Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 06, June - 2024

Performance of U-Net Based Architecture for Brain Tumor Segmentation: Choice of Backbone

P Lokeshwar Reddy, Ritesh Jha, V. Bhattacharjee
Birla Institute of Technology, Mesra, Ranchi

ABSTRACT:
In this paper we present a 3D U-Net architecture to perform segmentation of brain tumors from multi-modal magnetic resonance scans. Detecting brain tumors from medical imaging scans, particularly magnetic resonance imaging, is vital for timely treatment planning and patient care in neuro-oncology. However, manual interpretation methods are time-consuming, subjective, and vary among clinicians, causing delays in diagnosis and potentially inaccurate results. This research work aims to address these challenges by creating an automated system for brain tumor detection and segmentation using machine learning techniques. Our main objective is to develop a reliable solution that accurately identifies and outlines tumor regions in MRI scans, enhancing diagnostic precision and speeding up clinical processes. Deep learning models have been trained using the BraTS2020 dataset. Different architectures have been used as the backbone for the 3D-U-Net model, and comparative results have been presented.

Keywords**: Image segmentation, U-Net Architecture, Convolutional neural networks**

1.INTRODUCTION

Artificial Intelligence (AI) and Machine Learning (ML) are revolutionizing various sectors globally, offering innovative solutions to complex problems. One such area is healthcare, where these technologies are enhancing medical diagnostics and decision-making. The human brain, an intricate organ that controls our bodies, is vulnerable to tumors. These abnormal growths can disrupt brain functions, making early detection and intervention crucial for patient well-being.

With the advent of AI and ML, we're witnessing a paradigm shift in our approach to diagnosing brain health issues. AI and ML are being increasingly adopted in medicine to improve patient outcomes. In the context of brain health, the integration of machine learning with medical imaging is redefining our diagnostic approach.

Magnetic Resonance Imaging (MRI), a medical imaging technique that uses strong magnetic fields and radio waves, is invaluable for examining the brain and detecting brain tumors. The MRI scanner captures cross-sectional images of the brain from various angles, revealing the different soft tissues in high resolution. This allows radiologists to identify any abnormal growths or masses that could indicate a tumor.

However, analyzing the vast amount of data from MRI scans can be challenging and time-consuming for radiologists. This has led to a growing interest in using AI and deep learning models to assist with the identification and segmentation of brain tumors from MRI scans.

Deep learning algorithms, trained on large datasets of labeled MRI images, can automatically identify brain tumors. These AI models can discern subtle patterns and irregularities that may suggest the presence of a tumor. Some models focus on classification tasks, determining whether a tumor is present or not. Others are designed for segmentation, outlining the exact boundaries and spatial extent of the tumor within the brain.

Automated tumor segmentation can significantly reduce the time radiologists spend on manual outlining. The deep learning algorithms can quickly process full 3D MRI volumes to segment tumor regions, providing a detailed map of the tumor location. This automated segmentation improves consistency and reproducibility compared to manual methods and enables quantitative analysis of tumor characteristics like volume and shape.

The rest of the paper is organized as follows. Section 2 presents the literature review, Section 3 presents the methodology. In Section 4 we present our results of experiments, and Section 5 presents the ablation study where we experiment with the augmentation techniques, the dropout and other parameters and study their impact on the results. Finally Section 6 concludes our work.

2. LITERATURE REVIEW

Convolutional neural networks (CNNs) were introduced as a solution for a variety of computer vision problems, demonstrating their accuracy and capability without sacrificing efficiency [1]. Deep learning models, such as AlexNet [2], VGGNet [3], ResNet [4], and DenseNet [5], have demonstrated efficacy in addressing a range of computer vision applications in recent times, garnering significant interest from both academic and industrial domains. Deep neural networks are rapidly being used in analysis of medical images due to their remarkable capacity to automatically extract highly discriminating characteristics [6–8].

Researchers have proposed a number of automatic picture segmentation techniques [9]. CNN and U-net are the two most widely used architectures for early image segmentation task solutions [10]. For instance, Chen et al. [11] proposed an auto-context version of the VoxResNet whereas, Feng et al. [12] created a 3D U-Net for brain tumor segmentation. Lee and colleagues [13] suggested a variation of U-Net design that was able to retain more local information while overcoming the limitations of traditional U-net. Attention gate-trained models acquire the implicit ability to emphasize important elements in an input image while suppressing irrelevant areas and such models were proposed by Oktay et al [14]and Noori et al. [15]. This multi-class segmentation challenge was broken down into three binary segmentation tasks based on subregion hierarchy by Wang et al. [16]. An ensemble of numerous UNets can increase segmentation accuracy since different models can have different error rates, according to Feng et al. [17]. Jun Ma [18] and Henry [19] proposed U-Net based architectures to create a segmentation map of brain tumors. Zahi [20] and his team proposed a Multi-View Pointwise U-Net (MPV U-Net) for segmenting brain tumors from multi-modality MRI scans. In the BRATS 2020 testing dataset, the enhanced tumor had a mean dice score of 0.715, the whole tumor had a score of 0.839, and the tumor core had a score of 0.768.

Savadikar [21] and his team used the Probabilistic U-Net to study the effects of applying various segmentation maps. The results from the BRATS 2020 testing data were 0.7988 for the whole tumor, 0.7771 for the tumor core, and 0.7249 for the enhancing tumor, while the scores for the validation data were 0.81898, 0.71681, and 0.68893, respectively. In [22] Tibe and colleagues developed a deep learning model using the 3D U-Net architecture for brain tumor segmentation from MRI scans of the BraTS 2020 dataset. Their methodology included mathematical models like edge detection and fuzzy clustering for tumor localization and pixel clustering. The model achieved an impressive accuracy of 98.5% for tumor segmentation, with high scores on key metrics like the dice coefficient. The researchers aim to extend their work to detect tumor severity and growth patterns, making significant contributions to diagnosis and treatment planning. In 2020, researchers Nagwa M. Aboelenein, Piao Songhao, and Alam Noor [23] introduced a novel architecture known as the Hybrid Two Track U-net (HTTU) for automatic brain tumor segmentation. This model employs various techniques including N4ITK Bias Correction, Focal loss, and Generalized dice score functions. It was trained on the BraTS 2018 dataset and achieved a dice score of 0.865 for the whole tumor. However, it fell short in identifying the underlying layers of the images used.

Kajal and Mittal's [24] research introduces a modified U-Net architecture with a 6-layer encoder-decoder CNN for precise brain tumor segmentation from MRI scans. Their work enhances the standard U-Net's segmentation accuracy by adding layers. They used the BRATS 2020 dataset for model training and testing, with preprocessing like cropping performed on images. The model, which uses focal and dice loss functions, achieved an impressive accuracy of 97.59% and an IOU score of 0.6413, surpassing other methods like LinkNet and FPN. Despite the lack of comparative analysis, their model contributes significantly to brain tumor segmentation, aiding diagnosis and treatment planning.

Díaz-Pernas [25] and team have developed a new deep learning method for identifying and outlining brain tumors. Their approach uses a multiscale CNN, which works at different scales, much like our eyes do. They tested their model on MRI scans from 233 patients with different types of brain tumors. The best part? No need for pre-processing to remove non-brain parts, making it more user-friendly for doctors. Their method outperformed seven other techniques, achieving an impressive accuracy of 97.3%. This shows the potential of multiscale learning in medical image analysis. Ranjbarzadeh [26] and team have developed a new deep learning method for detecting brain tumors from MRI scans. Their approach uses smart preprocessing and a special network called a C-CNN to extract both local and global features. They also consider the tumor's location with a "Distance-Wise Attention" mechanism. Tested on BRATS 2018, their model outperformed several others, although it struggled with very large tumors. Despite this, their method shows great promise for clinical use. Alpeshkumar and several others [27-30] have developed deep learning methods for finding and outlining brain tumors in MRI scans. Their results show how effective deep learning can be for analyzing brain tumors in medical images.

3. METHODOLOGY

3.1 Dataset

For this work, we utilized the Multimodal Brain Tumor Segmentation Challenge (BraTS) 2020 dataset, which is a widely used and publicly available resource for brain tumor segmentation research. The BraTS dataset provides a large collection of clinically acquired brain MRI scans along with expert-annotated tumor segmentations, making it a valuable resource for developing and evaluating brain tumor segmentation methods.

The BraTS 2020 dataset consists of 369 preoperative MRI scans from patients diagnosed with glioblastoma (GBM) or lower-grade glioma (LGG). These MRI scans were collected from multiple institutions, ensuring a diverse and representative dataset. Each patient's MRI scan includes four different imaging modalities: T1-weighted (T1), post-contrast T1-weighted (T1ce), T2-weighted (T2), and T2 Fluid Attenuated Inversion Recovery (FLAIR). These different modalities provide complementary information about the tumor's appearance and characteristics.

The dataset is divided into a training set (369 cases), a validation set (125 cases), and a test set (166 cases). The tumor regions in each MRI scan have been meticulously annotated by expert radiologists, following a standardized protocol. The annotations delineate three distinct tumor sub-regions: the enhancing tumor (ET), the peritumoral edema (ED), and the necrotic and non-enhancing tumor core (NCR/NET). These annotations serve as ground truth segmentation masks, which are essential for training and evaluating the deep learning models.

The BraTS 2020 dataset presents several challenges for brain tumor segmentation algorithms, such as varying tumor sizes, shapes, and locations within the brain, the presence of multi-focal tumors and diffuse lesions, heterogeneous intensity distributions within tumors and across modalities, and intensity similarities between tumor regions and surrounding healthy tissues. By working with this diverse and challenging dataset, our proposed deep learning model will be trained and evaluated on realistic and clinically relevant scenarios, ensuring its applicability and robustness in real-world settings.

3.2 Data Pre-processing:

In this work, several important data preprocessing steps were carried out to prepare the BraTS 2020 dataset for training the deep learning model. The first step involved loading the MRI images and their corresponding segmentation masks from the dataset. The paths to these files were organized into separate lists for T1-weighted, post-contrast T1-weighted, T2-weighted, FLAIR, and segmentation mask files. To ensure that the input features were on a consistent scale, intensity normalization was performed on the MRI images. This step is crucial as it can improve the model's training process and convergence by preventing any single feature from dominating the others due to differences in scale.

The code then iterates through the dataset, loading each set of MRI images (T2, T1ce, and FLAIR) and the corresponding segmentation mask. The MRI images were reshaped, normalized using a MinMaxScaler, and then stacked together along the channel dimension to create a multi-channel input volume. Regarding the segmentation masks, a preprocessing step was applied to reassign the label value of 4 to 3, as per the guidelines of the BraTS challenge. This step was necessary to align with the expected label values for the segmentation task.

To facilitate efficient training and inference processes, the combined images and masks were cropped to dimensions that were divisible by 64. This step ensured that the input volumes could be divided into patches of size 128x128x128 during the training process. Furthermore, the segmentation masks were one-hot encoded using Keras' function. This step converts the integer labels into a binary matrix representation, which is necessary for training the model on multi-class segmentation tasks. After the preprocessing steps, the dataset was split into a training set (258 samples) and a validation set (86 samples), which would be used for training and evaluating the deep learning model, respectively. These preprocessing steps ensured that the MRI images and segmentation masks were properly loaded, normalized, formatted, and prepared for input into the deep learning model, enabling efficient and effective training and inference processes.

4. MODEL ARCHITECTURE

This research work leverages a modified 3D U-Net model, which is a well-known architecture in the realm of image segmentation, as shown in Figure 1 [10]. What makes our model unique is the integration of a ResNet50 and ResNet101 backbone, acting as the encoder in our U-Net model. This combination is realized using the segmentation models 3D package.

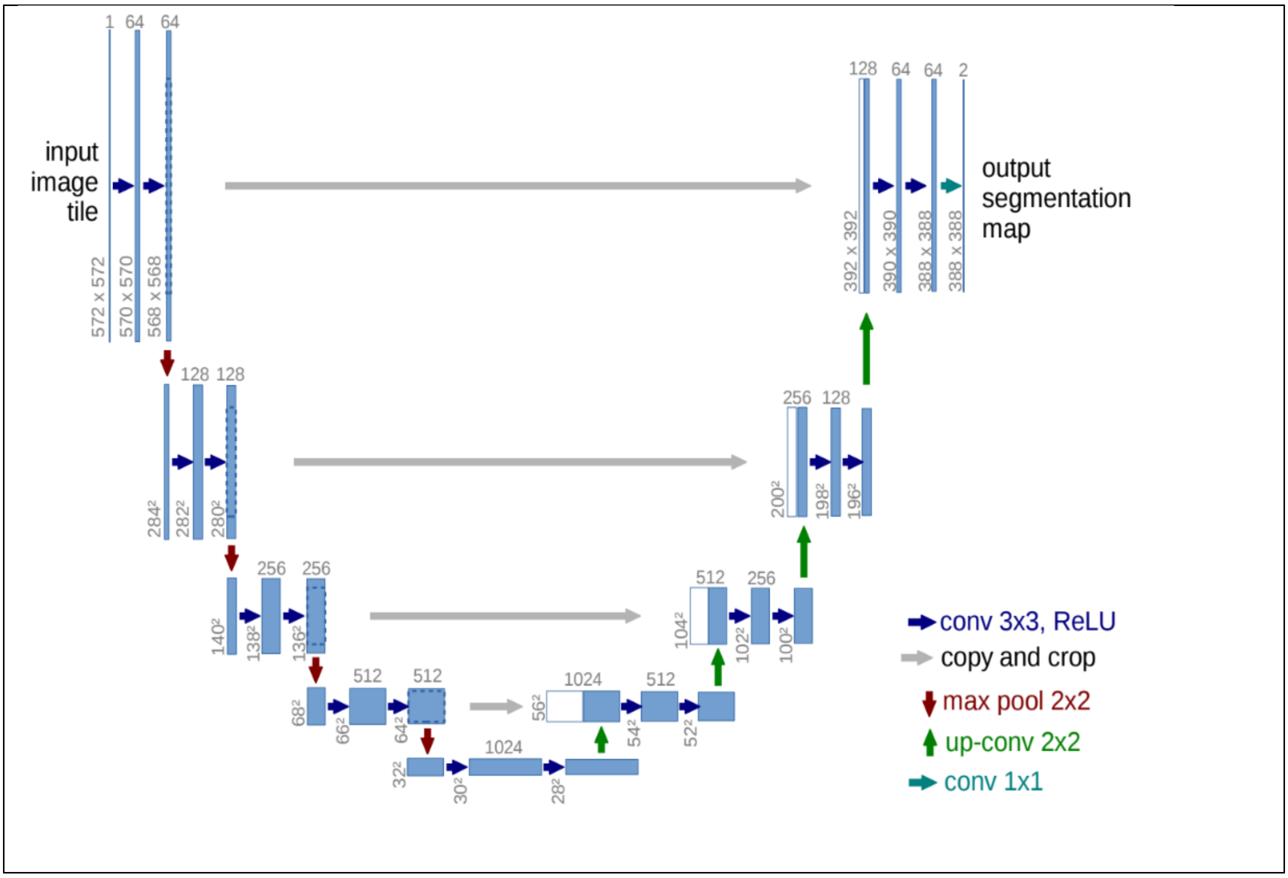

Figure 1. A model U-Net Architecture [10]

4.1 Encoder (ResNet50, ResNet101 Backbone)

The ResNet50 backbone, a pre-trained convolutional neural network (CNN), serves as the powerhouse for feature extraction in our model. The architecture of this backbone is organized into five blocks:

- Block 1: The input image first encounters a series of convolutional layers with a 7x7 kernel size and a stride of 2. This process is repeated four times, shrinking the spatial resolution of the feature map to a quarter of the original image.

- Block 2: The output from Block 1 is then passed through another series of convolutional layers, this time with a 3x3 kernel size and a stride of 1. After four repetitions, the feature map's spatial resolution is reduced to an eighth of the original image.
- Block 3: The process continues in a similar fashion, with the output from Block 2 undergoing another series of 3x3 convolutional layers with a stride of 1. After four repetitions, the feature map's spatial resolution is now a sixteenth of the original image.
- Block 4: The output from Block 3 is subjected to yet another series of 3x3 convolutional layers with a stride of 1. Four repetitions later, the feature map's spatial resolution is a thirty-second of the original image.
- Block 5: Finally, the output from Block 4 goes through a final series of 3x3 convolutional layers with a stride of 1. After four repetitions, the feature map's spatial resolution is a sixty-fourth of the original image.

4.2 Decoder (U-Net Architecture)

- The output from the ResNet50 backbone is then processed through a series of convolutional and Upsampling layers to generate the final segmentation mask. The architecture of the decoder is as follows:
- Upsampling Block 1: The output from the ResNet50 backbone is first upscaled by a factor of 2, followed by a 3x3 convolutional layer with a stride of 1.
- Upsampling Block 2: The output from the previous block undergoes a similar process, with another Upsampling layer (factor of 2) and a 3x3 convolutional layer with a stride of 1.
- Upsampling Block 3: The process is repeated again with the output from Block 2, with another Upsampling layer (factor of 2) and a 3x3 convolutional layer with a stride of 1.
- Upsampling Block 4: The output from Block 3 is upscaled once more by a factor of 2, followed by a 3x3 convolutional layer with a stride of 1.
- Upsampling Block 5: The output from Block 4 undergoes the final Upsampling (factor of 2) and convolutional layer (3x3 kernel size, stride of 1).
- Final Convolutional Layer: The output from Block 5 is passed through a final 1x1 convolutional layer with a stride of 1 to generate the final segmentation mask.

4.3 Output:

- The end product of the model is a segmentation mask that maintains the same spatial resolution as the input image. Each pixel value in this 3D array corresponds to the class label of the respective pixel in the input image. The model accepts an input of size 128x128x128x3 and generates an output of size 128x128x128x4.

4.4 Model Training

The training of the model is a crucial step where the model learns to segment brain tumors from 3D images. This is achieved by adjusting the model's internal parameters to minimize the difference between the model's predictions and the actual data (ground truth). Here's a more detailed explanation:

Optimizer:The optimizer is an algorithm that adjusts the internal parameters of the model to minimize the loss function. In this case, the Adam optimizer is used. Adam, short for Adaptive Moment Estimation, is a popular optimizer because it combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp. It adapts the learning rate for each weight in the model individually and computes adaptive learning rates for different parameters.

Loss Function:The loss function, also known as the objective function, is a measure of the model's error. The model's goal is to minimize this function. In this research work, a custom loss function based on the Dice coefficient is used. The Dice coefficient is a statistical metric that measures the similarity between two samples. In this case, it is used to measure the similarity between the predicted segmentation mask and the true mask. The Dice loss is then calculated as one minus the Dice coefficient, meaning that the model is trained to maximize the Dice coefficient, or equivalently, to minimize the Dice loss.

Dice Coefficient**:**The Dice coefficient is a measure of the overlap between two samples. In the context of image segmentation, it can be used to measure the similarity between the predicted segmentation mask and the true mask. The formula for the Dice coefficient is:

$$Dice = \frac{2 * | X \cap Y |}{| X | + | Y |}$$

Where:

$| X \cap Y |$ is the cardinality of the intersection of sets X and Y (i.e., the number of common elements between X and Y).

$| X |$ and $| Y |$ are the cardinalities of sets X and Y respectively.

Callbacks: Callbacks are special functions that are called at specific points during the training process, such as at the end of each epoch. They are used to save the model, adjust the learning rate, stop training early, etc. In this work, several callbacks are used:

- ModelCheckpoint: This callback saves the model after each epoch. The model weights are written to a file only if the validation Dice coefficient (as specified by the monitor argument) has improved compared to the previous epoch. This ensures that the best model is saved.
- EarlyStopping: This callback stops the training process if the validation loss does not improve for a specified number of epochs (as specified by the patience argument). This is useful to prevent overfitting and to save computational resources.
- TensorBoard: This callback logs the training and validation metrics for each epoch. These logs can be visualized in TensorBoard, a tool for visualizing learning curves and other training metrics.
- ReduceLROnPlateau: This callback reduces the learning rate if the validation loss does not improve for a specified number of epochs. Reducing the learning rate can help the model to overcome local minima in the loss landscape and to converge to a better solution.

Data Generators: Data generators are used to load and preprocess the data in batches, allowing for efficient memory usage. In this work, a custom data generator function is used that loads the images and masks in batches. This is especially useful when working with large datasets that do not fit into memory.

Training Process: The model is trained for a total of 10 epochs. An epoch is one complete pass through the entire training dataset. The number of steps per epoch is calculated as the total number of training images divided by the batch size. The same calculation is done for the validation steps using the total number of validation images.

During each epoch, the model's parameters are updated in an attempt to minimize the loss function. After each epoch, the validation loss is calculated on a separate validation dataset that the model has not seen during training. This allows for monitoring the model's performance on unseen data and helps to detect overfitting.

5. RESULTS AND DISCUSSION

In this work, our aim was to compare the performance of segmentation models built on two different backbone architectures - ResNet101 and ResNet50. Figure 2 presents a sample test case of segmentation. The aim was to understand the impact of the choice of backbone on the accuracy of segmentation. We evaluated the models using several key metrics such as Dice Coefficient, Jaccard Index, Precision, Recall, and F1 Score.

5.1 ResNet101 Performance

Dice Coefficient: The model built on ResNet101 achieved a Dice Coefficient of 0.974, indicating a high degree of overlap between the predicted and actual segmentations.

Jaccard Index: The Jaccard Index for the ResNet101 model stood at 0.949, demonstrating a strong agreement between the predicted and actual segmentations.

Precision: The precision of the ResNet101 model was 0.974, showcasing its ability to accurately identify positive predictions.

Recall: The ResNet101 model had a recall of 0.974, indicating its effectiveness in capturing true positive instances within the dataset.

F1 Score: The F1 Score for the ResNet101 model was 0.974, reflecting a balanced harmony between precision and recall.

5.2 ResNet50 Performance

Dice Coefficient: The model built on ResNet50 scored a Dice Coefficient of 0.954, indicating a significant overlap between the predicted and actual segmentations.

Jaccard Index: The Jaccard Index for the ResNet50 model was 0.912, showing a considerable agreement between the predicted and actual segmentations.

Precision: The precision of the ResNet50 model was 0.954, showcasing its ability to accurately identify positive predictions.

Recall: The ResNet50 model had a recall of 0.954, indicating its effectiveness in capturing true positive instances within the dataset.

F1 Score: The F1 Score for the ResNet50 model was 0.954, reflecting a balanced harmony between precision and recall.

6. CONCLUSION

The comparative analysis of the segmentation models based on ResNet101 and ResNet50 provided some interesting insights. Firstly, the ResNet101 model outperformed the ResNet50 model across all evaluation metrics, indicating its superior segmentation accuracy. This can be attributed to the depth and complexity of the ResNet101 architecture, which allows the model to capture more complex features within the images, leading to enhanced segmentation accuracy.

While both models demonstrated high segmentation accuracy, it's important to consider practical aspects such as computational complexity and resource requirements. The ResNet101 model, being deeper, may require more computational resources compared to the ResNet50 model. Therefore, the choice of backbone architecture should be a careful balance between segmentation accuracy and computational efficiency, especially in real-time or resource-constrained applications.

In conclusion, the findings highlight the importance of choosing the right backbone architecture when designing segmentation models. While deeper architectures like ResNet101 provide superior performance, shallower architectures like ResNet50 can also deliver competitive segmentation accuracy. These insights offer valuable guidance for researchers and practitioners in the field of medical image segmentation and computer vision. Future research could explore additional backbone architectures and optimization techniques to further enhance segmentation accuracy and efficiency, catering to a variety of application scenarios and requirements.

Several challenges need to be overcome, including the diverse nature of tumor characteristics across patients and imaging types, the scarcity of expertise in brain tumor diagnosis, and ensuring the model works effectively across different patient groups and imaging protocols. By utilizing machine learning algorithms and datasets like the BRATS (Brain Tumor Segmentation) dataset, we aim to create a scalable solution that supports clinical decision-making, alleviates resource constraints, and ultimately improves patient outcomes in neuro-oncology. Our vision is to revolutionize brain tumor diagnosis and treatment by providing healthcare providers with accurate, efficient, and scalable tools.

DECLARATIONS

- Competing Interests: The authors have no competing interests to declare that are relevant to the content of this article.
- Funding: No funds, grants, or other support was received.
- Ethics approval: Not Applicable
- Consent: All authors have consented to submit this article to the present journal. Data Availability: Data is available at www.kaggle.com
- Code availability: Shall be made available on request.
- Authors’ contribution: Lokeswar: Implementation, Data Preparation, Initial draft writing; Ritesh Jha: Conceptualization, Supervision, Draft preparation; Vandana Bhattacharjee: Conceptualization, Supervision, Draft revision.

REFERENCES

[1] Yamashita R, Nishio, M, Do RKG., Togashi (2018) Convolutional neural net works: an overview and application in radiology. Insights into Imaging 9(4), 611- 629. https://doi.org/10.1007/s13244-018-0639-9

[2]Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks, in Proc. Adv. Neural Inf. Pro cess. Syst. :1097-1105.

[3] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv:1409.1556. [Online]. Avail able: http://arxiv.org/abs/1409.1556

[4] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770778.

[5] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, Densely connected convolutional networks, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 47004708.

[6] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature, vol. 542, no. 7639, pp. 115118, Feb. 2017.

[7] V.Gulshan,L.Peng,M.Coram,M.C.Stumpe,D.Wu,A.Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster, Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs, JAMA, vol. 316, no. 22, p. 2402, Dec. 2016.

[8] Y. Liu, K. Gadepalli, M. Norouzi, G. E. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Q. Nelson, G. S. Corrado, J. D. Hipp, L. Peng, and M. C. Stumpe, Detecting cancer metastases on gigapixel pathology images, 2017, arXiv:1703.02442. [Online]. Avail able: http://arxiv.org/abs/1703.02442

[9] Bakas, S., et al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATSchallenge. CoRR abs/1811.02629 (2018). http://arxiv.org/abs/1811.02629

[10] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedi cal image segmentation. CoRR abs/1505.04597 (2015). http://arxiv.org/abs/1505. 04597

[11] Chen, H., Dou, Q., Yu, L., Qin, J., Heng, P.A.: Voxresnet: deep voxelwise residual networks for brain segmentation from 3D MR images. NeuroImage 170, 446–455 (2018). https://doi.org/10.1016/j.neuroimage.2017.04.041

[12] Feng, X., Tustison, N.J., Patel, S.H., Meyer, C.H.: Brain tumor segmentation using an ensemble of 3D U-nets and overall survival prediction using radiomic features. Front. Comput. Neurosci. 14, 25 (2020). https://doi.org/10.3389/fncom. 2020.00025

[13] Lee, B., Yamanakkanavar, N., Choi, J.Y.: Automatic segmentation of brain MRI using a novel patch-wise U-net deep architecture. PloS one 15(8), e0236493 (2020). https://doi.org/10.1371/journal.pone.0236493

[14] tay, O., et al.: Attention U-Net: Learning where to look for the pancreas. https://arxiv.org/pdf/1804.03999

[15] Noori, M., Bahri, A., Mohammadi, K.: Attention-guided version of 2D UNet for automatic brain tumor segmentation. In: 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE). pp. 269–275. IEEE, 24–25 Octo ber 2019. https://doi.org/10.1109/ICCKE48569.2019.8964956

[16] Wang, G., Li, W., Ourselin, S., Vercauteren, T.: Automatic brain tumor segmenta tion using cascaded anisotropic convolutional neural networks. In: Crimi, A., Bakas, S., Kuijf, H., Menze, B., Reyes, M. (eds.) BrainLes 2017. LNCS, vol. 10670, pp. 178–190. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75238-9

[17] Feng, X., et al.: Brain Tumor Segmentation Using an Ensemble of 3D U-Nets and Overall Survival Prediction Using Radiomic Features. Frontiers Comput. Neurosci. 14 25 (2020). https://doi.org/10.3389/fncom.2020.00025

[18] J. Ma, "Estimating Segmentation Uncertainties Like Radiologists." [Online]. Available: https://qubiq.grand-challenge.org/Home/

[19] T. Henry et al., "Brain tumor segmentation with self-ensembled, deeply-supervised 3D U-net neural networks: a BraTS 2020 challenge solution," Oct. 2020, [Online]. Available: http://arxiv.org/abs/2011.01045

[20] Z. and Z. Q. and F. Y. Zhao Changchen and Zhao, "MVP U-Net: Multi-View Pointwise U-Net for Brain Tumor Segmentation," in Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, S. Crimi Alessandro and Bakas, Ed., Cham: Springer International Publishing, 2021, pp. 93–103.

[21] R. and G. B. Savadikar Chinmay and Kulhalli, "Brain Tumour Segmentation Using Probabilistic U-Net," in Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, S. Crimi Alessandro and Bakas, Ed., Cham: Springer International Publishing, 2021, pp. 255–264.

[22] K. S. Tibe, A. Kangude, P. Reddy, P. Kharate, V. Lomte, and F. Year, "Brain Tumor Detection and Segmentation using UNET," International Research Journal of Engineering and Technology, 2022, [Online]. Available: www.kaggle.com

[23] N. M. Aboelenein, P. Songhao, A. Koubaa, A. Noor, and A. Afifi, "HTTU-Net: Hybrid Two Track U-Net for Automatic Brain Tumor Segmentation," IEEE Access, vol. 8, pp. 101406–101415, 2020, doi: 10.1109/ACCESS.2020.2998601.

[24] M. Kajal and A. Mittal, "A Modiied U-Net Based Architecture for Brain Tumour Segmentation on BRATS 2020," 2022, doi: 10.21203/rs.3.rs-2109641/v1.

[25] F. J. Díaz-Pernas, M. Martínez-Zarzuela, D. González-Ortega, and M. Antón-Rodríguez, "A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network," Healthcare (Switzerland), vol. 9, no. 2, Feb. 2021, doi: 10.3390/healthcare9020153.

[26] R. Ranjbarzadeh, A. Bagherian Kasgari, S. Jafarzadeh Ghoushchi, S. Anari, M. Naseri, and M. Bendechache, "Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images," Sci Rep, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-90428-8.

[27] J. Alpeshkumar Doshi, D. Jeel Alpeshkumar, M. Student, P. J. Patel, A. Professor, and T. Hitesh Shah, "Brain Tumor Detection and Segmentation." [Online]. Available: https://www.researchgate.net/publication/352210377

[28] Chen, W., Liu, B., Peng, S., Sun, J., Qiao, X.: S3D-UNet: separable 3D U-Net for brain tumor segmentation. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.) BrainLes 2018. LNCS, vol. 11384, pp. 358–368. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11726-9 32

[29] Kotowski K., Adamski S., Malara W., machura B., Zarudzki L. and Nalepa J. (2021) Segmenting Brain Tumors from MRI Using Cascaded 3D U-Nets, A. Crimi and S. Bakas (Eds.): BrainLes 2020, LNCS 12659, pp. 265–277.

[30] www.kaggle.com