Brain Tumor detection

Harshith Sheshan (sheshan.h@northeastern.edu), Shiyao Wang (wang.shiyao@northeastern.edu), Albert Sankoh (sankoh.a@northeastern.edu), Michael Hart (hart.mi@northeastern.edu)

Abstract

Brain tumor is the abnormal growth of cells in the brain. The tried and tested method to detect a tumor by(Magnetic Resonance Imaging)MRI scans has proven to be a great help only in the presence of a specialist doctor who has spent years in studying and has accumulated enough experience to be able to detect a tumor from the scans. There have been multiple Machine learning models which have been used to solve this problem. In the proposed work we are trying to improve the efficiency of thede models with the help of Convolutional neural Networks and Deep encoders-decoders.

Introduction

Brain tumors are uncontrolled cell proliferations. Some are primary because they originate in the brain. Others, known as secondary, spread to this location from elsewhere in the body via metastasis. Primary brain tumors do not spread to other parts of the body and can be either malignant or benign.

Recent advancement in artificial intelligence and machine learning with competing accuracy to human has found the use of this technology extensively in healthcare. The market value of Artificial intelligence in healthcare is expected to reach 45.2 billion USD (US Dollars) by 2026. The problem we have chosen to solve is related to improving the accuracy and efficiency of detecting a localizing brain tumor based on MRI scans with the help of CNN (convolutional Neural Networks) and Deep encoders-decoders.

Integrating AI into the healthcare ecosystem provides numerous benefits, including the ability to automate tasks and analyze large patient data sets to provide better healthcare faster and at a lower cost. According to statistics, radiologists are now reading 12 MRI images per minute compared to 3 a decade ago which makes way for finding innovative and efficient methods to diagnose without increasing the stress on radiologists.

Literature review

In recent years, researchers have proposed various automated systems to assist the diagnosis and stage classification of brain tumors. In 2012 Rathi et.al performed feature selection on 140 brain MRI images, with Principal Component Analysis (PCA) and Linear Discriminant Analysis

(LDA) applied on the training set, and Support Vector Machine (SVM) classifier comparing nonlinear vs. linear technique [1]. In 2017 Lavanyadevi et.al extracted statistical and texture features from the brain images with the Gray Level Co-occurrence Matrix (GLCM), achieving automated brain tumor stage classification with probabilistic neural network (PNN)[2]. Kumar et.al attempted the Particle Swarm Optimization (PSO) for feature selection hybrid with SVM classifier, which yielded a higher accuracy score compared to the simple SVM classifier [3]. Recently Sharma et.al did comparative analysis of Convolutional Neural Network (CNN) based transfer learning models, evaluated SVM classifier, Random Forest Classifier, VGG-16, Inception-V3 and ResNet architectures [4].

According to Sharma's work, among all the architectures of CNN, the ResNet architecture had good performance in tumor classification, with highest training accuracy of 99.7% and validation accuracy of 82%.

Methodology

In this project, our primary research question is to train and tune a deep learning model that is able to predict whether a brain MRI image indicates that the patient has a tumor.
And with reference to previous researchers' work, we came up with our own design, which is a CNN with ResNet architecture trained on a primary dataset, the LGG brain tumor MRI image data, and then verify the model on a second brain MRI image dataset.

Our primary dataset is the LGG Segmentation Dataset, which contains 3929 brain MRI images from 110 patients, together with manual FLAIR abnormality segmentation masks of each image.

Images and masks are organized in 110 folders with naming convention of
'TCGA_<institution-code>_<patient-id>_<slice-number>'

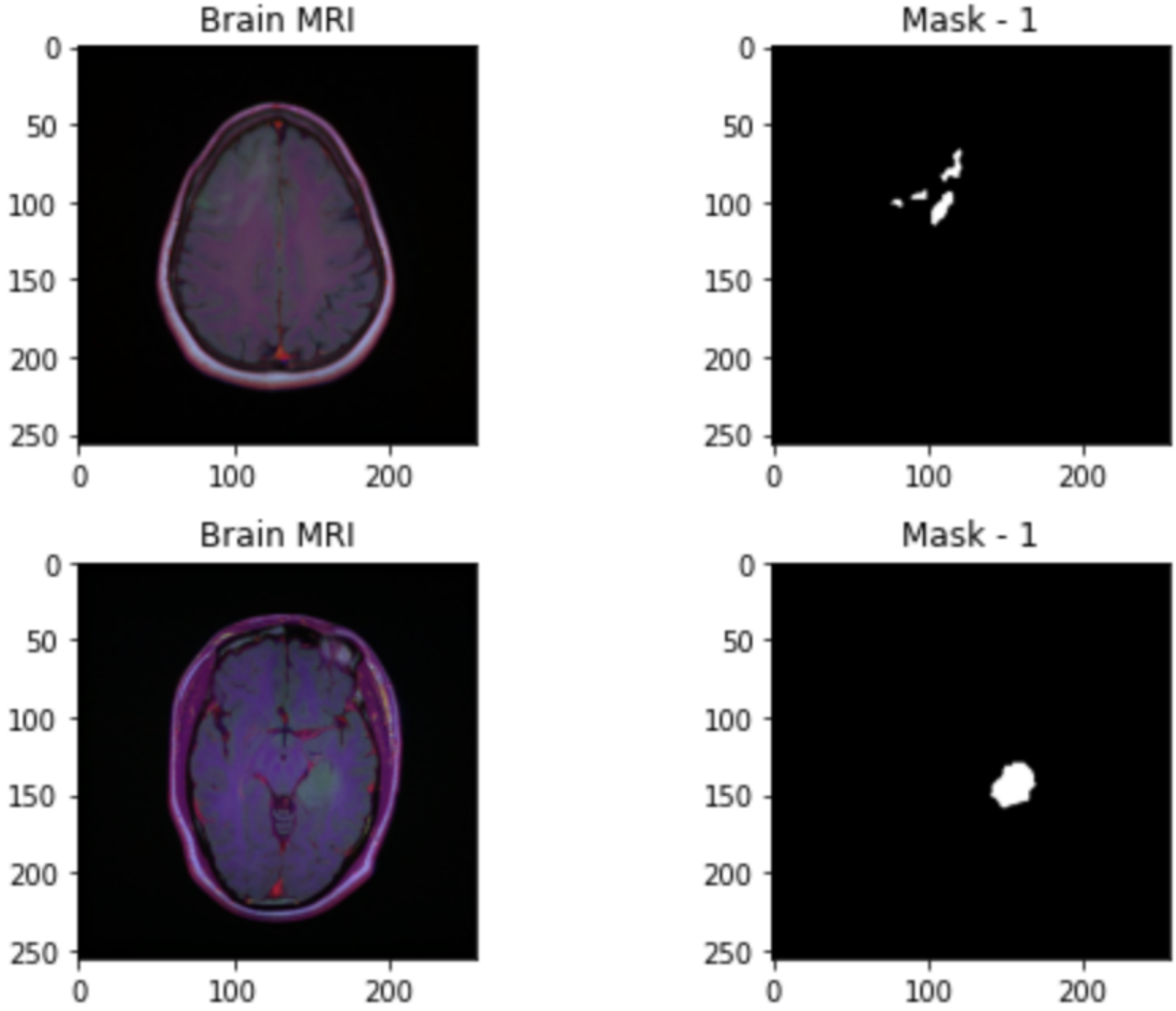
All images are provided in '.tif' format with 3 channels per image. Masks are binary, 1-channel images, named the same with the corresponding image with a '_mask' suffix.

There is also a lookup table 'data_mask.csv' that stores the paths of each image and the mask corresponding to it.

Columns of the table include:

- patient_id: the id of the patients, will be dropped in preprocessing since our analysis regards each image as a datapoint
- image_path: the path to an image
- mask_path: the path of above image's mask
- mask: a binary variable
  - 1: if the mask has a light region, displaying a existing tumor
  - 0: if the mask is empty, saying no tumor seen on this MRI image

Attached is a snapshot of our visualized input data, with the MRI image on the left and a mask on the right. The masks are measured and provided by the researchers in hospitals who initially performed the medical scan and collected the data.



The secondary dataset for verification purpose is more straightforward and relatively light-weighted, that is composed of 155 MRI images with tumor, and 98 without tumor.

Our base architecture is a ResNet model. Layers are presented in detail as we defined it.

```
basemodel = ResNet50(
    weights = 'imagenet',
    include_top = False,
    input_tensor = Input(shape=(256, 256, 3)))
```

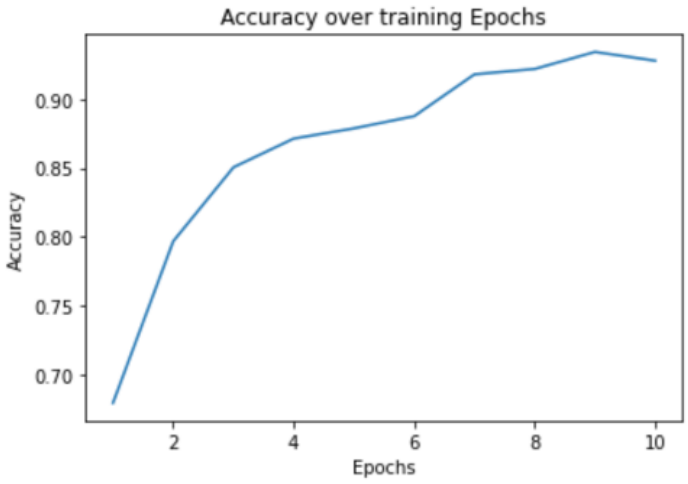
```
# Add classification head to the base model

headmodel = basemodel.output
headmodel = AveragePooling2D(pool_size = (4,4))(headmodel)
headmodel = Flatten(name= 'flatten')(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)#
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(2, activation = 'softmax')(headmodel)

model = Model(inputs = basemodel.input, outputs = headmodel)
```

We splitted the primary dataset into train, test and validation model by 70:15:15, extracted features from image pixels using tensorflow library ImageDataGenerator, trained the ResNet50 base model, performed predictions on test and validated with the validation set. Result scores are visualized in the confusion matrix.

Then we processed the secondary dataset similarly with ImageDataGenerator to get the features, and built the index table similar to that of the primary set. Predictions are made by the trained model and accuracy scores are recorded.

Results

We evaluated our model based on the accuracy, precision, and recall. After training the model through 9 epochs we achieved an accuracy of 0.98, precision of 0.98, and a recall of 0.96. This means we are able to use the trained model to predict whether an image in the test dataset has a brain tumor or not correctly 98% of the time.



The model converged after 10 epochs. Final Results:

Accuracy	Precision	Recall	F1 Score
0.98	0.98	0.97	0.97

Of the 590 predictions made using the test dataset, 98% of instances were predicted correctly:

	Negative Prediction	Positive Prediction	Total
True Negative	359	4	363
True Positive	10	217	227
Total	369	221	590

False positive rate: 1.1%
False negative rate: 4.4%
Specificity: 98.9%
Sensitivity: 95.6%

A false negative is the worst outcome as it means a patient would be told they do not have a tumor when in fact they did have a tumor. The model achieved a false negative rate of 4.4%.

Future scope

As we can observe that when we run the model against other datasets which were not used for training our model, there is a significant drop in the performance. To overcome this drawback and to improve the versatility of our model we can improve the model.

The following are some of the steps which can be taken towards this direction.

- The model can be further trained with other datasets extending the compatibility and improving accuracy of the model when handling classification.
- We can also preprocess the images to reduce the impact of difference in chromaticity and the resolution of the scans on the result.

Conclusion:

The main goal of this research work was to design efficient automatic brain tumor classification with high accuracy using deep learning and not classic algorithms (such as SVM, KNN...), which are old algorithms that required feature extraction and such old techniques don't give accurate results compared to the new approach (deep learning).

In this work we used a well-known technique called Transfer learning : A pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task such as Image Net Data set. The weights of the pre-trained model can be utilized for the classification of another task. You don't have to train your model from scratch.

In our case the large dataset is "ImageNet" (The ImageNet dataset is a very large collection of human annotated photographs designed by academics for developing computer vision algorithms) and The pre-trained model is "Resnet-50" (ResNet-50 is a convolutional neural network that is 50 layers deep).

We used 2 different datasets (let's name them D1 and D2).

D1 contains 3929 image samples of the human brain, belonging to 2 classes
(0 : without tumor, 1 : with tumor).

We used transfer learning with the dataset and pre-trained model mentioned above (resnet50 and ImageNet), and then we added new dense layers which we trained after freezing the base model layers (resnet50) on D1 and then we got about 91% accuracy, and then used a full-pre-trained model with its weights (files "resnet-50-MRI.json" and its weights "weights.hdf5") which performs well on brain tumor images where we got about 98% accuracy.

D2 contains 3929 image samples of the human brain, belonging to 2 classes
(No : without tumor, Yes : with tumor).

We used the same loaded pre-trained model (the one with uploaded files "resnet-50-MRI.json) with its weights ("weights.hdf5") didn't give good results (51% accuracy on test).

By using transfer learning the same as D1, and training on new dense layers on enough epochs we could reach 96% accuracy on train and almost 89% on test.

We can reach higher accuracy on D2 by using data augmentation (rotation,shearing...) and get new samples which then can be used in the training process, not only that we can also use another technique called "fine tuning" which is similar to transfer learning.

Acknowledgement

We would like to extend our gratitude to Professor Uzair Ahmad, who inspired us to make this project. We would also like to thank Northeastern University which provided us with the opportunity to learn the subject and apply its concept into making this project.

References

[1] V.P. Rathi and S.Palani, “Brain Tumor MRI Image Classification With Feature Selection and Extraction Using Linear Discriminant Analysis,” 2012 International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.4, July 2012 (Accessed October 27, 2022)

[2] R. Lavanyadevi, M. Machakowsalya, J. Nivethitha and A. N. Kumar, "Brain tumor classification and segmentation in MRI images using PNN," 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), 2017, pp. 1-6, doi:10.1109/ICEICE.2017.8191888.

[3] A. Kumar, A. Ashok and M. A. Ansari, "Brain Tumor Classification Using Hybrid Model Of PSO And SVM Classifier," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2018, pp. 1022-1026, doi: 10.1109/ICACCCN.2018.8748787.

[4] S. Arora and M. Sharma, "Deep Learning for Brain Tumor Classification from MRI Images," 2021 Sixth International Conference on Image Information Processing (ICIIP), 2021, pp. 409-412, doi: 10.1109/ICIIP53038.2021.9702609.