Classification of Brain Tumor Types Using Convolutional Neural Networks

by A.V, D. M

Abstract - **This paper delineates the methods and results of a project focused on classifying several types of brain tumors using Convolutional Neural Networks (CNN), particularly 4 different network architectures – ResNet, MobileNet, DenseNet, and EfficientNet. Data for this study was sourced from two different sets of MRI images. Results highlight the effectiveness of the implemented models and the practical application of a Fast API web service for real-time inference written in Python.**

I. Introduction

THE primary objective of this project is to automate the classification of brain tumors using machine learning and deep learning techniques. A web service, built with Fast API, provides an interface for real-time predictions. This project seeks to demonstrate the feasibility and accuracy of using CNN in medical diagnostics, while offering a user-friendly interface for utilizing the model.

II. Methodology

The objective of this project is the classification of brain tumors using several Convolutional Neural Network (CNN) with the architectures and analyzing which one performed the best. The project leveraged several Python libraries, such as TensorFlow for model training, Fast API for API creation, and various data manipulation and visualization libraries like Pandas and Matplotlib. The Fast API web service was deployed locally for testing. The architectures were chosen due to their proven ability in image classification tasks, which will be explained further.

EfficientNet was designed with compound scaling in mind, which means it scales not just the depth of the model but also the width and the resolution of the input. This three-dimensional scaling results in models that are optimized for both accuracy and computational efficiency.

MobileNet, on the other hand, is constructed around the idea of depth-wise separable convolutions. These convolutions split the standard convolutional operation into depth-wise and point-wise processes, drastically reducing computational costs. This makes MobileNet particularly suitable for mobile devices and edge computing, where computational resources are limited.

ResNet uses a method known as ‘skip connections’ to jump over certain layers – effectively mitigating the vanishing gradient problem. As a result, ResNet’s can be trained to have hundreds, or even thousands, of layers without succumbing to the challenges posed by deep architectures.

DenseNet connects each layer to every other layer in a feed-forward fashion. This means that every layer receives feature maps from all preceding layers, fostering feature reuse. This dense connectivity results in improved gradient flow, lower parameter counts, and enhanced feature propagation, making DenseNet particularly adept at extracting intricate patterns from data.

CustomNet is a custom convolutional neural network built specifically for this project's purpose. This model employs a combination of convolutional and dense layers. The models convolutional and dense blocks have convolutional layers with batch normalization, Leaky ReLU activation, and max pooling. The number of filters increases progressively.

In summary, the performance of each model will vary based on the nature of the dataset, computational resources, and specific task requirements. It is these inherent architectural differences that can lead to variations in performance across tasks and datasets, as will be demonstrated later in this paper.

When it comes to callbacks, we leveraged two particularly effective ones: Early Stopping and ReduceLROnPlateau. Early Stopping monitors a specified metric, in our case the validation loss, during the model's training. When the metric stops improving for a certain number of epochs, the training process is halted. ReduceLROnPlateau applies a similar kind of logic, except it reduces the learning rate of the model when the validation loss stops improving.

The models are structured in a series of blocks, where each block consists of a set of convolutional layers followed by batch normalization and a ReLU activation function. These blocks are grouped into 'bottleneck layers' to reduce the number of parameters and computational complexity.

Choosing these network architectures for the brain tumor classification task offers deep architectures which allows for the learning of high-level abstract features from MRI images, essential for distinguishing between subtle differences in tumor types, and the residual connections facilitate the backpropagation of gradients, ensuring that our models learn effectively even in its deeper layers.

In our project, the different network architectures were fine-tuned to cater to our specific classification task. The final fully connected layer in each network was adapted to output four classes, corresponding to glioma, meningioma, pituitary, and no tumor. The models were trained using a combination of the balanced dataset, an appropriate loss function to measure discrepancies between predicted and actual labels, and an optimization strategy to iteratively improve model weights.

To train the models, a dataset comprising MRI images of different brain tumor types was gathered. The different classes of tumors included were glioma, meningioma, pituitary, and no tumor. These images serve as the foundation for our supervised learning strategy.

For each type of tumor, at least 2500 MRI images were used, creating a balanced dataset that aids in reducing model bias. Given the nature of MRI images, feature normalization was an essential preprocessing step. All pixel values were normalized to the range of 0 to 1, ensuring that the model would not be disproportionately influenced by high pixel values. The images were visualized using matplotlib, to review the initial data and its features.

One of the most critical aspects of building a reliable and interpretable Convolutional Neural Network (CNN) model is understanding which regions in an input image contribute the most to the model’s decision. This understanding is especially crucial in medical applications like brain tumor classification, where interpretability can offer additional trust and validation of the model's predictions.

To provide this insight, we utilized Class Activation Maps (CAMs) as a visualization technique after training each model. CAMs allow us to see the regions in the MRI images that the CNN model focuses on during the classification process. These heatmaps are generated by projecting the output of the final convolutional layer back onto the input space, effectively highlighting the critical areas in the original MRI image.

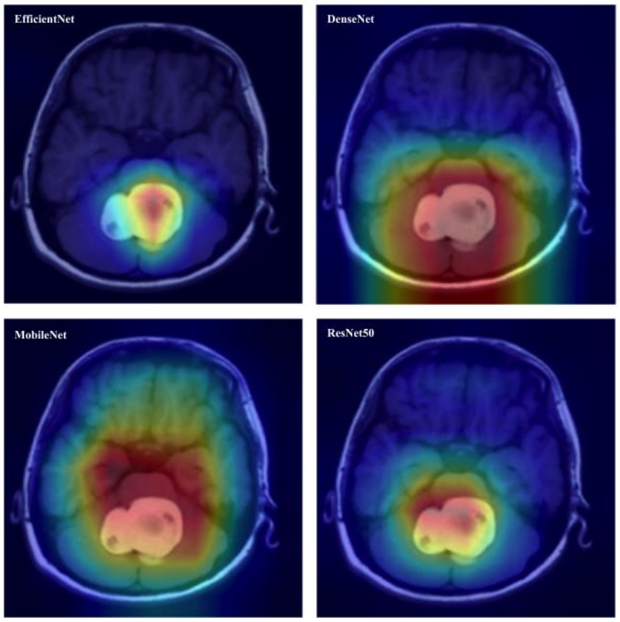

Fig. 1. Class Activation Map for Meningioma

The heatmap shows that the models focus on one specific part of the MRI image, namely the part consisting of the tumor area.

One tumor type which was exceedingly difficult for most models to classify correctly was pituitary tumors. According to these activation maps, some of the areas the models deem the most influential are not representative.

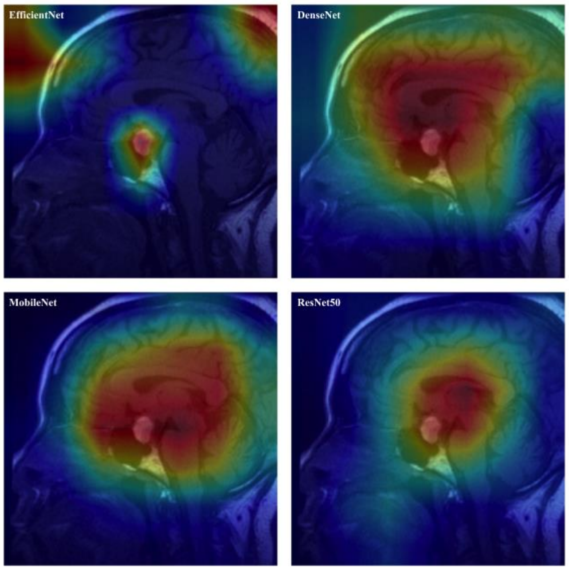

Fig. 2. Class Activation Map for Pituitary tumors

Another interesting metric is the Mean Pixel Value Distribution Across Classes. A histogram of this offers compelling insights into the characteristics of MRI images from different tumor types. Notably, there is a significant spike in images with a mean pixel value around the 30 range. This suggests that a massive portion of MRI scans across various categories have darker or lower intensity regions.

These could be indicative of non-tumorous sections of the brain or regions within the tumor with lower radiological density.

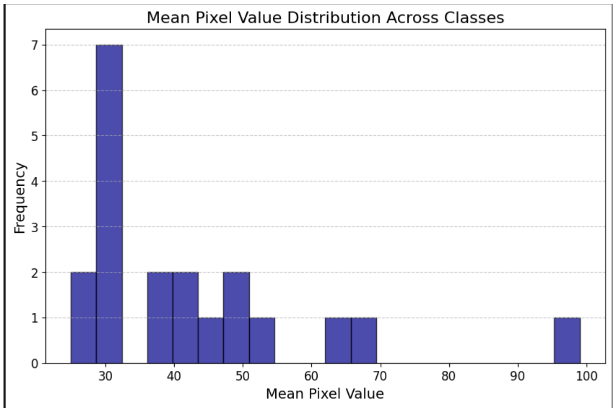

Fig. 3. Mean Pixel Value Distribution Across Classes

Furthermore, as we move to higher mean pixel values, the frequency distribution becomes more varied, with images from the 40 to 60 range being moderately represented, while there is a notable decrease in frequencies as we approach the 80 to 100 range. This could suggest that brighter or high-intensity regions, which could correspond to more aggressive or dense tumor sections, are less prevalent in the test set. Such insights are crucial for understanding the inherent intensity characteristics of MRI images and can have implications in preprocessing and model training, as balancing the input data's intensity distribution might be a consideration.

Analyzing the Class Distribution in the Training Dataset, we can deduce several observations regarding the composition of our data. The glioma and meningioma classes are well-represented, each with around 3000 images. The pituitary class also demonstrates a substantial representation, albeit slightly less than the former two. However, the MRI images representing an absence of tumors are slightly underrepresented, indicating a potential class imbalance.

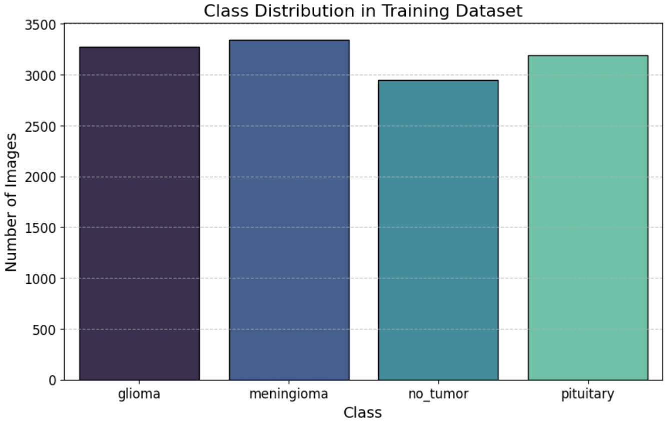

Fig. 4. Class Distribution in Training Dataset

This discrepancy in class distribution is critical, as models trained on imbalanced datasets might exhibit biases towards over-represented classes. For instance, our model may become particularly adept at detecting gliomas and meningiomas due to the abundance of data, while struggling or being less confident in identifying the ‘no tumor’ cases due to its comparative scarcity. Addressing this imbalance might necessitate strategies such as oversampling the under-represented class, employing weighted loss functions, or other data augmentation techniques to ensure robust and unbiased model performance.

To operationalize the trained models, Fast API was chosen as the web service framework. It not only offers an asynchronous model but also automatically creates API documentation, making the service more user-friendly. For evaluating the four models, the following metrics were used: Loss, Accuracy, and Area Under the Curve (AUC). These metrics provide a comprehensive view of how well the model generalizes to unseen data.

III. RESULTS

Our classification models showed accuracies ranging from 60% to 97%. The worst performing model uses the EfficientNet architecture, and the best performing model uses the ResNet50 architecture. The most problematic model is the EfficientNet model, for example it reaches a recall of 0.99 for class ‘no tumor,’ but only manages to recall 0.05 of pituitary tumors, as can be visually deduced from the class activation map in Fig. 2. However, even with a high recall for the ‘no tumor’ class, it comes with the cost of lower precision – indicating that the model is more prone to false positives. It seems as if the areas where the model focuses are too dispersed and not representative enough of the type of tumor it is classifying.

Model	Accuracy	AUC	Loss
EfficientNet	59.51%	83.19%	1.12
DenseNet	95.54%	99.23%	0.19
MobileNet	95.60%	99.38%	0.20
ResNet	96.89%	99.64%	0.16

Table 1. Model Evaluation Results

Another explanation for the model’s inferior performance could be due to the number of parameters the EfficientNet has, causing the model to overfit the data.

We will analyze and focus on the best and the worst models; ResNet and EfficientNet, respectively. Each row corresponds to the actual tumor classes, and each column indicates the predicted classifications. The labels 0 to 3 signify glioma, meningioma, no tumor, and pituitary tumors, respectively.

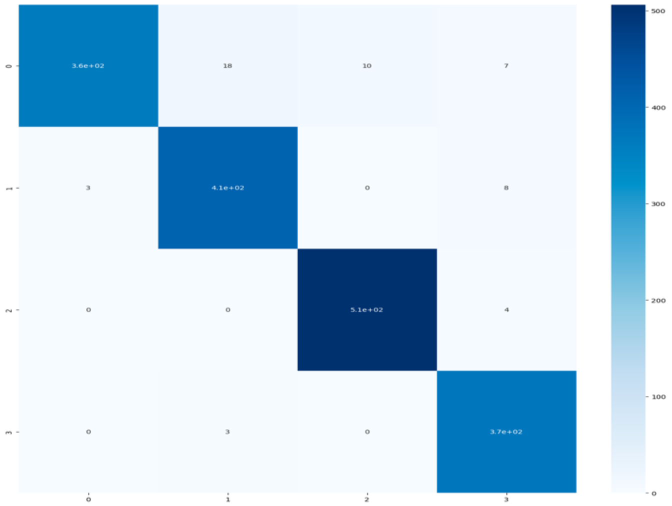

Fig. 5. Confusion Matrix for ResNet50 Model Performance

The matrix's diagonal highlights the correct predictions: notably, the class ‘no tumor’ observed the highest accuracy with 510 correct predictions, followed closely by class ‘meningioma’ at 400, the class ‘pituitary’ at 370, and the class ‘glioma’ at 340. The matrix also reveals areas where the model faced challenges. For instance, the class glioma had 33 instances misclassified as no tumor, and meningioma had 12 instances misinterpreted as no tumor.

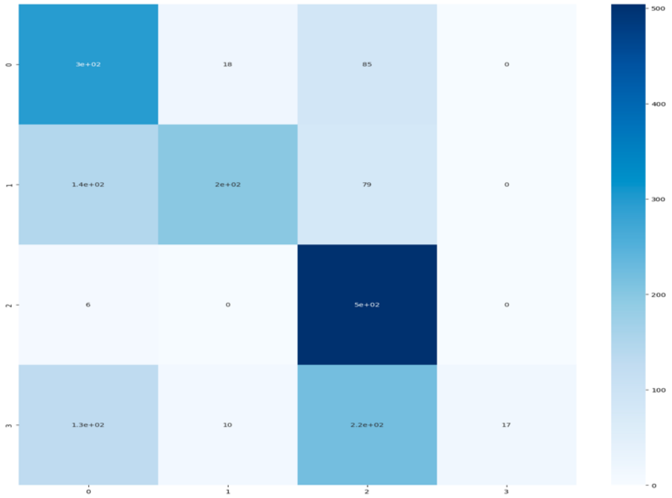

Fig. 6. Confusion Matrix for EfficientNet Model Performance

Observing the diagonals, the model most accurately predicts when an image displays no signs of a tumor, with around 500 true positive predictions. The other true positive predictions lay in the range of 130 to 300. There are many cases where the model misclassifies pituitary tumors. The predictions for other classes also show many misclassifications. Given the exceeding misclassifications between certain classes, particularly between pituitary and non-tumors, further investigation is needed to understand why these misclassifications are happening. More data or different class weighting could help improve the model’s performance.

Our CustomNet model achieved accuracy of around 83%. We consider this a decent result in comparison to the previous models that have their own predefined weights. The execution time of this model falls somewhere between the execution time of EfficientNet and DenseNet. Below we can see the produced confusion matrix (Fig. 7.):

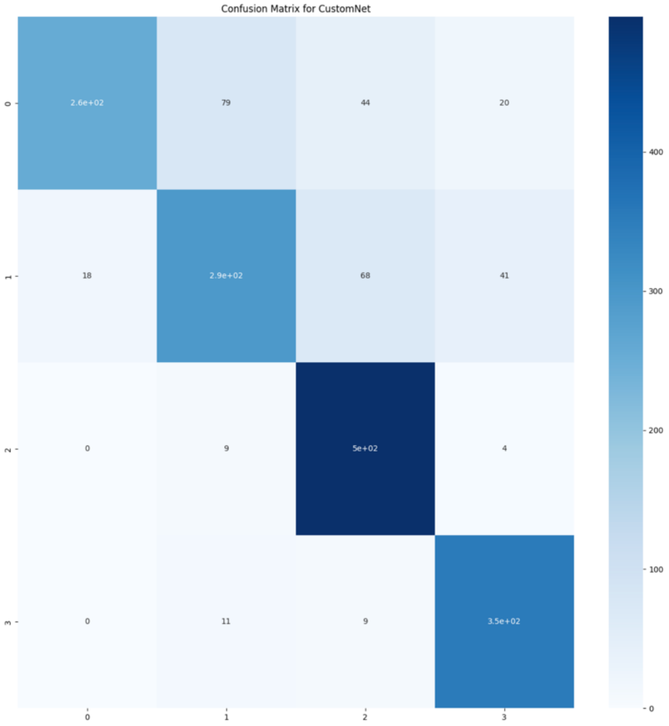

Fig. 7. Confusion Matrix for CustomNet Model Performance

Observing the diagonals, the model most accurately predicts when an image displays either no signs of a tumor or signs of a pituitary tumor, having around 500 true positive predictions and 350 true positive predictions, respectively. The model performs worst in the first two rows of the confusion matrix, scoring 290 true predictions for the “meningioma” class and 260 true predictions for the “glioma” class. Our custom model miss-classifies those two classes which could be improved by further fine tuning.

CustomNet has the highest loss value of all our trained models which means that the model could be even further improved more so than the previous models (Table 2.):

Model	Accuracy	AUC	Loss
CustomNet	83.30%	96.18%	5.02

Table 2. Custom Model Evaluation Result

To improve the models, we tried improving the training by using K-fold cross validation together with hold-out validation. In theory each model should be improved due to several reasons; determining the appropriate number of skip connections for ResNet, optimizing depth-wise separable convolutions for MobileNet, determining the optimal scaling coefficients, network width, depth, and resolution for EfficientNet and so on. In our approach we ran the models through a helper function that would train them with five-fold cross validation.

The training time for all models overall was much longer, due to using k-fold cross validation. When it comes to accuracy results and metrics, we noticed a substantial increase in the performance of EfficientNet, scoring 92% accuracy after k-fold cross validation. When it comes to other models the increase was also large, averaging around 3% - 4% increase in accuracy across all models. This is a significant increase and improves the accuracy of predictions greatly.

Model	Accuracy	Prev. Acc.	Difference	Loss
EfficientNet	92.07%	59.51%	32.56%	0.51
DenseNet	99.76%	95.54%	4.22%	0.0506
MobileNet	99.83%	95.60%	4.23%	0.0514
ResNet	99.87%	96.89%	2.89%	0.0473

Table 3. Model Accuracy Using KFCV

IV. Discussion

The goal of this project was to automate classification of distinct types of brain tumors, specifically by using Convolutional Neural Networks (CNN's). The findings show that the architectural design and inherent features of CNN models play a significant role in the accurate classification of brain tumors given by MRI images.

The ResNet model architecture outperformed the three other models, with an accuracy of around 97%. However, DenseNet and MobileNet were not too far behind, at around 95.5% accuracy. The skip connections might have played a part in their superior performance, as it utilizes flow of information across layers, tackling the 'vanishing gradient' problem in deeper networks, as discussed in many other papers in the field of Machine Learning[1].

Even though EfficientNet also uses skip connections and a sophisticated scaling approach (compound scaling), our specific model seemed to not fit our data very well.

All of the models consistently struggled with pituitary tumors, especially the EfficientNet model, which was demonstrated in the class activation map earlier.

[1] *Basodi, Sunitha; Ji, Chunyan; Zhang, Haiping; Pan, Yi (September 2020). "Gradient amplification: An efficient way to train deep neural networks". Big Data Mining and Analytics.* ***3*** *(3): 198. doi:10.26599/BDMA.2020.9020004. ISSN 2096-0654. S2CID 219792172.*