[Team 28] Brain Tumor Detection using Convolutional Neural Network

Piyush Umesh Devikar
pdevika@ncsu.edu

Nisarg Chokshi
nmchoks2@ncsu.edu

Sriram Sundararaj
ssunda24@ncsu.edu

***Abstract*—In order to improve human judgement in diagnosis advent of new technology into health care can be witnessed. An entrance of computer vision into diagnosis would reduce human error in judgment. For example, examining MRI results for brain related ailments requires extremely high concentration and wrong judgement would be catastrophic. The MRI scans are capable of identifying even the smallest aberrations in the human body. Here we train a model to specifically identify these tiny aberrations from MRIs and predict presence of a tumor with high accuracy. Convolutional Neural Network (CNN) is one of the most effective techniques for this problem statement. Thus using image preprocessing and transfer learning using VGG16, we built a highly reliant and robust model to solve this problem. [1]**

***Keywords*—Brain Tumor, MRI, OpenCV, Data Augmentation, CNN, Transfer Learning, VGGNet.**

I. Model Training and Selection

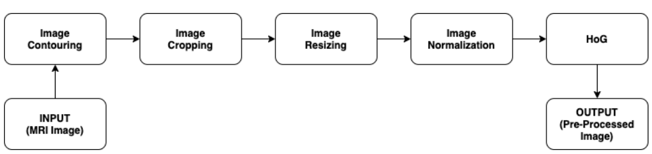


Fig. 1: Image Pre-processing

Before augmenting the raw dataset and feeding it to the model for training, we processed the images by performing some image processing operations as shown in Fig.2. We used OpenCV library for this task. To remove all the unnecessary regions in the input image, we did contouring followed by cropping the image. Subsequently, to make all the image of the same size, we performed image resize operation. Lastly, to maintain the consistency in dynamic range of pixel intensity values, we performed image normalization and histogram of oriented gradients (HOG).

A. The Model

A Convolutions Neural Network (CNN) model is found to be the best suited approach for the problem statement. It is comprised of one or more convolution layers (often with a sub-sampling step) and then followed by one or more fully connected layers as in a standard multi-layer neural network. The main goal of the convolutional base is to generate features from the image. The architecture of a CNN is designed to take advantage of the 2D structure of an input image [2].

Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned [4]. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

Transfer learning is usually expressed through the use of pre-trained models. A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem similar to the one that we want to solve. Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature (e.g. VGG, ResNet [10], MobileNet).

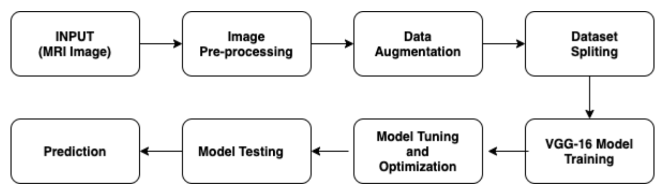


Fig. 2: Code flow block diagram

To improve on the baseline model that was a fully connected network, we implemented a VGG16 [3] network architecture. VGG-16 is a convolutional neural network architecture, it's name VGG-16 comes from the fact that it has 16 layers. It's layers consists of Convolutional layers, Max Pooling layers, Activation layers, Fully connected layers. There are 13 convolutional layers, 5 Max Pooling layers and 3 Dense layers which sums up to 21 layers but only 16 weight layers. Conv 1 has number of filters as 64 while Conv 2 has 128 filters, Conv 3 has 256 filters while Conv 4 and Conv 5 has 512 filters. The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. All hidden layers are equipped with the rectification (ReLU) non-linearity. Reducing volume size is handled by max pooling. Two fully-connected layers, each with 4,096 nodes are then followed by a sigmoid classifier.

The best performance was observed by unfreezing the last 2-3 dense layers to make them identify the variations of the image data and optimize the weights. We used imagenet weights as the initial weights. We attempted more complex models like ResNet50 and VGG19, but we observed overfitting and hence low accuracy on the test set. This is why we decided to stick to simpler transfer learning models.

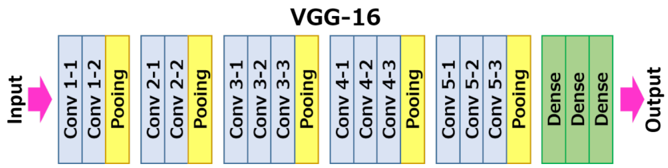


Fig. 3: VGG-16 Network Architecture

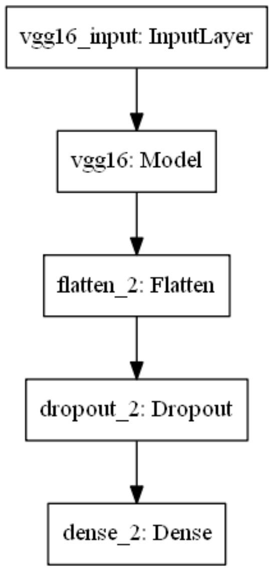


Fig. 4: Final Network Architecture

B. Baseline

Our baseline is a simple 3 layer fully connected network. For this model we are currently getting a test accuracy of 63%.

We use the same image preprocessing techniques along with k-fold cross validation with optimized hyperparameters. The hyperparameters used for this model are:
Train-Val-Test Split: 60%-20%-20%
Learning Rate: 0.0001
Optimizer: Adam
Image Size: 240*240

II. Experimental Section

A. Metrics

There are various metrics available for performance evaluation. Some of them are accuracy, precision, confusion matrix, specificity, F1 score, etc. The problem statement is a multi-class classification problem with 2 classes. Therefore, **accuracy and cross-entropy loss** are the good choices for the evaluation metric. Apart from the performance, **complexity of the network structure** is also an important factor for a machine learning model. A complex model costs more both in terms of time and hardware requirements. Overly complex models are less easily interpreted, at greater risk of overfitting. Therefore, in this project, we tried to keep the model simple along with getting the similar performance as on a more complex model.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 5: ConvNet configurations (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv(receptive field size)-(number of channels)". The ReLU activation function is not shown for brevity.

Accuracy is the quintessential classification metric and is a decent choice of evaluation for classification problems which are well balanced and not skewed or class imbalanced. Also, it is pretty convenient to analyze and easily suited for binary as well as a multi-class classification problems.

Cross-entropy [8] is the classical loss function which is often used for multi-class classification problems. Cross-entropy calculates a score that summarizes the average difference between the actual and predicted probability distributions for all classes in the problem. This score is to be minimized and an ideal cross-entropy value is 0.

To fully evaluate the effectiveness of a model, examining both precision and recall is necessary. Precision helps when the costs of false positives are high. Recall helps when the cost of false negatives is high. There is always trade off between precision and recall i.e. improving precision typically reduces recall and vice versa. While recall expresses the ability to find all relevant instances in a dataset, precision expresses the proportion of the data points our model says was relevant actually were relevant.

F1 score [9] is a measure of a test's accuracy. The F1 score conveys the balance between the precision and the recall. The F1 score is the harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall). F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution (large number of Actual Negatives).

B. Model Selection

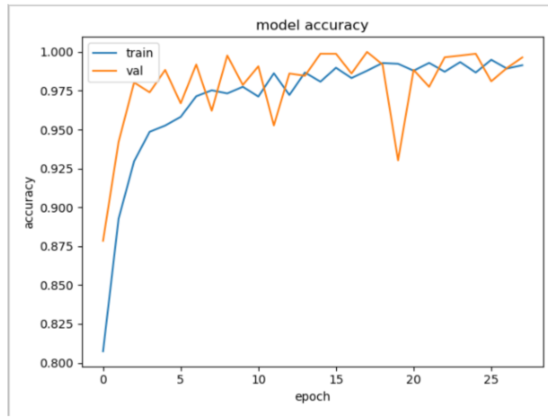
Hyperparameter tuning was done by monitoring 2 things mainly, **Testing Accuracy, Stability of Loss curves.**

The hyperparameters of choice are as follows:
Train-Val-Test Split: 60%-20%-20%
Learning Rate: 0.0001
Optimizer: Adam
Batch Size: 32
Image Size: 240*240
Loss: Binary Cross-entropy
Epochs: 50

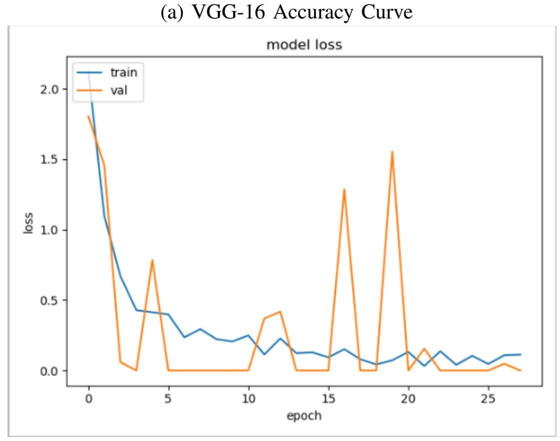
Some of the Accuracies before and after hyperparameter tuning are shown in Table I

TABLE I: Model Performance to Hyperparameter changes

Parameter	Value	Test Accuracy	Stability of Losses
Learning Rate	1e-1	87.51%	N
Learning Rate	1e-5	95.87%	Y
Batch Size	8	94.82%	N
Batch Size	64	96.18%	Y
Optimizer	SGD	89.84%	N
Optimizer	RMSprop	93.74%	N
Dropout	0.5	88.74%	N
Image Size	160*160	91.17%	Y



(a) VGG-16 Accuracy Curve



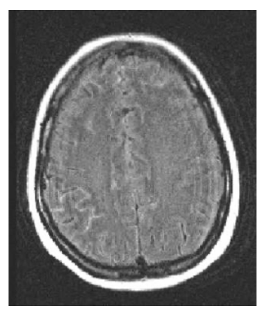
(b) VGG-16 Loss Curve

Fig. 6: Model Performance Plots

C. Performance and Comparison to Baseline

TABLE II: Model Performances

Model	Training Acc	Val Acc	Test Accuracy
MLP (Baseline)	85.2	76.89	63.2
SVC	89	88	89.48
SVC with HOG	95	94	94.68
CNN	92.10	92.46	90.04
VGG16	95.58	95.68	96.68
VGG16 (trained 2 layers)	**98.48**	**98.45**	**97.56**
VGG16 (trained 3 layers)	97.78	98.45	96.68
VGG19 unfreeze 3 layers	100	97.37	87.86
VGG19 unfreeze 2 layers	99.9	96.87	72.86
VGG19 unfreeze 1 layer	99.9	98	66.13
ResNet50 (trained 1 layer)	100	100	84.65


(a) Y185.png

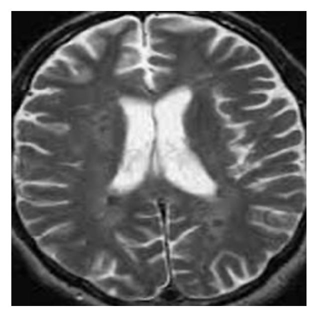

(b) 21 no.png

Fig. 7: Sample misclassified Images

TABLE III: Evaluations Results on Test Set

Test Accuracy	RMSE Loss	Precision	Recall	F1 Score
97.56%	0.93	0.975	0.975	0.975

Based on the data observed above in Table II, we made a model that is able to provide us a reliable results which is able to predict the presence of brain tumour in patients.

The current model of choice provides about **98%** test accuracy compared to a **63%** provided by the baseline. Table III provides the other performance metrics for the model of choice. As we have very less data (about 250 images) the model is not able to predict all the cases properly. With more data, this model will become more robust and provide perfect prediction results.

Two of the misclassified images are shown in Fig.7. The image on the left is classified as ”No Tumor” and the image on the right is classified as ”Tumor”. As seen from the Fig.7, even the human eye seems to be incapable of detecting a tumor in the image on the left. The model is trained to detect small blobs within the brain as tumour which is undecipherable here. On the other hand, the model has detected a tumor in the second image. As can be seen from the image, the centre region which is a part of the anatomy of the brain, would fulfil the criteria of being a small blob, and hence classified as a tumour.

REFERENCES

[1] H. E. M. Abdalla and M. Y. Esmail, ”Brain Tumor Detection by using Artificial Neural Network,” 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, 2018, pp. 1-6.

[2] V. Amsaveni and N. A. Singh, ”Detection of brain tumor using neural network,” 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, 2013, pp. 1-5.

[3] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

[4] Emilio Soria Olivas, Jose David Martin Guerrero, Marcelino Martinez Sober, Jose Rafael Magdalena Benedito, and Antonio Jose Serrano Lopez. 2009. Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA.

[5] https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

[6] https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection

[7] https://keras.io/preprocessing/image/

[8] https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

[9] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[10] https://keras.io/applications/resnet