HackAI 2023 Challenge: Brain Tumor Detection

Technical Report

Team: Maverick

Mayur Patil
Jaeha Lee
Jonathan

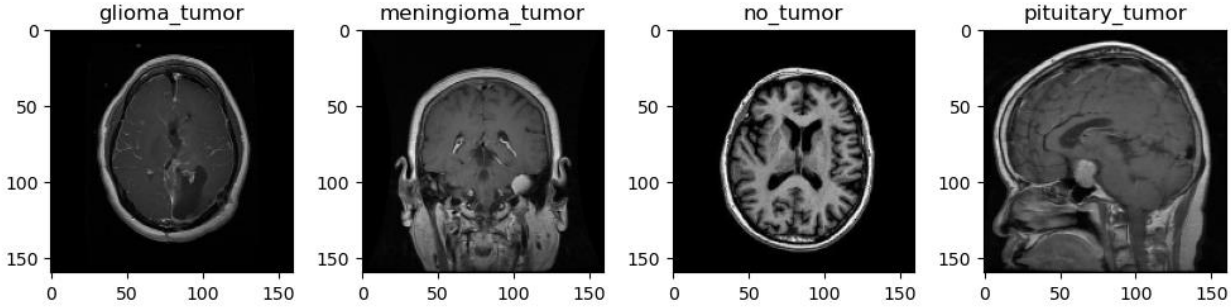
Problem Description and Motivation

One of the most deadliest cases of brain tumors takes thousands of lives every year. These are growth of cells in the brain or near it. There 85~90% of CNS Tumor cases and around 11,700 people are diagnosed annually. The proper treatment, planning, and accurate diagnostics can improve the life expectancy as there are lot of abnormalities in the sizes and location.

So, we tried to build a detection model using a convolutional neural network (CNN) in Tensorflow & Keras for detecting 4 types of brain tumors using MRI images. We used the dataset available on Kaggle [https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri]

Data Preparation & Pre-Processing

- Resizing images: Resize the image (because the images in the dataset come in different sizes (meaning width, height and # of channels). So, we want all of our images to be (160, 160, 3) to feed it as an input to the neural network.
- Apply normalization because we want pixel values to be scaled to the range 0-1.
- Data Augmentation: Since this is a small dataset, there wasn't enough data to train the neural network model. Also, data augmentation is useful in addressing the data imbalance issue in the data.
- Visualizing Samples



Data augmentation, training

The data set is trained in the model shown below The batch size is 11. As, there was small data set, so we augmented the data by using image data generator.

For image classification, we compared Xception and EfficientNetV2S which showed an accuracy of 86.47% and 90.54% respectively.

So, we used **EfficientNetV2S** for our project.

```
Epoch 42/100
220/220 [==============================] - 24s 107ms/step - loss: 0.1647 - accuracy: 0.921
0 - val_loss: 0.2058 - val_accuracy: 0.9022
Epoch 43/100
220/220 [==============================] - 23s 103ms/step - loss: 0.1591 - accuracy: 0.923
5 - val_loss: 0.2066 - val_accuracy: 0.8991
Epoch 44/100
220/220 [==============================] - 23s 106ms/step - loss: 0.1607 - accuracy: 0.920
3 - val_loss: 0.1916 - val_accuracy: 0.9022
Epoch 45/100
220/220 [==============================] - 23s 106ms/step - loss: 0.1632 - accuracy: 0.924
5 - val_loss: 0.2004 - val_accuracy: 0.9022
Epoch 46/100
220/220 [==============================] - 23s 104ms/step - loss: 0.1574 - accuracy: 0.923
8 - val_loss: 0.1999 - val_accuracy: 0.9054
Epoch 47/100
220/220 [==============================] - 23s 105ms/step - loss: 0.1643 - accuracy: 0.923
5 - val_loss: 0.2038 - val_accuracy: 0.9022
Epoch 48/100
220/220 [==============================] - 23s 104ms/step - loss: 0.1535 - accuracy: 0.924
2 - val_loss: 0.2063 - val_accuracy: 0.9054
Epoch 49/100                                                    Training Accuracy after callbacks
220/220 [==============================] - 23s 106ms/step - loss: 0.1594 - accuracy: 0.923
8 - val_loss: 0.2248 - val_accuracy: 0.8896
10/10 [==============================] - 1s 67ms/step - loss: 0.1905 - accuracy: 0.9054
Accuracy: 90.536278 Final testing accuracy
10/10 [==============================] - 1s 50ms/step - loss: 0.1905 - accuracy: 0.9054
```

Model Training

Final Results

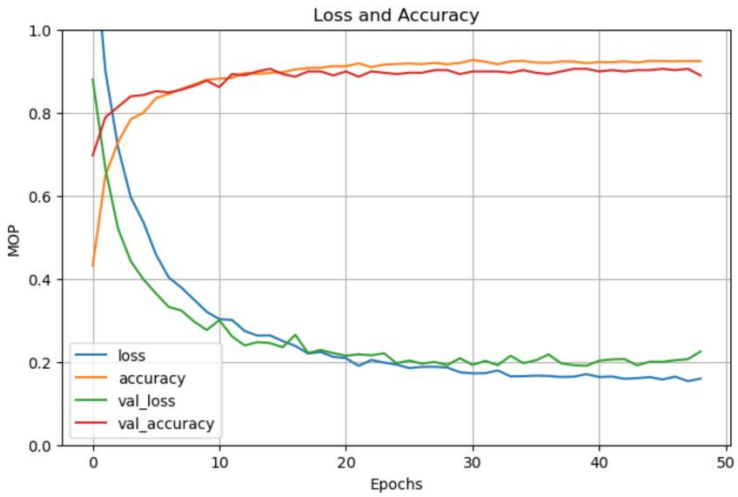
Training Accuracy - 0.9238

Testing Accuracy - 0.9054

Loss - 0.1905

Validation Accuracy - 0.8896

EfficientNetV2S showed an improvement of 4% over Xception model!!



Accuracy and Loss Plots

References

[1] S. Bhuvaji, A. Kadam, P. Bhumkar, S. Dedge, and S. Kanchan, “Brain Tumor Classification (MRI),” www.kaggle.com. https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

[2] M. Tan and Q. V. Le, “EfficientNetV2: Smaller Models and Faster Training,” arxiv.org, Apr. 2021, doi: https://doi.org/10.48550/arXiv.2104.00298.

[3] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” arXiv.org, 2016. https://arxiv.org/abs/1610.02357