BrainSegNet: DCGAN-Driven Brain Tumor Classification and Segmentation - Final Report

Abdulaziz Houtari, Deeksha Mohanty, Avi Lochab, Samir Shakir, Seyeon Park
Department of Computer Science and Engineering, Michigan State University
houtaria@msu.edu, mohant11@msu.edu, lochabav@msu.edu, shairsk@msu.edu, seyeon@msu.edu

Abstract

We present the progress of the implementation of our BrainSegNet, a framework for brain tumor classification and segmentation leveraging DCGANs for data augmentation, U-Net as a baseline segmentation model, and CLIP-enhanced U-Net and ViT models for improved performance. Thus far, we have implemented baseline models, trained on the BraTS 2020 dataset, and achieved promising results for segmentation using Dice coefficient and IoU metrics. Challenges, such as class imbalance and computational constraints, are being addressed, with planned refinements including CRF post-processing, gradient-based boundary refinement, and further hyperparameter tuning.

1. Introduction

Brain tumor segmentation is a critical step in medical imaging, enabling accurate diagnosis and treatment planning. However, challenges such as limited annotated data and class imbalance make it difficult to train robust models for this task. To address these challenges, we have focused on implementing and evaluating existing state-of-the-art models, including U-Net, CLIP-enhanced U-Net, Vision Transformers (ViT), and SegNet, using the BraTS 2020 dataset.

Our goal is to benchmark these architectures for segmentation accuracy and classification performance while exploring complementary techniques such as DCGAN-based data augmentation and Conditional Random Field (CRF) post-processing. By incorporating diverse preprocessing methods and evaluation metrics like Dice coefficient and IoU, we aim to understand the strengths and limitations of these models. Our progress so far demonstrates the effectiveness of baseline approaches while identifying areas for improvement, such as refining boundary precision, addressing class imbalance, and optimizing computational efficiency.

2. Experiments

We used PyTorch [8] for implementing and experimenting our models.

2.1. Segmentation

For all segmentation models, we used a weighted combined loss function which incorporated **Dice Loss** and **BCEWithLogitsLoss**. Due to computational limitations, training utilized different batch sizes for each model based on available resources at the time. For the most part, we used a batch size of 64 and trained our models for 5-20 epochs.

We used the Dice coefficient as the primary metric to evaluate the model's performance, as it measures the overlap between predicted and ground truth segmentation masks. The Dice coefficient is particularly suitable for medical image segmentation tasks, where it effectively captures both false positives and false negatives.

The experiments utilized the BraTS dataset [7] [3] [1] [2] [4], a widely used benchmark for brain tumor segmentation. The dataset contains MRI volumes, each consisting of 155 slices. For this model, each slice was treated as an individual sample and fed into the model separately. We focused on the T1CE and FLAIR modalities for input from the dataset.

Preprocessing steps included normalizing each image channel to zero mean and unit variance for consistent input scaling. All images and masks were resized to 160×160 to standardize input dimensions across the dataset. Finally, the dataset was split into training (80%) and test (20%) subsets.

2.1.1 Baseline Segmentation with U-Net

As a baseline, we implemented a U-Net architecture for the segmentation task [10]. The U-Net model is an encoder-decoder architecture with skip connections that effectively combine spatial and semantic information. The encoder extracts multi-scale features through successive down-sampling, while the decoder reconstructs these fea-

tures to produce pixel-wise segmentation masks via upsampling. Skip connections directly pass features from the encoder to the decoder at corresponding levels, preserving fine-grained spatial details that might otherwise be lost.

2.1.2 SegNet

To further explore segmentation models, we implemented SegNet, a fully convolutional encoder-decoder architecture originally designed for semantic segmentation tasks. SegNet leverages pooling indices during the decoding stage, enabling efficient reconstruction of high-resolution feature maps. This makes it suitable for brain tumor segmentation, where precise boundary delineation is critical.

The SegNet architecture consists of an encoder-decoder framework:

1. **Encoder:** The encoder comprises convolutional layers followed by max-pooling operations. Pooling indices are saved during down-sampling to guide the decoder during up-sampling. The encoder progressively extracts high-level spatial features while reducing resolution.

2. **Decoder:** The decoder reconstructs the input spatial resolution using max-unpooling operations guided by pooling indices from the encoder. It employs convolutional layers to refine the upsampled feature maps.

3. **Output:** The final output is a pixel-wise segmentation map with raw logits corresponding to the three target classes.

2.1.3 CLIP-Based Encoder with U-Net Decoder Implementation

We integrated a CLIP-based ViT-B/32 encoder [9] to leverage large-scale pretrained features for MRI brain tumor segmentation. CLIP's first convolutional layer was adapted to accept two MRI modalities (T1CE and FLAIR) by averaging and replicating the pretrained RGB weights.

1. **Encoder:** We retained the class token, entire transformer stack, and dynamically resized positional embeddings to match the new input resolution. After processing, the class token was removed, and the remaining embeddings were reshaped into a spatial feature map enriched with pretrained semantic context.

2. **Decoder:** A U-Net-inspired decoder without skip connections was used, as CLIP does not produce intermediate feature maps. A series of transpose and standard convolutions with ReLU reconstructed a high-resolution segmentation mask, which was then interpolated to the desired output size.

We trained the model using AdamW at 1×10^{-4} with limited batches per epoch and restricted validation due to computational constraints. Despite these limitations, the CLIP-enhanced encoder and streamlined decoder showed promising potential in harnessing pretrained contextual features for brain tumor segmentation.

2.1.4 DeepLabV3

DeepLabV3 [5] employs atrous (dilated) convolutions and Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale context without repeatedly downsampling. This approach inherently balances local detail with global semantics, making it well-suited for segmenting irregular tumor regions in MRI scans. The model consists of the following:

- **Backbone**: A pre-trained ResNet-50 backbone extracts feature maps at various resolutions.

- **ASPP** Module: Parallel atrous convolutions at different dilation rates aggregate multi-scale features.

2.2. DCGAN for Tumor Image Generation and Segmentation

To further enhance our experimentation, we implemented a Deep Convolutional Generative Adversarial Network (DCGAN) for generating synthetic MRI brain tumor images. The goal was to augment the dataset with realistic tumor images and explore how these synthetic samples can aid in segmentation tasks.

2.2.1 Model Architecture

The DCGAN comprises two primary components:

- **Generator**: The generator network maps a 100-dimensional latent vector sampled from a uniform distribution to a $256 \times 256 \times 3$ synthetic MRI image. The architecture includes:
  - Dense layers followed by reshaping to an initial spatial resolution.
  - Multiple transpose convolution layers with ReLU activation, progressively upsampling to the target image size.
  - A final transpose convolution layer with a `tanh` activation for output normalization.

- **Discriminator**: The discriminator evaluates whether an input image is real or generated. The architecture consists of:
  - Convolutional layers with Leaky ReLU activations for feature extraction.
  - Dropout layers to prevent overfitting.

– A fully connected layer with sigmoid activation for binary classification (real or fake).

2.2.2 Training Methodology

- The DCGAN was trained using adversarial loss, where the generator attempts to maximize the discriminator's error, and the discriminator aims to minimize classification error. This interplay ensures realistic image synthesis.
- Training data was preprocessed by normalizing pixel intensities to the range $[-1, 1]$ and resizing images to 256×256.
- We used the Adam optimizer for both the generator and discriminator, with learning rates of 0.0002 and β_1 set to 0.5.

2.2.3 Experiments and Observations

- We generated synthetic images resembling the tumor and non-tumor regions in the BraTS dataset. Visual inspection confirmed that the generator effectively captured global structures but struggled with fine details in complex regions.

2.3. Classification with Vision Transformer (ViT)

We also implemented a Vision Transformer (ViT)-based model to classify brain tumor images into four categories.

2.3.1 Model Architecture

- **Patch Embedding**: Instead of directly processing the entire image, ViT divides the input image into fixed-size patches, flattens them, and linearly projects them into embeddings. These embeddings, along with positional encodings to retain spatial information, form the input sequence for the Transformer.
- **Transformer Encoder**: The core architecture uses a stack of Transformer encoder layers. Each layer comprises Multi-Head Self-Attention (to capture relationships between patches) and Feed-Forward Neural Networks, with Layer Normalization and Residual Connections to enhance stability and learning efficiency.
- **Classification Head**: A special learnable token (class token) is prepended to the sequence of patch embeddings. After processing through the Transformer, the output of this token represents the overall image and is fed into a classification head, typically a fully connected layer, for predictions.

2.3.2 Training and Inference

The training setup involves optimizing the model using cross-entropy loss, a standard choice for multi-class classification tasks, and the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01. The training is conducted with a batch size of 32 for both training and validation datasets and spans 1000 epochs, during which the best-performing model is saved based on validation accuracy.

During inference, the trained model predicts individual test images by outputting the predicted class, the associated confidence score, and class-wise probabilities. These predictions are visualized with the input image alongside a bar chart that showcases the prediction probabilities for all classes, allowing for an intuitive interpretation of the results.

3. Results and Discussion

3.1. Segmentation Results and Comparisons

3.1.1 Baseline U-NET

We evaluated the baseline U-Net model on the BraTS dataset using the Dice coefficient as the primary metric to assess segmentation accuracy. Over the course of 50 epochs and approximately 2 hours of training on a T4 NVIDIA GPU, the model achieved a final validation Dice score of **0.6497**, demonstrating its ability to effectively segment tumor regions. Comparison to GT and DeepLabV3 can be seen in Figure 4.

3.1.2 SegNet

Figure 1 provides a qualitative visualization of the segmentation results. The predicted segmentation closely aligns with the ground truth segmentation mask, successfully identifying tumor regions. While larger tumor regions were segmented with high precision, the model struggled to correctly classify the tumor type when more than one category was present. Figure 2 provides another example where this model struggles when all 3 classes of tumors are present.

The model was trained only for 6 epochs because of the limited computational resources. This might explain its limited capability to distinguish between all the classes.

3.1.3 CLIP encoder-enhanced U-Net

Our CLIP-integrated BrainSegNet model achieved a validation Dice score of approximately 0.7866 after just 5 epochs, suggesting that the pretrained CLIP encoder provides valuable semantic features for MRI segmentation. As shown in Figure 3, the model accurately identified large tumor regions and maintained coherent shapes, demonstrating the benefits of global context from CLIP.

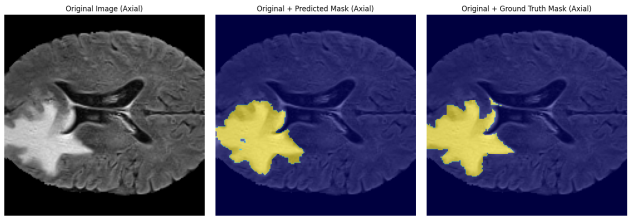

Figure 1. SegNet Segmentation Results: Comparison of the original MRI image, ground truth segmentation, and predicted segmentation.

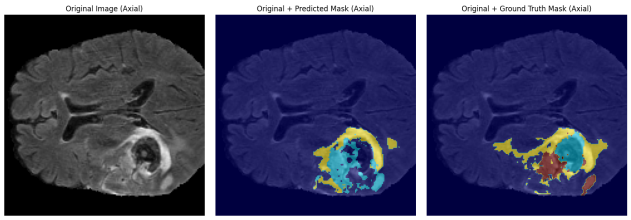

Figure 2. SegNet Segmentation Results: Comparison of the original MRI image, ground truth segmentation, and predicted segmentation.

Figure 3. CLIP Encoder-Based U-Net Segmentation Results: Comparison of the original MRI image, ground truth segmentation, and predicted segmentation.

Nevertheless, boundary precision and the segmentation of smaller lesions remain challenging. Occasional oversegmentation and missed details indicate room for refinement and possible regularization. Training time and computational limitations restricted us to a few epochs and reduced batches per epoch. With longer training, hyperparameter tuning, and domain-specific fine-tuning of CLIP, it would achieve better segmentation performance.

3.1.4 DeepLabV3

This model outperformed the baseline U-NET by a small amount achieving a validation dice score of **0.7007** which is roughly a 5% increase demonstrating a decent ability in this task. Comparison to GT and U-NET can be seen in Figure 4.

Table 1. Comparison of Validation Dice Values Across Models

Model	Dice Value (Validation)
U-NET	0.6497
SegNet	0.1123
BrainSegNet	0.7866
DeepLabV3	0.7007

3.1.5 Discussion and Comparison

While the U-Net performed well as a baseline, certain challenges and limitations were observed during the experiments which suggest opportunities for improvement:

1. **Computational Intensity**: The computational demands of the U-Net architecture required constraining the training process to 64 batches per epoch with a batch size of 64, while validation and testing were limited to 16 batches per epoch. These constraints may have hindered the model's ability to fully leverage the available data, potentially affecting generalization.

2. **Slice-Based Input**: The dataset was structured such that each MRI volume was divided into 155 slices, which were treated as individual samples. While this approach simplifies the input format for the U-Net, it may not have been optimal. Many slices contain little to no tumor-related information, which could dilute the learning signal and reduce the model's focus on informative slices. Incorporating methods to prioritize slices with higher tumor content or processing volumes in 3D could improve performance.

These findings validate the U-Net as a strong baseline but underscore the need for architectural refinements, improved data handling strategies, and enhanced training protocols to achieve state-of-the-art performance in brain tumor segmentation. Future work will explore incorporating 3D processing for volumetric inputs, attention-based mechanisms, and methods to address class imbalance for more robust and accurate segmentation.

3.2. Result for Vision Transformer (ViT)

We implemented a Vision Transformer (ViT) model to classify brain MRI scans into four tumor categories: pituitary, glioma, meningioma, and no tumor. The model achieved near-perfect training accuracy of 100%; however, the validation accuracy fluctuated significantly before stabilizing around 90% after 50 epochs, indicating potential overfitting. During testing, the model demonstrated its capability to predict tumor types with confidence scores for each class, highlighting its ability to distinguish among tumor categories effectively. The results were visualized using confidence bar charts, emphasizing the model's

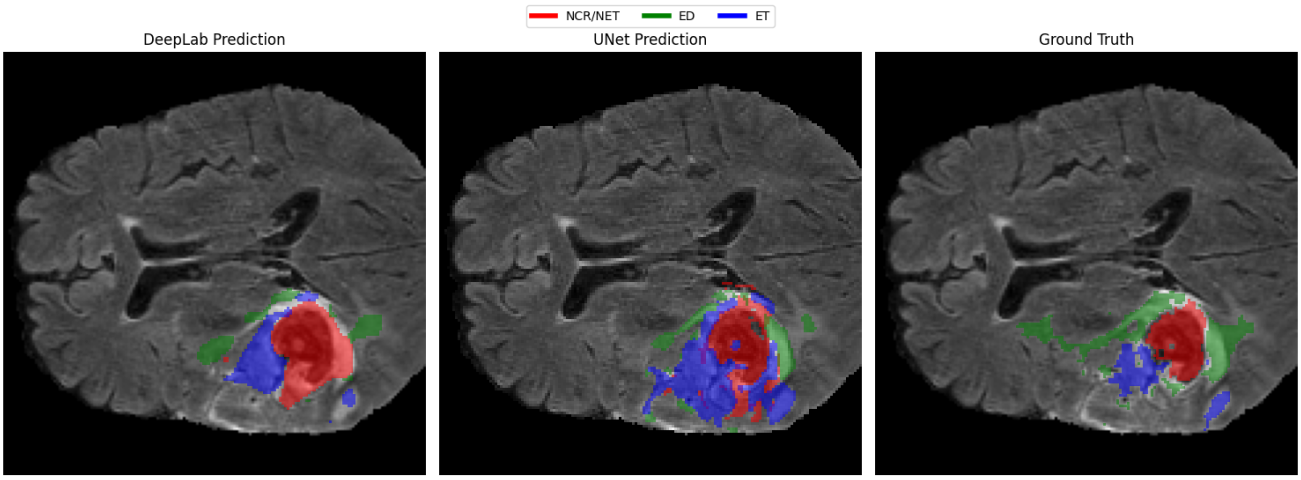

Figure 4. Comparison of the, ground truth segmentation, and predicted segmentation for U-NET and DeepLabV3.

strengths in identifying the most probable class while offering insights into alternative predictions. This feature underscores the model's potential utility in supporting clinical decision-making. To further enhance performance, future work could focus on mitigating overfitting through advanced data augmentation strategies, regularization techniques, or ensemble approaches, ultimately improving the robustness and generalizability of the model's predictions.

3.3. Web Interface

We created a simple website using **NextJS** and a small backend using **FastAPI** to generate 3D GIFs of the brain tumor regions given T1CE and FLAIR images.

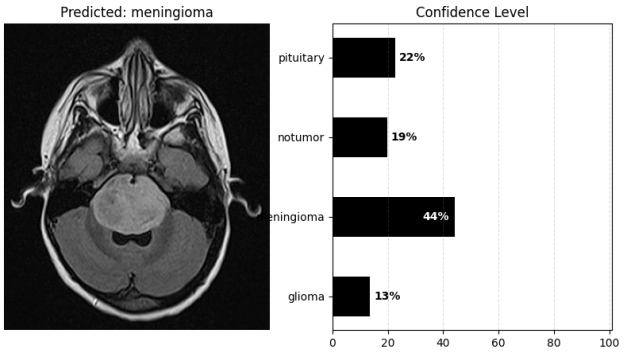

Figure 5. ViT Classification

4. Final Remarks & Challenges

This project achieved significant progress despite several computational challenges. Limited GPU resources on Google Colab frequently caused disconnections and loss of progress, disrupting training workflows. Additionally, long training times, averaging 25 minutes per epoch, constrained our ability to run all models under consistent conditions. Some models were trained for more epochs than others, affecting the comparability of results. Addressing these limitations in future work, through access to more reliable and powerful computational resources, would enhance consistency, streamline experimentation, and allow for deeper model evaluation.

Google Drive link with all project files: https://drive.google.com/drive/u/1/folders/1Js_updyjydAYJVjmC3a4cYUBbF6A1aLu

5. Distribution of Tasks

- Deeksha: Implemented a CLIP-enhanced U-Net model for brain tumor segmentation, including preprocessing the BraTS 2020 dataset, adapting CLIP's encoder for two-channel MRI inputs, and training the model. Also evaluated performance using Dice metrics, experimented with loss functions, and conducted hyperparameter tuning.

- Samir: I have implemented the SegNet as an alternative segmentation model on the BraTs2020 dataset. I also worked on pre-processing the dataset and visualized segmentation results with the SegNet model.

- Seyeon: Integrated a Vision Transformer (ViT)-based U-Net for the brain tumor segmentation task, trained and evaluated the model on the BraTS dataset, and compared its performance with the baseline U-Net. Analyzed improvements and documented insights.

- Abdulaziz: Implemented a baseline U-NET and

DeepLabV3. Developed a website to showcase the results of the model allowing users to upload their own MRI images and return the model's output.

- Avi: Developed a DCGAN-based approach for data augmentation and assisted in generating evaluation metrics (Dice). Documented insights from the augmented data experiments and contributed to overall performance analysis.

References

[1] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, et al. Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection, 2017. 1

[2] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, et al. Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection, 2017. 1

[3] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, et al. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Nature Scientific Data*, 4:170117, 2017. 1

[4] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge, 2018. 1

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 2

[6] G. Dharani Devi, S. Sandra Doss, S. Sanjitha, and N. Sai Chaithanya. Gcnn-based combined denoising and classification for improved mri brain tumor identification. In *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 401–412, Bengaluru, India, 2023. IEEE.

[7] 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. 1

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 1

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1

[11] K. Sailunaz, D. Bestepe, S. Alhajj, T. Özyer, J. Rokne, and R. Alhajj. Brain tumor detection and segmentation: Interactive framework with a visual interface and feedback facility for dynamically improved accuracy and trust. *PLOS ONE*, 18(4), Apr. 2023.

[12] M. Umamaheswari, J. Sivadasan, R. K. Dwibedi, B. Senthilkumar, L. P. Rani, and S. Oviya. Classification of brain tumor using generative adversarial network with res net discriminator. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 01–07, Chennai, India, 2024. IEEE.