Final Project For ECE228 Track Number #1

Group Number #35: Achyuth Mahesh Esthuri Devanshi Panchal Krish Mehta

Prashil Parekh

Abstract

Applying Machine Learning techniques on Medical Imaging tasks has always been a challenging problem. Brain Tumour detection using MRI scans is one such application which has a huge demand and potential upsides from automation, but is still not a solved problem. In this project, we explore 2 major CNN-based approaches to Brain Tumour detection, 1. A pixel-by-pixel Cascading CNN, which trains on focused mini-patches of MRIs, 2. An image-based UNet with Attention, which trains on complete MRI scans and predicts tumour classes. We make some enhancements in the training process and explore the trade-offs of each of these models in order to get the best possible classification results.

0.1 Evaluation

Achyuth Mahesh Esthuri: I certify that I have filled the evaluation.

Devanshi Panchal: I certify that I have filled the evaluation.

Krish Mehta: I certify that I have filled the evaluation.

Prashil Parekh: I certify that I have filled the evaluation.

1 Introduction

Medical imaging, such as MRIs and CT scans, play a critical role in diagnosing and treating brain tumors. Accurately identifying and delineating these tumors is vital for medical professionals to understand their size, location, and characteristics, aiding in treatment planning and patient care. However, manual segmentation of brain tumors from imaging data is time-consuming and prone to human error.

Our project tackles this challenge by investigating fully automated approaches utilizing Deep Neural Networks (DNNs), drawing from existing research and our own enhancements. Through the utilization of sophisticated deep learning algorithms, our goal is to increase the accuracy and effectiveness of tumor segmentation. This advancement not only accelerates the analysis process but also offers the potential for improved prognoses and personalized treatment plans for patients.

In essence, the motivation behind this project lies in the urgent need to develop reliable and efficient tools that assist medical professionals in accurately identifying and delineating brain tumors from imaging data, ultimately leading to improved patient outcomes and personalized care.

2 Related work

Over the past several decades, there has been a significant increase in publications focused on automated brain tumor segmentation. These efforts leverage domain-specific prior knowledge about the appearance of both healthy and tumorous tissues, often utilizing anatomical models obtained by aligning 3D MR images with atlases or templates derived from multiple healthy brains.

Traditional brain tumor segmentation methods typically fall into two categories: generative and discriminative models. Generative models depend on this domain-specific knowledge to simulate the appearance of brain tissues [3], while discriminative models focus on extracting low-level image features such as raw pixel values, local histograms, texture features, and alignment-based features. Discriminative models require manually designed features, which can be computationally intensive and memory-demanding [8, 7] .

Hand-engineered features in discriminative models generally use broad edge-related information without specific adaptation to brain tumors. In contrast, deep Convolutional Neural Networks (CNNs) can automatically compose and refine features into higher-level, task-specific representations. Although CNN-based segmentation models are well-established in natural scene labeling, their application in medical imaging has been comparatively limited. However, some researchers like Huang and Jain have successfully used CNNs to delineate neural tissue boundaries in electron microscopy images.

Despite the success of CNNs in non-medical tasks, their architectures often need more suitability for medical imaging and brain tumor segmentation. Preliminary studies, such as those presented at the BRATS 2014 challenge by Davy et al., Zikic et al., and Urban et al., indicate the promising potential of deep CNNs for brain tumor segmentation.

3 Methodology

We aim to harness the power of deep learning to improve the accuracy and efficiency of brain tumor segmentation. For this, we explore two primary approaches to utilizing deep CNNs for this task-Pixel-by-Pixel, which uses a Two-Path CNN, and Image-based technique, which uses a U-Net based architecture.

3.1 Pixel-by-Pixel Approach

Based on the model proposed in [1], this method involves training of the network using a small patch of the MRI. The input to the model is a one big patch 65x65 or 56x56 in size and one small patch of 33x33 size and 4 channels corresponding to each pixel. The significance of using two patches corresponding to the same input slice will be discussed in detail in the model architecture section. The output of the network is the classification of the center pixel of the 4x33x33 patch into one of the five categories of tumors, as defined by the dataset. By doing this, the model goes over the entire image producing labels pixel-by-pixel, essentially posing the pixel-wise segmentation problem as a classification problem.

3.1.1 BRATS Dataset

We utilized the BRATS 2013 and 2020 training dataset [5] to analyze the proposed methodology. This dataset includes both real patient images and synthetic images generated by SMIR. The dataset is categorized into two folders: High Grade (HG) and Low Grade (LG) images, we used the HG images.

Each patient's data comprises of horizontal slices of brain MRI scans in the following four imaging modalities, along with a fifth image that provides ground truth labels for each pixel:

- **T1-weighted (T1):** Produces high-resolution anatomical images of the brain. While it excels at visualizing brain structures, it is less effective at detecting tumor tissue compared to other imaging sequences.
- **T1-weighted post contrast (T1c or T1Gd):** Obtained after administering a contrast agent, typically gadolinium, which enhances areas with increased vascularity and blood-brain barrier disruption. This makes it particularly effective for identifying malignant tumors.
- **T2-weighted (T2):** Provides strong contrast for the brain's fluid spaces and is sensitive to edema, often found surrounding tumors. It is useful for visualizing both tumors and the surrounding tissue changes.
- **Fluid Attenuated Inversion Recovery (FLAIR):** Suppresses the fluid signal to help detect peritumoral edema and distinguish it from cerebrospinal fluid. It is especially useful for identifying lesions near or within the ventricles.

The image dimensions for HG is a 176 x 160 image with 4 channels each, representing the four modalities.

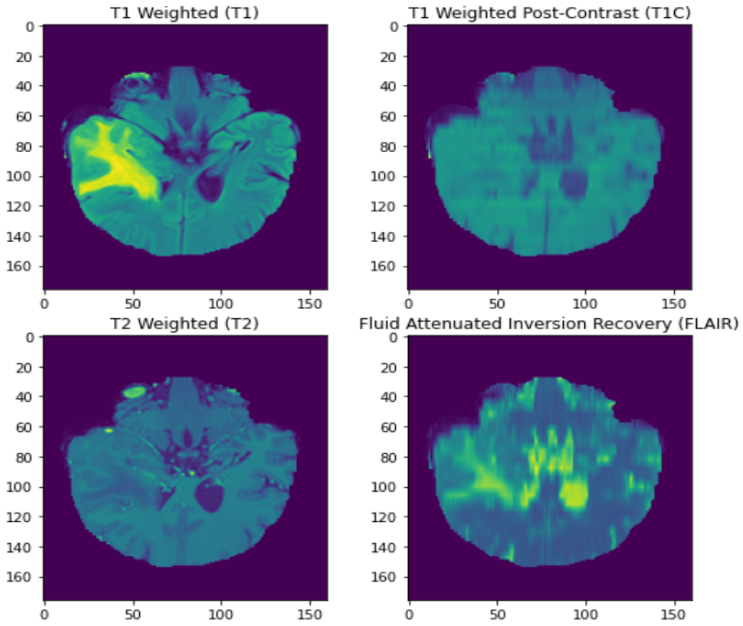

Figure 1: The four imaging modalities of brain MRI scans

The target labels for our dataset indicate segmentations that highlight areas of interest within the brain scans (Figure 2), particularly focusing on abnormal tissue associated with brain tumors. We use these labels to train models to differentiate brain tumors from normal brain tissue. These are the possible labels and what they represent:

- **Necrotic(NEC):** Marks the necrotic (dead) part of the tumor, which does not enhance with a contrast agent.
- **Edema (ED):** Identifies the edema, indicating swelling or fluid accumulation around the tumor.
- **Enhancing Tumor (ET):** Highlights the enhancing tumor, the region of the tumor that absorbs contrast material, often indicative of the most aggressive part of the tumor.
- **Non Enhancing Tumor (NET):** Tumorous portion, but not growing/advancing.
- **Everything else:** Non-tumorous part.

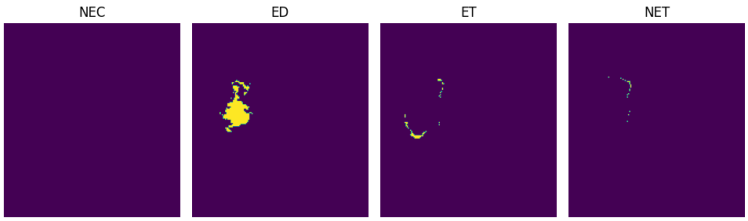

Figure 2: Labels Highlighting Tumorous Regions.

From Fig. 2 it is immediately apparent that an overwhelming majority of pixels are actually non-tumorous, and will not be highlighted in these ground truth images. This is one of the challenges of training a model on this dataset, which we discuss in detail in the experiments section.

3.1.2 Two-Path CNN Architecture

For the pixel-by-pixel approach, Cascading architectures are used, which are created by connecting two two-path CNNs in different ways. A two-path CNN architecture integrates two streams: a local pathway featuring 7x7 receptive fields and a global pathway with 13x13 receptive fields. As the local path has smaller kernel, it processes finer details because of small neighbourhood. Opposed to this, global path is used for a global feature extraction.The authors [1] chose this architectural design to ensure that the prediction of a pixel's label is influenced by two key factors: the visual details of the region around the pixel and its broader context, meaning the general area where the patch is located in the brain.

Figure 3 provides a detailed illustration of the entire architecture, which we refer to as the TwoPathCNN. To concatenate the top hidden layers of both pathways, the local pathway consists of two layers, with 3×3 kernels in the second layer. This setup ensures that the effective receptive field of features in the top layer of each pathway is identical, while the global pathway's parametrization more directly and flexibly models features in the same area. Ultimately, the feature maps from both pathways are combined.

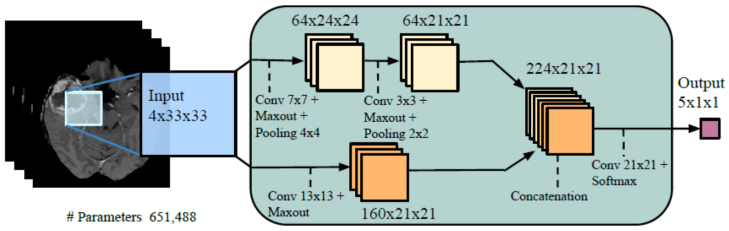

Figure 3: Two-Path CNN Architecture

3.1.3 Input Cascade CNN Architecture

The traditional CNN models, like the one described above, predict labels independently, lacking consideration for spatial dependencies. Addressing this issue, the paper explores different cascading architectures to incorporate such dependencies efficiently. By cascading two CNNs, the output probabilities of the first CNN serve as additional inputs to the second, enhancing label predictions during training of the model.

For our project, we implemented an Input Cascade CNN (shown in Figure 3) where the first two-path CNN's output is directly added to the second two-path CNN's input channels, treated as additional image channels to the network.

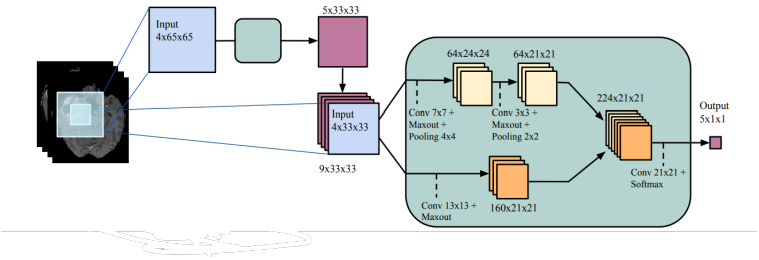

Figure 4: Input Cascade CNN Architecture

3.1.4 Training

We train the InputCascadeCNN with the Adam Optimizer, with a learning rate of 0.001. The categorical cross entropy loss is chosen as the loss function, as we're basically calculating the loss between the target label and predicted confidence values for each label. The training is done for 5 epochs per MRI slice, for the subset of slices that has non-negligible tumor presence. Training the

InputCascade CNN for the BRATS dataset, however, poses some unique challenges. We can't simply run the dataset through the model, because of the label distribution being extremely skewed. So, we experiment with a few training methodologies which are described in detail in the Experiments section.

3.2 Image-based Approach

Further in our project, we tried a full image based tumour classification approach using U-Net variants with Attention Mechanism based on the work of Oktay et. al [2]. The reason to pursue this approach is that the pixel-by-pixel approach was fairly difficult to train due to the extreme skew in the dataset. So, even if the model gets trained well for a given split of the dataset, it was not performing consistently well on MRI slices which had very scarce tumors. Unlike our previous approach, here we train a CNN on the entire MRI scan image and produce a processed image output which has tumorous regions highlighted or classified.

3.2.1 Dataset

In this part of the project, we use the a different variation of the BRATS Dataset to ensure compatibility with the UNet network. The dataset includes brain scan images accompanied by output images or masked labels that delineate regions of abnormal tissue. Input images for this variation are 240x240 and have the 4 modalities of MRI scans as described in the Dataset section, corresponding to each pixel. The model processes the entire scan as an image with input dimensions of 4x240x240. The output dimensions are also 240x240 but with 3 channels, with each channel representing a specific label: Necrosis (NEC), Edema (ED) or Enhancing Tumor (ET). If none of these channels have a non-zero value for a pixel, it is considered Non-Enhancing Tumor or Non-tumorous.

3.2.2 U-Net Architecture with Attention Mechanism

U-Net is a deep learning neural network originally designed [4] for image segmentation tasks, but it has since been applied to GANs and latent diffusion models. Like autoencoders, UNet includes an encoder to compress the input into a lower-dimensional representation and a decoder to reconstruct the output from this compressed form. However, UNet distinguishes itself with several features.

At its center is the bottleneck layer, which captures the most abstract representation of the input. The encoder and decoder are symmetrically designed, facilitating the implementation of skip connections that directly link layers of the encoder to their corresponding layers in the decoder. These skip connections, which bypass the bottleneck, transfer contextual information from the spatially rich encoder to the feature-heavy decoder, aiding in the precise localization necessary for detailed segmentation. Figure 5 depicts the original U-Net network

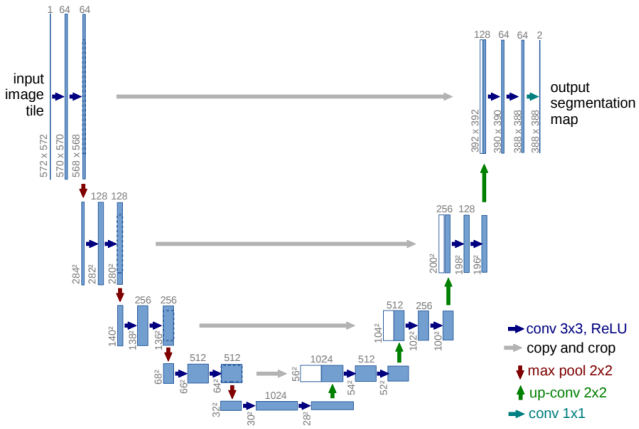

Figure 5: The original U-Net architecture [4]

Since the encoder captures good spatial representations but weaker feature representations, while the decoder excels in feature representations but may lose spatial details as the input moves deeper into

the network [2], we modified the U-Net architecture to include attention mechanism over the skip connection, allowing our model to be able to allocate more resources to relevant regions, and tune out the less important parts.

Figure 6 depicts the attention mechanism implemented. The attention process begins with the *query* (denoted as g for *gate* in the diagram), which is derived from the decoder's feature maps. These maps are anticipated to contain a detailed representation of the target structure due to the progressive decoding process.

The *key* is generated from the encoder's feature maps (marked x in the diagram), capturing the input image's contextual information. The objective is to pinpoint regions in the encoder's features that are pertinent to the query from the decoder.

The *query* and the *key* interact to create the attention map, which functions as a filter to highlight significant areas. This map is produced by aligning the query with the key, resulting in weights that indicate the importance of each feature in the encoder's output.

The *value*, also sourced from the encoder's feature maps, represents the content to be enhanced or suppressed. The attention map is applied to the value through element-wise multiplication, adjusting the features accordingly. Relevant features are emphasized, while less relevant ones are reduced.

This results in a weighted feature representation that concentrates on the most informative parts of the input, improving the network's information flow and enhancing segmentation quality. The attention mechanism ensures that the model focuses more on specific areas, increasing the precision and relevance of the final segmentation output.

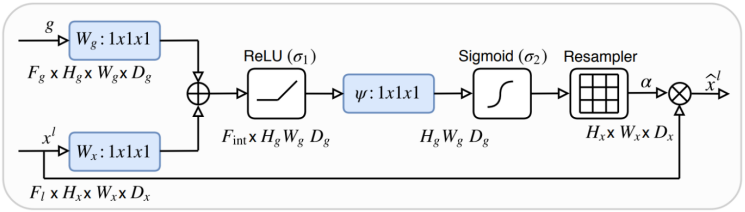

Figure 6: The attention mechanism implemented over the skip connections of U-Net [2]

Overall, our architecture leverages both encoder-decoder structures and attention mechanisms to improve the accuracy and precision of brain tumor segmentation. The encoder blocks compress the input features progressively, while the decoder blocks reconstruct the output. The attention blocks enhance feature learning and spatial focus by computing attention maps as described above. Inside the Attention class, the encoder downsamples input dimensions, and the bottleneck layer compresses features before decoding. Each decoder block integrates attention mechanisms to refine segmentation based on feature importance.

3.2.3 Training

The training of our Attention U-Net model begins by setting up the model and defining the training configuration. We use 25 epochs and the learning rate is 0.001, with a decay factor of 0.93. An Adam optimizer is initialized for parameter updates, while binary cross-entropy loss is chosen as the loss function, suitable for binary classification tasks like image segmentation.

4 Experiments

The primary challenge with the pixel-by-pixel is the skewed dataset problem. We did a bunch of experiments to try and mitigate this problem.

4.1 Skewed Dataset Problem and Two Step Training

The distribution of label classes in the BRATS dataset is extremely skewed. *Only about 98% of all pixels correspond to a non-tumorous label, i.e. label 0. Only 2% of pixels are tumors.* This poses a challenge in the sense that, we may be thinking that we have trained the model on N slices of the

MRI scan data but the actual number of datapoints that contribute to a tumor label is much smaller than that, and the model could get highly biased towards just classifying everything with a 0 label.

The proposed solution for this [1] is Two-Step Training:

1. First train on a distribution of patches which has equal probability of all tumour classes
2. Then fine-tune train on the full data distribution to remove any bias introduced by the first step

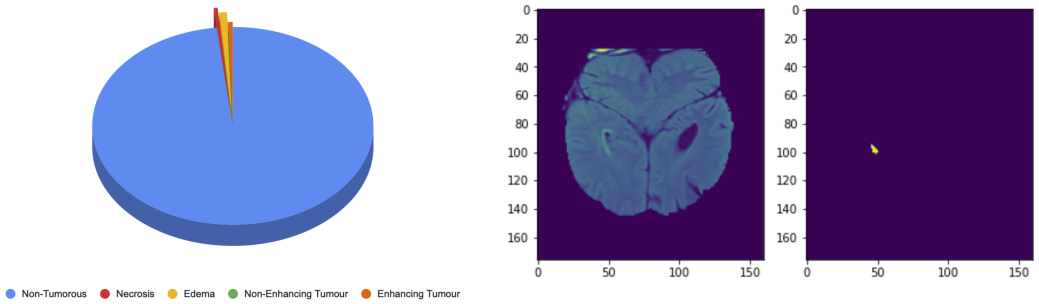

Figure 7: Above MRI slice's ground truth show how the presence of a tumour can be as small as a few pixels (none at all, for many slices) among 10^5 pixels

We implemented this training routine. In the first step we just feed the model patches corresponding to pixels which have tumor labels. This way the model learns what features within a patch contribute to tumors. However, just doing this will cause a lot of misclassification of healthy cells as tumorous. So, in order to avoid this, we do a fine-tuning step with 9X the data size as the first step (the ratio would be a hyperparameter) in order to avoid this bias. The inputs in this second step are random patches of the MRI scan, irrespective of whether they have a tumor or not.

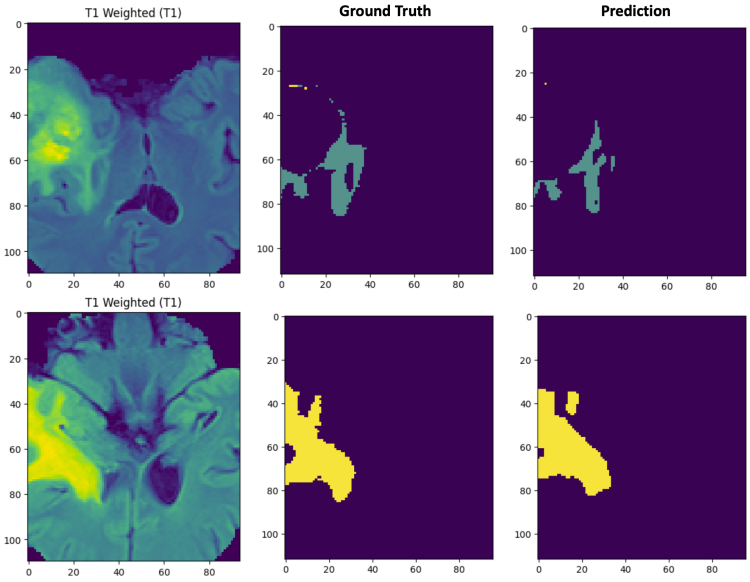

Figure 8: Cascade CNN Validation

We also implemented training with a blend of tumorous and non-tumorous patches, in different ratios. We also experimented an enhanced one-shot training, where the training loop only goes through slices with large enough tumour presence, thereby mitigating the distribution skew issue (although this has its own downsides in terms of over-classifying to tumour labels). The InputCascadeCNN takes two different patches of the same image as inputs: one big patch of 4x65x65 and one small

patch of 4x33x33. Note that both patches are centered on the same pixel, the pixel which the model will try to classify. Fig. 8 shows some of the best results achieved with the InputCascadeCNN model.

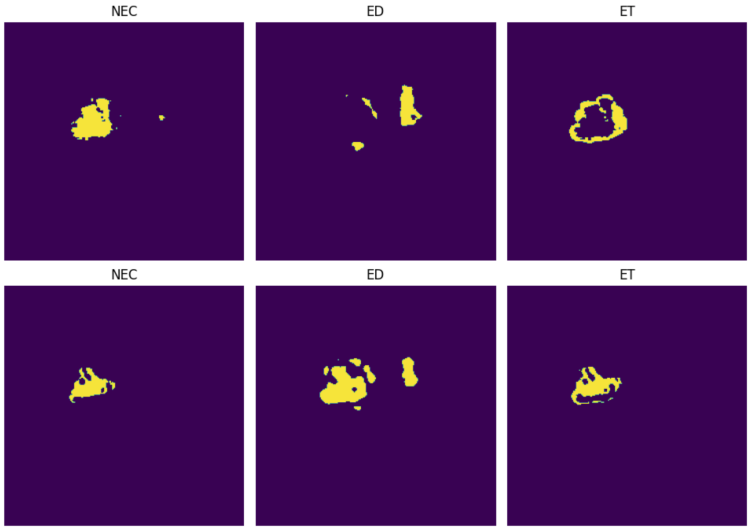

Figure 9: Attention UNet Validation, Ground Truth (Top) vs Prediction (Bottom)

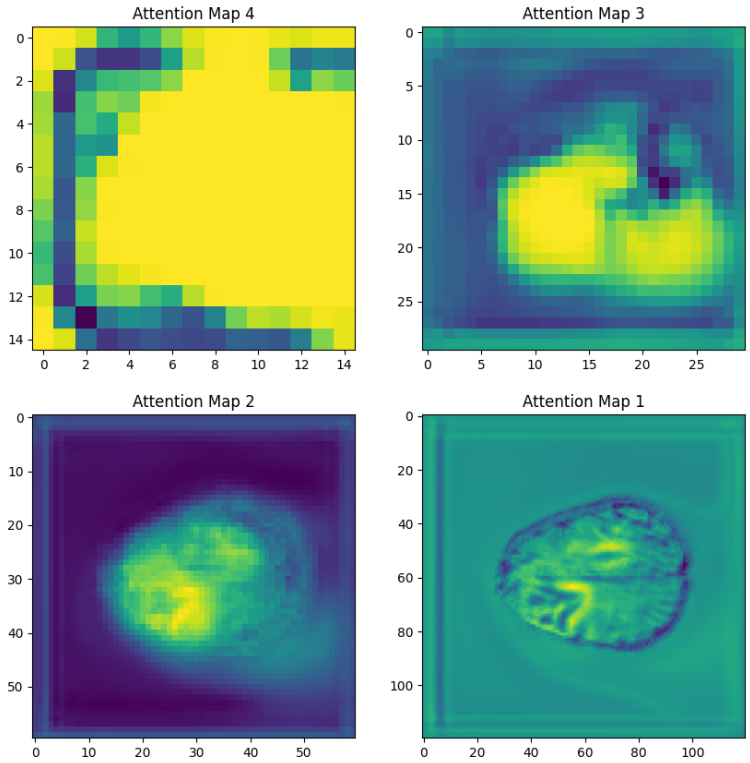

Figure 10: Attention Map for a given MRI scan slice passed through the Attention-UNet model

While the InputCascadeCNN model is performing well for slices with dense tumours, we have observed that huge variation in MRI slices adversely affects performance due to the skewed dataset problem. To take a different approach to this problem, we tried the Attention-UNet model as described in Section 3.2.2.

The performance of the UNet model is greatly boosted by the Attention Mechanism. This can be visualised using the attention maps that the model learns, as shown in Fig 10. Our implementation source code is referenced below [6].

5 Conclusion

After trying out these models and experimenting with their training we were able to get both the InputCascadeCNN and the Attention UNet models to recognize the rough position and shape of brain tumours in the data. Adding Attention mechanism to UNet helped in substantially improving the results. The pixel-by-pixel approach was harder to handle, and required lot of experimentation and manual fine-tuning of the training process in order to get the model to learn the true features of a tumorous patch inspite of the extereme skew in the data. There were some critical trade-offs between the pixel-by-pixel approach and the image-based approach:

- Since the pixel-by-pixel approach takes small patches as inputs, the number of forward passes required to train the model on 1 slice are significantly larger that number of forward passes needed by the image based model, which would consume the 1 slice in just 1 pass.
- If doing two-step training, the balance between training with tumorous data and random data is a very critical hyperparameter, and as per our experiments and literature review, there is no way to automatically strike this balance. A lot of experimentation is needed to get it right.
- Because the image-based approach is basically trying to output a 3 channel RGB image, the model doesn't produce one-hot equivalent outputs. That is, a given pixel could have predicted non-zero magnitudes for more than one label, which is not correct. This is something that could be enhanced in future work with more time and resources.

All in all, both methods produce sensible results, with the Attention UNet being less computationally intensive and easier to train. However, both these models are susceptible to some extent of misclassification of MRI scans, simply because there is a huge variation in the extent of tumors, even across slices of the same patient's MRI scan. That said, even if we are able to provide some form of automated intelligent assistance, it could save many man-hours of medical professionals.

References

[1] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18–31, January 2017.

[2] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas, 2018.

[3] Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig. A brain tumor segmentation framework based on outlier detection. *Medical image analysis*, 8(3):275–283, 2004.

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

[5] SMIR. Brats 2013 dataset. `https://www.smir.ch/BRATS/Start2013`.

[6] GitHub Source code. Achyth Esthuri, Devanshi Panchal, Krish Mehta, Prashil Parekh. `https://github.com/Kkrrish/ece228_brain_tumour_segmentation`.

[7] Nagesh Subbanna, Doina Precup, and Tal Arbel. Iterative multilevel mrf leveraging context and voxel information for brain tumour segmentation in mri. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 400–405, 2014.

[8] Nagesh K Subbanna, Doina Precup, D Louis Collins, and Tal Arbel. Hierarchical probabilistic gabor and mrf segmentation of brain tumours in mri volumes. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013: 16th International Conference, Nagoya, Japan, September 22-26, 2013, Proceedings, Part I 16*, pages 751–758. Springer, 2013.