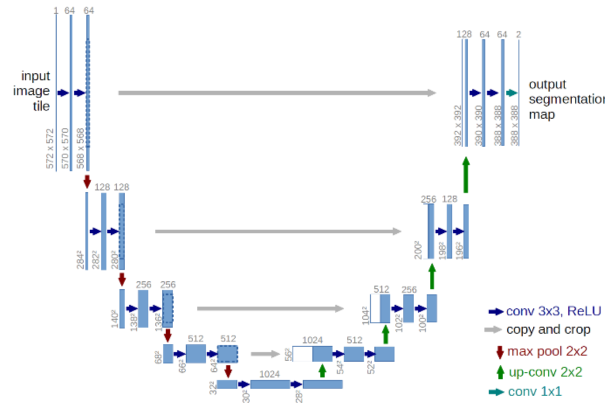
Preface:

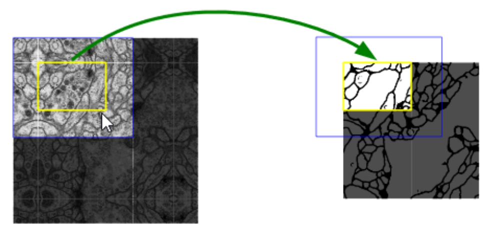


U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

U-Net: Convolutional Networks for Biomedical Image Segmentation

https://arxiv.org/pdf/1505.04597.pdf

Network Architecture (mentioned precisely due to 15k+ citations):



- In this paper, the authors build upon the “fully convolutional network” (UNET meets this classification); however they had to modify and extend this architecture such that it works with very few training images and yields more precise segmentations;
    - This is done via ‘Overlap-tile’ strategy: Since unpadded convolution is used, output size is smaller than input size. Instead of downsizing before network and up sampling after network, **overlap tile strategy** is used.
    - Thereby, the whole image is predicted part by part as in the figure above. The yellow area in the image is predicted using the blue area. At the image boundary, image is extrapolated by mirroring.

BU-Net: Brain Tumor Segmentation Using Modified U-Net Architecture

Relevant summary:

- This paper proposes a 2D image segmentation method, BU-Net, to contribute to brain tumor segmentation research. Residual extended skip (RES) and wide context (WC) are used along with the customized loss function in the baseline U-Net architecture. The modifications contribute by finding more diverse features, by increasing the valid receptive field. The contextual information is extracted with the aggregating features to get better segmentation performance. The proposed BU-Net was evaluated on the high-grade glioma (HGG) datasets of the BraTS2017 Challenge. This paper also exercises dice score.

Dataset:

- The publicly available benchmark databases used in this study. The proposed BU-Net model was evaluated on two benchmark datasets. These datasets are BraTS 2017 and BraTS 2018. The BraTS 2017 dataset consists of images collected from 285 glioma patients, out of which 210 were HGG cases and the remainder belong to LGG cases. Further, the validation dataset of BraTS 2017 carries images of 46 patients with unknown grade. The ground truths of the training data were labeled by the experts, and the labels of validation dataset are not made publicly available; therefore, the results can only be generated from the online web-server of BraTS. The dataset is labeled as four main classes which are:
    - Enhancing tumor.
    - Necrosis and non-enhancing tumor.
    - Edema.
    - Healthy tissue

Preprocessing:

- One of the weaknesses of deep learning models is that they are robust to noise; therefore, data processing is an important task to be carried out before the image is given to the network. For this purpose, N4ITK algorithm [34], a bias correction technique, is used on all images to make them homogeneous

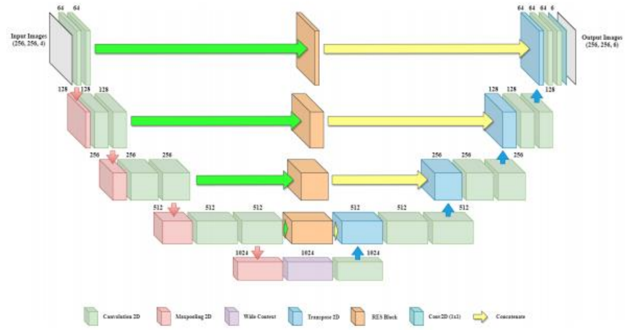
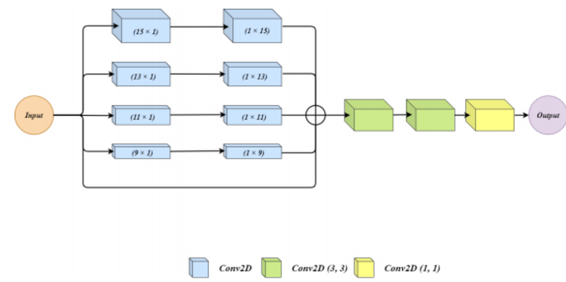


Figure 2. Overall architecture of the proposed BU-Net including RES blocks and wide context block.

Network architecture:

- In previous baseline architecture, no contextual information is shared between the shallow and deep layers. There is a need to introduce a module which can create an information bridge between shallow and deep layers so that local and global features of the network can be enhanced. Figure 2shows the overall architecture of the proposed BU-Net, which includes RES blocks and a WC block
  - The architecture takes input images of resolution 256 × 256 and outputs the images with the same dimensions. The left part of the model act as an encoder and the right part of the model acts as a decoder. The convolution layers with padding are used in BU-Net. This allows getting the same sized image as the output as that given as input.
  - The encoder and decoder of the network are divided into blocks. On the encoder side every block consists of two convolution layers along with a single max-pooling layer and a dropout layer. Every block of the decoder side starts with the Conv2DTranspose layer applied on the output of the previous block.
  - The output of Conv2DTranspose layer is concatenated with the output generated from the associated RES block. Dropout is applied to the concatenated output followed by two convolution layers. The last block of the decoder includes another convolution layer with six filters of size 1 × 1. The encoder side performs the contraction process on the image.



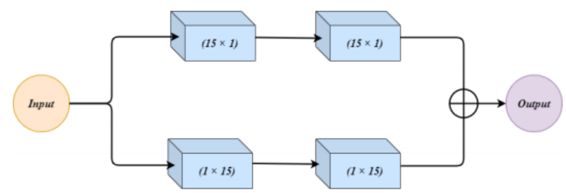


Figure 4. The architecture for the Wide Context (WC) block.

  - All convolution layers of BU-Net are followed by batch normalization and ReLU activation function, except for the last convolution layer which utilizes sigmoid

    - **Hyperparameters:**

      To set the dropout ratio, they applied hyper-parameter tuning—a range of dropout ratios were tested to get the most optimal dropout ratio; 0.3 proved to be the most optimal dropout ratio for the network. Adam optimizer was used along with the customized loss function. The learning rate was set to 0.01 with a momentum of 0.9. The batch size was 16, and early stopping based on validation loss with patience level of 10 was utilized for the maximum number of training iterations.
    - **Residual Extended Skip (RES):**

      The input to the architecture is given to 5 parallel connections. In the first four of them, two convolutions layers are applied. In each connection with convolution layers, we have used N × 1 filter size for first convolution layer and 1 × N filter size for second convolution layer. We used two cascaded convolution layers rather than using a single convolution layer with the filter size of N × N. Using two convolution layers generates a lesser number of parameters which benefits the overall architecture. The RES block generates the middle level features from low level features which helps to control information degradation.
    - **Wide Context (WC):**

      The input to WC is given to two parallel connections. Both the connections have 2 convolution layers. In the first connection, the two convolution layers use N × 1 and 1 × N respectively. The second connection first uses the 1 × N filter size, and then the next convolution layer has filter size N × 1. This change in combination in both the connections makes up a good feature set which can contribute towards the performance. The observation was made that a change in combination changes the extracted features, and both the combinations can contribute towards the final result. The outputs from both connections are summed up and treated as an output of WC.
    - **Customized-Loss function:**

BU-Net utilizes a combined loss function that sums weight cross-entropy (WCE) and Dice loss coefficient (DLC) as one of the challenges with brain tumor segmentation is relatd to the imbalance of class data from Brats dataset.

Table 2. Comparison of results with the BraTS 2017 HGG data.

Architecture	Whole	Core	Enhancing
CNN [38]	0.840	0.720	0.620
U-Net [39]	0.831	0.801	0.750
Densely CNN [40]	0.720	0.830	0.810
ResU-Net [41]	0.88	0.850	0.750
FCNN [35]	0.865	0.864	0.816
Proposed BU-Net	0.901	0.867	0.835

Table 3. Comparison of results with the whole BraTS 2017 dataset (57 MRI scans).

Architecture	Whole	Core	Enhancing
Seg-Net [42]	0.833	0.703	0.496
U-Net [39]	0.870	0.762	0.700
ResU-Net [41]	0.873	0.768	0.716
PSPNet [43]	0.809	0.701	0.554
NovelNet [43]	0.876	0.763	0.642
Proposed BU-Net	0.892	0.783	0.736

Table 4. Comparison of results with the BraTS 2018 validation dataset. (66 MRI scans).

Architecture	Whole	Core	Enhancing
U-Net [39]	0.860	0.790	0.767
3DU-Net [44]	0.885	0.718	0.760
ResU-Net [41]	0.867	0.803	0.768
Ensemble Net [45]	0.881	0.777	0.773
TTA [46]	0.873	0.783	0.754
S3DU-Net [47]	0.894	0.831	0.749
MCC [48]	0.882	0.748	0.718
Proposed BU-Net	0.901	0.837	0.788

A Modified U-Net Convolutional Network Featuring a Nearest-neighbor Re-sampling-based Elastic-Transformation for Brain Tissue Characterization and Segmentation

Summary

- The goal of this work was to improve the U-net model by replacing the de-convolution component with an up-sampled by the Nearest-neighbor algorithm and also employing an elastic transformation to augment the training dataset to render the model more robust, especially for the segmentation of low-grade tumors.

Dataset:

- All experiments reported in this paper were conducted on the BRATS 2017 image database. In addition, for testing purposes, they also used images from 146 patients featuring brain tumors of unknown grade available from the same MICCAI 2017 Challenge on Multi-modal Brain Tumor Segmentation.

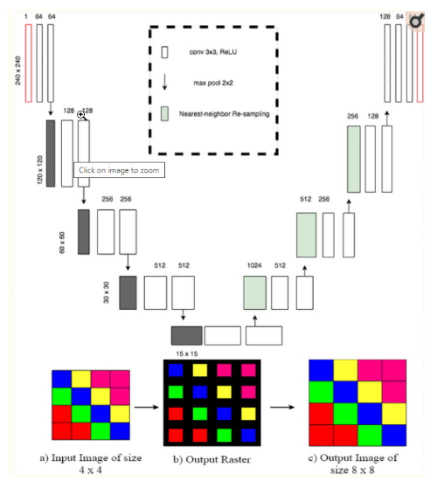


Fig. 4.

Schematic diagram of proposed NNRET U-net architecture (the concatenation process is not shown in the image) and nearest neighbor up-sampling for image enlargement.

Network Architecture:

- The image shows a schematic diagram of our proposed NNRET U-net deep convolution neural network. We replaced the fully connected layers with a down-sampling layer along with max-pooling layers. Every two layers form a block and the encoding path consists of 5 convolutional blocks. The network is to a large extent similar to the approach presented in [6]. The convolution layers in all the blocks detect the local features from the previous layers and map their appearance to a feature map. The kernel size and the stride were both chosen such that the filter size can be divided by the stride. For a kernel size of 3×3 and a stride of 1 in both directions coupled with a ReLU activation function, the stride will move the filters one pixel at a time. This down-sampling path decreases the feature size while increasing the number of feature maps from 1 to 1024.
- To decrease computational cost, we reduced the input volume size from 240 × 240 into 15 × 15, further down-sampled via a 2 × 2 × then max-pooling operation (i.e., inserting pooling layer in-between successive convolutional layers), resulting in low-resolution feature maps. To decrease the spatial size of the image, we used the max pooling of size 2 × 2 with a stride of 2 that will move the filter 2 pixels at a time, resulting in 75% fewer activations. To avoid the cropping operation during the concatenation of feature maps, we used zero padding on every convolution layer.

- Used nearest neighbor up-sampling layer with the scale factor of 2 at the beginning of each block, followed by two convolution layers and ReLU function that increased the spatial dimension in each block by a factor of 2
- The nearest neighbor up-sampling works like convolution that performs a mathematical operation on every pixel and its neighbors by using interpolation to increase the spatial dimension of an image. The process being the nearest neighbor up-sampling to increase image resolution is shown in Figure 4. As an example, here we explain how a 4 × 4 pixel image would be up-sampled by NN interpolation method. The cell centers of the output raster are equally separated and a location value needs to be determined from the input raster for each output cell. The nearest neighbor algorithm selects select those cells centers from the input raster that are closest to that of output raster. The black areas of the image can be filled either with the copies of the center pixel or the weighted combinations of the surrounding pixels.

Training accuracy after the first three epochs: every epoch consists of 6 sub-steps and upon the completion of each epoch, the mean DSC, IoU and computing time are computed. The training accuracy increased to 91% at convergence after the completion of 15 epochs.

Epoch	Dice	IoU	Time (s)
1	0.0433	0.5588	10.600
1	0.1918	0.6223	10.442
1	0.1388	0.7349	10.385
1	0.0961	0.5256	10.463
1	0.1606	0.8142	10.513
1	0.8084	0.7338	10.385
1/100	0.1937	0.6931	6817.6
2	0.7219	0.5447	10.516
2	0.4983	0.8910	10.516
2	0.3646	0.7633	10.462
2	0.5008	0.7498	10.499
2	0.9915	0.7739	10.709
2	0.4068	0.0000	10.519
2/100	0.5629	0.7162	6860.7
3	0.5019	0.6125	10.471
3	0.3489	0.8541	10.543
3	0.8318	0.9049	10.535
3	0.5502	1.0000	10.574
3	0.6907	0.7348	10.721
3	0.3784	0.7411	10.511
3/100	0.5532	0.7342	6880.5
…	…	…	…
100/100	0.9104	0.9075	6834.3

Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks

Data acquisition:

- The proposed method was tested and evaluated on the BRATS 2015 datasets, which contain 220 high-grade glioma (HGG) and 54 low-grade glioma (LGG) patient scans. Multimodal MRI data is available for every patient in the BRATS 2015 datasets and four MRI scanning sequences were performed for each patient using T1-weighted (T1), T1-weighted imaging with gadolinium enhancing contrast (T1c), T2-weighted (T2) and FLAIR.

Data Augmentation:

- Simple transformation such as flipping, rotation, shift and zoom can result in displacement fields to images but will not create training samples with very different shapes. Shear operation can slightly distort the global shape of tumor in the horizontal direction, but is still not powerful to gain sufficient variable training data, as tumors have no definite shape. To cope with this problem, we further applied elastic distortion that can generate more training data with arbitrary but reasonable shapes.

Table 1. Summary of the applied data augmentation methods (γ controls the brightness of the outputs; α and σ control the degree of the elastic distortion).

Methods	Range
Flip horizontally	50% probability
Flip vertically	50% probability
Rotation	$\pm 20^{o}$
Shift	10% on both horizontal and vertical direction
Shear	20% on horizontal direction
Zoom	±10%
Brightness	γ=0.8~1.2
Elastic distortion	α=720, σ=24,

Hyperparameters:

- Soft Dice metric was used as the cost function of the network rather than the cross-entropy based or the quadratic cost function
- Adopted the adaptive moment estimator (Adam) to estimate the parameters
- The parameters of our Adam optimizer were set as: learning rate = 0.0001 and the maximum number of epochs = 100. All weights were initialized by normal distribution with mean of 0 and standard deviation of 0.01, and all biases were initialized as 0

Table 3. Quantitative results of our proposed fully automatic brain tumor segmentation method compared to the results from other recently published deep learning based methods. Here we tabulated the Dice Similarity Coefficient (DSC) for HGG, LGG and combined cases, respectively. Grey background highlighted the experiments on the BRATS 2015 datasets. Bold numbers highlighted the results of the best performing algorithm.

			DSC		
Method	**Data**	**Grade**	**Complete**	**Core**	**Enhancing**
Proposed	Cross-Validation on BRATS 2015 Training Datasets	HGG	0.88	0.87	0.81
		LGG	0.84	0.85	0.00
		Combined	0.86	**0.86**	0.65
Pereira16	BRATS 2013 Leaderboard	HGG	0.88	0.76	0.73
		LGG	0.65	0.53	0.00
		Combined	0.84	0.72	0.62
	BRATS 2013 Challenge	HGG	0.88	0.83	0.77
	BRATS 2015 Challenge	Combined	0.79	0.65	0.75
Havaei16	BRATS 2013 Training	Combined	0.88	0.79	0.73
	BRATS 2013 Challenge	Combined	0.88	0.79	0.73
	BRATS 2013 Leaderboard	Combined	0.84	0.71	0.57
	BRATS 2015 Challenge	Combined	0.79	0.58	0.69
Kamnitsas17	BRATS 2015 Training	Combined	**0.90**	0.76	**0.73**
	BRATS 2015 Challenge	Combined	0.85	0.67	0.63

WHAT IS THE BEST DATA AUGMENTATION APPROACH FOR BRAIN TUMOR SEGMENTATION USING 3D U-NET?

Methods

In this project a standard 3D U-Net architecture was used, which was trained with 4 MR images (T1, T1Gd, T2, FLAIR) to perform a 4 class segmentation: background, whole tumour (WT), tumour core (TC), and enhancing tumour (ET). Although more advanced segmentation networks have been proposed, a standard U-Net won the BraTS 2020 challenge.

The augmentation techniques used for this projects are:

- Patch extraction: from each original volume a sub-volume of shape 128 × 128 × 128 is extracted around its centre. In this way each sub-volume mostly contains brain tissue and not the surrounding background.
- Flipping: random flipping of one of the three different axes with 1/3 probability. • Rotation: rotation applied to each axis with angles randomly chosen from a uniform distribution with range between 0◦ and 15◦ , 30◦ , 60◦ , or 90◦ .
- Scale: scaling applied to each axis by a factor randomly chosen from a uniform distribution with range ±10% or ±20%.
- Brightness: power-law γ intensity transformation with its parameters gain (g) and γ chosen randomly between 0.8 - 1.2 from a uniform distribution. The intensity (I) is randomly changed according to the formula: Inew = g · I γ .
- Elastic deformation: elastic deformation with square deformation grid with displacements sampled from from a normal distribution with standard deviation σ = 2, 5, 8, or 10 voxels [10], where the smoothing is done by a spline filter with order 3 in each dimension

Table 1. Non-parametric p-values for comparing different types of data augmentation, obtained through a sign flipping test using the 125 validation subjects. ET = enhancing tumor, WT = whole tumor, TC = tumor core. The p-values have been multiplied with 36 (36 one sided tests) as Bonferroni correction for multiple comparisons.

	p-value for Dice score		
Comparison	**ET**	**WT**	**TC**
Flipping + PE > PE	0.0082	1.0	0.19116
Brightness + PE > PE	0.00036	0.00036	0.00036
Scale $\pm 10\%$ + PE > PE	0.00036	1.0	1.0
Scale $\pm 20\%$ + PE > PE	1.0	0.00036	0.00396
Rotation 0° - 15° + PE > PE	1.0	1.0	1.0
Rotation 0° - 30° + PE > PE	1.0	1.0	1.0
Rotation 0° - 60° + PE > PE	0.4676	0.2422	1.0
Rotation 0° - 90° + PE > PE	0.00036	1.0	1.0
Elastic deformation 2 + PE > PE	0.00036	0.00036	0.00036
Elastic deformation 5 + PE > PE	0.00036	1.0	0.44208
Elastic deformation 8 + PE > PE	0.00036	0.00071	0.13464
Elastic deformation 10 + PE > PE	0.00036	1.0	1.0

Results

They used 100,000 sign flips per test and the p-values are given in Table 1. Brightness augmentation and elastic deformations with a σ = 2 result in significantly higher Dice scores for all 3 tumor classes, scaling with ± 20% significantly improves the Dice scores for 2 classes, while flipping and 90◦ rotation only significantly improve one tumor class.

Here too, looking at the ranking, is it possible to say that brightness and elastic deformations with a σ = 2 are the two best augmentation techniques for this dataset, having 1st and 2nd position in the ranking respectively. Moreover, the other most important techniques are scaling with ±20%, rotation with random angle chosen between 0◦ - 90◦ , and flipping with having 4th, 7th , and 8th rank position respectively.