Brain tumor detection and segmentation using VGG-16 and U-net, advanced data augmentation techniques and hyperparameter tuning. (DA-202 course project)

Bhumit Arora
bhumit_a@mfs.iitr.ac.in
22125010

VaibhavLohia
vaibhav_l@mfs.iitr.ac.in
22125038

Abstract

The project presents a classification and segmentation system for lower-grade gliomas (brain tumor) based on two CNN architectures (VGG-16 and U-net). Transfer learning is used with VGG-16 for the classification task. The LGG Segmentation dataset is available on Kaggle has been used for both tasks. Different data augmentation techniques have been used to make the results more accurate, as our dataset contained a very small number of training examples. The results indicate that the pre-trained VGG-16 model can now classify whether there is tumor with an accuracy of 89%. For the segmentation task, the model can now tell the location of the brain tumor with around 0.81 dice coefficient.

1. Main Objectives

- Classification of brain tumor using VGG-16 pre-trained model.
- Using U-net to segment the tumor region in the tumor classified images.

2. Status and other details

- Completed
- Total time spent on the project: 4 weeks

3. Introduction

According to the Ten Leading Causes of Death Statistics Report by the Ministry of Health and Welfare in 2021, cancer ranks as the leading cause of death nationwide. Among them, pleomorphic glioblastoma, also known as multiple morphologic glioblastoma, is a common brain cancer. This tumor can be categorised into lower-grade glioma and higher-grade glioma. Lower-grade gliomas have a lower risk, whereas higher-grade gliomas are malignant tumors characterised by fast growth and invasive properties.

This lower-grade glioma brain tumor, although better manageable than their higher-grade counterparts, often occurs in younger adults and eventually progresses to a fatal stage with an average survival of 7 years after diagnosis. Therefore, it is vital that they can be accurately detected from MRI images. According to a study, skilled radiologists reach accuracies of between 92 and 95% in detecting tumor from MRI scans (Yan et al., 2016).

For this project, the data used comes from 110 patients who had a lower-grade glioma brain tumor from The Cancer Genome Atlas (TCGA) and consists of images obtained from The Cancer Imaging Archive (TCIA) as well as manually created fluid-attenuated inversion recovery (FLAIR) masks. The FLAIR technique for MRIs is meant to suppress darkening cerebrospinal fluid effects on the images, which brings out the contrast between the pixels. There are 3929 images in the data set (Buda, Saha & Mazurowski, 2019). We have used advanced data augmentation techniques such as affine and elastic transformation.

This project not only aims to detect tumors in an MRI but also to segment the part of the brain where the tumor is located. Image segmentation techniques have been applied which cluster the parts of the brain images into tumor or non-tumor areas. To evaluate the performance of this segmentation task, we use the Dice coefficient, a metric of similarity between a predicted segmentation and the ground truth, which is most often used for this type of task. We have used U-net architecture which is often used for this type of tasks in which pooling operations are replaced by upsampling operations.

4. Dataset

We have used the LGG Segmentation dataset, which is available on Kaggle. Link: (https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation).

This dataset contains brain MRI images together with manual FLAIR abnormality segmentation masks.

The images were obtained from The Cancer Imaging Archive (TCIA). They correspond to 110 patients in The Cancer Genome Atlas (TCGA) lower-grade glioma collection with at least fluid-attenuated inversion recovery (FLAIR) sequence and available genomic cluster data. Tumor genomic clusters and patient data are provided in the data.csv file.

5. Data preprocessing

All images (110 total) were provided in .tif format format with 3 channels per image. For 101 cases(patients), 3 sequences were available, i.e. pre-contrast, FLAIR, and post-contrast (in this order of channels). For 9 cases, the post-contrast sequence was missing, and for 6 cases, the pre-contrast sequence was missing. Missing sequences were replaced with FLAIR sequences to make all images 3-channel. Masks are binary, 1-channel images.

We combined the brain scans with their corresponding 'masks' in a dataframe. Next, we used a function to determine which masks contained a tumour and which were empty. Then, we applied the function to the masks in the data frame and added a column with 0s and 1s, where 0 indicates a mask with no tumour, and 1 indicates a mask with a tumour. This dataframe is needed for the classification task because then the model can get the brain scans as input and train the column with the 0s and 1s as output.

6. Data augmentation

We have elected several data augmentation techniques that have most often and most successfully been used in similar tumour classification and segmentation tasks on the BraTS18 data set, dubbed a "standard benchmark for validating existing and emerging brain-tumour detection and segmentation techniques”. We used affine and elastic transformations. With affine transformations, images are flipped, cropped, rotated, and zoomed to produce more images.

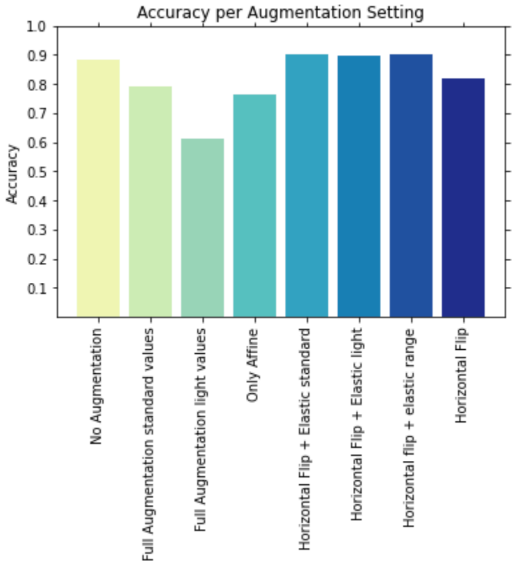
We used the following techniques only:

* Flipping (horizontal, because the left and right lobe of a brain are symmetrical looking),

* Changes in width, height, and a subtle zoom because not all brains on the MRIs are of the same size, and height and width adjustments of the brain's positions on the scan, so the model learns location invariance.

Then, the elastic transformation was used for shape variations. It draws a gridline over images and distorts the image along those lines.

We tuned the augmentation settings to try and find the highest accuracy whilst not overfitting too much. The augmentation techniques that gave the best accuracies compared to the model trained on un-augmented data were horizontal flips and elastic transformations. The rest of the affine techniques (zoom and shift) gave lower accuracies than unaugmented data.



7. Classification using VGG-16

Classifying based on the MRI images whether image contains tumour or not.

The VGG-16 was used as the pre-trained base, loaded from the tensor flow pre-trained on the imagenet.

Then, we added the classifier head to this pre-trained model. In the classifier head, the Flatten layer transforms the base's two-dimensional outputs into the one-dimensional inputs the head needs.

Added Regularisation: The drop-out layer omits some of the neurons at each step from the layer, making the neurons more independent from the neighbouring neurons. It helps in avoiding overfitting. We tried different drop-out parameters, between 0.2 and 0.5, but the parameters used in our code were given the best results.

The Dense layer is the output layer, classifying the image into 1 of the 2 possible classes using Sigmoid as the activation function.

Out of binary and categorical cross entropy, we used binary cross entropy as our cost function, with sigmoid as the activation function with a scalar target and not a one-hot encoded target.

The binary cross entropy is defined as

$$\prod_{c=1}^{C}(y_c(x, w_c))^{t_c}(1 - y_c(x, w_c))^{(1-t_c)}$$

And categorical cross entropy is defined as

$$\prod_{c=1}^{C}y_c(x, w_c)^{t_c}$$

As an optimizer, Adam is used. Adam is an SGD algorithm that has an adaptive learning rate that makes it suitable for most problems without any parameter tuning.

Adam is nothing but a combination of RMSProp, momentum and bias correction.

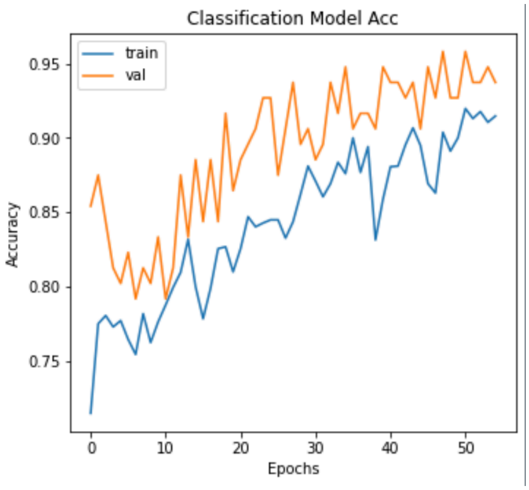
It is the best neural network optimisation algorithm.

$$m_t = \beta_1 * m_{t-1} + (1-\beta_1) * \nabla w_t$$
$$v_t = \beta_2 * v_{t-1} + (1-\beta_2) * (\nabla w_t)^2$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} * \hat{m}_t$$

We also used certain callbacks like **EarlyStopping** (to avoid overfitting), **modelcheckpoint** (helps to save a model by monitoring a specific parameter) and **ReduceLRonPlateau** (reduce learning rate when there is very less improvement).

Training was initialised for 100 epochs but early stopped after 25 epochs.

We got a **test accuracy of 89%** in the classification task.



8. Segmentation using U-net

The difference between segmentation and classification is that in classification, we assign a single label for each image, but in classification, we assign a label for each pixel value. Hence, the output is the localisation of the object.

We will be using a fully convolutional neural network with the U-Net architecture. The U-Net architecture is chosen because it can first convert an image into a vector and then convert it back into a segmented image (Ronneberger et al., 2015).

This architecture can work with a lot fewer training images and can give more precise segmented outputs. A recent research paper has shown that when a classification model is used along with a segmentation model, it is better that the classification model first classifies an image to determine whether there is a tumor. Then, according to conventional methods we have to train the segmentation model using only those images with tumors (Li et al., 2021). This increases the accuracy of the segmentation task.

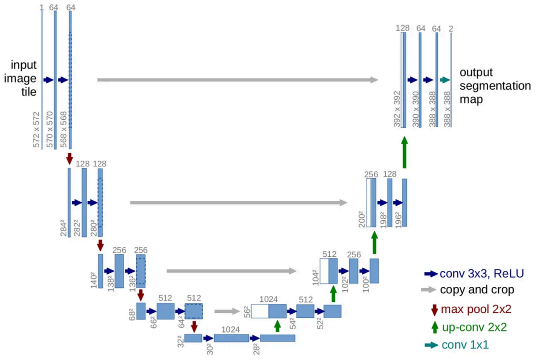
However, we have decided to use all scans to train the segmentation model because we are also curious whether our segmentation model will detect and localise tumours in scans that do not contain any. This could mean that patients may be diagnosed unnecessarily, which should be avoided as much as possible. That is why we chose to use all scans in the segmentation model.

8.1 U-net architecture

The U-Net below consists of a contracting path, the bottleneck and an expansive path. The contracting path is similiar to a normal convulational network. Each block begins with a 3x3 convolutional layer, followed by a rectified linear unit (Relu) activation function, then again a 3x3 convolution layer, which is followed by a batch normalization and a Relu function. A batchnormalization is added to a network to normalize each batch with its own mean and standard deviation. This can help the training, the optimization process and sometimes the prediction performance. At last, there is a 2x2 max pooling operation added in each contracting block. Also, after each block, the number of feature maps is doubled, so the model can train complex structures.

The second part of the U-Net is the bottleneck. This bottleneck combines the contracting path with the expansive path. In this U-Net the bottleneck has the same content as the blocks in the contracting path, but it doesn't have the pooling operation.

The last part of the U-net is the expansive path. Each block has a 2x2 transpose convolutional layer performing inverse convolution operation. So, the number of feature maps is being halved. There is also a concentration with the corresponding cropped feature map from the block in the contracting path included, which ensures that the gradients can flow freely through the model and that the features can be transmitted from the contracting to the expansive path, which can help with adding more information for the training. Next, the structure is the same as for the blocks in the contracting path: a 3x3 convolutional layer, followed by a Relu function, then again a 3x3 convolution layer, which is followed by a batch normalization and a Relu function. The final layer consists of a 1x1 convolution layer with a sigmoid activation to produce an output with the same size as the input.



UNet architecture

8.2 Metrics

Dice coefficient: Generally in case of image segmentation tasks, dice coefficient is used to measure the similarity. It is defined as:

$$\frac{2 * areaofoverlap}{totalpixels}$$

To smoothen the dice coefficient and avoid division by zero, we added a smooth term with a value of 100.

Smooth = 100

So the dice coefficient becomes:

$$\frac{2 * areaofoverlap + smooth}{totalpixels + smooth}$$

Intersection over Union(IoU): IoU is the ratio of the intersection area between the predicted region and the ground truth region to the area of their union.

It is defined as:

$$\frac{areaofintersection}{areaofunion}$$

After smoothening:

$$\frac{areaofintersection + smooth}{areaofunion + smooth}$$

8.3 Loss

Loss used is dice coefficient loss which I defined as

-(dice_coeff(y_true,y_pred))

8.4 Training

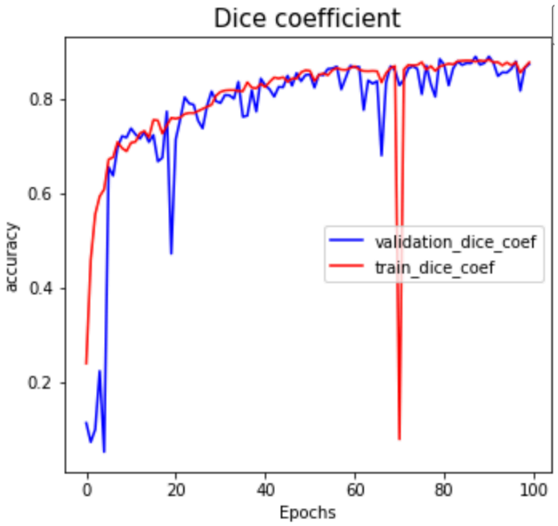
The model was trained for 100 epochs using 2828 images.

Kaggle's P100 GPU was used for this task.

And then, it was evaluated on 393 images, and the accuracy came out as:

Test IoU: 0.9979254007339478

Test dice coefficient: 0.81458407640



9. References

[1] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[2] Ronneberger, Olaf & Fischer, Philipp & Brox, Thomas. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. LNCS. 9351. 234-241. 10.1007/978-3-319-24574-4_28.

[3] Yan, Zhengbing & Han, Wenxuan & Penuelas, Josep & Sardans, Jordi & Elser, James & Reich, Peter & Fang, Jingyun. (2016). Yan et al-2016-Ecology Letters.sup-1.

[4] LGG segmentation dataset (https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation)

[5] Buda M, Saha A, Mazurowski MA. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. Comput Biol Med. 2019;109:218-225. doi:10.1016/j.compbiomed.2019.05.002

[6] Krishnan, Praveen & Jawahar, C.. (2019). HWNet v2: An Efficient Word Image Representation for Handwritten Documents. International Journal on Document Analysis and Recognition (IJDAR). 22. 10.1007/s10032-019-00336-x.

[7] Jabir, Brahim & Noureddine, Falih. (2021). Dropout, a basic and effective regularization method for a deep learning model: A case study. Indonesian Journal of Electrical Engineering and Computer Science. 24. 1009. 10.11591/ijeecs.v24.i2.pp1009-1016.

[8] Ruby, Usha & Yendapalli, Vamsidhar. (2020). Binary cross entropy with deep learning technique for Image classification. International Journal of Advanced Trends in Computer Science and Engineering. 9. 10.30534/ijatcse/2020/175942020.

[9] Nabiyev, Nabi & Malekzadeh, Saber. (2021). Anomalous Sound Localization Estimation. 10.13140/RG.2.2.25949.95201.

[10] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[11] Kar, Mithun. (2021). A Review on Progress in Semantic Image Segmentation and Its Application to Medical Images. SN Computer Science. 2. 10.1007/s42979-021-00784-5.

∈ ℝ