



3D Anisotropic Hybrid Network: Transferring Convolutional Features from 2D Images to 3D Anisotropic Volumes

Siqi Liu¹(✉), Daguang Xu¹, S. Kevin Zhou¹, Olivier Pauly², Sasa Grbic¹,
Thomas Mertelmeier², Julia Wicklein², Anna Jerebko², Weidong Cai³,
and Dorin Comaniciu¹

¹ Medical Imaging Technologies, Siemens Healthineers, Princeton, NJ, USA
siqi.liu@siemens-healthineers.com

² X-Ray Products, Siemens Healthineers, Erlangen, Germany

³ School of Information Technologies, University of Sydney, Sydney, Australia

Abstract. While deep convolutional neural networks (CNN) have been successfully applied to 2D image analysis, it is still challenging to apply them to 3D medical images, especially when the within-slice resolution is much higher than the between-slice resolution. We propose a 3D Anisotropic Hybrid Network (AH-Net) that transfers convolutional features learned from 2D images to 3D anisotropic volumes. Such a transfer inherits the desired strong generalization capability for within-slice information while naturally exploiting between-slice information for more effective modelling. We experiment with the proposed 3D AH-Net on two different medical image analysis tasks, namely lesion detection from a Digital Breast Tomosynthesis volume, and liver and liver tumor segmentation from a Computed Tomography volume and obtain state-of-the-art results.

1 Introduction

3D volumetric images (or volumes) are widely used for clinical diagnosis, intervention planning, and biomedical research. However, given the additional dimension, it is more time consuming and sometimes harder to interpret 3D volumes than 2D images by machines. Many imaging modalities come with anisotropic voxels, meaning not all of the three dimensions have equal resolutions, for example the Digital Breast Tomosynthesis (DBT) and sometimes Computed Tomography (CT). Directly applying 3D CNN to such images remains challenging due to the following reasons: (1) It may be hard for a small $3 \times 3 \times 3$ kernel to learn useful features from anisotropic voxels. (2) The capability of 3D networks is bounded by the GPU memory, constraining both the width and depth of the networks. (3) 3D tasks mostly have to train from scratch, and hence suffer from the lack of large 3D datasets. In addition, the high data biases make the 3D

D. Xu—Equal contribution.

networks harder to generalize. Besides the traditional 3D networks built with $1 \times 1 \times 1$ and $3 \times 3 \times 3$ kernels, there are other methods for learning representations from anisotropic voxels. Some studies process 2D slices separately with 2D networks [9]. To make a better use of the 3D context, more than one image slice is used as the input for 2D networks [8]. The 2D slices can also be viewed sequentially by combining a fully convolutional network (FCN) architecture with Convolutional LSTM [1]. Anisotropic convolutional kernels were used to distribute more learning capability on the xy plane [7].

In this paper, we propose the 3D Anisotropic Hybrid Network (AH-Net) to learn informative features for object detection and segmentation tasks in 3D medical images. To obtain the 3D AH-Net, we firstly train a 2D fully convolutional ResNet [10] which is initialized with pre-trained weights and uses multiple 2D image slices as inputs. The feature encoder of such a 2D network is then transformed into a 3D network by extending the 2D kernel with one added dimension. Then we add a feature decoder sub-network to extract the 3D context. The feature decoder consists of anisotropic convolutional blocks with $3 \times 3 \times 1$ and $1 \times 1 \times 3$ convolutions. Different anisotropic convolutional blocks are combined with dense connections [5]. Similar to the U-Net [11], we use skip connections between the feature encoder and the decoder. A pyramid volumetric pooling module [13] is stacked at the end of the network before the final output layer for extracting multiscale features. Since the AH-Net can make use of 2D networks pre-trained with large 2D general image datasets such as ImageNet [12], it is easier to train as well as to generalize. The anisotropic convolutional blocks enable the exploiting of 3D context. With end-to-end inference as a 3D network, the AH-Net runs much faster than the conventional multi-channel 2D networks regarding the GPU time required for processing each 3D volume.

2 Anisotropic Hybrid Network

The AH-Net is designed for the object detection and the segmentation tasks in 3D medical images. It is able to transfer learnt 2D networks to 3D learning problems and further exploit 3D context information. As an image-to-image network, the AH-Net consists of a feature encoder and a feature decoder as shown in Fig. 1. The encoder, transformed from a fine-tuned 2D network, is designed for extracting the deep representations from 2D slices with high resolution. The decoder built with densely connected blocks of anisotropic convolutions is responsible for exploiting the 3D context and maintaining the between-slice consistency. The network training is performed in two stages: the 2D encoder is firstly trained and transformed into a 3D encoder; then the 3D decoder is added and fine-tuned with the encoder parameters locked.

2.1 Pre-training a Multi-Channel 2D Feature Encoder

To obtain a pre-trained 2D image-to-image network, we train a 2D multi-channel global convolutional network (MC-GCN) similar to the architecture in [10] to

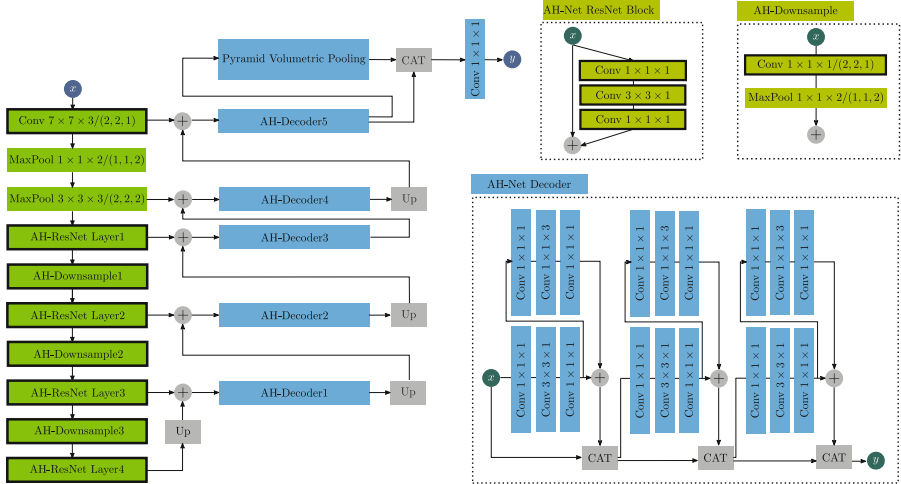


Fig. 1. The architecture of 3D AH-Net. We hide the batch normalization and ReLU layers for brevity. The parameters of the blocks with bold boundaries are transferred from the pre-trained 2D ResNet50 encoder.

extract the 2D within-slice features at different resolutions. We choose the 2D ResNet50 model [4] as the backbone network, which is initialized by pre-training with the ImageNet images [12]. The network is then fine-tuned with 2D image slices extracted from the 3D volumes. The inputs to this network are three neighbouring slices (as RGB channels). Thus, the entire architecture of the ResNet50 remains unchanged. With a 2D decoder upscaling the responses to the original resolution as described in [10], the MC-GCN outputs response maps with the same dimensions as the input slices. To fine tune this network, the scaled L2 loss is used for object detection and weighted cross entropy is used for segmentation.

2.2 Transferring the Learned 2D Features into 3D AH-Net

We extract the parameters of the trained ResNet50 encoder from the 2D MC-GCN and transfer them to the corresponding encoder layers of the AH-Net. The decoder of the MC-GCN is thus discarded. The input and output of AH-Net are now 3D volumes. The transformation of the convolution tensors from 2D to 3D aims to perform 2D convolutions on 3D volumes slice by slice in the encoder of the AH-Net. Overall, we permute the weight tensors of the first convolution layer so the channel dimension becomes the z-dimension. For the rest of the encoder, we treat 2D filters as 3D filters by setting the extra dimension as 1.

Notations. A 2D convolutional tensor is denoted by $T_{n \times m \times h \times w}^i$, where n , m , h , and w respectively represent the number of output channels, the number of input channels, the height and width of the i^{th} convolution layer. Similarly, a 3D weight tensor is denoted by $T_{n \times m \times h \times w \times d}^i$ where d is the filter depth. We use

$P^{(b,a,c,d)}(T_{a \times b \times c \times d})$ to denote the dimension permutation of a tensor $T_{a \times b \times c \times d}$, resulting in a new tensor $T_{b \times a \times c \times d}$ with the 1st and 2nd dimensions switched. $P^{(a,*,b,c,d)}(T_{a \times b \times c \times d})$ adds an identity dimension between the 1st and 2nd dimensions of the tensor $T_{a \times b \times c \times d}$ and gives $T_{a \times 1 \times b \times c \times d}$. We define a convolutional layer as Conv $K_x \times K_y \times K_z / (S_x, S_y, S_z)$, where K_x, K_y and K_z are the kernel sizes; S_x, S_y and S_z are the stride step size in each direction. Max pooling layers are denoted by MaxPool $K_x \times K_y \times K_z / (S_x, S_y, S_z)$. The stride is omitted when a layer has a stride size of 1 in all dimensions.

Input Layer Transform. The input layer of the 2D ResNet50 contains a convolutional weight tensor $T_{64 \times 3 \times 7 \times 7}^1$. The 2D convolutional tensor $T_{64 \times 3 \times 7 \times 7}^1$ is transformed into 3D as

$$P^{(1,*,3,4,2)}(T_{64 \times 3 \times 7 \times 7}^1) = T_{64 \times 1 \times 7 \times 7 \times 3}^1 \tag{1}$$

in order to form a 3D convolution kernel that convolves 3 neighbouring slices. To keep the output consistent with the 2D network, we only apply stride-2 convolutions on the xy plane and stride 1 on the third dimension. This results in the input layer Conv $7 \times 7 \times 3 / (2, 2, 1)$. To downsample the z dimension, we use a MaxPool $1 \times 1 \times 2 / (1, 1, 2)$ to fuse every pair of the neighbouring slices. An additional MaxPool $2 \times 2 \times 2 / (2, 2, 2)$ is used to keep the feature resolution consistent with the 2D network.

ResNet Block Transform. All the 2D convolutional tensors $T_{n \times m \times 1 \times 1}^i$ and $T_{n \times m \times 3 \times 3}^i$ in the ResNet50 are transformed as

$$P^{(1,2,3,4,*)}(T_{n \times m \times 1 \times 1}^i) = T_{n \times m \times 1 \times 1 \times 1}^i \tag{2}$$

and

$$P^{(1,2,3,4,*)}(T_{n \times m \times 3 \times 3}^i) = T_{n \times m \times 3 \times 3 \times 1}^i. \tag{3}$$

In this way, all the ResNet Conv $3 \times 3 \times 1$ blocks only perform 2D slice-wise convolutions on the 3D volume within the xy plane. The original downsampling between ResNet blocks is performed with Conv $1 \times 1 / (2, 2)$. However, in a 3D volume, a Conv $1 \times 1 \times 1 / (2, 2, 2)$ skips a slice for every step on the z dimension. This would miss important information when the image only has a small z -dimension. We therefore use a Conv $1 \times 1 \times 1 / (2, 2, 1)$ following by a MaxPool $1 \times 1 \times 2 / (1, 1, 2)$ to downsample the 3D feature maps between the ResNet blocks.

2.3 Anisotropic Hybrid Decoder

Accompanying the transformed encoder, an anisotropic 3D decoder sub-network is added to exploit the 3D anisotropic image context with chained separable convolutions as shown in Fig. 1. In the decoder, anisotropic convolutional blocks with Conv $1 \times 1 \times 1$, Conv $3 \times 3 \times 1$ and Conv $1 \times 1 \times 3$ are used. The features are passed into an xy bottleneck block at first with a Conv $3 \times 3 \times 1$ surrounded by two layers of Conv $1 \times 1 \times 1$. The output is then forwarded to another bottleneck block with a Conv $1 \times 1 \times 3$ in the middle and summed with itself before being

forwarded to the next block. This anisotropic convolution block decomposes a 3D convolution into 2D and 1D convolutions. It receives the inputs from the previous layers using a 2D convolution at first, preserving the detailed 2D features. Conv $1 \times 1 \times 3$ mainly fuses the within-slice features to keep the z dimension output consistent.

Three anisotropic convolutional blocks are connected as the densely connected neural network [5] using feature concatenation for each resolution of encoded features. The features received from each resolution of the encoder are firstly projected to match the number of features of the higher encoder feature resolution using a Conv $1 \times 1 \times 1$. They are then upsampled using the 3D tri-linear interpolation and summed with the encoder features from a higher resolution. The summed features are forwarded to the decoder blocks in the next resolution.

At the end of the decoder network, we add a pyramid volumetric pooling module [13] to obtain multi-scaled features. The output features of the last decoder block are firstly down-sampled using 4 different Maxpooling layers, namely MaxPool $64 \times 64 \times 1$, MaxPool $32 \times 32 \times 1$, MaxPool $16 \times 16 \times 1$ and MaxPool $8 \times 8 \times 1$ to obtain a feature map pyramid. Conv $1 \times 1 \times 1$ layers are used to project each resolution in the feature pyramid to a single response channel. The response channels are then interpolated to the original size and concatenated with the features before downsampling. The final outputs are obtained by applying a Conv $1 \times 1 \times 1$ projection layer on the concatenated features.

2.4 Training the AH-Net

Training the AH-Net using the same learning rate on both the pre-trained encoder and the randomly initialized decoder would make the network difficult to optimize. To train the 3D AH-Net, all the transferred parameters are locked at first. Only the decoder parameters are fine-tuned in the optimization. All the parameters can be then fine-tuned altogether afterwards to the entire AH-Net jointly. Though it is optional to unlock all the parameters for fine-tuning afterwards, we did not observe better performance. We use the scaled L2 loss for training the network for object detection tasks and the weighted cross entropy for segmentation tasks. We use ADAM [6] to optimise all the compared networks with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We use the initial learning-rate 0.0005 to fine-tune the 2D MC-GCN. Then, the learning rate is increased to 0.001 to fine-tune the AH-Net after the 2D network is transferred.

3 Experimental Results

To demonstrate the efficacy and efficiency of the proposed 3D AH-net, we conduct two experiments, namely lesion detection from a DBT volume and liver tumor segmentation from a CT volume. All the evaluated networks are implemented in Pytorch (<https://github.com/pytorch>).

3.1 Breast Lesion Detection from DBT

We use an in-house database containing 2809 3D DBT volumes acquired from 12 sites globally. The DBT volume has an anisotropic resolution of $0.085 \text{ mm} \times 0.085 \times 1 \text{ mm}$. We have experienced radiologists annotate and validate the lesions in DBT volumes as 3D bounding boxes. To train the proposed networks for lesion detection, we generate 3D multi-variant Gaussian heatmaps based on the annotated 3D boxes that have the same sizes as the original images. We randomly split the database into the training set with 2678 volumes (1111 positives) and the testing sets with 131 volumes (58 positives). We ensure the images from the same patient could only be found either in the training or the testing set. For training, we extract $256 \times 256 \times 32$ 3D patches. 70% of the training patches are sampled as positives with at least one lesion included, considering the balance between the voxels within and without a breast lesion. The patches are sampled online asynchronously to form the mini-batches.

Along with the proposed networks, we also train 2D and 3D U-Nets with the identical architecture and parameters [2,11] as the two baselines. The 2D U-Net is also trained with input having three input channels. The 3D U-Net is trained with the same patch sampling strategies as the AH-Net. We measure the GPU inference time of networks by forwarding a 3D DBT volume of size $384 \times 256 \times 64$ 1000 times on an NVIDIA GTX 1080Ti GPU respectively. The GPU inference of the AH-Net (17.7 ms) is 43 times faster than that of the 2D MC-GCN (775.2 ms) though the AH-Net has more parameters. The speed gain could be achieved mostly by avoiding repetitive convolutions on the same slices required by multi-channel 2D networks.

By altering a threshold to filter the response values, we can control the balance between the False Positive Rate (FPR) and True Positive Rate (TPR). TPR represents the percentage of lesions that have been successfully detected by the network. FPR represents the percentage of lesions that the network predicted that are false positives. The lesion detected by the network is considered a true

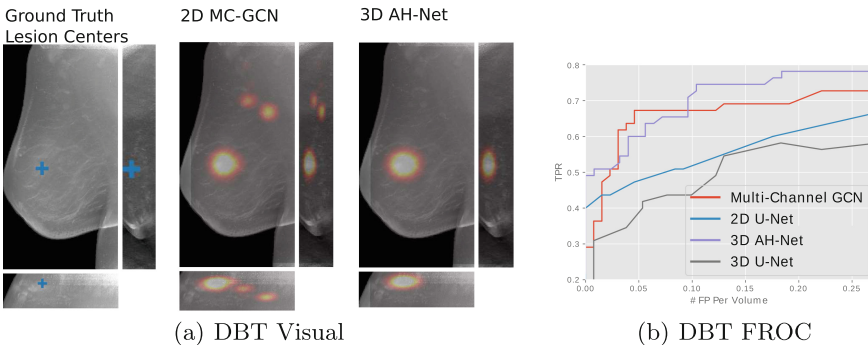


Fig. 2. Left: The visual comparisons of the network responses on a DBT volume from 2D MC-GCN and the 3D AH-Net with the encoder weights transferred from it. Right: FROC curves of the compared networks on the DBT dataset.

positive finding if the maximal point resides in a 3D bounding box annotated by the radiologist. Similarly, if a bounding box contains a maximal point, we consider it is detected by the network. They are otherwise considered as false positives. We evaluate the lesion detection performance by plotting the Free Response Operating Characteristic (FROC) curves as shown in Fig. 2(b). The proposed AH-Net outperforms both the 2D and 3D U-Net with large margins. Compared to the performance of the 2D MC-GCN, the 3D AH-Net generates higher TPR for a majority of thresholds, except the region around 0.05 per volume false positives. It is noticeable that AH-Net also obtains nearly 50% TPR even when only 0.01 false positive findings are allowed per volume.

3.2 Liver and Liver Tumor Segmentation from CT

The second evaluation dataset was obtained from the liver lesion segmentation challenge in MICCAI 2017 (lits-challenge.com), which contains 131 training and 70 testing 3D contrast-enhanced abdominal CT scans. The ground truth masks contain both liver and lesion labels. Most CT scans consist of anisotropic resolution: the between-slice resolution ranges from 0.45 mm to 6.0 mm while the within-slice resolution varies from 0.55 mm to 1.0 mm. In preprocessing, the abdominal regions are truncated from the CT scans using the liver center landmark detected by a reinforcement learning based algorithm [3]. Due to the limited number of training data, we applied random rotation (within $\pm 20^\circ$ in the xy plane), random scaling (within ± 0.2 in all directions), and random mirror (within xy plane) to reduce overfitting.

The performance of the AH-Net is listed in Table 1, together with other top-ranked submissions retrieved from the LITS challenge leaderboard. These submissions employ various types of neural network architectures: 2D, 3D, 2D-3D hybrid, and model fusion. Two evaluation metrics are adapted: (1) Dice Global (DG) which is the dice score combining all the volumes into one; (2) Dice per Case (DPC) which averages the dice scores of every single case. The Dice score between two masks is defined as $DICE(A, B) = 2|A \cap B|/(|A| + |B|)$. Our results achieve state-of-the-art performance in three of the four metrics.

Table 1. The liver lesion segmentation (LITS) challenge results with the dice global (Dice-G) and dice per case (Dice-PC). The compared results were obtained from the LITS challenge leaderboard (lits-challenge.com/#results) before the paper submission.

Method	Lesion		Liver	
	Dice-G	Dice-PC	Dice-G	Dice-PC
leHealth	0.794	0.702	0.964	0.961
H-DenseNet [8]	0.829	0.686	0.965	0.961
deepX	0.820	0.657	0.967	0.963
MC-GCN	0.788	0.593	0.963	0.951
3D AH-Net	0.834	0.634	0.970	0.963

4 Conclusion

In this paper, we propose the 3D Anisotropic Hybrid Network (3D AH-Net) which is capable of transferring the convolutional features of 2D images to 3D volumes with anisotropic resolution. By evaluating the proposed methods on both a large-scale in-house DBT dataset and a highly competitive open challenge dataset of CT liver and lesion segmentation, we show our network obtains state-of-the-art results. The GPU inference of the AH-Net is also much faster than piling the results from a 2D network.

Disclaimer: This feature is based on research, and is not commercially available. Due to regulatory reasons, its future availability cannot be guaranteed.

References

1. Chen, J., Yang, L., Zhang, Y., Alber, M.S., Chen, D.Z.: Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation. In: NIPS (2016)
2. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. ArXiv eprints [arXiv:1606.06650](https://arxiv.org/abs/1606.06650) (2016)
3. Ghesu, F.C., Georgescu, B., Grbic, S., Maier, A.K., Hornegger, J., Comaniciu, D.: Robust multi-scale anatomical landmark detection in incomplete 3D-CT data. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10433, pp. 194–202. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66182-7_23
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of CVPR, pp. 770–778 (2016)
5. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely Connected Convolutional Networks. ArXiv eprints [arXiv:1608.06993](https://arxiv.org/abs/1608.06993) (2016)
6. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. ArXiv eprints [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
7. Lee, K., Zung, J., Li, P., Jain, V., Seung, H.S.: Superhuman Accuracy on the SNEMI3D Connectomics Challenge. ArXiv e-prints [arXiv:1706.00120](https://arxiv.org/abs/1706.00120) (2017)
8. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.W., Heng, P.A.: H-DenseUNet: Hybrid Densely Connected UNet for Liver and Liver Tumor Segmentation from CT Volumes. ArXiv e-prints [arXiv:1709.07330](https://arxiv.org/abs/1709.07330) (2017)
9. Liu, F., Zhou, Z., Jang, H., Samsonov, A., Zhao, G., Kijowski, R.: Deep convolutional neural network and 3D deformable approach for tissue segmentation in musculoskeletal MR imaging. *Magn. Reson. Med.* (2017)
10. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large Kernel Matters - Improve Semantic Segmentation by Global Convolutional Network. ArXiv eprints [arXiv:1703.02719](https://arxiv.org/abs/1703.02719) (2017)
11. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv eprints [arXiv:1505.04597](https://arxiv.org/abs/1505.04597) (2015)
12. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. *IJCV* **115**(3), 211–252 (2015)
13. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid Scene Parsing Network. ArXiv eprints [arXiv:1612.01105](https://arxiv.org/abs/1612.01105) (2016)