U-Net++: Nested U-Net Architecture for Bio-Medical Image Segmentation

Contents

List of Figures

UNet++ Introduction

UNet++, a convolutional neural network dedicated for biomedical image segmentation, was designed, and applied in 2018 by (Zhou et al., 2018). UNet++ was basically designed to overcome some of the short comings of the UNet architecture. UNet works on the idea of skip connections. U-Net concatenates them and add convolutions and non-linearities between each up-sampling block. The skip connections recover the full spatial resolution at the network output, making fully convolutional methods suitable for semantic segmentation. UNet and other segmentation models based on the encoder-decoder architecture tend to fuse semantically dissimilar feature maps from the encoder and decoder sub-networks, which may degrade segmentation performance. This is where UNet++ is shown to have an edge over the other players as it bridges the semantic gap between the feature maps of the encoder and decoder prior to fusion thus improving the segmentation performance and output.

Evolution of UNet++

UNet and FCNs have attained the state-of-the-art status in the field of medical image segmentation. The encoder-decoder structure are widely used in almost every semantic and instance segmentation task. Their success is largely attributed to the design of the *skip connections that combine the deep, semantic, coarse-grained feature maps from the decoder sub-network with shallow, low-level, fine-grained feature maps from the encoder sub-network*. However, the network structure and the design of the skip connections suffer from the following limitations.

1. The **network depth** could vary from task to task largely attributed to the amount of the data available and the complexity of the segmentation task.
2. The **design of skip connections employed is very much restrictive**, such that it expects encoder and decoder feature maps to be fused be at the same scale.

The evolution goes through 3 different architectural phases with each phase improving he limitations of the previous one. The three different phases are -

1. Ensemble UNets
2. UNet+
3. UNet++

Ensemble UNets

The authors in the UNet paper (Zhou et al., 2019), experiment with UNets of varying depths and observe that deeper UNets does not always tend to increase the segmentation performance and the depth of the network also depends on the amount of data available and the complexity of the task. To overcome these limitations, the authors proposed an ensemble UNet combining UNets of varying depths. The ensemble architecture is represented as **UNet^{e}**. The individual UNets in the ensemble structure (partially) share a common encoder path while having their own respective decoders. Each UNet is trained with a different loss function to enable the ***deep supervision***. The output from respective UNets is averaged at the inference time. this kind of an ensemble structure provides an increase in performance, it still has a few drawbacks.
The ***decoders are disconnected*** such that the deeper UNets do not offer any supervision to their counterparts in the shallower UNets. The ***design of skip connections is still very much restrictive*** and expects the encoder and decoder feature maps to be on the same scale.

UNet+

To eliminate the above limitations of UNet^{e}, the authors modify the skip connections where they remove the original one to one skip connections and ***connect every two adjacent nodes in the ensemble***. This enables the UNet+ structure to connect feature maps which are at different scales enabling gradient back-propagation from the deeper decoders to the shallower counterparts. UNet+ also eliminates the restrictive behavior of skip connections by presenting each node in the decoders with the aggregation of all the feature maps flowing from the shallower UNets.

UNet++

To further improve the performance of UNet+ network, the structure of the skip connections is modified in way that enables ***dense connectivity along the skip pathways***. Each node in the decoder sees the original same-scale feature maps from the encoder, the intermediate aggregated feature maps, and the final aggregated feature maps. This enables the network to learn more detailed features from multiple aggregated feature maps.
UNet++ employs *deep supervision*, enabling the model to operate in two modes: **1) *accurate mode*** wherein the outputs from all segmentation branches are averaged; **2) *fast mode*** wherein

the final segmentation map is selected from only one of the segmentation branches, the choice of which determines the extent of model pruning and speed gain. However, deep supervision is not mandatory for both UNet+ and UNet++.

Figure 1 shows the evolution of UNet to UNet++.

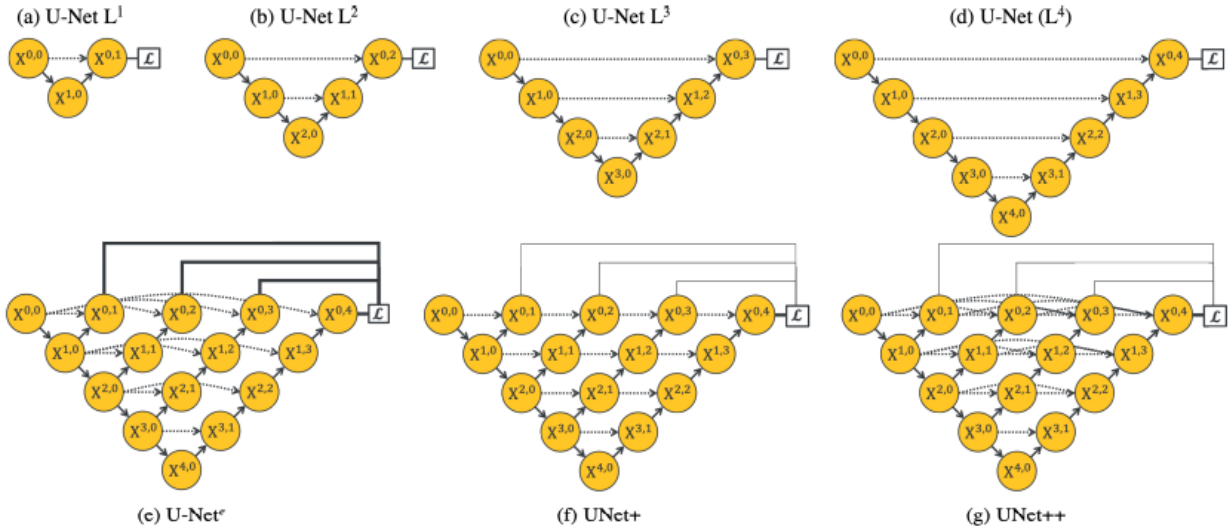

Figure 1 - Evolution from U-Net to UNet++

Image Source - (Zhou et al., 2019)

figures - (a - d) represents UNets of varying depths, figure - (e) represents the ensemble architecture, figure - (f) represents the UNet+ variant, figure - (g) represents the final UNet++ variant.

UNet++ Architecture

"UNet++ is constructed from U-Net^e by connecting the decoders, resulting in densely connected skip connections, enabling dense feature propagation along skip connections and thus more flexible feature fusion at the decoder nodes. As a result, each node in the UNet++ decoders, from a horizontal perspective, combines multiscale features from its all preceding nodes at the same resolution, and from a vertical perspective, integrates multiscale features across different resolutions from its preceding node. This multiscale feature aggregation of UNet++ gradually synthesizes the segmentation, leading to increased accuracy and faster convergence."

(Zhou et al., 2019)

Network Connectivity (Re-designed Skip Connections)

UNet++ consists of an encoder and decoder that are connected through a series of nested dense convolutional blocks. The type of connectivity between the encoder and decoder pathway is determined by the equation shown in Figure 2.

$$x^{i,j} = \begin{cases} \mathcal{H}\left(\mathcal{D}(x^{i-1,j})\right), & j = 0 \\ \mathcal{H}\left(\left[\left[x^{i,k}\right]_{k=0}^{j-1}, \mathcal{U}(x^{i+1,j-1})\right]\right), & j > 0 \end{cases}$$

Figure 2 - Network Connectivity Formulation

Image Source - (Zhou et al., 2019)

$\mathcal{H}(\cdot)$ is a convolution operation followed by an activation function, $\mathcal{D}(\cdot)$ and $\mathcal{U}(\cdot)$ denote a down-sampling layer and an up-sampling layer respectively, and [ ] denotes the concatenation layer.

The above equation implies that when j = 0, which is along the encoder pathway involves only down sampling operations and receives input only from the previous layer. However, when j > 0 which is along the skip pathway a node receives multiple inputs - from the previous node in the same skip level and the up-sampled output from the $(j + 1)^{th}$ node at the lower (skip) level.

The use of dense block along each skip connection enables to accumulate and propagate all the previous feature maps to the current node.

Deep Supervision

Deep supervision is a technique to generate multiple segmentation maps each at a different resolution level. The feature maps at each level are transposed/up-sampled to create secondary segmentation maps which are then combined and concatenated. *The secondary segmentation maps help in the speed of convergence by "encouraging" earlier layers of the network to produce better segmentation results* (Turečková et al., 2020).

UNet++ implements deep supervision via 1x1 convolutions with C (no. of classes) kernels followed by sigmoid activation to the outputs from the nodes - $X^{0,1}$, $X^{0,2}$, $X^{0,3}$, and $X^{0,4}$ along the top level skip pathway.

Deep supervision enables the model to operate in 2 different modes - (Zhou et al., 2018)

1. **Accurate Mode**: In this mode, the outputs from all segmentation branches are combined and averaged to get the final output.

2. **Fast Mode**: In this mode, the final segmentation map is selected from only one of the segmentation branches, the choice of which determines the extent of model pruning and speed gain.

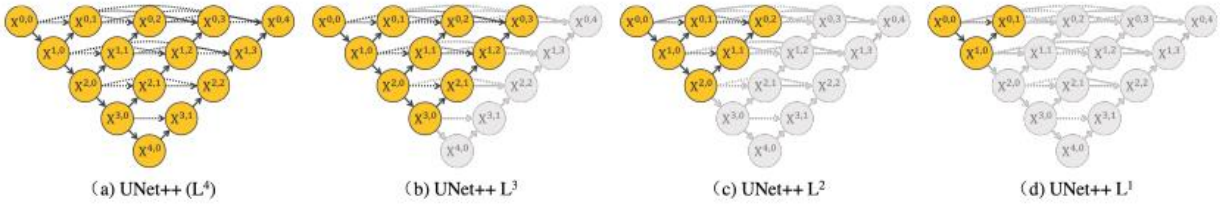

Figure 3 - Deep Supervision

Image Source - (Zhou et al., 2019)

The different UNets seen in Figure 3 communicate through the nested skip pathways and generate the full resolution feature maps at each levels which corresponds to deep supervision. To get the best output at each semantic levels, a combination of ***binary cross-entropy*** and ***dice coefficient*** is chosen as the loss function at each level. The hybrid loss function takes advantages of both loss functions resulting in a ***smooth gradient*** and ***efficient handling of class imbalance***. The hybrid loss function is defined as -

$$\mathcal{L}(Y, \hat{Y}) = -\frac{1}{N}\sum_{b=1}^{N}\left(\frac{1}{2}\cdot Y_b \cdot \log \hat{Y}_b + \frac{2\cdot Y_b \cdot \hat{Y}_b}{Y_b + \hat{Y}_b}\right)$$

Figure 4 - Hybrid Loss

where,

$\hat{Y}_b$ denotes the flatten predicted probabilities of b^{th} image.

Y_b denotes the flatten ground truths of b^{th} image respectively, and

N indicates the batch size

Model Pruning

Deep supervision paves the way for model pruning. It allows the model to be deployed in two different operating modes.

1. **Ensemble Mode (Accurate Mode)**: In this mode, the segmentation results from all segmentation branches are collected and then averaged to get the final output.

2. **Pruned Mode (Fast Mode)**: In this mode, the segmentation output is selected from only one of the segmentation branches, the choice of which determines the extent of model pruning and speed gain.

Architecture Diagram

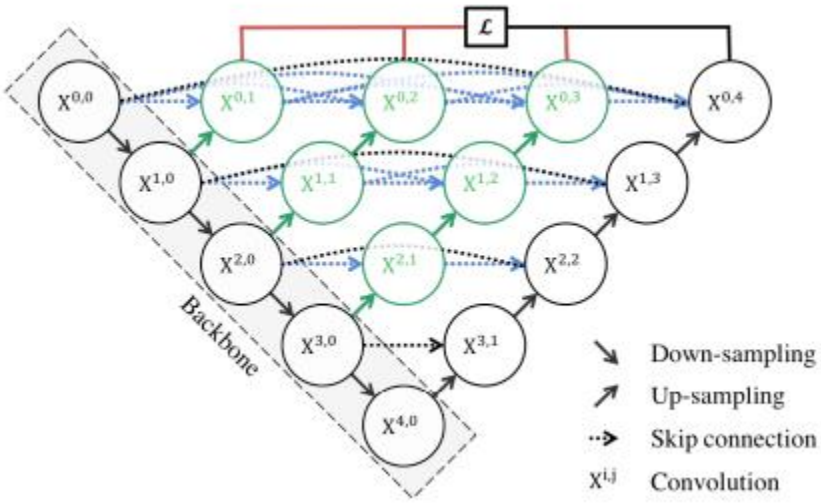

Figure 5 - UNet++ Architecture

Skip connections between any two different nodes is depicted as -

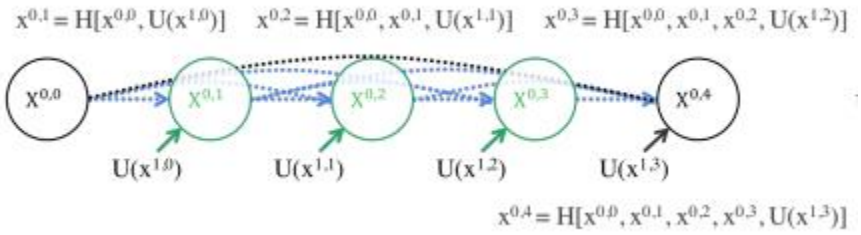

Figure 6 - Skip Pathway

The skip pathway between nodes $\mathbf{X^{0,0}}$ and $\mathbf{X^{1,3}}$ consists of a series dense convolution block with three convolution layers where **each convolution layer is preceded by a concatenation layer that fuses the output from the previous convolution layer of the same dense block with the corresponding up-sampled output of the lower dense block**. In Figure 6, the corresponding unit in the decoder block $X^{0,4}$ is a combination of the outputs of convolutions - X^{00}, X^{01}, X^{02}, X^{03} and up sampled convolution X^{13}.

Image Source - (Zhou et al., 2018)

UNet++ Metrics

UNet++ uses a combination of different metrics and loss functions like the *Binary Cross-Entropy*, *Dice Coefficient*, and the *Intersection over Union (IoU)* coefficient.

Binary cross-entropy

A loss function for binary classification for measuring the probability of misclassification. Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1 and calculates the score that penalizes the probabilities based on the distance from the expected value. UNet++ computes the ***pixelwise cross-entropy*** loss.

Dice Coefficient

It measures the overlap between the predicted and the ground truth.
It is computed as **- 2 * the area of overlap (*between the predicted and the ground truth*) divided by the total area (*of both predict and ground truth combined*).** It ranges between 0 and 1 where a 1 denotes perfect and complete overlap.

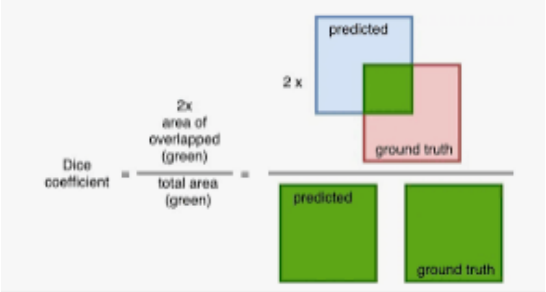

Figure 7 - Dice Coefficient

The model uses a combination of ***binary cross-entropy*** and ***dice coefficient*** is chosen as the loss function at each level. The hybrid loss function takes advantages of both loss functions resulting in a ***smooth gradient*** and ***efficient handling of class imbalance***. Figure 4 shows the mathematical representation of the hybrid loss function employed to train the model.

Intersection over Union (IoU) Coefficient

IoU tells how accurate the predicted mask is with the ground truth mask. It is computed as the area of overlap (*between the predicted and the ground truth*) and divide by the area of the union (*of predicted and ground truth*).

It ranges between 0 to 1 where 0 signifying no overlap whereas 1 signifying perfectly overlapping between predicted and the ground truth.

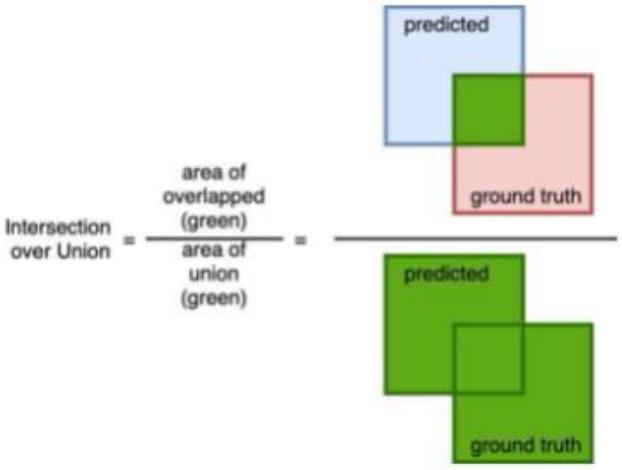

Figure 8 - Intersection over Union

Model Training and Results

UNet++ model is designed and put to training for a brain tumor segmentation task. The network parameters are chosen as per the implementation in the original paper (Zhou et al., 2018). The model is trained over 30 epochs on brain tumor data available at (*Brain Tumor Dataset*, n.d.). The dataset consists of 3064 brain tumor images along with their masks. For training purpose, the data is divided into training, validation and tests sets each having 2800, 200 and 64 images respectively.

Training Parameters

The network is trained with the below parameters set.

Epochs: 30

Batch Size: 64

Image Size: (128, 128)

Loss Function and Metric: Combination of Binary Cross Entropy and Dice Coefficient, IoU coefficient.

For this task, early stopping was not considered.

Segmentation Results

The image below shows the segmentation results from some of the images from the test set.

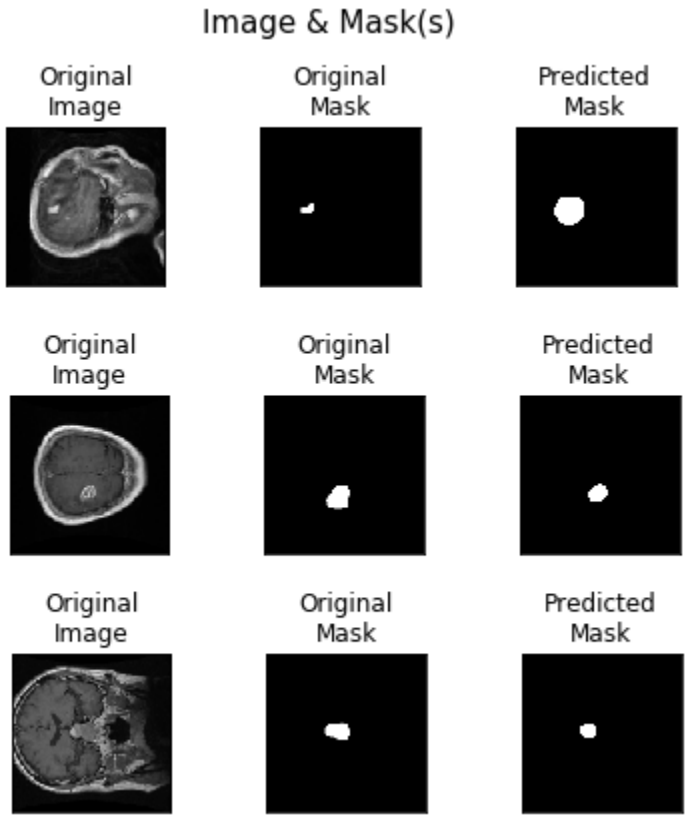

Figure 9 - Segmentation Results

Comparing the original image, original mask and the predicted mask, the model based on the UNet++ architecture is correctly able to segment the brain tumor location and generate the masks. Though there are some differences seen in the visualizations above, these can be improved with further training and fine tuning the model itself.

Implementation details and code is uploaded @

1. https://github.com/sauravmishra1710/UNet-Plus-Plus---Brain-Tumor-Segmentation/blob/main/BrainTumorSegmentation.ipynb

2. https://github.com/sauravmishra1710/UNet-Plus-Plus---Brain-Tumor-Segmentation/blob/main/UNetPlusPlus%20-%20Nested%20UNet.ipynb

Conclusion and Summary

- UNet++ aims to improve segmentation accuracy, with a series of nested, dense skip pathways.
- Redesigned skip pathways make optimization easier by getting the semantically similar feature maps.
- Dense skip connections improve segmentation accuracy and make the gradient flow smoother.
- Deep supervision allows for model complexity tuning to balance between speed and performance optimization by allowing the model to toggle between 2 different training modes in the fast mode and the accurate mode.
- UNet++ differs from the original U-Net in three ways - (refer Figure 5)
    - It has convolution layers (green)on skip pathways, which bridges the semantic gap between encoder and decoder feature maps.
    - It has dense skip connections on skip pathways (blue), which improves gradient flow.
    - It employs deep supervision (red), which enables model pruning and improves or in the worst case achieves comparable performance to using only one loss layer.

References

1. *brain tumor dataset*. (n.d.). Retrieved July 5, 2021, from https://figshare.com/articles/dataset/brain_tumor_dataset/1512427
2. Turečková, A., Tureček, T., Komínková Oplatková, Z., & Rodríguez-Sánchez, A. (2020). Improving CT Image Tumor Segmentation Through Deep Supervision and Attentional Gates. *Frontiers in Robotics and AI*, *7*, 106. https://doi.org/10.3389/frobt.2020.00106
3. *UNet++: A Nested U-Net Architecture for Medical Image Segmentation | Papers With Code*. (n.d.). Retrieved July 5, 2021, from https://paperswithcode.com/paper/unet-a-nested-u-net-architecture-for-medical
4. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. https://arxiv.org/abs/1807.10165
5. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2019). *UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation*. http://arxiv.org/abs/1912.05074