**Brain Tumor Semantic Segmentation Using a U-Net Model**

Ethan Wen

**Abstract**

A brain tumor is an abnormal growth of cells in brain tissue and can cause death if not treated properly. Patients who might have a brain tumor usually get a magnetic resonance imaging (MRI) scan which is then interpreted by a radiologist, a process that could take days or even weeks. However, this process can be bypassed with machine learning models. Machine learning models can read MRI scans and interpret them within a few hours, which saves time for both the patient and physician involved. This increases the efficiency of brain tumor detection processes and can save lives in some cases.

A common image detection technique is semantic segmentation, which classifies each pixel in an image as a certain class. For example, it can classify one area of pixels to be a brain tumor, and the rest of it as background. U-Net is a convolutional neural network (CNN) architecture that connects the down-sampled images to the corresponding up-sampled images and is often used in semantic segmentation models to produce greater accuracy. This model uses U-Net architecture in a semantic segmentation model to predict if there is and where the brain tumor is in brain scans.

The main goal of this experiment is to predict brain tumors accurately and quickly compared to other models. The model then outputs the image to show where the predicted brain tumor is in the given image. Trained on a public brain tumor detection dataset from Kaggle, this model detects brain tumors in brain scans with a fairly high intersection over union score at a quick rate, showing high efficiency and decent accuracy

**Introduction**

Being able to detect brain tumors in medical images is crucial to saving the lives of many people. However, it can be difficult to identify the brain tumor in scans and take up valuable time for both the patient and the doctor. This presents an issue since brain tumors should be removed quickly to prevent any further harm. By streamlining this process with a tumor detection program accuracy can be increased and time can be saved for both parties.

Semantic segmentation identifies pixels and classifies them based on their characteristics. This helps machines classify different groups of pixels as different objects. The difference between semantic segmentation and object detection models is they use pixels instead of bounding boxes. This makes semantic segmentation more precise and more useful than object detection in some scenarios. A brain tumor dataset for semantic segmentation is perfect for this project. Using semantic segmentation, the model outputs a smaller and more precise area for where the tumor should be.

This project is meant to innovate previously created code to make the model output more accurate predictions in a shorter amount of time. With more accurate and precise predictions the model’s reliability will be increased and it will be able to work in real settings as well. By decreasing the time it takes to output the predictions, the model allows medical professionals and patients to receive results back quicker, increasing the efficiency of the model as well.

**Dataset**

The dataset Brain Tumor Image DataSet: Semantic Segmentation is published on Kaggle. The dataset contains 1502 training images and 215 test images. There are two classes of images, one with no tumor present and the other contains tumors. The dataset gives annotations on each image which contains coordinates for a bounding box around where the tumor is located. These boxes are rectangles and contain parts of the image that are not part of the tumor.A close-up of a brain

Description automatically generated

**Fig 1.** These images contain the original image, randomly taken from the dataset, the bounding box around the tumor with the image, the coordinates given by the annotations, and just the bounding box without the original image.

**Model**

My model is a convolutional neural network (CNN) that classifies images using semantic segmentation. There are two parts to the model, an encoder and a decoder. These serve a similar purpose to an auto encoder which compresses an image and the pixels down to a few numbers for easier training and processing, then decompresses it back to the original image. This serves to represent a large number of pixels with a couple of numbers, allowing easier storage and reducing noise and redundancy in the image. The decoder then takes these numbers and decompresses them back into the original image.

A good example of this is a zip file. A zip file contains multiple files but compresses it into a single file to lower the storage space. This is just like an encoder as they both compress something down to be able to store things more efficiently. When the zip file is extracted, the files inside get taken out and are deconstructed into whole files again, just like a decoder.

My model utilizes U-NET architecture. U-NET architecture is different from standard CNNs because of skip connections. Skip connections connect steps from the encoding path to the corresponding step in the decoding path. To localize, high resolution features from the compressing path are combined with the upsampled output, giving a more precise output (Ronneburger et al, 2015). This ensures greater precision between the original and the decompressed image from the decoder. The output of the model is a segmentation mask that separates the class of every pixel in the image.

A diagram of a flowchart

Description automatically generated

**Fig 2.** U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. (Ronneberger et al, 2015)

**Method**

***Data Augmentation***

The data needs to be altered in various ways to ensure that the model is trained on different kinds of images and isn’t used to only seeing a certain kind, preventing overfitting. First, the images are resized to all be the same size and a mask is created to find the tumor. Each image is randomly selected to be randomly selected to be resize cropped so the model will be able to work with different types of images and randomly selected to be flipped horizontally. The image and mask is then converted to tensors and the image is normalized. Each annotation is also connected to the corresponding image. Masks are aimed at being as accurate as possible and try to contain the whole tumor while minimizing surrounding pixels.

***Training Details***

My U-NET function specifically uses 1 residual block per group, 1 class, 3 input channels, starting dimensions of 64, 16 groups, and dimension multipliers of 1, 2, 4, and 8. Having fewer residual blocks and classes prevents overfitting and allows the program to run smoothly. I tested the U-NET with different sized parameters, and these gave the best results.

My training function uses a learning rate of 0.001, batch size of 32, gradient accumulation of 4, 50 epochs, weight on tumor pixels of 5, and an image size of 256. After testing different learning rates, epochs, and batch sizes, these variables give the best results. My model uses a standard BCE loss function and an AdamW loss function, both from Pytorch. Overall, these specific parameters allow my model to be trained quickly and has the most accurate results after testing with smaller and larger epoch sizes and weights.

A graph of a graph showing the results of a training loss

Description automatically generated

**Fig 3.** Training and test loss curves over 50 epochs

**Results**

A close-up of a mri

Description automatically generated

**Fig 4**. An accurate mask prediction and a high IoU score with the bounding box

A comparison of a brain scan

Description automatically generated with medium confidence

**Fig 5.** A mask with a poor mask prediction and a low IoU score

The model has 97% accuracy which is not perfect and an Intersection over Union (IoU) score 0.55. Since much of each image is background space, the accuracy is heavily inflated. However, the IoU score is moderate, and on average the majority of predicted pixels overlap with the bounding box given by the dataset.

**Discussion**

The semantic segmentation model classifies each pixel into either being part of a tumor or not part of a tumor. The annotations use bounding boxes around the tumors. However, the masks from the model aren’t rectangles. The masks are irregular shapes that contain every pixel that is classified as a tumor, making them more accurate than the bounding boxes. This presents an issue in the intersection over union (IoU) calculation, since the intersecting mask and bounding box will almost never be perfect since the bounding box contains many pixels that aren’t the tumor. If the model is tested on data with a circular bounding area for the annotations, the IoU may be better and the results could show a more accurate prediction from the model. Training the model for more than 50 epochs was also an issue, since it takes more computational power that wasn’t available.

The results of the model were accurate most of the time but had some major drawbacks. This is seen in figures 4 and 5 the model can still improve in terms of prediction consistency. Some images, like Figure 4, are predicted very accurately while others, like Figure 5, are predicted poorly. Some tumor predictions were completely off the mark and the IoU was offset by such predictions and other very accurate predictions. A limitation to this experiment was the laptop used, since it ran 50 epochs quite slowly. Being able to run through more epochs at a quicker rate would allow for more testing with the epoch number and weights to see which one would allow for the optimal IoU score. Running the model with other datasets would allow for insight on if the model is overfitted, which wasn’t able to be done with this specific project. As seen in Figure 3, the training loss is still decreasing and shows there can still be improvement as the number of epochs increases. This could allow for more accurate predictions and a higher IoU between the masks and bounding boxes.

**Conclusion**

Future extensions to the project could include using different types or sizes of datasets. Larger or smaller datasets could increase or decrease accuracy and speed of the model, and provide more insights as to how to improve it. Different models may work better for this kind of image as well. Testing out different types of machine learning models with many datasets can improve upon the mask predictions and provide more insight as to the most efficient model for brain tumor image predictions. Training the model further and then testing the accuracy against other machine learning models for this dataset and for other similar datasets to find the most efficient option are good options to continue this project.

**References**

Ibm. (2024, October 17). *What are convolutional neural networks?*. IBM. https://www.ibm.com/topics/convolutional-neural-networks

Ibm. (2024b, December 2). *What is loss function?*. IBM. https://www.ibm.com/think/topics/loss-function

Priyammaz. (n.d.). *Priyammaz/Pytorch-Adventures: This repository contains an exhaustive coverage of a hands on approach to pytorch along side powerful tools to accelerate model tuning and training*. GitHub. https://github.com/priyammaz/PyTorch-Adventures

Ronneberger, O., Fischer, P., & Brox, T. (1970, January 1). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. SpringerLink. https://link.springer.com/chapter/10.1007/978-3-319-24574-4\_28

Rosebrock, A. (2024, November 18). *Intersection over union (IOU) for object detection*. PyImageSearch. https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

*Semantic segmentation*. Semantic Segmentation - MATLAB & Simulink. (n.d.). https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html#:~:text=Semantic%20segmentation%20is%20a%20deep,pavement%2C%20and%20other%20road%20features.

Wang, R., Zhuang, L., Chen, H., Xu, B., & Cai, R. (2024). TP-UNet: Temporal Prompt Guided UNet for Medical Image Segmentation. *arXiv preprint arXiv:2411.11305*.