

# 2D Brain Tumor MRI Segmentation Using U-Net

Final Project – Data Science and AI Diploma

Moamen Mohamed Ahmed Hassan  
moamengaber2002@gmail.com

July 28, 2025

## Executive Summary

This project presents a deep learning solution for automatic brain tumor segmentation in 2D MRI images using the U-Net architecture. Manual segmentation of brain tumors is often time-consuming, subject to variability, and requires expert knowledge. Automating this process can assist radiologists in improving diagnostic accuracy and speed.

The dataset used in this project was obtained from Kaggle and consisted of 4,715 paired MRI images and corresponding segmentation masks, stored in HDF5 format. A total of 3,017 images were used for training, 943 for validation, and 755 for testing. The U-Net model was trained to generate binary masks highlighting tumor regions within the brain MRIs.

Data preprocessing included intensity normalization, resizing, and decoding of the HDF5 file structure. Model performance was evaluated using Dice Similarity Coefficient and Intersection over Union (IoU) metrics. The final model achieved a Dice score of **0.9789** and an IoU score of **0.9588** on the test set, demonstrating excellent tumor segmentation performance.

This project was completed as the final capstone of a Data Science and AI diploma program, and highlights how U-Net can be effectively applied to real-world medical imaging tasks.

# Contents

<b>Executive Summary</b>	<b>2</b>
<b>1 Introduction</b>	<b>4</b>
<b>2 Data Description</b>	<b>4</b>
<b>3 Data Preprocessing</b>	<b>5</b>
<b>4 Modeling</b>	<b>6</b>
4.1 Model Architecture . . . . .	6
4.2 Loss Function and Metrics . . . . .	6
4.3 Optimizer and Training Parameters . . . . .	7
4.4 Callbacks and Training Strategy . . . . .	7
<b>5 Results and Evaluation</b>	<b>7</b>
5.1 Quantitative Metrics . . . . .	7
5.2 Training vs Validation Performance . . . . .	8
5.3 Qualitative Results . . . . .	8
<b>6 Conclusion</b>	<b>10</b>
<b>7 Future Work</b>	<b>10</b>
<b>8 Appendix</b>	<b>11</b>
8.1 Dataset Link . . . . .	11
8.2 Model Summary . . . . .	11
8.3 Training Configuration . . . . .	11
8.4 Additional Resources . . . . .	12
<b>9 References</b>	<b>12</b>

# 1 Introduction

Brain tumor segmentation from magnetic resonance imaging (MRI) plays a critical role in the diagnosis, treatment planning, and monitoring of patients with neurological conditions. Traditional manual segmentation is a time-consuming task that requires trained radiologists and is prone to inter-observer variability. Automated segmentation techniques, especially those based on deep learning, offer a promising alternative by providing fast, consistent, and accurate results.

This project aims to develop a deep learning model capable of performing 2D brain tumor segmentation using MRI scans. Specifically, a U-Net convolutional neural network architecture was implemented to generate binary segmentation masks that highlight tumor regions. U-Net has proven to be effective in biomedical image segmentation tasks due to its encoder-decoder structure and skip connections that preserve spatial information.

The dataset used for this project was sourced from Kaggle<sup>1</sup> and contains 4,715 grayscale brain MRI slices along with corresponding tumor masks. These were divided into training, validation, and test sets with 3,017, 943, and 755 images respectively. One of the key challenges encountered during the project was the dataset's HDF5 format, which required additional preprocessing to convert into usable image and mask pairs.

This project was conducted as the final capstone of a Data Science and AI diploma program and serves as a practical application of convolutional neural networks in the medical imaging domain. The success of this project reinforces the applicability of U-Net and similar architectures in solving real-world healthcare problems.

## 2 Data Description

The dataset used in this project was obtained from Kaggle<sup>2</sup> and consists of 4,715 2D brain MRI slices, each paired with a corresponding binary segmentation mask indicating tumor regions. All images were originally grayscale with a resolution of  $240 \times 240$  pixels and were rescaled to  $256 \times 256$  pixels to ensure consistency during model training.

The images were stored in HDF5 format, with two primary files: one containing the raw MRI scans and the other containing the binary masks. Each mask uses a pixel value of 0 to represent the background and 1 to represent the tumor region, making it a binary segmentation task.

The dataset was split into:

- **Training set:** 3,017 images
- **Validation set:** 943 images
- **Test set:** 755 images

All images in the dataset contained visible tumor regions, which helped avoid issues related to class imbalance. The dataset covered a range of tumor shapes and sizes, making

---

<sup>1</sup><https://www.kaggle.com/datasets/balakrishncodes/brain-2d-mri-imgs-and-mask>

<sup>2</sup><https://www.kaggle.com/datasets/balakrishncodes/brain-2d-mri-imgs-and-mask>

it suitable for training a robust segmentation model. A sample image-mask pair is shown in Figure 1.

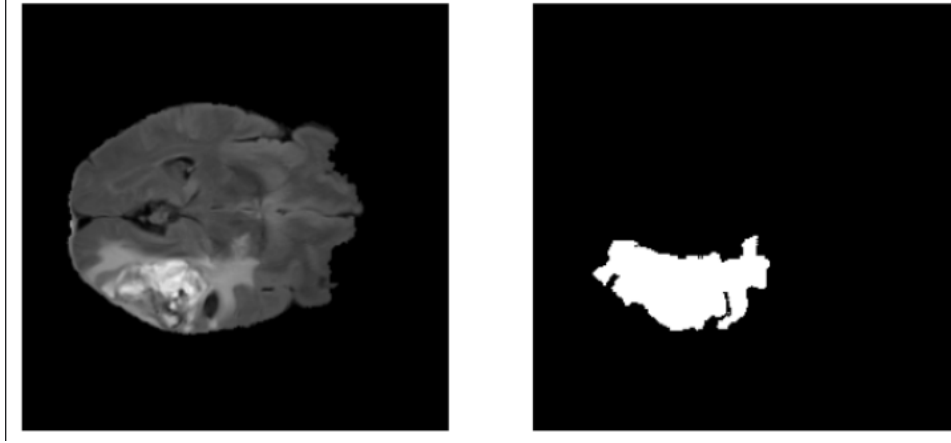


Figure 1: Sample brain MRI slice (left) and its corresponding tumor mask (right).

### 3 Data Preprocessing

The original dataset was provided in HDF5 format, which required conversion to a usable image format before training. Each HDF5 file contained two datasets: a 2D brain MRI slice and its corresponding tumor mask. These were extracted using the `h5py` library in Python. During extraction, the following preprocessing steps were applied:

- **Normalization:** Pixel intensity values of the MRI images were linearly normalized to the range  $[0, 255]$  to enhance contrast and ensure consistency.
- **Binarization:** The tumor masks were converted into binary format, where 0 represented the background and 1 represented tumor regions. Any non-zero pixel was treated as part of the tumor.
- **Resizing:** Both images and masks were resized from their original resolution of  $240 \times 240$  pixels to  $256 \times 256$  pixels to standardize input dimensions across the dataset.
- **Format Conversion:** The normalized images and binary masks were saved as PNG files for easier handling during training.

After preprocessing, the data was split into training (3,017 images), validation (943 images), and test (755 images) sets using a stratified and reproducible approach. The filenames were carefully paired to ensure that each image corresponded exactly to its mask.

To enhance the training set and reduce overfitting, data augmentation was applied using the `Albumentations` library. Specifically:

- **Horizontal flipping** was applied to all training images and masks.

- **Random shifting, scaling, and rotation** were used to simulate variations in position and orientation.

Each original training image generated two augmented variants, tripling the size of the training dataset. No augmentation was applied to the validation or test sets to ensure unbiased evaluation.

All preprocessed and augmented images were organized into separate directories for training, validation, and testing, with further subfolders for images and masks to streamline model training and loading.

## 4 Modeling

For this brain tumor segmentation task, a U-Net-based convolutional neural network was implemented using TensorFlow and Keras. U-Net is a well-established architecture for biomedical image segmentation due to its encoder-decoder structure and its ability to learn from relatively few annotated images while preserving spatial information.

### 4.1 Model Architecture

The model consisted of a symmetric U-Net with an encoder-decoder structure:

- **Input Shape:**  $256 \times 256 \times 1$  (grayscale MRI slices)
- **Encoder:** Comprised of four convolutional blocks, each containing two  $3 \times 3$  convolutional layers with ReLU activation, batch normalization, and a  $2 \times 2$  max pooling operation.
- **Bottleneck:** Two convolutional layers with 1024 filters.
- **Decoder:** Each decoder block consisted of a transposed convolution for upsampling, concatenation with the corresponding encoder feature maps (skip connections), followed by two  $3 \times 3$  convolutional layers with ReLU activation and batch normalization.
- **Output Layer:** A  $1 \times 1$  convolution with sigmoid activation to produce a binary segmentation mask.
- **Regularization:** A dropout layer with rate 0.4 was used at the end of the decoder to reduce overfitting.

### 4.2 Loss Function and Metrics

Given the nature of the segmentation task and the class imbalance typical in medical image data, the Dice loss was used. It is defined as:

$$\text{Dice Loss} = 1 - \frac{2 \sum y_{\text{true}} \cdot y_{\text{pred}} + \epsilon}{\sum y_{\text{true}} + \sum y_{\text{pred}} + \epsilon}$$

where  $\epsilon = 1 \times 10^{-6}$  is a small constant to avoid division by zero.

For evaluation during training, both Dice Coefficient and Intersection over Union (IoU) were monitored.

### 4.3 Optimizer and Training Parameters

The model was compiled using the Nadam optimizer with a learning rate of  $1 \times 10^{-4}$ . The training was conducted with the following hyperparameters:

- **Batch size:** 32
- **Epochs:** 100 (early stopping triggered at epoch 65)
- **Learning Rate Scheduler:** ReduceLROnPlateau was used to reduce the learning rate by a factor of 0.5 when the validation loss plateaued, with a minimum learning rate of  $1 \times 10^{-7}$

### 4.4 Callbacks and Training Strategy

To ensure optimal training and avoid overfitting, the following callbacks were utilized:

- **EarlyStopping:** Monitored the validation loss with a patience of 8 epochs, halting training when performance stopped improving.
- **ModelCheckpoint:** Saved the best-performing model based on validation loss.
- **ReduceLROnPlateau:** Dynamically adjusted the learning rate during training.
- **CSVLogger** and **TensorBoard:** Used for tracking training logs and visualizations.

The model achieved its best performance at epoch 65 with a final learning rate of  $1.5625 \times 10^{-6}$ , indicating the learning rate scheduler successfully helped fine-tune the training in its later stages.

## 5 Results and Evaluation

To assess the performance of the trained U-Net model, both quantitative metrics and qualitative visualizations were employed on the test set.

### 5.1 Quantitative Metrics

The model was evaluated using several common segmentation metrics: Dice Coefficient, Intersection over Union (IoU), Accuracy, Precision, Recall, and F1-score. The results on the held-out test set are summarized as follows:

- Dice Coefficient: **0.979**
- IoU Score: **0.959**

- Accuracy: **0.998**
- Precision: **0.983**
- Recall: **0.975**
- F1 Score: **0.979**

These results demonstrate that the model is highly effective at segmenting brain tumors from MRI images, achieving strong performance across all major evaluation metrics.

## 5.2 Training vs Validation Performance

During training, the model showed no significant signs of overfitting. The best training loss was achieved with a Dice Coefficient of 0.9855, IoU of 0.9710, and loss of 0.0145. The corresponding validation metrics were Dice Coefficient of 0.8211, IoU of 0.7759, and loss of 0.1816. These values suggest that the model maintained good generalization to unseen data.

## 5.3 Qualitative Results

To further evaluate the model's predictions, the following examples illustrate the input MRI slice, ground truth tumor mask, and the model's predicted mask side by side.

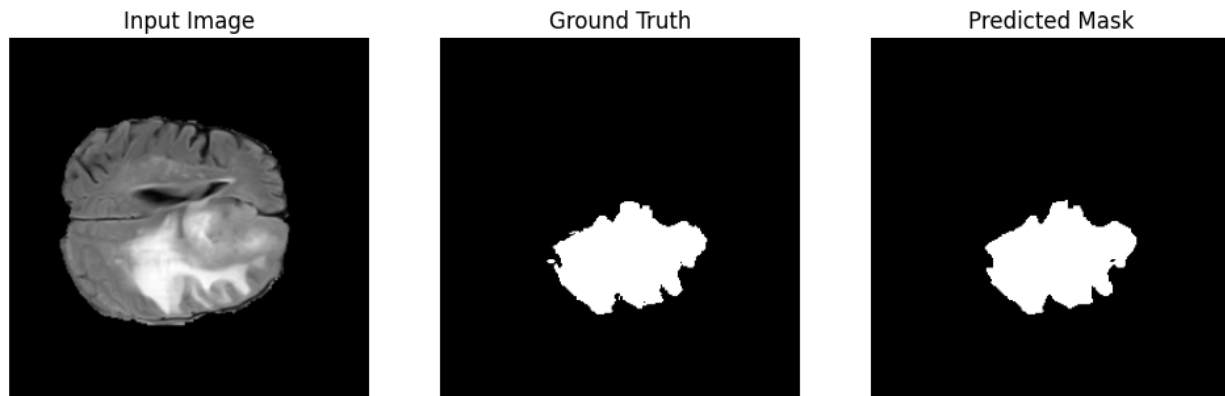


Figure 2: Example 1: Input image, Ground Truth, and Predicted Mask



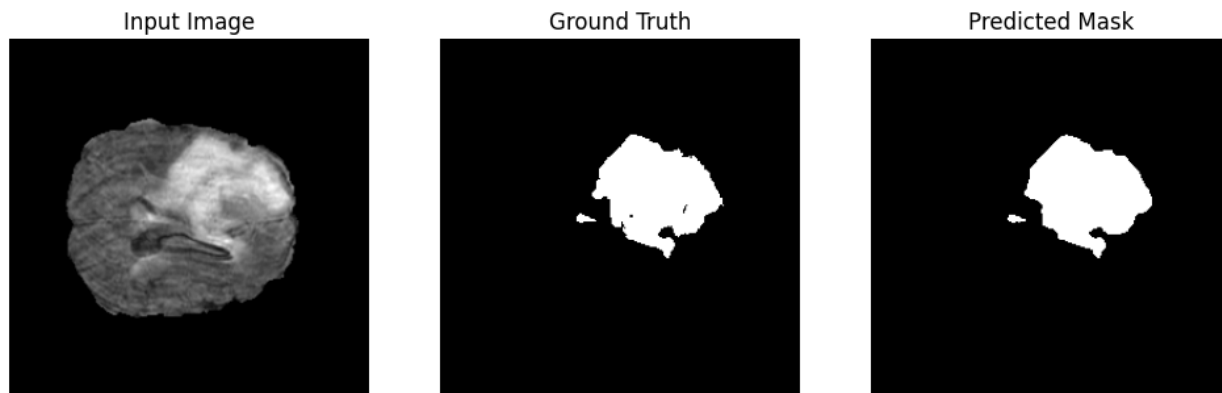


Figure 3: Example 2: Input image, Ground Truth, and Predicted Mask

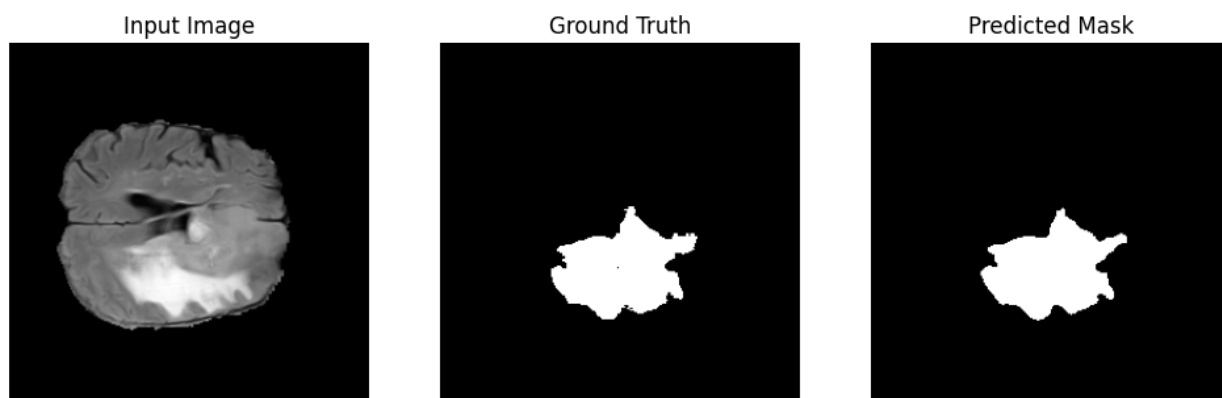


Figure 4: Example 3: Input image, Ground Truth, and Predicted Mask

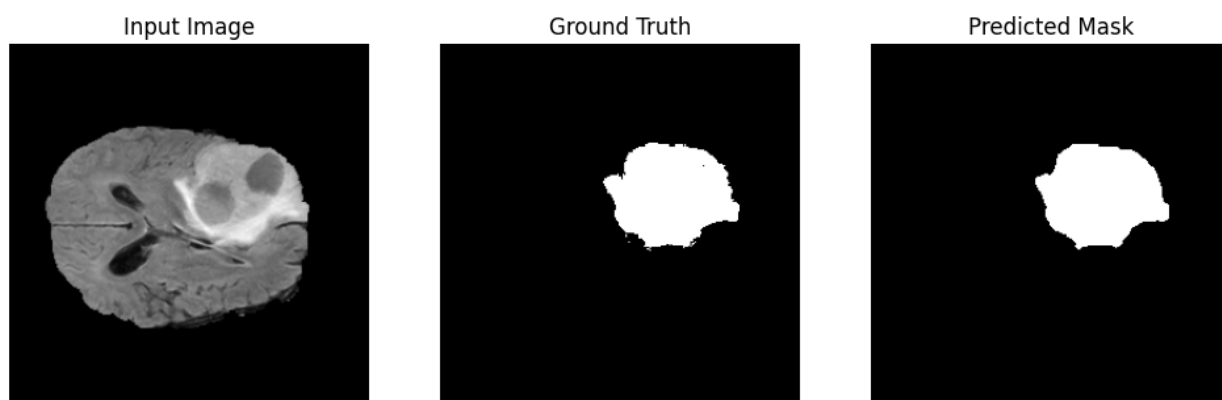


Figure 5: Example 4: Input image, Ground Truth, and Predicted Mask

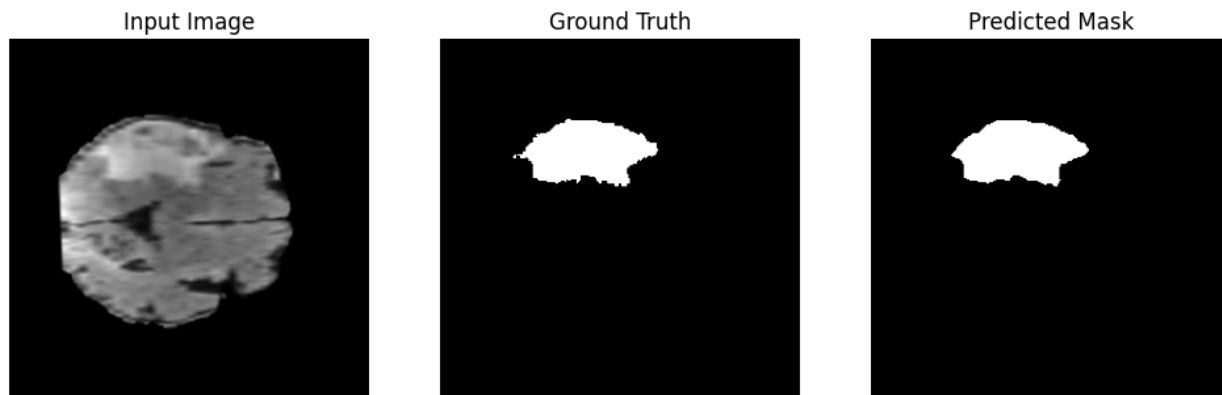


Figure 6: Example 5: Input image, Ground Truth, and Predicted Mask

## 6 Conclusion

This project successfully implemented a U-Net-based deep learning model to perform 2D segmentation of brain tumors using MRI scans. The dataset consisted of 4715 grayscale MRI images and their corresponding binary masks, and the model was trained, validated, and tested on appropriately split subsets. Through careful preprocessing, data augmentation, and model tuning, the model achieved strong performance on the test set, with a Dice coefficient of **0.9789** and an Intersection over Union (IoU) score of **0.9588**.

In addition to the strong quantitative metrics, visual inspection of the predicted masks showed that the model was capable of learning the spatial structures of tumors effectively. The model generalized well without significant signs of overfitting, as confirmed by consistent validation performance and well-behaved learning curves.

This work demonstrates the effectiveness of convolutional neural networks—specifically the U-Net architecture—for medical image segmentation tasks, and highlights their potential for supporting radiologists and healthcare professionals in analyzing complex brain scans efficiently and accurately.

## 7 Future Work

While the current model achieved high accuracy in segmenting brain tumors, there are several potential directions for future improvement:

- **3D Segmentation:** Extending the model to handle full 3D MRI volumes rather than 2D slices could improve spatial continuity and context awareness.
- **Multi-Class Segmentation:** If more detailed labels become available (e.g., differentiating between tumor types or tissue zones), the model could be trained to perform multi-class segmentation.
- **Transfer Learning:** Using pre-trained encoders from larger datasets or medical imaging models may improve performance, especially on smaller datasets.

- **Post-Processing:** Applying morphological operations or conditional random fields (CRFs) as post-processing steps might further refine mask boundaries.
- **Model Deployment:** Integrating the model into a clinical decision-support tool or a GUI interface could provide real-time assistance to radiologists.

These improvements would help move the model closer to real-world medical applications, increasing its clinical utility and reliability.

## 8 Appendix

### 8.1 Dataset Link

The dataset used in this project was sourced from Kaggle: <https://www.kaggle.com/datasets/balakrishcodes/brain-2d-mri-imgs-and-mask>

### 8.2 Model Summary

Below is the summary of the U-Net model architecture used:

Model: "functional"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 256, 256, 1)]	0
conv2d (Conv2D)	(None, 256, 256, 64)	640
batch_normalization (BN)	(None, 256, 256, 64)	256
...		
conv2d_18 (Conv2D)	(None, 256, 256, 1)	65

Total params: 31,042,113

Trainable params: 31,036,353

Non-trainable params: 5,760

### 8.3 Training Configuration

- Optimizer: Nadam, Learning rate:  $1e-4$  with LR scheduler
- Loss Function: Dice Loss
- Batch Size: 32
- Epochs: 100 (Early Stopping triggered at epoch 65)
- Callbacks: EarlyStopping, ReduceLROnPlateau, ModelCheckpoint, CSVLogger, TensorBoard

## 8.4 Additional Resources

The full source code, trained model, and GUI application are available on GitHub at the following link:

[github.com/MoamenGaber2/brain\\_mri\\_2d\\_segmentation](https://github.com/MoamenGaber2/brain_mri_2d_segmentation)

The repository includes:

- Data preprocessing and model training pipeline
- Evaluation and visualization scripts
- A PyQt5-based GUI for testing the model interactively
- Instructions for reproducing the results

## 9 References

### References

- [1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. MICCAI, 2015. Available at: <https://arxiv.org/abs/1505.04597>
- [2] Balakrish Codes. *Brain 2D MRI Images and Mask Dataset*. Kaggle, 2022. Available at: <https://www.kaggle.com/datasets/balakrishcodes/brain-2d-mri-imgs-and-mask>
- [3] Martín Abadi et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. Available at: <https://www.tensorflow.org/>