



AutoPath: Image-Specific Inference for 3D Segmentation

Dong Sun, Yi Wang*, Dong Ni and Tianfu Wang*

National-Regional Key Technology Engineering Laboratory for Medical Ultrasound, Guangdong Provincial Key Laboratory of Biomedical Measurements and Ultrasound Imaging, School of Biomedical Engineering, Health Science Center, Shenzhen University, Shenzhen, China

In recent years, deep convolutional neural networks (CNNs) has made great achievements in the field of medical image segmentation, among which residual structure plays a significant role in the rapid development of CNN-based segmentation. However, the 3D residual networks inevitably bring a huge computational burden to machines for network inference, thus limiting their usages for many real clinical applications. To tackle this issue, we propose *AutoPath*, an image-specific inference approach for more efficient 3D segmentations. The proposed *AutoPath* dynamically selects enabled residual blocks regarding different input images during inference, thus effectively reducing total computation without degrading segmentation performance. To achieve this, a policy network is trained using reinforcement learning, by employing the rewards of using a minimal set of residual blocks and meanwhile maintaining accurate segmentation. Experimental results on liver CT dataset show that our approach not only provides efficient inference procedure but also attains satisfactory segmentation performance.

Keywords: segmentation, 3D residual networks, reinforcement learning, policy network, image-specific inference

OPEN ACCESS

Edited by:

Long Wang,
University of Science and Technology
Beijing, China

Reviewed by:

Ziyue Xu,
Nvidia, United States
Wellington Pinheiro dos Santos,
Federal University of Pernambuco,
Brazil

*Correspondence:

Yi Wang
onewang@szu.edu.cn
Tianfu Wang
tfwang@szu.edu.cn

Received: 02 June 2020

Accepted: 19 June 2020

Published: 24 July 2020

Citation:

Sun D, Wang Y, Ni D and Wang T
(2020) AutoPath: Image-Specific
Inference for 3D Segmentation.
Front. Neurobot. 14:49.
doi: 10.3389/fnbot.2020.00049

1. INTRODUCTION

Automated segmentation is useful to assist doctors in disease diagnosis and surgical/treatment planning. Since deep learning (LeCun et al., 2015) has utilized widely, medical image segmentation has made great progresses. Various architectures of deep convolutional neural networks (CNNs) have been proposed and successfully introduced to many segmentation applications. Among various architectures, the residual structures in ResNet (He et al., 2016) play an important role in the rapid development of CNN-based segmentation. The backbone which contains the residual blocks has become essential support for many segmentation models, such as DeepLab V3 (Chen et al., 2017), HD-Net (Jia et al., 2019), Res-UNet (Xiao et al., 2018), and so on. Despite the superior performance of residual blocks, these structures inevitably bring a huge computational burden for network inference. This leads to the difficulty of introducing deep models (such as 3D ResNet-50/101) in clinical practice.

Recently, some strategies for model compression have been devoted to tackling the problem of large computation, among which the network pruning approaches have been extensively investigated (Li et al., 2016; He et al., 2017, 2018; Liu et al., 2017; Luo et al., 2017). In addition, other researches are focusing on lightweight network architectures (Howard et al., 2017; Ma et al., 2018; Mehta et al., 2018). However, all aforementioned methods including pruning, knowledge distillation, and lightweight structures all require network retraining and hyper-parameters retuning, which may consume plenty of extra time.

This paper explores the problem of dynamically distributing computation across all residual blocks in a trained ResNet for image-specific segmentation inference (see **Figure 1**). Relevant studies have been investigated in classification tasks. Teerapittayanon et al. (2016) developed BranchyNet to conduct fast inference via early exiting from deep neural networks. Graves (2016) devised an adaptive computation time (ACT) approach for recurrent neural network (RNN), by designing a halting unit whose activation indicates the termination probability of computations. Huang et al. (2016) proposed stochastic depth for deep networks, a training strategy that enables the seemingly contradictory setup to train short networks and use deep networks at test time. Veit et al. (2016) proposed a description of residual networks in classification showing that residual blocks can be seen as a collection of many paths and they do not strongly depend on each other thus can be selectively dropped. However, there are two obvious differences between segmentation and classification tasks: (1) The neural networks for classification often take a short approach, such as identifying a car by its shadow. Segmentation is classifying each pixel, thus the neural networks cannot be lazy; (2) Classification can work with local features, but segmentation needs to take global information (such as shape priors) into consideration.

In this study, we propose *AutoPath*, an image-specific strategy to design the inference path that uses minimal residual blocks but still preserves satisfactory segmentation accuracy. Specifically, a policy network is trained using reinforcement learning, by using the rewards of involving a minimal set of residual blocks and meanwhile maintaining accurate segmentation. To the best of our knowledge, this is the first investigation of the dynamic inference path for 3D segmentation using residual networks. Experimental results demonstrate that our strategy not only

provides efficient inference procedure but also attains satisfactory segmentation accuracy.

2. METHODS

Figure 2 illustrates the proposed framework. Given a new 3D volume, the policy network outputs “keep” or “drop” decisions for each residual block in the pretrained 3D DeepLab V3 network. Such image-specific inference path is then used for the segmentation prediction. The policy network is trained using reinforcement learning by rewarding accurate segmentation with minimal involved blocks.

2.1. 3D Residual Backbone

We implement a 3D DeepLab V3 based on the original DeepLab V3 (Chen et al., 2017), and further set its backbone to 3D ResNet according to ResNet (He et al., 2016) (the classification head is removed). We then specifically modify the 3D residual blocks to achieve better baseline results. Optimized 3D residual blocks assign the stride of each convolutional layer along z axis to 1 to constrain downsampling along the slice direction. In addition, we replace the straightforward upsampling operations after atrous spatial pyramid pooling (ASPP) with a decoder consisting of a series of convolutions according to Chen et al. (2019).

2.2. *AutoPath* Strategy

Backbone, as an indispensable part of segmentation networks, occupies most of the memory and calculation. Generally, the popular residual backbone consists of multiple repeated residual blocks. We regard each block as an independent decision unit by assuming that different blocks do not share strong dependencies in segmentation. Then as shown in **Figure 2**, we introduce

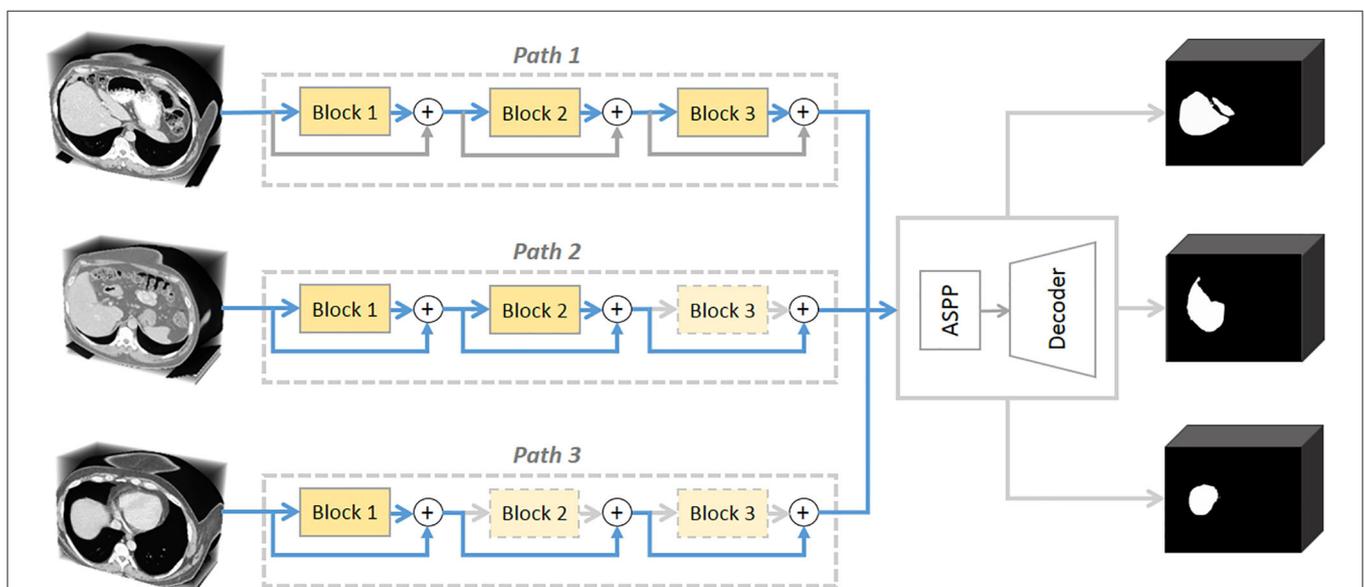
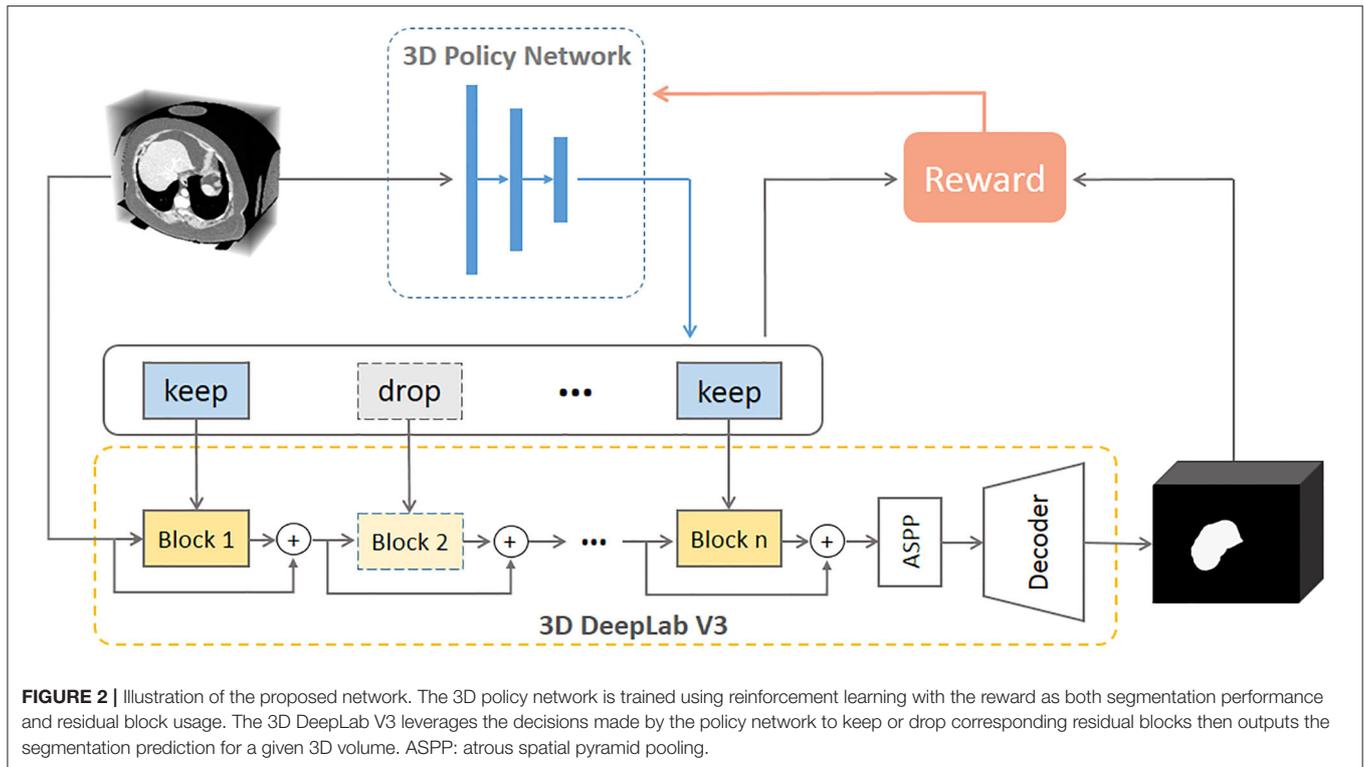


FIGURE 1 | A conceptual illustration of the proposed *AutoPath*. The motivation is to dynamically distributing computation across a ResNet. *AutoPath* selectively drops unnecessary residual blocks for image-specific inference. Such inference simultaneously achieves efficiency and accuracy.



reinforcement learning to train a policy network to intelligently conduct block dropping and explore the inference path that generates accurate segmentation with fewer blocks.

2.2.1. 3D Policy Network

Considering the actions for residual blocks can only be “keep” or “drop,” we define the policy for block dropping as a Bernoulli distribution (Wu et al., 2018):

$$\pi_{\mathbf{W}}(\mathbf{a}|\mathbf{x}) = \prod_{n=1}^N \mathbf{p}_n^{a_n} (1 - \mathbf{p}_n)^{1-a_n}, \quad (1)$$

where \mathbf{x} denotes the input 3D image and N is the total number of residual blocks in the pretrained 3D DeepLab V3 network. \mathbf{W} denotes the weights of the policy network. \mathbf{p} represents the “drop” likelihood ($\mathbf{p}_n \in [0, 1]$), and is the policy network’s output after sigmoid activation. The action vector \mathbf{a} is determined by \mathbf{p} , where $a_n = 0$ refers to drop the n -th block, otherwise keeping the corresponding block.

2.2.2. Reward Function

3D Segmentation is generally considered to be voxel-level classification, thus every voxel has to receive evaluation feedback for the corresponding action. Thus, we design the following voxel-level reward function:

$$R(\mathbf{a}) = \begin{cases} 1 - (\frac{|\mathbf{a}|_0}{N})^2 & \text{if } VDC(i) = \frac{1}{2} \\ -\tau & \text{if } VDC(i) = 1, \end{cases} \quad (2)$$

where $(\frac{|\mathbf{a}|_0}{N})^2$ calculates the block usage. When the prediction of a voxel i is the same as the label, we encourage dropping more blocks by giving a larger reward to the policy. On the other hand, we penalize using τ , which balances block usage and segmentation accuracy. When τ is a large value, the policy is prone to have a more solid segmentation result; otherwise, it is more likely to drop blocks. We designed Voxel Dice Coefficient (VDC) to identify which pixel should be penalized:

$$VDC(i) = \frac{2 * (y'_i \cap y_i) + 1}{y'_i + y_i + 1}, \quad (3)$$

where $y_i \in \{0, 1\}$ and $y'_i \in \{0, 1\}$ denotes the ground truth and prediction for voxel i , respectively.

2.2.3. Learning Strategy

Finally, we maximize the expected reward $\mathbb{E}_{\mathbf{a} \sim \pi_{\mathbf{W}}}[R(\mathbf{a})]$ to train the policy network. We employ policy gradient to calculate the gradient of the expected reward:

$$\nabla_{\mathbf{W}} \mathbb{E}_{\mathbf{a} \sim \pi_{\mathbf{W}}}[R(\mathbf{a})] = \mathbb{E}[R(\mathbf{a}) \nabla_{\mathbf{W}} \log \pi_{\mathbf{W}}(\mathbf{a}|\mathbf{x})] \quad (4)$$

$$= \mathbb{E}[R(\mathbf{a}) \nabla_{\mathbf{W}} \log \prod_{n=1}^N \mathbf{p}_n^{a_n} (1 - \mathbf{p}_n)^{1-a_n}] \quad (5)$$

$$= \mathbb{E}[R(\mathbf{a}) \nabla_{\mathbf{W}} \sum_{n=1}^N \log[\mathbf{p}_n^{a_n} + (1 - \mathbf{p}_n)(1 - a_n)]] \quad (6)$$

The Equation (6) can be approximated by Monte-Carlo sampling strategy.

To achieve efficient training, we further employ curriculum learning (Bengio et al., 2009) to train the policy network. Specifically, for epoch c ($c < N$), the first $N - c$ blocks are kept, while the learning is conducted on the last c blocks. As c increases, more and more blocks join the optimization until all blocks are involved. Algorithm 1 shows the training procedure of the proposed network.

3. EXPERIMENTS AND RESULTS

3.1. Materials and Implementation Details

3.1.1. Materials

Experiments were carried on liver CT images from LiTS challenge (Bilic et al., 2019). LiTS dataset contains 131 contrast-enhanced CT images acquired from six clinical centers around the world. 3DIRCADb is a subset from the LiTS dataset with 22 cases. Our network was trained using 109 cases from LiTS without data from 3DIRCADb,

and then evaluated on the 3DIRCADb subset using Dice metric.

3.1.2. Implementation Details

The experiments were conducted using 3D DeepLab V3 network with 18-layers and 50-layers, respectively. We adopted Adam (Kingma and Ba, 2014) with learning rate of 0.01 and batch size of 4 and 11 for ResNet-18 and ResNet-50, respectively. In addition, we utilized learning rate scheduler that decreases by 0.1 for every 100 epochs. The maximum epoch was set to 400.

For the policy network, we set learning rate to 0.001, τ to 50 and used the batch size of 1 and 5 for ResNet-18 and ResNet-50, respectively.

Algorithm 1 | Training of the policy network

Input : An input 3D image \mathbf{x} and corresponding label \mathbf{y}

- 1 Initialize policy network \mathbf{W} randomly
- 2 Set epochs for curriculum training to C
- 3 **for** $c \leftarrow 1$ to C **do**
- 4 $\mathbf{p} \leftarrow$ output of policy network
- 5 **if** $c < N$ **then**
- 6 | set $\mathbf{p}[1:N-c] = 1$ \triangleright curriculum training
- 7 **end**
- 8 $\mathbf{a} \sim \text{Bernoulli}(\mathbf{p})$
- 9 Run the 3D DeepLab V3 based on \mathbf{a}
- 10 Calculate $R(\mathbf{a})$ using Eq (2)
- 11 Gradients back-propagation using Eq (6)
- 12 **end**

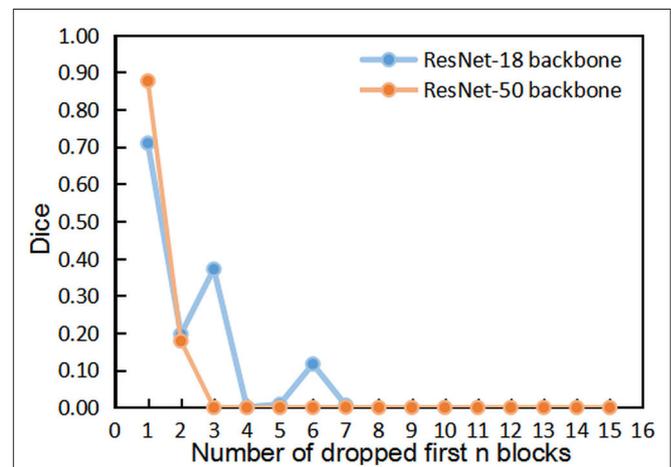


FIGURE 4 | The comparisons of segmentation performance from dropping the first n blocks of residual backbones.

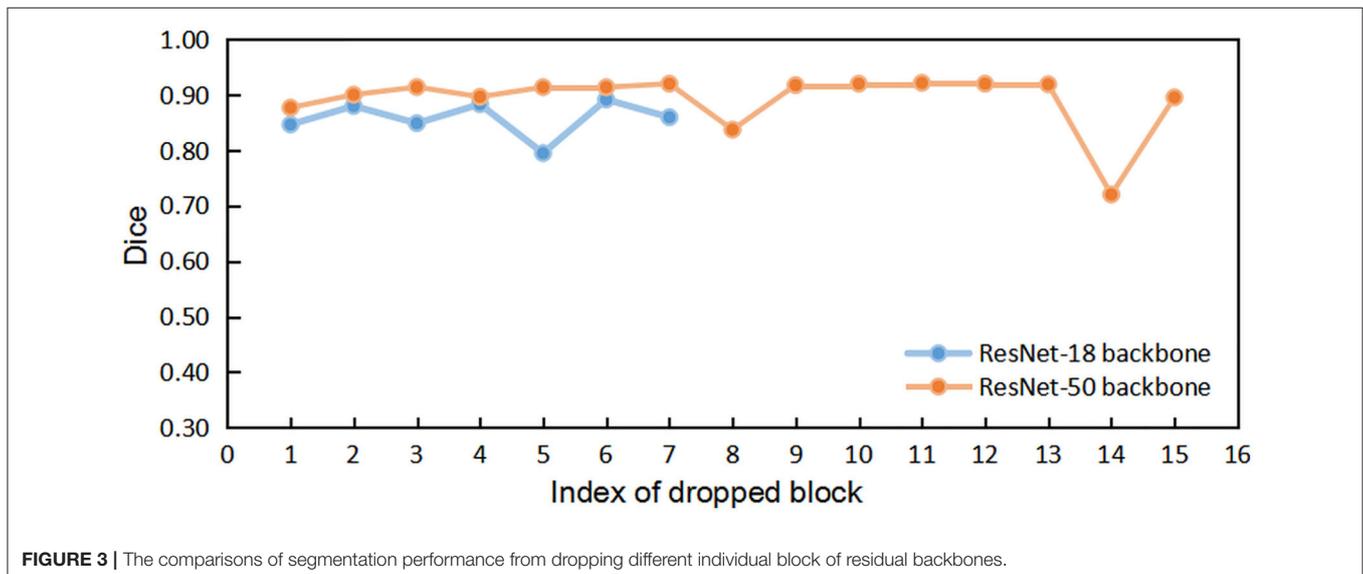
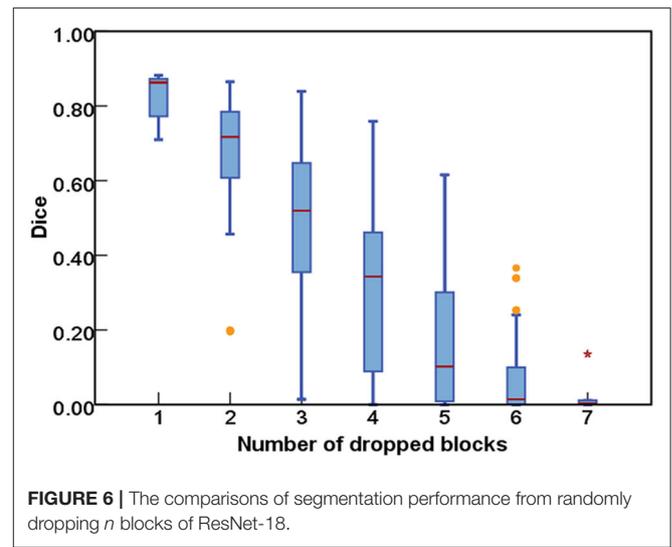
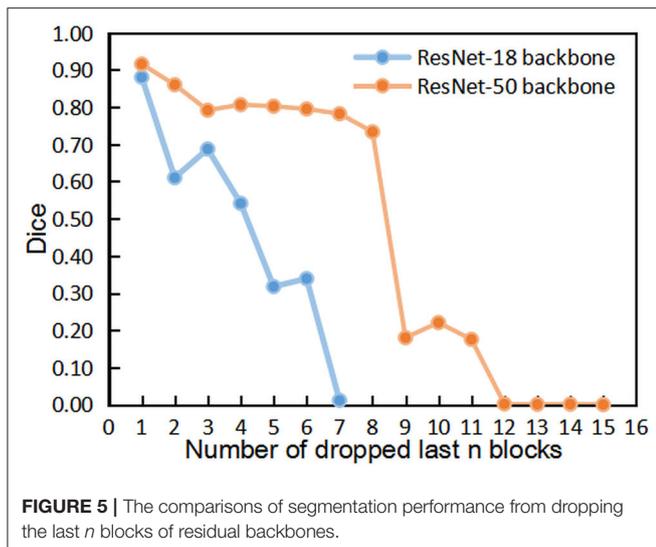


FIGURE 3 | The comparisons of segmentation performance from dropping different individual block of residual backbones.



3.2. Performance Evaluation

3.2.1. Investigation of Blocks' Dependencies

We first implemented the DropN strategy, which means dropping a single n -th block in residual backbone, to observe the dependencies of different blocks. We executed this strategy to ResNet-18 and ResNet-50 backbone which has 8 and 16 residual blocks, respectively. **Figure 3** shows that dropping individual block from residual backbone has a minimal impact on Dice evaluation except for few blocks. This suggests that different blocks in ResNet backbone do not share strong dependencies and most blocks are not indispensable for the accurate segmentation. Thus, dropping blocks in inference is feasible for segmentation.

3.2.2. Heuristic Dropping Strategies

We then evaluated three manual dropping strategies as follows:

- 1) DropFirstN, which means to drop all blocks before the n -th block;
- 2) DropLastN, which means to drop all blocks after the n -th block;
- 3) DropRandomN, which means to drop n blocks randomly.

Note that DropRandomN is a random strategy, thus for each n we performed 100 and 500 repeated experiments for ResNet-18 and ResNet-50, respectively.

Figures 4, 5 show the results of DropFirstN and DropLastN, respectively. It can be observed from **Figure 4** that first several blocks were relatively important for ResNet backbone. The Dice value dropped to almost 0 when the first three blocks were dropped. As shown in **Figure 5**, for ResNet-50, dropping the last 8 blocks didn't affect segmentation performance sharply.

As for the DropRandomN, with the increase of dropped blocks in the shallow backbone, the segmentation performance gradually decreased, as seen in **Figure 6**. In contrast, for the ResNet-50, the average Dice was almost 0 when 8 blocks were randomly dropped, as shown in **Figure 7**.

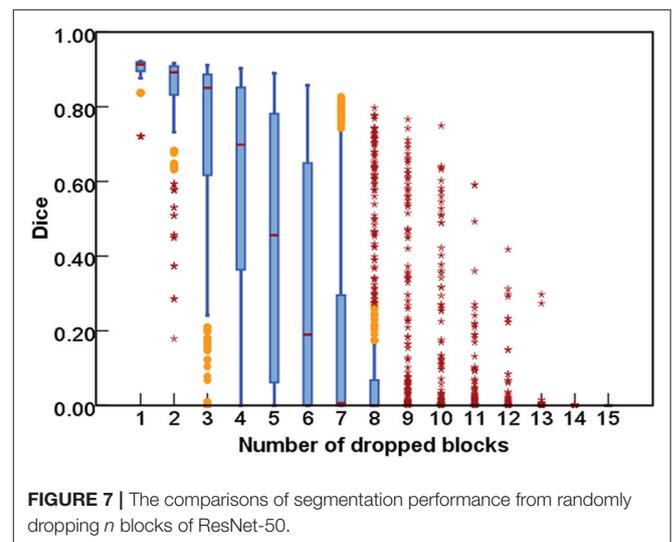


TABLE 1 | The comparisons of segmentation performance from heuristic strategies and *AutoPath* at the same dropping level.

	ResNet-18 backbone		ResNet-50 backbone	
	Dice	Dropped blocks	Dice	Dropped blocks
DropFirstN	19.67%	2	0.00%	4
DropLastN	61.03%	2	80.66%	4
DropRandomN	67.80%	2	59.11%	4
Ours	87.11%	2.4	89.14%	4.5
Full backbone	89.54%	8 used	93.16%	16 used

3.2.3. *AutoPath*

We compared the segmentation performance of heuristic strategies (i.e., DropFirstN, DropLastN, DropRandomN) and the proposed *AutoPath* at the same dropping level. **Table 1** reports that by considering image-specific input, our *AutoPath* can achieve an average block dropping ratio of more than

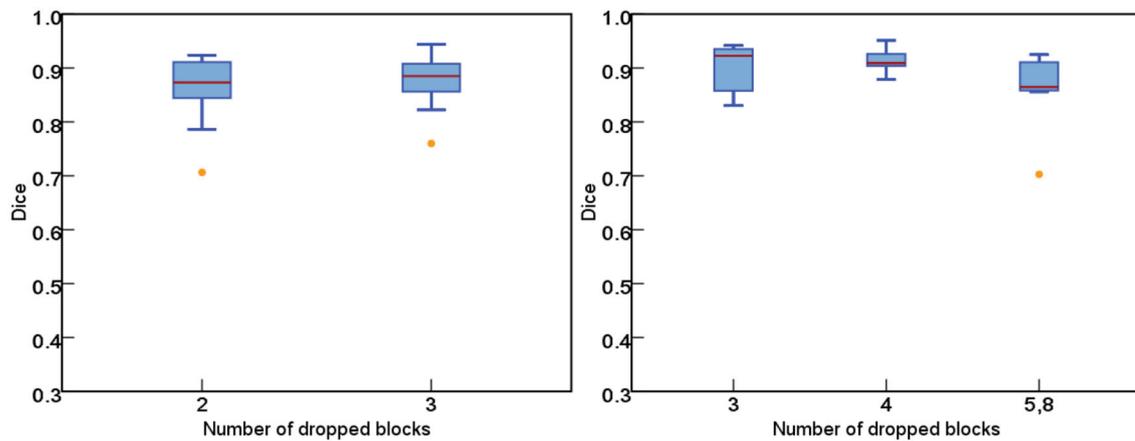


FIGURE 8 | The statistics of the *AutoPath* for all testing data. For most cases, *AutoPath* can provide high quality segmentation with fewer block usage. Left: ResNet-18; Right: ResNet-50.

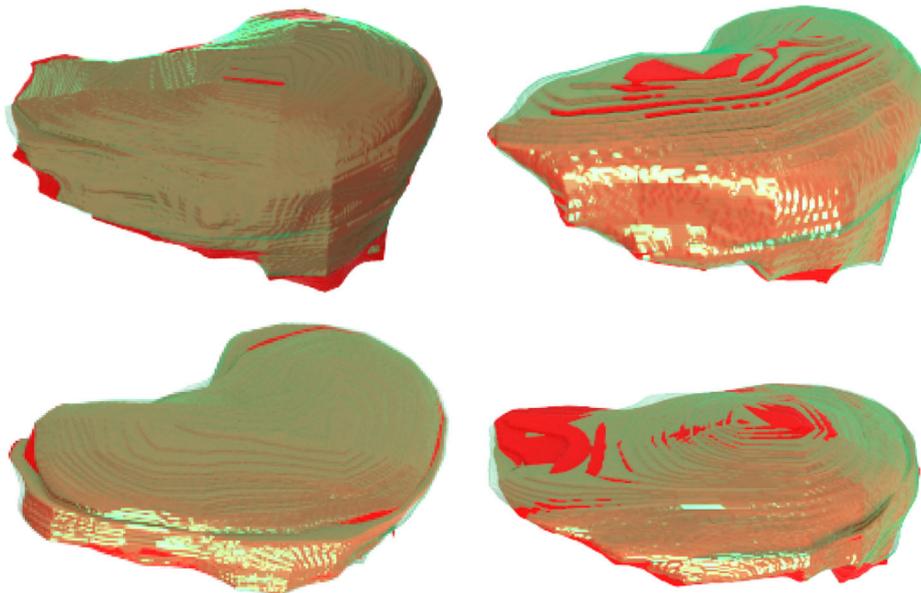


FIGURE 9 | 3D visualizations of some segmentation results obtained using *AutoPath* (green) and full backbone (red), respectively. It can be observed that the segmentation performance from the proposed *AutoPath* was comparable to that of the full backbone architecture.

25%, meanwhile with only 2 and 4% decrease of Dice values for ResNet-18 and ResNet-50, respectively. Also can be seen from **Table 1** that by dropping the same number of residual blocks, our *AutoPath* outperformed other heuristic strategies by a large margin.

Figure 8 further plots the statistics of the *AutoPath* for all testing data. For the image-specific inference, the *AutoPath* selectively dropped 2 or 3 blocks for ResNet-18, and 3, 4, 5, or 8 blocks for ResNet-50. For most cases, *AutoPath* can maintain a high quality segmentation with fewer block usage, which demonstrates its promising application for real clinical circumstance. **Figures 9, 10** visualizes some 3D and 2D segmentation results obtained using *AutoPath* and full

backbone, respectively. It can be observed that the segmentation performance from the proposed *AutoPath* was comparable to that of the full backbone architecture.

4. DISCUSSION AND CONCLUSION

This paper develops a reinforcement learning method to select image-specific and efficient inference paths for 3D segmentation, which addresses the problem of huge computational burden for 3D segmentation networks. To our best knowledge, this is the first study of the dynamic inference path for 3D segmentation. We refer to it as *AutoPath*, which can leverage an image-specific path including fewer residual blocks to attain accurate prediction.

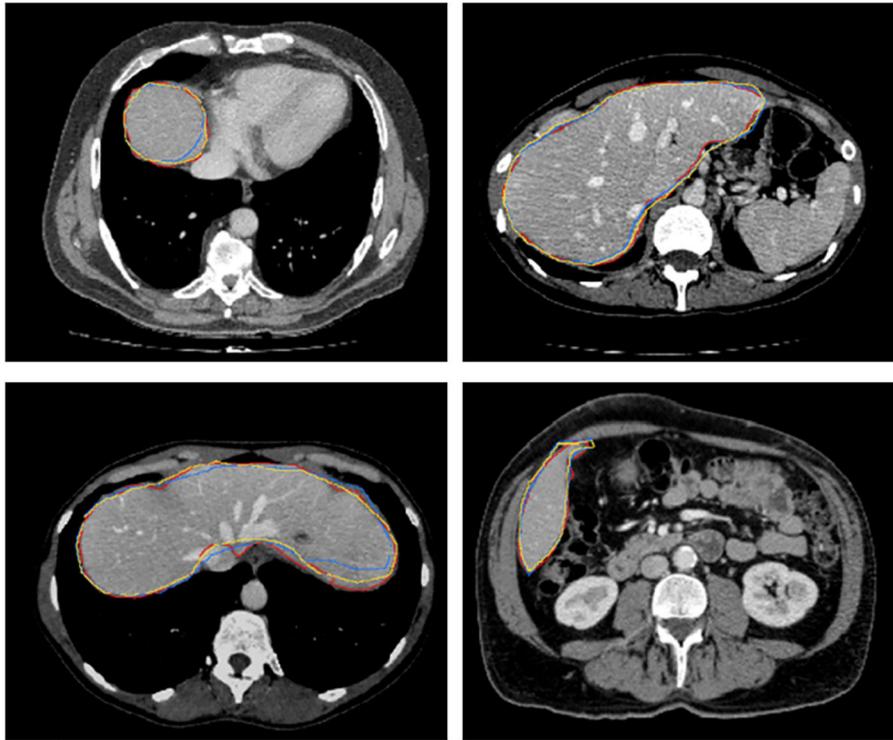


FIGURE 10 | 2D slices of some segmentation results obtained using *AutoPath* (yellow), full backbone (blue), and the ground truth (red), respectively.

To achieve this, we train a network to determine the policy of block dropping and the pretrained segmentation network executes inference according to this policy. We conducted extensive experiments on the liver CT dataset using 3D DeepLab V3 network with 18-layers and 50-layers, respectively. Experimental results demonstrate that *AutoPath* is a reliable method for the dynamic inference in 3D segmentation.

Deep neural networks offer excellent segmentation performance, yet their computational expense restrict their clinical usage, especially for the 3D segmentation. To tackle this issue, various compressed models have been proposed (Li et al., 2016; He et al., 2017, 2018; Liu et al., 2017; Luo et al., 2017). While the network efficiency has been improved somehow, the solution is a one-size-fit-all network that omits different inputs' complexity. In contrast, we investigate adaptively allocating computation across a CNN model according to specific input. Furthermore, our image-specific inference is conducted on the trained network, thus do not have to spend extra time for the network retuning.

In this study, although the DeepLabV3 network was employed as backbone to equip with residual structures, it could be replaced using other backbone architectures. With regard to our image-specific inference method, the residual structures are the most crucial components but not the design of backbones. In medical segmentation tasks, various CNN-based approaches have employed residual structures. For example, the encoder of HD-Net (Jia et al., 2019) is based on 3D ResNet-101 and BOWDA-Net (Zhu et al., 2019) utilizes dense connection

multiple times. In addition, Xiao et al. (2018) propose a weighted Res-UNet which replaces the convolution block with residual block to achieve remarkable results in retina vessel segmentation. Furthermore, there are some improved structures based on ResNet, such as ResNext (Xie et al., 2017), SE-Net (Hu et al., 2018), and SK-Net (Li et al., 2019). Our proposed method can be utilized to dynamically distribute computation across their residual blocks for image-specific segmentation inference.

Our current scheme mainly focuses on the usage reduction of the residual blocks due to their independent design. It may not be directly adopted to other network configurations without residual structures. Future work may further investigate the dynamic inference for other network configurations. In addition, although our method attained satisfactory performance on liver CT volumes, further validations on large amount of various medical images will be conducted to investigate the robustness and generalization ability of the proposed scheme.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://chaos.grand-challenge.org/Combined_Healthy_Abdominal_Organ_Segmentation.

AUTHOR CONTRIBUTIONS

DS, YW, DN, and TW response for study design. DS implemented the research. DS and YW conceived the

experiments. DS conducted the experiments. DS, YW, and TW analyzed the results. DS and YW wrote the main manuscript text and prepared the figures. All authors reviewed the manuscript.

FUNDING

This work was supported in part by the National Natural Science Foundation of China under Grant 61701312, in part by

the Guangdong Basic and Applied Basic Research Foundation (2019A1515010847), in part by the Medical Science and Technology Foundation of Guangdong Province (B2019046), in part by the Shenzhen Key Basic Research Project (No. JCYJ20170818094109846), in part by the Natural Science Foundation of SZU (Nos. 2018010 and 860-000002110129), in part by the SZU Medical Young Scientists Program (No. 71201-000001), and in part by the Shenzhen Peacock Plan (KQTD2016053112051497).

REFERENCES

- Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09* (New York, NY: Association for Computing Machinery), 41–48.
- Bilic, P., Christ, P. F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., et al. (2019). The liver tumor segmentation benchmark (lits). *arXiv preprint arXiv:1901.04056*.
- Chen, L.-C., Papandreu, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.
- Chen, S., Ma, K., and Zheng, Y. (2019). Med3d: transfer learning for 3d medical image analysis. *arXiv preprint arXiv:1904.00625*.
- Graves, A. (2016). Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 770–778.
- He, Y., Kang, G., Dong, X., Fu, Y., and Yang, Y. (2018). Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*.
- He, Y., Zhang, X., and Sun, J. (2017). "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 1389–1397.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 7132–7141.
- Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. (2016). "Deep networks with stochastic depth," in *European Conference on Computer Vision* (Amsterdam: Springer), 646–661.
- Jia, H., Song, Y., Huang, H., Cai, W., and Xia, Y. (2019). "Hd-net: hybrid discriminative network for prostate segmentation in mr images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Shenzhen: Springer), 110–118.
- Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *corr abs/1412.6980*.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539
- Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. (2016). Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.
- Li, X., Wang, W., Hu, X., and Yang, J. (2019). "Selective kernel networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 510–519.
- Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. (2017). "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 2736–2744.
- Luo, J.-H., Wu, J., and Lin, W. (2017). Thinet: a filter level pruning method for deep neural network compression. in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 5058–5066.
- Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. (2018). "Shufflenet v2: practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich), 116–131.
- Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., and Hajishirzi, H. (2018). "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich), 552–568.
- Teerapittayanon, S., McDanel, B., and Kung, H.-T. (2016). "Branchynet: fast inference via early exiting from deep neural networks," in *2016 23rd International Conference on Pattern Recognition (ICPR)* (Amsterdam: IEEE), 2464–2469.
- Veit, A., Wilber, M. J., and Belongie, S. (2016). "Residual networks behave like ensembles of relatively shallow networks," in *Advances in Neural Information Processing Systems* (Barcelona), 550–558.
- Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Davis, L. S., Grauman, K., et al. (2018). "Blockdrop: dynamic inference paths in residual networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 8817–8826.
- Xiao, X., Lian, S., Luo, Z., and Li, S. (2018). "Weighted res-unet for high-quality retina vessel segmentation," in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)* (Hangzhou: IEEE), 327–331.
- Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 1492–1500.
- Zhu, Q., Du, B., and Yan, P. (2019). Boundary-weighted domain adaptive neural network for prostate mr image segmentation. *IEEE Trans. Med. Imaging* 39, 753–763. doi: 10.1109/TMI.2019.2935018

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Sun, Wang, Ni and Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.