Performance Analysis of baseline CNN and Pre-Trained Deep Learning Models for Multi-Class Brain Tumor Detection

Abstract:

Accurate and efficient brain tumor classification from MRI images is essential for early diagnosis and improved clinical decision-making. This study evaluates the performance of a baseline Convolutional Neural Network (CNN) trained from scratch against three state-of-the-art transfer learning models; ResNet50, DenseNet121, and EfficientNet-B0 to determine their effectiveness in multi-class brain tumor detection. Using a balanced dataset comprising glioma, meningioma, pituitary, and no-tumor images, all models were trained under identical conditions and assessed using accuracy, F1-score, class-wise performance, and computational efficiency. Results show that transfer learning models significantly outperform the baseline CNN, achieving faster convergence, stronger generalization, and reduced computational cost. EfficientNet-B0 achieved the highest validation accuracy (0.9297) and test accuracy (0.9271), along with superior class-wise performance, particularly in the visually challenging meningioma class. Despite balanced class distribution, meningioma remained difficult to classify due to subtle feature overlap with other tumor types. Overall, the findings demonstrate that transfer learning. especially EfficientNet-B0 provides a highly effective and efficient approach for MRI-based brain tumor classification, offering strong potential for real-world medical imaging applications.

1. Introduction

Brain tumors represent one of the deadliest cancer types which require immediate detection and classification to improve patient outcomes and treatment effectiveness. Brain tumor detection through proper classification methods enables doctors to start treatment early which leads to better survival rates and fewer complications. The current diagnostic methods for brain tumors depend on manual MRI scan interpretation which produces long and subjective results. Medical diagnosis now depends heavily on automated and intelligent systems because medical imaging datasets continue to grow in complexity and size. Medical image analysis has experienced substantial progress through deep learning methods which emerged as a leading approach during the previous several years. By integrating AI technology in the problem domain, we can significantly simplify the process of finding effective solutions [1].

The visual data processing abilities of Convolutional Neural Networks (CNNs) have achieved outstanding results in medical image analysis through their ability to learn hierarchical representations which leads to enhanced tumor detection and classification performance [2],[3]. The model requires improvement for its ability to recognize new data patterns and its speed of operation during real-time processing[4]. In this context, transfer learning can help leverage pre-trained models, saving time and computational resources while improving model accuracy. Moreover, application of transfer learning on CNN models, particularly pre-trained on ImageNet dataset improves generalization performance [5],[6]. This study aims to evaluate the effectiveness of transfer learning in the context of brain tumor detection by comparing the performance of a baseline CNN trained from scratch and pre-trained models: ResNet, DenseNet and EfficientNet. These models will be trained and fine-tuned on an MRI dataset from Kaggle with 2176 labeled brain scan images classified into four classes: glioma, meningioma, pituitary, and no tumor.

The key contributions of the work are:
1) A comparative evaluation of various CNN models applied to medical imaging.
2) Improved generalization via transfer learning on unseen medical data.
3) A comprehensive evaluation of trade-off between predictive precision and real-time inference capability.
4) Detailed tumor level model performance analysis.

2. Methodology

2.1 Dataset details

The CNN models were trained using a comprehensive dataset of brain scans from several imaging modalities, including MRI. The dataset was carefully chosen to a wide range of clinical conditions. The dataset was gathered from Kagle and Hayatabad Medical Complex in Peshawar. The dataset's criteria include 550 healthy brain samples, 456 gliomas, 551 meningiomas, and 620 pituitary brain tumor samples. Each class in the dataset show different views, including axial, coronal, and sagittal.

The following data transformations were performed to make them suitable for CNN modelling.

- **Resizing**: Images were resized to 224x224 pixels.
- **Conversion to Tensor**: Images were converted into PyTorch tensors.
- **Normalization**: Pixel values were normalized using the standard ImageNet mean and standard deviation values
- **Data splitting**: The original dataset will be divided into 70:15:15 split for train, validation, and test datasets.

2.2 Baseline CNN architecture

The baseline convolutional neural network (CNN) built for image classification. It consists of several key components, which we will explain in detail below. The architecture is designed to work with images and classify them into one of several categories.

2.2.1. Convolutional Layers

The model uses three **convolutional layers**, which are the core building blocks of CNNs. Convolutional layers apply filters (kernels) to the input image to extract important features, such as edges, textures, and patterns.

- **Conv1**: The first convolutional layer takes an input image with **3 channels** (RGB, or Red, Green, and Blue), and it applies **16 filters** of size 3x3 to extract features. The padding is set to 1 to ensure the spatial dimensions (height and width) of the image are preserved.
- **Conv2**: The second convolutional layer takes the output from the first convolution (16 feature maps) and applies **32 filters** of size 3x3.
- **Conv3**: The third convolutional layer takes the output from the second convolution (32 feature maps) and applies **64 filters** of size 3x3.

These layers progressively learn more complex and abstract features of the input images, starting from simple patterns (like edges) in the first layer and moving to more complex features (like shapes or textures) in later layers.

2.2.2. Max-Pooling Layers

After each convolutional layer, we apply **max-pooling** to reduce the spatial size (height and width) of the feature maps. Max-pooling helps reduce the number of parameters, lowers the computational cost, and provides some degree of spatial invariance (the ability of the model to recognize patterns even when they are shifted or rotated).

- **Pool1**: This pooling layer follows the first convolutional block. It applies a **2x2 window** with a stride of 2, effectively halving the height and width of the feature map.

- **Pool2**: Similarly, this pooling layer follows the second convolutional block and also uses a **2x2 window** with stride 2, reducing the spatial dimensions.
- **Pool3**: The final pooling layer reduces the size of the feature map from the third convolutional block, again using a **2x2 window** with stride 2.

The pooling layers help retain the most important features while reducing the computational burden by decreasing the spatial dimensions of the feature maps.

2.2.3. Fully Connected Layers

After the convolutional and pooling layers, the model flattens the output (i.e., converts the 2D feature maps into a 1D vector) to pass it through **fully connected (dense) layers**. These layers learn to combine the extracted features and make the final predictions.

- **fc1**: This is the first fully connected layer, which has **512 neurons**. It takes the flattened feature maps (which have a size of 64 channels by 28x28 spatial dimensions) and outputs a vector of size 512. The activation function used is **ReLU** (Rectified Linear Unit), which helps introduce non-linearity into the model, allowing it to learn more complex patterns.
- **fc2**: This is the final fully connected layer, which outputs the predictions. Since the task is to classify the images into one of four categories (e.g., glioma, meningioma, pituitary tumor, and no tumor), this layer has **4 neurons** (one for each class). The activation function used here is typically a **softmax**, which outputs probabilities for each class.

2.2.4. Flow of Data

The flow of data through the SimpleCNN model can be summarized as follows:

1. **Input Image**: The model takes an image of size (e.g., 224x224x3 for RGB images).
2. **Convolution and Pooling**: The image goes through three convolutional layers (conv1, conv2, conv3), each followed by max-pooling (pool1, pool2, pool3). These layers progressively extract features and reduce the spatial dimensions.
3. **Flattening**: The output of the last pooling layer is a multi-dimensional feature map, which is flattened into a one-dimensional vector.
4. **Fully Connected Layers**: The flattened vector is passed through two fully connected layers (fc1 and fc2). The first fully connected layer (fc1) learns to combine the features, and the second fully connected layer (fc2) makes the final prediction.
5. **Output**: The output is a vector of size 4, representing the probabilities for each of the four classes (glioma, meningioma, pituitary tumor, no tumor).

2.3 Transfer learning models

In this work, three main transfer learning models were used for brain tumor classification: **ResNet50**, **DenseNet121**, and **EfficientNet-B0**. All these models were initially pre-trained on the large-scale **ImageNet dataset**.

The fine-tuning process for each of these models followed a similar, common transfer learning strategy [5],[6]:

1. **Loading Pre-trained Weights**: Each model was initialized with its respective pre-trained weights from pre-trained models.
2. **Freezing Base Layers**: To leverage the powerful feature extraction capabilities learned from ImageNet, we froze the weights of all layers except the final fully connected layer.
3. **Modifying the Classification Head**: The original final classification layer (or 'head') of each model, which was designed for ImageNet's 1000 classes, was replaced with a new layer. This new fully connected layer was configured to output 4 classes, corresponding to the specific brain tumor types ('glioma', 'meningioma', 'no tumor', 'pituitary') in our dataset.
4. **Training Only the New Head**: During the training phase, only the parameters of this newly added, randomly initialized classification head were updated. This strategy allows the model to quickly adapt to the new task using the highly relevant features learned by the pre-trained backbone, while minimizing computational cost and the risk of overfitting on the smaller brain tumor dataset.

This approach effectively transfers the general image recognition knowledge from ImageNet to the more specific task of brain tumor classification.

Training details and hyperparameters for the ResNet, DenseNet, and EfficientNet models are as follows:

- **Optimizer:** Adam
- **Learning Rate:** 0.001
- **Loss Function:** Cross Entropy Loss
- **Number of Epochs:** 40 epochs
- **Early Stopping Patience:** 5 epochs
- **Batch Size:** 32
- **Layers Trained:** Only the final fully connected layer (transfer learning)

3. Experiments and results

3.1 Training/ Validation curves

3.1.1 Baseline CNN

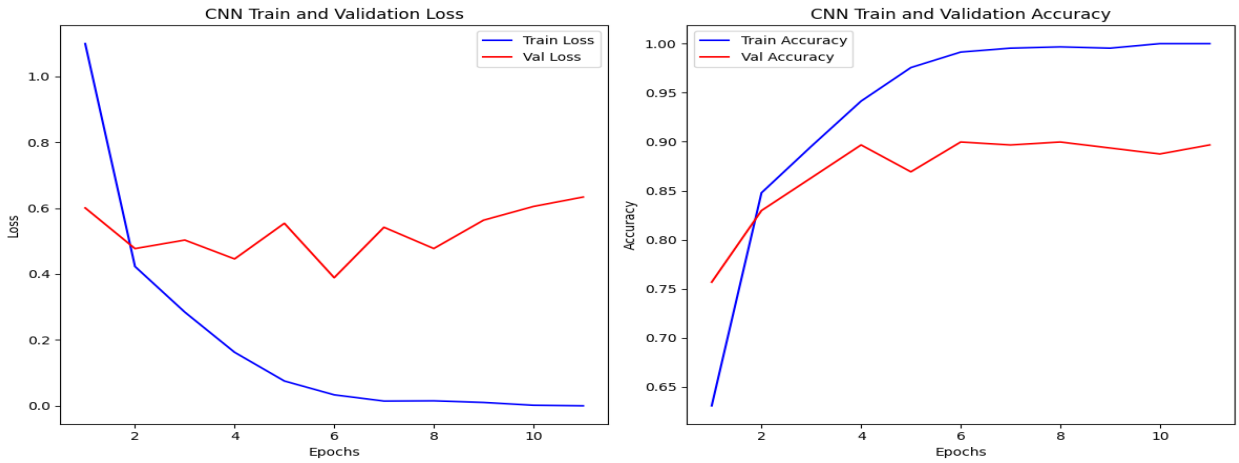


Figure 1. Baseline CNN Training Curves

- There is strong evidence of overfitting in this model. The training loss continues to drop to almost zero and training accuracy reaches 100%, indicating the model is memorizing the training data. Simultaneously, the validation loss starts to fluctuate and slightly increase after epoch 6, and validation accuracy increases steadily in the early epochs, reaching its peak of 0.8997 at epoch 6, and then slightly fluctuates around this value. The early stopping mechanism correctly identified this trend and stopped training at epoch 11 when validation accuracy didn't improve for 5 consecutive epochs.

3.1.2 Resnet50

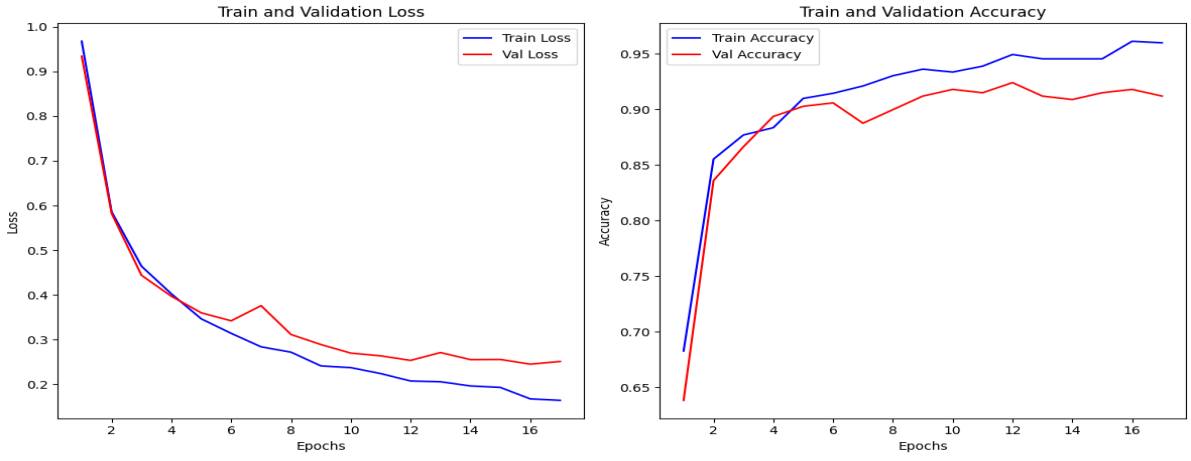


Figure 2. ResNet50 Training Curves

Both the training loss and validation loss curves show a rapid decrease in the initial epochs, indicating that the model quickly learned to make better predictions. The validation loss started to flatten out around epoch 12, and early stopping was applied in epoch 17. Similarly, both training accuracy and validation accuracy increased sharply in the initial epochs. The training accuracy continued to climb, eventually reaching very high levels (close to 0.96), while the validation accuracy also showed consistent improvement but plateaued around 0.9-0.92.

3.1.3 DenseNet121

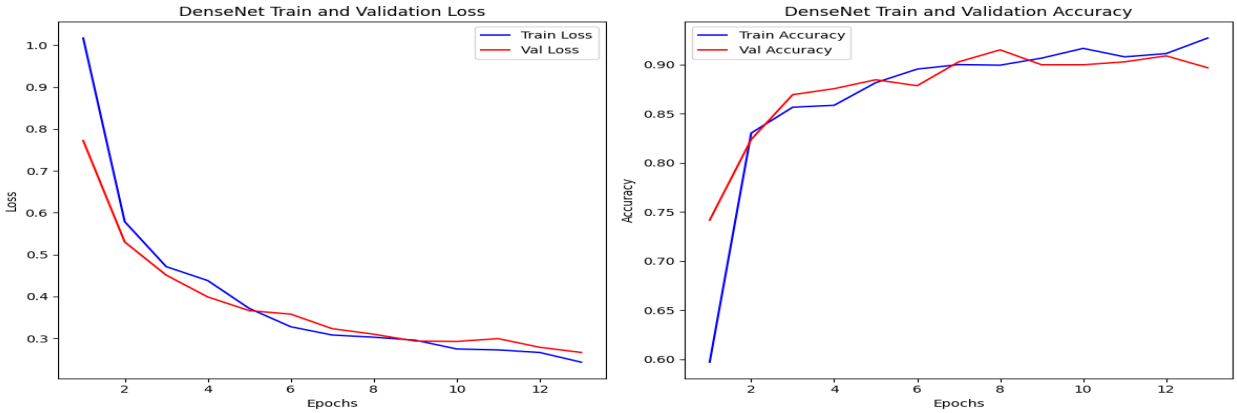


Figure 3. DenseNet121 Training Curves

The training loss shows a consistent and relatively steep decrease from Epoch 1to Epoch 13. The validation loss also shows a general downward trend, decreasing from 0.7722 in Epoch 1 to its lowest point around 0.2662 in Epoch 13. However, validation accuracy after reaching its peak of 0.9149 around Epoch 8, shows fluctuations and early stopping is triggered.

3.1.4 EfficientNet-B0

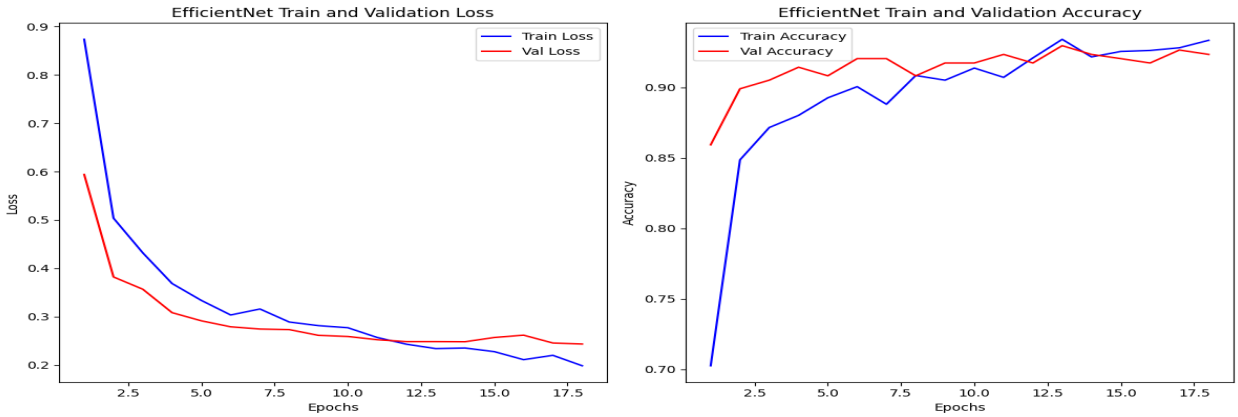


Figure 4. EffiecientNet-B0 Training Curves

The training loss decreases rapidly in the initial epochs and continues to decrease, while the validation loss also decreases significantly, then stabilizes around epoch 11-13 before a slight

increase. Early stopping mechanism effectively prevents severe overfitting by halting training when validation accuracy after reaching peak value 0.9297 and no longer improves significantly.

3.1.5 Overall Comparisons

Pre-trained models (ResNet, DenseNet, EfficientNet) significantly outperformed the custom SimpleCNN in terms of generalization and efficiency. They reached high accuracy much faster and maintained better stability, confirming the immense benefits of transfer learning for medical image classification tasks with potentially limited datasets. Among the pre-trained models, **EfficientNet** emerged as the top performer, achieving best validation accuracy (0.9297).

3.2 Test Accuracy

Here also, **EfficientNet achieved the highest test accuracy** of 0.9271 among all models. And in terms of Precision, Recall and F1-Score it showed best performance as shown in the table-1. This indicates superior generalization capability, attributed to its efficient architecture and effective transfer learning. **Following EfficientNet, DenseNet shows better results** with accuracy 0.9174. ResNet and SimpleCNN show similar test results despite the lower validation performance of baseline CNN. It can be primarily attributed to the effective use of early stopping, which preserved the model weights from the point of best generalization. The test set's characteristics likely also contributed to this performance, suggesting it was well-aligned with the model's learned patterns at its optimal state.

Table 1. Test set performance of selected models.

Model	Accuracy	Precision	Recall	F1-Score
EfficientNet	0.9271	0.9271	0.9271	0.9264
DenseNet	0.9174	0.9180	0.9174	0.9169
ResNet	0.9144	0.9159	0.9144	0.9133
BaselineCNN	0.9144	0.9143	0.9144	0.9138

3.2 Class wise Performance

All models show generally strong class-wise performance, with F1-scores often above 0.90 for most classes. Consistently across all models, **'meningioma' tends to have the lowest precision, recall, and F1-score compared to other classes.** This suggests it might be the most difficult class

for the models to distinguish, indicating potential areas for further improvement or specific data augmentation strategies. **Class "pituitary" performs strongly** among all models with the highest precision, recall and F1-scores. **EfficientNet demonstrates very competitive performance** across all classes, often achieving some of the highest metrics or being very close to the top, reinforcing its strong overall performance.

Table 2. Class wise performance of models on test data

Class	Metric	SimpleCNN	ResNet	DenseNet	EfficientNet
Glioma	Precision	0.9375	1.0000	0.9841	0.9692
	Recall	0.8824	0.8971	0.9118	0.9130
	F1-Score	0.9091	0.9457	0.9466	0.9403
Meningioma	Precision	0.8861	0.8625	0.8734	0.8846
	Recall	0.8434	0.8313	0.8313	0.8313
	F1-Score	0.8642	0.8466	0.8519	0.8571
Notumor	Precision	0.8736	0.9000	0.9070	0.8989
	Recall	0.9157	0.9759	0.9398	0.9639
	F1-Score	0.8941	0.9364	0.9231	0.9302
Pituitary	Precision	0.9588	0.9271	0.9192	0.9588
	Recall	1.0000	0.9570	0.9785	0.9894
	F1-Score	0.9789	0.9418	0.9479	0.9738

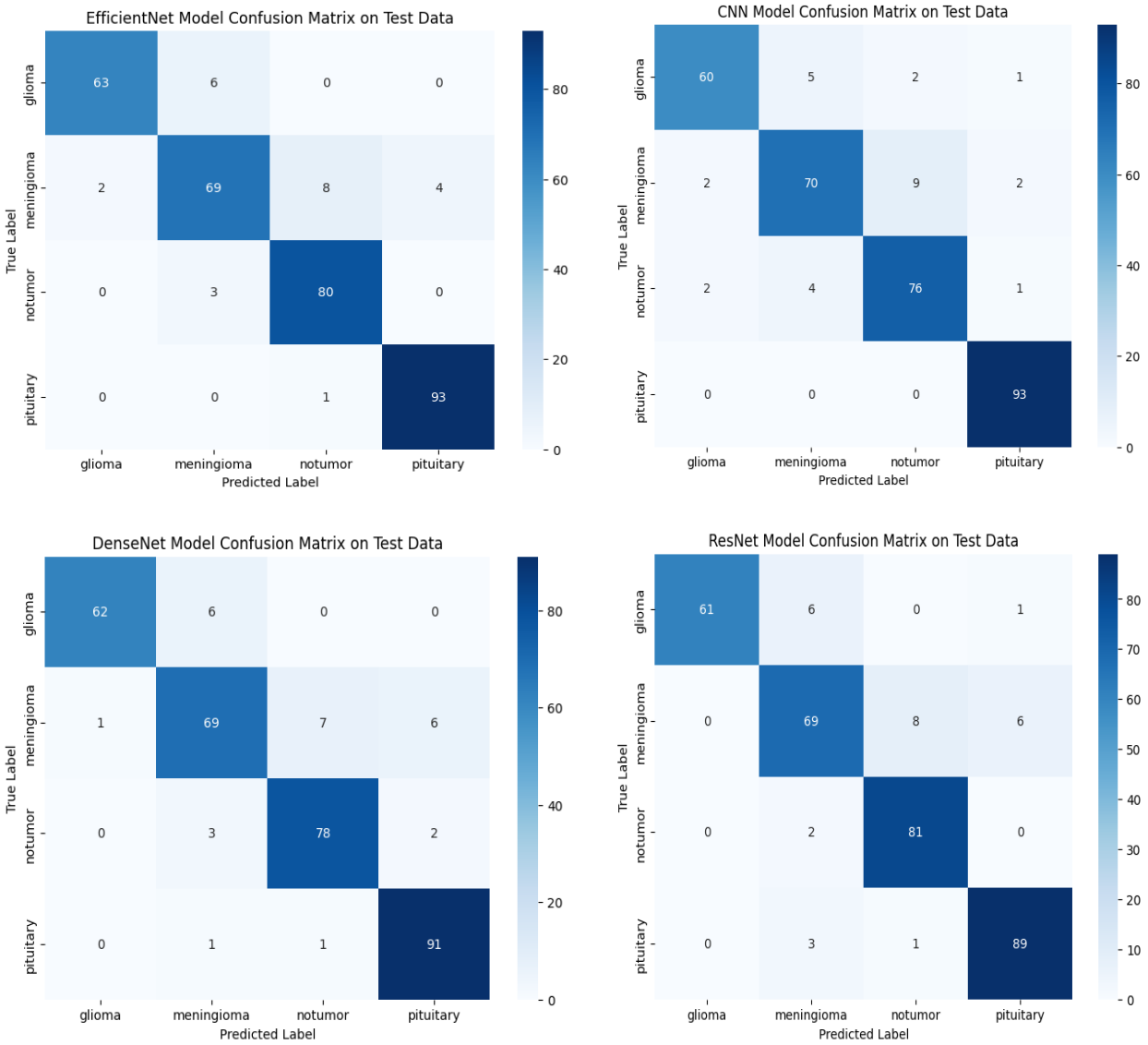
3.3 Top-k accuracy

Model	Top-1 Accuracy	Top-2 Accuracy	Top-3 Accuracy
EfficientNet	0.9633	0.9969	0.9969
DenseNet	0.9174	0.9878	0.9969
ResNet	0.9144	0.9817	0.9969
SimpleCNN	0.9144	0.9725	0.9969

EfficientNet clearly leads with a Top-1 Accuracy of 0.9633. This means it makes the absolute best prediction correctly most often. Its Top-2 Accuracy is 0.9969, indicating that for nearly all remaining cases, the true class is its second-best guess. This makes it a highly reliable model. DenseNet performs very well with a Top-1 Accuracy of 0.9174 and a strong Top-2 Accuracy

of 0.9878. This suggests that even when its top prediction is incorrect, it often ranks the true class as its second choice. ResNet and SimpleCNN have comparable Top-1 Accuracies (0.9144 each), but ResNet shows a slightly better Top-2 Accuracy (0.9817) compared to SimpleCNN (0.9725). This implies that ResNet's second best guess is more frequently correct than SimpleCNN's. All models achieved a very high Top-3 Accuracy of 0.9969. This implies that for 99.69% of the test images, the true class was among the top three most probable predictions.

3.4 Confusion Matrices



Across all models, misclassifications (false negative) are very high in class “meningioma” and very less in class “pituitary”. Across all models, EfficientNet seems to perform the best across the board with high recall and low false positives.

3.5 Efficiency Analysis

Model	Total Trainable Parameters	Average train time per epoch (s)	Average inference time per epoch (s)
Baseline CNN	25,716,260	22.44	5.43
ResNet	8,196	10.66	3.19
DenseNet	4,100	11.94	2.55
EfficientNet	5,124	10.69	2.52

The SimpleCNN model, trained from scratch, has a significantly larger number of trainable parameters (25.7 million) compared to the transfer-learned models (ResNet, DenseNet, EfficientNet). This is because for the pre-trained models, only the final classification layer was trained, while the vast majority of their layers were frozen. Among the transfer-learned models, ResNet has the highest number of trainable parameters (8,196), followed by EfficientNet (5,124), and DenseNet has the fewest (4,100).

Average train time and inference time are higher for baseline CNN. This could be due to the large number of parameters compared with pre-trained models. EfficientNet had the lowest inference time that was followed by DenseNet. **For high performance with less computational cost, EfficientNet and DenseNet are strong contenders, especially for real-time inference.**

3.6 Overall comparison for baseline CNN with transfer learning models

3.7 Explainable AI of EfficientNet

We generated and displayed a saliency map for a sample test image to visualize which pixels are most influential in the EfficientNet's predictions. It highpoints the area where tumor present [7].

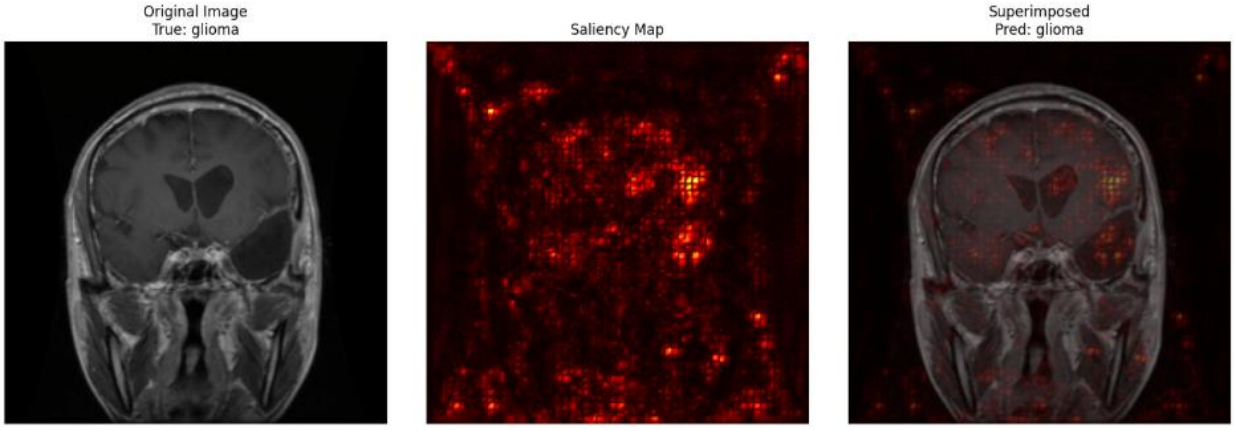


Figure 5. Saliency map of EfficientNet model on glioma image

Similarly, we identified a selection of correctly classified and misclassified images from the EfficientNet model's predictions for Grad-CAM visualization. After that, Grad-CAM heatmaps were displayed for the selected correctly classified and misclassified images using the EfficientNet

model. This will visually highlight regions of input images that were most influential in the model's predictions.

For correctly classified images, the Grad-CAM heatmaps typically highlight relevant regions in the brain where the tumor (or lack thereof) is present, aligning well with human intuition and medical expertise. This indicates that the model is focusing on the correct features to make its predictions. For example, in a 'glioma' image, the heatmap might concentrate on the glioma region, or in a 'notumor' image, it might show a diffuse pattern across healthy brain tissue.

In misclassified cases, Grad-CAM often reveals that the model is either focusing on irrelevant background features, or on a part of the tumor that is ambiguous, or even on a different lesion/area that it incorrectly associates with the predicted class. For example, an image of 'glioma' predicted as 'meningioma' might have the heatmap focusing on some normal brain structures that vaguely resemble meningiomas to the model, or there could be artifacts in the image drawing the model's attention.

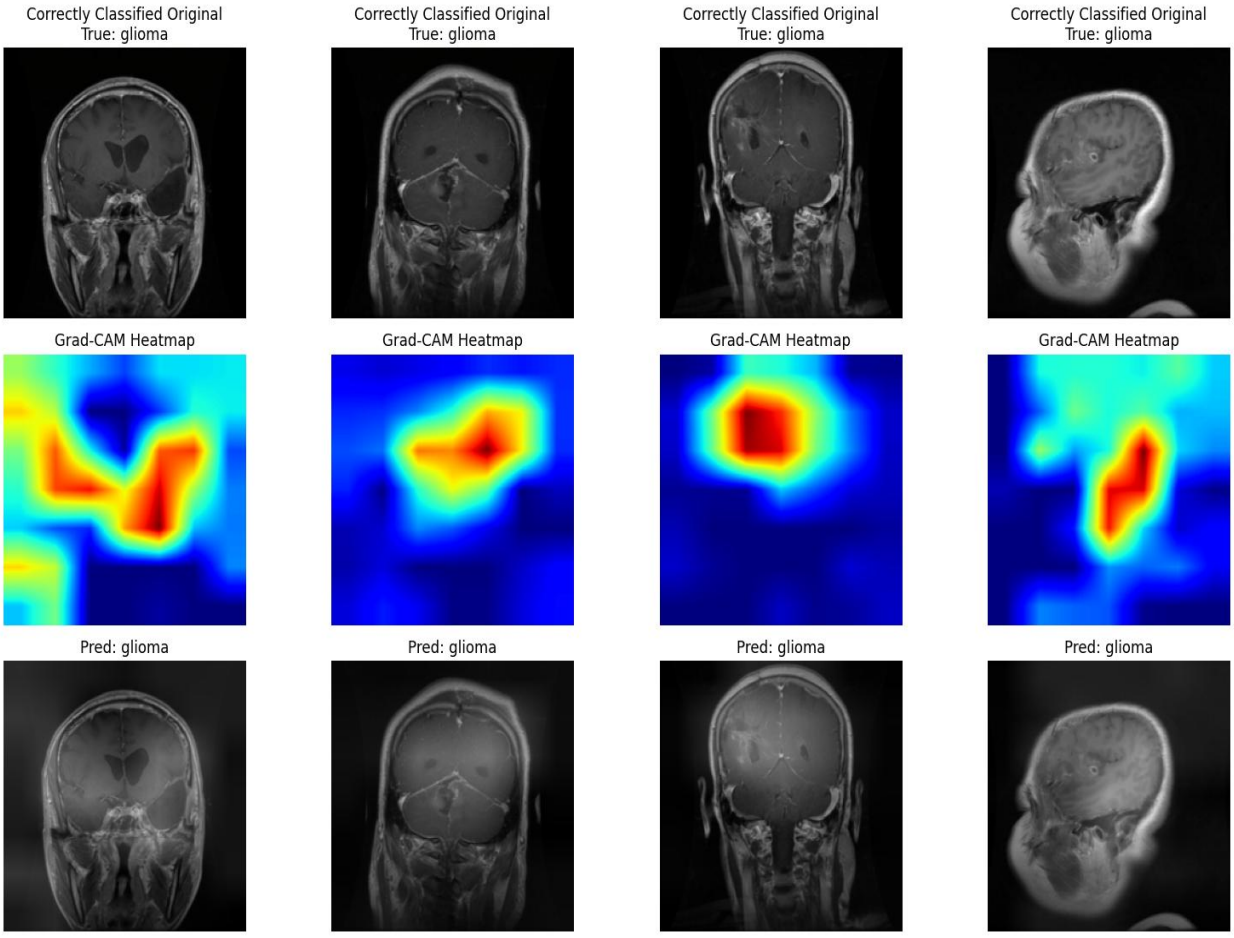


Figure 6. Grad-CAM maps of correctly classified images.

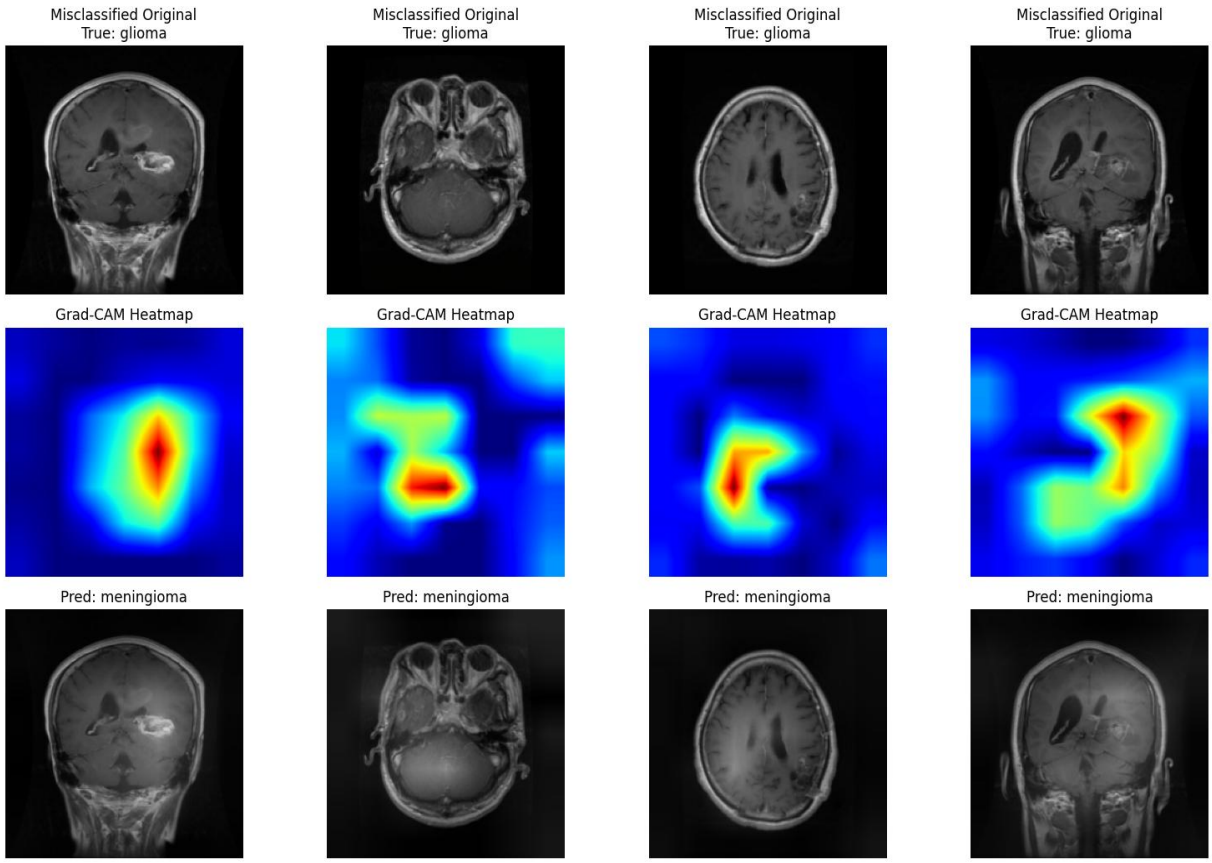


Figure 7. Grad-CAM maps of mis-classified images.

4. Discussion

The comparison reveals a consistent and strong advantage of transfer learning models over the baseline CNN trained from scratch. While the baseline CNN achieved reasonable performance, it suffered from heavy overfitting, slower convergence, and significantly higher computational cost. In contrast, the pre-trained models—ResNet, DenseNet, and EfficientNet—showed faster learning, superior generalization, and far more efficient use of parameters.

Performance-wise, the transfer learning models reached higher validation and test accuracies. EfficientNet-B0 emerged as the best-performing model overall, achieving the highest validation accuracy (0.9297), test accuracy (0.9271), and strongest class-wise performance, especially in challenging categories like *meningioma* where most models struggle. Although all four classes in the dataset are almost equally represented—with glioma, meningioma, pituitary, and no-tumor **samples being well-balanced and showing no class imbalance**, underperformance is likely due to the high visual similarity between meningioma and other tumor types, overlapping intensity patterns, and the presence of diverse imaging orientations (axial, coronal, sagittal), which make feature discrimination more difficult. To improve performance on this class, several strategies can be applied: targeted data augmentation (e.g., contrast enhancement, tumor-region–focused augmentation), incorporating attention-based mechanisms to help the models focus on subtle

regions. DenseNet followed EfficientNet closely with high accuracy and stable class metrics, while ResNet also delivered strong results but slightly below the other two.

Generalization was a major difference: the baseline CNN quickly memorized the training data, achieving 100% training accuracy but struggling to improve validation accuracy beyond ~0.89. Meanwhile, transfer learning models maintained smooth learning curves and stable validation performance due to the rich, pre-learned feature representations from ImageNet.

Efficiency analysis further strengthens the case for transfer learning. The baseline CNN required training all 25.7 million parameters from scratch, resulting in the longest training time and slowest inference. In comparison, the transfer-learned models trained only a few thousand parameters, drastically reducing computation time. EfficientNet delivered the fastest inference and best accuracy, making it the best choice for real-time or resource-limited environments.

Overall, the study clearly shows that **transfer learning models, specially EfficientNet outperform the baseline CNN in accuracy, stability, efficiency, and class-level consistency**, making them far more suitable for medical image classification tasks where both precision and computational efficiency are critical.

5. Conclusions

The models were trained, validated, and tested on a well-balanced dataset covering all four tumor types. This study successfully demonstrated the effectiveness of transfer learning for brain tumor classification, showing that pre-trained models, particularly EfficientNet-B0 can deliver high accuracy, robust generalization, and fast inference when applied to MRI-based tumor recognition. Follwing that, DenseNet121 shows better performance across the measures. Class wise, “pituitary” achieved highest predictability. By comparing a baseline CNN trained from scratch with ResNet50, DenseNet121, and EfficientNet-B0, the work established a clear performance advantage for transfer learning approaches, supported by strong class-wise results and efficient training behavior. Despite these achievements, several limitations point toward opportunities for improvement. Future work could explore larger and more diverse datasets to enhance feature variability, apply more advanced augmentation strategies tailored to tumor texture and contrast, and incorporate attention mechanisms or vision transformers to better capture fine-grained tumor characteristics. Additional improvements may include fine-tuning deeper layers instead of only the classification head, experimenting with hybrid CNN–ViT architectures, applying advanced regularization techniques, and leveraging ensemble methods for more stable predictions. Expanding the study to 3D MRI volumes and multimodal imaging could further elevate clinical relevance. Overall, these directions present promising avenues to refine model performance and build more reliable, clinically deployable brain tumor classification systems.

References

1. Xu, M., Guo, L. & Wu, H. C. Novel robust automatic Brain-Tumor detection and segmentation using resonance imaging. IEEE Sens. J. 24 (1 April1), 10957 10964.
2. Shah, H. A. et al. A robust approach for brain tumor detection in magnetic resonance images using finetuned EfficientNet. IEEE Access. 10, 65426–65438 (2022).
3. Alam, I. & Kaur, A. A novel approach to classify brain tumor with an effective transfer learning based deep learning model. Braz. Arch. Biol. Technol. 67, 1–10.
4. Khushi, H. M. T., Masood, T., Jaffar, A., Akram, S. & Bhatti, S. M. Performance analysis of stateoftheart CNN architectures for brain tumour detection.
5. Deepak S, Ameer PM. Brain tumor classification using deep CNN features via transfer learning. Computers in biology and medicine. 2019 Aug 1;111:103345.
6. Rehman A, Naz S, Razzak MI, Akram F, Imran M. A deep learning-based framework for automatic brain tumors classification using transfer learning. Circuits, Systems, and Signal Processing. 2020 Feb;39(2):757-75.
7. Hossain S, Chakrabarty A, Gadekallu TR, Alazab M, Piran MJ. Vision transformers, ensemble model, and transfer learning leveraging explainable AI for brain tumor detection and classification. IEEE Journal of Biomedical and Health Informatics. 2023 Apr 12;28(3):1261-72.