Research Report: 3D U-Net Development and Implementation

Mingzhe Ma
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
nick.ma.cheers@gmail.com

***Abstracts* - Deep learning and neural networks has been actively researched and explored during the past decade. U-Net model generally outperforms other models particularly in biomedical areas for segmentation tasks, due to its high accuracy while given only small size of training data. Various available libraries and public code repositories enable fast implementations and stable deployments of 3D U-Net applications, which notably contributes to the future utilisation of AI in medicine.**

I. INTRODUCTION

3D U-Net was first systematically illustrated and published by Özgün Çiçek et al. regarding their research on volumetric segmentation on images of Xenopus kidney embryos in 2016 [1], and then derived multiple variations modifications and got repetitively examined in many experiments and applications, especially in biomedical and public health areas. In this report, I would firstly introduce the evolution of the 3D U-Net and key concepts behind each of the core techniques; secondly, discussed the pros and cons of basic U-Net and other iconic modifications with support of real cases; and thirdly specify the usage of a few popular libraries and source codes that could be applicable in developing light 3D U-Net models. The report was drafted for the purpose of presenting my discoveries and understanding of the related topics, and thus ensured my preparation for future development and further research work.

II. KEY CONCETPS AND ARCHITECTURE

The development of 3D U-Net was an extension of the original Fully Convolutional Networks (FCN) based 2D U-Net, inspired by the earlier attempts of using 3D CNNs [2] and other usage of deconvolution [3] in 3D domain in biomedical researches.

A. Fully Convolutional Network (FCN)

The convolutional networks had been well known for its ability to extract hierarchies of image features since early 2010s. Such attribute made CNNs naturally fit in the classification tasks, which was enabled by a sequential set of convolutional layers and pooling layers and a final fully connected layer, transforming pixel-based spatial information into class-based semantic information.

Prior to the research on FCN, the localisation tasks were typically done by fine tuning the models to specialise in finding bounding box coordinates and key points in the image [4]. While method of FCN removes the fully connected layer, but instead adds another convolutional layer that up-samples the matrix by reversing the previous convolution operation [5]. (Therefore, the matrix with semantic information could be later concatenated with the original more spatially informative matrix derived from the input images.) This technique, known as Devolution or Transposed Convolution became the very fundamental unit of the U-Net architecture.

B. 2D U-Net

As symmetry was introduced in the CNN based segmentation architecture, U-Net was invented in 2015 by Olaf Ronneberger from University of Freiburg, which yielded "very good performance" and bested other known models at the time in different segmentation tasks [6].

The U-Net uses the same idea as in FCN to apply transposed convolutional layers and concatenations, so as to restore the spatial information corresponding to the semantic classifications. But some essential modifications were made: a) the architecture was perfectly symmetric, which means each convolutional layer and pooling layer had a reciprocal convolution layer and up-sampling layer, b) skip connections between each of those layers were built accordingly, c) a bottleneck layer was added other than encoder and decoder, supposing that the redundant channel features could be transformed into the only useful information to the localisation task, d) both skip connection and up-sampling layers had a learnable weight, which kept changing throughout the iterations. All these improvements granted U-Net an elegant and effective implementation.

The original work in [6] used unpadded convolution, therefore border pixels lost and cropping was necessary, and even height and width of the input image were also required to achieve desirable pooling. However, these could be easily optimised by using zero-padding, shifting, mirroring and other common padding techniques [7].

C. 3D U-Net

3D U-Net uses 3-dimensional kernels to execute the convolution and devolution operations, while maintains the general architect of the 2D U-Net. This allowed the network to extract the features involving all the adjacent pixels as in practice, which achieved significantly greater IoU on both fully and partially automated setups, compared to the simple method where results were obtained by adding up the output of 2D U-Net on all slices individually [1].

Moreover, unlike its predecessor, the model in [1] also avoided of using bottleneck, as was suggested in [8]. Other data processing techniques, such as batch normalisation, dropout, augmentations, were used in experiments to boost the performance of the model, too. More variants of the 3D U-Net models will be introduced and discussed in the following sessions.

III. Applications

The design of U-Net models excellently satisfied the needs of privacy and accuracy in biomedical image processing tasks. Different optimisations on 3D U-Net were practiced to handle specialised tasks.

A. General Discussion

Deep learning models usually rely on well annotated data and large training data set, the recent advances of big data engineering and data mining provided remarkable support for the progress of such industry. However, the special focus on privacy and high proficiency required for annotation makes extra obstacles in building models in biomedical areas.

The U-Net architecture, although considered slightly heavy and tedious, could make it possible to accurately accomplish localisation tasks with small size of training data. The 2D version managed to achieve an average IoU of 85.4% by only being trained from 168 images, and outperformed by the other two powerful competitors, ENet and BoxENet [9]. By using pre-trained model, either upon the entire architecture or encoder only, could greatly accelerate the training process while obtain even better outcome, e.g. VGG [10] (but it might be outdated now).

Its primitive 3D version achieved 72.3% IoU with only 3 raw training samples [1] (but many slices and augmentations could it provide). The popular variations of the U-Net include: AttUNet, ResUNet, RAUNet, UNet++, SDUNet, DCUNet, UNext, LcmUNet and so on [11]. All of them could have potentials to be utilised in 3D context, with different advantages and trade-offs.

B. Case Study

Segmentation of pelvic lymph nodes on difusion-weighted images could be a convincing example of showing the advantages of 3D U-Net on automated segmentation tasks in biomedical. Despite the heterogeneity of objects' shapes and sizes, the model achieved a precision of 0.97 and a recall of 0.98 with a total 393 images as input data (training:validation:testing = 8:1:1), however, correct annotation by senior radiologists were considered as a crucial precondition to the success[12].

In an earlier attempt to identifying Covid-19 infection, a two-stage cascaded 3D U-Net was used to analyse lung CT volume [13]. In the method, the first U-Net with residual skip connections would extract lung parenchyma from the CT input, and after appropriate post-processing, the second U-Net would then segment the infected 3D volumes. The proposed algorithm yielded a specificity of 99.84% and sensitivity of 83.33% on a data set of 20 CT volumes.

In the recent research of Jakhongir Nodirov on brain tumor segmentation, Attention 3D U-Net and Multiple Skip Connections were implemented [14]. Attention was achieved by applying a query and score function using a pre-proposed attention gate, i.e., Attention Gates, so that the model would better focus on the highlighted areas. And by replacing the one-to-one parallel skip connections in the traditional symmetric U-Net architecture with one-to-many connections and up-samplings, the Multiple Skip Connections could be achieved. Both techniques improved the performance of the model, but slightly increased training time. MobileNetV2 backbone was also used to optimise memory usage and training efficiency.

IV. Development Tools

Most mainstream deep learning libraries natively support 3D convolution, transposed convolution and pooling operation, which makes it possible to build up and train a simple 3D U-Net from the scratch. The popular choices for includes TensorFlow [15,16,17] and PyTorch [18,19,20] for Python, TensorFlow C++ API [21] and Caffe [22] for C++ (but I didn't find any 3D U-Net friendly classes in Caffe documentation yet), and so on.

Furthermore, MONAI [23] has its in-built 3D U-Net class, under monai.networks.nets.unet module[24] and monai.networks.nets.basic_unet module [25].

Besides, open source codes of previous milestone experiments and applicable built networks could be found in GitHub. The PyTorch implementation of the simple 3D U-Net method in [1] can be found on [26], or on [27] for Tensorflow; Another implementation of the U-Net as well as Attention U-Net and other researches can be found on [28]. Diverse styled tutorials can be found in different websites and forums [29, 30], too.

REFERENCES

[1] Özgün Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and Olaf Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," Jun. 2016, doi: https://doi.org/10.48550/arxiv.1606.06650.

[2] F. Milletari et al., "Hough-CNN: Deep learning for segmentation of deep brain regions in MRI and ultrasound," Computer Vision and Image Understanding, vol. 164, pp. 92–102, Nov. 2017, doi: https://doi.org/10.1016/j.cviu.2017.04.002.

[3] D. Tran, Lubomir Bourdev, R. Fergus, L. Torresani, and Manohar Paluri, "Deep End2End Voxel2Voxel Prediction," Nov. 2015, doi: https://doi.org/10.48550/arxiv.1511.06681.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. "DeCAF: A deep convolutional activation feature for generic visual recognition," In ICML, 2014, https://doi.org/10.48550/arXiv.1310.1531.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), doi:10.1109/cvpr.2015.7298965.

[6] Olaf Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015, doi: https://doi.org/10.48550/arxiv.1505.04597.

[7] J. L. Rumberger et al., "How shift equivariance impacts metric learning for instance segmentation," 2016 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, doi:10.1109/iccv48922.2021.00704.

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, doi:10.1109/cvpr.2016.308.

[9] A. F. Karimov et al., "Comparison of UNet, ENet, and BoxENet for Segmentation of Mast Cells in Scans of Histological Slices," Oct. 2019, doi: https://doi.org/10.1109/sibircon48586.2019.8958121.

[10] Iglovikov Vladimir, and A. Shvets. "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation," 2018, doi:10.48550/arxiv.1801.05746.

[11] S. Zhang and Y. Niu, "LcmUNet: A Lightweight Network Combining CNN and MLP for Real-Time Medical Image Segmentation," Bioengineering, vol. 10, no. 6, pp. 712–712, Jun. 2023, doi: https://doi.org/10.3390/bioengineering10060712.

[12] X. Liu et al., "Development and validation of the 3D U-Net algorithm for segmentation of pelvic lymph nodes on diffusion-weighted images," BMC Medical Imaging, vol. 21, no. 1, Nov. 2021, doi: https://doi.org/10.1186/s12880-021-00703-3.

[13] A. A. L. and V. C. S. S., "Cascaded 3D UNet architecture for segmenting the COVID-19 infection from lung CT volume," Scientific Reports, vol. 12, no. 1, 2022, doi: https://doi.org/10.1038/s41598022-06931-z.

[14] J. Nodirov, A. B. Abdusalomov, and T. K. Whangbo, "Attention 3D U-Net with Multiple Skip Connections for Segmentation of Brain Tumor Images," Sensors, vol. 22, no. 17, p. 6501, Aug. 2022, doi: https://doi.org/10.3390/s22176501.

[15] https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv3D (accessed Aug 2023)

[16] https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv3DTranspose (accessed Aug 2023)

[17] https://www.tensorflow.org/api_docs/python/tf/keras/layers/MaxPooling3D (accessed Aug 2023)

[18] https://pytorch.org/docs/stable/generated/torch.nn.Conv3d.html (accessed Aug 2023)

[19] https://pytorch.org/docs/stable/generated/torch.nn.ConvTranspose3d.html#convtranspose3d (accessed Aug 2023)

[20] https://pytorch.org/docs/stable/generated/torch.nn.MaxPool3d.html (accessed Aug 2023)

[21] https://www.tensorflow.org/api_docs/cc (accessed Aug 2023)

[22] https://caffe.berkeleyvision.org/doxygen/annotated.html (accessed Aug 2023)

[23] https://github.com/Project-MONAI/MONAI/ (accessed Aug 2023)

[24] https://docs.monai.io/en/stable/_modules/monai/networks/nets/unet.html (accessed Aug 2023)

[25] https://docs.monai.io/en/stable/_modules/monai/networks/nets/basic_unet.html (accessed Aug 2023)

[26] https://github.com/AghdamAmir/3D-UNet (accessed Aug 2023)

[27] https://github.com/ellisdg/3DUnetCNN (accessed Aug 2023)

[28] https://github.com/Hsankesara/DeepResearch (accessed Aug 2023)

[29] Anonymous, "3D U-Net Segmentation," https://miccai-sb.github.io/materials/Hoang2019b.pdf (accessed Aug 2023)

[30] N. Adaloglou, "3D Medical image segmentation with transformers tutorial," AI Summer, https://theaisummer.com/medical-segmentation-transformers/ (accessed Aug 2023)