Applying U-Net for Medical Image Segmentation: A Case Study

Carson Keller

Department of Computer Science, Boise State University

1 Introduction

In computer vision, image segmentation is the process of algorithmically partitioning an image into meaningful regions from which objects of interest are identified as defined by pixel characteristics relative to those regions [1]. It serves many functions, including as a means to detect and isolate objects in autonomous driving systems or to compare and distinguish between likenesses in facial recognition applications. Notably, automated segmentation methods are an important development in medical image processing, providing the ability to automatically segregate and demarcate structures and other areas of interest across a range of radiological modalities, including ultrasound (US), computed tomography (CT), and magnetic resonance imaging (MRI).

Radiological imaging technology is a crucial tool for modern medicine in confirming diagnoses, informing prognoses, and aiding treatments. As its capability, accuracy, and resolution has increased, so too has its usefulness. The application of image segmentation to the robust field of radiology serves only to enhance those benefits. It poses the unique opportunity for augmenting and streamlining doctors' workflows by providing computer-aided diagnoses, allowing better surgical planning and navigation, informing targeted treatments, and enabling enhanced efficiency, ultimately improving overall patient outcomes through revolutionized patient care [2]. When so many patients' prognoses can be drastically impacted by these factors, when an early brain cancer diagnosis and quick, tailored treatment can mean the difference between death and survival, this technology has the potential to greatly impact the health and well-being of our society and communities, our friends and family, and, potentially, even ourselves.

This particular experiment serves as a case study, applying image segmentation methods to a dataset of MRI scan image files to assess the viability of this process as a proof of concept. Additionally, it also aims to evaluate the performance of different combinations and variations of deep learning models architectures, metrics, and loss functions informed by academic literature. This is done in the hope of finding a most successful, most performant composite model structure that yields the best results.

2 Related work

Traditional approaches to image segmentation include direct region detection methods [3], graph-based methods [4], and active contour and level set models [5]. Later approaches have attempted to adapt and apply traditional machine learning algorithms [6], such as support vector machines (SVMs) [7] and unsupervised clustering [8]. However, in recent years, more significant progress has been achieved utilizing deep learning models [9] [10] [11].

The most well-known deep learning model architecture for image segmentation, U-Net [12], is the modification of the convolutional neural network (CNN) architecture into an encoder-decoder network. While CNNs alone have demonstrated significant performance when handling image input [13], the hybridization of these two network types has yielded incredible efficacy in image segmentation applications. Through the use of convolutional layers, CNNs determine the presence of meaningful features and construct feature maps that are passed on to deeper layers in the model. Further, pooling layers allow dimensionality reduction, losing information but significantly reducing the overall number of parameters needed by the model. Implementing an encoder portion into the model architecture, as U-Net does, produces various feature maps or "hidden states" that capture the image input's relevant characteristics and elements before combining them to generate a context vector, an "encoded" numeric matrix rep-

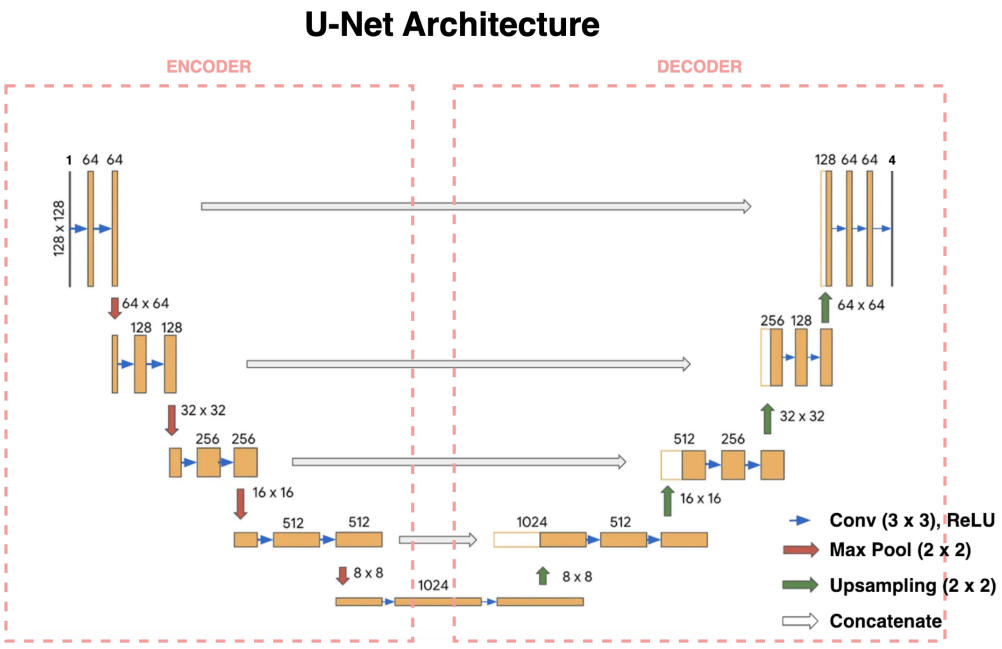

Figure 1: Modified U-Net architecture with relevant input and output dimensions, layers, convolutional filter sizes, and other features accurate to this particular implementation.

resentation of the most important features. The decoder portion is then responsible for translating this context vector into a mask, or segmented version of the original image, maintaining the important features and segmenting the meaningful region as accurately as possible. This architecture thus enables end-to-end feature extraction and pixel classification. As depicted in Figure 1, the structure of the baseline model is modified for this particular implementation, with accurate input and output dimensions, number of layers, convolutional filter sizes, and other features.

Different variations of the popular U-Net architecture have been developed, primarily specialized for different tasks. In image segmentation, specifically medical image segmentation, the most performant models beyond the base U-Net architecture have historically included variants of three principle categories: 3D U-Net, enabling 3-dimensional, volumetric segmentation [14]; Attention U-Net, allowing the ability to focus on areas of importance and ignore more unnecessary regions [15]; and Residual U-Net, hybridizing the traditional U-Net model and the ResNet deep learning model and enabling it to overcome difficulty in training [16].

In training deep neural networks such as U-Net, backpropagation is used to update and optimize model parameters in accordance with the dictated loss function. Cross entropy loss, commonly used in classification applications, is used in the original U-Net implementation [12]. Other loss functions have been utilized, such as dice loss which is based on the common dice metric used for segmentation applications and is, therefore, a direct form of loss minimization. Generally, loss functions for image segmentation applications can be classified as distribution-based (such as cross entropy), region-based (such as dice loss), boundary-based (such as boundary loss), or as compound losses which combine multiple, independent loss functions [17]. Generalizations of dice loss and cross entropy-based losses, such as unified focal loss for mitigating errors arising from class imbalance, have proven particularly robust and effecitve in medical image segmentation applications [18].

Choice of evaluation metric is just as critical as choice of loss function in deriving optimal performance from deep learning models. While loss functions adapt the model parameters during training, it is done with the purpose of ultimately optimizing chosen evaluation metrics. These metrics provide a holistic assessment of model performance after training is complete on validation and test data the model was not trained on. In image segmentation, common metrics include simple pixel accuracy (Rand index), precision and recall or the minimization of false positives and false negatives, respectively, the dice coefficient, the intersection-over-union value (IOU or Jaccard Index), and the Matthews correlation coefficient. Other proposed metrics include Otsu's thresholding or the Coye and Grabcut algorithms, with the

latter showing particular strength isolating skin lesions in medical image segmentation applications [19].

A predominant issue in medical image segmentation arises from class imbalance, or the significant, unequal distribution in area between meaningful foreground regions and their background. For example, foreground elements such as organs in automatic organ segmentation tend to be substantially smaller than the entire scan itself, resulting in a skewed distribution that favors background elements [20]. This issue is significantly more prevalent in the realm of oncology where a tumor is usually even smaller than its organ of origin. This can be described as input imbalance whereas difficulty handling class imbalance resulting in classification errors arising during inference can be described as output imbalance [17]. Such errors can include false positives and false negatives, respectively described as background pixels incorrectly included with the foreground region and foreground pixels incorrectly categorized with the background region and excluded from the foreground. Both errors are incredibly important in medical image segmentation as too many false positive pixels could significantly increase the target region's area, leading to larger radiation fields or surgical margins, and too many false negatives could decrease the target region's area, leading to inadequate radiation delivery or incomplete surgical excision.

3 Methodology

Through this experiment, image segmentation methods were applied to the BraTS2020 dataset of MRI scan image files to determine a baseline viability of this process as well as improve upon that baseline model, to the extent allowed by the availability of time and resources, to find the most successful, performant model possible and that yielded the best results.

3.1 BraTS2020 Dataset

This experiment utilized the BraTS2020 dataset, comprised of clinically-acquired pre-operative multimodal MRI scans of glioblastoma and lower grade glioma with pathologically confirmed diagnoses. All scan instances were available as NIfTI files (.nii.gz) in five treatment variations: 1. native (T1) and 2. post-contrast T1-weighted (T1Gd), 3. T2-weighted (T2), and 4. T2 Fluid Attenuated Inversion Recovery (T2-FLAIR) volumes, all of which were acquired from multiple institutions, with the fifth variation being a manually-derived segmentation. Each segmentation was a composite comprised of three partitions, annotations denoting the GD-enhancing tumor (ET — label 4), the peritumoral edema (ED — label 2), and the necrotic and non-enhancing tumor core (NCR/NET — label 1) [21]. From the 369 scans provided, 250 were utilized for training, 74 for validation, and 45 were reserved as test scans for model evaluation. Each scan file had a native image resolution of 240×240 pixels, with 155 voxels or slices to be examined. Due to variation in the first and last five slices in some of the scan files, processing was limited to voxels 5-150, or the middle 145 slices, of each scan file. Preparing each slice for input results in input dimensionality of $128 \times 128 \times 2$.

3.2 Deep Learning Model

Given its reputation for high performance in image segmentation, the traditional U-Net architecture was implemented as a baseline to assess performance of subsequent model variants. Upon establishing baseline metric values, the baseline model was modified, with various model features applied systematically to determine their respective impact on model performance with the ultimate goal of achieving the highest model metrics with the most accurate inferences. Such model variants were generated by alternating the activation function that determines each node's output between ReLU and PReLU, the latter of which has shown promise in other medical image segmentation applications [22], adding various numbers of dropout layers and passing a range of values to determine the proportion of nodes to drop, and including or excluding normalization layers between convolutional layers.

3.3 Loss Functions

Motivated by the original U-Net implementation, the initial choice of loss function was the distribution-based categorical cross entropy loss [12]. This loss function is ideal for its ease of implementation and optimization but might have a potentially deleterious impact on performance with significantly imbalanced data. Thus, it constituted a sufficient baseline to measure other, potentially more effective, loss functions against. Had time allowed, other such loss functions to implement might have included unified focal loss which generalizes dice loss and cross entropy loss to mitigate issues that arise due to class imbalance, a loss function that has demonstrated to be quite effective in medical image segmentation applications

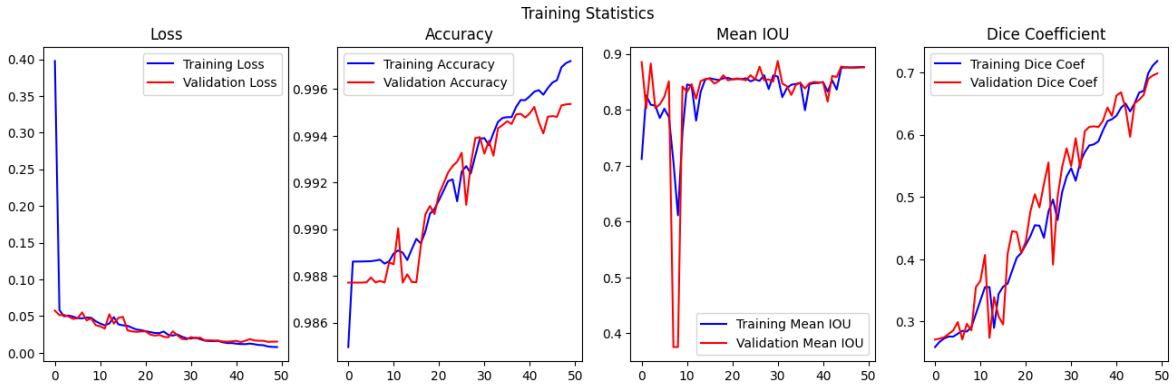

Figure 2: Most performant model loss, accuracy, mean IOU, and dice coefficient metrics from training and validation.

[18]. However, due to time constraints, computational demands of training, and hardware limitations, the ability to implement other loss functions beyond categorical cross entropy were out of reach.

3.4 Evaluation Metrics

Opting, instead, to examine the impact of model architecture variants and various loss functions in lieu of various evaluation metrics, hypothesizing they would have a greater impact on overall model performance, traditional metrics were chosen to evaluate the model throughout each trial. Accuracy, mean IOU, and dice coefficient (equations 1, 2, and 3, respectively) were all provided to the model for consideration as it optimized its parameters using its specified loss function. These metrics were held as constant throughout the entirety of experimentation despite adjusting other features and model parameters.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$meanIOU = \frac{1}{n} \times \frac{TP}{TP + FP + FN} \tag{2}$$

$$dice = \frac{2 \times TP}{(2 \times TP) + FP + FN} \tag{3}$$

3.5 Implementation

Due to this experiment's computational demands and limitations of local hardware availability, the processing of this experiment was conducted using GPU hardware acceleration available through Google's online python coding environment, Colab. This allowed access to remote hardware resources and exponentially faster processing. Each training epoch on local, non-GPU-accelerated hardware took between 5 and 6 hours whereas offloading training and validation onto a remote GPU resource shortened that time to approximately 210 seconds per epoch, allowing the completion of a full, 50-epoch training session to be completed in just under 3 hours. The use of such resources is highly recommended if reprocessing is to be attempted as the computational demand of training this model and its variants is incredibly high.

4 Experiment & Results

Comparing the results of the different model variants generated and tested, the outcomes were notably unremarkable. None of the predetermined modifications majorly improved or decreased performance of the model. That is, model performance was relatively consistent regardless of the activation function chosen, regardless of the inclusion or exclusion of dropout layers, the number of dropout layers included, or the coefficient for the dropout proportion selected, and regardless of the inclusion of normalization layers. All final pixel accuracy metric values came within 0.5% of the simple, traditional U-Net architecture implementation, utilizing cross-entropy loss, that yielded the best results in terms of pixel accuracy with an impressive 99.42% on the test data. These results reaffirm the performance of the U-Net architecture and suggest the need for further exploration regarding model-only optimization. That is to say, if one seeks to derive major improvement from a model alone, an entirely different model architecture may be more likely to provide more significant results than slightly augmenting or minorly

U-Net Implementation Statistics

	Loss	Accuracy	Mean IOU	Dice Coefficient
Training	0.78%	99.72%	87.66%	71.80%
Validation	1.53%	99.54%	87.60%	69.81%
Test	1.78%	99.42%	82.00%	66.15%

Table 1: Resulting evaluation metrics after training, validation, and testing of standard U-Net implementation.

modifying features in one's current model architecture. Further, these results suggest substituting loss function or implementing some other more major changes to the experiment structure rather than lesser modifications to one's current model architecture might yield a more significant impact on model performance.

For the sake of completing this case study though, upon finding no improvement after the attempted feature modifications beyond the standard U-Net implementation, that baseline model was assumed for the remainder of the procedure. The statistics generated in its training, validation, and testing were logged and graphed. Examining these statistics from its training run, epoch-to-epoch, as depicted in Figure 2, U-Net displayed its strengths, with performance in each metric steadily moving in the desired direction. Aside from some more drastic variance in the Mean IOU values, loss, accuracy, and dice coefficient values trend fairly consistently. The final, resultant numeric values of each metric are noted in Table 1.

Out of context, such high pixel accuracy scores fail to convey the significance of the seemingly negligible 0.58% error remaining. However, when an inference is made and a mask is generated and compared to the manually segmented ground truth mask, differences become clear with even that degree of error. Such can be seen in Figure 3.

While the derived mask is primarily true to form, false positive and false negative pixels, pixels incorrectly included with the foreground or background, respectively, become noticeable. In certain areas, the mask stretches beyond its border, but in others, it atrophies. It is true that the general shape is true to form and primarily accurate. However, that 0.58% error remaining comes into stark relief at the margins and boundaries.

Despite the general success and capability of this technology, this insufficiency only yields further reason to continue this work. There is only room but to improve. With technology as capable as this and with countless avenues yet unexplored, including many potential optimization tactics named within this work itself, there is a path forward and progress to make, for science, for medicine, for doctors, and for patients the world over.

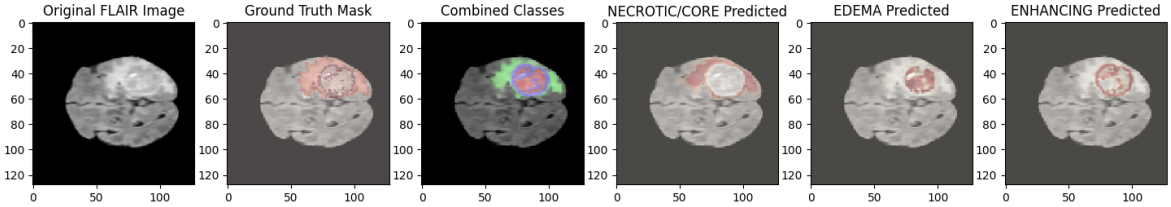

Figure 3: Original FLAIR image and manually segmented mask compared to class-by-class generated masks and combined composite mask.

References

1. Pal, N. R. & Pal, S. K. A review on image segmentation techniques. *Pattern Recognition* **26,** 1277–1294. ISSN: 0031-3203 (1993).
2. Saldanha, J. A. *Medical Image Segmentation and Its Real-World Applications: UNet and Beyond* 2023. https://medium.com/@jervisaldanha/medical-image-segmentation-and-its-real-world-applications-unet-and-beyond-9cd06eeebcb6 (2024).
3. Rundo, L. *et al.* Combining split-and-merge and multi-seed region growing algorithms for uterine fibroid segmentation in MRgFUS treatments. *Medical & biological engineering & computing* **54,** 1071–1084 (2016).
4. Chen, X. & Pan, L. A Survey of Graph Cuts/Graph Search Based Medical Image Segmentation. *IEEE Reviews in Biomedical Engineering* **11,** 112–124 (2018).
5. Khadidos, A., Sanchez, V. & Li, C.-T. Weighted Level Set Evolution Based on Local Edge Features for Medical Image Segmentation. *IEEE Transactions on Image Processing* **26,** 1979–1991 (2017).
6. Rundo, L. *et al.* A survey on nature-inspired medical image analysis: a step further in biomedical data integration. *Fundamenta Informaticae* **171,** 345–365 (2020).
7. Wang, S. & Summers, R. M. Machine learning and radiology. *Medical Image Analysis* **16,** 933–951. ISSN: 1361-8415 (2012).
8. Ren, T. *et al.* Study on the improved fuzzy clustering algorithm and its application in brain image segmentation. *Applied Soft Computing* **81,** 105503. ISSN: 1568-4946 (2019).
9. Ker, J., Wang, L., Rao, J. & Lim, T. Deep Learning Applications in Medical Image Analysis. *IEEE Access* **6,** 9375–9389 (2018).
10. Rueckert, D. & Schnabel, J. A. Model-Based and Data-Driven Strategies in Medical Image Computing. *Proceedings of the IEEE* **108,** 110–124 (2020).
11. Castiglioni, I. *et al.* AI applications to medical images: From machine learning to deep learning. *Physica medica* **83,** 9–24 (2021).
12. Ronneberger, O., Fischer, P. & Brox, T. in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* 234–241 (Springer International Publishing, 2015). ISBN: 9783319245744.
13. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *ImageNet Classification with Deep Convolutional Neural Networks* in *Advances in Neural Information Processing Systems* (eds Pereira, F., Burges, C., Bottou, L. & Weinberger, K.) **25** (Curran Associates, Inc., 2012).
14. Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T. & Ronneberger, O. *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation* in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (eds Ourselin, S., Joskowicz, L., Sabuncu, M. R., Unal, G. & Wells, W.) (Springer International Publishing, Cham, 2016), 424–432. ISBN: 978-3-319-46723-8.
15. Oktay, O. *et al. Attention U-Net: Learning Where to Look for the Pancreas* 2018. arXiv: 1804.03999 [cs.CV].
16. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *CoRR* **abs/1512.03385.** arXiv: 1512.03385 (2015).
17. Taghanaki, S. A. *et al.* Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized Medical Imaging and Graphics* **75,** 24–33 (2019).
18. Yeung, M., Sala, E., Schönlieb, C.-B. & Rundo, L. Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics* **95,** 102026. ISSN: 0895-6111 (2022).
19. Setiawan, A. W. *Image Segmentation Metrics in Skin Lesion: Accuracy, Sensitivity, Specificity, Dice Coefficient, Jaccard Index, and Matthews Correlation Coefficient* in *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)* (2020), 97–102.
20. Roth, H. R. *et al. DeepOrgan: Multi-level Deep Convolutional Networks for Automated Pancreas Segmentation* in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (eds Navab, N., Hornegger, J., Wells, W. M. & Frangi, A.) (Springer International Publishing, Cham, 2015), 556–564. ISBN: 978-3-319-24553-9.

21. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging* **34,** 1993–2024 (2015).

22. Rizwan I Haque, I. & Neubert, J. Deep learning approaches to biomedical image segmentation. *Informatics in Medicine Unlocked* **18,** 100297. ISSN: 2352-9148 (2020).