Efficient Detection of Tumorous Tissue in Brain using Machine Learning

07.2020

Md. Sadek Hossain Asif

Safwan Shahid Labib

Md. Nazmus Sakib

Notre Dame College, Dhaka

Registration Id : 3611173

Abstract

Brain tumor is one of the most fearsome medical problems globally today. With the advent of various natural factors, this is among one of those medical problems that is seeing a significant rise in happenstance. Our project aims to utilize the analytical power of neural networking in engineering a method for accurate detection of tumorous tissue in magnetic resonance (MR) images. Based on a type of network known as Deep Neural Network (DNN), our solution proposes, on a limited scale, to identify low and high grade glioblastomas (a type of tumor) pictured in a cerebral MRI scan. These tumors can appear anywhere in the brain, and have variable shapes and contrasts in an MRI, which translates to the requirement of a machine learning solution for detection of these phenomena that exploits a flexible and high capacity DNN, while at the same time, being extremely efficient. Our aim is to produce a rough-edged, efficient, but effective assistive tool that learns as it is used for radiologists' use in machine-assisted detection of tumors, in order that the physician has a time-effective and accurate tool in their disposal in diagnosis and subsequent treatment of this life-threatening malady.

Table of contents

Hypothesis

Tumorous tissue is most commonly detected in MRI scans utilizing glucose usage differentials. Tumorous tissue, being of the nature that it rapidly reproduces without control from the cell's normal mechanisms for regulating cellular division, is a prolific user of glucose and water, among other necessary nutritions. Resultantly, the comparatively higher use of resources shows up on an MR image. These differences are often subtle and require qualitative decisions on part of the radiologists. We propose a sophisticated yet low-volume neural network using cascading architecture, however in an unconventional approach. Our Cascading Neural Network (CNN) proposes to exploit both local features as well as more global contextual features simultaneously. Also, different from most traditional uses of CNNs, our network also proposes to use a final layer that is a convolutional implementation of a fully connected layer; this allows high speed data processing. As this program is a network that will 'learn', as in self-program its refinements in image processing and detection, we propose a 2-phase training procedure that allows us to tackle difficulties related to the imbalance of tumor labels. Finally, we propose to explore a cascade architecture in which the output of a basic CNN is treated as an additional source of information for a subsequent CNN. The import of this multi-layered approach shall hopefully translate into high accuracy in data processing, implementation, and final application in the detection process.

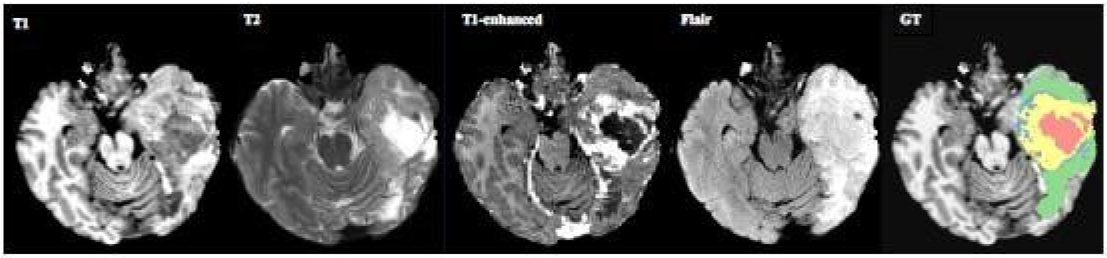
Required Technologies

```
- Keras 2.2.4
- Tensorflow 1.13.1
- Matplotlib
- SimpleITK
- Numpy
```

Source of training data: BRaTS 2015, 2017, 2018

What is BRaTS Dataset?

We have used BRaTS 2015, 2017 and 2018 training dataset for analysis and training of the proposed methodology. It consists of real patient images as well as synthetic images created by SMIR. Each of these folders are then subdivided into High Grade and Low Grade images. For each patient, four modalities (T1, T1-C, T2 and FLAIR) are provided. The fifth image has ground truth labels for each pixel. The dimensions of the image are different in LG and HG. For HG, the dimensions are (176,261,160) and for LG are (176,196,216).



Dataset 2

This dataset contains 3064 T1-weighted contrast-enhanced images with three kinds of brain tumor. Reference and details regarding the dataset can be found on this site. Version 5 of this dataset is used in this project. Each image is of dimension 512*512*1 , these are black and white images thus having a single channel.

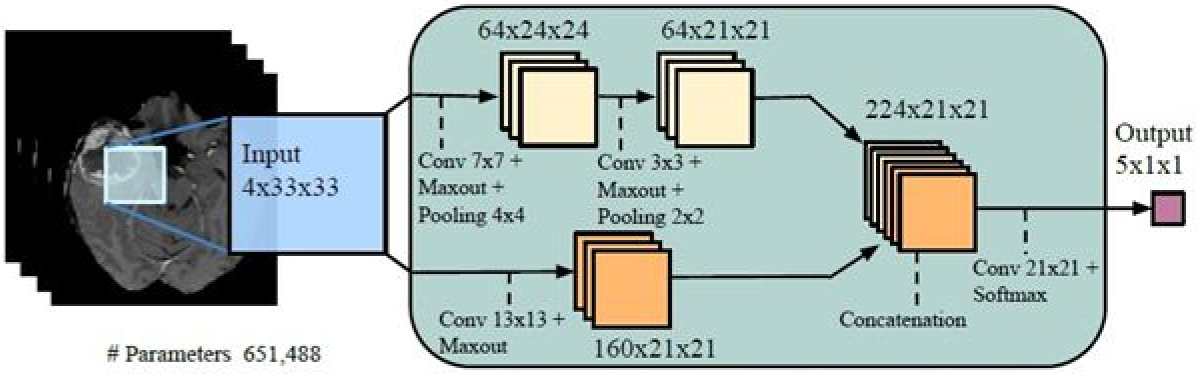
Samples of data[1]

[1] As per the requirement of the algorithm, slices with the four modalities as channels are created. For taking slices of 3D modality images, We have used 2D rendering. At time of training/testing, we need to generate patches centered on the pixel which we would be classifying. We are ignoring the border pixels of images and taking only inside pixels. Generating a dataset per slice. We am filtering out blank slices and patches. Also, slices with all non-tumor pixels are ignored.

Model Architecture

I. TwoPathCNN

TwoPathCNN represents the 2 paths an input patch has to go through. The 1st path, where 2 convolutional layers are used, is the local path. The 1st convolutional layer is of size (7,7) and 2nd one is of size (3,3). The 2nd and global path consists of (21,21) filter. As the local path has smaller kernel, it processes finer details because of small neighborhood. Opposed to this, global path processes data in more global way. After the convolutional layer, Max-Out [Goodfellow et.al] processing is used, after which data max-pooling is used with stride '1'. We have changed the max-pooling to convolution with same dimensions in the global path, after convolution max-out is carried out. There is no max-pooling in the global path. After activation is generated from both paths, both data pieces are concatenated and final convolution is carried out. Then Softmax activation is applied to the output activations. As there is no fully-connected layers in model, the number of parameters is substantially smaller than standard image processing systems, and very high-speed computation is made possible.



II. Cascading Architecture

Cascading architecture uses TwoPathCNN models joined at various positions. The paper defines 3 of them -

- InputCascadeCNN: 1st's output joined to 2nd's input.
- LocalCascadeCNN: 1st's output joined to 2nd's hidden layer (local path 2nd conv input.
- MFCCascadeCNN: 1st's output joined to 2nd's concatenation of two paths.

Data is also augmented, as data needs to be used as efficiently as possible in order to increase accuracy levels. Above 96% accuracy is predicted based on initial testing.

III. Data Augmentation

The basic forms of data augmentation are used here to diversify the training data. All the augmentation methods are used from Pytorch's Torchvision module.

- Horizontal Flip
- Vertical Flip
- Rotation Between 75°-15°

Training the network

Loss function

Loss function is defined as 'Categorical cross-entropy' summed over all pixels of a slice. We have modified the loss function in 2-ways:

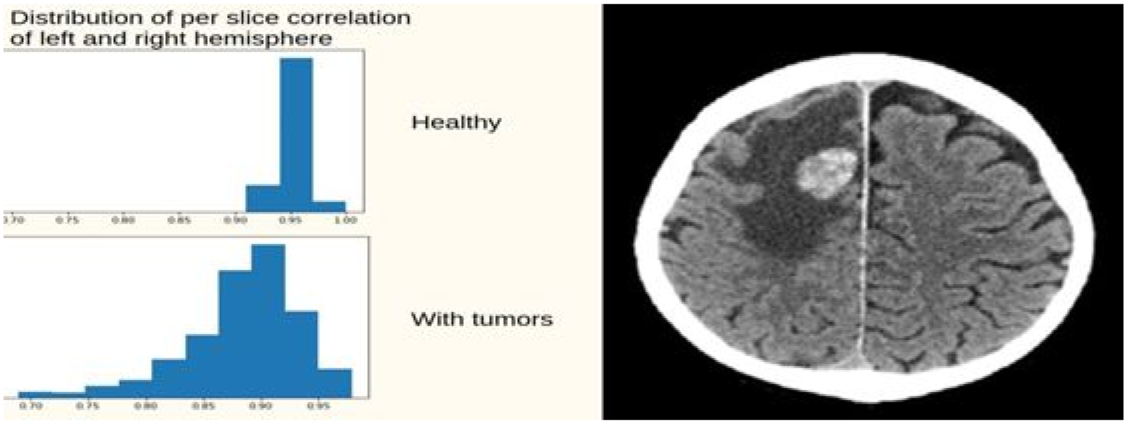
- The dataset per slice is being directly fed for training with mini-batch gradient descent i.e., we are calculating and back-propagating loss for much smaller number of patches than the whole slice.
- For each dataset, we are calculating weights per category, resulting into weighted-loss function. This is taken as measure to skewed dataset, as number of non-tumor pixels mostly constitutes dataset.

Regularization

The paper uses drop-out for regularization. We have used batch-normalization, which is used for regularization also. Batch-normalization helps training because it smoothens the optimization plane; This helps in stable gradients and faster reaching optima. When training without regularization and weighted-loss function, we found out that model gets stuck at local optima, such that it always predicts 'non-tumor' label. After utilizing both the techniques, the accuracy of the model increases notably.

Results

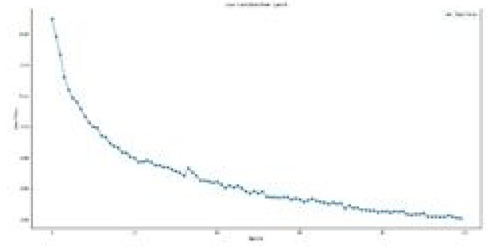
As the dataset is very large because of patch-per-pixel-wise training scheme, we have not been able to train the models on all of the dataset. For now, both cascading models have been trained on 4 HG images and tested on a sample slice from the new brain image. We have computed f-measure for the complete tumor region.

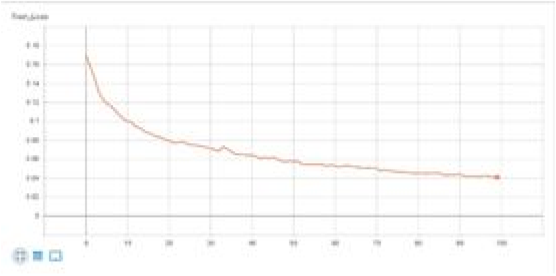


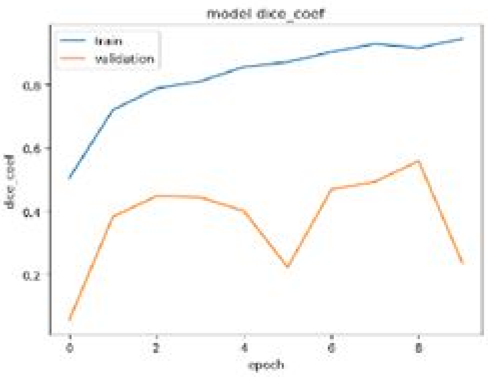
Complete score: labels 1+2+3+4 for patients data.

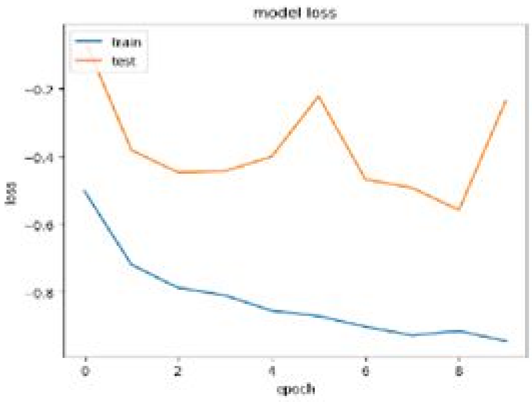
Slice Number	F1-Score (complete)	
(HG 0027)	InputCascadeCNN	MFCcascadeCNN
105	0.9250	0.80091
106	0.9271	0.8029
107	0.9269	0.8085
108	0.9280	0.8114
109	0.92691	0.8056
110	0.9277	0.7986
111	0.9291	0.7929
112	0.9297	0.7868
113	0.9273	0.79228

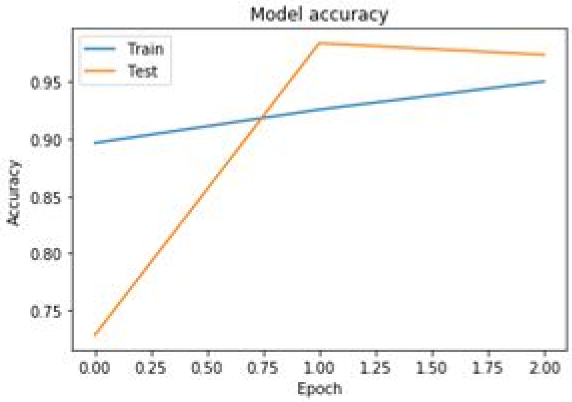
Graphs

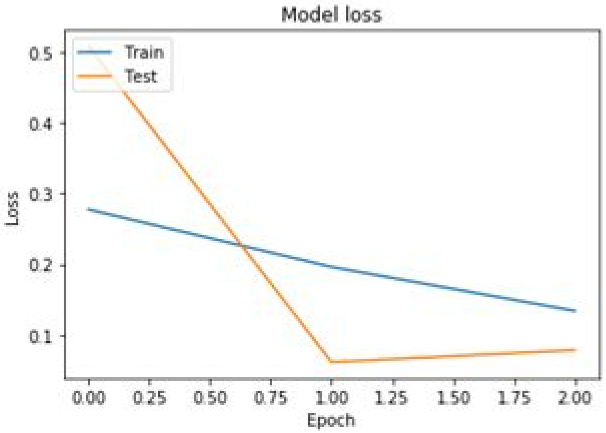





model dice_coef
train
validation
dice_coef
0.2
0.4
0.6
0.8
0
2
4
6
8
epoch


model loss
train
test
loss
−0.2
−0.4
−0.6
−0.8
0
2
4
6
8
epoch


Model accuracy
Train
Test
Accuracy
0.75
0.80
0.85
0.90
0.95
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Epoch


Model loss
Train
Test
Loss
0.1
0.2
0.3
0.4
0.5
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Epoch

Conclusion

The mean Dice Score our model gained was 0.74461 in testing dataset of 600 images. From this we can conclude that in our testing dataset our constructed mask has a similarity of about 74% with the original mask.

Some samples from our training dataset output are below. The top best results can be found here :
https://drive.google.com/drive/folders/1vwwUipaH9Yb0NLelv3lW-04E6WnVJ3nh?usp=sharing

References

1. Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, Hugo Larochelle, “Brain Tumor Segmentation with Deep Neural Networks”. Cornell University Arxiv, arXiv:1505.03540 [cs.CV]
2. B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, et al. "The Multimodal Brain Tumor Image Segmentation Benchmark (BRaTS)", IEEE Transactions on Medical Imaging 34(10), 1993–2024 (2015) DOI: 10.1109/TMI.2014.2377694
3. S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J.S. Kirby, et al., "Advancing The CancerGenome Atlas glioma MRI collections with expert segmentation labels and radiomic features", Nature Scientific Data, 4:170117 (2017) DOI: 10.1038/sdata.2017.117
4. S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, et al., "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRaTS Challenge", arXiv preprint arXiv:1811.02629 (2018)
5. S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, et al., "Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-GBM collection", The Cancer Imaging Archive, 2017. DOI: 10.7937/K9/TCIA.2017.KLXWJJ1Q
6. S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, et al., "Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-LGG collection", The Cancer Imaging Archive, 2017. DOI: 10.7937/K9/TCIA.2017.GJQ7R0EF
7. U-Net: Convolutional Networks for Biomedical Image Segmentation, https://arxiv.org/abs/1505.04597
8. Intelligent Scanning Using Deep Learning for MRI, https://medium.com/tensorflow/intelligent-scanning-using-deep-learning-for-mri36dd620882c4
9. VGG Image Annotator
10. Mask RCNN
11. github.com/zsdonghao/u-net-brain-tumor/
12. github.com/JooHyun-Lee/BraTs