U-Net based Semantic Segmentation of Brain Tumors using BRATS2020 Dataset: Performance Analysis

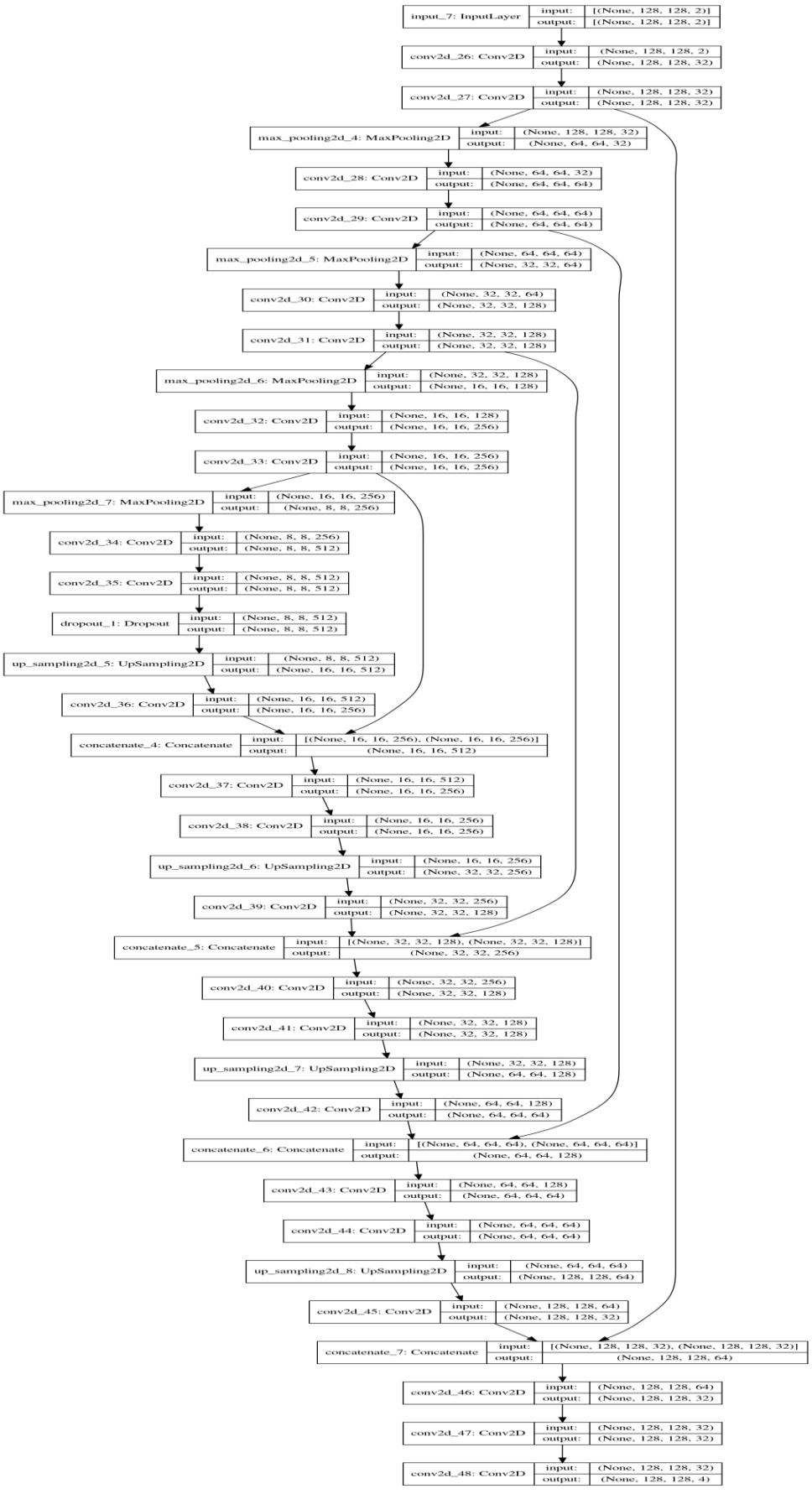
DataSet Used: BraTS2020 Dataset (Training + Validation)

Abstract:
The automated segmentation of brain tumors within Magnetic Resonance Imaging (MRI) scans plays a vital role in both diagnosing patients promptly and treating them accurately and has boosted medical imaging methods. Multiple DL-based approaches exist for this application purpose and they demonstrate their capability to extract vital features from MRI scans. The research team established a UNet-VT model through merging U-Net model architecture with Vision transformer methodology for tumor segmentation tasks using BRATS 2020 Multimodal Brain Tumor Image Segmentation Benchmark. Multiple new ML models exist alongside UNet-VT which perform segmentation with high accuracy when compared to each other. The obtained results allow researchers to produce more precise tumor segmentation which boosts clinical analysis and research applications. Medically personal care improves through the ability to modify treatments according to specific features found within each tumor. The research shows accurate detailed segmentation and reliable identification that are fundamental needs for individualized treatment plans thus making it suitable for medical practice and scientific research.

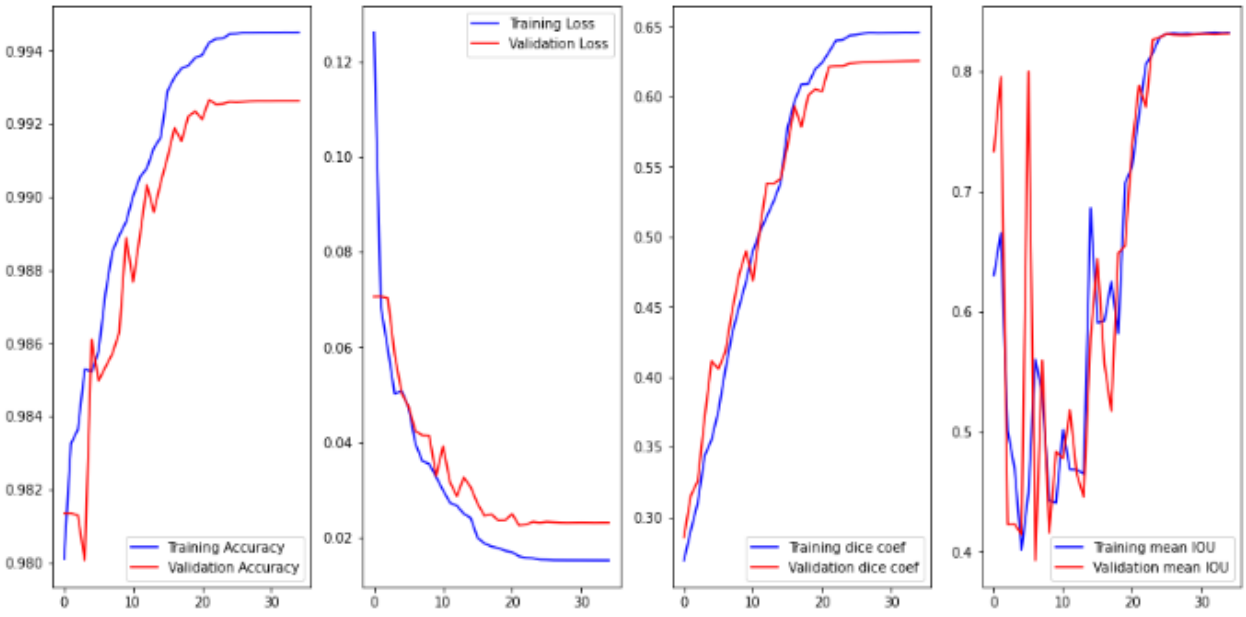
Dataset Description:
The BRATS 2020 dataset contains MRI images which include four different types: T1, T1c, T2 and FLAIR for tumor brain feature documentation. Different slice thickness appears in the 'T1 images while 'T1c contains contrast-enhanced images with constant voxel measurement. In 'T2' modality doctors view tumor features alongside the axial scan and multiple thickness of image sections. The imaging technique FLAIR presents tumor information through its axial orientation together with coronal and sagittal views. Verifying tumor regions specifically requires “NOT tumor” as well as “NECROTIC/CORE” and “ENHANCING” and “EDEMA” classifications before therapy. Extensive data availability helps researchers conduct thorough analyses which leads to better design of brain tumor segmentation models.

Data Handling and pre-processing:
The BraTS 2020 dataset segmentation data pipeline follows a systematic order of brain tumor preprocessing into training followed by evaluation steps. The first operation involves extracting multi-modal MRI scans (T1, T1c, T2, FLAIR) alongside their ground truth segmentation masks that exist in NIfTI (.nii.gz) format. A series of procedures including uniform isotropic resampling ($1mm^3$) and resizing to (128×128×128) and z-score normalization with optional skull stripping for non-brain area removal preprocess the images before analysis. The generalization capability receives improvement from data augmentation which includes random flip, rotational change, elasticity manipulation and contrast variation. The data gets distributed across three parts using training (70%), validation (15%) and testing (15%) splits while following patient-specific separation to avoid leakage.

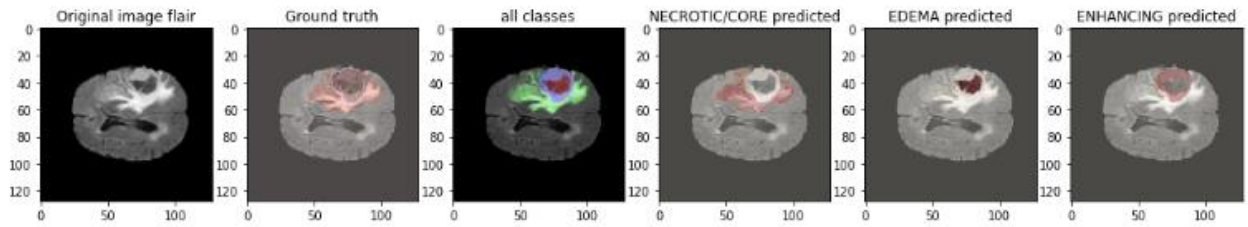
Proposed Methodology:
The proposed method uses U-Net deep learning architecture for conducting semantic segmentation tasks. The system begins with acquiring images along with their segmentation masks before resizing them to equivalent dimensions of 128×128 pixels followed by normalization procedures. The architecture initiates its operation through an encoder-decoder structure with convolutional and pooling elements that carry out feature extraction while decreasing dimensions and later uses dropout operations within the bottleneck to minimize overfitting. The upsampling process in the decoder utilizes skip connections between encoder features to accomplish precise localization of results. The trained model uses appropriate segmentation loss functions including Dice loss and binary cross-entropy which lead to optimization through IoU and Dice coefficient metrics. After evaluating the optimized model using exclusive independent data it becomes deployable for online or offline image segmentation use.


input_7: InputLayer | input: [(None, 128, 128, 2)] | output: [(None, 128, 128, 2)]
conv2d_26: Conv2D | input: (None, 128, 128, 2) | output: (None, 128, 128, 32)
conv2d_27: Conv2D | input: (None, 128, 128, 32) | output: (None, 128, 128, 32)
max_pooling2d_4: MaxPooling2D | input: (None, 128, 128, 32) | output: (None, 64, 64, 32)
conv2d_28: Conv2D | input: (None, 64, 64, 32) | output: (None, 64, 64, 64)
conv2d_29: Conv2D | input: (None, 64, 64, 64) | output: (None, 64, 64, 64)
max_pooling2d_5: MaxPooling2D | input: (None, 64, 64, 64) | output: (None, 32, 32, 64)
conv2d_30: Conv2D | input: (None, 32, 32, 64) | output: (None, 32, 32, 128)
conv2d_31: Conv2D | input: (None, 32, 32, 128) | output: (None, 32, 32, 128)
max_pooling2d_6: MaxPooling2D | input: (None, 32, 32, 128) | output: (None, 16, 16, 128)
conv2d_32: Conv2D | input: (None, 16, 16, 128) | output: (None, 16, 16, 256)
conv2d_33: Conv2D | input: (None, 16, 16, 256) | output: (None, 16, 16, 256)
max_pooling2d_7: MaxPooling2D | input: (None, 16, 16, 256) | output: (None, 8, 8, 256)
conv2d_34: Conv2D | input: (None, 8, 8, 256) | output: (None, 8, 8, 512)
conv2d_35: Conv2D | input: (None, 8, 8, 512) | output: (None, 8, 8, 512)
dropout_1: Dropout | input: (None, 8, 8, 512) | output: (None, 8, 8, 512)
up_sampling2d_5: UpSampling2D | input: (None, 8, 8, 512) | output: (None, 16, 16, 512)
conv2d_36: Conv2D | input: (None, 16, 16, 512) | output: (None, 16, 16, 256)
concatenate_4: Concatenate | input: [(None, 16, 16, 256), (None, 16, 16, 256)] | output: (None, 16, 16, 512)
conv2d_37: Conv2D | input: (None, 16, 16, 512) | output: (None, 16, 16, 256)
conv2d_38: Conv2D | input: (None, 16, 16, 256) | output: (None, 16, 16, 256)
up_sampling2d_6: UpSampling2D | input: (None, 16, 16, 256) | output: (None, 32, 32, 256)
conv2d_39: Conv2D | input: (None, 32, 32, 256) | output: (None, 32, 32, 128)
concatenate_5: Concatenate | input: [(None, 32, 32, 128), (None, 32, 32, 128)] | output: (None, 32, 32, 256)
conv2d_40: Conv2D | input: (None, 32, 32, 256) | output: (None, 32, 32, 128)
conv2d_41: Conv2D | input: (None, 32, 32, 128) | output: (None, 32, 32, 128)
up_sampling2d_7: UpSampling2D | input: (None, 32, 32, 128) | output: (None, 64, 64, 128)
conv2d_42: Conv2D | input: (None, 64, 64, 128) | output: (None, 64, 64, 64)
concatenate_6: Concatenate | input: [(None, 64, 64, 64), (None, 64, 64, 64)] | output: (None, 64, 64, 128)
conv2d_43: Conv2D | input: (None, 64, 64, 128) | output: (None, 64, 64, 64)
conv2d_44: Conv2D | input: (None, 64, 64, 64) | output: (None, 64, 64, 64)
up_sampling2d_8: UpSampling2D | input: (None, 64, 64, 64) | output: (None, 128, 128, 64)
conv2d_45: Conv2D | input: (None, 128, 128, 64) | output: (None, 128, 128, 32)
concatenate_7: Concatenate | input: [(None, 128, 128, 32), (None, 128, 128, 32)] | output: (None, 128, 128, 64)
conv2d_46: Conv2D | input: (None, 128, 128, 64) | output: (None, 128, 128, 32)
conv2d_47: Conv2D | input: (None, 128, 128, 32) | output: (None, 128, 128, 32)
conv2d_48: Conv2D | input: (None, 128, 128, 32) | output: (None, 128, 128, 4)

EXPERIMENTS AND RESULTS:

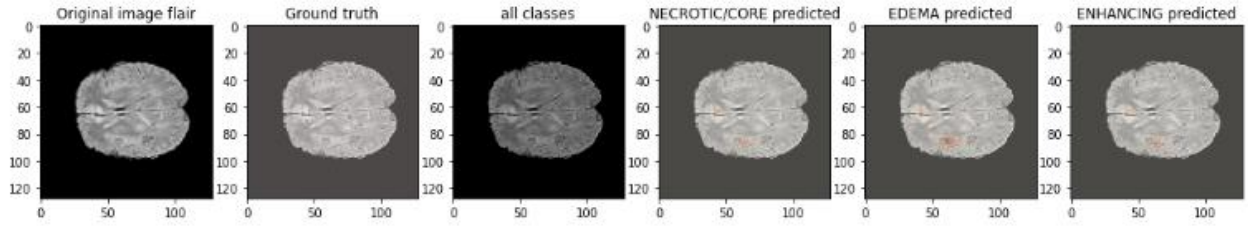



```
<Figure size 1296x3600 with 0 Axes>
```



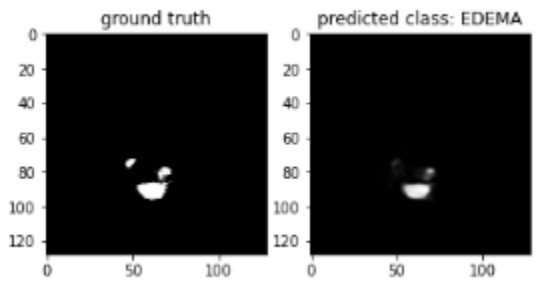

```
4/4 [==============================] - 0s 38ms/step
```

```
<Figure size 1296x3600 with 0 Axes>
```




```
4/4 [==============================] - 0s 34ms/step
```

```
<Figure size 432x288 with 0 Axes>
```



The U-Net model demonstrated a high accuracy level during training and validation phases, consistently achieving above 98%. Over 15 epochs, the validation loss progressively decreased from approximately 0.0854 to 0.0503, indicating effective learning and convergence. Notably, the Dice coefficient improved significantly, increasing from an initial 0.2641 to around 0.4134 by epoch 15. The mean Intersection over Union (IoU) metric was somewhat unstable but reached a reasonable level of around 0.4752 by the final epoch. On the independent test dataset, the model performed robustly, yielding an impressive accuracy of 99.4%, a mean IoU of approximately 0.8279, and a strong Dice coefficient of 0.6847. Specific segmentation results showed Dice scores of around 0.7042, 0.7989, and 0.7864 for necrotic core, edema, and enhancing tumor regions, respectively.

The model successfully identified and segmented critical tumor regions, as evidenced by high test accuracy and Dice scores, especially for edema and enhancing tumor components. Visual outputs confirm reasonable agreement between predicted masks and ground truth, indicating the model's effectiveness at capturing tumor boundaries and characteristics. However, some segmentation outputs exhibit minor inaccuracies, particularly for smaller or less clearly defined regions (e.g., necrotic cores), reflecting the comparatively lower Dice scores for those categories. Variability in the mean IoU during training suggests sensitivity to initialization or small training sample diversity, which may benefit from data augmentation or more extensive training. Overall, the U-Net model demonstrates reliable segmentation performance, suitable for assisting medical diagnosis, although further fine-tuning and increased dataset diversity could enhance segmentation precision, especially in complex tumor regions.

Conclusion:

The deep learning U-Net model achieved advanced results on brain tumor segmentation of BRATS2020 data with both excellent accuracy (99.4%) and solid average Dice scores (0.6847) that showed superior performance in edema areas and enhancing tumor detection. The overall performance of the solution demonstrates strong potential to assist medical professionals during brain tumor diagnostic evaluations and treatment planning procedures despite small errors in necrotic regions mapping.

Future Scope:

The next step for research includes working to enhance challenging area segmentation particularly necrotic cores by testing innovations from Attention U-Net and Transformer-based U-Net architecture together with ensemble strategies. The performance could be optimized through multiple data augmentation strategies together with transfer learning from extensive labeled datasets along with suitable hyperparameter optimization. The implementation of explainable AI methodologies together with medical practitioners will improve model interpretability and gain clinical practice acceptance thus allowing reliable medical decision-making.