Glioma Classification from MRI using CNN

Abhinav Rawat

Table of contents

Chapter 1

1. Introduction: A Brain tumour is considered as one of the aggressive diseases among both children and adults. They account for more than 85% of all primary Central Nervous System tumours. Each year around 11,700 people are diagnosed with a brain tumor, the survival rate for people with a cancerous brain or CNS tumor is 1 in 3. The best technique to detect brain tumors is a Magnetic Resonance Imaging (MRI), these images are examined by a radiologist and a manual examination is conducted to detect or rule out the presence of a tumour.

1.1 Overview: In this project our main focus is to build a classification model to identify the presence of brain tumor, in this study we are particularly working with image data of Glioblastoma (Glioma) MRI scans. Usually an MRI scans are observed by a radiologist to detect the presence of tumours, this can usually take anywhere from 2 days to a week of time, using Deep Learning algorithms we can fasten this process to within a minute or two. This helps with the diagnosis of Brain tumors which are time sensitive, matter of fact any cancer is time sensitive.

1.2 Objective of the Study: Automate the identification of Glioma tumor from MRI scans to an accuracy of above 90%.

Chapter 2

2. Methodology: The data was collected from kaggle, it consists of around 7000 images labelled. Unlike, normal MRI scans which are in .dicom formats, these have already been converted to .jpg which are easier to work with. The data is labelled into tumor and non-tumor. The model is built using 'Python' and 'Pytorch', Python is a high level, general purpose programming language, it is designed for ease of use as code readability. It has multiple libraries which contain methods and functions which help make coding easy for beginners and scales to high level programming. Pytorch is one such library which is an optimised tensor library for deep learning using GPU's and CPU's. We will utilize Convolutional Neural Network CNN and Residual CNNs algorithms for training and compare which method gives best/better results.

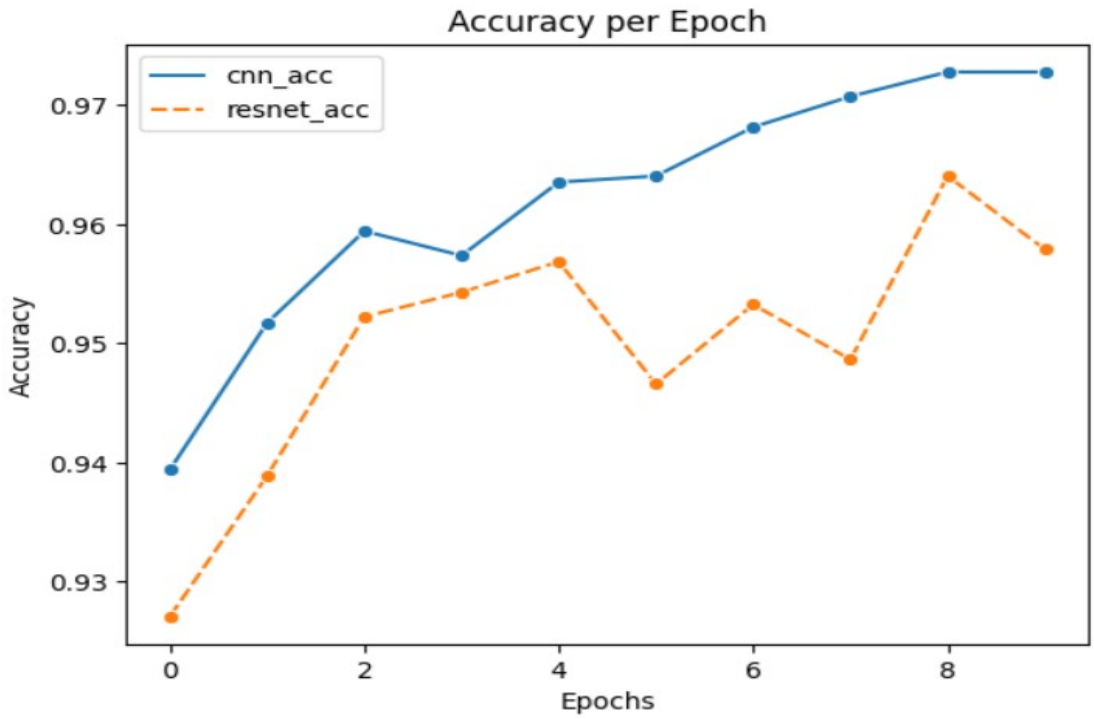
2.1 Data Preparation: The data consists of around 7000 images in '.jpg' format which are easier to work with compared to a '.dicom' format which is the typical format for an MRI scan. While performing Exploratory Data Analysis (EDA) we observed images of various shapes, so converting all the images to a 2D (black and white) '64×64' size to standardize the data. Since we are working with image data and memory can be an issue we will create a file location data-loader to feed images as batches to the training model. The file location of the images are added to a column in a data frame with a label column indicating the presence (1) or absence (0) of tumor. A training set with 5000 images and a validation set of 1950 images was created, the training data-loader has a batch size of 25 images meaning, at a time 25 images enter the model for training.

2.2 Model and Training: Convolutional Neural Network (CNN) and Residual CNNs (Resnet18) was utilized for model training. A CNN is a regularized type feed-forward neural network that learns features by itself via a filter/kernel optimization. This deep learning network has been applied to train data including text, images and audio. Resnet18 is a multiple layer CNN, which is 2 layers deep, to go in detail it consists of five CNNs, an average pooling layer, a fully connected layer and a softmax. We will train the model for 10 epochs and compare the validation set accuracy and run times.

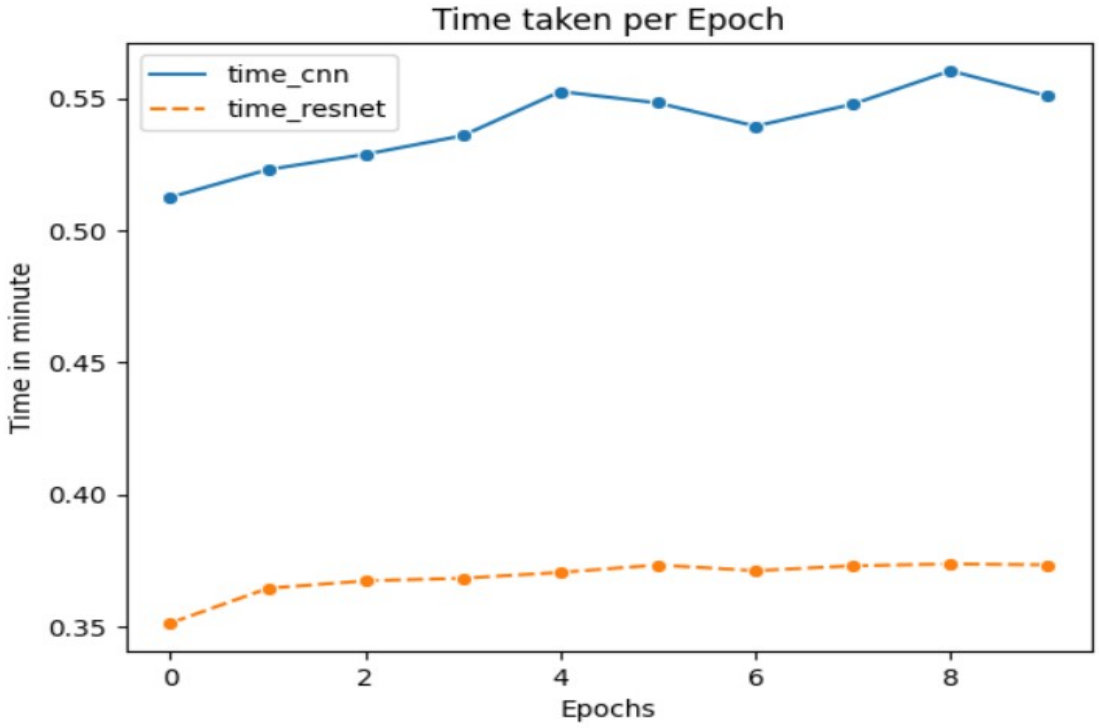
2.3 Results: After running each model for 10 epochs the following table was created with accuracy measures and time measures from individual runs on respective models.

	epoch	cnn_acc	resnet_acc	time_cnn	time_resnet
0	Epoch1	0.939394	0.927067	0.512474	0.351347
1	Epoch2	0.951721	0.938880	0.522947	0.364505
2	Epoch3	0.959425	0.952234	0.528706	0.367333
3	Epoch4	0.957370	0.954289	0.535804	0.368316
4	Epoch5	0.963534	0.956857	0.552475	0.370520
5	Epoch6	0.964047	0.946584	0.548158	0.373328
6	Epoch7	0.968156	0.953261	0.539380	0.371165
7	Epoch8	0.970724	0.948639	0.547717	0.373016
8	Epoch9	0.972779	0.964047	0.560332	0.373723
9	Epoch10	0.972779	0.957884	0.550776	0.373374

From the table we can see that the simple CNN performed slightly better than resnet18, but the later was slightly faster than CNN.



As can be seen in the graph where we see increment in accuracy with epochs for both resnet18 and CNN, with the end of final epoch CNN achieved an accuracy of 97.27% and resnet18 achieved an accuracy of 95.78%. The average times as seen in the graph is around half a minute (30 secs) for CNN and 1/3rd minute (20 secs) for resnet18.



Chapter 3

3. Discussion and Conclusion: From the results we can conclude that CNN had better accuracy on the validation data at the end of 10th Epoch, but resnet18 was more than 50% faster than CNN. The total time taken to run CNN was around 5 mins and 3 mins for resnet18, the highest accuracy obtained by CNN was 97.27% and 95.78% for resnet18. An interesting observation was made when we ran the resnet18 model for a total of 16 Epochs, since that will surmount to the time taken to run CNN, an accuracy of 96.2% was achieved. A simple CNN with 2 layers outperforms a resnet18 model.