the best of our knowledge, we are one of the first to explore the idea of NAS/AutoML in medical imaging field. Previous work [9] used reinforcement learning and the search restricts to 2D based methods, whereas we use differentiable NAS and search between 2D, 3D and P3D. 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 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 [8,11] structure while the architecture for each cell, *i.e.*, 2D, 3D,

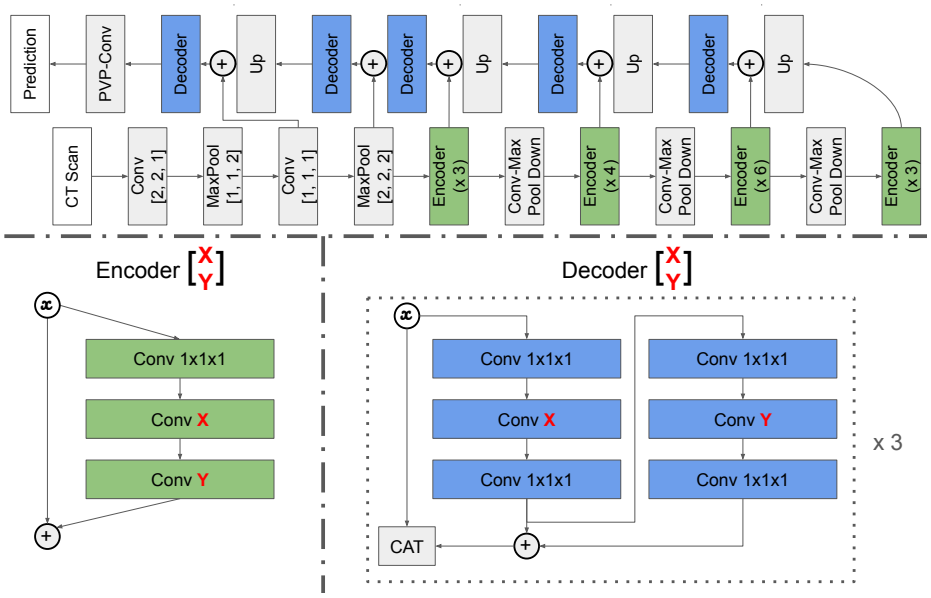

Fig. 1. The segmentation network architecture. Each Encoder cell and Decoder cell has two candidate conv layers X and Y, which are chosen between 2D, 3D, or P3D. The Encoder along the encoding path is repeated by $3, 4, 6, 3$ times, respectively. The encoder path is designed based on ResNet-50, while the decoder path takes advantage of dense block and pyramid volumetric pooling (PVP) [7]. The first two convolutional layers adopt a kernel size $7\times7\times1$ with stride $[2, 2, 1]$ and $1\times1\times3$ with stride $[1, 1, 1]$, which are manually set to efficiently extract low-level features at the beginning [7].

or P3D, is learned in a differentiable way [5,6]. The whole network structure is illustrated in Fig. 1, where green Encoder and blue Decoder are in the search space. Similar to [5,6], we start with describing the search space of Encoder and Decoder of network, followed by optimization and search process.

Encoder Search Space The set of possible Encoder architecture is denoted as $\mathcal{E}$, which includes the following 3 choices (*c.f.*, Fig.1 for Encoder $\begin{bmatrix}\mathrm{X}\\ \mathrm{Y}\end{bmatrix}$):

$$\mathcal{E} = \{\underbrace{\text{Encoder}\begin{bmatrix}3\times3\times1\\1\times1\times1\end{bmatrix}}_{E_0\text{: 2D}}, \underbrace{\text{Encoder}\begin{bmatrix}3\times3\times3\\1\times1\times1\end{bmatrix}}_{E_1\text{: 3D}}, \underbrace{\text{Encoder}\begin{bmatrix}3\times3\times1\\1\times1\times3\end{bmatrix}}_{E_2\text{: P3D}}\} \quad (1)$$

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. $3\times3\times1$ 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 $O_e^l \in \mathcal{E}$ 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_e^l as a softmax over all 3 Encoder convolution cells. By Eq. 2, the

Algorithm 1: V-NAS

Partition the whole labeled dataset $\mathcal{S}$ into the **disjoint** $\mathcal{S}_{\text{train}}$, $\mathcal{S}_{\text{val}}$ and $\mathcal{S}_{\text{test}}$
Create the mixed operations $\bar{O}_e^l$ and $\bar{O}_d^b$ parametrized by α_i^l and β_i^b, respectively
while *training not converged* **do**
1. Update weights w by descending $\nabla_w \mathcal{L}_{\text{train}}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$
2. Update $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by descending $\nabla_{\boldsymbol{\alpha},\boldsymbol{\beta}} \mathcal{L}_{\text{val}}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$

Replace $\bar{O}_e^l$ with $O_e^l = E_i, i = \texttt{argmax}_k \exp(\alpha_k^l) / \sum_{j=0}^{2} \exp(\alpha_j^l)$
Replace $\bar{O}_d^b$ with $O_d^b = D_i, i = \texttt{argmax}_k \exp(\beta_k^b) / \sum_{j=0}^{2} \exp(\beta_j^b)$

relaxed weight choice is parameterized by the encoder architecture parameter $\boldsymbol{\alpha}$, where α_i^l determines the probability of encoder convolution E_i in the l-th cell.

$$x^{l+1} = O_e^l(x^l) \approx \bar{O}_e^l(x^l) = \sum\nolimits_{i=0}^{2} \frac{\exp(\alpha_i^l)}{\sum_{j=0}^{2} \exp(\alpha_j^l)} E_i(x^l), l = 1, \ldots, L. \quad (2)$$

Decoder Search Space Similarly, the set of possible Decoder architectures is denoted as $\mathcal{D}$, consisting of the following 3 choices (*c.f.*, Fig. 1 for Decoder $\begin{bmatrix} \text{X} \\ \text{Y} \end{bmatrix}$):

$$\mathcal{D} = \{\underbrace{\text{Decoder} \begin{bmatrix} 3 \times 3 \times 1 \\ 3 \times 3 \times 1 \end{bmatrix}}_{D_0\text{: 2D}}, \underbrace{\text{Decoder} \begin{bmatrix} 3 \times 3 \times 3 \\ 3 \times 3 \times 3 \end{bmatrix}}_{D_1\text{: 3D}}, \underbrace{\text{Decoder} \begin{bmatrix} 3 \times 3 \times 1 \\ 1 \times 1 \times 3 \end{bmatrix}}_{D_2\text{: P3D}}\} \quad (3)$$

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 [3,7]. 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_d^b 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_d^b as a softmax over all 3 Decoder convolution cells, parameterized by the decoder architecture parameter $\boldsymbol{\beta}$, where β_i^b is the choice probability of decoder convolution cell D_i in the b-th dense block.

$$x^{b+1} = O_d^b(x^b) \approx \bar{O}_d^b(x^b) = \sum\nolimits_{i=0}^{2} \frac{\exp(\beta_i^b)}{\sum_{j=0}^{2} \exp(\beta_j^b)} D_i(x^b), b = 1, \ldots, B. \quad (4)$$

Optimization After relaxation, our goal is to jointly learn the architecture parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ 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 [6]. The algorithm for searching the network architecture parameters is given in Alg. 1. After obtaining optimal encoder and decoder operations O_e^l and O_d^b by discretizing the mixed relaxations $\bar{O}_e^l$ and $\bar{O}_d^b$ through `argmax`, we retrain the searched optimal network architectures on the $\mathcal{S}_{\text{trainval}} = \{\mathcal{S}_{\text{train}}, \mathcal{S}_{\text{val}}\}$ and then test it on $\mathcal{S}_{\text{test}}$.

3 Experiments

3.1 Neural Architecture Search Implementation Details

We consider a network architecture with L=3+4+6+3=16 and B=5, shown as color blocks in Fig. 1. The search space contains $3^{L+B}=3^{21}\approx10^{10}$ 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 0.05 with polynomial decay (the power is 0.9), a 0.9 momentum and weight decay of 0.0005. When learning the architecture parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we use Adam optimizer with a learning rate of 0.0003 and weight decay 0.001. Instead of optimizing $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ from the beginning when w are not well-trained, we start updating them after 20 epochs. After the architecture search is done, we retrain the weights w of the optimal architecture from scratch for a total of 40,000 iterations. The searching process only takes 1.2 V100 GPU days for one partition of train, val and test. 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 $\mathcal{S}_{\text{test}}$ while the remaining 3 folds as the $\mathcal{S}_{\text{train}}$ and $\mathcal{S}_{\text{val}}$ with a train *v.s.* val ratio as 2 : 1. Therefore, there are in total 4 architecture search processes considering the 4 different $\{\mathcal{S}_{\text{train}}, \mathcal{S}_{\text{val}}\}$. The searched architecture might be different for each fold due to different $\{\mathcal{S}_{\text{train}}, \mathcal{S}_{\text{val}}\}$. In this situation, the ultimate architecture is obtained by summing the choice probabilities ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) across the 4 search processes and then discretize the aggregated probabilities. Finally, we retrain the optimal architecture on each $\mathcal{S}_{\text{trainval}}$ and evaluate on the corresponding $\mathcal{S}_{\text{test}}$. All experiments use the same split of cross-validation and adopts Cross-Entropy loss, evaluated by the Dice-Sørensen Coefficient (DSC).

3.2 NIH Pancreas Dataset

We conduct experiments on the NIH pancreas segmentation dataset [12], which contains 82 normal abdominal CT volumes. Following [17] for the data preprocessing and data augmentation, we truncate the raw intensity values to be in $[-100, 240]$; then normalize each CT case to have zero mean and unit variance. Our training and testing procedure take patches as input to make more memory for the architecture design, where the training patch size is 96×96×64 and the testing patch size is 64×64×64 for the fine scale testing.

First of all, we manually choose the architecture of Encoder and Decoder cells. As shown in Table 1, 2D, 3D, and P3D kernels contribute differently to the segmentation. The first row denotes the pure categorical choice for the Encoder cells while the second row for the Decoder. The P3D as Encoder and the P3D as Decoder outperforms all the other manual configurations. It is conjectured that the P3D takes advantage of the anisotropic data annotation of the NIH dataset, where the annotation was done slice-by-slice along the z-axis.

As shown in Table 2, our searched optimal architecture outperforms state-of-the-arts segmentation algorithms. It is worth noting that state-of-the-arts [15,17] adopt the two-stage coarse-to-fine framework whereas our method outperforms

Encoder	3D			2D			P3D		
Decoder	3D	2D	P3D	3D	2D	P3D	3D	2D	P3D
Mean DSC	84.09%	83.77%	84.20%	83.66%	83.29%	84.08%	84.32%	84.69%	**84.75**%

Table 1. Performance of manual settings of different encoder and decoder on the NIH.

them by one stage segmentation. We also obtain the smallest standard deviation and the highest Min DSC, which demonstrates the robustness of our segmentation. Furthermore, we implement the "Mix" baseline that equally initializes all architecture parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and keep them frozen during the training and testing, which basically means the output takes exactly equal weight from 2D, 3D, and P3D in the encoder and decoder paths. The search mechanism outperforms the "Mix" baseline by 3.17% and 0.79% in terms the Min and Mean DSC, respectively, which verifies the effectiveness of the searching framework.

Method	Categorization	Mean DSC	Max DSC	Min DSC
V-NAS	Search	$\mathbf{85.15 \pm 4.55}$%	91.18%	**70.37**%
Baseline	Mix	84.36 ± 5.25%	91.29%	67.20%
Xia *et al.* [15]	2D/3D	84.63 ± 5.07%	**91.57**%	61.58%
Zhu *et al.* [17]	3D	84.59 ± 4.86%	91.45%	69.62%
Cai *et al.* [1]	2D	82.40 ± 6.70%	90.10%	60.00%
Zhou *et al.* [16]	2D	82.37 ± 5.68%	90.85%	62.43%
Roth *et al.* [13]	2D	78.01 ± 8.20%	88.65%	34.11%

Table 2. Performance of different methods on the NIH dataset evaluated by the 4-fold cross validation. The architecture searched on NIH 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.

3.3 Medical Segmentation Decathlon Lung Tumors

We also evaluate our framework on the Lung tumor dataset from the MSD Challenge, which contains 64 training and 32 testing CT scans. It is aimed for the segmentation of a small target (tumor) in a large image. 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 64 training set. The patch size is set to be 64×64×64 for training and testing.

In Table 3, our method (V-NAS-Lung) beats all other approaches by at least 1.53% in terms of the mean DSC, including the 3D UNet [2] and VNet [8], the manual architectures of 3D/3D, 2D/2D and P3D/P3D, where "3D/3D" stands for 3D Encoder and 3D Decoder cell. The search process consistently outperforms the "Mix" version which takes equally the 2D, 3D and P3D. Furthermore, we report results of directly training the searched architecture on NIH dataset (V-NAS-NIH) on the Lung tumors dataset. The searched architecture generalizes well, and achieves better performance than other baselines. By looking closer into the two searched architectures from NIH Pancreas and MSD Lung, we find

that the two optimal architectures share 68% (11 out of 16 Encoder cells) for the encoder path and 60% (3 out of 5 Decoder blocks) for the decoder path. All of those approaches miss some lung tumors considering the lowest DSC to be 0, which shows that small lung tumors segmentation is a challenging task.

Method	Categorization	Mean DSC	Max DSC	Median
V-NAS-Lung	Search	$\mathbf{55.27 \pm 31.18}\%$	90.32%	66.95%
V-NAS-NIH	Search	$54.01 \pm 31.39\%$	92.17%	**68.93**%
Baseline	Mix	$52.27 \pm 31.40\%$	89.57%	61.71%
3D/3D	3D	$53.74 \pm 30.66\%$	91.44%	60.55%
2D/2D	2D	$52.01 \pm 31.50\%$	92.58%	63.27%
P3D/P3D	P3D	$51.48 \pm 32.46\%$	92.40%	63.89%
UNet	3D	$52.94 \pm 31.28\%$	93.58%	61.08%
VNet	3D	$50.47 \pm 31.37\%$	**93.85**%	57.82%

Table 3. 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 2 0 1, 2 1 2 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.

3.4 Medical Segmentation Decathlon Pancreas Tumors

The MSD Pancreas Tumors dataset is labeled with both normal pancreas regions and pancreatic tumors. The original training set contains 282 portal venous phase CT cases. The patch size is set to be $64 \times 64 \times 64$ for training and testing. As shown in Table 4, our searched architecture consistently outperforms the UNet and VNet, especially the pancreas tumors DSC delivers an improvement of around 2%, which is regarded as a fairly significant advantage. The 7.68% improvement on the pancreas tumors proves the advantage of the architecture search over the manual "Mix" setting in the volumetric image segmentation field.

Method	Categor.	Pancreas DSC			Pancreas Tumors DSC		
		Mean	Max	Min	Mean	Max	Median
V-NAS	Search	$\mathbf{79.94 \pm 8.85}\%$	**92.24**%	36.99%	$\mathbf{37.78 \pm 32.12}\%$	92.49%	**38.32**%
Baseline	Mix	$78.41 \pm 9.40\%$	92.21%	40.08%	$30.10 \pm 31.40\%$	92.95%	18.05%
UNet	3D	$79.20 \pm 9.43\%$	91.95%	**40.72**%	$35.61 \pm 32.20\%$	**93.66**%	32, 23%
VNet	3D	$79.01 \pm 9.44\%$	92.05%	28.15%	$35.99 \pm 31.27\%$	92.95%	35.91%

Table 4. 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 0, 2 2 1 2 1 1, 0 1 1] and [1 0 2 0 1].

4 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. 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.

References

1. Cai, J., Lu, L., Xie, Y., Xing, F., Yang, L.: Improving deep pancreas segmentation in CT and MRI images via recurrent neural contextual learning and direct loss function. In: MICCAI (2017)
2. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D u-net: learning dense volumetric segmentation from sparse annotation. MICCAI (2016)
3. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.W., Heng, P.A.: H-denseunet: Hybrid densely connected unet for liver/tumor segmentation from ct volumes. TMI (2018)
4. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. MIA (2017)
5. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A., Fei-Fei, L.: Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In: CVPR (2019)
6. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
7. Liu, S., Xu, D., Zhou, S.K., Pauly, O., Grbic, S., Mertelmeier, T., Wicklein, J., Jerebko, A., Cai, W., Comaniciu, D.: 3d anisotropic hybrid network: Transferring convolutional features from 2d images to 3d anisotropic volumes. MICCAI (2018)
8. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV (2016)
9. Mortazi, A., Bagci, U.: Automatically designing cnn architectures for medical image segmentation. In: International Workshop on MLMI (2018)
10. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: ICCV (2017)
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
12. Roth, H.R., Lu, L., Farag, A., Shin, H.C., Liu, J., Turkbey, E.B., Summers, R.M.: Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation. In: MICCAI (2015)
13. Roth, H.R., Lu, L., Farag, A., Sohn, A., Summers, R.M.: Spatial aggregation of holistically-nested networks for automated pancreas segmentation. MICCAI (2016)
14. Wang, G., Li, W., Ourselin, S., Vercauteren, T.: Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In: MICCAI Brainlesion Workshop (2017)
15. Xia, Y., Xie, L., Liu, F., Zhu, Z., Fishman, E.K., Yuille, A.L.: Bridging the gap between 2d and 3d organ segmentation. In: MICCAI (2018)
16. Zhou, Y., Xie, L., Shen, W., Wang, Y., Fishman, E.K., Yuille, A.L.: A fixed-point model for pancreas segmentation in abdominal CT scans. In: MICCAI (2017)
17. Zhu, Z., Xia, Y., Shen, W., Fishman, E., Yuille, A.L.: A 3d coarse-to-fine framework for volumetric medical image segmentation. In: 3DV (2018)

Supplementary Material

Global Image Caption The visualization illustration of predicted segmentation for "VNAS", "Mix", "UNet" and "VNet" on the MSD Pancreas Tumors dataset, which is the most challenging task among our 3 segmentation tasks. The masked blue and red regions denote for the normal pancreas regions and tumor regions, respectively. Best viewed in color.

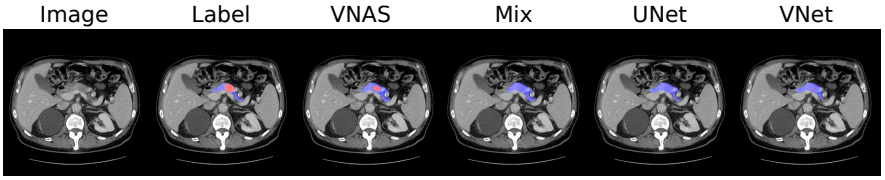

Fig. 1. The segmentation visualization for the case number 309. "VNAS" successfully detects the tiny tumor regions.

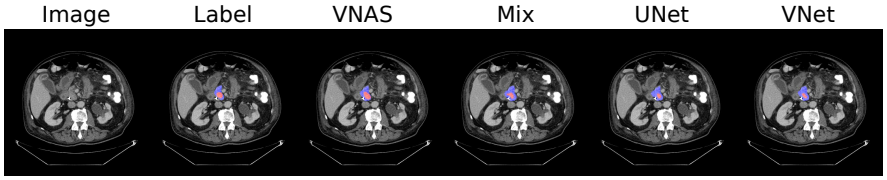

Fig. 2. The segmentation visualization for the case number 329. "VNAS" detects the tiny tumor regions better than others.

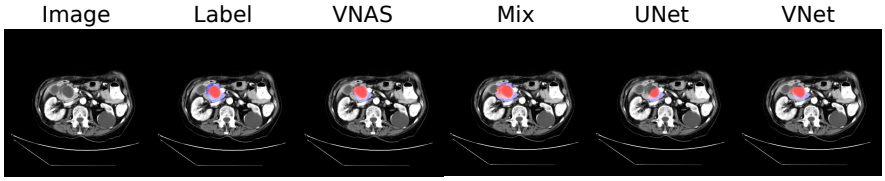

Fig. 3. The segmentation visualization for the case number 069. "VNAS" detects the medium-size tumor regions better than others.