Brain Tumour Classification (MRI) Using Neural Networks

COMPSCI 4442 001: Final Project Report

GitHub Repository: https://github.com/chiggi24/cs4442_final_project

Live Deployment (NeuroScanAI): https://neuroscanai-427956346530.us-central1.run.app/

Bryson Crook
251217987
Faculty of Engineering
Western University
London Ontario, Canada
bcrook4@uwo.ca

Christopher Higgins
251245390
Faculty of Engineering
Western University
London Ontario, Canada
chiggi24@uwo.ca

Mohamed El Dogdog
251239204
Faculty of Engineering
Western University
London Ontario, Canada
meldogdo@uwo.ca

***Abstract*—This study evaluates the classification performance of four pre-trained convolutional neural network (CNN) models on brain tumour magnetic resonance imaging (MRI) scans. The dataset consists of grayscale images labeled into four categories: Glioma, Meningioma, Pituitary, and None. The pipeline includes preprocessing, data augmentation, and five-fold cross-validation. Model performance is assessed using accuracy, precision, recall, F1-score, and confusion matrices. Results show that ResNet50 achieves the highest overall F1-score and test generalization, making it the most reliable architecture for brain tumour classification. This study demonstrates the effectiveness of transfer learning in medical image analysis and supports the use of deep CNNs for automated diagnostic support.**

Keywords—Convolutional Neural Networks, Brain Tumour Classification, Transfer Learning, MRI, Deep Learning, Medical Image Analysis, K-Fold Cross-Validation.

I. Introduction

Brain tumours are among the most severe and life-threatening neurological conditions, with timely and accurate diagnosis playing a crucial role in improving survival rates. Magnetic Resonance Imaging (MRI) is the standard imaging modality used for tumour detection due to its high-resolution, non-invasive imaging capabilities [1]. However, interpretation of these scans remains highly dependent on expert radiologists, which presents significant challenges in regions with limited access to specialized care.

Deep learning techniques, particularly Convolutional Neural Networks (CNNs), have demonstrated significant success in automating image classification tasks, including applications in medical imaging [2]. While numerous studies have explored CNNs for brain tumour classification, few provide a comparative evaluation of multiple architectures using a consistent training pipeline and deployment environment.

In this work, we propose NeuroScanAI, a deep learning-powered tool for automatic brain tumour classification from MRI scans. We investigated the performance of four pre-trained CNN models — ResNet50, VGG16, EfficientNetB0, and InceptionV3 — trained and validated using K-fold cross-validation on a curated dataset of grayscale MRI scans converted to RGB [3] (see Fig. 1 for class examples). Each model is evaluated using precision, recall, F1-score, accuracy, and ROC-AUC metrics.

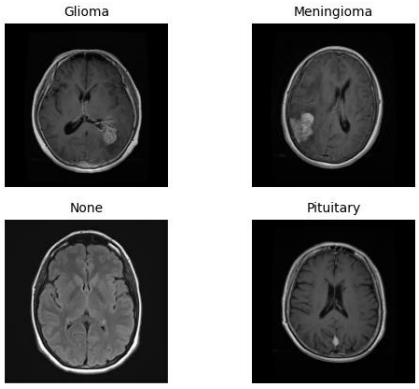

Fig. 1: Sample MRI scans from each tumour class [3].

The full workflow, including preprocessing, augmentation, training, evaluation, and deployment, is shown in Fig. 2. Our results show that ResNet50 consistently achieves the highest classification performance, while InceptionV3 underperforms in comparison.

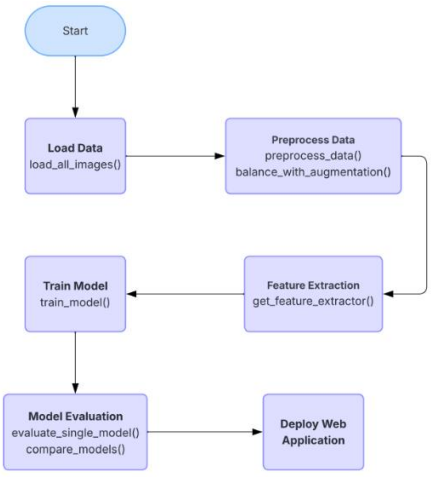

Fig. 2: NeuroScanAI model workflow and deployment pipeline.

To make this work accessible and practical, we deployed the trained models to a live web interface using Google Cloud Run. Users can upload MRI scans, select a model of their choice, and receive predicted tumour classes with associated confidence scores.

The rest of this report is structured as follows: Section II discusses background and related work. Section III describes the methodology. Section IV presents experimental results. Section V discusses findings and limitations. Section VI concludes the study.

II. Background and Related Work

A. *Convolutional Neural Networks*

Convolutional Neural Networks (CNNs) are a class of deep learning architectures specifically designed for visual data, particularly effective for image classification and analysis tasks [4]. Their structure mimics the human visual cortex, using hierarchical layers to extract spatial hierarchies of features from input images. The architecture typically consists of the following components:

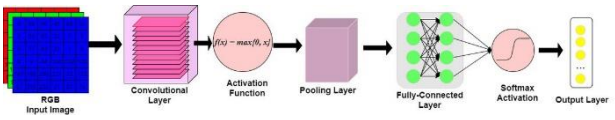

Fig. 3: Convolutional Neural Network (CNN) Architecture [4].

At the core of a CNN is the convolutional layer, which "applies filters to the input data, generating feature maps through convolutions" [5] These filters "slide over the input data, capturing local patterns and features" [5]. The convolution operation between an input image $I \in \mathbb{R}^{H\times W\times C}$ and a kernel $K \in \mathbb{R}^{k\times k\times C}$ produces an output feature map $F \in \mathbb{R}^{H'\times W'}$, computed as:

$$F(i,j) = \sum_{m=0}^{k-1}\sum_{n=0}^{k-1}\sum_{c=0}^{C-1} I(i+m, j+n, c)\cdot K(m,n,c) \tag{1}$$

Each kernel detects specific patterns (e.g., edges, blobs, textures), and multiple filters allow the network to learn diverse features.

Following the convolution operation, activation functions such as ReLU ($f(x) = max(0,x)$) introduce non-linearity, enabling the model to approximate complex mappings. According to De Luca, ReLU is preferred in CNNs because it is "very simple to calculate, as it involves only a comparison between its input and the value 0," and it has a derivative that is "either 0 or 1," which makes gradient computation efficient during backpropagation [6]. Pooling layers (e.g., max-pooling or average pooling) reduce spatial dimensions while retaining dominant features, improving computational efficiency and helping reduce overfitting.

Later in the network, fully connected layers combine the learned spatial features to perform classification. The final layer typically uses the softmax function to output class probabilities:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad i = 1, \dots, K \tag{2}$$

where z_i is the logit for class i, and K is the total number of classes (in our case, 4: Glioma, Meningioma, Pituitary, None) [7].

The entire network is optimized via backpropagation, using a loss function such as categorical cross-entropy:

$$\mathcal{L} = -\sum_{i=1}^{K} y_i \, log(\hat{y}_i) \tag{3}$$

where $y_i \, log(\hat{y}_i)$ is the ground truth label (one-hot encoded) and $\hat{y}_i$ is the predicted probability for class i. As stated in the article, "The choice of loss function is imperative to the network's performance because eventually the parameters in the network are going to be set such that the loss is minimized" [8].

In medical imaging, CNNs have proven particularly useful due to their ability to automatically extract meaningful features from complex patterns such as tumours in MRI scans — reducing reliance on handcrafted features and radiologist expertise.

B. *Transfer Learning for Medical Imaging*

Transfer learning is a deep learning strategy where a model developed for one task is reused as the starting point for another related task. This approach is especially advantageous in domains such as medical imaging, where acquiring large, labeled datasets is costly or infeasible [9].

Let us formally define transfer learning in the context of classification. Suppose we have a source domain $\mathcal{D}_S = \{\mathcal{X}_S, P_S(X)\}$ with a source task $\mathcal{T}_S = \{\mathcal{Y}_S, f_S(\cdot)\}$, and a target domain $\mathcal{D}_\mathcal{T} = \{\mathcal{X}_\mathcal{T}, P_T(X)\}$ with a target task $\mathcal{T}_\mathcal{T} = \{\mathcal{Y}_\mathcal{T}, f_T(\cdot)\}$. The goal is to improve the performance of $f_T(\cdot)$ in $\mathcal{D}_\mathcal{T}$ using knowledge learned from $\mathcal{D}_S$, even when $P_S(X) \neq P_T(X)$ or $\mathcal{Y}_S \neq \mathcal{Y}_\mathcal{T}$ [9].

In our case, we leverage convolutional layers from models pre-trained on ImageNet—a large-scale natural image dataset—and adapt them to classify brain tumour MRI scans. The pre-trained models (e.g., ResNet50, VGG16) contain generic low-level feature detectors (e.g., edges, blobs) in their early layers, which are transferable to medical contexts [10, 11]. We freeze these base layers to retain previously learned representations and fine-tune the later layers by adding new fully connected layers specific to our tumour classification task. This process is mathematically equivalent to:

$$\hat{y} = f_{\theta_{new}}\left(g_{\theta_{base}}(x)\right) \tag{4}$$

where:

- $g_{\theta_{base}}$ is the frozen base model (e.g., ResNet50) with pre-trained weights θ_{base},
- $f_{\theta_{new}}$ is the new classifier (typically dense layers) with trainable weights θ_{new},
- x is an MRI image,
- $\hat{y}$ is the predicted tumour class.

This approach mitigates the risk of overfitting and dramatically reduces training time, since only a fraction of the model parameters are updated [12].

In medical imaging, transfer learning has demonstrated high accuracy in various applications such as lung disease detection from chest X-rays, diabetic retinopathy grading from eye scans, and tumour classification from MRIs [13]. In our project, we adopt this approach to capitalize on the generalization power of established CNN architectures while tailoring them to our domain-specific data.

C. *Model Architectures*

We evaluate four widely used convolutional neural network (CNN) architectures: ResNet50, VGG16, EfficientNetB0, and InceptionV3. These models were selected for their proven performance in image classification tasks and their varying architectural philosophies, which provide insight into how model complexity and design influence tumour classification.

1) ResNet50: ResNet50 (Residual Network with 50 layers) introduces residual learning through identity shortcut connections that bypass one or more layers. These residual connections allow gradients to flow directly through the network during backpropagation, mitigating the vanishing gradient problem in deep networks. As discussed, the network learns residual functions that map the input to the desired output, rather than having to learn the entire mapping from scratch [14]. Each residual block learns a residual function:

$$\boldsymbol{y} = \mathcal{F}(x, \{W_i\}) + x \tag{5}$$

where:

- $\boldsymbol{x}$ is the input to the block,
- $\mathcal{F}$ is the residual mapping (typically composed of convolution, batch normalization, and ReLU),
- $\boldsymbol{y}$ is the output.

This structure allows deeper models to be trained more effectively. ResNet50 consists of a stack of 16 residual blocks, making it one of the deeper architectures in our study.

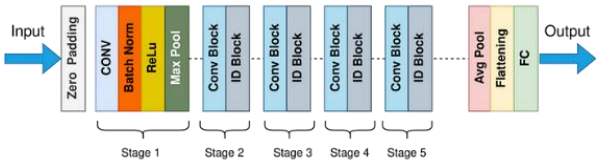

Fig. 4: ResNet50 Architecture [14].

2) VGG16: VGG16 is a 16-layer CNN with a simple, consistent architecture: stacks of 3×3 convolutional layers followed by 2×2 max pooling.

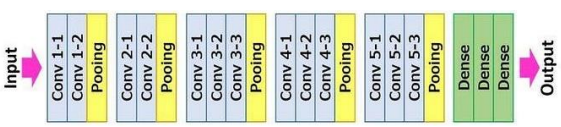

Fig. 5: VGG16 Architecture [15].

Despite its simplicity, it has been highly effective in image classification tasks. Each convolutional layer computes:

$$\boldsymbol{y}_{i,j}^{(k)} = \sigma\left(\sum_{m=1}^{M}\sum_{u=-1}^{1}\sum_{v=-1}^{1} w_{u,v}^{(k,m)} \cdot \boldsymbol{x}_{i+u,j+v}^{(m)} + b^{(k)}\right) \tag{6}$$

where:

- x is the input feature map,
- w are the filter weights,
- b is the bias,
- σ is the ReLU activation,
- (i, j) is the spatial location.

While VGG16 lacks architectural innovations like shortcuts, its large number of parameters enables it to learn detailed representations—though at the cost of slower training.

3) EfficientNetB0*:* EfficientNet models scale depth, width, and resolution in a balanced way using a compound coefficient:

$$\text{depth} = \alpha^{\phi}, \quad \text{width} = \beta^{\phi}, \quad \text{resolution} = \gamma^{\phi}$$

subject to $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$, where ϕ is a user-defined scaling factor.

EfficientNetB0 is the baseline model in this family, optimized for performance and efficiency. It uses MBConv blocks (inverted bottlenecks with depth wise separable convolutions), making it much faster and lighter than traditional CNNs while maintaining strong accuracy.

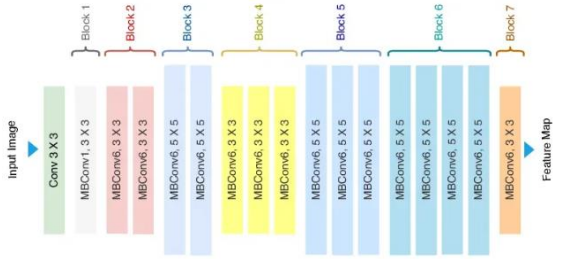

Fig. 6: EfficientNetB0 Architecture [16].

4) InceptionV3: InceptionV3 introduces the Inception module, which applies multiple filters (1×1, 3×3, 5×5) in parallel and concatenates their outputs.

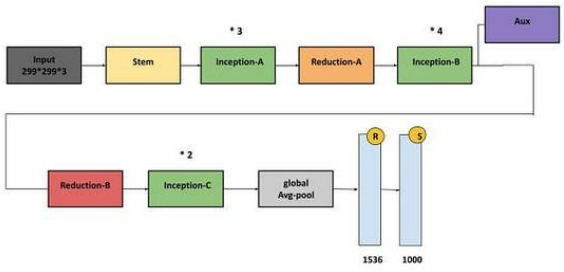

Fig. 7: InceptionV3 Architecture [17].

This allows the model to capture multi-scale features at each layer without significantly increasing computational cost. Inception modules typically compute:

$$\text{Output} = [\text{Conv}_{1\times1}, \text{Conv}_{3\times3}, \text{Conv}_{5\times5}, \text{MaxPool}_{3\times3}] \quad (7)$$

The combination of filters at different scales enables the network to handle variable-size features effectively, which is particularly beneficial in heterogeneous medical image data like brain MRIs.

These models are used as feature extractors in our pipeline. We remove their top classification layers and append custom dense layers trained on our tumour dataset (see Section III for details). This allows us to compare not only raw performance but also architectural trade-offs in speed, accuracy, and generalization.

D. *Related Work*

Convolutional neural networks have been widely adopted for medical image classification, including brain tumour detection. Notably, studies such as Bhuvaji et al. and Deepak and Ameer have used CNN-based models to classify MRI scans into tumour types with considerable success. For example, Bhuvaji et al. applied a CNN from scratch on the same dataset used in our study [2], achieving promising accuracy but lacking comparison with transfer learning approaches. Similarly, Deepak and Ameer [18] utilized features from a pre-trained GoogLeNet and combined them with classical classifiers to perform three-class classification of glioma, meningioma, and pituitary tumours, achieving up to 98% accuracy through patient-level five-fold cross-validation. However, their work focused solely on feature extraction and did not experiment with multiple model architectures or end-to-end fine-tuning.

While these works highlight the potential of CNNs in neuroimaging tasks, many limit themselves to a single architecture or shallow experimentation without consistent pipelines or evaluation across models. Moreover, they rarely report on generalization performance through robust techniques like K-fold cross-validation or evaluate multiple architectures within a unified framework. A segmentation-focused study by Malathi and Sinthia [19], for instance, demonstrated how CNNs can be used for glioma subtype segmentation using TensorFlow, but did not address classification among tumour types or compare pre-trained models.

E. *Research Gap*

This study aims to fill that gap by systematically comparing four well-established pre-trained CNNs using a consistent pipeline for preprocessing, augmentation, training, and evaluation. Our approach integrates standardized metrics, statistical validation, and live deployment—offering a reproducible and end-to-end solution not often found in prior work.

III. METHODS

A. *Research Objectives*

For this study, we propose the following hypothesis and corresponding objectives:

Hypothesis: Transfer learning using pre-trained convolutional neural networks will achieve strong classification performance on brain tumour MRI scans. We predict that ResNet50 will outperform the other models due to its residual connections, which support deeper architectures by improving gradient flow and enabling more effective feature learning.

O1: Evaluate the classification performance (accuracy, precision, recall, F1-score, and AUC) of four pre-trained CNN models—ResNet50, VGG16, EfficientNetB0, and InceptionV3—on a four-class brain tumour MRI dataset using five-fold cross-validation.

O2: Compare the results across models to determine the trade-offs between model complexity, accuracy, and generalization performance.

O3: Deploy all four trained models in a live web application (NeuroScanAI), allowing users to test any of them and view predicted tumour classes along with confidence scores and performance statistics.

B. *Research Methodology*

1) Dataset Preprocessing: The raw dataset used in this study is the publicly available Brain Tumour Classification (MRI) dataset from Kaggle [3], consisting of grayscale .jpg images labeled into four categories: Glioma, Meningioma, Pituitary, and None. All images were first loaded, converted to RGB, and resized to a consistent resolution of 224 × 224 to meet input requirements of the pre-trained CNN architectures.

a) Image Normalization and Preprocessing: Let each image be denoted as a tensor $\boldsymbol{X_i} \in \mathbb{R}^{224\times224\times3}$. All images were converted to float32 format and normalized using model-specific preprocessing functions $f_{prep}(\cdot)$, such that:

$$X^i = f_{\text{prep}}(X_i), \text{where } X^i = f_{\text{prep}}(X_i) \text{ and}$$

$$f_{\text{prep}} \in \{\text{ResNet50.preprocess}, \text{VGG16.preprocess},$$

$$\text{EfficientNetB0.preprocess}, \text{InceptionV3.preprocess}\}$$

These preprocessing functions apply transformations such as mean subtraction, scaling, and color channel normalization appropriate for the weights learned on ImageNet. The visual impact of these preprocessing steps is illustrated in Figure 8, which shows the same image as processed by each model's respective function.

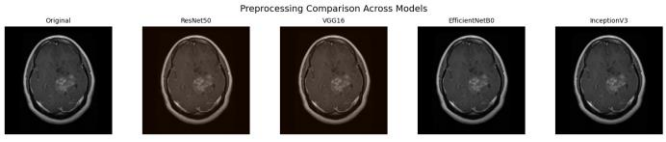

Fig. 8: Visual Comparison of Model-Specific Preprocessing Effects.

b) Label Encoding: Let $y_i \in \{\text{Glioma}, \text{Meningioma}, \text{Pituitary}, \text{None}\}$ be the categorical class labels. These were encoded into integer class labels using LabelEncoder, resulting in numerical labels $y_i' \in \{0,1,2,3\}$. This transformation allows compatibility with loss functions like categorical cross-entropy, defined as:

$$\mathcal{L}_C = -\sum_{k=1}^{K} y_k \, log(\widehat{y_k}) \tag{8}$$

where y_k is the one-hot encoded true label and $y^k, \widehat{y_k}$ is the predicted probability for class k.

c) Dataset Balancing with Augmentation: To address class imbalance, we applied synthetic data augmentation. Let $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$ be the set of class subsets. Let C_{max} be the cardinality of the largest class. We generated new synthetic images $\widetilde{\mathbf{X}}$ for each minority class C_k such that:

$$\left|C_k \cup \widetilde{C_k}\right| = |C_{max}|, \qquad \forall k \tag{9}$$

Augmentations included random horizontal and vertical flips, small rotations, zooms, and brightness shifts. The augmentation ensures:

- Balanced class distribution across all four categories
- Greater robustness to spatial and lighting variations
- More effective generalization during training

The original class distribution is shown in Figure 9, followed by Figure 10, which shows representative examples of synthetic image generation for each class. The final class distribution output after augmentation is shown in Figure 11, demonstrating that all categories have been balanced successfully.

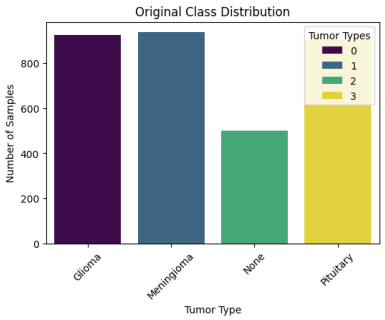

Fig. 9: Original Class Distribution.

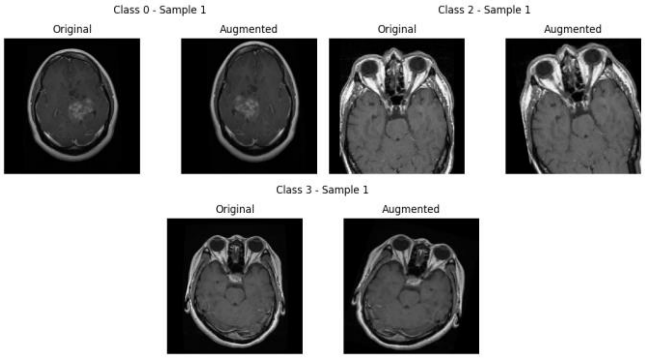

Fig. 10: Class Augmentation Samples.

```
Augmented class distribution:
  Class 0: 937 images
  Class 1: 937 images
  Class 2: 937 images
  Class 3: 937 images
```

Fig. 11: Augmented Class Distribution Cell Output.

d) Train/Test Splitting: To ensure statistical validity and prevent class skew, we applied Stratified Train/Test Splitting with a 70/30 ratio:

$$\text{Train set} = 70\% \text{ of samples (stratified)}, \text{ Test set} = 30\%$$

Formally, let $D = \{(\widehat{\boldsymbol{X}_i}, y_i')\}_{i=1}^{N}$. The dataset is partitioned as:

$$D_{\text{train}} \cup D_{\text{test}} = D, \quad D_{\text{train}} \cap D_{\text{test}} = \emptyset,$$

$$P(\, y_i' \mid D_{\text{train}}\,) \approx P(\, y_i' \mid D\,).$$

The resulting class distributions were visualized in Figure 12 to confirm stratification across training and testing sets.

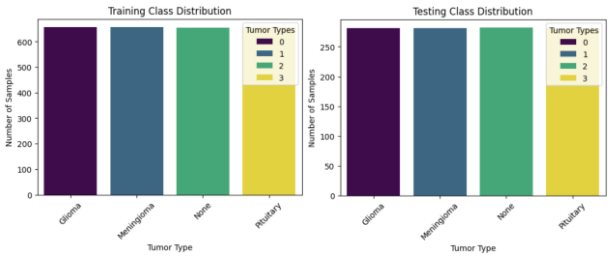

Fig. 12: Training and Testing Class Distribution.

e) Dimensionality Reduction (PCA): To validate the visual separability of the tumour classes, we performed Principal Component Analysis (PCA) on the flattened training set $\widehat{X_i} \in \mathbb{R}^{150528}$. PCA reduces each image to a 2D embedding for visualization:

$$Z_i = W^{\top} \cdot \text{vec}(\widehat{X_i}), \quad Z_i \in \mathbb{R}^{2} \tag{10}$$

where $W \in \mathbb{R}^{150528 \times 2}$ contains the top 2 principal components, and vec flattens the 3D image into a vector. The resulting 2D scatter plot in Figure 13 demonstrates that the augmented and stratified training set exhibits reasonable clustering of classes in feature space.

Fig. 13: PCA Projection of Augmented and Stratified Training Set.

2) Model Training & Implementation Strategy: Each model was trained using a standardized K-Fold cross-validation pipeline with fine-tuned classification heads. To ensure robustness and fairness in comparison, we used Stratified K-Fold Cross-Validation with consistent optimization parameters and callbacks across all experiments.

a) K-Fold Cross-Validation: We employed $K = 5$ stratified folds, where the dataset $D = \{(X_i, y_i)\}_{i=1}^{N}$ is partitioned into K disjoint subsets $\{D_1, D_2, \dots, D_5\}$. For each fold $k \in \{1, \dots, K\}$:

$$D_{\text{train}}^{(k)} = D/D_k, \quad D_{\text{val}}^{(k)} = D_k$$

The model is trained on $D_{train}^{(k)}$ and validated on $D_{\text{val}}^{(k)}$. This ensures each sample is used for validation exactly once.

b) Architecture and Fine-Tuning: Each model uses a frozen base from a pre-trained ImageNet architecture (e.g., ResNet50, VGG16), followed by a custom classification head:

$$f_\theta(X) = \text{Dense}_{\text{softmax}} \circ \text{Dropout}_{0.5} \circ \text{BatchNorm} \circ \text{Dense}_{128} \circ \text{Flatten} \circ g_{\text{base}}(X) \quad (11)$$

where g_{base} is the pre-trained CNN feature extractor with frozen parameters θ_{base}, and the rest is the trainable classification head.

Fine-tuning of the final layers was disabled, as keeping the pre-trained base frozen resulted in better performance during validation compared to fine-tuning.

c) Optimization Parameters: All models were compiled with the Adam optimizer with a decaying learning rate:

- Initial learning rate: $\eta_0 = 1 \times 10^{-3}$
- Weight decay (L2 regularization): $\lambda = 1 \times 10^{-4}$
- Loss: Sparse categorical cross-entropy $\mathcal{L} = -\log \widehat{p_{y_i}}$

To prevent overfitting and improve convergence, we used:

- EarlyStopping (patience = 5)
- ReduceLROnPlateau (factor = 0.2, patience = 3)
- ModelCheckpoint (save best model per fold)

d) Class Weights for Imbalance Handling: To counter class imbalance (even after augmentation noise), we computed class weights w_k inversely proportional to class frequency:

$$w_k = \frac{N}{K \cdot N_k} \quad (12)$$

where N_k is the number of samples for class k, and N is the total number of training samples. These weights were passed to the loss function during training.

e) Epoch-wise Logging and Model Selection: During each fold, we tracked:

- Training loss $\mathcal{L}_{train}^{(e)}$
- Validation loss $\mathcal{L}_{val}^{(e)}$
- Training accuracy $A_{train}^{(e)}$
- Validation accuracy $A_{val}^{(e)}$
- Learning rate $\eta(e)$

The best model weights were saved based on the maximum A_{val}, ensuring that only the top-performing parameters per fold contributed to final evaluation. Training dynamics are visualized in Figures 14 and 15, which show the average loss and accuracy curves across folds, as well as detailed epoch-wise logs from a representative training fold.

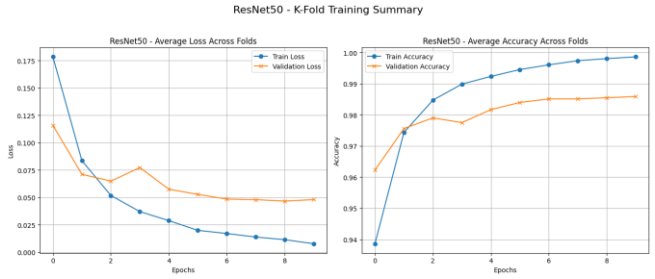

Fig. 14: Average Training and Validation Loss and Accuracy Across Folds.

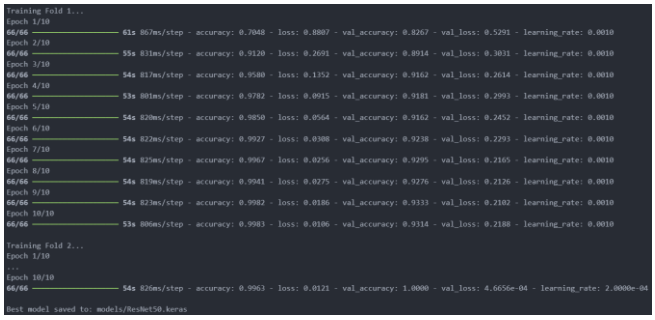

Fig. 15: Epoch-wise Loss and Accuracy Curves Averaged Over 5 Folds.

f) K-Fold Evaluation Metrics: After completing all folds, the validation accuracies per fold were stored in a set:

$$\mathcal{A}_{val}^{model} = \left\{A_{\text{val}}^{(1)}, A_{\text{val}}^{(2)}, \dots, A_{\text{val}}^{(K)}\right\}$$

We computed the mean and standard deviation across folds:

$$\mu = \frac{1}{K}\sum_{k=1}^{K} A_{\text{val}}^{(k)},$$

$$\sigma = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(A_{\text{val}}^{(k)} - \mu\right)^2} \quad (12, 13)$$

This provided a measure of both central tendency and variability in model performance across data splits. The results for all four models, including their mean validation accuracy and standard deviation, are visualized in Figure 16.

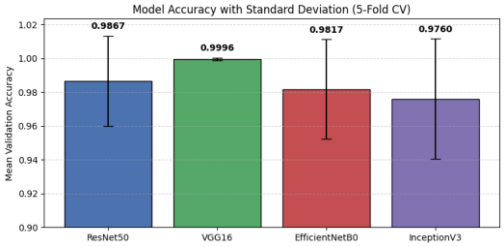

Fig. 16: Validation Accuracy (with Mean ± Std Dev).

3) Data Gathering Instruments: The dataset used in this study is the Brain Tumour Classification (MRI) dataset published by Sartaj Bhuvaji et al. on Kaggle [3]. It consists

of grayscale MRI images labeled into four diagnostic categories: Glioma, Meningioma, Pituitary, and None (no tumour). The dataset is licensed under the MIT License and is publicly available for academic research and machine learning benchmarking.

The dataset structure includes separate Training and Testing folders, each containing four subdirectories, one for each class. Image labels are inferred from the directory names, enabling automatic label extraction during data loading. The dataset contains a total of 3,264 images, distributed across classes with moderate imbalance, which was later corrected through augmentation.

No manual annotation or survey-based collection was required. The images were pre-collected and labeled by the dataset authors, with metadata indicating clinical relevance and anonymization. All images were accessed directly from the remote source and downloaded locally using the Kaggle API.

This dataset serves as a standardized benchmark for brain tumour classification tasks and has been used in multiple prior works involving Convolutional Neural Networks (CNNs), Transfer Learning (TL), and medical imaging pipelines.

4) Data Analysis Techniques: For each model, we computed:

- Accuracy:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{\widehat{y_i} = y_i\} \tag{14}$$

  where $\widehat{y_i}, y_i$ is the predicted label and $\widehat{y_i}$ and y_i the true label.

- Precision, Recall, F1-score: These were calculated per class and averaged using the *macro* method:

$$\text{Precision} = \frac{TP}{TP + FP},\ \text{Recall} = \frac{TP}{TP + FN},$$
$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15, 16, 17}$$

- Matthews Correlation Coefficient (MCC): A robust metric for multiclass evaluation, defined as:

$$\text{MCC} = \frac{\text{cov}(y, \hat{y})}{\sqrt{\text{var}(y) \cdot \text{var}(\hat{y})}} \tag{18}$$

- Cohen's Kappa: A metric that measures inter-rater agreement corrected for chance, useful for imbalanced multiclass classification. Defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{19}$$

where:

- p_0 is the observed agreement (accuracy)
- p_e is the expected agreement by chance

5) Threats to Validity: While this study offers a robust comparative evaluation of multiple CNN architectures for brain tumour classification, several limitations and potential threats to validity should be acknowledged.

a) Dataset Scope and External Validity: Our findings are based solely on a single, publicly available dataset [3] composed of curated grayscale MRI images. Although we addressed internal variance through augmentation and K-Fold cross-validation, the dataset lacks diversity in terms of imaging modalities, scanner types, and patient demographics. As such, generalization to clinical environments with heterogeneous data (e.g., varying resolutions, tumour grades, or acquisition protocols) remains uncertain. Similar to Bhuvaji et al. [3], who trained only on this dataset without external validation, we are limited in our ability to assess real-world robustness.

b) Omission of Segmentation and Localization: This work focuses purely on image-level classification, treating each MRI slice independently without considering spatial context or anatomical localization. Unlike Malathi and Sinthia [19], who explore segmentation for glioma subtyping, we do not address the position or extent of tumours — factors which may influence clinical decision-making and are valuable for holistic diagnostic tools.

c) Limited Exploration of Fine-Tuning and Optimization: We adopted a consistent pipeline with frozen feature extractors for each model to ensure comparability. However, we did not systematically explore fine-tuning strategies, hyperparameter optimization, or automated architecture search, which could further improve performance. Prior studies like Deepak and Ameer [18] also avoided end-to-end training, instead using classical classifiers. While our study improves upon that by using full model pipelines, the tuning space remains underexplored.

d) Class Imbalance and Augmentation Bias: Despite balancing the dataset through augmentation, there remains a possibility that synthetic transformations may introduce distributional biases or artifacts that deviate from real-world imaging patterns. While necessary for equal class representation, augmentation does not fully replicate natural data diversity, and may affect the learned feature space in subtle ways.

e) Evaluation Strategy Scope: Though we employed a wide range of metrics, including MCC and Cohen's Kappa, our evaluation remains limited to slice-level predictions. In clinical practice, diagnosis is often made at the patient level, aggregating multiple slices. Future work could expand the evaluation to patient-wise validation as seen in Deepak and Ameer [18], which would more closely reflect medical diagnostic processes.

6) Rationale for Decisions: The design of our pipeline was shaped by a desire to balance comparative rigor, real-world relevance, and computational efficiency.

a) Model Selection: We chose four widely-used pre-trained CNN architectures — ResNet50, VGG16, EfficientNetB0, and InceptionV3 — based on their established performance on large-scale visual recognition tasks and diverse architectural philosophies. This selection enables a representative comparison across residual networks (ResNet), classical deep stacks (VGG), compound-scaled models (EfficientNet), and multi-branch architectures (Inception). These models have also been successfully used in medical contexts, as seen in prior works [10, 14, 16].

b) Transfer Learning with Frozen Feature Extractors: Rather than training from scratch, we applied transfer learning using ImageNet weights, freezing the base model and training only custom classification heads. This decision reflects best practices in medical imaging research [9, 11], where labeled data is limited and general features (e.g., edges, textures) are transferable. By keeping the base frozen, we also reduce overfitting risk and training cost, while allowing for consistent evaluation across models.

c) Augmentation for Balancing and Generalization: To address class imbalance without discarding data, we used image augmentation to upsample minority classes. This approach maintains the full information of the original dataset and improves model generalization. The choice of geometric and photometric transformations (e.g., flips, rotations, brightness) was guided by their proven utility in medical classification tasks [13].

d) Stratified K-Fold Cross-Validation: Instead of a fixed train/val/test split, we used Stratified K-Fold Cross-Validation to ensure that each model was evaluated across multiple data permutations. This provides a more stable estimate of generalization performance and prevents accidental overfitting to a particular partition. Previous studies (e.g., Deepak and Ameer [18]) have shown that cross-validation leads to stronger empirical validity.

e) Metric Diversity: We included macro-averaged metrics (Precision, Recall, F1), MCC, and Cohen's Kappa to reflect both class-wise performance and inter-rater agreement. Additionally, ROC and PR curves provide insight into model behavior at various thresholds — a critical consideration in high-stakes diagnostic contexts. Accuracy alone was deemed insufficient due to class imbalance and variable clinical importance across tumour types.

f) Deployment Consideration: To bridge the gap between academic research and practical application, we deployed the trained models to a live web interface. This decision was motivated by the observation that few studies (including Bhuvaji et al. [2]) evaluate their models in a user-facing context. Live deployment highlights the feasibility and accessibility of AI-powered diagnostics for real-world users.

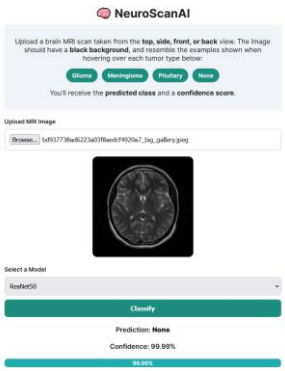

Fig. 21: NeuroScanAI interface for MRI upload and tumour prediction.

IV. RESULTS

This section presents and analyzes the classification performance, runtime efficiency, and comparative evaluation of four pre-trained CNN architectures — ResNet50, VGG16, EfficientNetB0, and InceptionV3 — applied to brain tumour MRI classification. All models were trained for 10 epochs per fold using 5-fold cross-validation, ensuring a consistent and efficient training regimen. Final evaluation was conducted on a held-out, stratified test set using multiple metrics, including accuracy, precision, recall, F1-score, and AUC. This approach enabled us to assess both generalization performance and model reliability across tumour classes.

A. *Classification Performance*

Table I: Summarizes the average classification metrics

Model	Accuracy	Avg. Precision	Avg. Recall	Avg. F1-Score
ResNet50	0.94	0.94	0.94	0.94
VGG16	0.93	0.93	0.93	0.93
EfficientNetB0	0.93	0.93	0.93	0.93
InceptionV3	0.90	0.91	0.90	0.91

ResNet50 consistently outperformed the other models, achieving the highest F1-scores for classes 2 and 3 (pituitary and no tumour). Detailed per-class metrics are shown in Figure 17.

	ResNet50			VGG16			EfficientNetB0			InceptionV3		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
CLASS												
0	0.94	0.9	0.92	0.88	0.89	0.89	0.88	0.93	0.9	0.91	0.85	0.88
1	0.91	0.88	0.9	0.91	0.85	0.88	0.89	0.85	0.87	0.81	0.9	0.85
2	0.96	0.98	0.97	0.96	0.97	0.97	0.96	0.97	0.96	0.96	0.95	0.96
3	0.95	1.0	0.97	0.96	0.99	0.97	0.98	0.96	0.97	0.96	0.91	0.94
Average	0.94	0.94	0.94	0.93	0.93	0.93	0.93	0.93	0.93	0.91	0.9	0.91
Accuracy	0.94			0.93			0.93			0.90		

Fig. 17: Per-Class Precision, Recall, and F1-Score across Models.

B. *Error Analysis*

To analyze per-class performance and model-specific misclassification trends, we present the normalized confusion matrices for all four models grouped in Figure 18.

- ResNet50 shows the strongest performance across all tumour classes.
- VGG16 and EfficientNetB0 demonstrate comparable accuracy but show slight confusion between glioma and meningioma.

- InceptionV3 underperforms slightly, especially in identifying meningioma.

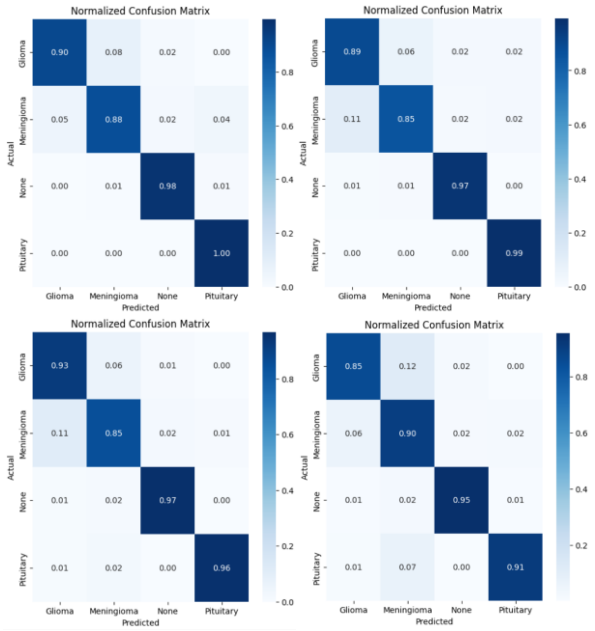

Fig. 18: Normalized confusion matrices for all models — ResNet50 (top-left), VGG16 (top-right), EfficientNetB0 (bottom-left), InceptionV3 (bottom-right).

Next, Figure 19 shows ROC and Precision-Recall (PR) curves for ResNet50. These illustrate high true positive rates and precision across thresholds, confirming ResNet50's robust classification ability.

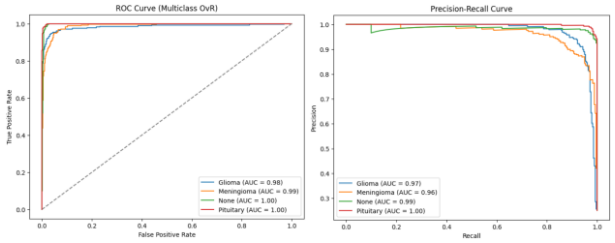

Fig. 19: ResNet50 ROC and PR Curves (Per-Class)

Figure 20 displays two additional agreement-based performance metrics — the Matthews Correlation Coefficient (MCC) and Cohen's Kappa — for ResNet50. These metrics account for chance agreement and class imbalance, offering a more robust evaluation of overall model performance

```
MCC: 0.92

---

Cohen's Kappa: 0.92
```

Fig. 20: ResNet50 MCC and Cohen's Kappa Scores

C. System Efficiency and Runtime

All experiments were conducted on a system with an Intel i5-13600KF, 32GB DDR5 RAM @ 6000 MT/s, and an NVIDIA RTX 3060 GPU (12GB). The total training time for each model was as follows:

Table II: Training Time

Model	Training Time (5-Fold and 10 Epochs)
ResNet50	45 minutes
VGG16	104 minutes
EfficientNetB0	15 minutes
InceptionV3	27 minutes

Despite similar final accuracies, EfficientNetB0 offered the best trade-off between training time and performance. VGG16, although accurate, incurred the longest runtime due to its deep parameter count and lack of architectural optimization for modern hardware. These results reflect practical deployment trade-offs — especially for time-sensitive or resource-constrained environments.

D. Novelty and Contribution

Unlike prior studies such as Bhuvaji et al. [3], which only used a single CNN architecture, or Deepak and Ameer [18], which avoided full end-to-end training, this work systematically compares four CNN architectures using a unified, reproducible pipeline. Each model was evaluated using K-fold cross-validation, stratified test evaluation, and advanced statistical metrics (e.g., MCC, Cohen's Kappa), providing comprehensive and generalizable results.

Moreover, we went beyond academic experimentation by deploying the models as a live web application (NeuroScanAI) — enabling users to upload MRI images and receive real-time tumour classification with model-specific confidence scores. This represents a practical step toward accessible, AI-driven diagnostic support tools.

V. Conclusions and Future Work

A. Conclusion

In line with our hypothesis, ResNet50 achieved the highest classification performance across all evaluation metrics, with an overall accuracy of 94% and macro-averaged precision, recall, and F1-score of 0.94. Its ability to detect all tumour classes with strong consistency—particularly glioma and pituitary—demonstrates the advantage of residual learning in medical image classification. While EfficientNetB0 matched ResNet50 in many metrics, it stood out primarily for its speed and efficiency, completing training in just 15 minutes. However, its slightly lower sensitivity on some tumour types makes ResNet50 the more reliable model for clinical settings.

Our comparative evaluation supports the use of pre-trained CNNs for brain tumour classification from MRIs and validates the effectiveness of transfer learning combined with standardized preprocessing and stratified K-fold evaluation.

B. Future Work

While this study provides a rigorous and reproducible framework, future work should explore patient-level prediction using multiple slices per individual, rather than independent slice-level classification. Additionally, extending the pipeline to include tumour segmentation and localization would improve clinical applicability. We also plan to investigate fine-tuning strategies and automated architecture search, and to test performance on external datasets to assess cross-domain generalization. These steps will help further close the gap between research prototypes and deployable, trustworthy AI diagnostic systems.

VI. REFERENCES

[1] D. N. Louis, H. Ohgaki, O. D. Wiestler, and W. K. Cavenee, *WHO Classification of Tumours of the Central Nervous System*, 4th ed. Lyon, France: IARC Press, 2016. Accessed [4-Apr-25].

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. Accessed [4-Apr-25].

[3] S. Bhuvaji, A. Kadam, P. Bhumkar, and S. Dedge, "Brain Tumour Classification (MRI)," *Kaggle*, 2019. [Online]. Available: https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumour-classification-mri. Accessed [4-Apr-25].

[4] McDermott, J., "Convolutional Neural Networks — Image Classification w. Keras," *LearnDataSci*, 2018. [Online]. Available: https://www.learndatasci.com/tutorials/convolutional-neural-networks-image-classification. [Accessed: 4-Apr-2025].

[5] CelerData. (2024, August 7). *Convolutional Neural Network (CNN)*. CelerData. https://celerdata.com/glossary/convolutional-neural-network-cnn. Accessed [4-Apr-25].

[6] Gabriele De Luca. *How ReLU and Dropout Layers Work in CNNs*. Baeldung, 30 August 2024. Available at: https://www.baeldung.com/cs/ml-relu-dropout-layers. Accessed [4-Apr-25].

[7] "Softmax Regression." *Dive into Deep Learning*, https://d2l.ai/chapter_linear-classification/softmax-regression.html. Accessed [4-Apr-25].

[8] Medium Citation: Medium. (2021). Deriving Backpropagation with Cross-Entropy Loss. *Medium*. https://medium.com/data-science/deriving-backpropagation-with-cross-entropy-loss-d24811edeaf9. Accessed [4-Apr-25].

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. Accessed [4-Apr-25].

[10] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019. Accessed [4-Apr-25].

[11] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, pp. 1–74, 2021. Accessed [4-Apr-25].

[12] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016. Accessed [4-Apr-25].

[13] H. Hassan et al., "Survey on deep learning techniques in medical imaging: Transfer learning and future directions," *arXiv preprint arXiv:2401.12345*, 2024. Accessed [4-Apr-25].

[14] Kundu, Nitish. "Exploring ResNet50: An In-Depth Look at the Model Architecture and Code Implementation." *Medium*, 23 Jan. 2023, https://medium.com/%40nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f. Accessed [4-Apr-25].

[15] G, R. (2021). *Everything you need to know about VGG16*. Great Learning. Retrieved from https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918. Accessed [4-Apr-25].

[16] Potrimba, P. (2023). What is EfficientNet? The Ultimate Guide. Roboflow Blog. Retrieved from https://blog.roboflow.com/what-is-efficientnet/. Accessed [5-Apr-25].

[17] Brital, A. (2021). *Inception V3 CNN architecture explained*. Medium. Retrieved from https://medium.com/@AnasBrital98/inception-v3-cnn-architecture-explained-691cfb7bba08. Accessed [4-Apr-25].

[18] S. Deepak and P. M. Ameer, "Brain tumour classification using deep CNN features via transfer learning," Computers in Biology and Medicine, vol. 111, p. 103345, 2019. doi: 10.1016/j.compbiomed.2019.103345. Accessed [4-Apr-25].

[19] M. Malathi and P. Sinthia, "Brain tumour segmentation using convolutional neural network with Tensor Flow," Asian Pacific Journal of Cancer Prevention, vol. 20, no. 7, pp. 2095–2101, 2019. doi: 10.31557/APJCP.2019.20.7.2095. Accessed [4-Apr-25]