UNet++: A Nested UNet Architecture for Medical Image Segmentation (Reimplementation by Abhiroop Talasila)

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee,
Nima Tajbakhsh, Jianming Liang

Arizona State University
{zongweiz, mrahmans, ntajbakh, jianming.liang}@asu.edu

Abstract

The UNet architecture [1], introduced in 2015, gained widespread popularity after it outperformed the prior best method (a sliding-window convolutional network) and proved its effectiveness with small datasets. But due to issues with imaging modality, optimal depth, and feature fusion, the authors of the mentioned paper [2] attempted to develop a new architecture to improve the quality of segmentation output and overall performance of the model. PyTorch is used to implement the paper and the original code can be found at: https://github.com/abhirooptalasila/HAI-Assignments/

1. Introduction

Biomedical Image Segmentation is a core component of medical care currently stimulated by deep learning. The earlier sliding window approach was very time-consuming and computationally expensive. UNet was designed specifically for medical image segmentation and became the state-of-art for many segmentation datasets. The authors felt they could make improvements to the original UNet architecture to address the issues detailed below.

1.1. Why?

Despite UNet's success, they have two limitations: their optimal depth for a particular task is unknown and their skip connections impose an unnecessarily restrictive fusion scheme, forcing aggregation only at the same-scale feature maps of the encoder and decoder subnetworks. The authors of this paper propose a new architecture for semantic and instance segmentation, alleviating the unknown network depth with an efficient ensemble of UNet's of varying depths, which partially share an encoder and co-learn simultaneously using deep supervision with redesigned skip connections to aggregate features of varying semantic scales at the decoder sub-networks, leading to a highly flexible feature fusion scheme. The proposed architecture promises better segmentation quality and invariance to imaging modality.

1.2. Architecture

The ensemble architecture (UNet^{e}) outlined in Figure 1 benefits from knowledge sharing, because all UNet's within the ensemble partially share the same encoder even though they have their own decoders. However, this architecture still suffers from two drawbacks. First, the decoders are disconnected — deeper UNet's do not offer a supervision signal to the decoders of the shallower UNet's in the ensemble. Second, the common design of skip connections used in the UNet^{e} is unnecessarily restrictive, requiring the network to combine the decoder feature maps with only the same-scale feature maps from the encoder. While striking as a natural design, there is no guarantee that the same-scale feature maps are the best match for the feature fusion. The paper presents UNet+ (Figure 2) first, which is just an intermediate proposal of UNet++ (Figure 3) without the dense skip connections.

The final decoder nodes $X^{0,1}, X^{0,2}, X^{0,3}, X^{0,4}$ all output segmentation masks corresponding to the depth of the underlying encoder. Deep supervision for these models basically means the loss is obtained by first calculating the individual loss between all four outputs and the expected segmentation mask and then aggregated. This enables the loss to propagate to earlier decoder stages while performing a gradient step. The authors also designed a wide UNet with similar number of parameters to the suggested architecture. This is to ensure that the performance gain yielded by this architecture is not simply due to increased number of parameters. So finally we have 4 models - a normal UNet, a wide UNet,

UNet+ and UNet++ with the last two either using deep supervision or not.

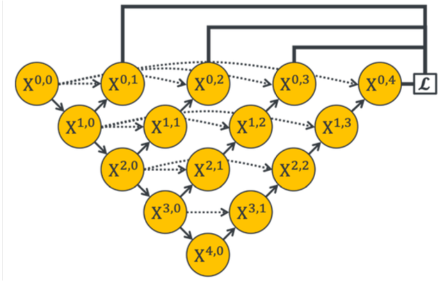

Figure 1. UNet^e

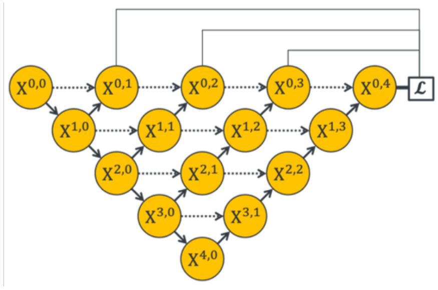

Figure 2. UNet+

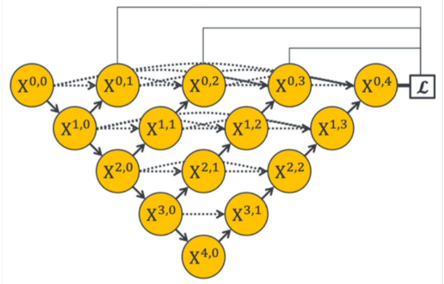

Figure 3. UNet++

1.3. Datasets

Application	Images	Input Size	Modality	Provider
EM	30	96×96	microscopy	ISBI 2012 [30]
Cell	354	96×96	Cell-CT	VisionGate [31]
Nuclei	670	96×96	mixed	Data Science Bowl
Brain Tumor	66,348	256×256	MRI	BraTS2013 [32]
Liver	331	96×96	CT	MICCAI 2017 LiTS
Lung Nodule	1,012	64×64×64	CT	LIDC-IDRI [33]

Table 1. Datasets used in the paper

Table 1 shows the 6 biomedical datasets used in this paper. The second dataset is not publicly available and the last one consists of 3D CT scans and all the others are processed as 2D images. I could test my implementation all of the 2D datasets but didn't have enough compute for the 3D dataset.

1. **Electron Microscope**: The dataset is provided by the EM segmentation challenge as a part of ISBI 2012. The dataset consists of 30 images (512×512 pixels) from serial section transmission electron microscopy of the Drosophila firt instar larva ventral nerve cord (VNC).

2. **Cell**: The dataset is acquired with a Cell-CT imaging system.

3. **Nuclei**: The dataset is provided by the Data Science Bowl 2018 segmentation challenge and consists of 670 seg- mented nuclei images from different modalities (brightfield vs. fluorescence). This is the only dataset used in this work with instance-level annotation where each nucleolus is marked in a different color.

4. **Brain Tumor**: The dataset is provided by BraTS 2013. The models are trained using 20 High-grade (HG) and 10 Low-grade (LG) with Flair, T1, T1c, and T2 scans of MR images from all patients. The ground truth segmentation have four different labels: necrosis, edema, non-enhancing tumor, and enhancing tumor. The paper considered all four labels as positive class and others as negative class.

5. **Liver**: The dataset is provided by MICCAI 2017 LiTS Challenge and consists of 331 CT scans. The ground truth segmentation provides two different labels: liver and lesion. The paper considered liver as positive class and others as negative class.

2. Methodology

To implement the architecture, I first built a basic UNet and then modified the decoder networks because that is where the redesigned skip connections come into picture. The implementation was fairly straightforward. The paper presents UNet+ first, which is just an intermediate proposal of UNet++ without the dense skip connections. The authors also designed a wide UNet with similar number of parameters to the suggested architecture. This is to ensure that the performance gain yielded by this architecture is not simply due to increased number of parameters. So finally we have 3 models - a normal UNet, wide UNet, UNet+ and UNet++ with the last two either using deep supervision or not. I used PyTorch's subclass `nn.Module` and standard `Dataset()` and `Dataloader()` classes to implement these. I also used an external library called Albumentations which makes it easier to apply the same augmentations to both images and masks in the dataset.

All the model architectures are given in `unet/models.py`, all dataset loaders are defined in `unet/datasets.py`, the training functions are given in `run.py` and the final results and visualisations are given in `unet/notebooks/training.ipynb`.

	DS	From Paper				My Implementation			
Dataset		**ISBI 2012**	**DS Bowl 2018**	**BraTS 2013**	**LiTS 2017**	**ISBI 2012**	**DS Bowl 2018**	**BraTS 2013**	**LiTS 2017**
UNet	✗	88.30	90.57	89.21	79.90	88.32	89.34	90.62	80.11
Wide UNet	✗	88.37	90.47	89.35	80.25	88.73	89.64	90.94	80.24
UNet+	✗	88.39	91.73	90.70	79.62	88.52	90.56	91.34	79.84
UNet+	✓	88.89	92.04	91.15	**82.83**	88.76	90.93	91.76	**83.02**
UNet++	✗	88.92	**92.44**	90.86	82.51	89.28	91.62	91.45	82.45
UNet++	✓	**89.33**	92.37	**91.21**	82.60	**89.55**	**91.80**	**92.07**	82.62

Table 2. Semantic Segmentation results measured by IoU%

2.1. Issues Faced

The main issue was processing the 3D datasets like BraTS and LIDC-IDRI. The former has MRI scans in `.mha` files, which I was unfamiliar with. It took some time to understand the different types of scans and the `SimpleITK` library. I could process the BraTS dataset but didn't have the compute to process the lung nodule dataset. Most implementations of UNet use `UpSampling` layers to implement the decoder network. But after a bit of research, I found out that `ConvTranspose` layers made more sense as `UpSampling` is just simple interpolation whereas `ConvTranspose` is a convolution layer with trainable parameters.

2.2. Results

The final results given in Table 2 are for semantic segmentation and Table 3 has results for instance segmentation using the DS Bowl dataset with a `ResNet101` backbone for UNet and UNet++. Instance segmentation consists of segmenting and distinguishing all object instances; hence, is more challenging than semantic segmentation. The Nuclei dataset is chosen because multiple nucleolus instances can be present in an image, in which case each instance is annotated in a different colour, and thus marked as a distinct object. Therefore, this dataset is amenable to both semantic segmentation where all nuclei instances are treated as foreground class, and also instance segmentation where each individual nucleus is to be segmented separately. As expected, UNet++ outperforms UNet for semantic segmentation too.

UNet++ is at the top of the leaderboard for the 2018 DS Bowl dataset with an IoU of 92.5% compared to my implementation with an IoU of 91.8%. Some sample output masks from all the models are given in Figure 4 for comparison.

2.3. Final Thoughts

As I mentioned earlier, my initial reason for picking up this paper was that it showcased a much better quality segmentation result than from a normal UNet.

	From Paper		My Implementation	
Architecture	**IoU**	**Dice**	**IoU**	**Dice**
UNet	91.03	75.73	91.47	75.62
UNet++	**92.55**	**89.74**	**92.64**	**89.77**

Table 3. Instance Segmentation Results

From the segmentation results in Figure 4 we can see that the results from the proposed architecture are much more structurally clear. This improved performance by UNet++ is attributed to its nested structure and redesigned skip connections, which aimed to address two key challenges of the UNet: 1) unknown depth of the optimal architecture and 2) the unnecessarily restrictive design of skip connections. From the results, we can see that UNet++ has demonstrated consistent performance improvement for both semantic and instance segmentation.

References

[1] Olaf Ronneberger, Philipp Fischer, Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. MICCAI, 2015. 1

[2] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, Jianming Liang. *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. IEEE Transactions on Medical Imaging, 2019. 1

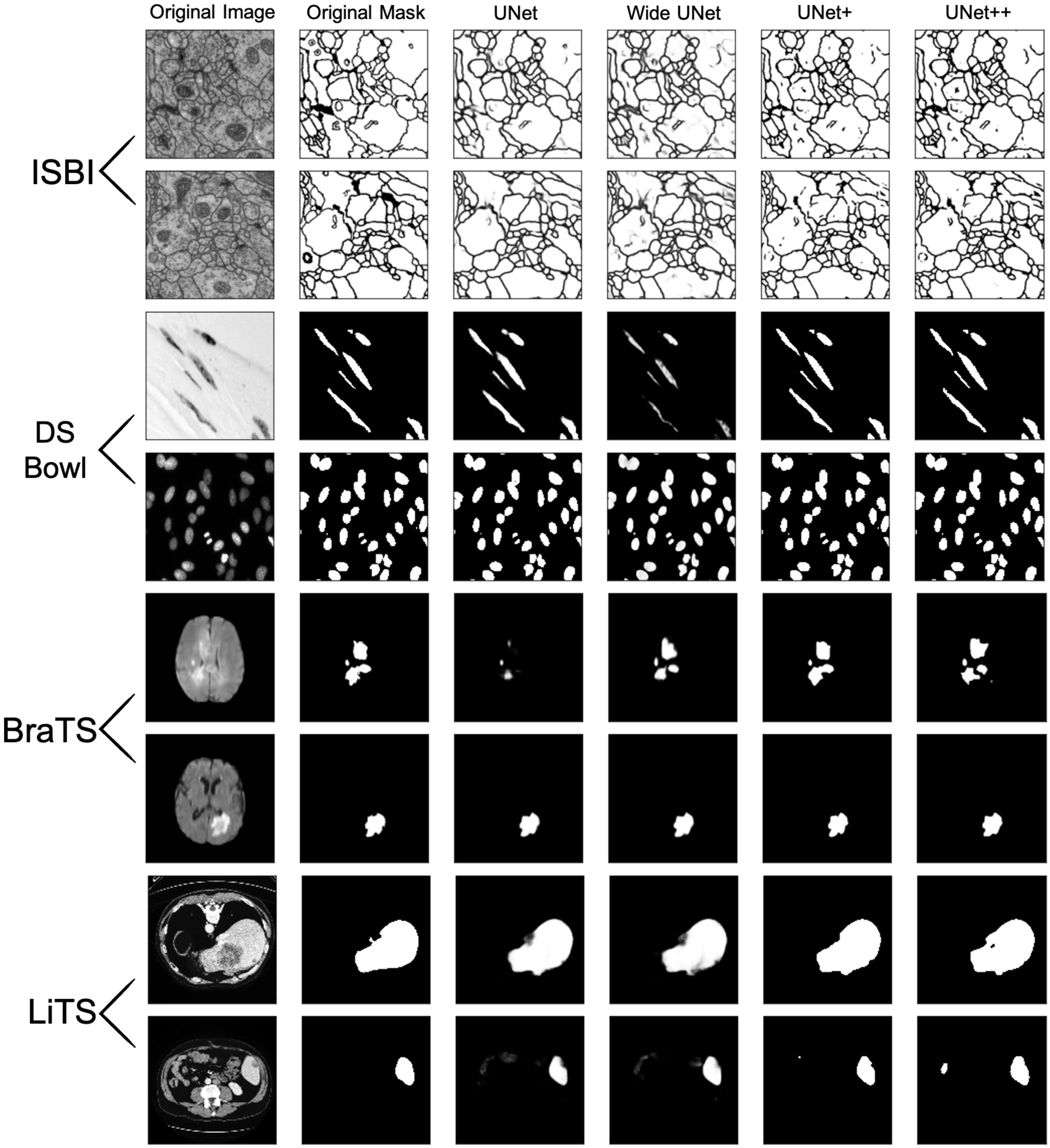

Figure 4. Segmentation Outputs