Brain Tumor Classification and Segmentation

Linoy Menda & Michelle Eidelman

Abstract

Brain tumors are not as common as many other cancers. However, the mortality rate for brain cancers is very high. Diagnosing a brain tumor usually begins with magnetic resonance imaging (MRI) which helps the doctor plan treatment. Therefore, in this project, we build a binary model that classifies between a brain with a tumor and a normal brain. In addition, after creating the model, we will use the segmentation technique to try detecting the specific tumor object and explore if there is a better segmentation tool.

1. Introduction

The data set was taken from kaggle, please find hereunder the link to the data set:
https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection
The data contain 155 images of brain tumors and 98 images without brain tumors.

As you can notice, the data set does not contain many images. However, instead of ignoring small databases, we have taken the challenge of using this small database to try to expand it with tools learned in the course.

2. Related work

There are various segmentation algorithms for object detection in images. Therefore, we have searched articles that related to this subject. One of the articles that we found it explains the importance of segmentation: "Due to the large amount of brain tumor images currently being generated in the clinics, clinicians cannot annotate and segment these images in a reasonable time manually. Hence, automatic segmentation has become inevitable. The requirement for accurate segmentation is essential as the precise location, size, and volume of unhealthy tissue are crucial for treatment, e.g., radiation treatment [1] In this article, the researchers used different segmentation techniques and compared them. The output results of all techniques were compression after segmented with each other by correlation and structural similarity.

In another article [2], the researchers compare segmentation types by the speed and precision of each technique.

These articles help us to understand the importance of segmentation and get to know different working methods.

3. Methods

3.1 Normalization

Pixel values are represented by numbers in the range between 0 and 255.

To prevent problems such as a slowdown in the model's training, we normalized the images and divided the pixels by 255.

The new range of the pixel values is between 0 and 1.

3. 2 Resized

In order to get uniformity we resized all images to 128X128 pixel.

3. 3 Convert to a grayscale

Our dataset contains black and white images, therefore we change the format from RGB to grayscale. We reduced the dimensional from 3 channels to 1 channel.

3.4 Augmentation

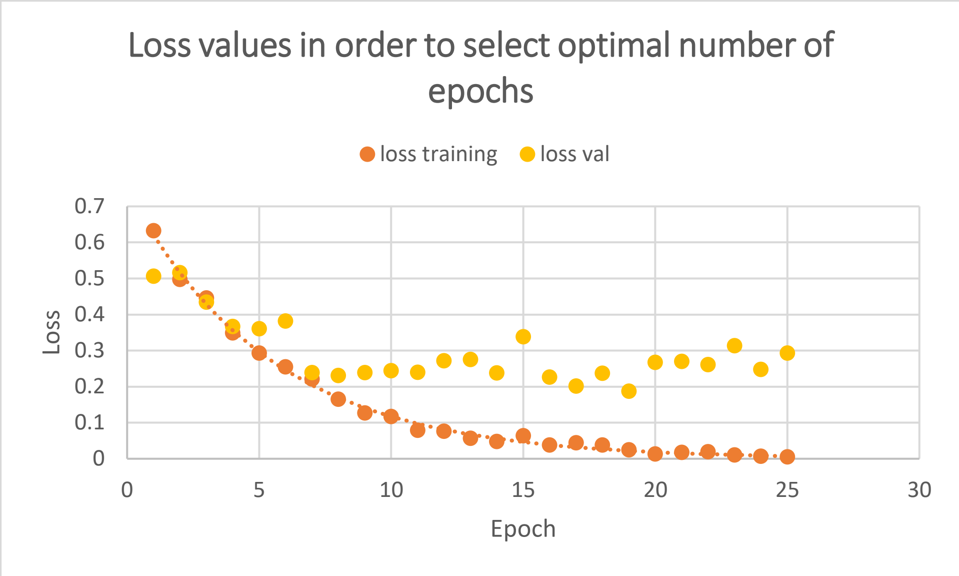
By using augmentation we increased the number of images in the dataset.
We used different types of augmentation, for example: flip horizontal, flip vertical, adding noise.

3.5 CNN model

There are many types of layers used to build Convolutional Neural Networks.
Our model **contains 14 layers** from the following types: Convolutional, Activation, Pooling, Dropout.
The Convolutional layer contains parameters learned during the training process.
In the Activation layer, we chose to use RELU function.
A CNN model that uses RELU function is easier to train and often achieves better performance.
The pooling layer reduces the spatial size of the input volume. Doing this allows us to reduce the number of parameters and computations in the network. Pooling also helps us control overfitting.
The Dropout layer help prevent overfitting by increasing testing accuracy.
Randomly dropping connections ensures that no single node in the network is responsible for "activating" when presented with a given pattern. Instead, dropout assures multiple, redundant nodes will activate when presented with similar inputs — this, in turn, helps our model to generalize.
In out model, we used in a **learning rate of 0.001** as recommended for **Adam optimizer**.
As mentioned in the literature, Adam optimization is a stochastic gradient descent method based on adaptive estimation of the first-order and second-order moments.
In addition, we chose to use **a sparse categorical cross-entropy loss function.**
The formula of the loss function is :

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i).$$

This loss function is used in class classification tasks. These are tasks where an example can only belong to one out of two or many possible categories, and the model must decide which one.
To avoid overfitting, choosing the optimal number of epochs is essential to train the CNN model. After building the network, we train the model until 25 epochs and plot the training loss values and validation loss values against the number of epochs.



As the number of epochs increases beyond 9, the training set loss decreases and becomes nearly to zero. Whereas, validation loss increases depicting the overfitting of the model on training data. As a result, we choose number epochs to be 9.

3.6 Segmentation

In this section we used in 3 types of segmentation of the images that were classify with brain tumor:

1. Threshold segmentation
2. Edge based segmentation
3. Region based segmentation

After applying segmentation to the images, we compared the different type by Dice coefficient:

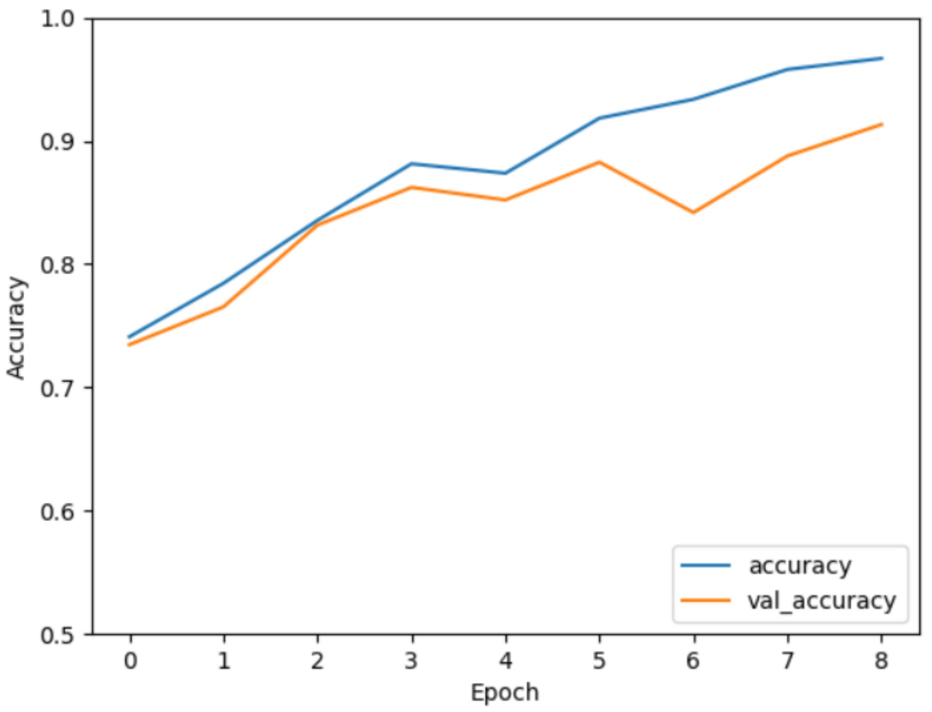
$$Dice = \frac{2\,|A \cap B|}{|A| + |B|}$$

Dice coefficient is essentially a measure of overlap between two samples.
This measure ranges from 0 to 1, where a Dice coefficient of 1 denotes perfect and complete overlap. The Dice coefficient was originally developed for binary data.
In order to use Dice coefficient we first found the target mask(ground truth) of the images by using a website that we found.
$|A \cap B|$ represents the common elements between sets A, the target mask, and B, which is the segmentation's prediction.
$|A|$ represents the number of elements in the target mask, while $|B|$ represents the number of elements in the segmentation prediction.

$|A \cap B|$ is calculated by multiplying the prediction and target mask and then summing the resulting matrix. Also, $|A|$ and $|B|$ are calculated by a simple sum.

4. Experiments

After building CNN model we got accuracy of 0.913.



To evaluate the improvement in our deep learning model's performance, we compared our model to a non-deep learning model.

The non-deep learning model contains one layer (Dense layer), the accuracy of the model is 0.675.

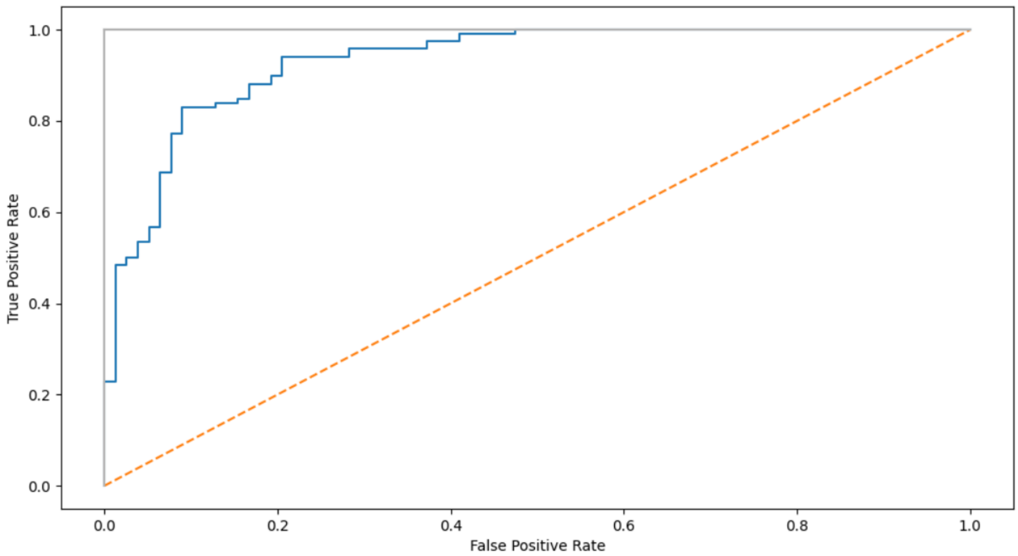
As we show, the CNN model provides better performance (as expected.(

Receiver Operating Characteristic

Roc plot presents the rate between False Positive results and True Positive results.

The area under the ROC curve is also a metric. The greater the area means the better performance.

ROC Curve – for CNN model



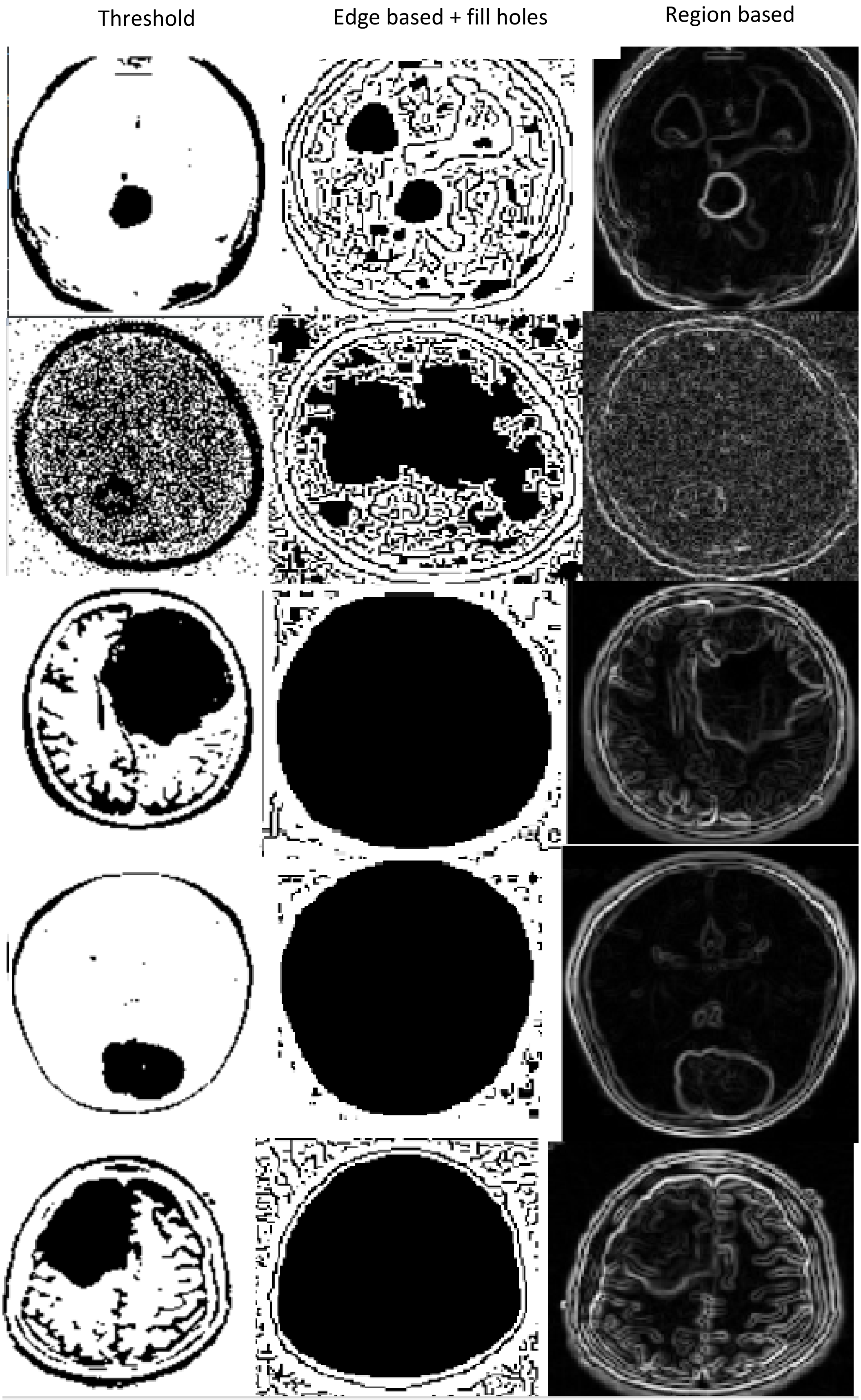
Confusion matrix

We used confusion matrix to gauge the performance of our classification.

True label	**TP = 48.9%**	**FN = 15.8%**
	FP = 2.7%	**TN = 32.6%**
	Predicted label	

After we build the classification model, we chose 5 images that were label with "yes" classification and we apply different types of segmentation.

Results of the compression between the segmentation



Dice coefficient			
score/type	Threshold	Edge-base	Region-base
img1	0.992445	0.37659	0.082971282
img2	0.950663	0.487306	0.212429862
img3	0.974815	0.595336	0.115682915
img4	0.996759	0.781043	0.087580574
img5	0.985533	0.643845	0.135454365
Average	0.980043	0.576824	0.1268238

Analyzing the results

As we mentioned before, CNN model is better than the non-deep classification model that we use. Clearly, networks that contain many layers will yield better results.

Regarding segmentation, the results show that the best segmentation is the threshold technique. After the threshold technique, the edge-base is the next, and the region base is the last.

Edge-base and region base have smaller dice scores due to the calculation of the score.

Dice score is calculated by mask target, which we create with tools from the Internet.

The mask target paints the tumor in black and the background in white. The edge-base and region base contains many black areas in the background, so there is not much overlap between the segmented image and the target mask. That is why the numerator in the dice score will be small and will affect the total score.

5. Conclusion and Future Work

In this project, we created a CNN model that classifies images with and without tumors with very high accuracy.

Furthermore, we compared different segmentation techniques, and the results show that the better approach was the threshold. We discussed why Edge-based and region base might return fewer effective results.

Of course, it is not possible to determine for sure that edge-base and region base techniques are less suitable because we chose to explore only five different images.

In addition, the mask quality should be taken into account because it affects the dice score.

In future work, we must find a better tool to create the mask target, which will bring more reliable results.

On a personal note, we learned a lot from the project by implementing the material learned in the course

.6. References

Code:
https://github.com/michelleeidelman/image_processing/blob/2119bad192251f297576d09c3e90661872128ed9/image_processing.ipynb

Articles:
1.https://www.researchgate.net/publication/337742433_Comparison_of_Different_Image_Segmentation_Techniques_on_MRI_Image
2. http://article.sapub.org/10.5923.j.ajbe.20160602.03.html#Sec3.1
Tools for create mask target:
3. https://www.apeer.com/app/dashboard
4. https://www.dcode.fr/binary-image
Another Information
5. https://www.jeremyjordan.me/semantic-segmentation/
6. https://www.upgrad.com/blog/basic-cnn-architecture/
7. https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea
8. https://www.tensorflow.org/tutorials/images/segmentation