DEEP LEARNING-BASED CLASSIFICATION FOR BRAIN TUMOR DETECTION

Saida Arobi *ID:2104010202287,* Adnan Kamal Chowdhury *ID:2104010202297,*
and Aysha Jahan *ID:2104010202302*

Abstract—When brain cells multiply uncontrollably and they form abnormal cells we know as tumors creating serious health challenges. Brain tumors remain one of the most critical conditions affecting neurological health. Magnetic Resonance Imaging (MRI) serves as a powerful and consistent imaging technique to detect these abnormalities. However, interpreting the hundreds of images generated from a single MRI scan demands significant time and expertise from radiologists. This manual review process is not only time-consuming but also prone to human error, which can delay crucial diagnosis.

The growth of deep learning has revolutionized decision-making across numerous fields from healthcare and finance to agriculture and security.These intelligent systems are now helping analyze Magnetic Resonance Imaging (MRI) scans with remarkable speed in medical imaging specifically. Yet, when dealing with something as critical as brain tumor, speed alone isn't enough we need exceptional accuracy to ensure patients receive the right treatment and doctors have reliable support for timely decisions.

Our work introduces a convolutional neural network approach designed specifically for brain tumor classification. We trained our model using a carefully organized dataset containing Magnetic Resonance Imaging scans across four categories: three distinct tumor types(glioma, meningioma, pituitary tumor) and healthy brain images. By implementing advanced data augmentation methods, we enhanced our training material to build a more effective system. Through detailed testing using performance metrics like validation loss and confusion matrices, our models particularly those based on ResNet50 and EfficientNet architectures demonstrated strong capabilities in accurately identifying brain tumors, showing positive results in precision and recall that could genuinely support clinical practice.

1 Introduction

Over the past decade, the fields of medical imaging and neuro-oncology have undergone a profound transformation driven by advances in computer vision and machine learning technologies. Among these, deep learning, a powerful subfield of machine learning, has demonstrated exceptional ability in the analysis of large-scale medical data with unexpected precision. Its ability to automatically learn complex patterns from high-dimensional images has made it a groundbreaking tool, particularly in the detection and classification of brain tumors from MRI scans. As neurological disorders continue to pose serious diagnostic challenges, the combination of deep learning techniques offers new possibilities to improve early detection, treatment and overall patient outcomes.

Brain tumors whether benign or malignant have complex variations in shape, texture, and anatomical location that make an accurate diagnosis difficult even for experienced radiologists. Magnetic Resonance Imaging (MRI) remains the most widely used non-invasive imaging modality for visualizing brain structures and tumor boundaries. However, manual interpretation of magnetic resonance images requires expert knowledge and is often prone to human limitations such as fatigue, visual errors and inter-observer variability. These factors highlight the urgent need for automated systems that can support radiologists by providing high-precision, consistent diagnostic information.

In recent years, Convolutional Neural Networks have appeared to be a dominant approach in medical image analysis due to their hierarchical feature extraction, robustness and ability to detect subtle patterns that may not be visible to the human eye. CNN-based models have shown remarkable performance in classifying brain tumors into clinically significant categories such as glioma, meningioma, pituitary tumors and healthy brain tissues. By utilizing large MRI datasets and advanced learning techniques machine learnig models can generate predictions that assist experts in making timely and accurate decisions.

Training a deep CNN from scratch is a com-

putationally intensive endeavor that typically requires enormous,summarized datasets a resource often scarce in the medical field due to privacy concerns and the cost of expert labeling. Transfer learning has emerged as a powerful strategy to sidestep this obstruct. This technique involves taking a CNN model that has been pre-trained on a massive, general-purpose image dataset (such as ImageNet) and re-purposing its learned feature detectors for a new, specific task in this case **Brain Tumor Classification**. By leveraging the generic visual knowledge already embedded in the pre-trained model, we can achieve high performance with comparatively smaller medical datasets and significantly reduced training times. This research connects state-of-the-art pre-trained architectures, including ResNet50 and EfficientNet, capitalizing on their proven feature extraction capabilities and robust learning mechanisms.

The performance and generalization of any deep learning model are intimately linked to the volume and diversity of its training data. In medical imaging, curating a massive database of labeled patient scans is often impractical. Data augmentation provides an elegant and effective solution to this problem by artificially expanding the training dataset through a set of label-preserving geometric and photometric transformations. Techniques such as random rotation, flipping, zooming, and brightness adjustment are applied to the original MRI images, effectively simulating a wider range of clinical presentation and imaging conditions. This process acts as a regularizer, forcing the model to learn more invariant features and drastically reducing the risk of overfitting, thereby fostering the development of a more robust and reliable diagnostic model.

The primary objective of this study is to design, implement, and rigorously evaluate an automated framework for the multi-class classification of brain tumors from MRI scans using pre-trained CNN models enhanced by data augmentation. The novel contributions of this work are delineated as follows:

1) **Structured Data Pipeline:** Implementation of a systematic workflow for data ingestion, partitioning into training, validation, and test sets, and the application of extensive data augmentation techniques to ensure model robustness.

2) **Model Fine-Tuning:** Strategic adaptation and fine-tuning of powerful pre-trained CNN architectures, namely *ResNet50* and *EfficientNet*, specifically optimized for the task of distinguishing between glioma, meningioma, pituitary tumor, and non-tumor MRI scans.

3) **Complete Performance Evaluation:** A thorough experimental analysis of the proposed models using a extensive suite of matrix including classification accuracy, precision, recall, F1-score, and confusion matrices, to provide a holistic view of diagnostic performance.

2 Literature Review

These are some related work to the field of brain tumor classification.They are :

- The multiclass classification of brain tumors using MRI images by developing and evaluating a generic Convolutional Neural Network (CNN) model and six state-of-the-art with pre-trained deep learning models like ResNet50, InceptionV3, InceptionResNetV2, Xception, MobileNetV2, and EfficientNetB0. This study reveals InceptionV3 model achieved the highest accuracy of 97.12%, demonstrating its effectiveness as the best-performing model forfour class brain tumor classification (Glioma, Meningioma, Pituitary, and No-tumor). [1]

- The development of an automated brain tumor detection and classification system using Convolutional Neural Networks (CNN)is the goal of this study. It utilizes MRI images from a publicly available dataset and applies data preprocessing steps such as cropping, segmentation using the GrabCut algorithm and one-hot encoding. To improve model performance with limited data, data augmentation is performed. Transfer learning is implemented using two pre-trained deep learning architectures ResNet50 and EfficientNet to extract features and classify brain tumors into four categories: glioma, meningioma, pituitary, and no tumor. Experimental results demonstrate that the proposed CNN-based method, especially with data augmentation and pre-trained models, achieves high accuracy and improves reliability compared to manual diagnosis or

models without augmentation. [2]

- This project develops and evaluates deep learning and machine learning models for accurate brain tumor classification using MRI images. A large dataset used to train two proposed deep learning architectures: a modified 2D CNN and a convolutional auto-encoder based classifier.The extracted features were classified using six traditional machine learning methods to compare performance. Experimental results showed that the proposed deep learning models achieved high accuracy (95–96%) and near-perfect AUROC scores (0.99–1) significantly outperforming classical techniques. Statistical analysis using one-way ANOVA confirmed meaningful performance differences among the eight methods tested, demonstrating that the proposed 2D CNN and auto-encoder models offer improved accuracy and generalization for brain tumor detection. [3]

- The main work of this study involved developing a complete MRI-based brain tumor classification pipeline capable of distinguishing six tumor types with advanced image processing, segmentation, feature engineering and machine learning techniques. The process began with enhancing edges and reducing noise.The optimized features were used to train a variety of classifiers including deep learning (MLP, RNN) and machine learning models (Random Committee, Random Forest, J48, BayesNet), evaluated using strict patient-level splitting and 10-fold cross-validation to avoid data leakage and overfitting. The study improved accuracy and reduced model training time, with the Random Committee classifier achieving the highest performance of 98.61%, confirming the effectiveness of the proposed methodology for a reliable multi-class brain tumor classification. [4]

- There is limited number of no-tumor images so augmentation techniques were applied to increase sample diversity and prevent bias during training. Existing literature highlights inherent limitations of machine learning-based tumor detection, noting that model performance depends heavily on data quality and quantity and may vary when exposed to unfamiliar image types. Since ML models do not account for the underlying physics of MRI imaging, preprocessing steps such as skull stripping, noise reduction, and bias correction are essential for improving reliability. Comparative analyses show that EfficientNetB0 achieves the highest accuracy (97.61%) among several tested architectures, outperforming models like ResNet50, Xception, MobileNetV2, and VGG16. Although strong classification performance is reported across multiple models, literature emphasizes the need for improved preprocessing, better handling of small datasets. [5]

3 Dataset Description

This research utilizes a strong neuroimaging repository specifically designed for brain tumor classification via deep learning. The Brain tumor Dataset, in its entirety has 7,023 high-quality magnetic resonance imaging (MRI) scans which is systematically sorted into four clinically meaningful categories: glioma, meningioma, pituitary tumor and no tumor.They represents healthy brain anatomy and serves as a critical control group.

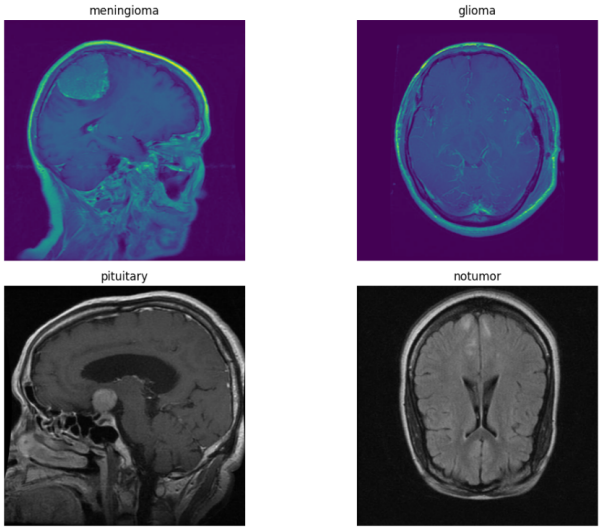


Fig. 1: Sample Data of four classes

Every image in the collection has undergone meticulous examination and labeling by experienced neuroradiologists, guaranteeing a high level of diagnostic reliability in the dataset. The expert-validated annotations significantly enhance the dataset's credibility making it exceptionally suitable

for training sophisticated deep learning models that require accurate ground truth.

The dataset is thoughtfully divided into separate training and testing directories allowing for systematic model development and dependable performance evaluation for training and testing directories. The training portion contains the majority of images, enabling deep learning models to effectively learn distinguishing features across all four brain tumor categories. Meanwhile, the independent test set serves as an essential benchmark for assessing the model's ability to generalize to new and unseen MRI scans.

The training directory contains large volume of 5,712 documented images, providing a rich foundation for the model to learn the special radiological features and complex patterns associated with each category. The distribution within the training set is as follows:

- Glioma: 1,321 images
- Meningioma: 1,339 images
- Pituitary Tumor: 1,457 images
- No Tumor: 1,595 images

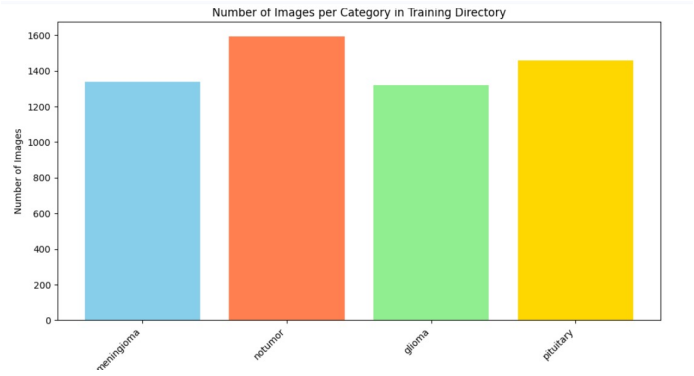


Fig. 2: Barchart of Training Data

The testing directory has 1,311 images, which are completely deducted during the training phase. This subset acts as an unbiased benchmark to exactly evaluate the model's ability to generalize its knowledge to unseen clinical data. The data distribution of the testing set is as follows:

- Glioma: 300 images
- Meningioma: 306 images
- Pituitary Tumor: 300 images
- No Tumor: 405 images

The key strength of this collection is its clinically relevant organization into four distinct categories that represent both common pathological conditions and normal neurological anatomy. This multi-class structure is essential for training algorithms

Fig. 3: Barchart of Testing Data

to perform precise differential diagnoses. The categories are defined like this:

- **Glioma:** It includes both low-grade and high-grade gliomas, capturing their hallmark infiltrative nature and mass effect an reflects the heterogeneity of glial cell–derived tumors, with variation in contrast enhancement, shape, and anatomical involvement.

- **Meningioma:** Represents extra-axial neoplasms with characteristic dural attachments. Contains images that demonstrate homogeneous contrast enhancement and, in some cases, calcification, across diverse anatomical locations such as the convexity and skull base.

- **Pituitary Tumor:** Presents both micro adenomas (confined to the sella) and macroadenomas (with suprasellar extension). Includes typical radiological features such as mass effect on the optic chiasm or invasion of the cavernous sinus.

- **No Tumor (Normal Brain):** Contains non-pathological MRI images that reflect normal brain anatomy. Serves as a negative-control group, allowing algorithms to learn the difference between healthy brain structure and pathological lesions.

This dataset provides a powerful foundation for developing multi-class brain tumor classification systems. It also supports broader research into feature extraction, transfer learning, and model interpretability. Beyond academic exploration, the dataset holds substantial potential for aiding the

creation of clinical decision-support tools that can assist radiologists in diagnosis, case prioritization, screening workflows, and generating reliable second opinions.

3.1 Dataset Preprocessing

1) **File Store:**
The dataset repository is stored in Google drive.To acquire the dataset Colab environment mounted on Google Drive and set the base directory for the brain tumor dataset stored in Google Drive. Then constructed file paths for the training and testing folders and verified that the dataset directory exists.

```
BASE_DIR = '/content/drive/MyDrive/MLL/Brain Tumor data/Brain Tumor data'
TRAIN_DIR = os.path.join(BASE_DIR, 'Training')
TEST_DIR  = os.path.join(BASE_DIR, 'Testing')

if not os.path.exists(BASE_DIR):
    raise FileNotFoundError("Base directory not found!")

print("Train classes:", os.listdir(TRAIN_DIR))
print("Test  classes:", os.listdir(TEST_DIR))
```

```
Train classes: ['meningioma', 'glioma', 'pituitary', 'notumor']
Test  classes: ['pituitary', 'notumor', 'glioma', 'meningioma']
```

Fig. 4: File path in Drive

The output of code confirms that the dataset is properly organized into labeled class folders (such as glioma, meningioma, pituitary, and no-tumor).

2) **Global Parameter Set:**
Three key configuration parameters used for model training. The Image size = (224, 224) sets the resolution to all input images in a consistent input shape. The Batch size = 32 specifies that the model will process 32 images at a time during each training step, balancing memory usage and training speed. Finally, seed = 42 sets a fixed random seed to make the data splitting, shuffling, and other random operations.

```
IMAGE_SIZE = (224, 224)
BATCH_SIZE = 32
SEED = 42
```

3) **Data Augmentation:**
The training image generator is used to prepare the images in a way that increases variety, helping the model learn more effectively and reduce the chances of memorizing the dataset. It applies a custom image processing function and introduces changes such as rotating pictures, moving them slightly in different directions, enlarging or shrinking them, changing their angles, flipping them sideways, and making them a little brighter or darker.

```
train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_func,
    rotation_range=25,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.20,
    shear_range=0.15,
    horizontal_flip=True,
    brightness_range=[0.8, 1.2],
    validation_split=0.2
)

val_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_func,
    validation_split=0.2
)
test_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_func
)
```

A small portion of the dataset (20%) is kept aside to check how well the model performs during development. The generator for validation images uses the same processing steps but does not make any visual changes, ensuring that evaluation results remain realistic and unbiased. In the same way, the image generator for testing only processes the images without any modifications, allowing the final performance check to be done on clean and untouched data.
The Augmented images of MRI Scans are shown below

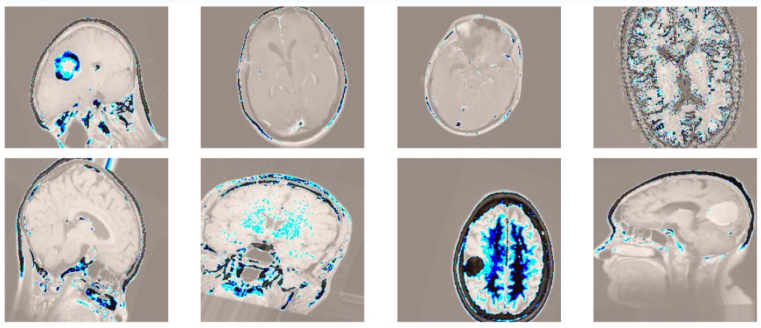


Fig. 5: Augmentated Sample Images

4) **Label Encoding:**
The categorical class labels were converted into numerical representations using label encoding. This step is essential because deep learning models can only process numerical inputs.

```
print("Class Indices (Label Encoding):")
for class_name, label in train_generator.class_indices.items():
    print(f"  '{class_name}': {label}")

class_names_encoded = list(train_generator.class_indices.keys())
print(f"\nOrdered Class Names: {class_names_encoded}")
```

Each brain tumor category was assigned a unique integer value, forming the class indices: glioma was encoded as 0, meningioma as 1, notumor as 2, and pituitary as 3. These encoded values served as the target labels during model training and evaluation. Additionally, an ordered list of class names ['glioma', 'meningioma', 'notumor', 'pituitary'] was maintained to ensure a consistent mapping between numerical predictions and their corresponding clinical categories.After encoding the classes the result shows:

```
Class Indices (Label Encoding):
  'glioma': 0
  'meningioma': 1
  'notumor': 2
  'pituitary': 3

Ordered Class Names: ['glioma', 'meningioma', 'notumor', 'pituitary']
```

Fig. 6: Label Encoding

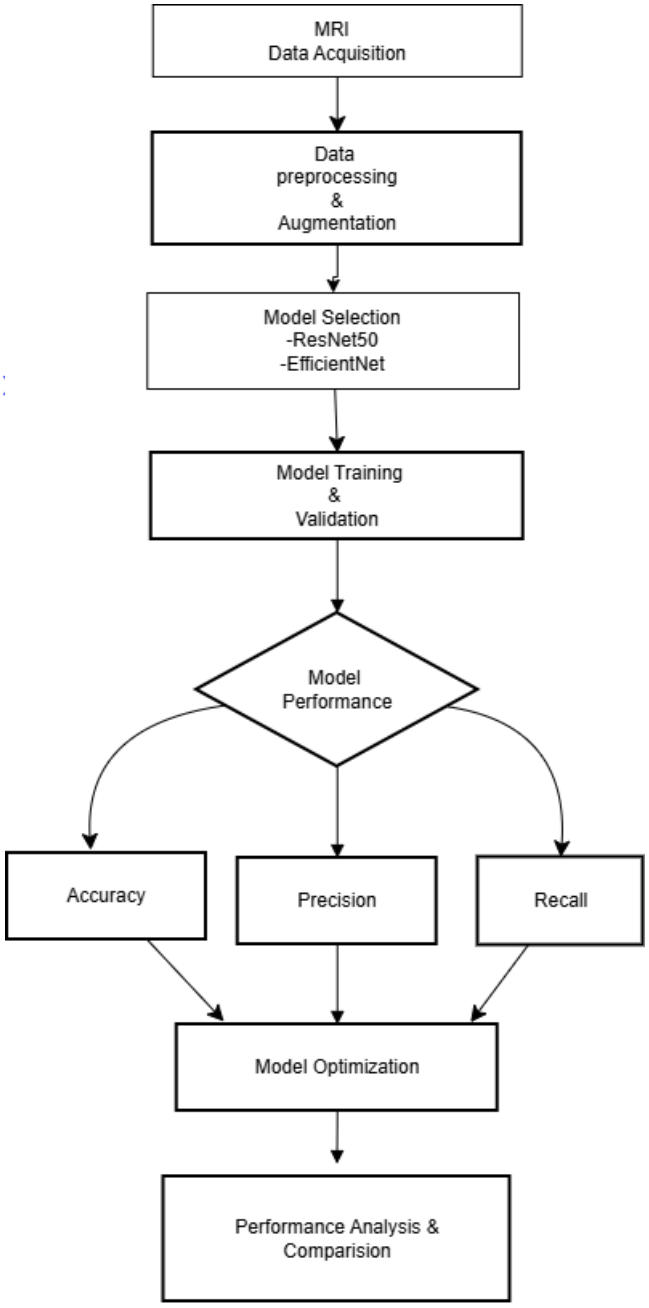


Fig. 7: Flowchart of the Proposed System

4 METHODOLOGY

The proposed system for this project is structured around a systematic, multi-stage pipeline designed to ensure strong predictive performance for brain tumor multiclass classification. The entire workflow, from data acquisition to final model evaluation, is visualized in the flowchart below

To handle the computational demands of deep learning–based medical image analysis all experiments were carried out on Google Colab, a cloud-based platform that provides high-performance computing resources for Python development. The implementation was performed using the Python programming language, with TensorFlow serving as the primary deep learning library for constructing, training and validating the neural network models.The high-end GPU significantly reduced model training time and made it easier to train computationally intensive architectures which would otherwise require long training cycles on standard hardware.

The foundation of this research lies in the acquisition of a large and diverse dataset of brain MRI scans. The dataset was obtained from a publicly accessible repository Brain Tumor Data and is a combination of two separate directories to provide a broader representation of tumor characteristics. In total, the dataset includes 7,023 high-resolution MRI

images, each categorized into one of four clinically significant classes: Glioma, Meningioma, Pituitary Tumor, and No Tumor. This multi-class structure is essential for differentiating between tumor types and identifying healthy brain tissue, making it a critical component in the early stages of automated disease classification.

Raw medical images require a lot more preprocessing to be suitable for deep learning models and to increase the model's ability to learn relevant features and predict.

- **Resizing and Normalization:** All MRI images were resized to a uniform resolution based on the input requirements of the pretrained models.Pixel intensity values were normalized to the range [0, 1] to stabilize gradient updates and accelerate training by maintaining consistent numerical values across the dataset.

- **Label Encoding:** Categorical text labels (glioma, meningioma, pituitary, no tumor) were converted into numerical form using label encoding.This enabled the neural network to correctly interpret each class during supervised learning.

- **Data Augmentation:** Data augmentation was applied to increase dataset diversity and reduce overfitting, as medical image datasets are typically small.

These transformations help the models learn more generalized patterns, making them more applicable to real-world clinical imaging scenarios. The models were chosen for their strong prediction and feature detection ability.

ResNet50 model was selected for its deep architecture and its use of residual connections, which overcome the vanishing gradient problem and enable effective training of networks with many layers. Residual learning allows deeper architectures to learn more abstract and powerful features. EfficientNet was chosen for its ability to scale network depth, width, and input resolution in a balanced manner using compound scaling. This results in exceptional performance while keeping computational costs low, making it highly suitable for medical image classification tasks.

The training pipeline was developed to ensure best learning while preventing overfitting or underfitting.

The augmented training data used to train the models, while a validation split allowed continuous monitoring of model performance after each epoch. The validation results provided insights into the model's learning behavior and help in tuning hyperparameters before final testing. Each model was compiled using Categorical Cross-Entropy as the loss function, suitable for multi-class problems. The Adam optimizer was employed due to its adaptive learning rate mechanism, which makes it effective for complex optimization landscapes.

Multiple evaluation metrics were used to evaluate model performance thoroughly.This was chosen to provide a better understanding of classification performance across different tumor types.

- **Accuracy:** Measures the overall correctness of predictions.
- **Precision:** Indicates the proportion of correctly identified positive cases among all predicted positives, reflecting model reliability.
- **Recall (Sensitivity):** Represents the proportion of actual positive cases detected by the model, which is critical in medical diagnosis to avoid missing tumors.

This analysis helped detect potential issues such as overfitting or underfitting and contributed to the final determination of which architecture performed best for the task.

4.1 Model Description

1) **Residual Network(ResNet50):** ResNet50, short for Residual Network with 50 layers, is one of the most influential and widely used deep learning architectures for image classification tasks. Developed by Microsoft Research, the model introduced a breakthrough concept known as residual learning, which allows very deep neural networks to be trained effectively without suffering from the common problems of vanishing gradients or performance degradation. Because of its powerful feature extraction abilities and stable training behavior, ResNet50 has become a preferred choice in many medical imaging applications, including brain tumor classification from MRI scans.
   Traditional deep neural networks often struggle when increasing depth beyond a certain point. Adding more layers does not always improve performance; in fact, very

deep networks tend to learn poorly due to the difficulty in propagating gradients during backpropagation. ResNet50 solves this problem through its signature architectural component: the residual block. A residual block includes shortcut or skip connections, where the input of a layer is added directly to the output of a deeper layer. This simple yet powerful design ensures that even if deeper layers fail to learn useful patterns, the network can still retain important information from earlier layers. As a result, ResNet50 can train extremely deep models successfully while maintaining high accuracy.

The architecture of ResNet50 consists of 50 layers, including convolutional layers, batch normalization, activation functions (ReLU), and fully connected layers. However, the defining feature is the presence of multiple stacked residual blocks, each containing convolutional operations and shortcut pathways. These blocks allow the model to learn complex hierarchical features—from basic edges and textures in the early layers to advanced structural and anatomical patterns in deeper layers. This layered feature extraction is essential for MRI analysis, where brain tumors often exhibit subtle differences in shape, contrast, and spatial distribution.

For brain tumor classification, ResNet50 takes MRI images typically resized to a standardized input dimension as input. The initial layers process low-level pixel information, identifying primitive patterns such as edges and intensity gradients. As the image passes deeper into the network, the residual blocks help extract higher-level features that correspond to tumor-specific characteristics, such as irregular boundaries, mass effect, heterogeneous texture, or contrast enhancement seen in gliomas, meningiomas, and pituitary tumors. The skip connections ensure that important spatial features are preserved and combined with newly learned patterns, resulting in robust and detailed feature maps.

One major advantage of using ResNet50 in medical imaging is that the model is pretrained on the ImageNet dataset, which contains millions of natural images. This pretraining helps the network learn general visual features that can be transferred to smaller medical datasets through transfer learning. Since medical MRI datasets are often limited in size, transfer learning significantly improves performance and speeds up the training process. Researchers typically replace the final fully connected layer of ResNet50 with a customized classification layer that outputs predictions for specific tumor categories such as glioma, meningioma, pituitary tumor, and no-tumor.

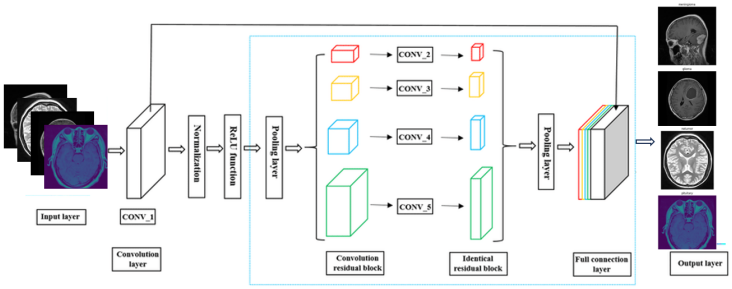


Fig. 8: ResNet50 Architecture

Another strength of ResNet50 is its strong generalization ability, meaning it performs well on unseen MRI images even when there are variations in lighting, noise, or imaging conditions. This is especially important in real-world medical scenarios, where MRI scanners, acquisition protocols, and patient anatomy can vary widely. The residual architecture helps prevent overfitting and ensures that the network learns meaningful tumor patterns rather than memorizing training images.

In brain tumor classification projects, ResNet50 is frequently used for tasks such as multi-class classification, feature extraction, and even segmentation when combined with other models. It has been proven effective in identifying small lesions, distinguishing normal brain structures from tumor tissue, and differentiating among tumor types with high accuracy. These abilities make ResNet50 a valuable tool for computer-aided diagnosis (CAD) systems that aim to assist radiologists in making faster and more accurate decisions.

2) **EfficientNet:** EfficientNet is a family of deep learning models specially designed to achieve high accuracy with fewer parameters, making it one of the most efficient and powerful neural network architectures available today. Unlike traditional CNN models that are scaled up by simply adding more layers or increasing width without a clear strategy, EfficientNet introduces a balanced and systematic scaling method that improves performance while maintaining computational efficiency. This makes it particularly suitable for medical imaging tasks such as brain tumor classification, where both accuracy and computational cost are important considerations.

Fig. 9: EfficientNetB0 Architecture

At its core, EfficientNet is built on a highly optimized base architecture known as MBConv blocks (Mobile Inverted Bottleneck Convolution). These blocks help the network learn complex visual patterns in MRI images while keeping the model lightweight and fast. The key idea behind MBConv is to first reduce the number of channels (compression), perform convolutions efficiently, and then expand the channels again to capture more detailed features. This inverted bottleneck structure allows the network to extract rich feature representations from MRI scans without requiring heavy computation.

One of the unique strengths of EfficientNet is its compound scaling method. Traditional models may scale depth (number of layers), width (number of channels), or resolution (input image size) independently, which often leads to inefficient architectures. EfficientNet solves this by scaling all three dimensions together, using a set of carefully designed scaling coefficients. This balanced approach helps the model learn more detailed features when working with higher-resolution MRI images while preventing overfitting or excessive computational load. For brain tumor classification, this means the model can better identify subtle characteristics in tumors such as texture, edges, intensity variations, and shape without requiring extremely large or slow models. EfficientNet also incorporates squeeze-and-

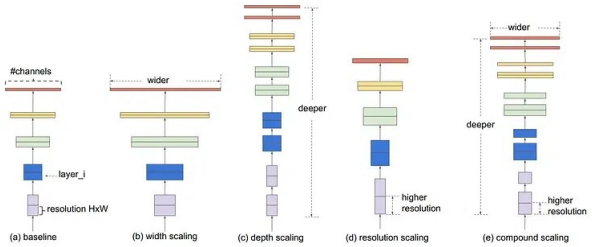


Fig. 10: EfficientNetB0 Pipeline

excitation blocks, which allow the model to focus on the most important features in an MRI image by re-weighting channel importance dynamically. In the context of brain tumor classification, this helps the network highlight critical tumor regions while suppressing irrelevant background information. This ability to emphasize meaningful patterns enhances the model's capacity to differentiate between glioma, meningioma, pituitary tumors, and normal brain images.

When used for brain tumor classification, EfficientNet typically takes MRI scans as input—often resized to a standardized resolution depending on the chosen EfficientNet variant (e.g., EfficientNet-B0, B1, B3, etc.). The model then processes the image through a series of MBConv layers, SE blocks, and convolutional operations, gradually converting low-level visual features (such as edges and textures) into high-level tumor-specific representations. Toward the end of the network, these features are passed through fully connected layers or a global average pooling layer, producing a final classification output that assigns the MRI scan to one of the predefined categories.

A major advantage of EfficientNet in medical imaging is that it offers strong accuracy even when trained on relatively small datasets, thanks to its efficient feature extraction mechanism. This is crucial for brain tumor datasets, where obtaining large amounts of annotated MRI images can be challenging. EfficientNet also tends to generalize well, meaning it performs reliably on

unseen MRI scans from different patients or imaging conditions.

4.2 Fine Tuning

- **ResNet50**
  Freezing Layer:
  The model was trained using augmented images from the training generator and evaluated after each epoch using clean validation data to measure generalization. Although the maximum training limit was set to 15 epochs, callback mechanisms automatically controlled the process prevente overfitting. This setup ensured efficient learning preserved model weights and captured all accuracy and loss values in a history log for later performance analysis.

```
history = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=15,
    callbacks=[checkpoint, early_stop]
)
```

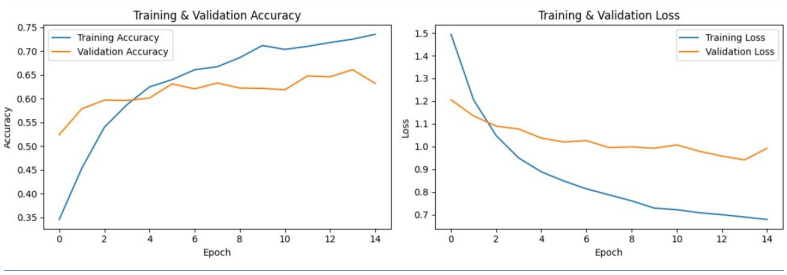


Fig. 11: Training Curve

The visualizations clearly show the model's progression and stabilization over time. The accuracy plot demonstrates steady improvement with training accuracy rising from around 0.34 to 0.74 and validation accuracy increasing before leveling off near 0.63–0.66, indicating mild overfitting due to the frozen pretrained base. The loss plot supports this trend as training loss decreases from about 1.5 to 0.67 while validation loss declines more gradually and stabilizes near 1.0. The widening gap between training and validation curves confirms moderate overfitting, which was effectively controlled through EarlyStopping and Dropout.

Unfreezing Layer:

1. After the initial training phase with the ResNet50 backbone frozen, a fine-tuning stage was conducted to enhance the model's ability to distinguish between brain tumor classes. In this phase, only the final 50 layers of the ResNet50 backbone were unfrozen to target high-level features, allowing the model to adapt its pre-learned representations to the subtle characteristics of MRI images while preserving the stability of the earlier layers. To ensure careful weight updates, the model was recompiled with a much lower learning rate (1×10), allowing gradual and precise adjustments without overwriting previously learned features. Fine-tuning was performed over 10 additional epochs, with augmented training data used to improve generalization and an untouched validation set providing an unbiased measure of performance.

```
for layer in base_model.layers[-50:]:
    layer.trainable = True
model.compile(
    optimizer=Adam(learning_rate=1e-5),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
history_finetune = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=15,
    callbacks=[checkpoint, early_stop]
)
```

The fine-tuning training runs for up to 15 epochs - giving the model more time to refine its understanding compared to a shorter training period. This extended duration allows the newly unfrozen layers to make subtle, effective adjustments

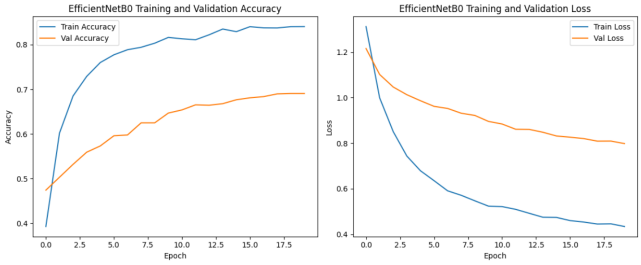
- **EfficientNetB0**
  Freezing Layer:
  1. The model was initially used as a fixed feature extractor by loading the pretrained ImageNet weights and freezing the convolutional layers.

```
monitor_metric_eff = 'val_accuracy'

checkpoint_eff = ModelCheckpoint(
    '/content/best_efficientnetb0_brain_tumor.h5',
    monitor=monitor_metric_eff,
    save_best_only=True,
    mode='max',
    verbose=1
)

early_stop_eff = EarlyStopping(
    monitor=monitor_metric_eff,
    patience=5,
    restore_best_weights=True,
    mode='max',
    verbose=1
)
history_efficientnet = model_efficientnet.fit(
    train_generator_eff,
    validation_data=val_generator_eff,
    epochs=20,
    callbacks=[checkpoint_eff, early_stop_eff]
)
```

This means that during the initial training phase, only the newly added classification head consisting of a Global Average Pooling layer, a Dropout layer, and a Dense softmax output layer was trainable while the base model's parameters remained unchanged.
The training results show that EfficientNetB0 achieved a final training accuracy of 84.05%, indicating that the model learned the patterns within the training samples reasonably well. However, the validation accuracy was significantly lower at 69.06%, suggesting generalization gap. This gap implies that the model may be overfitting, as it performs much better on the training data than on unseen validation data.

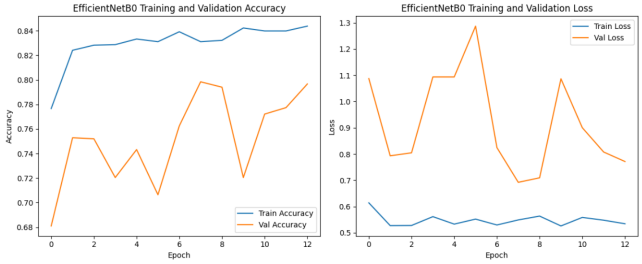



```
EfficientNetB0 Final Training Accuracy:   0.8405
EfficientNetB0 Final Validation Accuracy: 0.6906
EfficientNetB0 Final Training Loss:       0.4341
EfficientNetB0 Final Validation Loss:     0.7982
```

2. After changing the dropout layer and learning rate there is no significant change in the curve and the result accuracy and loss.

```
x = base_model_efficientnet.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.5)(x)
outputs = Dense(NUM_CLASSES, activation='softmax')(x)
```

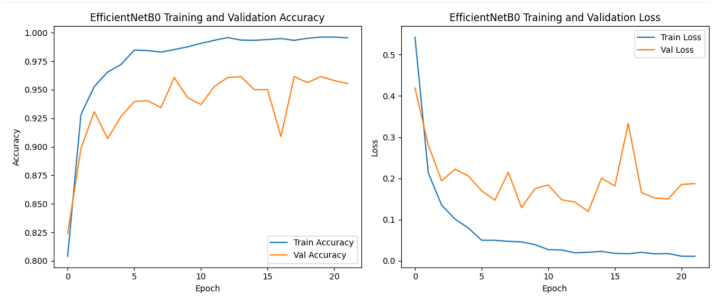
```
model_efficientnet.compile(
    optimizer=Adam(learning_rate=1e-2),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```



The result curve shows a significant underfitting due to model's inability to learn features.

Unfreezing Layers:
1.The last 30 layers of the EfficientNetB0 model was selectively unfreezed so they can be trained without overfitting in the code.

```
for layer in base_model_efficientnet.layers[-30:]:
    layer.trainable = True
```



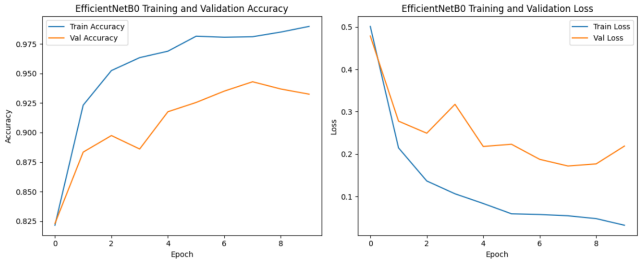
The unfreezing strategy was successful the model learned

effectively without significant overfitting, demonstrating that fine-tuning the entire architecture significantly improved performance for brain tumor classification but the result shows a lots of fluctuations throughout the training process which means model is not learning properly.

2.Training now proceeds for 10 epochs with early stopping patience reduced to 5.

```
early_stop_eff = EarlyStopping(
    monitor='val_accuracy',
    patience=5,
    restore_best_weights=True,
    mode='max',
    verbose=1
)

history_finetune = model_efficientnet.fit(
    train_generator_eff,
    validation_data=val_generator_eff,
    epochs=10,
    callbacks=[checkpoint_eff, early_stop_eff]
)
```



The curve is significantly smooth from before which means the feature extraction is also smooth.

5 Training Procedure

- **ResNet50:**

Model Build:
This study adopted a transfer learning approach using the ResNet50 architecture as the foundational network to develop an effective image classification model. Instead of training a deep neural network from the ground up the pre-trained ResNet50 model—trained with reliable feature extractor. The model was initialized by loading ResNet50 without its original top classification layer. During the initial training phase, all layers of the ResNet50 base were frozen meaning their weights remained unchanged. This preserved the rich visual features learned from ImageNet while reducing computational cost by training only the newly added layers. Classification heads were built by applying Global Average Pooling to reduce dimensionality ,a dropout layer with a 0.5 rate to minimize overfitting and finally a dense softmax layer to classify images into the target classes. The combined model was compiled using appropriate hyperparameter and a model summary was generated to verify the architecture and ensure correct configuration.

```
base_model = ResNet50(
    weights='imagenet',
    include_top=False,
    input_shape=(IMAGE_SIZE[0], IMAGE_SIZE[1], 3)
)

base_model.trainable = False

x = GlobalAveragePooling2D()(base_model.output)
x = Dropout(0.5)(x)
output = Dense(NUM_CLASSES, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=output)

model.compile(
    optimizer=Adam(learning_rate=1e-4),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

model.summary()
```

The other variable that was considered for training procedure was

1. **Loss Function:**
The model was trained using categorical cross-entropy as the loss function, which is standard for multi-class classification problems. This function measures the difference between the predicted class probabilities and the true class labels.

2. **Optimizer:**
The Adam optimizer was selected due to its efficiency, adaptive learning capabilities, and suitability for deep learning tasks. A low learning rate was used to make precise updates to the network weights, especially important when fine-tuning a pre-trained

model.

3. **Environment and Hardware:**
The model was built and trained within a controlled computational environment in Google Colab that provided GPU acceleration. This environment ensures consistent execution of the code managed dependencies such as TensorFlow and Keras, and offers sufficient computational power for handling large image datasets and deep learning models.

4. **Seed:**
A fixed random seed (Seed=42) was used to ensure the reproducibility of the results.The randomness involved in operations such as data shuffling, weight initialization and augmentation by controlling.The seed ensures that each run of the model produces consistent and comparable outcomes. This is essential for scientific reliability and repeatability of experiments.

- **EfficientNetB0:**

Model Bulid:
Building an EfficientNetB0-based deep learning model for brain tumor classification. First, the EfficientNetB0 architecture is loaded with pretrained ImageNet weights, excluding its top classification layers, and its input size is set to the defined image dimensions. The base model is frozen (trainable = False) so its pretrained weights are not updated during initial training. A custom classification head is then added by applying global average pooling to reduce feature dimensions, followed by a dropout layer with a 0.4 rate to reduce overfitting. A final dense layer with softmax activation is used to classify the images into the specified number of classes.Models architecture is displayed using summary().
There are other variable taken into consideration which are -
1. **Loss Function:** Model was trained using categorical cross-entropy as a loss function. This loss is standard for multi-class classification tasks and is suitable when the output layer uses a softmax activation function.

```
base_model_efficientnet = EfficientNetB0(
    weights='imagenet',
    include_top=False,
    input_shape=(IMAGE_SIZE[0], IMAGE_SIZE[1], 3)
)
base_model_efficientnet.trainable = False
x = base_model_efficientnet.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.4)(x)
outputs = Dense(NUM_CLASSES, activation='softmax')(x)
model_efficientnet = Model(inputs=base_model_efficientnet.input, outputs=outputs)
model_efficientnet.compile(
    optimizer=Adam(learning_rate=1e-4),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
model_efficientnet.summary()
```

2. **Optimizer:** Training was optimized using the Adam optimizer with a learning rate that differ in various training. Adam was chosen due to its adaptive learning rate capabilities, which help stabilize training and improve speed especially for deep architectures such as EfficientNetB0.

3. **Environment and Hardware:** Training was conducted in a GPU-accelerated environment (Google Colab), which supports efficient execution of deep learning workloads. There was use of TensorFlow/Keras within a Python environment optimized with GPU usage. The GPU support accelerates computationally intensive tasks like forward propagation, backward propagation and updating millions of model parameters.

4. **Seed:**
Fixed random seed was used to ensure reproducibility of the results. By controlling the randomness involved in operations such as data shuffling, weight initialization, and augmentation, the seed ensures that each run of the model produces consistent and comparable outcomes. This is essential for scientific reliability and repeatability of experiments.

6 Result

1) **ResNet50:**
The accuracy curve shows a steady improvement in both training and validation performance after unfreezing the top layers of the model. Training accuracy increased smoothly from 0.92 to 0.95, indicating that

the model was able to learn deeper and more relevant feature representations from the images. Similarly, validation accuracy rose from 0.82 to 0.88, showing that the model was also improving on unseen data. Although the gap between the two curves suggests mild overfitting, the consistent upward trend in validation accuracy indicates that the model maintained good generalization ability throughout training.

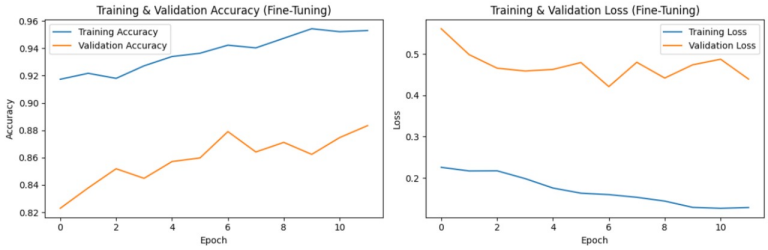


Fig. 12: Training Accuracy of ResNet50

The loss curve further supports these observations by showing a continuous decrease in training loss from 0.23 to 0.14 which confirms that the model's predictions were becoming more confident and accurate. The validation loss, while more variable and fluctuating between 0.42 and 0.50, still shows an overall downward movement. The higher and more unstable validation loss compared to the training loss reflects mild overfitting, a common effect when fine-tuning deep models. Nonetheless, the general decreasing trend indicates that the model was still improving its performance on unseen data despite the fluctuations.

```
Test Loss: 0.3287
Test Accuracy: 0.8986
```

Fig. 13: Test Accuracy of ResNet50

The trained model was evaluated on the test dataset using the model.evaluate() function. The evaluation measured both the loss and accuracy to assess the model's performance on unseen data. The results indicate a test loss of 0.3287 and a test accuracy of 0.8986. This means that the model correctly classified approximately 89.86% of the test samples, demonstrating strong generalization from the training data. The relatively low loss further confirms that the model predictions are close to the actual labels, indicating effective learning.

Classification Report:
The model achieved an overall accuracy of 90%, indicating a strong capability to correctly classify MRI scans across the four categories.

	precision	recall	f1-score	support
glioma	0.94	0.78	0.85	300
meningioma	0.81	0.82	0.81	306
notumor	0.93	0.99	0.96	405
pituitary	0.90	0.97	0.94	300
accuracy			0.90	1311
macro avg	0.90	0.89	0.89	1311
weighted avg	0.90	0.90	0.90	1311

Fig. 14: ResNet50 Classification Report

However, performance varied across specific tumor types:

- High Performance ('Notumor' & 'Pituitary'): The model demonstrated exceptional reliability in identifying the notumor and pituitary classes. Precision and recall scores for these categories approached or exceeded 0.95. This suggests that the distinct visual features of pituitary tumors and healthy brains are easily successfully captured by the model, resulting in very few false alarms or missed diagnoses.

- Challenges with Glioma: The glioma class proved to be the most difficult for the model to characterize, with a recall score of 0.78. A lower recall indicates that while the model is generally accurate when it does identify a glioma, it misses a noticeable portion of actual glioma cases, frequently misclassifying them as other tumor types.

Confusion Matrix:
The Confusion Matrix (visualized as a heatmap) provides a granular view of these predictions. The strong concentration of values along the diagonal confirms that the vast majority of predictions were correct.
However, the off-diagonal errors reveal specific patterns of confusion:

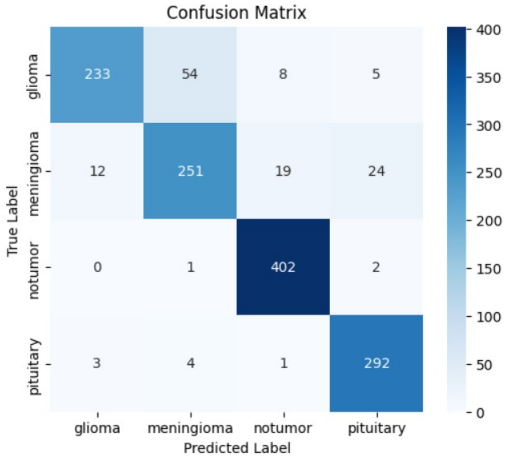


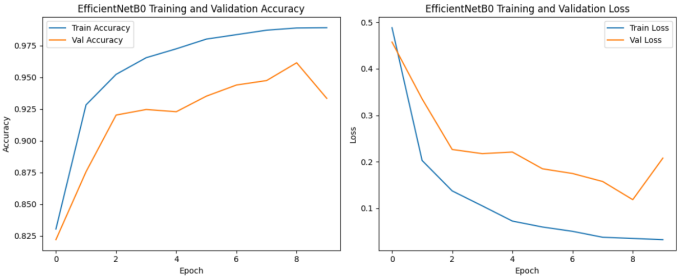
Fig. 15: ResNet50 Confusion Matrix

• Tumor Overlap: The most significant error occurred between glioma and meningioma. Specifically, 54 glioma cases were incorrectly classified as meningioma. This pattern suggests that these two tumor types likely share similar structural or textural features in MRI scans, making it challenging for the model to distinguish between them without further training or higher-resolution data.

• Robustness of 'Notumor': In contrast, the notumor class was almost perfectly isolated. With only 3 total misclassifications, the model proved highly effective at distinguishing between a healthy brain and one containing a tumor.

2) **EfficientNetB0:**
The final result after increasing the dropout rate to 0.3, the model showed significant improvement.The curve was significantly smoother than before. The training accuracy rose to 98.91% with a low training loss of 0.0323, while the validation accuracy improved to 93.34% with a validation loss of 0.2077. This indicates that the model is now less overfitted and performs more reliably on unseen MRI images. The higher dropout helped the network regularize better and learn more unseen features.

```
x = base_model_efficientnet.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.3)(x)
outputs = Dense(NUM_CLASSES, activation='softmax')(x)
```




```
EfficientNetB0 Final Training Accuracy:   0.9891
EfficientNetB0 Final Validation Accuracy: 0.9334
EfficientNetB0 Final Training Loss:       0.0323
EfficientNetB0 Final Validation Loss:     0.2077
```

Classification Report:
The EfficientNetB0 model achieved excellent performance across all brain tumor classes.

EfficientNetB0 Classification Report:

	precision	recall	f1-score	support
glioma	0.98	0.96	0.97	300
meningioma	0.97	0.96	0.97	306
notumor	0.98	1.00	0.99	405
pituitary	0.98	1.00	0.99	300
accuracy			0.98	1311
macro avg	0.98	0.98	0.98	1311
weighted avg	0.98	0.98	0.98	1311

Fig. 16: EfficientNetB0 Classification Report

Precision measures how many predicted positive instances were actually correct. Values range from 0.97 to 0.98, indicating very few false positives. Recall measures how many actual positives were correctly identified. Values range from 0.96 to 1.00, showing the model effectively detected almost all true cases. F1-score is the harmonic mean of precision and recall, with values 0.97–0.99, reflecting balanced and reliable predictions. Accuracy across all 1,311 test images is 98%, highlighting strong overall performance.

Confusion Matrix:
The model performs for a strong classification across all four brain tumor categories, with particularly high accuracy in identifying pituitary and notumor cases.
The model correctly classified 289 glioma

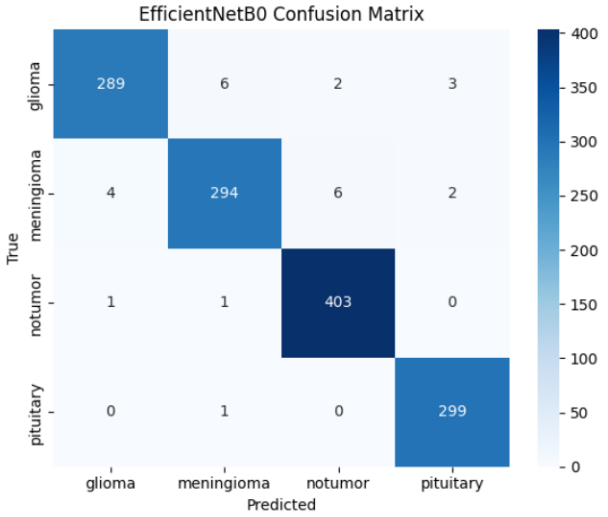


Fig. 17: EfficientNetB0 Confusion Matrix

cases, though it misclassified 6 as meningioma and 3 as notumor. For meningioma, 294 cases were accurately identified with minor confusion with glioma and pituitary classes. The pituitary class achieved near-perfect detection with 403 correct predictions and only 2 misclassifications as meningioma, while the notumor class showed excellent specificity with 299 correct identifications and only one false positive as meningioma.

overall, we can say that,the performance comparison between the two models shows a clear advantage for EfficientNet over ResNet. While ResNet achieved an accuracy of 89.86% with precision and recall around 0.90, EfficientNet significantly outperformed it, reaching 98.02% accuracy and both precision and recall at 0.98. This indicates that EfficientNet not only correctly classified a higher proportion of images but also did so consistently across all classes, making it more reliable and effective for brain tumor classification. The results suggest that EfficientNet's architecture and feature extraction capabilities are better suited for capturing the subtle patterns in MRI images compared to ResNet.

Model	Accuracy	Precision	Recall
ResNet	89.86%	0.90	0.90
EfficientNet	98.02%	0.98	0.98

TABLE 1: Comparison of ResNet and EfficientNet

7 Discussion

The experimental findings of this study reveal a distinct performance difference between the two transfer learning models used for brain tumor classification: ResNet50 and EfficientNetB0. In the initial training phase both models were trained with frozen convolutional layers to utilize their pree-trained feature extractors. However, nether model achieved high accuracy at this stage, indicating that frozen weights alone were not sufficient for capturing the complex patterns present in MRI scans. ResNet50, in particular, showed limited improvement even after further training. Its architectural depth and reliance on residual connections, while beneficial for natural image datasets may not have align well with the unique structural characteristics of brain MRI images. Additionally, ResNet50 often requires large and diverse datasets for optimal adaptation, and the variations in MRI intensity levels may have contribute to inconsistent feature extraction and reduced predictive precision.

In contrast, EfficientNetB0 demonstrated a substantial boost in performance, especially after fine-tuning and selectively unfreezing deeper layers. Its compound scaling strategy effectively balances depth, width, and resolution, enabling the model to extract more discriminative and task-specific features from the MRI scans. After unfreezing layers and performing full fine-tuning, EfficientNetB0 exhibited stable learning behavior, improved generalization capability, and high accuracy across nearly all tumor categories. This indicate that EfficientNetB0 is more capable of identifying subtle intensity variations and localized structural differences essential for precise medical image classification.

The comparison based on evaluation metrics highlights the performance gap between the two models. ResNet50 achieved an accuracy of 89.86% with precision and recall both at 0.90, reflected moderate performance with noticeable limitations in correctly identifying each tumor class. EfficientNetB0, on the other hand, achieved an accuracy of 98.02% with precision and recall both reaching 0.98, demonstrating far superior reliability in detecting tumor types and minimizing false predictions. Such strong performance confirms EfficientNetB0's suitable for medical imaging tasks.

Throughout the study from relevant research papers and prior works significantly influenced model selection, preprocessing strategies and evaluation approaches. Ethical considerations were also maintained, as the dataset involved publicly avail-

able MRI scans, ensuring patient confidentiality and adherence to responsible AI research practices. Overall, the findings highlight that while deeper architectures such as ResNet50 may struggle with limited medical datasets, efficiently scaled models like EfficientNetB0 can deliver exceptional results when properly fine-tuned.

8 Conclusion and Future work

This brain tumor classification project represents a practical application of modern artificial intelligence in the field of medical diagnostics. The core methodology centers on using transfer learning, a technique that takes pre-existing, powerful image recognition models like ResNet50 and EfficientNet and fine-tunes them for a highly specialized purpose. This approach is both efficient and effective, as it bypasses the need for an impossibly large dataset and immense computational power that would be required to train a complex model from scratch.

The workflow followed a logical and thorough process. It began with a careful inspection and organization of the brain MRI dataset, ensuring the images across the four categories glioma, meningioma, pituitary tumor, and no tumor were sufficiently balanced for reliable training. The subsequent adaptation of the pre-trained models involved strategically replacing their final layers to re-purpose their general pattern-finding capabilities for the specific task of identifying neurological anomalies. Furthermore, the implementation of training safeguards, such as early stopping to prevent over-learning and checkpoints to save the best-performing version of the model, reflects a professional and considered approach to machine learning.

This study can focus on extending the current classification approach by approaching tumor localization, allowing the detection of exact tumor regions in addition to classification. Since this project used ResNet50 and EfficientNetB0 it can improved by experimenting with deeper or more advanced architectures like ensemble learning.Expanding the dataset with more MRI scans from multiple sources would also help improve model generalization and feature extraction.Finally, deploying the model as a clinical decision-making tool could enhance its understanding and practical usefulness for experts.

Ultimately, the true value of this work lies in its potential real-world impact. The goal is to produce a robust and accurate AI tool that can analyze a brain MRI and provide a preliminary classification. Such a system is not intended to replace the experts but to act as a powerful assistant of a tecnology. By offering a fast and consistent second opinion, it has the potential to reduce human error and contribute to earlier detection of brain tumors which is often a critical factor in improving patient health and survival rates. This project serves as a proof-of-concept for the ongoing combination of AI into clinical practice.

REFERENCES

[1] Marco Antonio Gómez-Guzmán,Laura Jiménez-Beristaín, Enrique Efren García-Guerrero,Oscar Roberto López-Bonilla,Ulises Jesús Tamayo-Perez ,José Jaime Esqueda-Elizondo ,Kenia Palomino-Vizcaino and Everardo Inzunza-González,
"Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks", 2023.
https://www.mdpi.com/2079-9292/12/4/955

[2] K. Nishanth Rao , Osamah Ibrahim Khalaf, V. Krishnasree , Aruru Sai Kumar , Deema Mohammed Alsekait , S. Siva Priyanka , Ahmed Saleh Alattas , Diaa Salama AbdElminaam ,
"An efficient brain tumor detection and classification using pre-trained convolutional neural network models,".
https://www.sciencedirect.com/science/article/pii/S2405844024128046

[3] Soheila Saeedi ,Sorayya Rezayi ,Hamidreza Keshavarz Sharareh R. Niakan Kalhori ,
"MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques,", 2025.
https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-023-02114-6

[4] Aqib Ali ,Xinde Li,Wali Khan Mashwani ,Mohammad Abiad, Faten Khalid Karim Samih M. Mostafa ,
"Multi-class brain tumor MRI segmentation and classification using deep learning and machine learning approaches,", 2024.
https://cancerimagingjournal.biomedcentral.com/articles/10.1186/s40644-025-00953-2

[5] Gopinath Balaji, Ranit Sen, Harsh Kirty,
"Detection and Classification of Brain tumors Using Deep Convolutional Neural Networks," 2022.
https://arxiv.org/abs/2208.13264