Artificial Intelligence

Professor: Dr. Sajedi

Feb 2025

Final Project Report
Brain MRI Segmentation Using U-Net
Anahita Hedayatifard

---

1 Methods and Implementation

1.1 Dataset

- **Source:** The Brain MRI dataset was obtained from Kaggle's "LGG MRI Segmentation" dataset.
- **Preprocessing:**
  - Images were resized to 128x128 pixels.
  - Pixel values were normalized to the range [0, 1].
  - Slicing and cropping were performed to remove extra black background regions, ensuring that only the relevant parts of the MRI images were retained.
- **Dataset Split:** The data was divided into training, validation, and test sets with a ratio of 80:10:10.

1.2 Model Architecture

The U-Net architecture was implemented with the following components:

- **Encoder:** Stacked convolutional layers with 3x3 kernels, followed by max-pooling layers for downsampling.
- **Decoder:** Transposed convolution layers for upsampling, paired with skip connections from the encoder.
- **Output Layer:** A single convolution layer with a sigmoid activation function to produce a binary segmentation mask.
- **Parameters:** Approximately 7,800,000 trainable parameters.

1.3 Loss Function and Metrics

- **Loss Function:** Binary cross-entropy
- **Metrics:**
  - Accuracy
  - Intersection over Union (IoU)
  - Dice Coefficient

- Precision
- Sensitivity
- Specificity

1.4 Training Setup

- **Optimizer:** Adam optimizer with an initial learning rate of 0.001.
- **Batch Size:** 8
- **Epochs:** 50
- **Callbacks:**
  - ReduceLROnPlateau to dynamically adjust the learning rate.
  - ModelCheckpoint to save the best model based on validation loss.

2 Results and Analysis

2.1 Model Performance

The U-Net model achieved the following metrics on the validation set:

- **Accuracy:** 0.9927
- **IoU:** 0.8336
- **Dice Coefficient:** 0.7209
- **Precision:** 0.7801
- **Sensitivity:** 0.7099
- **Specificity:** 0.9976

2.2 Challenges Faced

1. **High Parameter Count**
   - To address this, the input images were resized to smaller dimensions (128×128), significantly reducing the computational cost.
2. **Metric Issues**
   - The TensorFlow `MeanIoU` metric produced misleading results due to non-thresholded outputs. A custom IoU metric with thresholding was implemented for accurate evaluation.
3. **Dimension Mismatch**

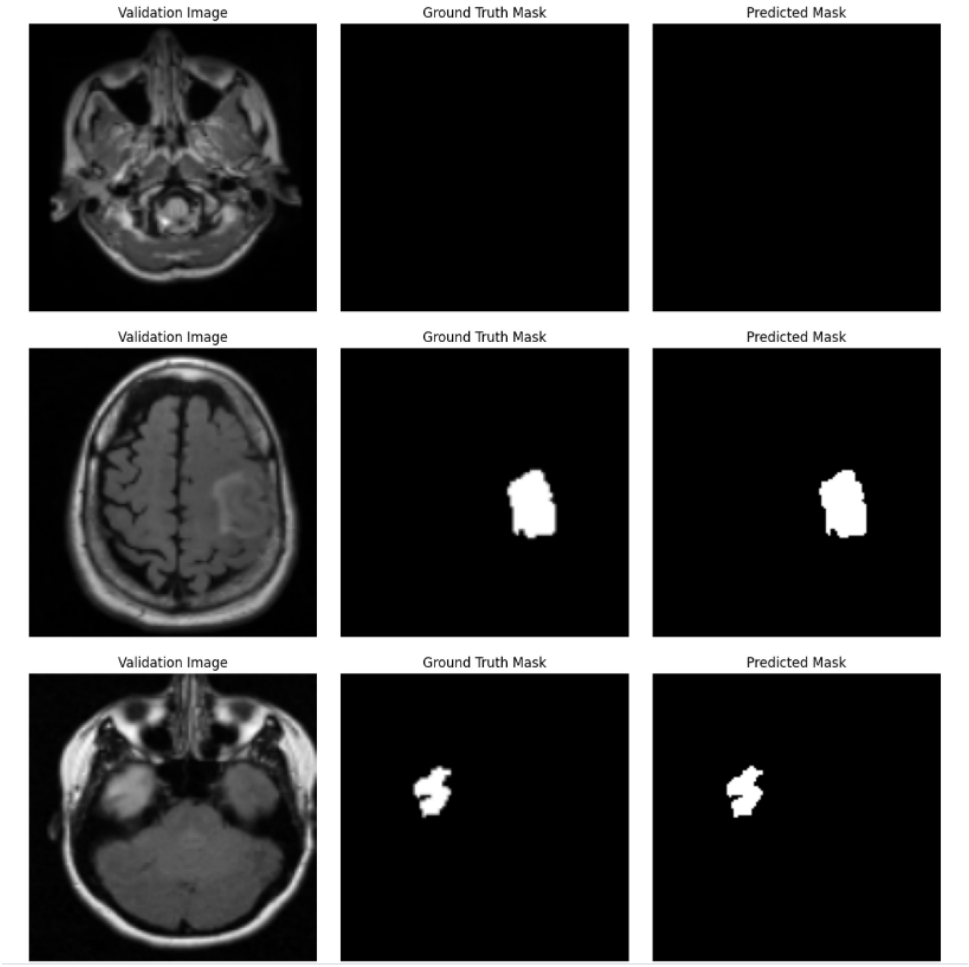
2.3 Comparison with Reference Paper

The implemented U-Net was compared with the performance metrics reported in a reference paper:

Metric	Implemented Model	Paper
Accuracy	0.9927	0.9981
Dice Coefficient	0.7209	0.8409
IoU Coefficient	0.8336	0.9130
Loss	0.0141	0.0054
Precision	0.7801	0.9974
Sensitivity	0.7099	0.9971
Specificity	0.9976	0.9991

2.4 Visualization of Results

The following examples illustrate the segmentation results:



3 Bonus Task: Brain MRI Segmentation Using ResUNet

3.1 Introduction: What is ResUNet?

ResUNet is an enhanced version of U-Net that integrates residual connections into the architecture. These residual blocks help improve gradient flow and ease the training of deeper networks, leading to more robust and effective segmentation performance.

Metric	Implemented Model	Paper
Accuracy	0.9940	0.9981
Dice Coefficient	0.7891	0.8409
IoU Coefficient	0.8543	0.9130
Loss	0.0131	0.0054
Precision	0.8237	0.9974
Sensitivity	0.7695	0.9971
Specificity	0.9984	0.9991

Table 1: 3.7. Comparison with Reference Paper

3.2 Architecture Changes

The ResUNet architecture incorporates residual blocks in both the encoder and decoder. This modification helps mitigate vanishing gradients and enhances feature propagation. In the implementation, the residual blocks were designed to preserve input information, which contributed to improved training stability.

3.3 Implementation Details

Similar to the U-Net, the input images were preprocessed (resized to 128×128 pixels, normalized, and cropped). However, additional convolutional layers were replaced with residual blocks, and skip connections were adapted to accommodate the residual structure.

3.4 Model Performance

3.5 Challenge: Resource-Intensive Model Training

Solutions:

- **Mixed Precision** Trained with float16 (half precision) instead of float32 since this uses less GPU memory and can speed up matrix operations.

3.6 Training Setup

- **Optimizer:** Adam optimizer with an initial learning rate of 0.001.
- **Batch Size:** 10
- **Epochs:** 80
- **Callbacks:**
  - ReduceLROnPlateau to dynamically adjust the learning rate.
  - ModelCheckpoint to save the best model based on validation loss.

3.7 Comparison with Reference Paper

There are noticeable improvements from the U-Net model shown in table 1.

3.8 Visualization of Results

The following examples illustrate the segmentation results: