Brain Tumor Classification

Capstone Project Final Report

Szeling Annie Hsu

I. Introduction

The brain is a major organ of the human body that controls all the body's functions. A brain tumor is the growth of abnormal cells in the brain. There exist many types of brain tumors. Some are cancerous (malignant) and others are noncancerous (benign). The growth rate and the location of a brain tumor determine how the function of the nervous system will be affected. As the brain is the control center of the body, more than any other cancer, brain tumors can have lasting and life-altering physical, cognitive, and psychological impacts on a patient's life.

A formal diagnosis of a brain tumor takes multiple steps. A series of tests will be undergone based on the symptoms, the location and the nature of the tumor, and the patient's health history. The diagnosis usually starts with magnetic resonance imaging (MRI). When the MRI scans show the presence of a brain tumor, a sample of the tumor's tissue (a biopsy) is usually needed to determine the type of brain tumor. However, manually reviewing the MRI scans is time-consuming and also prone to human error.

The goal of this project is to use machine learning techniques to determine the types of brain tumors from MRI scans. The diagnosis time is then shortened and the doctors can proceed with the treatment plan earlier. We will use deep learning algorithms to build a convolutional neural network as our predictive model. We will also explore the transfer learning technique to build several convolutional neural networks based on the pre-trained models.

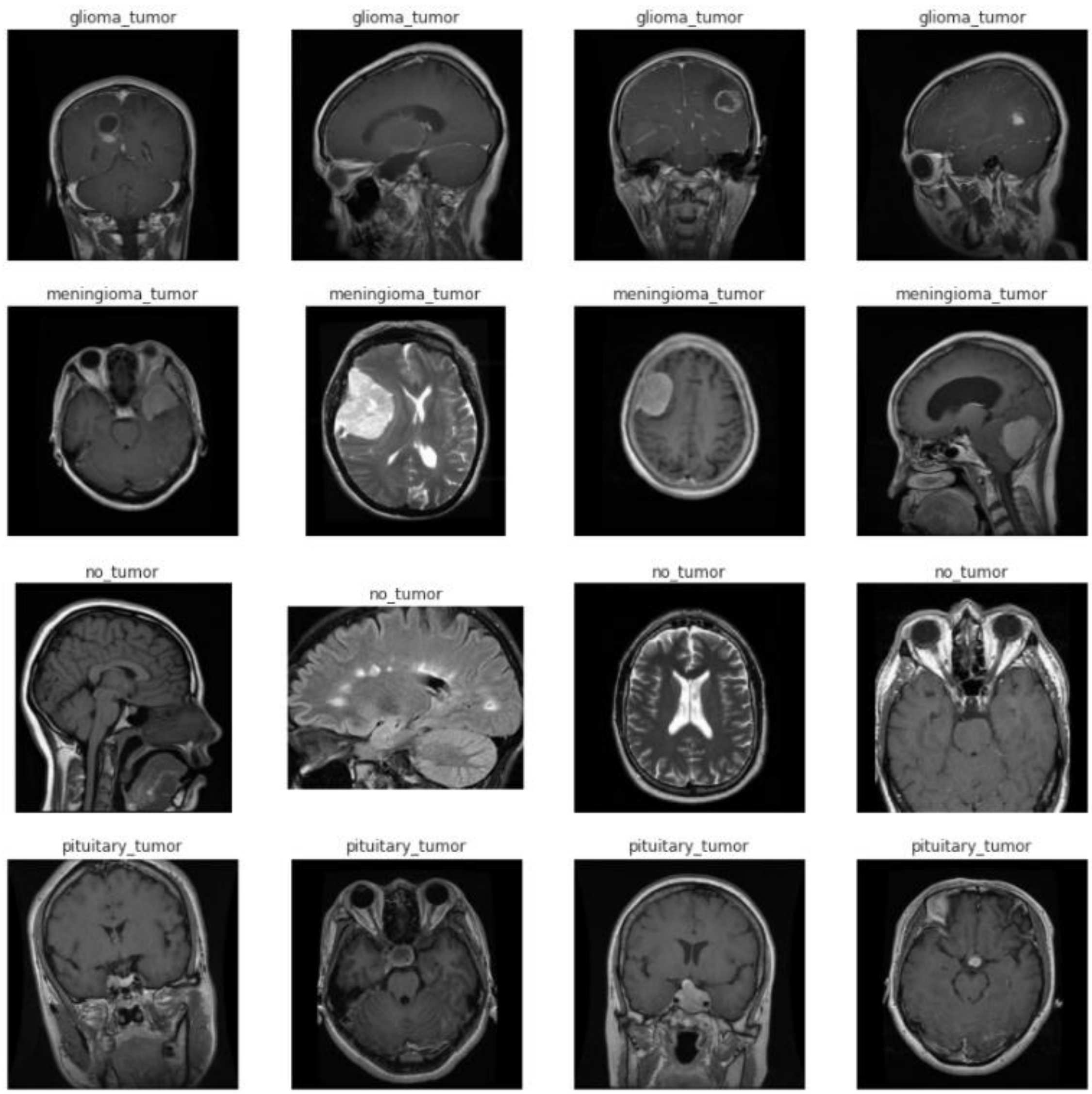
II. Data Source

We have taken a brain MRI images dataset [1] from Kaggle.com. It contains 2870 MRI images of the brain in the training dataset and 394 images in the testing dataset.

We reserved 20% of the training data for validation. Thus, there are 2296 images for training, 574 images for validation, and 394 images for testing. Our task is to classify the MRI images into four types of brain tumors – glioma tumor, meningioma tumor, pituitary tumor, and no tumor.

III. Exploratory Data Analysis

Here are some random sample images of the four types of brain tumors:



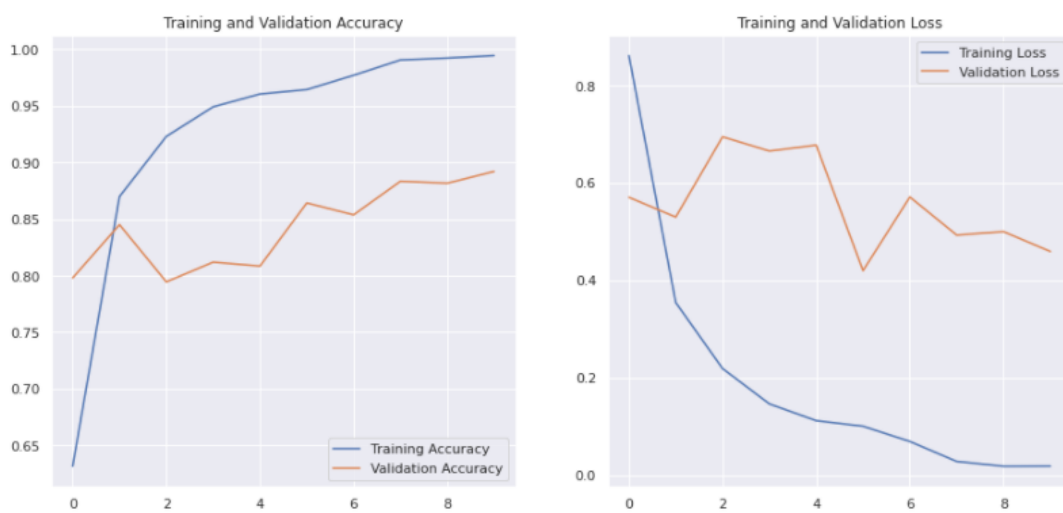
In the training dataset, the no tumor class has less than half of the data comparing to the other three classes. The dataset is imbalanced. Here is the plot of class distribution:

IV. Models

To solve this multi-class classification problem, we first built a baseline convolutional neural network model to get a baseline performance. Then, we apply the transfer learning technique to build another CNN model. In order to further improve the model performance, we tried preprocessing the images before fitting into the model to train.

A. <u>Baseline Model</u>

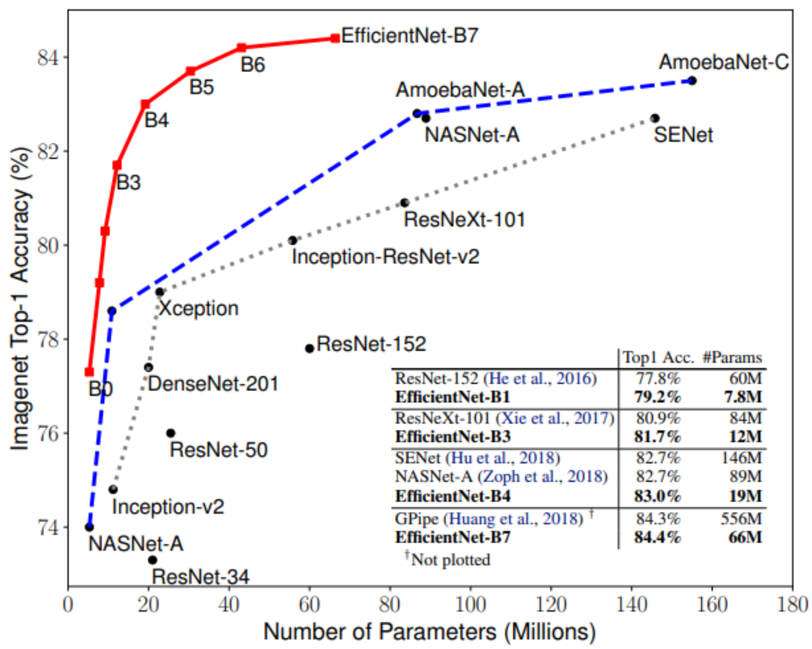
Our baseline model consists of two blocks. Each block has a convolution layer (Conv2D) to extract features from the images. Then follows a max-pooling layer to reduce the number of parameters to learn. A dropout layer is added after the two blocks to prevent the model from overfitting. The baseline model has a training accuracy of 99.43%, a validation accuracy of 89.20%, and a testing accuracy of 68.53%. Here are the plots of the training and validation accuracy and loss:



Plots of the training and validation accuracy and loss

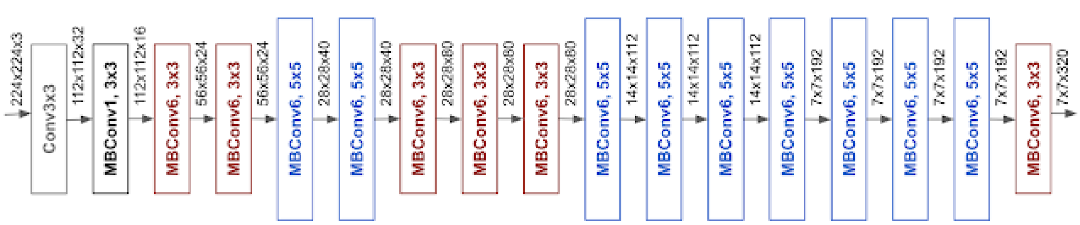
B. <u>EfficientNet-B1 Model</u>

Next, we used the transfer learning technique in deep learning to build other CNN model. We chose the EfficientNet family with the ImageNet pre-trained weights. The EfficientNet models are smaller in architecture yet yielding high performance. A smaller structure will benefit in reducing the computational time and in turn saving cost. Below is a graph of some popular CNN models comparing in model size vs. ImageNet accuracy:



Convolutional Neural Networks comparison in model size vs. ImageNet accuracy
Image source: EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling [2]

The EfficientNet models all built on their baseline network EfficientNet-B0. Other models in the family are the scaled up version of this baseline network. Here is the architecture of the EfficientNet-B0 model:

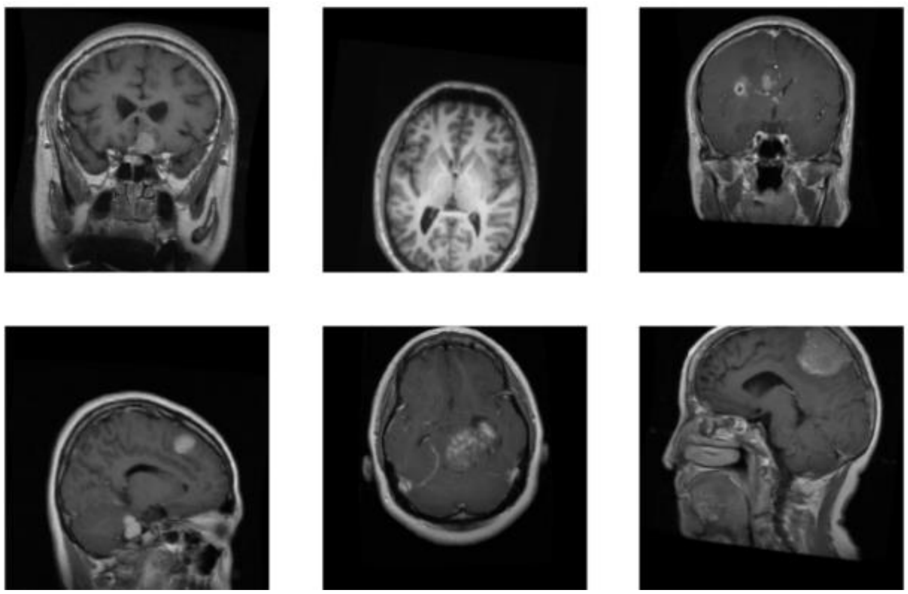


EfficientNet-B0 architecture
Image source: EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling [2]

Steps of workflow:

1. Image Augmentation

   In order to expand the size of our dataset for training, we used the technique of image augmentation to apply different transformations to the original images. We used Keras ImageDataGenerator to perform real-time data augmentation while the model was still in the training stage. The images were augmented by rotation, shifting and horizontal flip. Here are some samples of the augmented images:



Samples of augmented images

2. Compute class weight

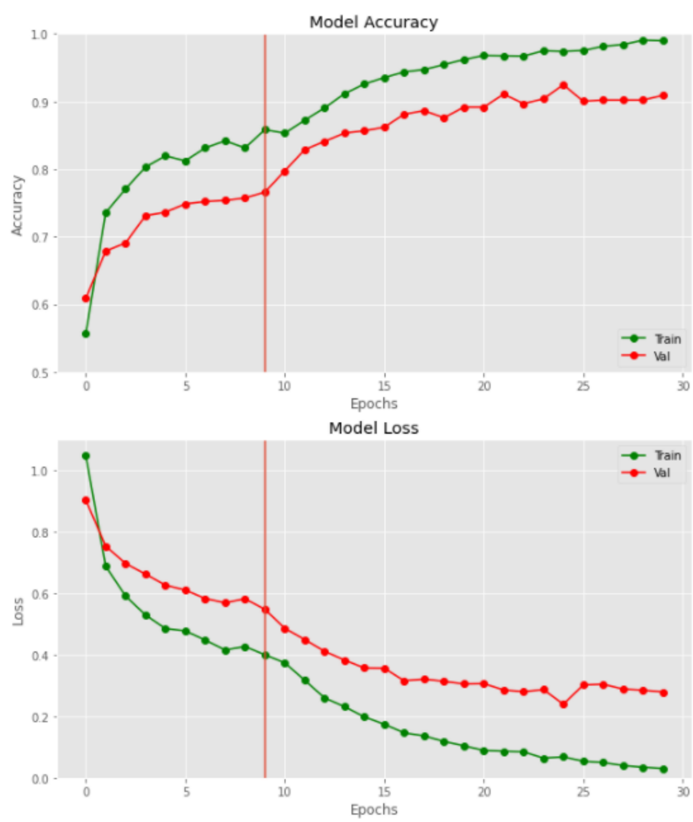
   To account the imbalanced dataset issue, we computed the class weights to use for training the model.

3. Build CNN model based on EffificentNet-B1 model

   We built our model using the EfficientNet-B1 model pre-trained with the ImageNet weights as a base and then built our own classifier at the top. The EfficientNet-B1 model has 339 layers and 7,856,239 parameters. We kept the convolutional base frozen before compiling and training the model. Freezing prevents the weights in a given layer from being updated during training. We trained the model with learning rate of 0.001 for 10 epochs. After training, the model had a training accuracy of 85.85%, validation accuracy of 76.61%, and the testing accuracy of 59.90%.

4. Fine-tune the entire model

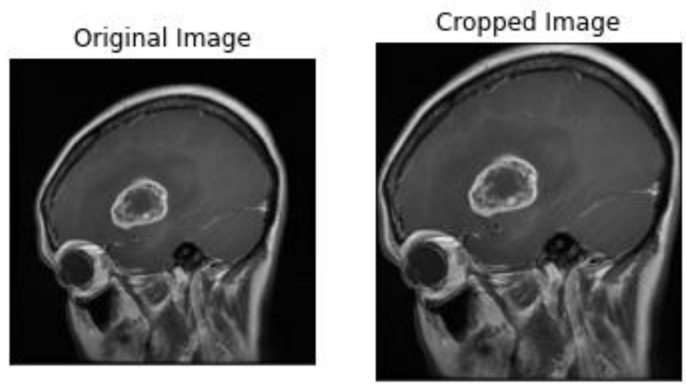
   After our model has converged on the data, we unfreeze the base model and train the entire model end-to-end with a low learning rate. The base model is still running in inference mode even when the base model becomes trainable, since we passed training=False when calling it when we built the model. The batch normalization layers inside will not update their batch statistics. We trained the model again with a low learning rate (1e-5) to incrementally adapting the pretrained features to the new data. We also increased the number of training epochs to 50. The training had an early stop at epoch 29 when the validation accuracy did not improve for 5 epochs. After fine-tuning, the model has a training accuracy of 99.00%, a validation accuracy of 90.92% and a test accuracy of 77.66%. Here are the plots of the training and validation accuracy and loss:



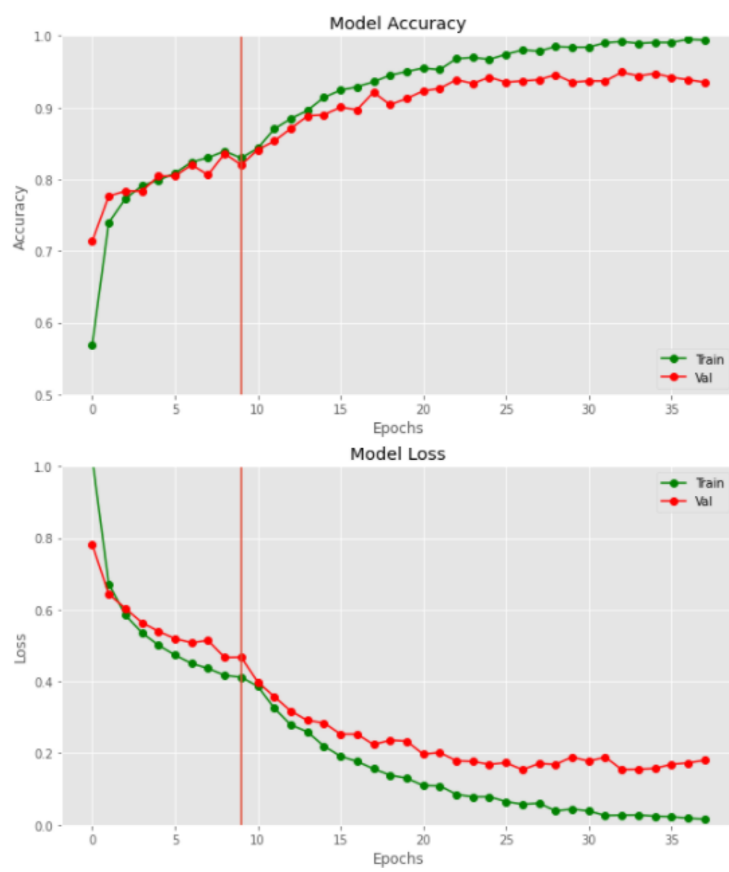
Plots of the training and validation accuracy and loss

C. <u>EfficientNetB1 Model with image preprocessing</u>

In an attempt to improve the model performance, we tried to preprocess the images before training the model. We cropped the images to reduce the background. Here is a sample of the original image and the cropped image.



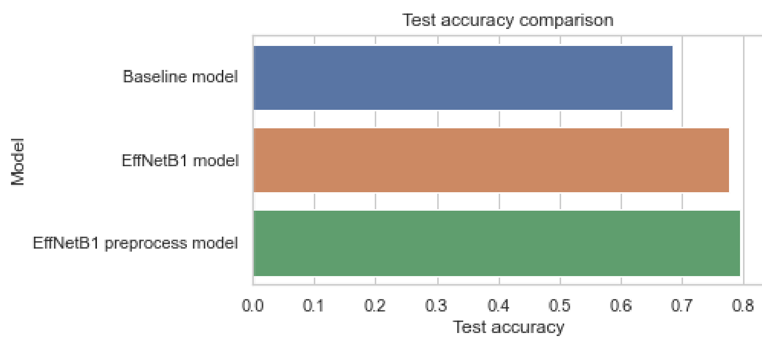
After the image preprocessing step, we followed the same workflow to train and fine-tune the model. The training stopped at 37 epochs with a training accuracy of 99.43%, a validation accuracy of 93.54%, and a test accuracy of 79.44%.



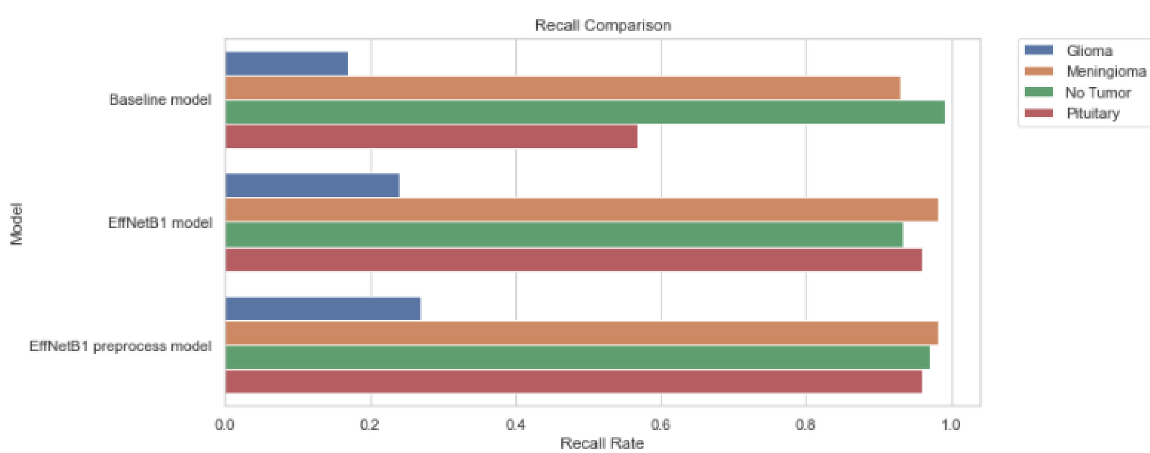
Plots of the training and validation accuracy and loss

V. Analysis

Comparing the three models, the model based on EfficientNet-B1 with image preprocessing yielding the highest test accuracy of about 80% which is about 11% higher than our baseline model. That means the model has the highest ratio of the total number of correct predications and the total number of predictions.



In the medical setting, we usually use recall rate as the metric to measure the performance of a classifier. The recall rate measures whether the model correctly identifying the true positives. It is very important not to mis-classified the brain tumor type or when a patient has a brain tumor but we mis-labeling as no tumor present. From the chart below, we see that all three models have very low recall rates on the Glioma class. The model with image preprocessing has the highest recall rate on the Glioma class. For the other three classes, the last model also has very high recall rates.

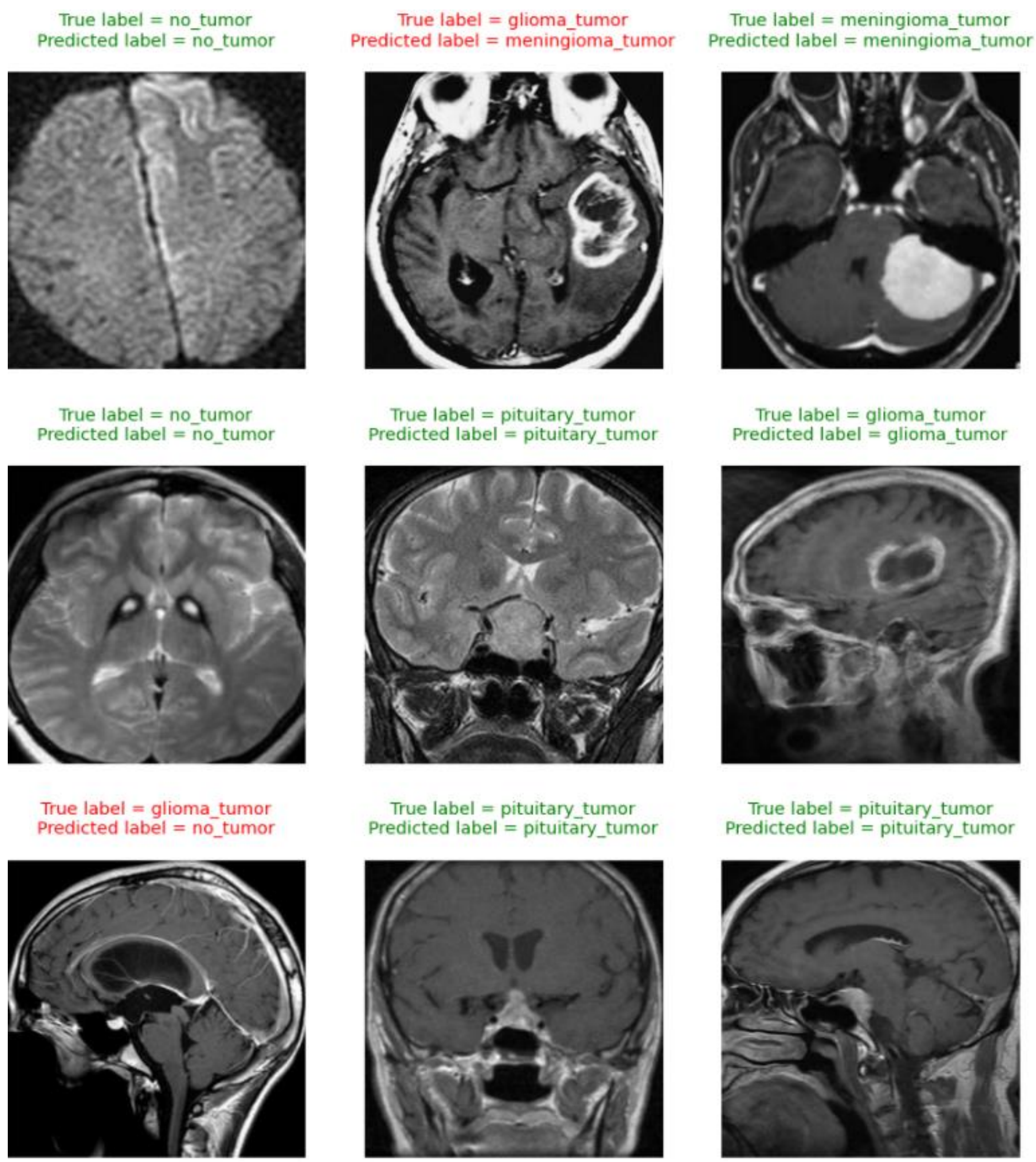


Let's take a look at the confusion matrix of the last model. We can see how the classifier labels the tumor classes. The classifier correctly labelled about one quarter of the Glioma Tumors, but wrongly labelled about half of the Glioma Tumor as Meningioma Tumor and one quarter as No Tumor.

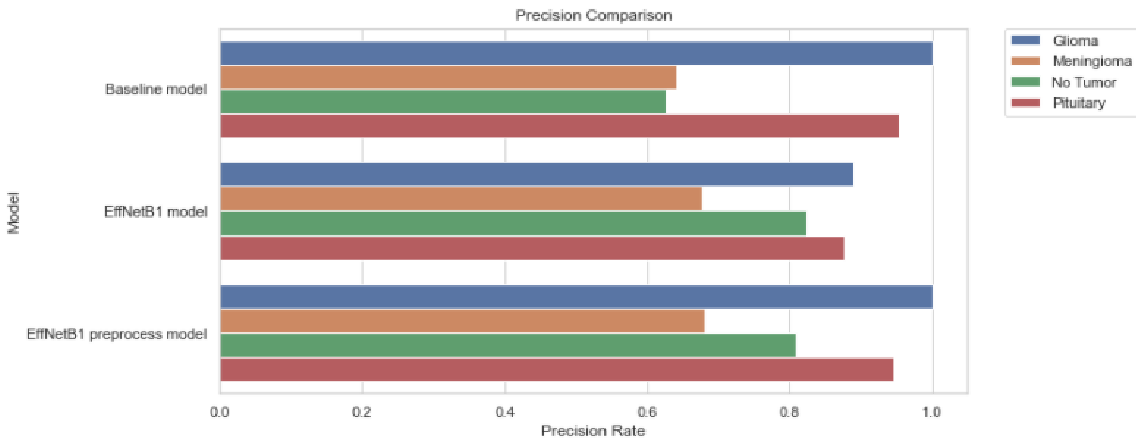
$$
\begin{array}{c c}
 & \begin{array}{cccc} \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{3} \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} &
\begin{bmatrix}
27 & 50 & 21 & 2 \\
0 & 113 & 0 & 2 \\
0 & 3 & 102 & 0 \\
0 & 0 & 3 & 71
\end{bmatrix}
\end{array}
$$

0: Glioma Tumor
1: Meningioma Tumor
2: No Tumor
3: Pituitary Tumor

We can look at some sample images with their predicted labels:



Since about 75% of the Glioma Tumors were wrongly mislabeled as Meningioma Tumor and No Tumor, we see that the precision rates of the Meningioma Tumor and No Tumor classes were affected.



VI. Conclusion

Out of the three models, the model built based on EfficientNet-B1 model with image preprocessing performs the best. It has the highest test accuracy and the recall rates on all four tumor classes. Transfer learning did help to build a classifier with better performance. Image preprocessing also helped to further improve the model performance. In the future, we shall collect more images to train the model, especially the Glioma tumor. We shall also explore other methods to preprocess the images like creating different filters and tumor segmentation.

VII. References

[1] Brain Tumor Classification (MRI) Kaggle Dataset
https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri

[2] EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling
https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

[3] Transfer learning & fine-tuning
https://keras.io/guides/transfer_learning/