GRNT: GATE-REGULARIZED NETWORK TRAINING FOR IMPROVING MULTI-SCALE FUSION IN MEDICAL IMAGE SEGMENTATION

Yizhe Zhang[1*] *Pengfei Gu*[2*] *Yejia Zhang*[2] *Chaoli Wang*[2] *Danny Z. Chen*[2]

[1]Nanjing University of Science and Technology, Nanjing, Jiangsu, China
[2]University of Notre Dame, Department of Computer Science and Engineering, Notre Dame, IN, USA

ABSTRACT

Multi-scale fusion is a key for semantic segmentation of medical images. Recent deep learning methods added network complexity to achieve better multi-scale fusion results. However, given the already very expressive architectures (e.g., U-Net), an interesting question is whether additional complexity is necessary for achieving more robust multi-scale fusion performance. In this paper, we proposed a new method for improving the multi-scale fusion performance of a medical image segmentation model. We create a set of binary gates at fusing locations that allow us to control forward and backward flows in training. A gate on-off schedule is imposed so that high-scale (level) features could drive the generation of segmentation, while low-scale features serve as complimentary for reconstructing shape details. Our method is effective, easy to implement, and occurs no extra cost when deploying the trained model. Experiments show that our gate-regularized network training (GrNT) improves widely adopted models (e.g., U-Net, Attention U-Net, and DenseVoxNet) on three segmentation datasets (2017 ISIC Skin Lesion segmentation (2D), MM-WHS CT (3D), and 2016 HVSMR (3D)).

1. INTRODUCTION

Fusing multi-scale information plays a key role in attaining robust semantic segmentation of medical images. Low-level features are essential for reconstructing the fine details of the segmented shapes, but lack sufficient semantic information to determine object classes due to the limited field of view and encoding parameters. On the other hand, high-level features are rich in semantic information, but lack sufficient resolution/details to reconstruct shape details. Combining high-level and low-level features effectively is a key to medical image segmentation. For example, U-Net [1] fuses multi-scale information from bottom to top incrementally with concatenation and convolution operations, DCN [2] combines multi-scale information via a one-time concatenation following convolution layers for information fusion, and DenseVoxNet [3] fuses multi-scale information with dense skip connections between convolution layers/blocks.

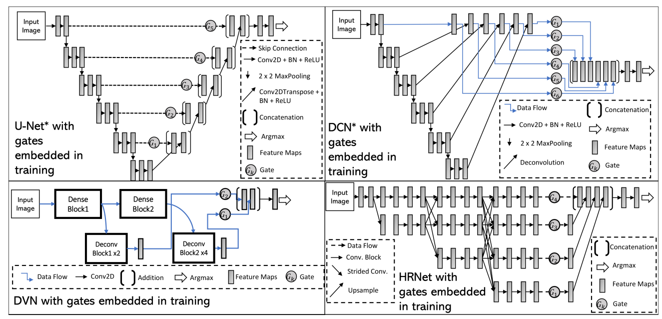

Fig. 1. U-Net [1], DCN [2], DVN [3], and HRNet [4] with gates embedded during the training process. * indicates a deeper version of the original network.

Recent developments designed additional modules for improving multi-scale fusion. For example, Attention U-Net [5] utilizes features from deeper layers and forms attention maps to apply on shallower layer features, aiming to guide low-level features in generating segmentation. In a similar principle, Inf-Net [6] designed reverse attention modules to allow higher-level features to guide lower-level features in generating segmentation of COVID-19 lung CT scans.

On the other hand, studies for better network training aimed to improve segmentation performance without adding additional complexity to the original networks. The No New-Net work [7] demonstrated that a well-trained U-Net is difficult to beat on the BRATS18 dataset. A few methods [8, 9, 10] opted to train better medical segmentation networks using data augmentation techniques. Recently, methods for learning adaptive embedding have been proposed to consider incremental classes and model capacities [11, 12].

In this paper, we propose a new gate-regularized network training (GrNT) method to improve multi-scale information fusion for medical image segmentation networks. Working in a training perspective, GrNT does not add additional complexity to the networks in test time. First, binary gates are created and placed at locations where multi-scale feature fusion occurs in a semantic segmentation network. Then, a

* Equal Contribution.

scheduled training process is designed to control the on/off (1/0) of these gates so that the trained network produces segmentation results in a way that higher-level features always drive the generation of segmentation and lower-level features serve only as complimentary for reconstructing fine shape details. In test time, these added gates are disabled (removed) to allow utilizing the full expressiveness of the network; thus, no additional inference time/computation cost is incurred. Experiments on three public segmentation datasets show that our new GrNT method brings considerable accuracy improvement with state-of-the-art networks. Further, our GrNT method enables a more informative error analysis that can measure how much a particular set of features may contribute to the segmentation errors.

Algorithm 1 Gate-regularized network training

function GRNT($x_i, y_i, i = 1, 2, \ldots, n, maxiter, batchsize$)
Suppose three gates are used in training: g_l is put on low-level features, g_m is put on mid-level features, and g_h is put on high-level features.
    Initialize segmentation network f^{g_h,g_m,g_l};
    for $iter \leftarrow 1$ to $maxiter$ **do**
        $g_l \leftarrow 0, g_m \leftarrow 0, g_h \leftarrow 0$;
        if iter $\bmod 3 == 1$ **then** $g_l \leftarrow 0, g_m \leftarrow 0, g_h \leftarrow 1$;
        else if iter $\bmod 3 == 2$ **then** $g_l \leftarrow 0, g_m \leftarrow 1, g_h \leftarrow 1$;
        else $g_l \leftarrow 1, g_m \leftarrow 1, g_h \leftarrow 1$;
        Randomly select $batchsize$ samples (X^{iter}, Y^{iter}) from the full training set;
        Update network f^{g_h,g_m,g_l} using the error from $\mathcal{L}(f^{g_h,g_m,g_l}(X^{iter}), Y^{iter})$;
    return segmentation network f^{g_h,g_m,g_l}.

2. GATE-REGULARIZED NETWORK TRAINING (GRNT)

2.1. Gate Design, Placement, and Training

Suppose a fully convolutional network (FCN) can be expressed as a series of encoding functions followed by a series of decoding functions. Given an input image x, on the encoding side, low level features are extracted via $\tau_{low} = \phi_{low}(x)$; middle-level features are then computed on top of the extracted low-level features, as $\tau_{mid} = \phi_{mid}(\tau_{low})$; finally, high-level features are extracted via $\tau_{high} = \phi_{high}(\tau_{mid})$. Low-level features are rich in image fine details but are not sufficient in providing semantic predictions, due to the limited field of view and encoding parameters. On the other hand, high-level features contain more reliable semantic information but lack image fine details due to the pooling operations (e.g., max-pooling, 2-stride Convolution (Conv)) applied along the encoding path.

With the low, mid, and high level features thus obtained, the decoding side is normally carried out to generate segmentation results. In many recent FCN designs, the decoder consists of not only up-sampling but also convolutional layers (e.g., U-Net). In a simple form, we express the decoding operations as $df_{high} = D_{high}(\tau_{high})$, $df_{mid} = D_{mid}(\tau_{mid}, df_{high})$, and $df_{low} = D_{low}(\tau_{low}, df_{mid})$. Finally, several convolution layers are used to transfer df_{low} into the final segmentation map as $seg = (\text{argmax}(\phi_{final}(df_{low})))$.

Given the above formulation, our GrNT method puts a binary gate g_h on top of ϕ_{high}, a binary gate g_m on top of ϕ_{mid}, and a binary gate g_l on top of ϕ_{low}. With these added gates, the decoding operations are expressed as $df_{high} = D_{high}(g_h \times \tau_{high})$, $df_{mid} = D_{mid}(g_m \times \tau_{mid}, df_{high})$, and $df_{low} = D_{low}(g_l \times \tau_{low}, df_{mid})$.

During network training, we explicitly control the on/off (1/0) of these gates to encourage the high-level information driving the generation of the segmentation results, and the low-level information serving only to supply segmentation shape details. Given a set of images $x_1, x_2, \ldots, x_n$ and their corresponding label maps $y_1, y_2, \ldots, y_n$, using an FCN $f^{g_l,g_m,g_h}(x)$ with gates embedded at the multi-scale fusion locations, our GrNT loss function can be written as

$$\sum_{i=1}^{n} \mathcal{L}(f^{g_h=1,g_m=0,g_l=0}(x_i), y_i) + \mathcal{L}(f^{g_h=1,g_m=1,g_l=0}(x_i), y_i) + \mathcal{L}(f^{g_h=1,g_m=1,g_l=1}(x_i), y_i) \tag{1}$$

where $\mathcal{L}$ is a typical cross-entropy loss for segmentation tasks. Using a mini-batch stochastic gradient descend method, we optimize this loss function using Algorithm 1. After network training, for normal inference, one can either remove the gates or set all the gates as 1.

2.2. Architecture Case Study

In this section, we showcase how to apply our proposed GrNT method to several widely used FCNs.

U-Net-type networks. We derive the above general formula to fit the U-Net [1] architecture. The exact placements of the gates are shown in Fig. 1 (top-left). During training, the gates are scheduled as follows. Iter-1: g_1 = 1, g_2, g_3, g_4, g_5 = 0; iter-2: g_1, g_2 = 1, $g3$, $g4$, $g5$ = 0; iter-3: g_1, g_2, g_3 = 1, g_4 ,g_5 = 0; iter-4: g_1, g_2, g_3, g_4 = 1, g_5 = 0; iter-5: g_1, g_2, g_3, g_4, g_5 = 1; iter-6: g_1 = 1, g_2, g_3, g_4, g_5 = 0; these patterns continue iteratively. These gate placements and the training schedule apply to Attention U-Net [5] since the concatenations between multi-scale features follow a similar design as the original U-Net.

DCN. Similar to U-Net, DCN [2] utilizes concatenation to combine features across scales. Different from the incremental fusion approach of U-Net, DCN concatenates all the features of multiple scales at once and uses a sequence of convolutions following the concatenation to fuse the multi-scale features and produce the final segmentation results. The exact placements of the gates are shown in Fig. 1 (top-right). During training, the gates are scheduled as follows. Iter-1: g_1 = 1, g_2, g_3, g_4, g_5, g_6= 0; iter-2: g_1, g_2 = 1, g_3, g_4, g_5, g_6 = 0; iter-3: g_1, g_2, g_3 = 1, g_4, g_5, g_6 = 0; iter-4: g_1, g_2, g_3, g_4= 1,

g_5, g_6 = 0; iter-5: g_1, g_2 ,g_3, g_4, g_5 = 1, g_6 = 0; iter-6: g_1, g_2 ,g_3, g_4, g_5, g_6 = 1; iter-7: g_1 = 1, g_2, g_3, g_4, g_5, g_6 = 0; these patterns continue.

DenseVoxNet (DVN). DenseVoxNet [3] utilizes dense between-layer skip connections to progressively combine earlier layer (lower-level) features with later layer (higher-level) features. Placing gates and controlling the forward and backward propagation flows in such fine-grained levels incur additional challenges in gate design and scheduling. Instead of putting gates at the layer-level, we create two gates and put them at the stage-level (see Fig. 1 (bottom-left)). The network output is then slightly modified from taking only the 2nd stage's output to combining (adding) the 1st stage's output with the 2nd stage's output. During training, the gates are scheduled as follows. Iter-1: g_1 = 1, g_2 = 0; iter-2: g_1, g_2 = 1; iter-3: g_1 = 1, g_2 = 0; these patterns continue.

HRNet. Fig. 1 (bottom-right) shows an architectural overview of the state-of-the-art segmentation network/backbone for HRNet [4]. It fuses multi-scale information between each computation stage, where lower-level features are fused into higher-level features. We put the multi-scale fusion gates at the last stage.

During training, the gates are scheduled as follows. Iter-1: g_1 = 1, g_2, g_3, g_4 = 0; iter-2: g_1, g_2 = 1, g_3, g_4 = 0; iter-3: g_1, g_2, g_3 = 1, g_4 = 0; iter-4: g_1, g_2, g_3, g_4 = 1; iter-5: g_1 = 1, g_2, g_3, g_4 = 0; these patterns continue.

2.3. Enabling a More Informative Error Analysis

As mentioned in Section 2.1, a segmentation network f^{g_h,g_m,g_l} after gate-regularized training can be applied in a normal way in which all the gates are set to 1. Setting all the gates to 1 is equivalent to removing all the gates from the network. However, if one aims to perform a more thorough error analysis when evaluating a segmentation network, these gates can be useful to provide more information for the error analysis.

Segmentation errors are due to various causes; some are caused by less accurate boundary reconstruction, and some are caused by incorrect semantic prediction of target objects. Traditionally, a trained network provides a single segmentation output, and we compare the output with the ground truth (GT) annotation for evaluating the overall segmentation performance. But, little effort has been conducted on determining whether a particular error (in an image) is caused by less robust high-level (semantic) features or due to less accurate low-level shape reconstruction. Our GrNT automatically provides a way to address this issue.

For a given test image x_i, we apply the trained f^{g_l,g_m,g_h} with different gates: $\hat{y}_i^h = f^{g_l=0,g_m=0,g_h=1}(x_i)$ and $\hat{y}_i^{full} = f^{g_l=1,g_m=1,g_h=1}(x_i)$. We compare $\hat{y}_i^h$ with the GT annotation y_i, and obtain a set P^h containing pixel locations exhibiting errors. Similarly, we obtain pixel locations with errors in a set P^{full} by comparing $\hat{y}_i^{full}$ with y_i. Let P be the set containing all the pixel locations in the test image x_i. We take $P^h \cap P^{full}$ to obtain the pixels with errors exclusively caused by high-level semantic features. This additional performance metric is a useful tool when one needs to find a more robust network backbone during architecture search and/or model selection, since errors caused by high-level features are directly caused by a less robust backbone architecture.

Table 1. Comparison of segmentation results on the 2017 ISIC Skin Lesion dataset. * indicates a deeper version of the original network.

Method	Jaccard index	Dice	Sensitivity	Specificity
Yuan et al. [13]	0.765	0.849	0.825	0.975
Li et al. [14]	0.765	0.866	0.825	0.984
Mirikharaji et al. [15]	0.773	0.857	0.855	0.973
Xie et al. [16]	0.788	0.868	0.884	0.957
U-Net [1]*	0.778	0.877	0.815	0.986
w/ GrNT	**0.785**	**0.882**	**0.829**	**0.986**
Ablation: w/ random gating	0.774	0.875	0.815	0.985
Attention U-Net [5]*	0.779	0.878	0.821	**0.985**
w/ GrNT	**0.791**	**0.886**	**0.846**	0.981
Ablation: w/ random gating	0.770	0.873	0.807	0.987
DCN [2]*	0.787	0.883	0.822	0.987
w/ GrNT	**0.794**	**0.887**	**0.832**	**0.987**
Ablation: w/ random gating	0.778	0.877	0.817	0.987

3. EXPERIMENTS AND RESULTS

Implementation Details. Our GrNT method is implemented with Tensorflow, trained on an Nvidia Tesla V100 Graphics Card with 32GB GPU memory using the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-10$). All the models are initialized using a Gaussian distribution and the "poly" learning rate policy, $L_r \times \left(1 - \frac{iter}{\#iter}\right)$, is applied; the initial learning rate is $5e-4$, and the maximum number of iterations is 60k times the number of gates embedded in the respective model. We apply the standard data augmentation (e.g., random cropping, rotation, and flipping) to reduce overfitting. All the experiments are performed 5 times, each time with a different random seed.

Only the mean values across the 5 runs are reported in the tables below.

(1) The 2017 ISIC Skin Lesion Segmentation Dataset. This dataset [17] aims to segment lesion boundaries in 2D dermoscopic images. It contains 2000 training, 150 validation, and 600 test images. Following [16, 13], we resize the images to 224×224 as input, and apply a dual-threshold method to generate the final results for all the settings. Table 1 presents the results. First, observe that our GrNt can improve the performance of multiple segmentation networks and backbones consistently across nearly all the evaluation measures. Second, by leveraging our GrNt, DCN [2]* attains the highest Jaccard index, Dice, and specificity, though its sensitivity is slightly lower than that of some other methods. This demonstrates the effectiveness of our GrNt method.

(2) The 2017 MM-WHS CT Dataset. This dataset [18] aims to segment seven cardiac structures (left/right ventricle blood

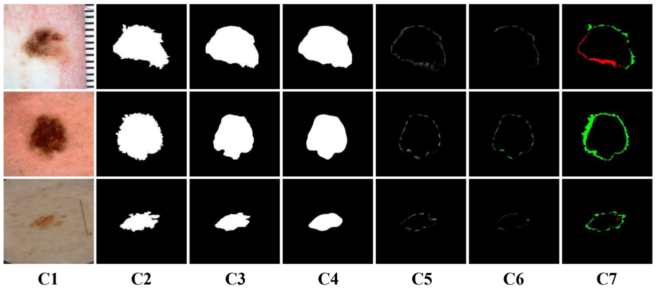

Fig. 2. From left to right: raw images (C1), GT (C2), segmentation using full features (C3), segmentation using only high-level features (C4), differences between these two types of segmentation results (C5), errors due to lower-level features (C6), and errors due to high-level features (C7). Red: false positives; green: false negatives.

cavity (LV/RV), left/right atrium blood cavity (LA/RA), myocardium of the left ventricle (LV-myo), ascending aorta (AO), and pulmonary artery (PA)) in 3D CT images. It contains 20 unpaired CT images.

Our training/test split (16 and 4 images) is the same as in [19]. Table 2 shows that our GrNT can consistently improve segmentation results across multiple networks. The improvements are especially noticeable for surface/boundary based metrics (ADB and Hausdorff).

(3) The 2016 HVSMR Dataset. This dataset [20] is for segmentation in 3D cardiovascular MR images, with two objects of interest: blood pool and myocardium. Evaluations are performed on the organizers' server. We show in Table 3 that 3D U-Net is improved by GrNT considerably.

Error Analysis. As described in Section 2.3, GrNT enables disentangling segmentation errors caused by high-level features and lower-level features. In inference, segmentation using high-level features is generated by setting $g_1 \leftarrow 1$ and all the other gates as 0. Segmentation using full features is attained by setting all the gates as 1. The last two columns of Fig. 2 showcase the disentangled segmentation errors caused by lower-level features and high-level features.

Ablation Study. We conduct ablation study using the 2017 ISIC Skin Lesion and MM-WHS CT datasets to examine the effectiveness of our scheduled gates in training. In each training iteration, instead of assigning specific values to the gates (i.e., using the scheduled gates), we randomly assign 0 or 1 to the gates (similar to random dropout), which we denote as "w/ random gating". From Table 1 and Table 2, one may observe: (1) when using random gating, the performance drops significantly compared to our GrNT; (2) when using random gating, in some cases, the performance can be even worse than that of the original models. These observations demonstrate the effectiveness of our GrNT.

Table 2. Comparison of segmentation results on the 2017 MM-WHS CT dataset. * indicates a deeper version of the original network. AB denotes ablation study.

Method	Metric	LV	RV	LA	RA	LV-myo	AO	PA	Mean
Payer et al. [21]	Dice	0.918	0.909	0.929	0.888	0.881	0.933	0.840	0.900
Dou et al. [22]	Dice	0.888	—	0.891	—	0.733	0.813	—	—
HFA-Net [19]	Dice	0.946	0.893	0.925	0.897	0.910	0.964	0.830	0.909
	Hausdorff	7.148	33.128	42.173	22.903	36.954	12.075	37.845	27.461
	ADB	0.076	0.562	0.210	0.334	0.225	0.103	1.685	0.456
Chen et al. [23]	Dice	0.919	—	0.911	—	0.877	0.927	—	0.909
U-Net [1]*	Dice	0.952	0.894	0.937	0.906	0.920	0.968	0.835	0.916
	Hausdorff	5.837	35.081	18.014	28.942	8.423	8.621	31.814	19.534
	ADB	0.073	0.682	0.158	0.266	0.113	0.067	1.567	0.418
w/ GrNT	Dice	0.952	0.901	0.933	0.911	0.921	0.973	0.843	**0.920**
	Hausdorff	5.587	13.928	19.345	15.148	7.326	7.795	37.065	**15.172**
	ADB	0.068	0.340	0.170	0.248	0.104	0.040	1.500	**0.353**
AB: w/ random gating	Dice	0.947	0.898	0.935	0.899	0.914	0.961	0.834	0.912
	Hausdorff	14.969	28.033	29.419	17.187	26.875	18.553	42.928	25.424
	ADB	0.150	0.605	0.185	0.311	0.129	0.138	1.671	0.456
Attention U-Net [5]*	Dice	0.952	0.898	0.937	0.908	0.921	0.969	0.838	0.918
	Hausdorff	5.404	32.863	18.069	17.844	15.980	16.237	31.687	19.726
	ADB	0.069	0.493	0.164	0.261	0.115	0.072	1.527	0.386
w/ GrNT	Dice	0.953	0.900	0.936	0.910	0.919	0.972	0.840	**0.919**
	Hausdorff	6.041	13.923	12.875	14.260	6.544	7.590	36.395	**13.947**
	ADB	0.068	0.325	0.165	0.243	0.110	0.041	1.547	**0.357**
AB: w/ random gating	Dice	0.949	0.897	0.936	0.882	0.912	0.954	0.833	0.909
	Hausdorff	6.550	29.766	18.648	25.080	14.352	22.907	48.808	23.730
	ADB	0.076	0.371	0.162	0.389	0.128	0.192	1.672	0.427
DCN [2]*	Dice	0.953	0.896	0.937	0.912	0.922	0.971	0.830	0.918
	Hausdorff	9.148	15.282	12.180	14.201	12.404	12.789	32.461	15.458
	ADB	0.069	0.385	0.163	0.238	0.108	0.066	1.628	0.380
w/ GrNT	Dice	0.951	0.905	0.940	0.906	0.922	0.972	0.834	**0.920**
	Hausdorff	5.993	14.933	11.987	15.959	8.208	7.923	34.428	**14.099**
	ADB	0.072	0.344	0.155	0.268	0.104	0.040	1.574	**0.365**
AB: w/ random gating	Dice	0.950	0.907	0.936	0.891	0.917	0.961	0.824	0.912
	Hausdorff	5.753	14.887	11.186	15.306	8.334	10.604	37.746	15.031
	ADB	0.073	0.376	0.150	0.318	0.111	0.080	1.658	0.395
DVN [3]	Dice	0.945	0.878	0.931	0.871	0.900	0.956	0.830	0.901
	Hausdorff	35.249	74.381	86.127	77.582	41.516	49.993	55.525	60.053
	ADB	0.096	0.765	0.335	1.858	0.761	0.300	1.684	0.828
w/ GrNT	Dice	0.948	0.883	0.931	0.874	0.913	0.960	0.833	**0.906**
	Hausdorff	10.322	70.463	34.175	111.613	37.569	52.133	52.328	**52.658**
	ADB	0.083	0.441	0.271	0.719	0.143	0.212	1.652	**0.503**

Table 3. Comparison of segmentation results on the 2016 HVSMR dataset.

Method	Myocardium			Blood pool			Overall score
	ADB	Dice	Hausdorff	ADB	Dice	Hausdorff	
3D U-Net [24]	0.858	0.791	5.026	0.848	0.934	8.125	−0.079
w/ GrNT	**0.778**	**0.805**	**4.475**	**0.800**	**0.934**	**7.287**	**0.083**

4. CONCLUSIONS

In this paper, we presented a new gate-regularized network training (GrNT) method for improving multi-scale information fusion, which is a key process in medical image segmentation. Theoretical analysis and empirical study demonstrated that our GrNT is effective in improving segmentation results by encouraging high-level features to drive the generation of the segmentation. As a bonus, GrNT offers a new means for a more informative error analysis and model interpretability by disentangling segmentation errors caused by features of different levels.

5. COMPLIANCE WITH ETHICAL STANDARDS

This research study was conducted retrospectively using human subject data made available in open access by two publicly available datasets [17, 18, 20]. Ethical approval was not required as confirmed by the licenses attached with the open access datasets.

6. ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant 62201263) Natural Science Foundation of Jiangsu Province (Grant BK20220949), and NSF Grant CCF-1617735.

7. REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[2] H. Chen, X. Qi, J.-Z. Cheng, and P.-A. Heng, "Deep contextual networks for neuronal structure segmentation," in *AAAI*, 2016, pp. 1167–1173.

[3] L. Yu, J.-Z. Cheng, Q. Dou, X. Yang, H. Chen, J. Qin, and P.-A. Heng, "Automatic 3D cardiovascular MR segmentation with densely-connected volumetric ConvNets," in *MICCAI*, 2017, pp. 287–295.

[4] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al., "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[5] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, et al., "Attention U-Net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[6] D.-P. Fan, T. Zhou, G.-P. Ji, Y. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Inf-Net: Automatic covid-19 lung infection segmentation from ct images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2626–2637, 2020.

[7] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "No New-Net," in *International MICCAI Brainlesion Workshop*, 2018, pp. 234–244.

[8] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[9] Y. Zhang, L. Yang, H. Zheng, P. Liang, C. Mangold, R. G. Loreto, D. P. Hughes, and D. Z. Chen, "SPDA: Superpixel-based data augmentation for biomedical image segmentation," in *MIDL*, 2019, pp. 572–587.

[10] R. Sheikh and T. Schultz, "Feature preserving smoothing provides simple and effective data augmentation for medical image segmentation," in *MICCAI*, 2020, pp. 116–126.

[11] Y. Yang, D. Zhou, D. Zhan, H. Xiong, and Y. Jiang, "Adaptive deep models for incremental learning: Considering capacity scalability and sustainability," in *KDD*, 2019, pp. 74–82.

[12] Y. Yang, Z.-Q. Sun, H. Zhu, Y. Fu, Y. Zhou, H. Xiong, and J. Yang, "Learning adaptive embedding considering incremental class," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[13] Y. Yuan and Y.-C. Lo, "Improving dermoscopic image segmentation with enhanced convolutional-deconvolutional networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 519–526, 2017.

[14] H. Li, X. He, F. Zhou, Z. Yu, D. Ni, S. Chen, T. Wang, and B. Lei, "Dense deconvolutional network for skin lesion segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 527–537, 2018.

[15] Z. Mirikharaji and G. Hamarneh, "Star shape prior in fully convolutional networks for skin lesion segmentation," in *MICCAI*, 2018, pp. 737–745.

[16] Y. Xie, J. Zhang, H. Lu, C. Shen, and Y. Xia, "SESV: Accurate medical image segmentation by predicting and correcting errors," *IEEE Transactions on Medical Imaging*, vol. 40, no. 1, pp. 286–296, 2021.

[17] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, et al., "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)," in *IEEE, ISBI*, 2018, pp. 168–172.

[18] X. Zhuang and J. Shen, "Multi-scale patch and multi-modality atlases for whole heart segmentation of MRI," *Medical Image Analysis*, vol. 31, pp. 77–87, 2016.

[19] H. Zheng, L. Yang, J. Han, Y. Zhang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen, "HFA-Net: 3D cardiovascular image segmentation with asymmetrical pooling and content-aware fusion," in *MICCAI*, 2019, pp. 759–767.

[20] D. F. Pace, A. V. Dalca, T. Geva, A. J. Powell, M. H. Moghari, and P. Golland, "Interactive whole-heart segmentation in congenital heart disease," in *MICCAI*, 2015, pp. 80–88.

[21] C. Payer, D. Štern, H. Bischof, and M. Urschler, "Multi-label whole heart segmentation using CNNs and anatomical label configurations," in *International Workshop, STACOM*, 2017, pp. 190–198.

[22] Q. Dou, C. Ouyang, C. Chen, H. Chen, and P.-A. Heng, "Unsupervised cross-modality domain adaptation of ConvNets for biomedical image segmentations with adversarial loss," in *IJCAI*, 2018, pp. 691–697.

[23] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng, "Unsupervised bidirectional cross-modality adaptation via deeply synergistic image and feature alignment for medical image segmentation," *IEEE Transactions on Medical Imaging*, 2020.

[24] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *MICCAI*, 2016, pp. 424–432.