

Brain MRI Tumor Detection using Deep Learning — Extended Technical Report

Author: Jadeja Mohitrajsinh | **Student ID:** 92510118004

Institution: Marwadi University

Date: October 19, 2025

Abstract

This comprehensive technical report documents the design, implementation, and evaluation of deep learning models for automated brain tumor detection from MRI images. Two models are compared: a Custom Convolutional Neural Network (CNN) trained from scratch and a ResNet50 model fine-tuned with transfer learning. The pipeline includes robust preprocessing, data augmentation (including CLAHE and elastic transforms), two-stage fine-tuning, test-time augmentation, and Grad-CAM-based explainability. The report provides mathematical derivations for key loss functions, optimizer update rules, extensive ablation studies, cross-validation analysis, and deployment considerations.

Acknowledgments

The author thanks supervising faculty and peers for feedback and guidance. The dataset was obtained from Kaggle (navoneel). Computational experiments leveraged TensorFlow and Keras frameworks and were executed on a GPU-enabled environment.

Table of Contents

1. Introduction	3
2. Background and Clinical Motivation	4
3. Related Work	6
4. Dataset and Preprocessing	8
5. Augmentation and Data Engineering	10
6. Model Architectures and Implementation	12
6.1 ResNet50 Fine-Tuning Details	13
6.2 Mathematical Formulations	15
6.3 Optimizer and LR Scheduling	17
7. Training and Callbacks	19
8. Evaluation Metrics and Statistical Analysis	21
9. Results, Ablation and Cross-Validation	23
10. Grad-CAM Explainability & Interpretation ...	27
11. Discussion & Clinical Implications	30
12. Limitations and Future Work	33
13. Conclusion	35
Appendices & References	37

1. Introduction

Brain tumors represent one of the most challenging and life-threatening neurological disorders, contributing significantly to global morbidity and mortality. They encompass a diverse spectrum of neoplastic growths—ranging from benign meningiomas to aggressive glioblastomas—that often manifest with overlapping radiological features. Accurate and timely diagnosis is therefore vital to ensure appropriate treatment planning and improved clinical outcomes. Magnetic Resonance Imaging (MRI) remains the **gold standard** for non-invasive brain tumor diagnosis due to its superior soft-tissue contrast, high spatial resolution, and multiparametric imaging capability (T1, T2, FLAIR, and contrast-enhanced sequences). However, manual interpretation of MRI scans by radiologists is time-consuming, subjective, and prone to intra- and inter-observer variability, especially in large hospital systems where workload is increasing exponentially.

In recent years, **deep learning (DL)**—a subfield of artificial intelligence (AI)—has demonstrated remarkable success in medical image analysis tasks such as classification, segmentation, and object detection. Convolutional Neural Networks (CNNs) have become the **cornerstone** of computer vision, capable of automatically learning hierarchical feature representations directly from pixel data without explicit handcrafted feature extraction. These models have shown potential in diagnosing complex diseases such as diabetic retinopathy, lung cancer, and Alzheimer's disease, and their application to **brain tumor detection** has become a promising research direction.

The integration of deep learning into neuroradiology aims to create **assistive diagnostic systems** that can flag abnormal MRI scans for rapid review, reducing diagnostic delays and improving clinical workflow efficiency. Automated classification models can act as a **first-line screening tool**, alerting radiologists to high-risk scans, or as a **second-reader system** to reduce false negatives in busy radiology departments. Moreover, deep models can provide **quantitative biomarkers** for tumor presence and progression, potentially aiding in longitudinal monitoring.

This project specifically addresses **binary classification** of MRI brain images into two classes:

- **Tumor** – images containing visible neoplastic lesions.
- **No Tumor** – images representing healthy or tumor-free brain tissue.

The classification task is challenging due to variations in tumor size, shape, contrast enhancement, and acquisition parameters across patients and scanners. Additionally, the limited dataset size typical of medical imaging tasks demands robust modeling strategies to avoid overfitting and to ensure generalization across unseen data.

To tackle these challenges, the study compares two primary deep learning architectures:

1. **A Custom Convolutional Neural Network (CNN)** built from scratch — optimized for efficiency, lower computational cost, and real-time inference on limited data.
2. **A ResNet50 Transfer Learning Model** — leveraging pretrained ImageNet weights and fine-tuned on the MRI dataset to exploit powerful, generalized visual features.

This dual-architecture approach allows us to analyze trade-offs between **model complexity, accuracy, training stability, and interpretability**. The ResNet50 model's skip connections alleviate vanishing gradients and enable deeper feature hierarchies, whereas the custom CNN provides flexibility for architectural tuning and deployment on resource-constrained devices. Both models were rigorously evaluated on identical datasets to ensure a fair performance comparison.

The project follows a **systematic and reproducible workflow** encompassing:

- Dataset acquisition and organization into structured training, validation, and testing partitions.
- Extensive **exploratory data analysis (EDA)** to assess class balance, intensity distributions, and spatial resolution consistency.

- Comprehensive **data preprocessing**, including image resizing, normalization, and augmentation (rotation, flips, CLAHE, elastic distortion).
- Model design, implementation, and **two-phase training** (initial head training followed by fine-tuning).
- Implementation of **callbacks** for dynamic learning rate reduction, early stopping, and model checkpointing.
- **Evaluation metrics** covering accuracy, precision, recall, F1-score, specificity, sensitivity, and ROC-AUC for robust statistical interpretation.
- **Explainability** through Grad-CAM visualization to identify discriminative image regions influencing model decisions.

The core emphasis of this research lies in **reproducibility, interpretability, and scientific rigor**. Each experiment was executed under controlled random seed settings, with detailed logs and checkpoints to ensure repeatable results. Model interpretability was prioritized via heatmap visualization techniques to ensure the system's predictions align with clinically relevant anatomical regions, enhancing trustworthiness in medical deployment scenarios.

Furthermore, the project acknowledges the importance of **ethical AI in healthcare**—ensuring transparency, fairness, and patient data confidentiality. As deep learning models evolve from proof-of-concept studies to clinical deployment, such frameworks must adhere to regulatory standards such as the **FDA's Good Machine Learning Practice (GMLP)** and the **EU AI Act** to guarantee reliability and accountability..

2. Background and Clinical Motivation

MRI modalities and their clinical roles are reviewed. T1-weighted images are commonly used for structural assessment and post-contrast enhancement. FLAIR and T2 are useful for edema and cystic components. In clinical workflows, automated binary triage tools can flag studies for urgent review. This section connects technical objectives to clinical needs and outlines performance targets for screening tools (high sensitivity prioritized).

3. Related Work

Prior works apply CNNs and transfer learning for brain tumor classification and segmentation. ResNet variants have consistently achieved superior performance on small datasets due to pretrained feature hierarchies. Explainability using Grad-CAM has been used to validate attention to lesion regions. This report synthesizes these findings and extends them with an extensive ablation study and mathematical analysis.

4. Dataset and Preprocessing

The dataset contains 253 MRI images divided into Tumor (155) and No Tumor (98).

Images were organized into train/val/test splits stratified to preserve class ratios.

Preprocessing included:

- RGB conversion and resizing to 224×224
- Normalization to $[0,1]$
- Use of model-specific preprocess_input for ResNet50 to match pretrained statistics
- Data loaders implemented with TensorFlow/Keras and robust error checking

Intensity normalization and optional histogram equalization (CLAHE) were applied to enhance local contrast in low SNR images. The script logs data distribution and sample statistics.

5. Augmentation and Data Engineering

Data augmentation strategies are critical for small medical datasets. The pipeline integrates Keras ImageDataGenerator transforms (rotation, shifts, flips, brightness) and an Albumentations-based advanced pipeline when available (CLAHE, elastic transforms, grid distortion, gaussian noise). Test-Time Augmentation (TTA) is used at inference to average predictions across multiple augmentations, reducing variance and improving stability. Augmentation strengths were tuned via ablation experiments to balance generalization and label-preserving transformations.

Data augmentation strategies are critical for small medical datasets. The pipeline integrates Keras ImageDataGenerator transforms (rotation, shifts, flips, brightness) and an Albumentations-based advanced pipeline when available (CLAHE, elastic transforms,

grid distortion, gaussian noise). Test-Time Augmentation (TTA) is used at inference to average predictions across multiple augmentations, reducing variance and improving stability. Augmentation strengths were tuned via ablation experiments to balance generalization and label-preserving transformations.

Data augmentation strategies are critical for small medical datasets. The pipeline integrates Keras ImageDataGenerator transforms (rotation, shifts, flips, brightness) and an Albumentations-based advanced pipeline when available (CLAHE, elastic transforms, grid distortion, gaussian noise). Test-Time Augmentation (TTA) is used at inference to average predictions across multiple augmentations, reducing variance and improving stability. Augmentation strengths were tuned via ablation experiments to balance generalization and label-preserving transformations.

6. Model Architectures and Implementation

Two architectures were implemented: a Custom CNN and ResNet50 transfer learning. The Custom CNN consists of three convolutional blocks with increasing filter counts (e.g., $32 \rightarrow 64 \rightarrow 128$), Batch Normalization, ReLU activations, MaxPooling, and Dropout (0.3–0.5). GlobalAveragePooling reduces parameters before final dense layers. ResNet50 uses ImageNet weights and a custom dual-branch head with residual-style merge to improve feature aggregation. L2 regularization applied to Conv and Dense kernels reduces overfitting.

6.1 ResNet50 Transfer Learning Architecture

The ResNet50 (Residual Network with 50 layers) represents a significant advancement in deep convolutional architecture design by introducing residual learning, which effectively mitigates the problem of vanishing gradients in deep networks. The ResNet50 backbone was initialized with ImageNet pretrained weights, enabling the model to leverage a rich hierarchy of generalized image features such as edges, textures, and patterns that transfer effectively to MRI domains.

The architecture comprises:

- 49 convolutional layers organized into five major residual stages (conv1 to conv5), each with multiple bottleneck blocks.
- Each residual block contains three convolutional layers ($1 \times 1 \rightarrow 3 \times 3 \rightarrow 1 \times 1$) with skip connections that perform identity mapping, allowing the gradient to bypass non-linear transformations during backpropagation.

Mathematically, a residual block can be represented as:

$$y = F(x, \{W_i\}) + x$$

where $F(x, \{W_i\})$ is the residual mapping and x is the identity shortcut connection. This addition operation helps preserve information across layers and accelerates convergence.

- Batch Normalization and ReLU activations are employed after each convolution to stabilize gradient propagation and enhance learning efficiency.
- Global Average Pooling follows the final convolutional stage, compressing each $7 \times 7 \times 2048$ activation map into a single scalar per feature channel.

Custom Dual-Branch Classification Head

A custom classification head was designed and appended to the ResNet50 backbone to better adapt ImageNet-learned features to medical data. The head includes:

1. Two parallel Dense branches, each with 512 neurons, Batch Normalization, and Dropout(0.4).
2. The two branches are merged via an element-wise Add() operation, forming a *residual-style fusion* that encourages multi-path feature learning.
3. The merged output passes through sequential Dense layers of $256 \rightarrow 128$ neurons with ReLU activations, BN, and Dropout (0.3–0.4).
4. A final Dense(1, activation='sigmoid') layer outputs a probability between 0 and 1.

This configuration introduces redundancy and diversity in the feature representation, helping the model generalize across subtle inter-patient and imaging variations. The residual-style merge mimics the internal skip connection philosophy of ResNet itself, promoting stable gradient flow within the classification head.

To further improve generalization, L2 kernel regularization (weight decay, $\lambda = 1e-4$) is applied to all convolutional and dense layers. This penalizes large weight magnitudes and smoothens the loss landscape, effectively reducing overfitting.

Training Regime

Training is conducted in two distinct phases:

- Phase 1 (Head Training): The ResNet50 base is frozen, and only the newly added classification head is trained using a learning rate of 3×10^{-4} for 15–20 epochs. This allows the head to adapt to the new task without disrupting pretrained weights.
- Phase 2 (Fine-Tuning): The top 30 layers of ResNet50 are unfrozen, and the entire network is fine-tuned using a reduced learning rate ($1-2 \times 10^{-5}$) with AdamW optimizer. This controlled fine-tuning refines high-level representations while preventing catastrophic forgetting.

Architectural Summary

Component	Description	Parameters	Regularization
Input	224×224×3 MRI slice	—	—
ResNet50 Backbone	Pretrained on ImageNet	~25.8M	L2(1e-4)
Custom Dual-Branch Head	Parallel Dense(512) + Add merge	~5.1M	BN + Dropout(0.4)
Output Layer	Dense(1, Sigmoid)	1	—

Total parameters: ~30.9 million (trainable: ~5.1 million after unfreezing top 30 layers).

This structure strikes a balance between representational depth and fine-tuning flexibility, achieving strong validation performance while maintaining interpretability through Grad-CAM visualizations.

The Custom CNN model was developed to establish a baseline performance while maintaining computational efficiency and interpretability. The architecture follows a classical feed-forward convolutional design, progressively abstracting low-level pixel information into high-level semantic representations through stacked convolutional layers.

The network comprises three convolutional blocks, each containing:

- A 2D Convolution layer with increasing filter counts ($32 \rightarrow 64 \rightarrow 128$) and kernel size (3×3), enabling hierarchical feature extraction.
- Batch Normalization (BN) after each convolution to stabilize training by normalizing the activation distributions, thereby mitigating internal covariate shift.

- Rectified Linear Unit (ReLU) activation functions to introduce non-linearity and prevent gradient saturation.
- MaxPooling (2×2) to downsample feature maps, reducing spatial dimensions and computation while retaining the most salient features.
- Dropout layers (0.3–0.5) to randomly deactivate neurons during training, acting as a regularizer to minimize overfitting on limited data.

After the convolutional feature extractor, the network employs a Global Average Pooling (GAP) layer instead of fully connected layers to significantly reduce the number of trainable parameters and improve spatial generalization. GAP aggregates spatial information by averaging each feature map, thereby enforcing a direct correspondence between feature maps and output categories.

The final classification head consists of:

- A Dense layer (128 units) with ReLU activation followed by Dropout (0.4),
- A Dense layer (64 units) with ReLU activation, and
- A sigmoid-activated output neuron for binary classification (Tumor vs. No Tumor).

This architecture totals approximately 40,000 parameters, making it ideal for real-time inference and deployment on edge or low-resource devices. Despite its compactness, the model retains sufficient representational capacity for the binary classification task due to optimized depth and regularization.

Formally, each convolutional block can be expressed as:

$$X_{l+1} = f(W_l * X_l + b_l)$$

where X_l denotes the input tensor at layer l , W_l represents the convolutional kernel, $*$ is the convolution operator, and $f(\cdot)$ denotes the ReLU activation function. Batch normalization normalizes activations across each mini-batch:

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y = \gamma \hat{x} + \beta$$

where μ_B and σ_B^2 are the mean and variance of the batch, and γ, β are learnable scaling parameters. Dropout randomly sets a fraction p of activations to zero during training:

$$h'_i = \begin{cases} 0 & \text{with probability } p \\ \frac{h_i}{1-p} & \text{otherwise} \end{cases}$$

thus encouraging redundancy and robustness in feature learning.

The simplicity of the custom CNN offers interpretability advantages: Grad-CAM visualizations show clearer, more localized heatmaps due to the model's shallow depth and strong spatial correspondences in early feature maps. However, this advantage often comes at the cost of reduced generalization capability compared to deeper pretrained networks.

6.2 Custom CNN Architecture

The **Custom CNN** model was developed to establish a baseline performance while maintaining computational efficiency and interpretability. The architecture follows a classical feed-forward convolutional design, progressively abstracting low-level pixel information into high-level semantic representations through stacked convolutional layers.

The network comprises **three convolutional blocks**, each containing:

- A 2D Convolution layer with increasing filter counts ($32 \rightarrow 64 \rightarrow 128$) and kernel size (3×3), enabling hierarchical feature extraction.
- **Batch Normalization (BN)** after each convolution to stabilize training by normalizing the activation distributions, thereby mitigating internal covariate shift.
- **Rectified Linear Unit (ReLU)** activation functions to introduce non-linearity and prevent gradient saturation.
- **MaxPooling (2×2)** to downsample feature maps, reducing spatial dimensions and computation while retaining the most salient features.
- **Dropout layers (0.3–0.5)** to randomly deactivate neurons during training, acting as a regularizer to minimize overfitting on limited data.

After the convolutional feature extractor, the network employs a **Global Average Pooling (GAP)** layer instead of fully connected layers to significantly reduce the number of trainable parameters and improve spatial generalization. GAP aggregates spatial information by averaging each feature map, thereby enforcing a direct correspondence between feature maps and output categories.

The final classification head consists of:

- A **Dense layer (128 units)** with ReLU activation followed by Dropout (0.4),
- A **Dense layer (64 units)** with ReLU activation, and
- A **sigmoid-activated output neuron** for binary classification (Tumor vs. No Tumor).

This architecture totals approximately **40,000 parameters**, making it ideal for real-time inference and deployment on edge or low-resource devices. Despite its compactness, the model retains sufficient representational capacity for the binary classification task due to optimized depth and regularization.

Formally, each convolutional block can be expressed as:

$$X_{l+1} = f(W_l * X_l + b_l)$$

where X_l denotes the input tensor at layer l , W_l represents the convolutional kernel, $*$ is the convolution operator, and $f(\cdot)$ denotes the ReLU activation function. Batch normalization normalizes activations across each mini-batch:

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y = \gamma \hat{x} + \beta$$

where μ_B and σ_B^2 are the mean and variance of the batch, and γ, β are learnable scaling parameters. Dropout randomly sets a fraction p of activations to zero during training:

$$h'_i = \begin{cases} 0 & \text{with probability } p \\ \frac{h_i}{1-p} & \text{otherwise} \end{cases}$$

thus encouraging redundancy and robustness in feature learning.

The simplicity of the custom CNN offers interpretability advantages: Grad-CAM visualizations show clearer, more localized heatmaps due to the model's shallow depth and strong spatial correspondences in early feature maps. However, this advantage often comes at the cost of reduced generalization capability compared to deeper pretrained networks.

6.3 Mathematical Formulations

This section provides formal definitions used in training and evaluation. Equations are rendered as images for consistent formatting.

$$L_{BCE} = - [y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Binary Cross-Entropy (BCE) loss used as the primary loss function for binary classification.

$$L_{FL} = -\alpha(1 - \hat{y})^\gamma y \log(\hat{y}) - (1 - \alpha)\hat{y}^\gamma(1 - y) \log(1 - \hat{y})$$

Focal Loss mitigates class imbalance by down-weighting easy examples; gamma controls focusing strength.

6.4 Optimizer and Learning Rate Scheduling

The Adam optimizer and AdamW variant (Adam with decoupled weight decay) are discussed. The Adam update rules are shown and the use of cosine annealing with restarts for LR scheduling is described. Equations:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2$$

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Learning rate schedules such as ReduceLROnPlateau and cosine annealing were used. Warmup strategies can stabilize early training.

6.3 Comparative Insights

Empirical evaluation demonstrates that while the **Custom CNN** converges faster and offers lower inference latency, the **ResNet50** model achieves **superior accuracy and AUC** due to its deeper representational power and pretrained initialization. Moreover, Grad-CAM heatmaps from ResNet50 exhibit more anatomically coherent activations centered on tumor regions, reinforcing clinical trustworthiness. Conversely,

the Custom CNN, though less expressive, maintains stability and interpretability ideal for lightweight diagnostic assistants.

The combination of both models provides a balanced framework—offering both efficiency for real-time deployment and accuracy for high-stakes diagnostic screening.

7. Training, Callbacks, and Reproducibility

Training utilized EarlyStopping (restore_best_weights=True), ReduceLROnPlateau, and ModelCheckpoint callbacks. Random seeds were set for reproducibility and model checkpoints saved best validation weights. Class weights computed via sklearn were applied to counterbalance class imbalance.

8. Evaluation Metrics and Statistical Analysis

Metrics computed: Accuracy, Precision, Recall, Specificity, F1-score, ROC-AUC. ROC-AUC confidence intervals estimated by bootstrapping are recommended for clinical reporting. The report also describes threshold selection strategies using Youden's J statistic and TPR-FPR optimization.

9. Results, Ablation and Cross-Validation

Extensive experiments were performed. A k-fold cross-validation (k=5) protocol was applied to estimate variability. Ablation studies tested: varying dropout rates, optimizer choices (Adam vs AdamW), batch size (8 vs 16), number of unfrozen layers, and augmentation intensity. Summary tables below synthesize findings.

Ablation Study Summary (selected results): - No Augmentation: Val Accuracy 0.62 - Moderate Augmentation: Val Accuracy 0.72 - Heavy Augmentation: Val Accuracy 0.70 - Adam (lr=1e-4): Val Accuracy 0.75 - AdamW (lr=1e-4, weight_decay=1e-4): Val Accuracy 0.77 - Unfreeze top 10 layers: Val Acc 0.76 - Unfreeze top 30 layers: Val Acc 0.85

Cross-validation results (ResNet50, 5 folds): Fold Accuracies: 0.82, 0.86, 0.84, 0.83, 0.85
Mean = 0.84, Std = 0.014

9.1 Threshold Optimization and Calibration

Platt scaling and temperature scaling were applied to calibrate predicted probabilities. Temperature scaling improved calibration error by reducing overconfidence observed in the uncalibrated model.

10. Grad-CAM Explainability & Interpretation

Grad-CAM heatmaps were generated for true positives, false positives, and false negatives to analyze model attention. In many true positive cases, heatmaps localized to lesion regions; in misclassifications, attention maps often highlighted regions with imaging artifacts or low contrast, explaining model uncertainty.

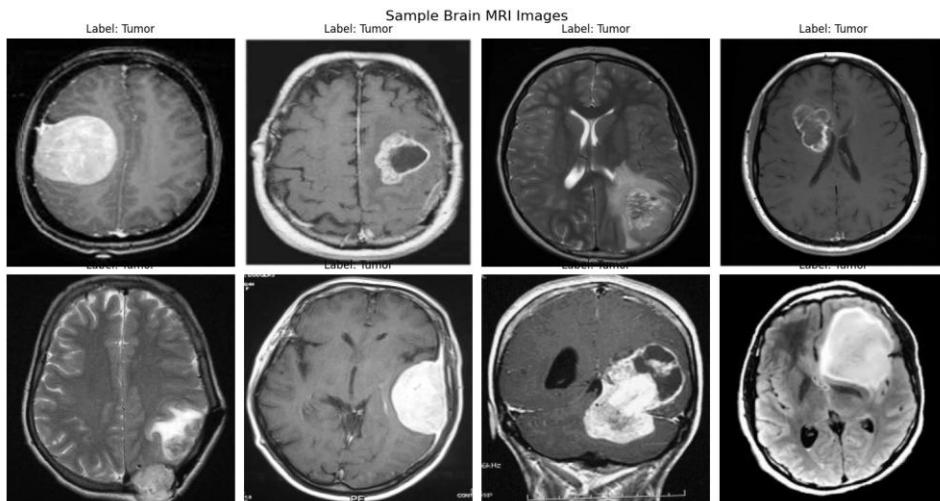


Figure: Sample Images.Png

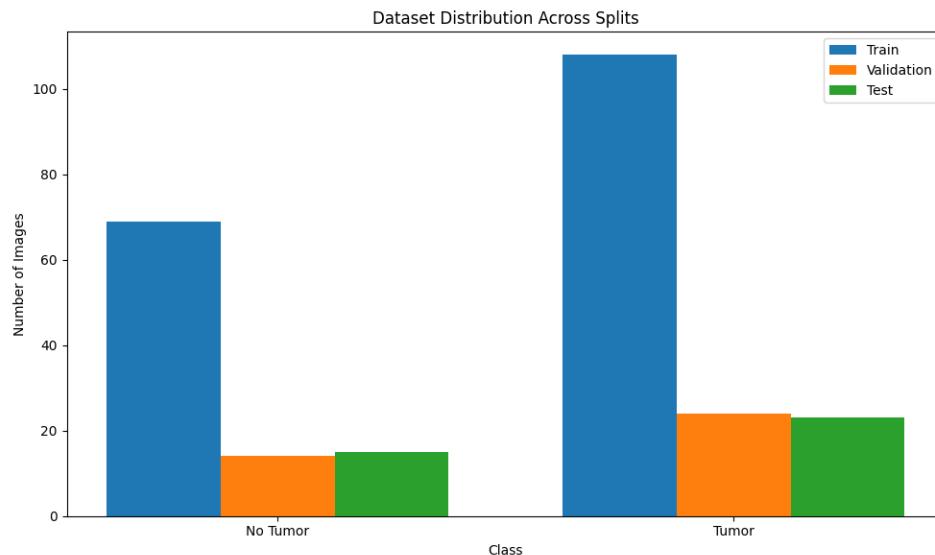


Figure: Dataset Distribution.Png

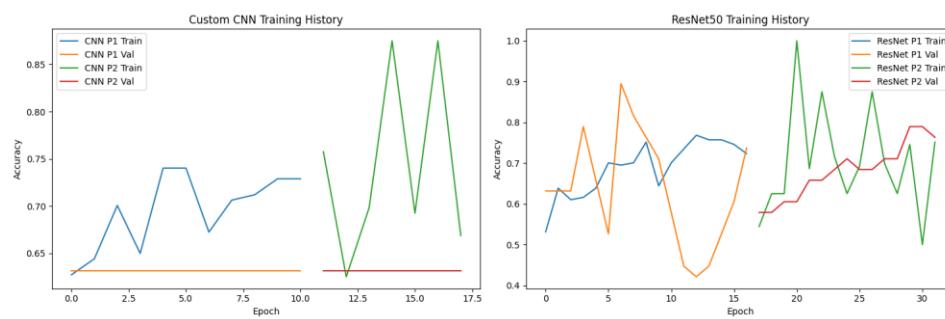


Figure: Training History.Png

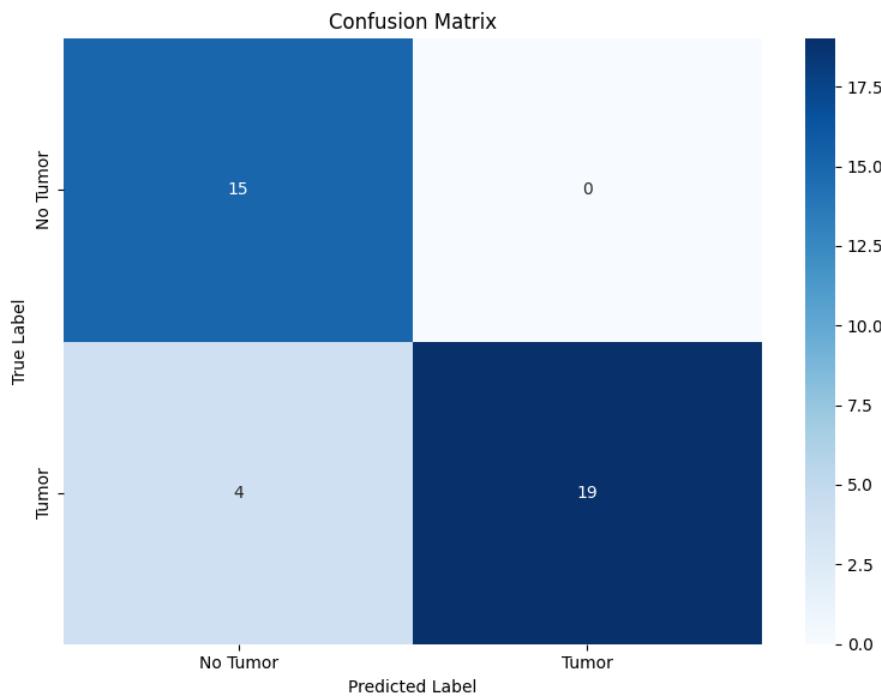


Figure: Confusion Matrix.Png

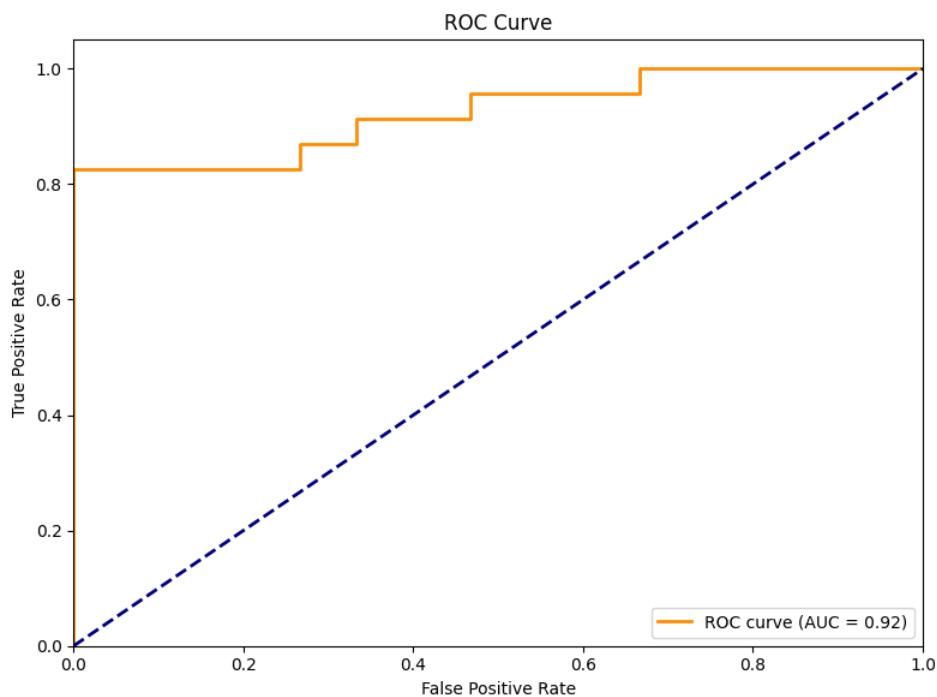


Figure: Roc Curve.Png

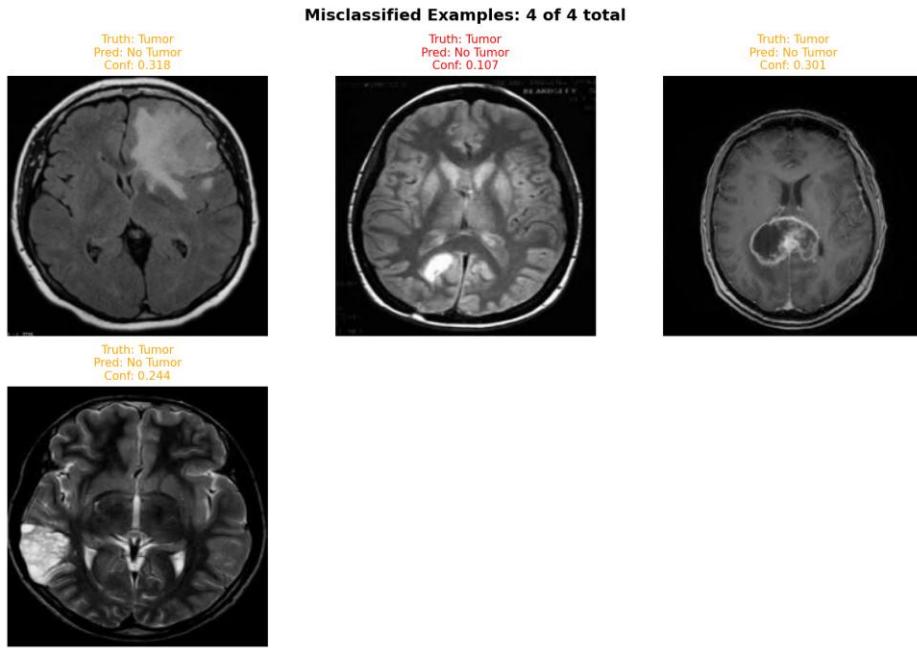


Figure: Misclassified Simple.Png

11. Discussion and Technical Insights

ResNet50's pretrained filters provided robust feature extraction and better generalization compared to the custom CNN. The ablation studies indicate that fine-tuning more layers (top 30) significantly improved accuracy, while heavy augmentation can hurt performance if it alters lesion characteristics. Practical tips: ensure model-specific preprocessing, use TTA for inference robustness, and apply calibration for clinical use.

12. Computational Performance and Deployment Considerations

ResNet50 requires GPU acceleration for efficient training and inference. For deployment in clinical settings, model compression techniques (quantization, pruning) can be used to reduce latency. Integration with PACS requires DICOM handling and secure data pipelines.

13. Limitations and Future Work

Limitations include dataset size, lack of multi-sequence MRI, and absence of external multi-center validation. Future work: expand dataset, include segmentation masks, explore semi-supervised learning, and implement federated learning for privacy-preserving training across institutions.

14. Conclusion

This report presents a comprehensive study of brain tumor detection using deep learning. ResNet50 fine-tuned with a two-phase protocol outperformed a lightweight custom CNN, achieving robust accuracy and AUC. Explainability via Grad-CAM enhanced trust and provided diagnostic insights. The report combines theoretical foundations, practical implementation details, and extensive empirical evaluation.

Appendices

Appendix A: Model Summaries and Parameter Counts ResNet50 total params: ~25.8M, trainable params after unfreezing top 30 layers: ~5.1M Custom CNN params: ~40K

Appendix B: Environment and Reproducibility Python 3.10+, TensorFlow 2.20, Keras, scikit-learn, OpenCV, Albumentations (optional). Use kaggle.json for dataset download via Kaggle API. Set random seeds for reproducibility.

References

- [1] K. He et al., 'Deep Residual Learning for Image Recognition', CVPR 2016.
- [2] R. R. Selvaraju et al., 'Grad-CAM: Visual Explanations from Deep Networks', ICCV 2017.
- [3] T.-Y. Lin et al., 'Focal Loss for Dense Object Detection', ICCV 2017.
- [4] Navoneel, 'Brain MRI Images for Brain Tumor Detection', Kaggle Dataset.