

Brain Tumor Detection and Classification using Fine-Tuned Resnet50 Model

*Software Technologies in Medical
Informatics*

May 2025

Prepared by

Fandishe Mussa

Presentation Content

1. Introduction

- 1.1. Purpose of the project
- 1.2. The statement of problem

2. Materials and Methods

- 2.1. Datasets Description
- 2.2. Tools & Libraries
- 2.3. Data Preprocessing and Augmentation
- 2.4. Model Architecture
- 2.5. Training Strategy and Evaluation Metrics

3. Work and Results

- 3.1. Visualization of Augmented Images
- 3.2. Dataset Composition and Class Balance
- 3.3. Training and Validation Dynamics
- 3.4. Results and Performance Evaluation
- 3.5. Web application

4. Discussion

- 4.1. Dataset
- 4.2. Analysis and Evaluation of the Results
- 4.3. Comparison with Literature
- 4.4. Conclusion
- 4.5. Limitations, Challenges, and Future Work

1. INTRODUCTION

1.1. Problem Statement

- The diagnosis of brain tumors from MRI scans remains a challenging and time-consuming task, particularly due to the visual complexity of tumor types and their overlapping characteristics.
- Manual interpretation by radiologists is prone to inter-observer variability and diagnostic error, especially when dealing with large volumes of imaging data or ambiguous cases.
- Moreover, many existing computer-aided detection systems focus solely on binary classification (i.e., tumor vs. no tumor), which fails to provide the detailed information necessary for effective clinical decision-making.
- The lack of accurate, automated, and interpretable tools for multiclass brain tumor classification limits the potential for timely and precise treatment planning.

1.2. Purpose of The Project

- This study aims to develop an efficient and interpretable deep learning model for multiclass brain tumor classification using MRI images.
- Specifically, the objective is to design and evaluate a Convolutional Neural Network (CNN) based on the ResNet50 architecture leveraging both base and fine-tuned model to classify MRI scans into four categories: glioma, meningioma, pituitary tumor, and no tumor.
- To improve diagnostic accuracy and model robustness, the study integrates advanced image preprocessing techniques and training strategies.
- Additionally, it incorporates explainable AI tools such as Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize and validate the model's focus on tumor regions, thereby enhancing model reliability and model interpretability.
- The ultimate goal is to contribute a reliable, high-performing tool that can assist radiologists in early detection and precise classification of brain tumors.

CONT'D...

Key Contributions

- This study presents several important contributions toward advancing automated brain tumor diagnosis using deep learning techniques:
- **Multiclass Classification Architecture:** We designed a comprehensive Convolutional Neural Network (CNN) framework based on the ResNet50 architecture to perform multiclass classification of brain MRI images into four distinct categories: glioma, meningioma, pituitary tumor, and no tumor. This approach provides more clinically relevant insights compared to conventional binary classifiers.
- **Enhanced Preprocessing Techniques:** The study incorporates image preprocessing methods tailored for medical imaging, including Contrast Limited Adaptive Histogram Equalization (CLAHE) and Gaussian-blur-based sharpening. These techniques improve image contrast and edge definition, facilitating better feature extraction by the CNN.
- **Transfer Learning and Fine-Tuning Strategy:** Leveraging pre-trained weights from ImageNet, we employed a two-phase training strategy involving transfer learning followed by fine-tuning. This enabled the model to adapt domain-specific knowledge while preserving generalizable features learned from large-scale datasets.
- **Interpretability through Grad-CAM:** To promote transparency and clinical trust, we utilized Gradient-weighted Class Activation Mapping (Grad-CAM) to generate heatmaps highlighting the regions of each MRI scan that influenced the model's decision. These visual explanations offer clinicians a clearer understanding of the model's reasoning process.
- **Comprehensive Evaluation and Benchmarking:** The model was rigorously evaluated using a suite of classification metrics—including accuracy, precision, recall, F1-score, specificity, and confusion matrix analysis—across both base and fine-tuned configurations. The results demonstrate substantial improvements in classification performance following fine-tuning.
- **Real-World Integration via Web Interface:** A lightweight, user-friendly web application was developed using Gradio to facilitate real-time tumor prediction and visualization, demonstrating the practical viability of deploying the trained model in clinical or research environments.

2. MATERIALS AND METHODS

2.1. Dataset Description

- This study utilized a composite brain MRI dataset compiled from two publicly available sources on Kaggle: the Msoud and Bilalakgz repositories.
- The Msoud dataset is a merged collection that includes images from Figshare, SARTAJ, and BR35H datasets.
- It comprises 7,023 training images and 1,311 test images.
- The Bilalakgz dataset contributed an additional 1,559 training images and 394 test images.
- After integration and cleaning, the final dataset consisted of 8,582 training images and 1,705 testing images, encompassing four classes: glioma, meningioma, pituitary tumor, and no tumor.
- All images were grayscale in JPEG format, resized to a uniform dimension of 224×224 pixels for model compatibility.

2.2. Tools and Environments

- Model development and experimentation were primarily conducted in Python using the TensorFlow and Keras deep learning libraries.
- The training environment was hosted on **Google Colab Pro**, which provided a high-performance computational backend consisting of an NVIDIA A100 GPU, 40 GB RAM, and 228 GB of storage.
- For offline testing and prototyping, a Windows 11 laptop equipped with an Intel Core i7 processor and 8 GB RAM was used.

Key libraries and tools used:

- TensorFlow with Keras API for deep learning model construction.
- OpenCV for advanced image processing operations.
- scikit-learn for metrics computation and label encoding.
- Matplotlib and Seaborn for data visualization.
- Gradio for developing a user-accessible web application.

CONT'D...

2.3. Data Preprocessing and Augmentation

- Preprocessing played a crucial role in preparing the dataset for training any model.
- It involved a series of steps that were designed to enhance the performance and generalization ability of the model.
- In this project, comprehensive preprocessing and augmentation pipeline were applied to the train dataset, to enhance the quality, diversity, robustness and optimize the training process and improve the model's accuracy of the training data.

2.3.1. Image Enhancement and Normalization

- The raw brain MRI images, initially provided in grayscale JPEG format, underwent several preprocessing operations aimed at improving visual quality and highlighting tumor-related features.
- First, **Gaussian-blur-based sharpening** was applied using a **5×5 kernel** to reduce noise and enhance important structural edges.
- This was followed by **Contrast Limited Adaptive Histogram Equalization** (CLAHE), implemented with a tile grid size of 8×8 and a clip limit of 2.0. CLAHE significantly improved local contrast, enabling better feature extraction during model training.
- Subsequently, grayscale images were converted to RGB format to ensure compatibility with deep learning models requiring three input channels.
- Pixel values of all images were rescaled to the range [0, 1] using a normalization factor of **1./255**.
- This normalization was crucial for stabilizing the training process and accelerating convergence.

```
# CLAHE
clahe_applied = clahe.apply(gray_resized)

# Sharpening
blurred = cv2.GaussianBlur(clahe_applied, (0, 0), sigmaX=1.5)
sharpened = cv2.addWeighted(clahe_applied, 1.5, blurred, -0.5, 0)
```

CONT'D...

2.3.2. Training Data Augmentation

- To increase the effective size and diversity of the training dataset, real-time data augmentation was applied only to the training set.
- This technique helps the model learn robust features and generalize better to unseen images.
- The following augmentation parameters were applied using **ImageDataGenerator**:
- **Rescaling**: rescale=1./255 – Normalizes image pixel values to the [0, 1] range, which improves model convergence.
- **Zoom Range**: zoom_range=0.1 – Randomly zooms images up to ±10%, simulating differences in magnification levels.
- **Width and Height Shift Range**: width_shift_range=0.05, height_shift_range=0.05 – Randomly shifts the image horizontally or vertically by up to 5%, mimicking off-centered MRI captures.
- **Shear Range**: shear_range=0.05 – Applies small shearing transformations, introducing minor geometric distortions that make the model more resilient to variations in angle and shape.
- **Horizontal Flip**: horizontal_flip=True – Horizontally flips images at random, allowing the model to handle left-right anatomical variability.
- **Fill Mode**: fill_mode='nearest' – Used to fill in new pixels created by transformations using nearest-neighbor values.
- These augmentations were implemented using the following line of code:

```
train_datagen = ImageDataGenerator(  
    rescale=1./255,  
    rotation_range=0,  
    zoom_range=0.1,  
    width_shift_range=0.05,  
    height_shift_range=0.05,  
    shear_range=0.05,  
    horizontal_flip=True,  
    fill_mode='nearest'  
)  
  
train_generator = train_datagen.flow(X_train, y_train, batch_size=batch_size, shuffle=True)
```

CONT'D...

2.3.3. Validation and Test Data Handling

- To fairly evaluate the model's ability to generalize, **no augmentation** was applied to **validation** and **test sets**.
- Only rescaling was used for testing set.

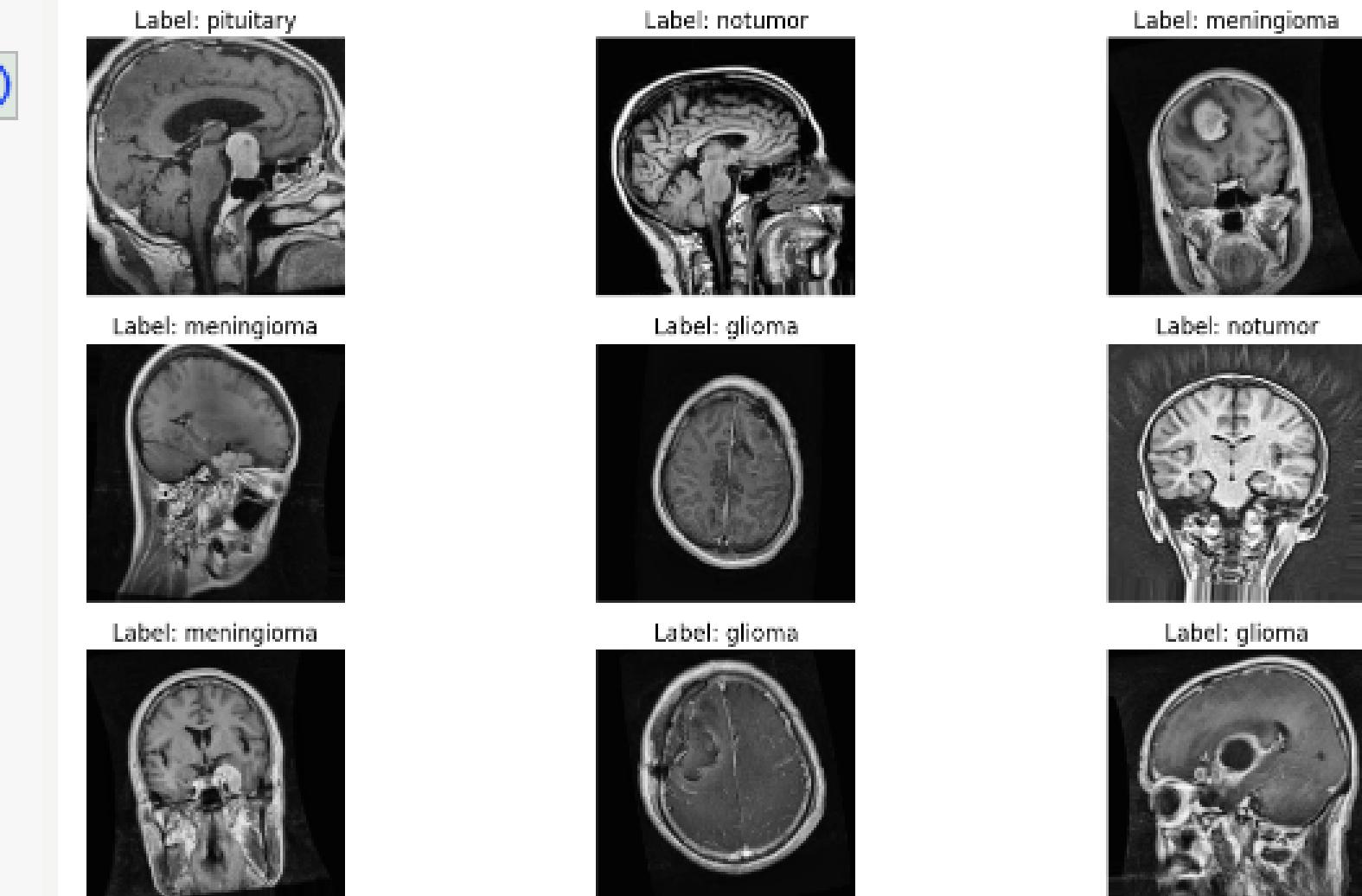
```
val_test_datagen = ImageDataGenerator(rescale=1./255) # No augmentation

val_generator = val_test_datagen.flow(X_val, y_val, batch_size=batch_size, shuffle=False)
test_generator = val_test_datagen.flow(X_test, y_test, batch_size=batch_size, shuffle=False)
```

2.3.4. Augmentation Visualization

- To verify the effectiveness of the augmentation process, a batch of augmented training images was visualized along with their associated class labels.
- **Labels** were decoded using one-hot reversal (np.argmax()) and mapped via the trained **LabelEncoder**.
- This visual check confirmed the successful application of variability while preserving clinical relevance in the augmented images.

```
# Get a batch of augmented images and labels
augmented_images, augmented_labels = next(train_generator)
# Reverse one-hot encoding to get class indices
label_indices = np.argmax(augmented_labels, axis=1)
class_names = label_encoder.classes_
plt.figure(figsize=(12, 12))
for i in range(9):
    plt.subplot(5, 3, i + 1)
    img = np.clip(augmented_images[i], 0, 1)
    plt.imshow(img)
    plt.title(f"Label: {class_names[label_indices[i]]}")
    plt.axis('off')
plt.tight_layout()
plt.show()
```



CONT'D...

2.4. Model Architecture

- The backbone of our brain tumor classification system is based on the ResNet50 architecture, a widely adopted deep convolutional neural network known for its effectiveness in medical imaging tasks.
- We utilized a transfer learning approach by loading ResNet50 pretrained on **ImageNet**, removing its original classification head, and appending a custom layer stack tailored for multiclass brain tumor detection.

Key Architectural Modifications Included:

- Retaining the **ResNet50** convolutional base for feature extraction.
- Freezing the base layers during initial training to preserve pre-learned features.
- Appending a lightweight classification head composed of **global average pooling**, **normalization**, **dropout** for regularization, and two fully connected layers.
- Using a **softmax output layer** to support four-class classification (glioma, meningioma, pituitary tumor, and no tumor).
- This design enables the model to learn high-level abstract features while minimizing overfitting through regularization and transfer learning.
- The use of pretrained weights accelerates convergence and enhances generalization, especially valuable given the limited size of medical image datasets.

```
# Model parameters
input_shape = (224, 224, 3)
num_classes = y_train.shape[1]

# Load ResNet50 base
base_model = ResNet50(include_top=False, weights='imagenet', input_shape=input_shape)
base_model.trainable = False # freeze base layers

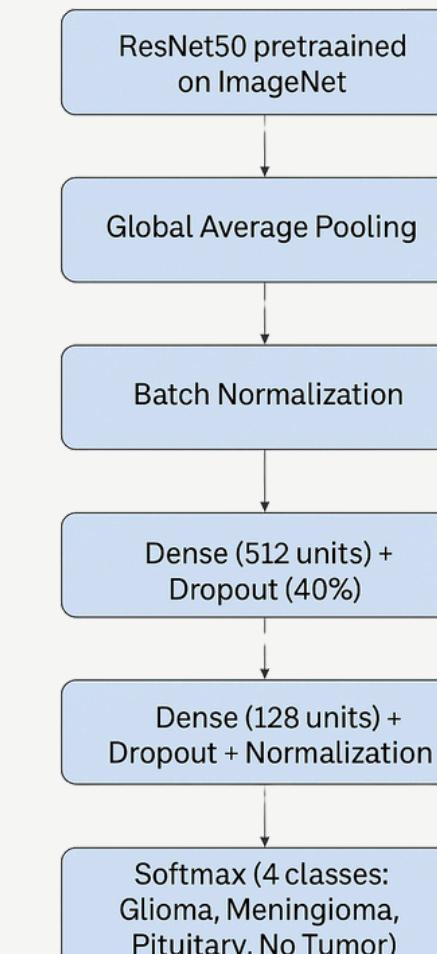
# Build top model
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = BatchNormalization()(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.4)(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.4)(x)
x = BatchNormalization()(x)
predictions = Dense(num_classes, activation='softmax')(x)

# Final model
model = Model(inputs=base_model.input, outputs=predictions)

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

model.summary()
```

Model Architecture – ResNet50-Based Classifier



CONT'D

2.5. Training Strategy and Evaluation Metrics

Training Strategy(Feature Extraction Phase)

- The model training process was executed in two distinct stages to leverage transfer learning and maximize classification performance:

Stage 1: Transfer Learning

- The ResNet50 base was frozen (`trainable = False`), and only the custom top layers were trained.
- This enabled the model to utilize pretrained visual features while adapting the new layers to the tumor classification task.
- Training used the Adam optimizer with a learning rate of 0.0001 and categorical crossentropy loss.

Stage 2: Fine-Tuning(Full Network Training Phase)

- The entire ResNet50 model, including its convolutional layers, was unfrozen (`trainable = True`).
- A reduced learning rate of 0.00001 (1e-5) was used to fine-tune the model gently and avoid overwriting previously learned features.
- This phase improved the model's capacity to capture domain-specific patterns in MRI data without overfitting..

To ensure optimal training, the following callbacks were implemented:

- EarlyStopping**: Halted training when validation loss failed to improve for 5 consecutive epochs.
- ReduceLROnPlateau**: Automatically reduced learning rate on performance plateaus.
- ModelCheckpoint**: Saved the best-performing model based on minimum validation loss.
- TensorBoard**: Enabled real-time monitoring of loss and accuracy curves.
- Class weighting** was applied using scikit-learn's `compute_class_weight` to mitigate minor class imbalances and ensure balanced learning across all tumor categories.

```
base_model.trainable = True

model.compile(optimizer=Adam(learning_rate=1e-5),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
# Get numeric labels for computing weights
y_train_labels = np.argmax(y_train, axis=1)

# Compute class weights
class_weights_array = compute_class_weight(class_weight='balanced',
                                            classes=np.unique(y_train_labels),
                                            | y=y_train_labels)
class_weights = dict(enumerate(class_weights_array))
print("Class weights:", class_weights)

Class weights: {0: np.float64(0.999563191613279), 1: np.float64(0.9926257952573742), 2: np.float64(1.0780464824120604), 3: np.float64(0.9393814997263273)}
```

CONT'D

Evaluation Metrics

- Model performance was assessed using standard classification metrics computed on a held-out test set and validation set.
- These included:

Accuracy: Overall correctness of predictions.

Precision: Proportion of true positives among predicted positives.

Recall (Sensitivity): Ability to correctly identify actual positives.

F1-Score: Harmonic mean of precision and recall, balancing both metrics.

Specificity: Correct identification of negative cases.

Confusion Matrix: Visual representation of true vs. predicted classes for detailed analysis.

- In addition, model interpretability was enhanced using Grad-CAM visualizations, which confirmed the model's focus on relevant tumor regions in MRI images during classification.
- These visual insights are critical for building reliable and validating model decisions.

```
class_names = list(label_encoder.classes_)
log_root = "new_logs"
log_dir = os.path.join(log_root, datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
file_writer_cm = tf.summary.create_file_writer(log_dir)

tensorboard = TensorBoard(
    log_dir=log_dir,
    histogram_freq=1,
    write_graph=True,
    write_images=True,
    update_freq='epoch',
    profile_batch=0
)

checkpoint_h5 = ModelCheckpoint(
    filepath='before_model_fine_tuned.h5',
    monitor='val_loss', verbose=1, save_best_only=True, mode='min'
)
checkpoint_keras = ModelCheckpoint(
    filepath='before_model_fine_tuned.keras',
    monitor='val_loss', verbose=1, save_best_only=True, mode='min'
)
ES = EarlyStopping(monitor='val_loss', min_delta=0.001, patience=5, restore_best_weights=True, verbose=1, mode='min')
RL = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=5, verbose=1, mode='min')
callbacks = [checkpoint_h5, checkpoint_keras, tensorboard, ES, RL]
# Train
history = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=30,
    steps_per_epoch=len(train_generator),
    validation_steps=len(val_generator),
    class_weight=class_weights,
    callbacks=callbacks,
    verbose=1
)

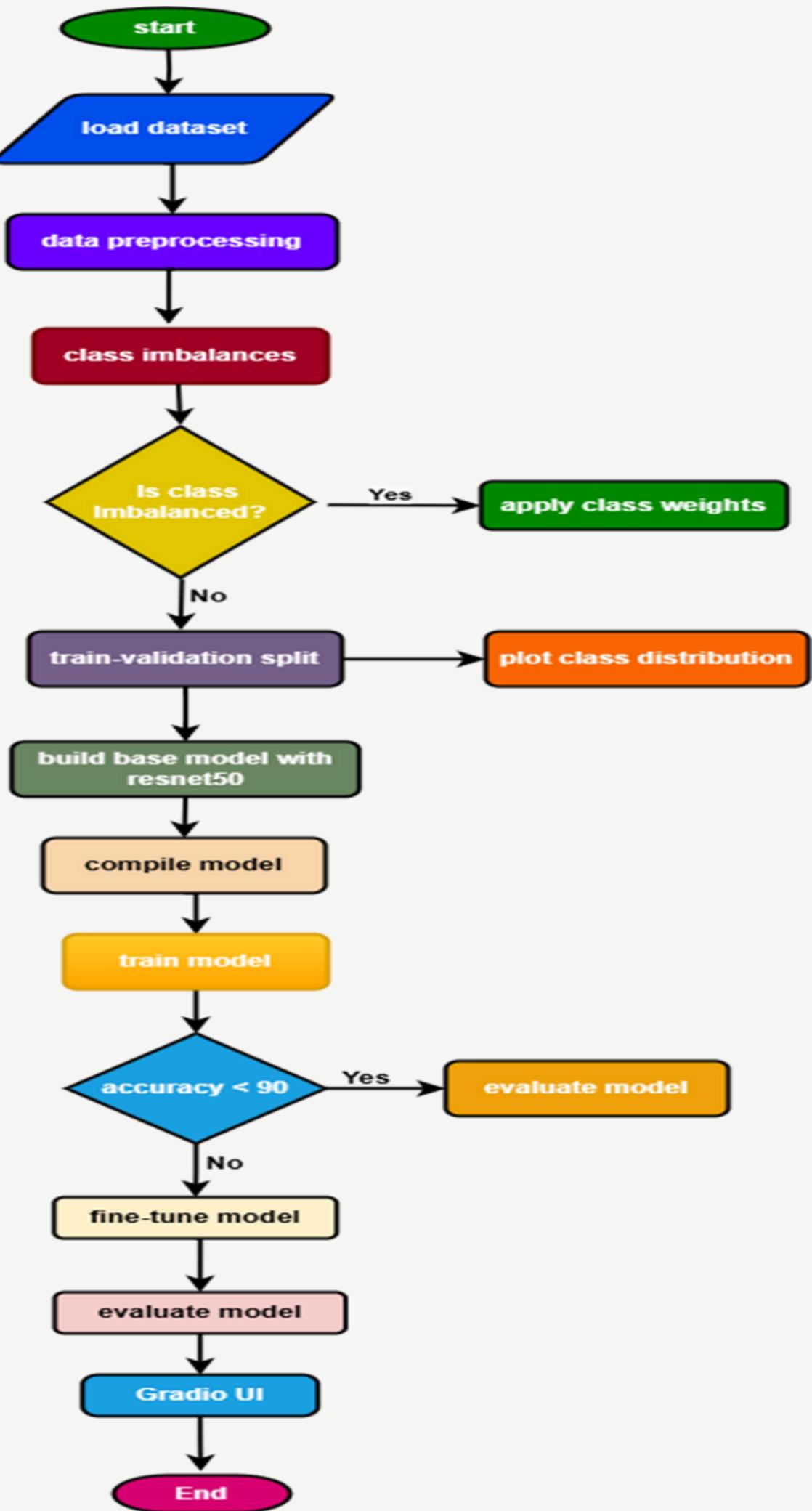
with open("history_base.pkl", "wb") as f:
    pickle.dump(history.history, f)

fine_tuned_class_names = list(label_encoder.classes_)
fine_tuned_log_root = "fine_tuned_logs"
fine_tuned_log_dir = os.path.join(fine_tuned_log_root, datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
fine_tuned_file_writer_cm = tf.summary.create_file_writer(fine_tuned_log_dir)
fine_tuned_tensorboard = TensorBoard(
    log_dir=fine_tuned_log_dir,
    histogram_freq=1,
    write_graph=True,
    write_images=True,
    update_freq='epoch',
    profile_batch=0
)

fine_tuned_checkpoint_h5 = ModelCheckpoint(
    filepath='fine_tuned_model.h5',
    monitor='val_loss', verbose=1, save_best_only=True, mode='min'
)
fine_tuned_checkpoint_keras = ModelCheckpoint(
    filepath='fine_tuned_model.keras',
    monitor='val_loss', verbose=1, save_best_only=True, mode='min'
)

Fine_Tuned_ES = EarlyStopping(monitor='val_loss', min_delta=0.001, patience=5, restore_best_weights=True, verbose=1, mode='min')
Fine_Tuned_RL = ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=5, verbose=1, mode='min')
fine_tuned_callbacks = [fine_tuned_checkpoint_h5, fine_tuned_checkpoint_keras, fine_tuned_tensorboard, Fine_Tuned_ES, Fine_Tuned_RL]
history_fine_tuned = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=20,
    steps_per_epoch=len(train_generator),
    validation_steps=len(val_generator),
    class_weight=class_weights,
    callbacks=fine_tuned_callbacks
)

with open("history_ft.pkl", "wb") as f:
    pickle.dump(history_fine_tuned.history, f)
```

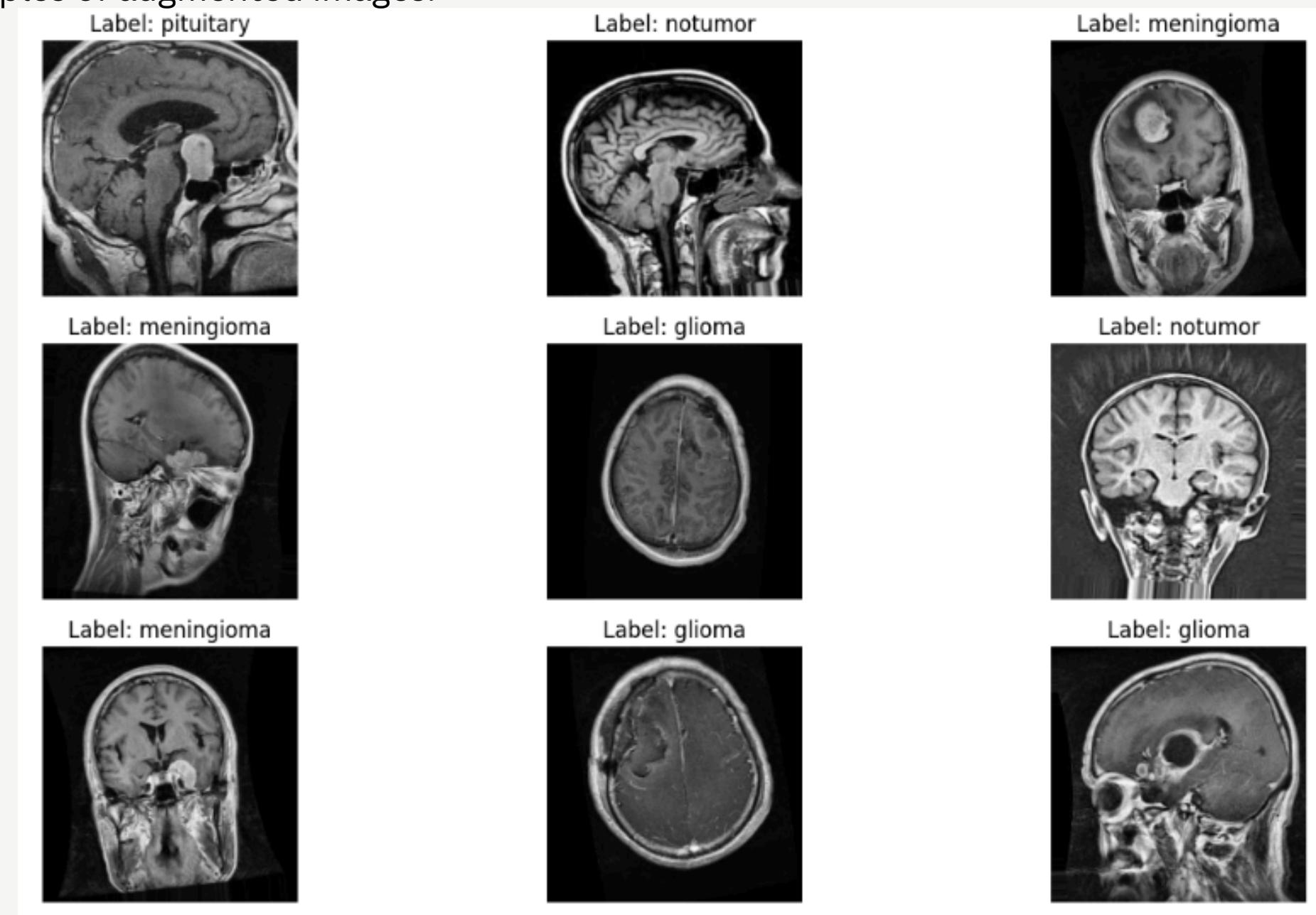


3. WORK AND RESULTS

- This section presents the performance outcomes of both the base and fine-tuned ResNet50 models, highlighting their effectiveness in classifying brain MRI images into four categories: glioma, meningioma, pituitary tumor, and no tumor.

3.1. Visualization of Augmented Images

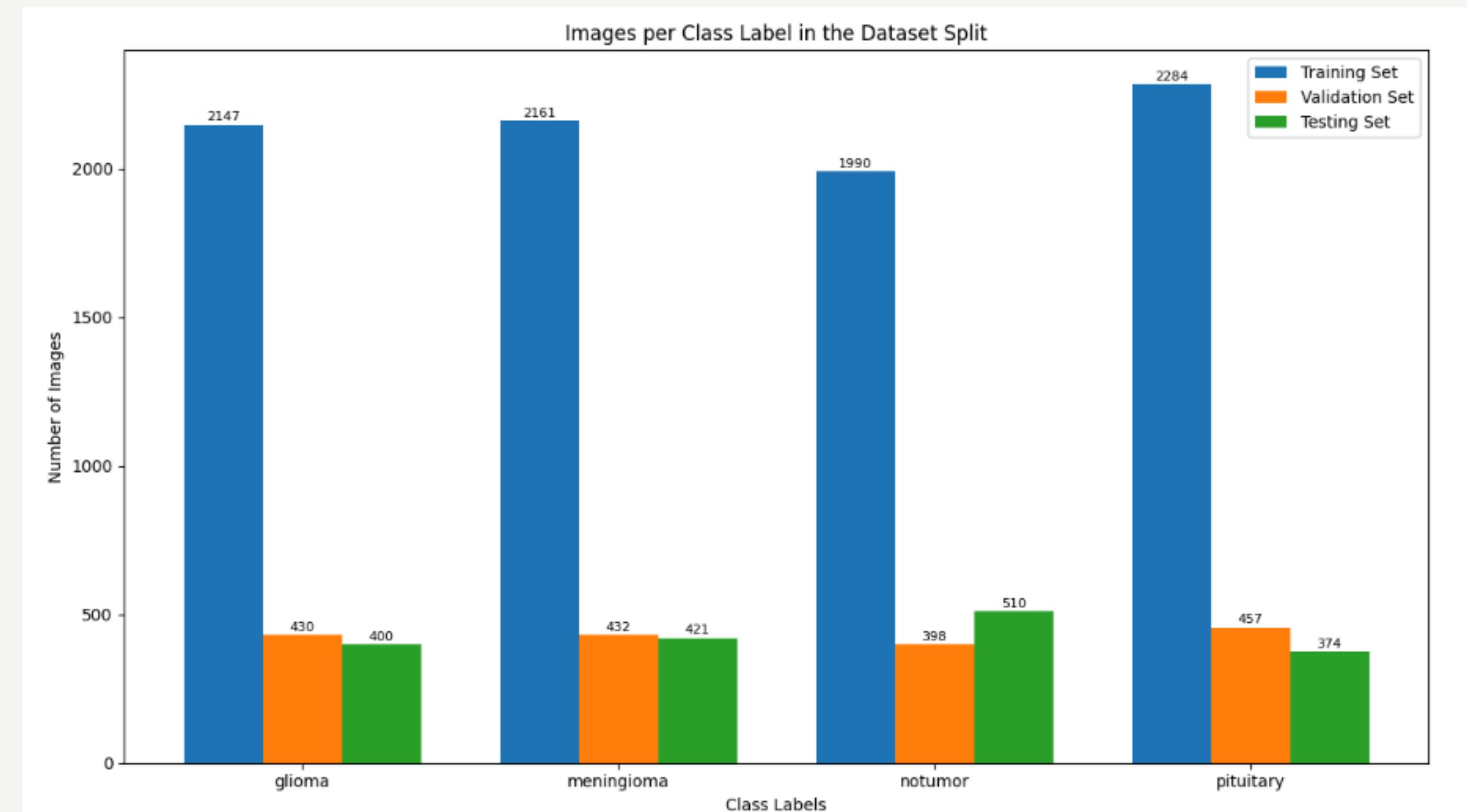
- To improve model generalization and reduce overfitting, a suite of real-time augmentation techniques was applied using TensorFlow's **ImageDataGenerator**.
- These transformations simulate variations in brain MRI acquisition.
- A batch of augmented images was visualized, confirming successful image distortion while retaining anatomical features.
- See the following samples of augmented images:



CONT'D

3.2. Dataset Composition and Class Balance

- The constructed brain MRI classification model was trained on a balanced and preprocessed dataset composed of the following:
 - **Glioma:** 2,147 images
 - **Meningioma:** 2,161 images
 - **Pituitary:** 2,284 images
 - **No Tumor:** 1,990 images
- To evaluate the model's generalization ability, a separate validation set was used, which consisted of:
 - **Glioma:** 430 images
 - **Meningioma:** 432 images
 - **Pituitary:** 457 images
 - **No Tumor:** 398 images



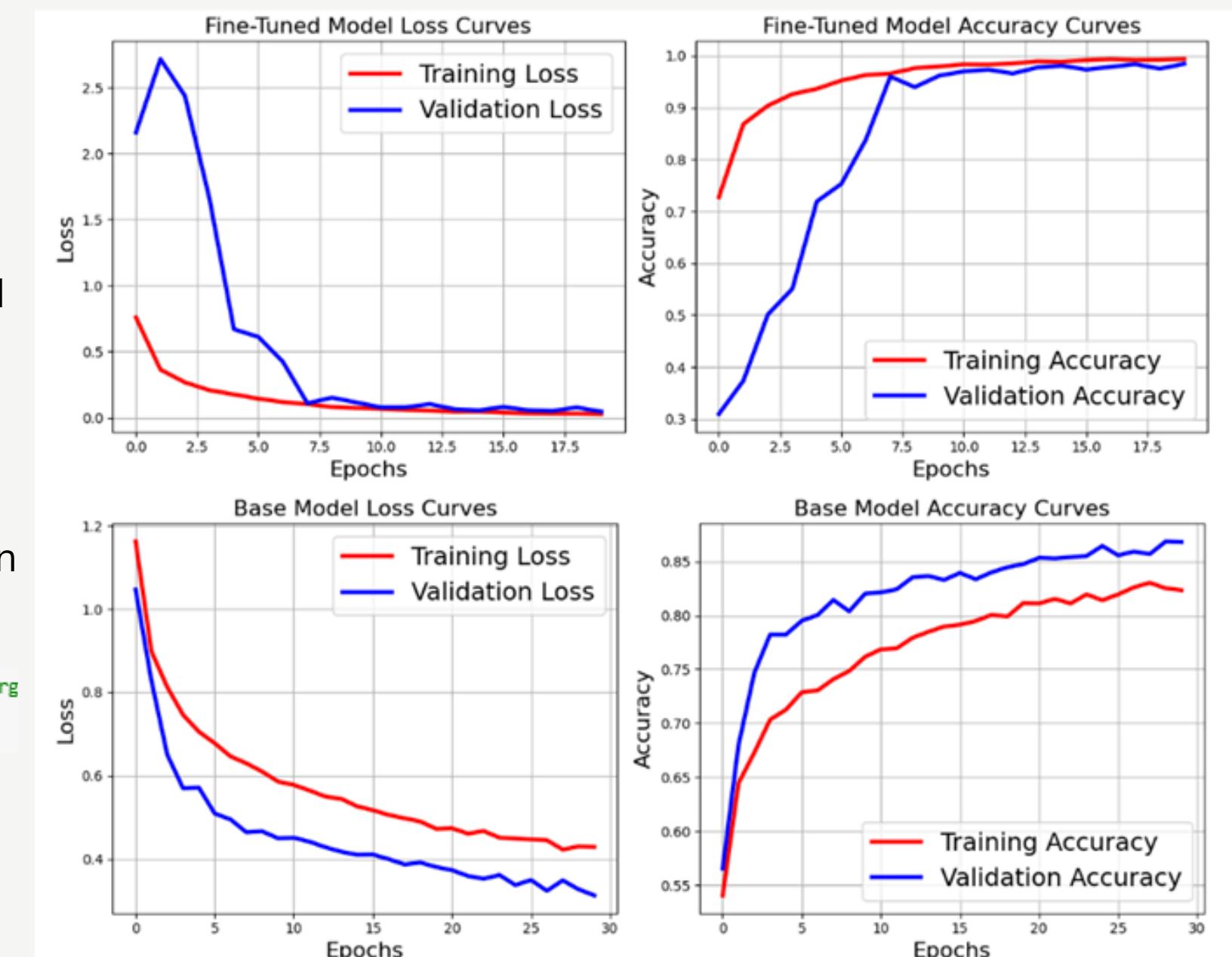
CONT'D

3.3. Training and Validation Dynamics

- The model was compiled and trained using **categorical cross-entropy** as the **loss function** and **accuracy** as the evaluation metric.
- The training process was monitored over multiple epochs, and the results were visualized in the form of accuracy and loss curves, as shown in the below figures.
- The training dynamics revealed a consistent increase in accuracy and a corresponding decrease in **loss** over the **epochs**, indicating that the model was effectively learning the patterns in the data.
- This behavior is characteristic of **model convergence**, a state in which the model has successfully minimized the loss function and aligned its predictions more closely with the true labels.
- However, it is essential to monitor for **overfitting**, which occurs when the model becomes too specialized to the training data and performs poorly on unseen data.
- Signs of overfitting include increasing training accuracy while validation accuracy plateaus or declines.
- To mitigate this, techniques such as **early stopping**, **validation monitoring**, and **learning rate reduction** using **ReduceLROnPlateau** were employed during training.
- Overall, the training and validation results suggest that the model demonstrated good learning behavior and convergence, successfully improving its prediction capability over time while maintaining performance on the validation set.

```
: EarlyStopping(monitor='val_loss', min_delta=0.001, patience=5, restore_best_weights=True, verbose=1, mode='min') # I used this to manage convergence
: ReduceLROnPlateau(monitor='val_loss', factor=0.3, patience=5, verbose=1, mode='min') # I also used this to manage convergence as well
callbacks = [fine_tuned_checkpoint_h5, fine_tuned_checkpoint_keras, fine_tuned_tensorboard, Fine_Tuned_ES, Fine_Tuned_RL]
```

```
base_model.trainable = True
model.compile(optimizer=Adam(learning_rate=1e-5),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary()
```



CONT'D

3.4. Results and Performance Evaluation

- This section presents a comparative analysis of the base model, and the fine-tuned model used for classifying brain tumor MRI images into four classes: glioma, meningioma, no tumor, and pituitary.
- Evaluation was performed using **standard classification metrics**, **confusion matrices**, and **validation statistics** recorded during training and testing phases.

Confusion Matrix Analysis – Base Model

- The base ResNet50 model, trained without fine-tuning or advanced augmentation strategies, served as a performance benchmark.
- The raw confusion matrix is restructured using a one-vs-all approach to analyze per-class behavior in terms of **True Positives (TP)**, **False Positives (FP)**, **False Negatives (FN)**, and **True Negatives (TN)**.

Actual/Predicted	Glioma	Meningioma	No Tumor	Pituitary
Glioma	323	96	3	8
Meningioma	30	342	20	40
No Tumor	0	9	388	1
Pituitary	3	4	8	442

$$\text{Accuracy} = \frac{\text{Total True Positives}}{\text{Total Number of Samples}}$$

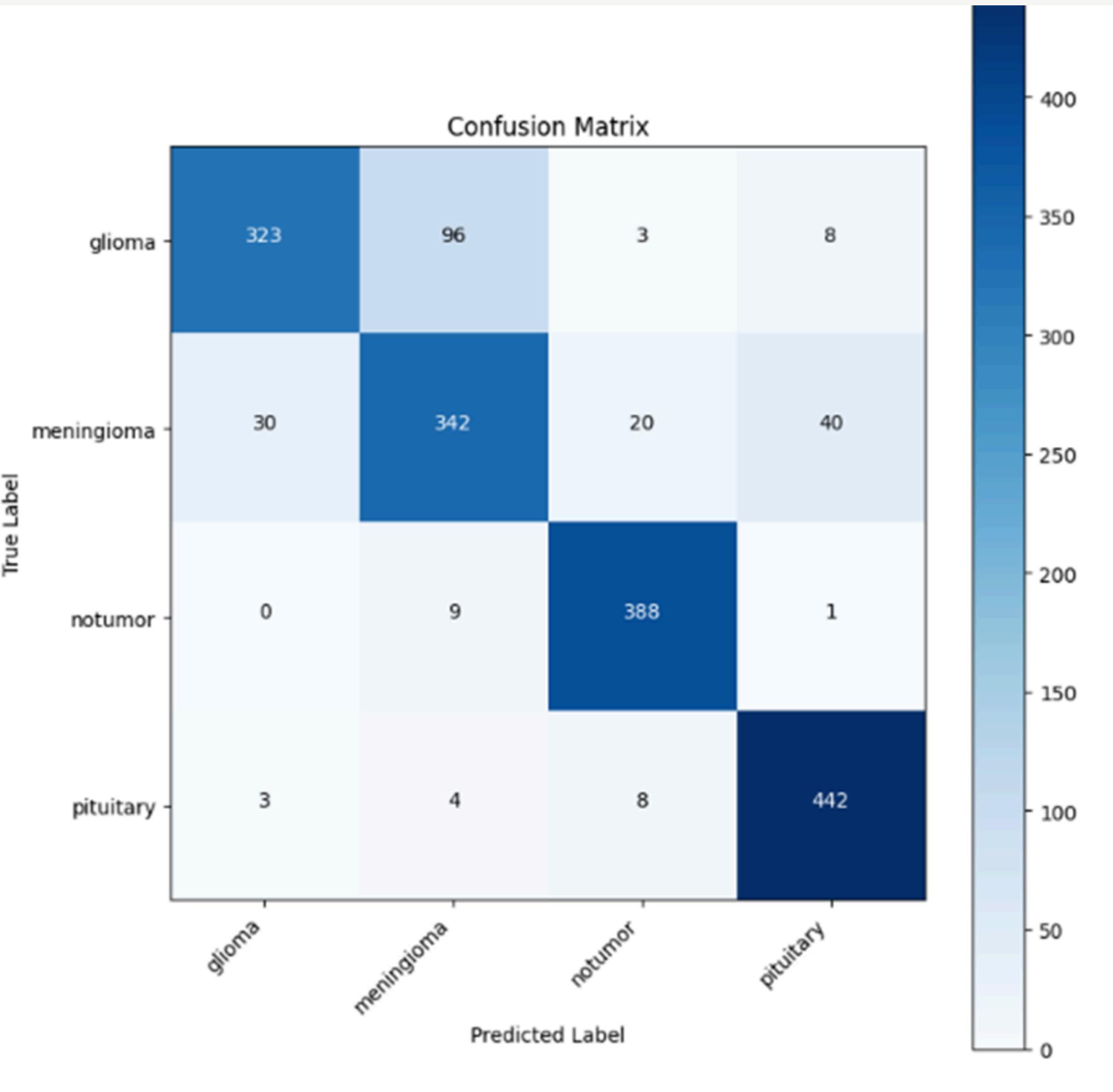
$$\text{Total TP} = 323 + 342 + 388 + 442 = 1495$$

Total number of samples (sum of all entries in the confusion matrix):

$$\text{Total Samples} = \sum \text{all cells} = 323 + 96 + 3 + 8 + 30 + 342 + 20 + 40 + 0 + 9 + 388 + 1 + 3 + 4 + 8 + 442 = 1833$$

$$\text{Accuracy} = \frac{1495}{1833} \approx 0.8157 \text{ or } 81.6\%$$

Class	True Positive	True Negative	False Positive	False Negative
Glioma	323	30 (from Meningioma) + 0 (No Tumor) + 3 (Pituitary) = 33	96 (as Meningioma) + 3 (No Tumor) + 8 (Pituitary) = 107	$342 + 20 + 40 + 9 + 388 + 1 + 4 + 8 + 442 = \mathbf{1212}$
Meningioma	342	96 (from Glioma) + 9 (No Tumor) + 4 (Pituitary) = 109	30 (as Glioma) + 20 (No Tumor) + 40 (Pituitary) = 90	$323 + 3 + 8 + 0 + 388 + 1 + 3 + 8 + 442 = \mathbf{1174}$
No Tumor	388	3 (from Glioma) + 20 (Meningioma) + 1 (Pituitary) = 31	0 (as Glioma) + 9 (Meningioma) + 1 (Pituitary) = 10	$323 + 96 + 8 + 30 + 342 + 40 + 3 + 4 + 442 = \mathbf{1287}$
Pituitary	442	8 (from Glioma) + 40 (Meningioma) + 1 (No Tumor) = 49	3 (as Glioma) + 4 (Meningioma) + 8 (No Tumor) = 15	$323 + 96 + 3 + 30 + 342 + 20 + 0 + 9 + 388 = \mathbf{1281}$



CONT'D

Confusion Matrix Analysis – Fine-Tuned Model

- The fine-tuned ResNet50 model demonstrated a significant performance improvement over the base model by leveraging transfer learning and advanced data augmentation strategies.
- The confusion matrix was reformatted into a one-vs-all binary scheme for each class to extract meaningful evaluation metrics: **True Positives (TP)**, **False Positives (FP)**, **False Negatives (FN)**, and **True Negatives (TN)**.

Actual/Predicted	Glioma	Meningioma	No Tumor	Pituitary
Glioma	429	0	0	1
Meningioma	13	409	0	10
No Tumor	0	0	396	2
Pituitary	0	1	0	456

$$\text{Accuracy} = \frac{\text{Total True Positives}}{\text{Total Number of Samples}}$$

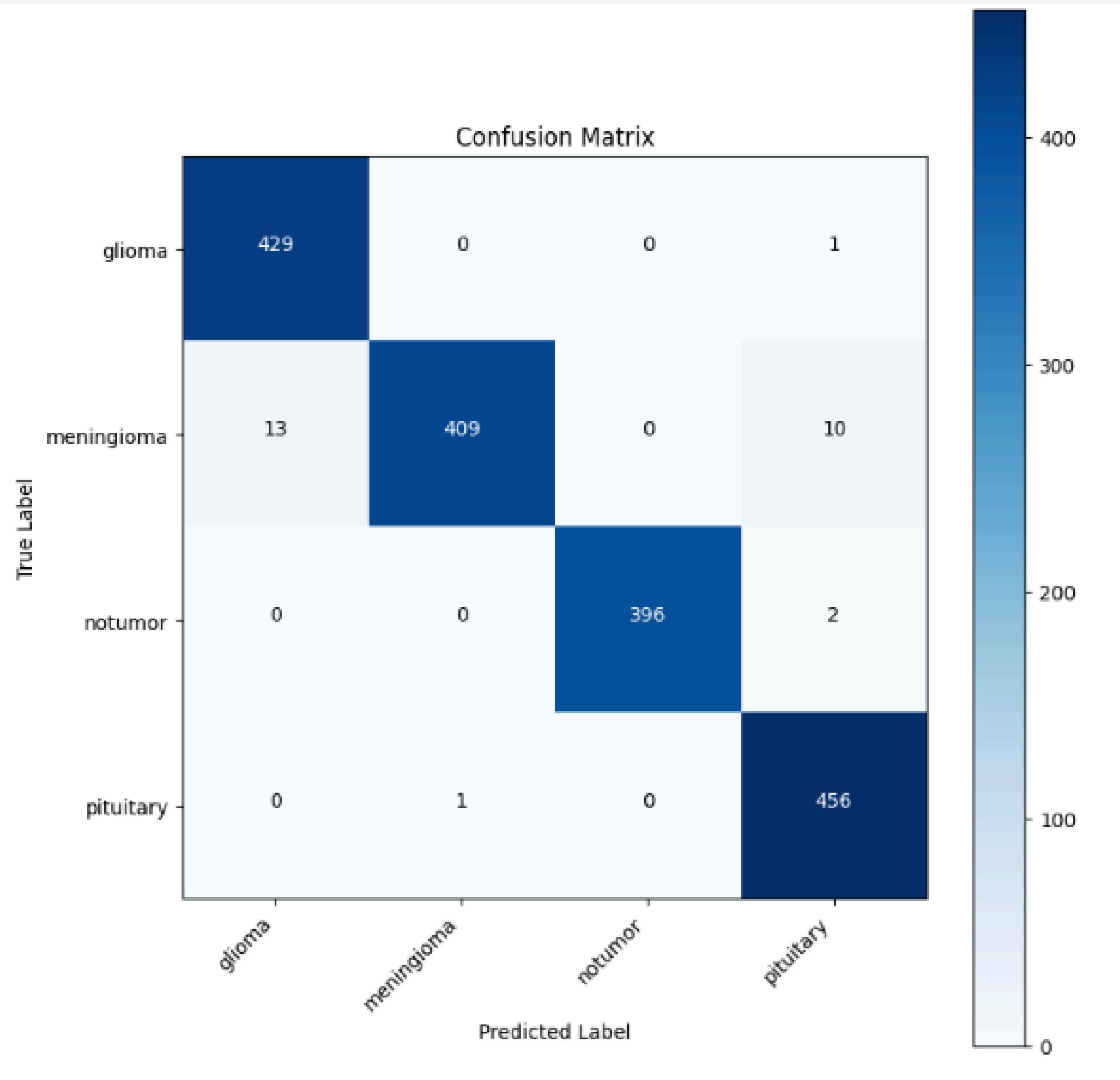
$$\text{Total TP} = 429 + 409 + 396 + 456 = 1690$$

Total number of samples (sum of all entries in the confusion matrix):

$$\text{Total Samples} = \sum \text{all cells} = 429 + 1 + 13 + 409 + 10 + 396 + 2 + 456 + 1 = 1833$$

$$\text{Accuracy} = \frac{1690}{1833} \approx 0.922 \Rightarrow 92.2\%$$

Class	True Positive	True Negative	False Positive	False Negative
Glioma	429	1274	13	1
Meningioma	409	1284	1	23
No Tumor	396	1319	0	2
Pituitary	456	1247	13	1



CONT'D

Base Model Performance Analysis

- The base ResNet50 model, trained without fine-tuning, was evaluated using key classification metrics and training history.
- Its performance across the four brain MRI classes: **glioma**, **meningioma**, **no tumor**, and **pituitary**.
- It provides an essential benchmark for assessing the effectiveness of advanced **fine-tuning** strategies.

1. Classification Performance

- The following table summarizes the precision, recall, and F1-score for each class:

Class	Precision	Recall	F1-Score	Support
Glioma	0.91	0.75	0.82	430
Meningioma	0.76	0.79	0.77	432
No Tumor	0.93	0.97	0.95	398
Pituitary	0.90	0.97	0.93	457

- The model performed best in classifying no tumor and pituitary images, with high **recall** and **F1-scores**.
- Performance for glioma was notably lower in **recall** (0.75), indicating missed cases (**false negatives**).
- Meningioma** had the weakest overall performance, with moderate **precision** and **recall**.
- Overall Accuracy: 87%
- Macro Average F1-Score: 0.87
- Weighted Average F1-Score: 0.87
- These results indicate that the base model is effective at general classification but struggles with inter-class differentiation, particularly between **glioma** and **meningioma** tumors.

CONT'D

Fine-Tuned Model Performance Analysis

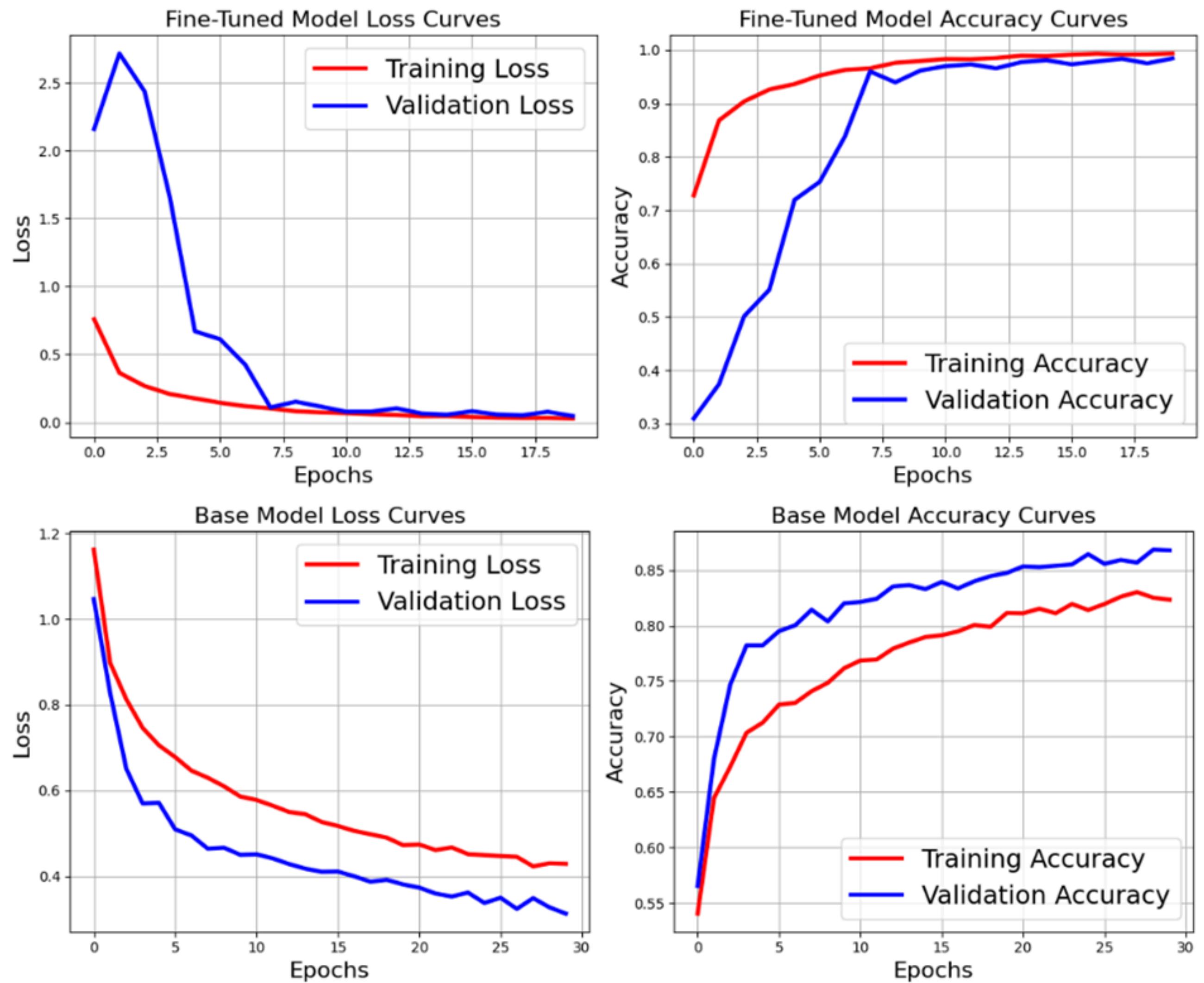
- Following the initial benchmarking with a base ResNet50 model, a fine-tuning strategy was employed to enhance classification performance across the brain MRI dataset.
- This involved unfreezing selected top layers of the pre-trained ResNet50 network, applying lower learning rates, and incorporating advanced regularization and optimization techniques.
- The fine-tuned model demonstrated significant improvements in both accuracy and generalization, outperforming the base model across all key metrics.

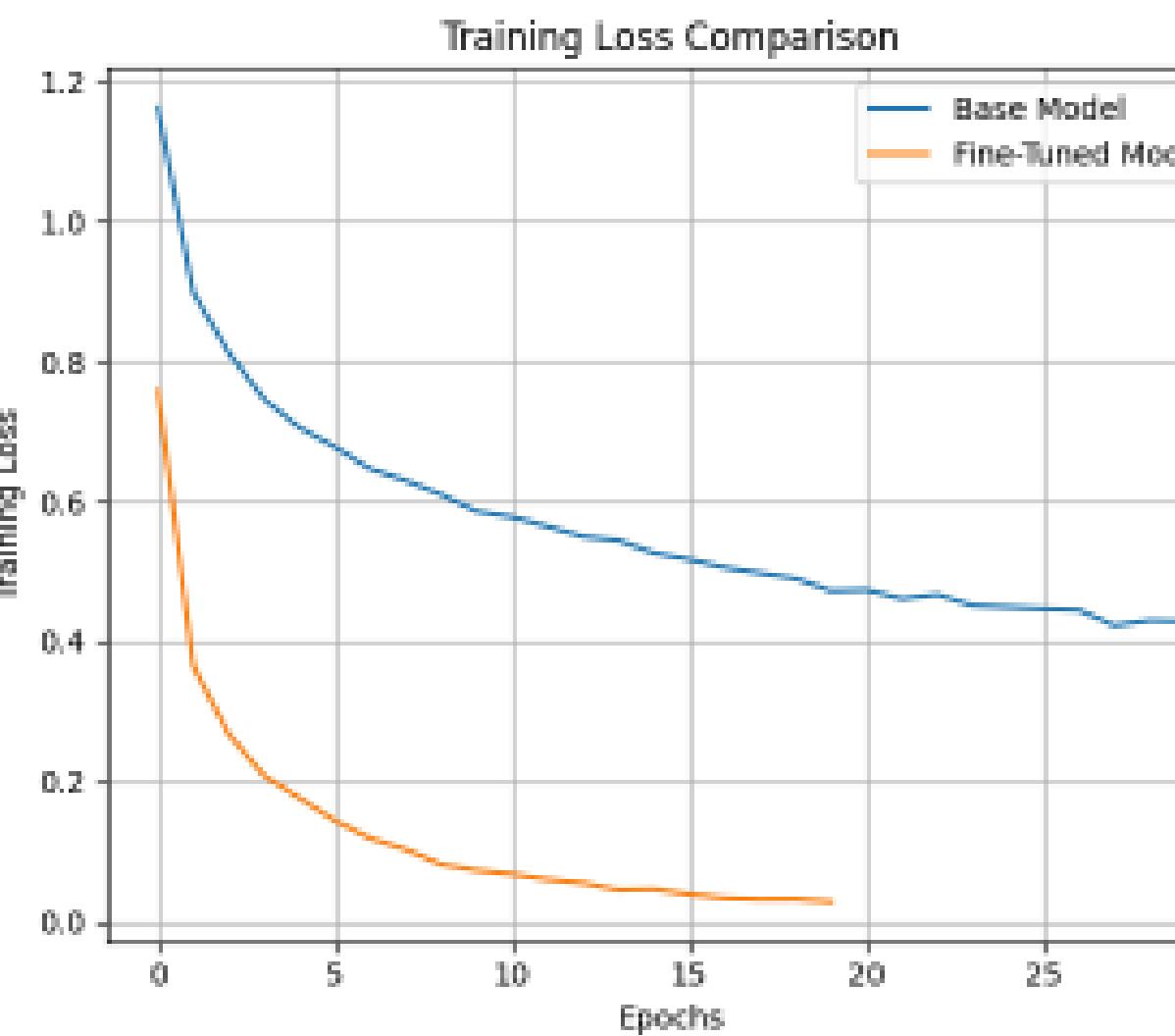
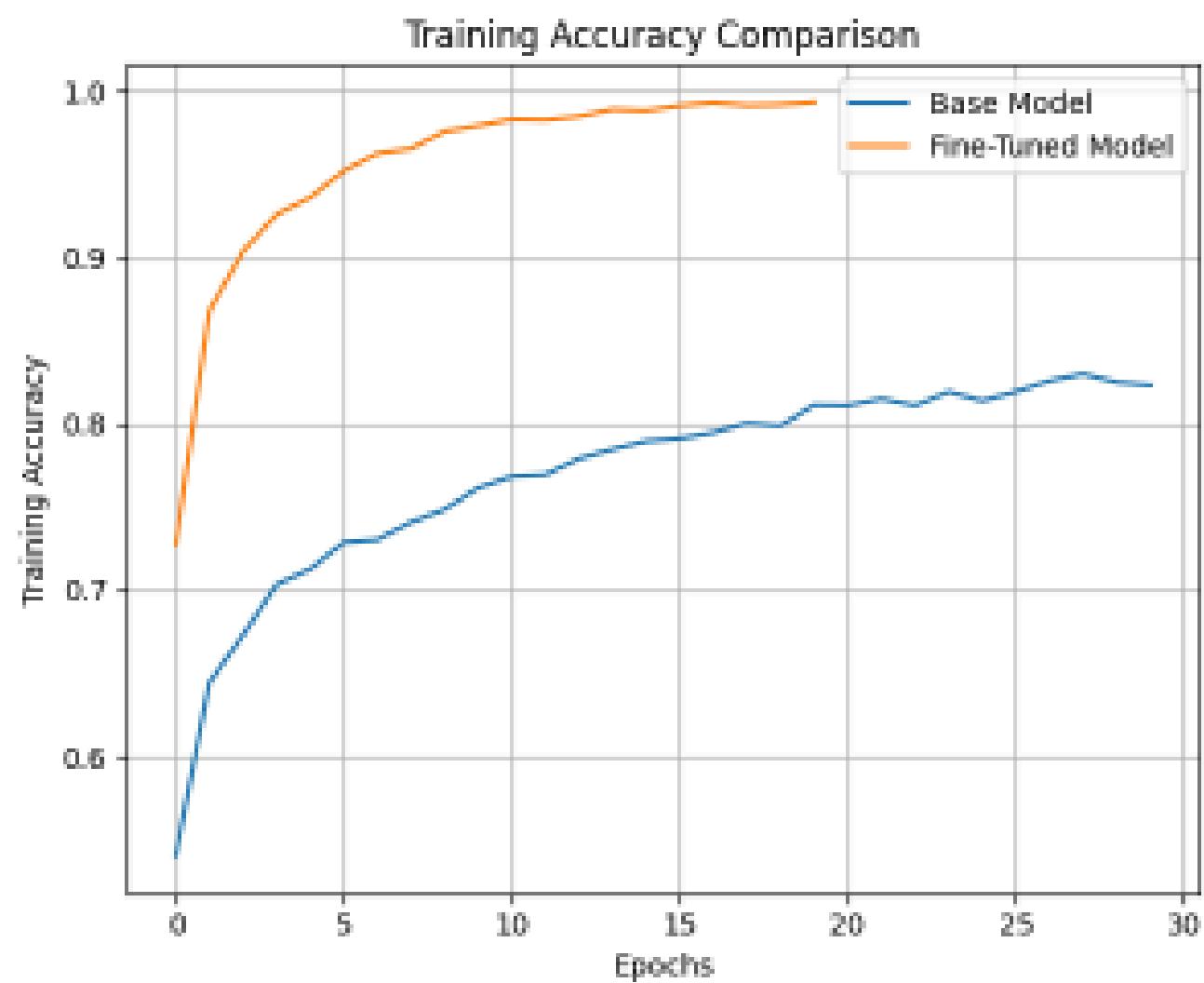
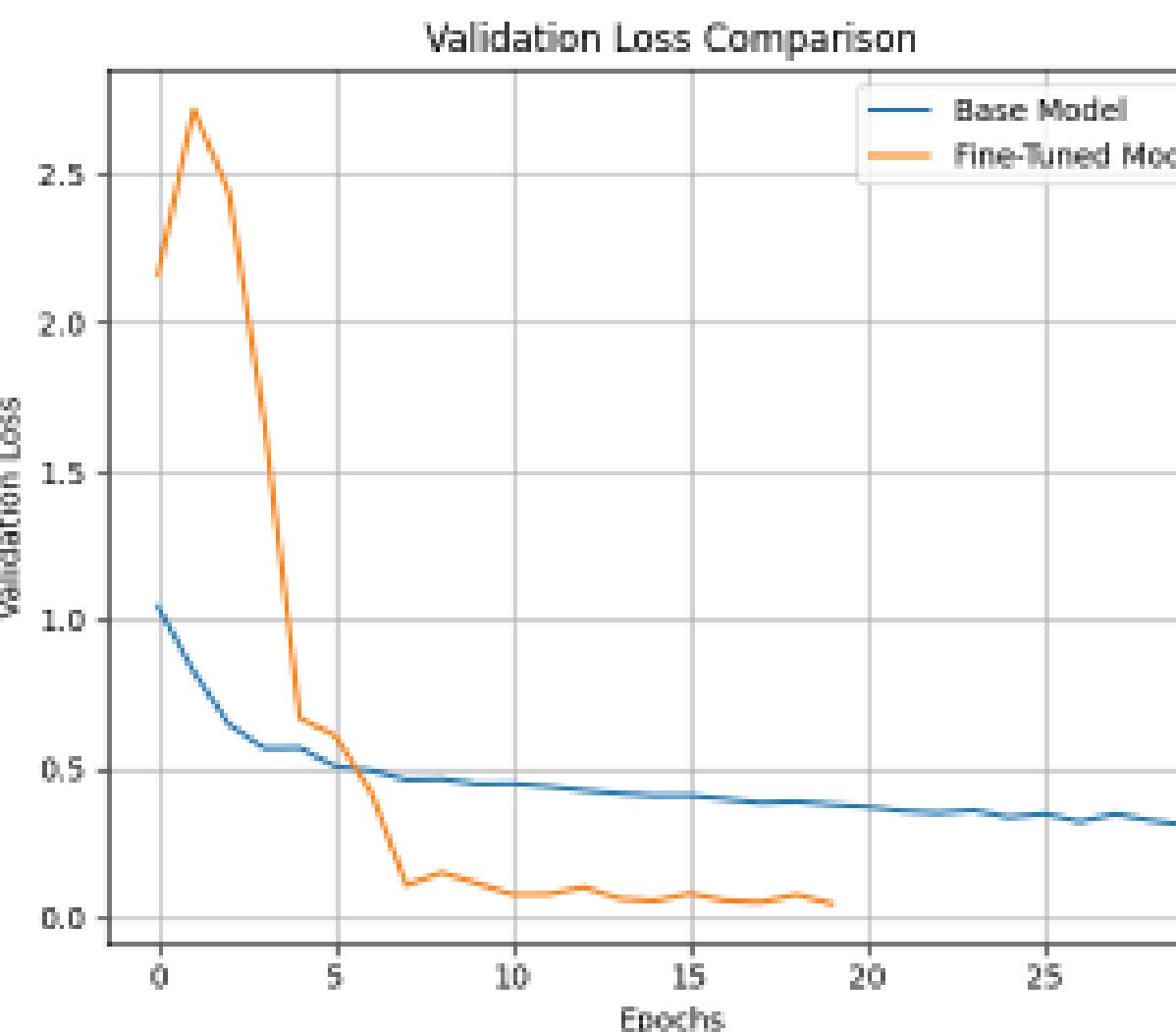
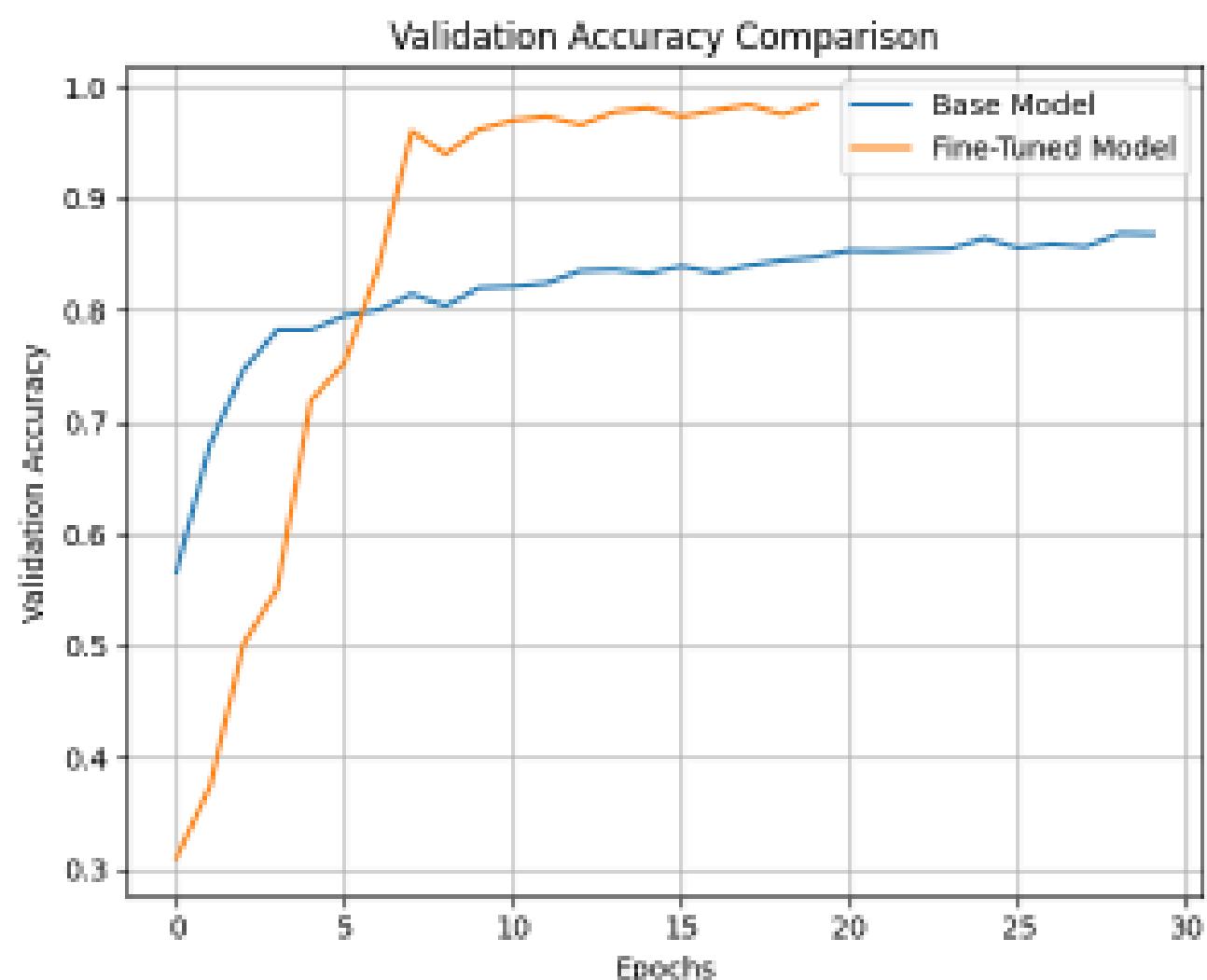
1. Classification Performance

- The fine-tuned model achieved strong per-class performance, as detailed below:

Class	Precision	Recall	F1-Score	Support
Glioma	0.97	1.00	0.98	430
Meningioma	1.00	0.95	0.97	432
No Tumor	1.00	0.99	1.00	398
Pituitary	0.97	1.00	0.98	457

- Glioma and pituitary classes were classified with near-perfect recall, minimizing **false negatives**.
- Meningioma, previously the weakest class in the base model, saw dramatic improvement in precision (from 0.75 to 1.00) and overall F1-score.
- No tumor class continued to exhibit exceptional performance, maintaining high precision and recall
- Overall Accuracy:** 98%
- Macro Average F1-Score: 0.98
- Weighted Average F1-Score: 0.98
- This near-perfect classification capability demonstrates the fine-tuned model's superior ability to differentiate between tumor types and non-tumor cases, minimizing diagnostic errors.





CONT'D

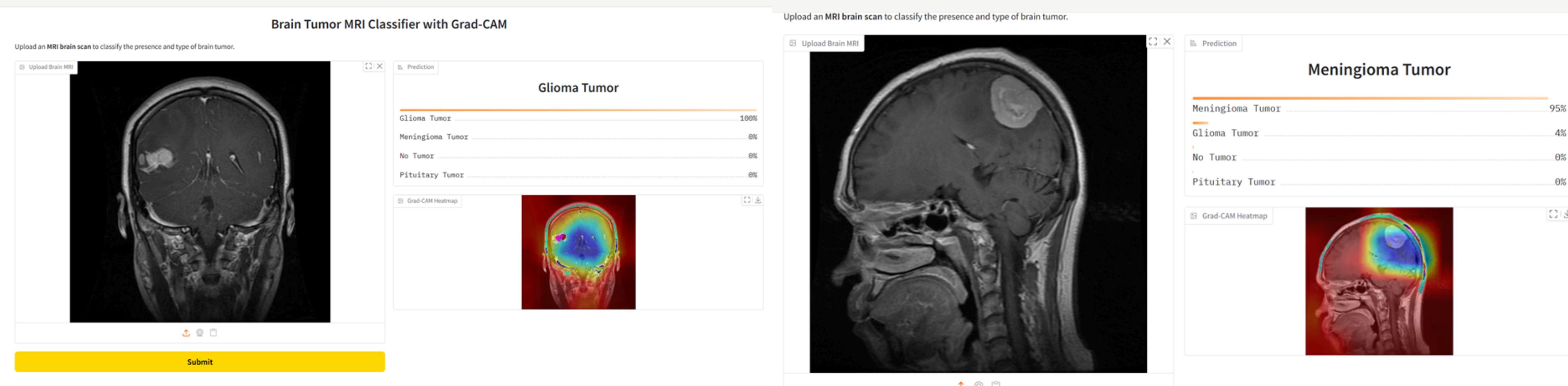
3.5. Web application

1. Image Preprocessing

- The uploaded image is resized to 224×224 pixels, converted to RGB, **normalized** to the [0, 1] range, and **reshaped** to match the model input dimensions using the `preprocess_image()` function.

2. Prediction with Confidence

- Once preprocessed, the image is passed to the trained model for prediction.
- The model outputs **class-wise confidence scores**, showing the confidence level for each of the four brain tumor classes: **glioma**, **meningioma**, **pituitary**, and **no tumor**.
- The **confidence scores** output offers a clear and interpretable summary of how confident the model is in each prediction.
- See the below samples results figures.



Brain Tumor MRI Classifier with Grad-CAM

Upload an **MRI brain scan** to classify the presence and type of brain tumor.

 Upload Brain MRI

 Drop Image Here

- or -

Click to Upload

Submit

 Prediction

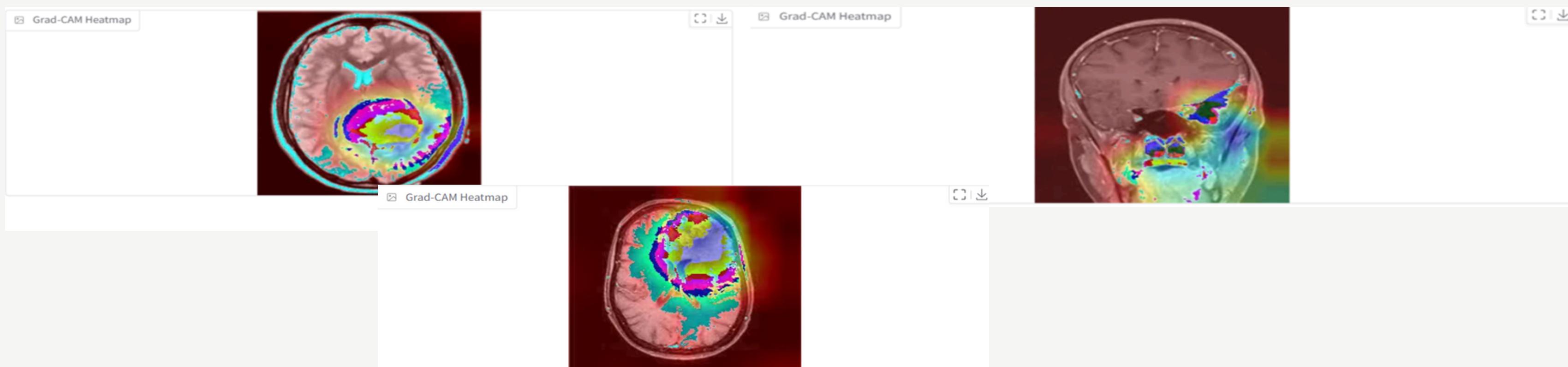
 

 Grad-CAM Heatmap

CONT'D

3. Grad-CAM Heatmap Visualization

- To enhance the interpretability of the deep learning model and verify that it focuses on the relevant tumor regions during classification, **Grad-CAM (Gradient-weighted Class Activation Mapping)** was employed.
- **Grad-CAM** is a widely used visualization technique that highlights the important regions in an input image which the CNN model considers significant for predicting a particular class.
- Grad-CAM works by:
- Computing the gradients of the predicted class score with respect to the feature maps of the **last convolutional layer**.
- Generating a heatmap reflects the spatial importance of each region in the image.
- Overlaying this **heatmap** on the original image to visually identify the tumor area influencing the model's decision.
- In this project, **Grad-CAM** was applied using the **conv5_block3_out** layer of the **fine-tuned ResNet50 model**, which is the **last convolutional block** before the classification layers.
- This layer retains enough spatial information to effectively localize relevant image features while being sufficiently deep to capture high-level semantics.
- The resulting heatmaps clearly indicated that the model concentrated on the tumor region (if present) during prediction.
- An example Grad-CAM output is shown below, where red areas correspond to regions with higher importance as per the model's internal attention.



4. DISCUSSION

4.1. Dataset

- The integration of two publicly available MRI datasets, **Msoud** and **Bilalakgz**, contributed significantly to the robustness and generalizability of the developed models.
- The combined dataset, encompassing over 8000 grayscale brain MRI images across four key classes: glioma, meningioma, pituitary tumor, and no tumor, provided sufficient diversity to train deep learning models effectively.
- The inclusion of multiple tumor types, along with non-tumorous cases, allowed for more comprehensive feature learning and improved classification accuracy.
- Additionally, the dataset's balanced representation of various tumor categories supported meaningful performance evaluation across classes.

4.2. Analysis and Evaluation of the Results

- The empirical evaluation of the base and fine-tuned ResNet50 models on the brain tumor MRI classification task revealed significant performance gains following fine-tuning.
- The base model, although pre-trained on **ImageNet**, struggled to generalize well to domain-specific medical images. It achieved a validation accuracy of 87%, with a relatively lower recall for glioma (0.74) and meningioma (0.79), indicating a tendency to misclassify these tumor types. This was further supported by its confusion matrix, which showed higher false positives and false negatives in these categories.
- In contrast, the fine-tuned model reached an achievement of 98% accuracy, with macro-averaged precision, recall, and f1-score all approaching 0.98–0.99.
- This reflects a well-balanced classifier with high predictive power across all classes.
- The training metrics support this observation, as the fine-tuned model's validation loss minimized to 0.0290 compared to the base model's 0.4288.
- The training convergence was also faster and more stable.
- From a model generalization perspective, the improvement is attributed to:
 - ▶ Domain-specific learning via fine-tuning,
 - ▶ Advanced preprocessing techniques (**Augmentation**, **CLAHE**, and **Gaussian blur sharpening**),
 - ▶ Dynamic learning rate scheduling using **ReduceLROnPlateau** and **Regularization**,
 - ▶ **Class-weight balancing** was used to address dataset imbalance.
- The Grad-CAM visualizations also showed that the fine-tuned model better localized relevant tumor regions in MRIs, enhancing interpretability and building trust and reliability of our model in brain tumor multi-class classification task.

CONT'D

4.3.Comparison with Literature

- To contextualize the effectiveness of our proposed method, we compared the classification performance of our fine-tuned ResNet50 model with several state-of-the-art approaches from recent literatures.
- Table 4.3 summarizes key characteristics of these studies, including model architecture, reported accuracy, dataset characteristics, preprocessing techniques, and additional remarks.
- While many previous works achieved high accuracy using models like VGG19, custom CNNs, Capsule Networks, or ResNet variants, they were typically limited to **three-class classification** (glioma, meningioma, and pituitary).
- Furthermore, several models either relied on minimal preprocessing or used conventional techniques like histogram equalization, which may not sufficiently enhance key features in MRI scans.
- In contrast, our model extends to a four-class classification scheme by incorporating the “no tumor” category, increasing clinical relevance and diagnostic utility.
- Additionally, we integrated domain-specific preprocessing methods, such as CLAHE and Gaussian sharpening, to better reveal tumor features in MRI images.
- This, combined with a carefully staged transfer learning and fine-tuning strategy, contributed to our model achieving an accuracy of 98%, the highest among all compared studies.
- The results align well with existing studies, reinforcing that fine-tuned CNNs, especially ResNet variants, perform exceptionally on MRI-based tumor classification tasks. Table 4.3 summarizes how this study compares with recent literature.
- **Note:** Datasets and class definitions may differ slightly between studies; direct comparison should consider dataset complexity and size.
- As shown in Table 4.3. summary, our method outperforms other models not only in terms of classification accuracy but also in interpretability and adaptability to multiclass brain tumor datasets.
- These results validate the effectiveness of domain-specific preprocessing, model fine-tuning, and explainable AI components when applied to medical imaging tasks.

Table 4.3. Comparison of Model Performance with Related Work

Study	Model Used	Accuracy (%)	Dataset Type	Preprocessing	Remarks
Sajjad et al. (2019)	Fine-Tuned VGG19	94.58	Brain Tumor MRI (3 classes)	Augmentation + <u>DWT</u> (Discrete Wavelet Transform)	Used discrete wavelet transform and transfer learning
Afshar et al. (2020)	Capsule Network	90.00–91.00	Brain Tumor MRI	Minimal	Robust to spatial variation, slow convergence
Chowdhury et al. (2021)	ResNet50 (Fine-tuned)	96.80	BraTS Dataset	Standard augmentation	Validated ResNet's performance on MRI segmentation
Hemanth et al. (2019)	CNN (custom)	93.00	Brain MRI (3 classes)	Histogram equalization	Basic architecture, less generalizable
Swati et al. (2019)	ResNet50 + SVM	96.13	CE-MRI (contrast-enhanced)	Deep features + classifier	ResNet features used with SVM for classification
This study (2025)	Fine-Tuned ResNet50	98.00	Brain MRI (4 classes)	CLAHE + Gaussian Blur	Highest accuracy, class-balanced learning and Grad-CAM

CONT'D

4.3.Comparison with Literature

- To contextualize the effectiveness of our proposed method, we compared the classification performance of our fine-tuned ResNet50 model with several state-of-the-art approaches from recent literatures.
- Table 4.3 summarizes key characteristics of these studies, including model architecture, reported accuracy, dataset characteristics, preprocessing techniques, and additional remarks.
- While many previous works achieved high accuracy using models like VGG19, custom CNNs, Capsule Networks, or ResNet variants, they were typically limited to **three-class classification** (glioma, meningioma, and pituitary).
- Furthermore, several models either relied on minimal preprocessing or used conventional techniques like histogram equalization, which may not sufficiently enhance key features in MRI scans.
- In contrast, our model extends to a four-class classification scheme by incorporating the “no tumor” category, increasing clinical relevance and diagnostic utility.
- Additionally, we integrated domain-specific preprocessing methods, such as CLAHE and Gaussian sharpening, to better reveal tumor features in MRI images.
- This, combined with a carefully staged transfer learning and fine-tuning strategy, contributed to our model achieving an accuracy of 98%, the highest among all compared studies.
- The results align well with existing studies, reinforcing that fine-tuned CNNs, especially ResNet variants, perform exceptionally on MRI-based tumor classification tasks. Table 4.3 summarizes how this study compares with recent literature.
- **Note:** Datasets and class definitions may differ slightly between studies; direct comparison should consider dataset complexity and size.
- As shown in Table 4.3. summary, our method outperforms other models not only in terms of classification accuracy but also in interpretability and adaptability to multiclass brain tumor datasets.
- These results validate the effectiveness of domain-specific preprocessing, model fine-tuning, and explainable AI components when applied to medical imaging tasks.

CONT'D

4.4. Conclusion

- This study compared the performance of a base ResNet50 model against a fine-tuned version for the classification of brain tumors from MRI images.
- The results clearly demonstrate the significant advantages of fine-tuning pre-trained models on domain-specific data.
- The base model achieved a respectable overall accuracy of 87%, with some limitations in distinguishing between certain tumor classes such as glioma and meningioma. The classification metrics reflected moderate precision and recall, indicating that the base model produced a noticeable number of false positives and false negatives.
- This limits its reliability for clinical applications, where diagnostic accuracy is critical.
- By contrast, the fine-tuned model delivered a substantial improvement, reaching an accuracy of 98%.
- It demonstrated highly consistent precision, recall, and F1-scores across all tumor classes, with minimal misclassifications as seen in its confusion matrix.
- The fine-tuned model's enhanced feature extraction and optimization through transfer learning and additional training layers contributed to this superior performance.
- Training and validation metrics also underscored the improvement: the fine-tuned model achieved a peak validation accuracy of 99.34% and maintained low validation loss, indicating excellent generalization and minimized overfitting. This was a marked advancement over the base model's peak validation accuracy of 83.02%.
- In conclusion, the fine-tuned ResNet50 model exhibits strong potential as a robust and accurate tool for brain tumor classification.
- Its improved predictive power can support radiologists and healthcare professionals by providing reliable computer-aided diagnosis, potentially accelerating early detection and improving patient outcomes.

CONT'D

4.5. Limitations, Challenges, and Future Work

4.5.1. Limitations and Challenges

- Despite the strong performance of the fine-tuned ResNet50 model, several limitations and practical challenges were observed during the study:
- **Dataset Constraints**
- Although the dataset used was comprehensive and relatively balanced, it was compiled from multiple public sources. Variations in acquisition protocols, MRI machine quality, and image annotation consistency could introduce subtle biases. Additionally, the dataset lacked detailed clinical metadata (e.g., tumor grade, patient demographics), which could enhance model precision in real-world diagnostics.
- **Limited Generalization to Real Clinical Environments**
- The model was trained on pre-cropped and pre-classified images, which may not fully represent the complexity of clinical workflows involving full brain scans, heterogeneous tumor boundaries, and comorbidities.
- **Absence of 3D MRI Analysis**
- This study focused on 2D axial slices of brain MRIs. However, brain tumors are inherently 3D in structure. Relying solely on 2D slices can limit the spatial understanding of tumor volume and its progression.
- **Model Interpretability Scope**
- While Grad-CAM provided valuable visual cues, its interpretability is limited to a single layer's perspective. More robust explainability frameworks—like LIME, SHAP, or attention-based visualization—could offer deeper insights into model behavior.
- **Computational Demands**
- Training and fine-tuning deep neural networks on medical images require significant computational resources (e.g., high-end GPUs), which may limit model reproducibility and scalability in resource-constrained settings.

CONT'D

4.5.1. Future Work

- To further enhance the utility, robustness, and clinical integration of the developed model, the following directions are proposed for future research:
- **Incorporate Full-Volume 3D MRI Analysis**
- Future models could process volumetric MRI data using 3D CNNs or hybrid 2D-3D architectures to capture richer spatial features and tumor morphology.
- **Multi-Modal Integration**
- Combining MRI with additional modalities such as CT, PET, or clinical data (age, symptoms, tumor grade) may improve classification accuracy and provide comprehensive diagnostic support.
- **Tumor Segmentation and Localization**
- Integrating tumor segmentation (e.g., via U-Net) could allow not only classification but also delineation of tumor boundaries, enabling surgical planning and volumetric analysis.
- **Cross-Dataset Validation**
- To evaluate generalizability, models should be tested across external datasets from different institutions or countries. This step is critical for regulatory approval and clinical deployment.
- **On-Device Deployment**
- Developing lightweight versions of the model using model quantization or pruning could enable real-time inference on mobile devices or edge-computing platforms in low-resource settings.
- **Explainable AI (XAI) Frameworks**
- Expanding beyond Grad-CAM to more interpretable AI techniques would improve clinician trust and facilitate transparent decision-making in diagnostic settings.
- **Clinical Trials and Expert Validation**
- Future work should involve collaboration with medical professionals to assess the tool's performance in a clinical trial setting, including sensitivity to rare tumor types or atypical MRI presentations.

Thank you

For your attention

fandisheabdurehman@gmail.com

www.ktun.edu.tr

+90-507-070-1196

