Brain Tumor Segmentation Project Report

Hannah Han (hannahh@mit.edu) Kathryn Le (kle@mit.edu)

1. Abstract

Brain tumor segmentation is an important part of a patient's diagnosis and treatment process, but it is currently difficult to do because of inefficient manual segmentation methods, and vague tumor boundaries and shapes [5][8]. Existing literatures use techniques such as the U-Net architecture and SPPF+ to create well-performing models [1][3]. We aim to create a new, better performing model architecture by combining existing U-Net architecture with SPPF techniques along with data augmentation to improve model accuracy and generalization. Our results show that adding a SPP or SPPF layer with or without data augmentation does not significantly improve model performance and the dice coefficient. We also notice that adding data augmentation does improve generalization for the U-Net model with the SPP layer (U+SPP model) but does not significantly improve generalization for the U-Net model with the SPPF layer (U+SPPF model). In conclusion, although our model did not improve better than the base model, we gained important insights into the SPP/F layer and data augmentation, as well as possible future experiments that can be done to further improve the model. With better models, we hope that brain tumor (and other cancer) segmentation can improve and help more people live longer.

2. Introduction

With 90,000 people diagnosed with primary brain tumors, approximately 17,200 people die from malignant brain tumors every year [7] and the glioblastoma patients have an average survival rate of 14 months after diagnosis [6]. Detecting and localizing the tumor are crucial for a proper diagnosis and treatment plan. However, manual tumor segmentation from brain MRI images is error-prone and time-consuming [8]. In addition, segmenting gliomas and glioblastomas is challenging due to their indistinct boundaries, lack of contrast, and ability to manifest anywhere in the brain, taking on any size and shape [3]. Thus, leveraging advances in computer vision for brain tumor segmentation can immensely aid medical personnel in more accurately diagnosing and treating brain tumors, giving patients hope of a longer life. Currently, there are many artificial intelligence and computer science methods using deep learning, convolutional neural networks, attention, and other techniques to solve this challenging segmentation problem [1][3][10]. Many have shown significant progress, and we hope to contribute to this rapidly improving field.

3. Related Work

3.1. Existing Methods

There are many existing methods for brain tumor segmentation. U-Net was the baseline model architecture introduced in 2015 [10] and NNU-Net built on top of this architecture using more advanced techniques such as residual connections, attention mechanisms, and leaky ReLUs to increase performance [3]. Other methods utilize pretrained models such as YOLOv7 and improve them with attention mechanisms, SPPF+, and BiFPN with high success of segmenting three types of brain tumors [1]. In these models, there is a baseline model or architecture used to solve the basic problem improved by different techniques to increase accuracy, which is the method we strive to work with.

3.2. U-Net

The U-Net architecture was first introduced in 2015 for biomedical segmentation applications [10]. The U-Net consists of a contracting path and an expansive path using convolutional layers for both sides, down-sampling for the contracting side, and up-sampling for the expansive side. ReLUs and max pooling were used as nonlinear activation functions [10]. The U-Net model was the state-of-the-art at the time, beating many other neural networks in biomedical segmentation competitions. Although the U-Net alone is no longer the state-of-the-art, it is a ground-breaking architecture that is the foundation of many recent state-of-the-art models.

3.3. NNU-Net

NNU-Net is a U-Net based model that automatically configures the rule-based parameters based on the dataset fingerprint on top of already fixed parameters. Through cross-validation, up to three configurations are trained and the best model is chosen [2]. They accomplished state-of-the-art performance on 53 diverse segmentation tasks. Although the showed strong generalizability on many of these tasks, it was not able to beat other models for every task indicating possible shortcomings in the model such as shortcomings in domain-specific knowledge or handling cases the model has not seen before.

3.4. YOLOv7, Attention, and SPPF+

YOLOv7 is a state-of-the-art real-time object detection algorithm that predicts boundaries and class probabilities in an image consisting of a backbone and a head network that extracts features of interest and processes them for object detection respectively. The attention mechanism captures long range dependencies from the input and extracts features. SPPF+ (Spatial Pyramid Pooling with Fusion) reuses features and

applies different filter sizes during max pooling to improve feature interaction and localization [1]. The model was able to reach 99% accuracy with their dataset, but its ability to generalize to new data has not been tested.

4. Methods

Our proposed method is to combine techniques from existing models such as SPPF to create our own method for brain tumor segmentation starting from a baseline U-Net architecture.

Data Collection We will be using the data provided in the GitHub repository **Error! Reference source not found.** we are r eferencing with data from the BraTS2020 dataset [2]. The dataset is split into 369 training data points and 125 validation data points. Each data point is a 3D image consisting of 4 modalities (flair, t1, t1ce, and t2). Each modality consists of 155 volume slices. In this project, we will only be using the flair and t1ce modalities and slices 60-134 inclusive (75 total volume slices). This dataset is very well known for its clean data, however, the dataset is relatively small making it very challenging to achieve good generalization.

Model Choice From previous works, the U-Net architecture and SPPF are crucial components to improving the performance of their segmentation model, so we aim to combine these two techniques. Our goal was to use a baseline U-Net model and apply SPPF to create precise boundaries of any segmented tumor area to accurately segment irregular tumor shapes while maintaining the high prediction accuracy rate of previous works. Because SPPF and SPP are similar in structure, we decided to experiment with both methods for more complete results [4]. For example, we can compare their results to see which pooling method is more suitable for brain tumor segmentation. We will also experiment with data augmentation (specifically horizontally flipping the images) to make the model more robust and generalizable.

The baseline U-Net model consists of convolutional layers, convolutional transpose layers, ReLU activation functions, max-pooling layers, he_normal weight initialization, Dice Loss Crossentropy loss, the Adam optimizer with a starting learning rate of 0.001, and the reduce learning rate on plateau function, which helps with dynamic learning rates during training. In our experiments, we added the SPP/F layer at the bottleneck of the U-Net (after the fourth convolutional layer) and the data augmentation into the data generator. The model was trained on 35 epochs, so we continued to train the model using 35 epochs with a batch size of 1 (where each batch consists of 75 volume slices of one t1ce and one flair 3D image).

Our new model architecture of combined techniques from successful models will improve generalization and accuracy of segmentation. If our model shows better segmentation than previous work, then there is potential for our architecture to be used to segment tumors or lesions in other areas of the body.

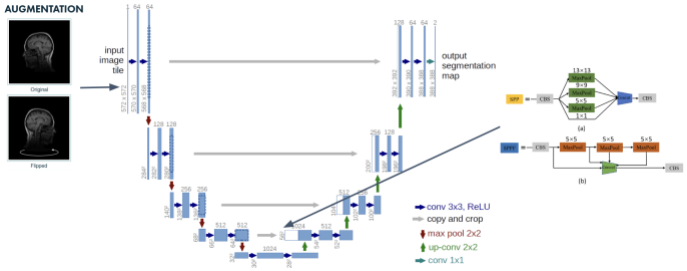

Figure 1: The U-Net architecture that we are using as our baseline architecture based on [10] along with data augmentation and SPP/F layers.

Training Process Using the training set of data, we will train our U-Net model with different techniques such as SPP and SPPF to pool the feature maps and data augmentation to prevent overfitting. Then, we will calculate Dice Loss and utilize the Adam optimizer and backpropagation to update our weight parameters.

Experiments We are conducting two experiments: adding a SPP or SPPF layer into the U-Net model and data augmentation. We added the SPP/F layer into the fourth contracting convolutional layer to help extract the most important features for segmentation. For data augmentation, because the model is training on a batch size of 1 with 75 volume slices, we are randomly generating a number from 0 - 1 for every batch and if the number is less than 0.2 (for 20% probability), the image will be horizontally flipped. This means that, on average, the model will be trained on data where 20% is augmented.

5. Experimental Results and Discussion

Result Evaluation: We will evaluate our results using the Dice Similarity Coefficient (DSC) metric, which is an overlap measure between the predicted segmentation and the ground truth segmentation provided by the dataset [9]. The DSC metric returns 1 if the two segmented areas overlap perfectly and 0 if the two segmented areas do not overlap at all. A higher DSC score represents stronger model performance.

5.1. SPP/F Layer

For our first experiment, we added an SPP layer into the U-Net model and trained it on the same dataset as the U-Net model [12]. SPP conducts a parallel max pooling operation of three different sized kernels [4]. To compare the performances of our U+SPP model to the base U-Net model, we mainly used the dice coefficient metric. During training, after 35 epochs for both models, our U+SPP model displayed a higher dice coefficient of 0.63 compared to the base model's dice coefficient of 0.62. Then

during testing, our U+SPP model performed worse than expected with a dice coefficient of 0.57 compared to the base model's dice coefficient of 0.60.

Observing that for our model, the training loss of 0.0163 seemed much lower than the test loss of 0.0282, we can conclude that our U+SPP model overfit the training data. Another reason that could explain why the U+SPP model did not perform as good as we expected was because the SPP layer was only one pooling layer among many other max pooling layers in the U-Net architecture, so the SPP layer's might not have contributed enough change to lead to a better performing model.

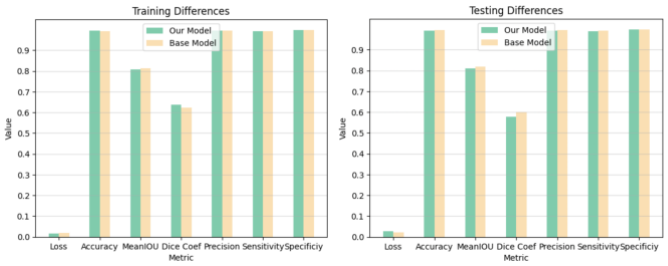

Figure 2: The comparison bar charts of various metrics of our U+SPP model (green) and the base U-Net model (yellow), from training (left) and testing (right).

For our second experiment, we replaced the SPP layer with a SPPF layer and trained using 35 epochs again. This experiment differs because SPPF conducts a serial operation of three same-sized kernels [4]. We trained this new model version on the same dataset as the U+SPP model. Similarly, we compared the performances of the U+SPPF model with the U+SPP model and the base model. During training, the U+SPPF model performed a little worse than the other two models with dice coefficient of 0.6063 and loss 0.0198. Then during testing, the U+SPPF generalized a little better than the U+SPP model with a dice coefficient of 0.5953. But the U+SPPF model had a higher loss value of 0.0322.

We observe from these results that an SPPF layer performs a little better at generalization than SPP for the tumor segmentation task.

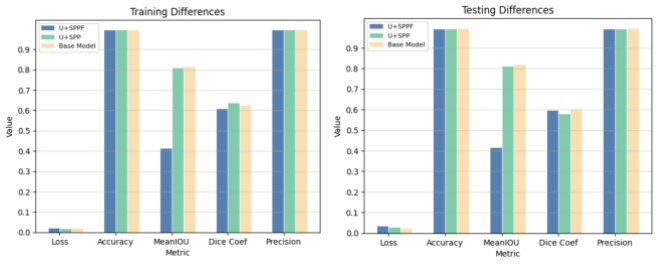

Figure 3: The comparison bar charts of various metrics of our U+SPPF model (blue), our U+SPP model (green) and the base U-Net model (yellow), from training (left) and testing (right).

5.2. Data Augmentation

Because we noticed that the U+SPP/F models were overfitting the data, we decided to experiment with including data augmentation to help our model generalize to the data better.

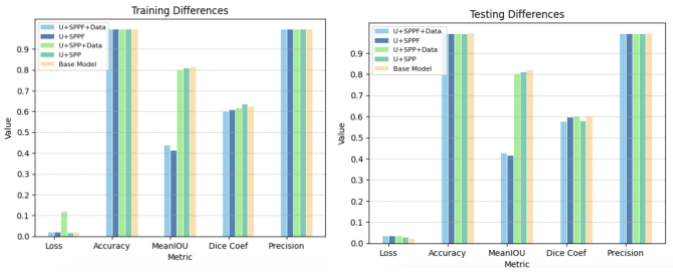

Figure 4: The comparison bar charts of various metrics of our U+SPPF model, our U+SPP model with and without data augmentation, and the base U-Net model, from training (left) and testing (right).

First, we tried to add data augmentation to improve the performance of the U+SPP model. Although adding data augmentation did not significantly improve model performance, the testing dice coefficient for the U+SPP model improved from around 0.57 (without data augmentation) to 0.60 (with data augmentation). Data augmentation seemed to prevent overfitting because the training dice coefficient for the U+SPP+Data augmentation model was lower than the training dice coefficient for the U+SPP model but the U+SPP+Data augmentation did better during testing. This means that the U+SPP+Data augmentation model is better with generalization.

However, when we tried to use data augmentation with the SPPF layer, the model seemed to do better without data augmentation for training and testing. For training, the U+SPPF model had a dice coefficient of around 0.60 compared to the dice coefficient of around 0.61 for the U+SPPF+Data model. For testing, the U+SPPF model had a dice coefficient of around 0.59 while the U+SPPF+Data had a dice coefficient of around 0.57. Overall, data augmentation is still be an effective method to improve model performance and generalization for certain cases.

6. Conclusion

Overall, our methods did not significantly improve the model performance. However, through our experimentation, we have possible explanations for reasons why the model did not improve and possible experimentation directions to improve future brain tumor segmentation U-Net models.

Possible reasons way our model did not improve are data loss, over-compression, and overfitting. Because the SPP/F layer is a pooling layer, we might be losing important information by including the additional pooling layer causing the model to struggle to construct the predicted segmentation. Another reason that the model did not improve is over-compression. We added

the SPP/F layer after the fourth convolutional layer when the input has already been significantly compressed and downsampled. Adding an extra pooling layer at this step may have caused data loss to be amplified since the receptive field is much larger per value after multiple convolutional layers. Finally, adding the SPP/F layer may have increased the chance of overfitting, which we tried to mitigate by including data augmentation. However, flipping 20% of the images horizontally may not have been the most effective data augmentation techniques and overfitting may still have occurred.

With these possible reasons for why our model did not improve, there are many experiments we can try in the future to improve the model such as adding the SPP/F layer in a different step of the U-Net pipeline and/or attempting different data augmentation methods. We could try to add the SPP/F layer earlier in the contracting path of the U-Net or replace the max-pooling layers with SPP/F layers to determine whether SPP/F can improve the U-Net performance and if it's more effective than other pooling methods. We can also experiment with different data augmentation methods such as increasing the amount of the data that is being augmented and/or combining different data augmentation approaches (scaling, rotating, blurring, etc.).

Although our experiments did not directly create a model with improved performance, they were able to help us gain deeper insight into how the model works and what future experiments can be done to improve the model.

7. Contributions to Society

Our findings reveal practices that may be less effective to improving brain tumor segmentation models and highlights some possible explanations for why. We also directly contribute to potential further research directions, such as additional variation for data augmentation or changing the frequency or location of the SPP/F layer. If we are able to discover a more accurate segmentation of brain tumor shapes, this new technology can help doctors treat brain tumor patients with more confidence. Because biopsies are important for doctors to do to determine brain tumor type while the tumor is still small, accurately outlining the boundary of a small, irregularly shaped tumor will benefit both doctors and patients.

8. Work Allocation

At the start of the project, we tried to build our own U-Net model from scratch. Kathryn wrote the model class while Hannah wrote the loss function and evaluation metrics (Dice loss and the Dice Similarity Coefficient). We also contributed equally to the Progress Report with Kathryn working on describing what we have already finished and Hannah working on describing the work allocation and goals we aim to reach by the end of the project. However, after a few attempts at debugging the model we wrote from scratch, we decided to use an existing baseline U-Net model instead. This is so that we can focus more on improving the model and running experiments instead of debugging a model that already exists.

After running the U-Net model, we aimed to reduce the resulting Dice loss through augmenting the data and adding the SPP/F layer. Hannah experimented on adding the SPP layer and SPPF layer while Kathryn experimented on augmenting the data after the addition of the different layers. Then, we compared our DSC score to the original DSC score along with other metrics to see if our new model performed better than the original model.

Throughout the process, we came up with many ideas on how to integrate the SPP/F layer and data augmentation into the existing model. At first, Hannah added the SPP layer after the fourth down-sampling convolutional layer. Kathryn tried to horizontally flip the data and append it to the original dataset to include data augmentation. Our first experiments did not show much improvement in the Dice Coefficient compared to the original model. In the second round of experiments, Kathryn tried to change the way we augmented the data by randomly horizontally flipping 20% of the data. Hannah tried to replace all max pooling operations with SPP, but due to challenges with tensor size matching, she was unable to move forward with this experiment. Instead, Hannah decided to move forward with just replacing the one SPP layer with an SPPF layer.

At the end, Kathryn and Hannah worked on and edited the final paper and presentation slideshow together. Kathryn found and cited all the resources, while Hannah created the comparison bar graphs.

References

[1] A. B. Abdusalomov, M. Mukhiddinov, and T. K. Whangbo, "Brain tumor detection based on deep learning approaches and Magnetic Resonance Imaging," *Cancers*, vol. 15, no. 16, p. 4172, Aug. 2023. doi:10.3390/cancers15164172

[2] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, et al. "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)", IEEE Transactions on Medical Imaging 34(10), 1993-2024 (2015) DOI: 10.1109/TMI.2014.2377694 F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "NNU-net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Dec. 2020. doi:10.1038/s41592-020-01008-z

[3] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "NNU-net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Dec. 2020. doi:10.1038/s41592-020-01008-z

[4] Huang, Liang. "(PDF) ASFF-Yolov5: Multielement Detection Method For ..." *ResearchGate*, July 2022, www.researchgate.net/publication/362193968_ASFF-YOLOv5_Multielement_Detection_Method_for_Road_Traffic_in_UAV_Images_Based_on_Multiscale_Feature_Fusion.

[5] M. Havaei et al., "Brain tumor segmentation with Deep Neural Networks," Medical Image Analysis, vol. 35, pp. 18–31, Jan. 2017, doi: 10.1016/j.media.2016.05.004.

[6] R. Ranjbarzadeh *et al.*, “Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images,” *Scientific Reports*, vol. 11, no. 1, May 2021. doi:10.1038/s41598-021-90428-8

[7] “Brain tumor faqs-learn more or donate today!: Abta,” American Brain Tumor Association, https://www.abta.org/about-brain-tumors/brain-tumor-education/#:~:text=Approximately%2090%2C000%20people%20are%20diagnosed,of%20a%20primary%20brain%20tumor. (accessed Mar. 27, 2024).

[8] D. Daimary, M. B. Bora, K. Amitab, and D. Kandar, “Brain tumor segmentation from MRI images using hybrid convolutional neural networks,” *Procedia Computer Science*, vol. 167, pp. 2419–2428, 2020. doi:10.1016/j.procs.2020.03.295

[9] M. Antonelli et al., “The Medical Segmentation Decathlon,” Nat Commun, vol. 13, no. 1, p. 4128, Jul. 2022, doi: 10.1038/s41467-022-30695-9.

[10] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” arXiv, May 18, 2015. doi: 10.48550/arXiv.1505.04597.

[11] Ovh, “Ai-training-examples/notebooks/computer-vision/image-segmentation/tensorflow/brain-tumor-segmentation-unet/notebook_image_segmentation_unet.ipynb at main · OVH/AI-training-examples,” GitHub, https://github.com/ovh/ai-training-examples/blob/main/notebooks/computer-vision/image-segmentation/tensorflow/brain-tumor-segmentation-unet/notebook_image_segmentation_unet.ipynb (accessed May 8, 2024).

[12] ShahzaibWaseem, “Shahzaibwaseem/SpatialPyramidPooling_tf2: Spatial Pyramid pooling is used when there are variable sized data in the dataset, the image shape is not specified at the start of the code and can be changed dynamically.,” GitHub, https://github.com/ShahzaibWaseem/SpatialPyramidPooling_tf2/tree/master (accessed May 8, 2024).