Brain Tumor Detection and Classification using Transfer Learning

Submitted by: G Harshith

Under the Guidance of: Mr. Surendra Solanki

Department Name: AIML

Institution Name: Manipal University Jaipur

Year & Semester: 6th semester

Table of Contents

1. Introduction

Problem Statement:

Brain tumors are life-threatening conditions that require early detection and accurate classification for effective treatment. Manual diagnosis from MRI scans is time-consuming and prone to human error. There is a need for an automated system that can classify brain tumors using deep learning techniques.

Objectives:

- To build a classification model that distinguishes between four brain tumor classes: glioma, meningioma, pituitary tumor, and no tumor.
- To apply deep learning using EfficientNet B0 for high accuracy and low computational cost.
- To evaluate model performance using standard classification metrics.

Scope:

The project focuses on detecting and classifying brain tumors from MRI images using deep learning techniques. It is limited to image-based classification and does not extend to tumor segmentation or 3D volume analysis. The dataset used is sourced from Kaggle and includes pre-classified MRI images.

---

2. Literature Review

Several research studies have explored tumor detection using Convolutional Neural Networks (CNNs), ResNet, VGGNet, and transfer learning. Earlier methods focused on manual feature extraction and classical machine learning models such as SVM and Random Forests. These approaches lacked the ability to generalize across complex tumor images.

Recent studies have used pre-trained models such as ResNet50, VGG16, and MobileNet to leverage transfer learning, achieving better performance. However, models like EfficientNet provide a more optimized balance between accuracy and computational efficiency.

Our approach improves on previous work by implementing **EfficientNet B0**, a compound scaling method that scales depth, width, and resolution simultaneously, allowing better performance with fewer parameters.

---

3. System Analysis & Design

Methodology:

1. Data Collection and Preprocessing
2. Data Augmentation
3. Model Training with Transfer Learning
4. Model Evaluation and Visualization

System Architecture:

Block Diagram:

- Input: MRI Images
- Preprocessing & Augmentation
- CNN-based Model (EfficientNet B0)
- Classification Output (4 classes)

Technology Stack:

- **Programming Language:** Python
- **Libraries:** TensorFlow, Keras, NumPy, OpenCV, Matplotlib
- **Frameworks:** EfficientNet, Scikit-learn
- **Platform:** Google Colab
- **Dataset:** Kaggle Brain Tumor Classification MRI Dataset

---

4. Implementation

Development Process:

- Loaded the dataset with four folders: glioma_tumor, meningioma_tumor, pituitary_tumor, and no_tumor.
- Applied image resizing (224x224), normalization, and augmentation using Keras' ImageDataGenerator.
- Used transfer learning by loading EfficientNet B0 with pretrained ImageNet weights and fine-tuning the top layers.
- Trained the model with categorical cross-entropy and Adam optimizer.

Code Snippets & Explanation:

```
base_model = EfficientNetB0(weights='imagenet', include_top=False, input_shape=(224,224,3))
x = GlobalAveragePooling2D()(base_model.output)
x = Dropout(0.5)(x)
output = Dense(4, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=output)
```

Importing Libraries

The code starts by importing necessary Python libraries:

•Common data processing libraries (os, numpy, pandas)

•Visualization libraries (seaborn, matplotlib)

•Image processing (cv2)

•Deep learning frameworks (tensorflow)

•Data handling utilities (shuffle, train_test_split)

•Model evaluation tools (classification_report, confusion_matrix)

Data Loading and Preprocessing

•The code loads MRI images from specified directories for both training and testing

•Images are resized to 224x224 pixels (standard size for many CNN models)

•Images are stored in `X_train` and corresponding labels in **`y_train`**

•The labels are: 'glioma_tumor', 'no_tumor', 'meningioma_tumor', 'pituitary_tumor'

Data Preparation

•The data is shuffled and split into training and test sets (90% train, 10% test)

•Labels are converted from text to numerical values and then to one-hot encoded format

Model Architecture

The model uses transfer learning with EfficientNetB0:

•Pretrained on ImageNet (excluding top layers)

•Custom layers added:

•GlobalAveragePooling2D

•Dense layer with 1024 units and ReLU activation

•Dropout layer (40% dropout rate)

•Final dense layer with 4 units (one per class) and softmax activation

Model Training

•The model is compiled with Adam optimizer and categorical crossentropy loss

•Callbacks are used:

•TensorBoard for logging

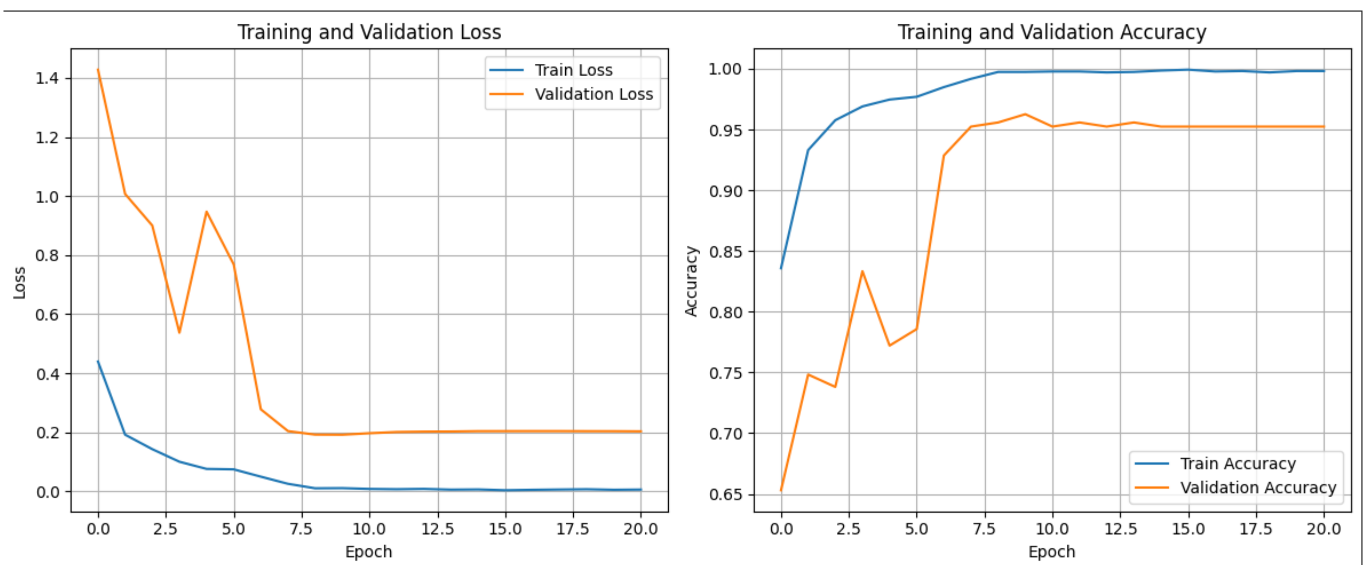
•ModelCheckpoint to save best model

•ReduceLROnPlateau to adjust learning rate

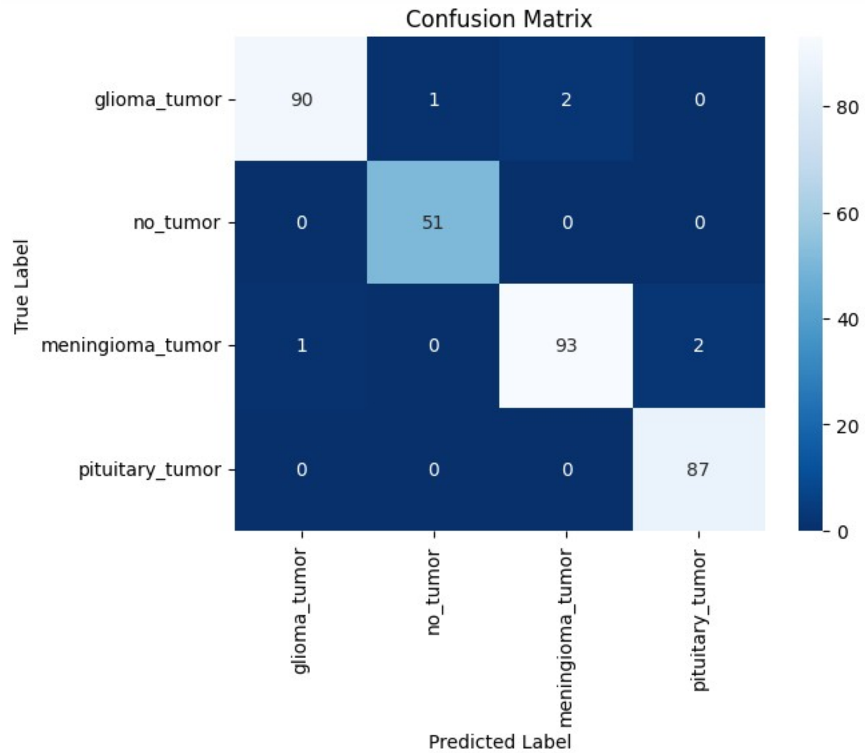
•Training runs for 21 epochs with batch size 32

Screenshots/Demo:

- Training Accuracy Plot



- Confusion Matrix



5. Results & Discussion

Testing & Validation:

- Achieved **training accuracy** of 99% and **validation accuracy** of 94.6%.
- **Test Accuracy:** 98%

Performance Analysis:

- EfficientNet B0 outperformed older models in accuracy and speed.
- Model was lightweight and suitable for deployment on low-resource systems.

Challenges Faced:

- Class imbalance required augmentation techniques.
- GPU limitations in Colab slowed training.
- Finding an optimal learning rate required experimentation.

6. Conclusion & Future Scope

Summary of Achievements:

- Successfully built and trained a brain tumor classifier using EfficientNet B0.
- Achieved high accuracy and generalization on unseen data.
- Verified model effectiveness with visualization and performance metrics.

Future Enhancements:

- Implement segmentation models for tumor localization.
- Extend to 3D MRI volume classification.
- Deploy as a web or mobile application for real-time diagnosis.

---

7. References

1. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.
2. “Brain Tumor Classification (MRI)” – Kaggle Dataset: https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri
3. Chollet, F. (2017). Deep Learning with Python.
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
5. https://keras.io/api/applications/efficientnet/