

Deep Learning Assignment Report

Submitted By
Amruth Karun M V

Introduction

Deep Learning helps in the analysis of large number of medical data at a very high speed and improves the diagnosis to a great extent. We can make use of the existing approaches like classification, detection and segmentation techniques to identify tumors and damages in human tissues directly from the medical imaging data. The intention of this assignment is to classify brain tumor into four classes, 'Glioma', 'Meningioma', 'No tumor' and 'Pituitary' using VGG-19 and SqueezeNet. Classification accuracy and loss is calculated, and plotted for each epoch.

Problem Statement

The problem is to identify abnormal cells or tumors in the brain and classify brain tumor Magnetic Resonance Imaging (MRI) images into four classes, Glioma', 'Meningioma', 'No tumor' and 'Pituitary' using the deep learning models VGG-19 and SqueezeNet.

Dataset

The dataset used for this problem is the Kaggle MRI Brain Tumor dataset. The link for the dataset is given below:

Link: <https://www.kaggle.com/masoudnickparvar/brain-tumor-mri-dataset>

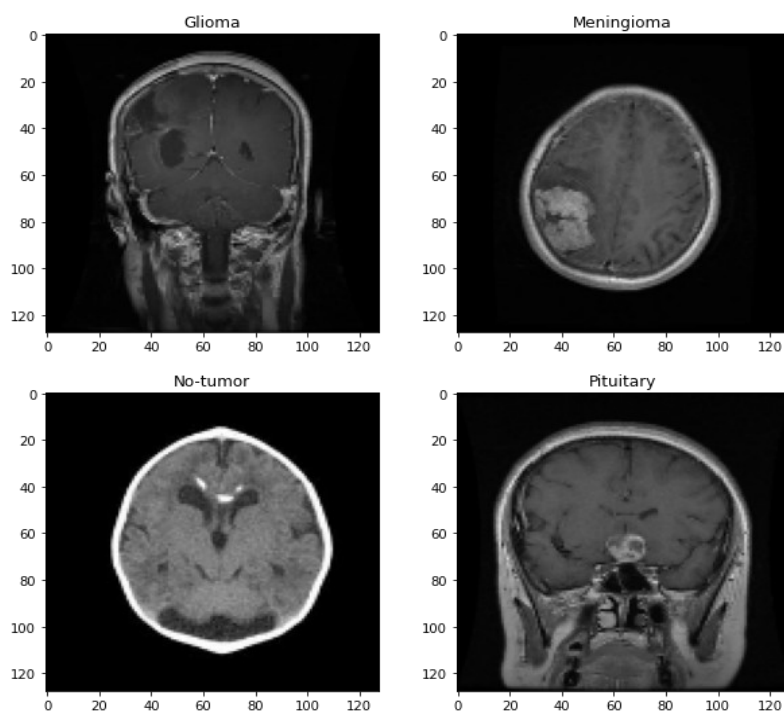
This dataset contains 7022 images of human brain MRI images which are classified into 4 classes: glioma, meningioma, no tumor, and pituitary tumor. This dataset is a combination of other two datasets from figshare and kaggle.

1. Kaggle: <https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri/metadata>

2. Figshare: https://figshare.com/articles/dataset/brain_tumor_dataset/1512427

- This brain tumor dataset containing 3064 T1-weighted contrast-enhanced images from 233 patients with three kinds of brain tumor: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices).

The final dataset consist of 1311 test images and 5712 train images.



Glioma:

A glioma is a type of tumor that starts in the glial cells of the brain or the spine. Gliomas comprise about 30 percent of all brain tumors and central nervous system tumours, and 80 percent of all malignant brain tumours.

Meningioma:

A meningioma is a tumor that arises from the meninges, the membranes that surround the brains and may compress or squeeze the adjacent brain, nerves and vessels.

Pituitary Tumor:

A pituitary tumor is an abnormal growth in the pituitary gland which is located behind the back of the nose. It makes hormones that affect many other glands and many functions in our body. Most pituitary tumors are not cancerous (benign).

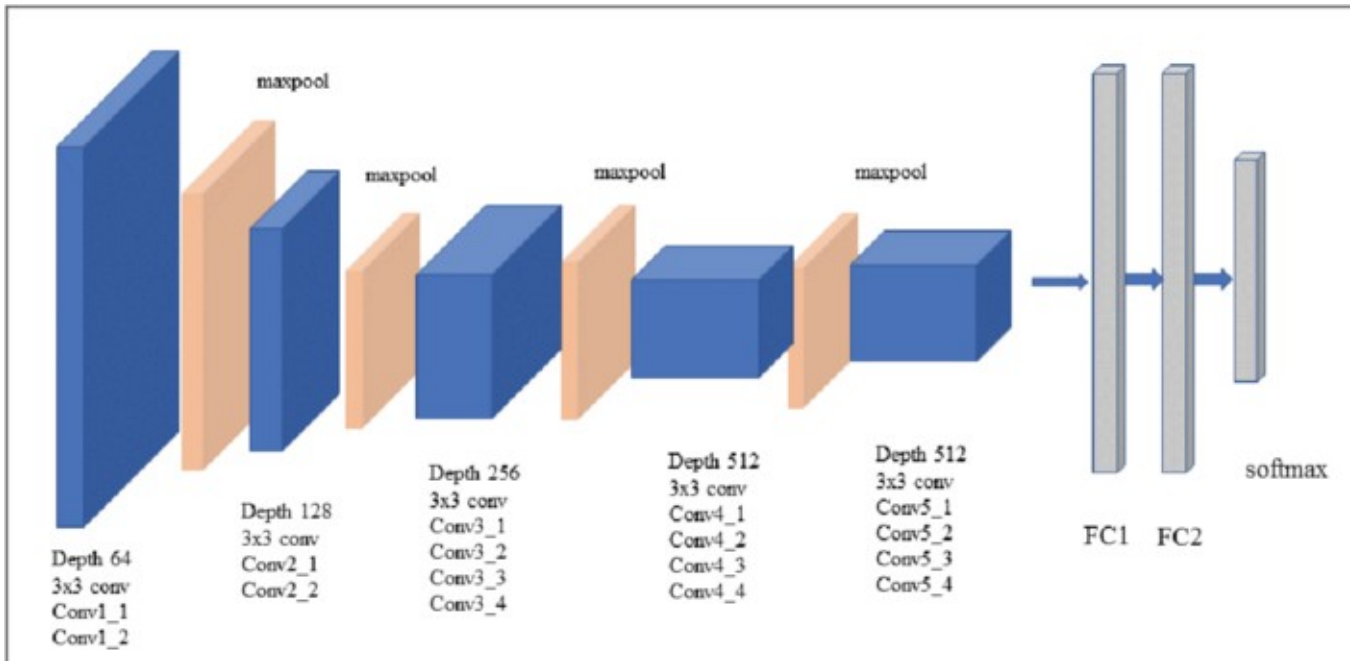
Deep Learning Based Solution: VGG-19 and SqueezeNet

The brain tumor classification can be carried out with the help of deep learning models with Convolutional Neural Networks (CNN). Proper classification of the MRI images can be done by training the dataset using state of the art models like VGG-19 and SqueezeNet. Using these approaches we are able to get an accuracy around 90% in train and test datasets. For each iteration loss and accuracy is calculated and the learning curve is plotted. Model is evaluated using the test set and confusion metrics is calculated using the predictions. Detailed model architecture and dimensions of the two models are given below.

VGG-19: Model Architecture:

VGG19 is a variant of VGG model which in short consists of **19 layers** (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer).

- The input to VGG-19 a 224x224 RGB image.
- The first two layers are convolutional layers with 3x3 filters, and first two layers use 64 filters that results in 224x224x64 volume as same convolutions are used. The filters are always 3x3 with stride of 1.
- After this, pooling layer was used with max-pool of 2x2 size and stride 2 which reduces height and width of a volume from 224x224x64 to 112x112x64.
- This is followed by 2 more convolution layers with 128 filters. This results in the new dimension of 112x112x128.
- After pooling layer is used, volume is reduced to 56x56x128. Before every pooling layer batch normalization is performed
- Four more convolution layers are added with 256 filters each followed by batch normalization and max-pooling.
- Same step is repeated twice with convolution blocks with 512 filters. This makes the fifth block of the VGG architecture.
- After the final pooling layer, the result is flattened into Fully Connected (FC) layer with 4096 channels with dropout and softmax output of 4 classes as shown in the below diagram.



VGG-19 Architecture

VGG-19: Model Dimension:

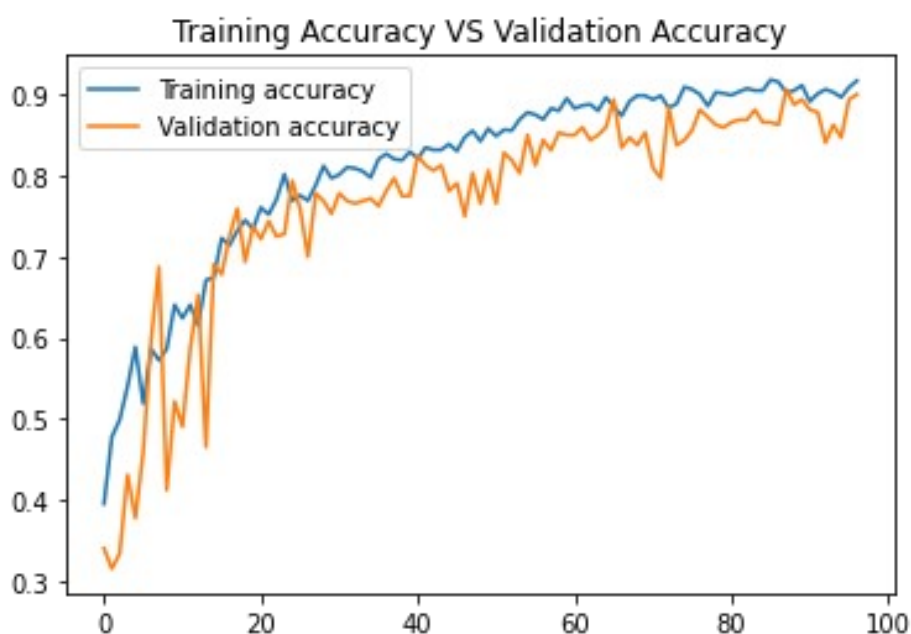
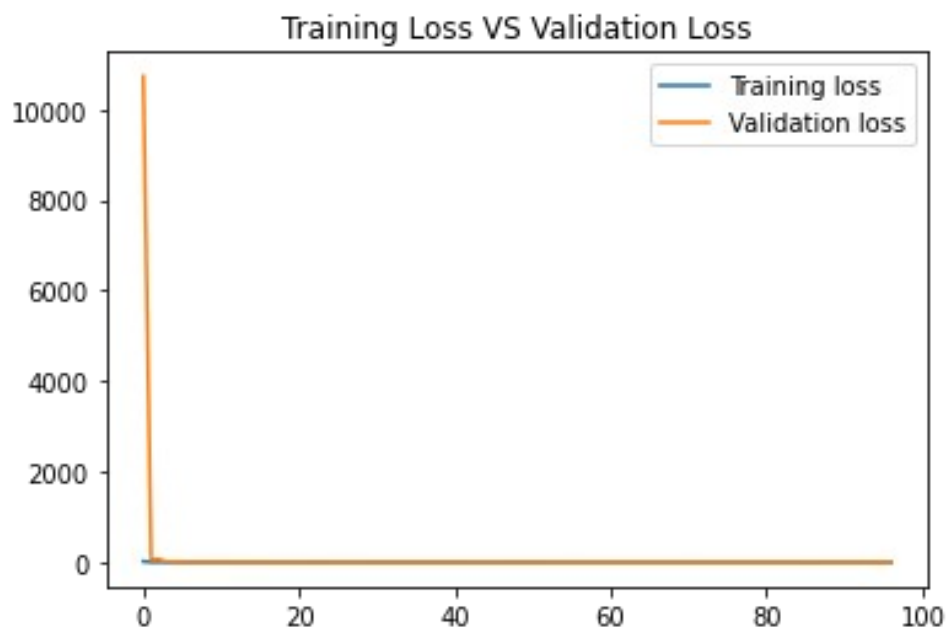
Based on the architecture given above, the dimensions of the model are as given below.

Layer name	#Filters	#Parameters	#Activations
input			150K
conv1_1	64	1.7K	3.2M
conv1_2	64	36K	3.2M
max pooling			802K
conv2_1	128	73K	1.6M
conv2_2	128	147K	1.6M
max pooling			401K
conv3_1	256	300K	802K
conv3_2	256	600K	802K
conv3_3	256	600K	802K
conv3_4	256	600K	802K
max pooling			200K
conv4_1	512	1.1M	401K
conv4_2	512	2.3M	401K
conv4_3	512	2.3M	401K
conv4_4	512	2.3M	401K
max pooling			100K
conv5_1	512	2.3M	100K
conv5_2	512	2.3M	100K
conv5_3	512	2.3M	100K
conv5_4	512	2.3M	100K
max pooling			25K
fc6		103M	4K
fc7		17M	4K
output		—	Num Class

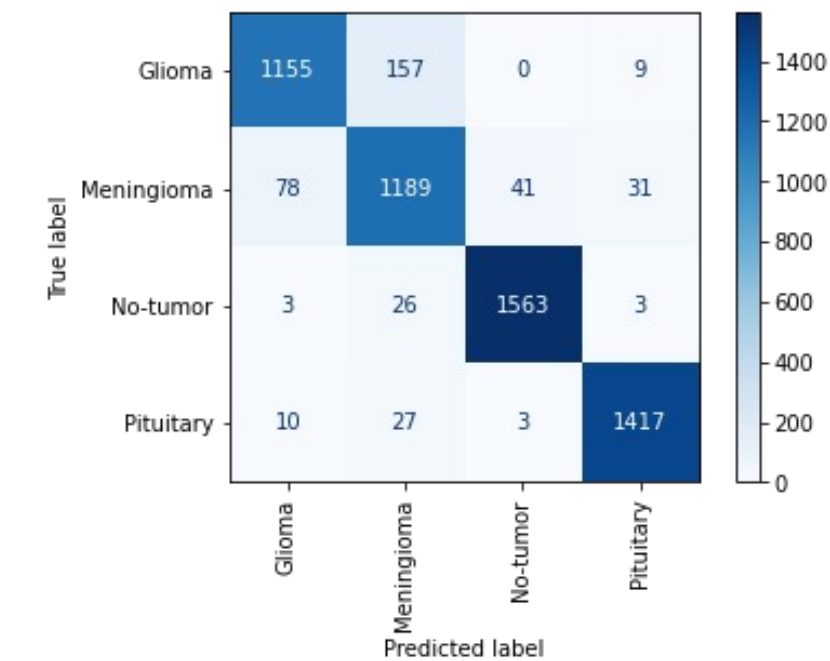
VGG-19 Model Dimension

VGG-19: Model Evaluation:

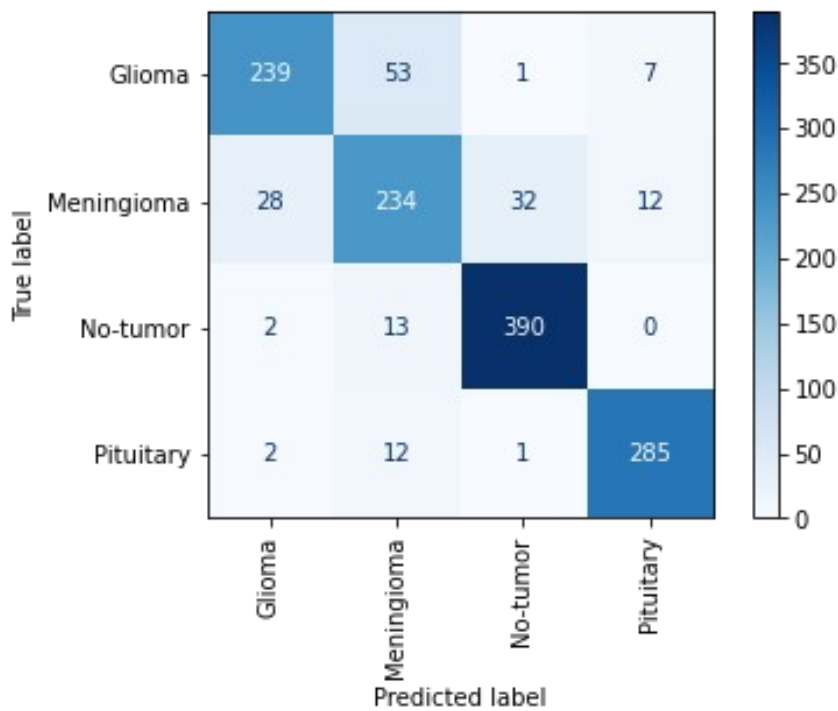
The VGG-19 model is trained for 100 epochs with a batch size of 128. I obtained an accuracy of 93% on the train set and 87% on the test set. The accuracy and loss curves are as given below. A loss of 0.19 on the training set and 0.31 on the validation dataset is observed.



Confusion matrix is also plotted against the training and validation dataset.



Confusion Matrix: Train Data



Confusion Matrix: Val Data

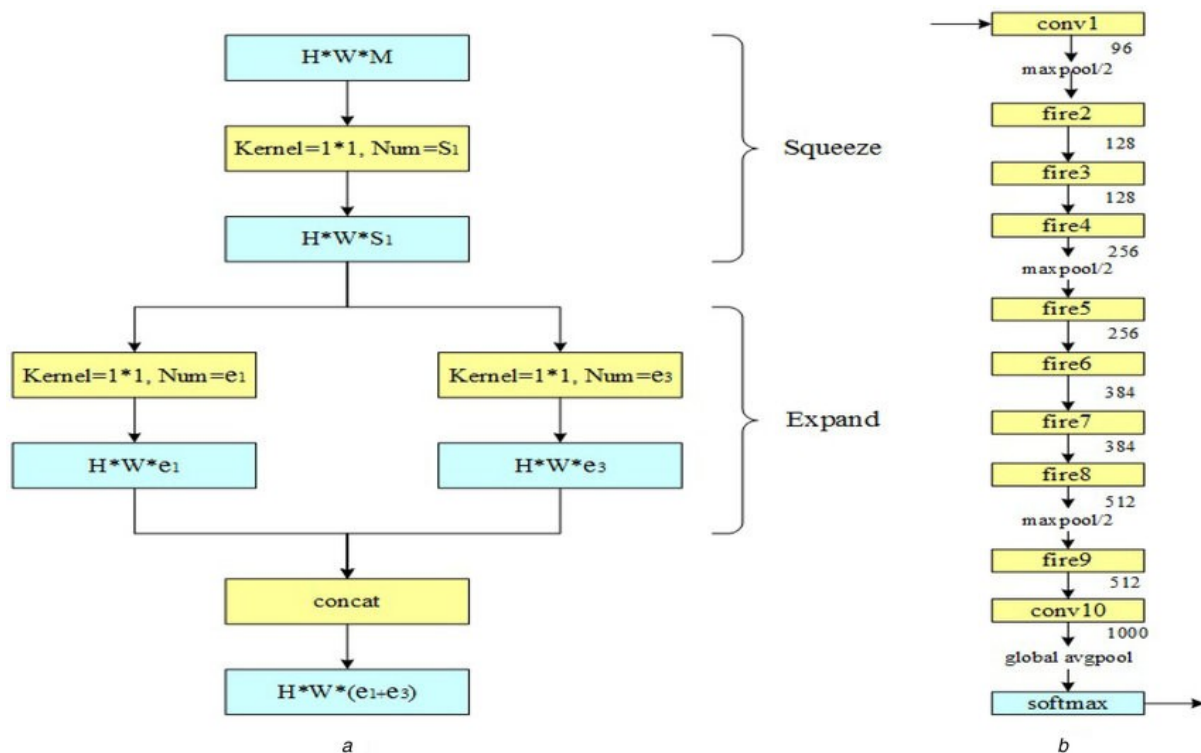
SqueezeNet: Model Architecture:

SqueezeNet is a convolutional neural network that employs design strategies to reduce the number of parameters, with the use of fire modules that "squeeze" parameters using 1×1 convolutions.

- Since 1×1 filters have 9x fewer parameters, 3×3 filters are replaced with 1×1 filters.
- Number of input channels are reduced to 3×3 filters using squeeze layers.
- Performs delayed downsampling to create large activation maps for higher classification accuracy.

Fire Module:

A Fire module is comprised of a squeeze convolution layer with only 1×1 filters, feeding into an expand layer that has a mix of 1×1 and 3×3 convolution filters. The number of 1×1 and 3×3 filters can be provided as hyper-parameters. The 1×1 filters in squeeze layer can be set less than $1 \times 1 + 3 \times 3$ of the expand layer to limit the number of filters in the input layer to 3×3 .



SqueezeNet Architecture

- SqueezeNet begins with a stand-alone convolution layer (conv1), followed by 8 Fire modules (fire2 – 9), ending with a final convolution layer (conv10).
- The number of filters per fire module is gradually increased from the beginning to the end of the network. Squeeze layers go from 16, 32, 48 to 64 and expand layers go from 64, 128, 192 to 256.
- Max-pooling with a stride of 2 is performed after layers conv1, fire4, fire8, and global average pooling is performed after conv10 (refer python notebook in SqueezeNet folder).

SqueezeNet: Model Dimension:

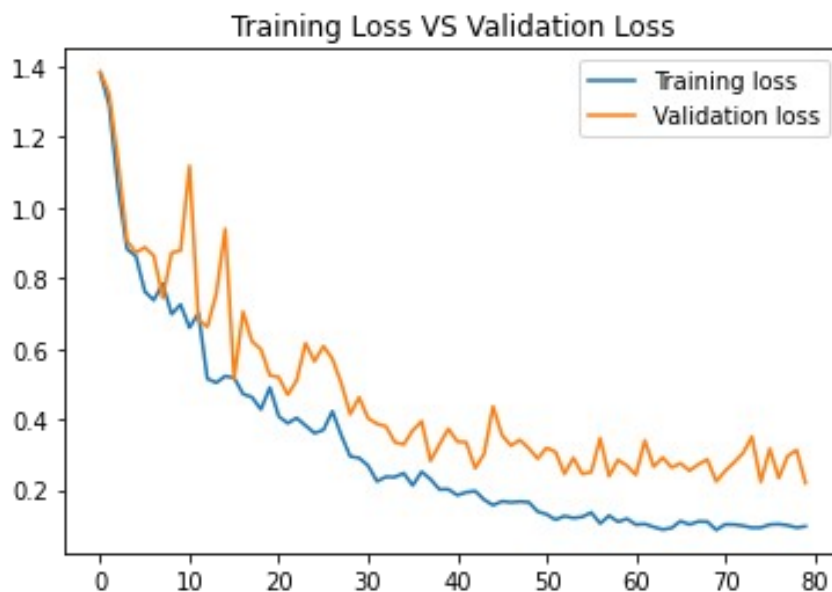
Compared to AlexNet, SqueezeNet has a 50% reduction in model size. Complex and simple bypass connections gives a better accuracy than vanilla SqueezeNet architecture. The detailed model dimension of the SqueezeNet architecture is given below.

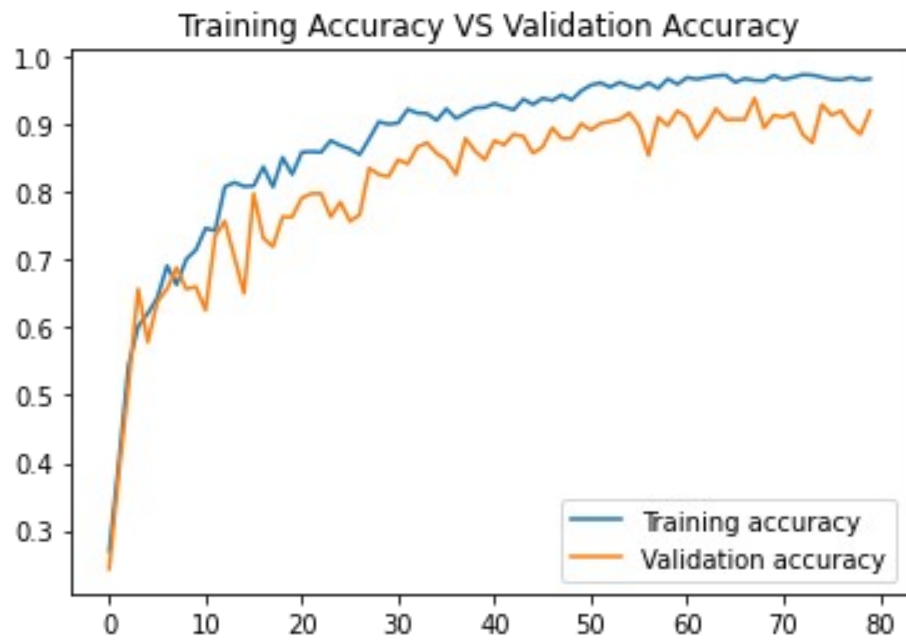
layer name/type	output size	filter size / stride (if not a fire layer)	depth	s _{1x1} (#1x1 squeeze)	e _{1x1} (#1x1 expand)	e _{3x3} (#3x3 expand)	s _{1x1} sparsity	e _{1x1} sparsity	e _{3x3} sparsity	# bits	#parameter before pruning	#parameter after pruning
input image	224x224x3										-	-
conv1	111x111x96	7x7/2 (x96)	1				100% (7x7)			6bit	14,208	14,208
maxpool1	55x55x96	3x3/2	0									
fire2	55x55x128		2	16	64	64	100%	100%	33%	6bit	11,920	5,746
fire3	55x55x128		2	16	64	64	100%	100%	33%	6bit	12,432	6,258
fire4	55x55x256		2	32	128	128	100%	100%	33%	6bit	45,344	20,646
maxpool4	27x27x256	3x3/2	0									
fire5	27x27x256		2	32	128	128	100%	100%	33%	6bit	49,440	24,742
fire6	27x27x384		2	48	192	192	100%	50%	33%	6bit	104,880	44,700
fire7	27x27x384		2	48	192	192	50%	100%	33%	6bit	111,024	46,236
fire8	27x27x512		2	64	256	256	100%	50%	33%	6bit	188,992	77,581
maxpool8	13x12x512	3x3/2	0									
fire9	13x13x512		2	64	256	256	50%	100%	30%	6bit	197,184	77,581
conv10	13x13x1000	1x1/1 (x1000)	1				20% (3x3)			6bit	513,000	103,400
avgpool10	1x1x1000	13x13/1	0									
<div> <div>activations</div> <div>parameters</div> <div>compression info</div> </div>											1,248,424 (total)	421,098 (total)

SqueezeNet Model Dimension

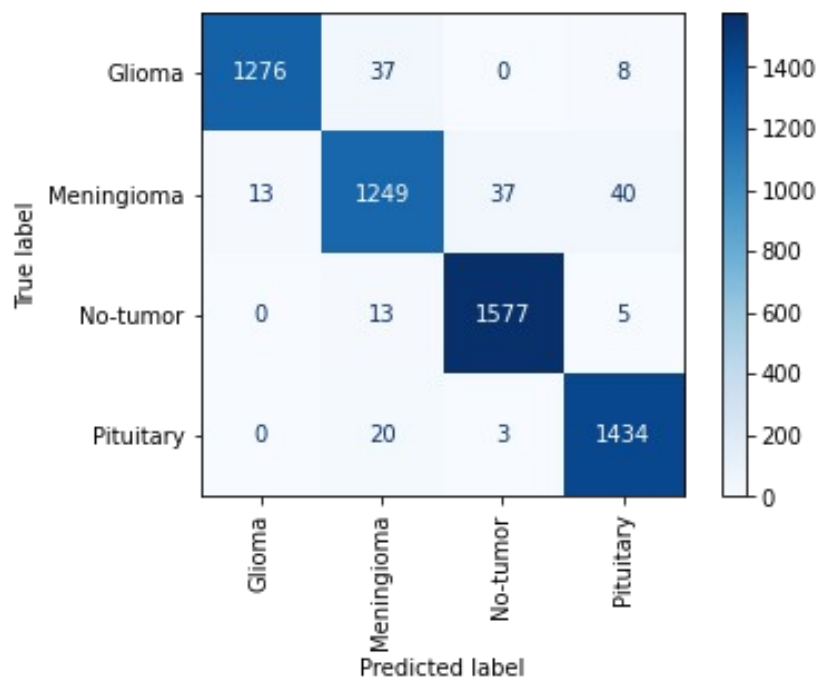
SqueezeNet: Model Evaluation:

The SqueezeNet model is trained for 100 epochs with a batch size of 128. Early stopping is done by monitoring the loss. Obtained an accuracy of about 97% on the train set and 90% on the val set. The accuracy and loss curves are as given below. A loss of 0.092 on the training set and 0.28 on the validation dataset is observed.

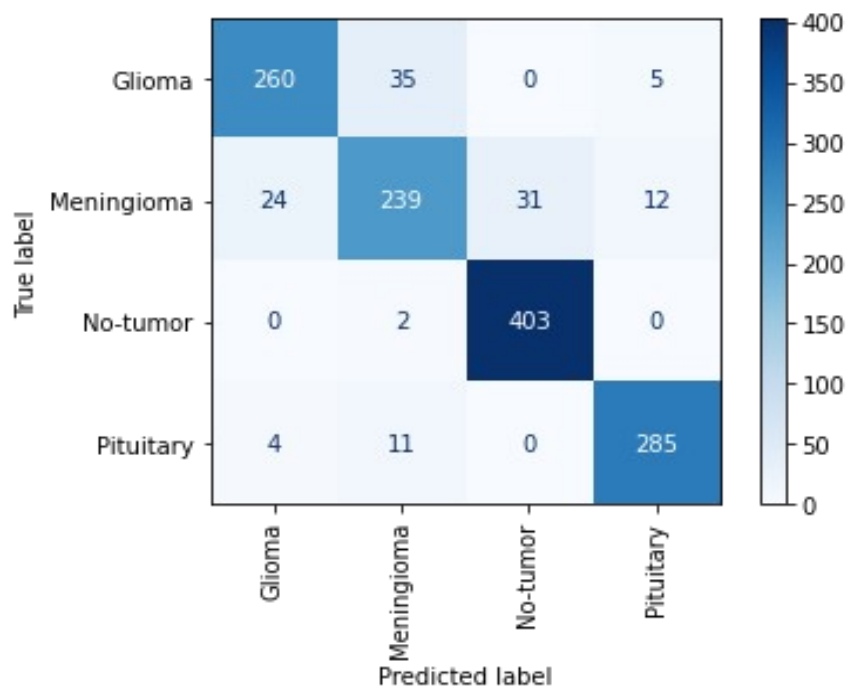




Predictions are made using the trained model and confusion matrix is plotted against the train and validation dataset.



Confusion Matrix: Train Data



Confusion Matrix: Val Data

Conclusion

Brain tumor classification is performed with the help of deep learning with convolutional neural network. Implemented two state of the art models, VGG-19 and SqueezeNet and obtained a train accuracy of about 97% for SqueezeNet. Model is evaluated and confusion matrix is calculated for the train and validation datasets. It is observed that most of the no tumor cases are correctly classified and there are some misclassification in the case of Glioma and Meningioma. Overall the model performance is observed to be good and further improvements can be made on the solution by modifying the existing architecture and tuning the hyper-parameters.
