Brain Tumor Classification using Transfer Learning

Saeed Bin Mizan
Department of Computer Science
b00093343@aus.edu

Mohamed Monir Eltamamy
Department of Computer Science
b00092938@aus.edu

Samad Sayed
Department of Computer Science
b00092829@aus.edu

***Abstract*—Accurate and efficient brain tumor classification is critical for improving patient outcomes and supporting clinical decision-making. In this project, we investigate the performance of several state-of-the-art deep learning models for classifying brain MRI images into four categories: Glioma, Meningioma, Pituitary Tumor, and No Tumor. We utilize the PMRAM: Bangladeshi Brain Cancer MRI Dataset, which comprises 6,000 images balanced across these classes and sourced from multiple hospitals in Bangladesh, ensuring diversity and robustness. Transfer learning is employed with pre-trained architectures including DenseNet-169, ResNet-50, AlexNet, and Vision Transformer (ViT), alongside a custom CNN baseline. Each model is fine-tuned and evaluated using accuracy, precision, recall, F1-score, and AUC metrics. Experimental results show that DenseNet-169 and ResNet-50 achieve near-perfect classification performance, significantly outperforming both AlexNet, ViT, and the custom CNN. These findings highlight the effectiveness of advanced transfer learning approaches for brain tumor detection and demonstrate their potential medical imaging diagnoses.**

***Index Terms*—Brain tumor classification, Magnetic Resonance Imaging (MRI), deep learning, convolutional neural networks (CNN), transfer learning, pretrained models**

I. Introduction

Brain cancer remains a formidable health challenge throughout the world, with an estimated 347,992 new cases and 246,253 deaths recorded globally in 2019 [1]. In the United States alone, projections for 2024 indicate approximately 25,400 new diagnoses and 18,760 deaths due to malignant brain and spinal cord tumors [2]. Although they represent a smaller fraction of overall cancer cases, brain tumors often result in significant morbidity and mortality, underscoring the need for accurate and early diagnostic methods. Brain tumors arise from rapid, uncontrolled cell proliferation within the brain or central nervous system and can be classified according to their origin, growth rate, and stage of progression [3]. Accurate classification is critical for effective, targeted therapy and improved patient outcomes. Traditional diagnosis relies heavily on manual interpretation of brain scans and examination of biopsy samples, which are time-consuming and require specialized expertise. Early detection of brain tumors is vital, allowing early intervention to increase the likelihood of complete tumor removal, improving overall survival rates [4]. Several publicized and well-established datasets have been used for classification for different types of brain tumors. To start, we have the Figshare Brain Tumor Dataset which stands out for its comprehensive collection of MRI images. This dataset contains T1-weighted contrast-enhanced MRI scans from 233 patients, covering key tumor types such as meningioma, glioma, and pituitary tumors [5]. Another prominent resource is the REMBRANDT (REpository for Molecular BRAin Neoplasia DaTa) dataset, hosted by Georgetown Lombardi Comprehensive Cancer Center.. This dataset integrates genomic, diagnostic, treatment, and outcomes data from 671 adult brain cancer patients across 14 institutions [6]. Lastly, the Brain MRI ND-5 Dataset is another option for brain tumor classification and detection tasks. It features MRI images labeled for multiple tumor types, including gliomas, meningiomas, pituitary tumors, and healthy controls [7]. For this project, we have chosen the PMRAM: Bangladeshi Brain Cancer - MRI Dataset from Kaggle. This dataset comprises of 1,600 raw MRI images, augmented to 6,000, evenly distributed among four categories: glioma, meningioma, pituitary tumor, and no tumor. Collected from several hospitals in Bangladesh, the dataset ensures diverse representation and is standardized to 512×512 pixels for compatibility with deep learning models. Developed in collaboration with medical professionals, PMRAM provides a robust foundation for training and evaluating state of the art methods [8].

II. Dataset Description

The PMRAM: Bangladeshi Brain Cancer Dataset is a comprehensive resource designed to support research and development in brain tumor detection and classification. This dataset comprises 6,000 MRI images, systematically categorized into four equally represented classes: Glioma, Meningioma, Pituitary tumor, and No Tumor, with 1,500 images per category. The images were collected from multiple hospitals across Bangladesh, including Ibn Sina Medical College in Dhaka, Dhaka Medical College & Hospital, and Cumilla Medical College, ensuring a diverse and representative sample of the Bangladeshi population.

Initially, the dataset consisted of 1,600 raw MRI scans (400 per class). To enhance the robustness and size of the dataset, various augmentation techniques were applied, expanding it to 6,000 images. These augmentations included random rotations of up to 45 degrees, horizontal and vertical shifts of up to 20% of the image dimensions, shear transformations of up to 20%, random zooming in and out of up to 20%, horizontal flipping, and filling empty areas with the nearest pixel values.

All images have been resized to a standardized 512×512 pixel format.

The PMRAM dataset is distributed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license, making it openly accessible to researchers.

III. Literature Review

In recent years, research in brain tumor classification has become a focal point within the computer vision domain, aiming to provide non-invasive and accurate diagnostic support for clinicians. This review synthesizes findings from recent studies, highlighting methodological advances and comparative performance in the context of deep learning-based brain tumour classification.

A. Hybrid Deep Learning Approach: GoogleNet and SVM

Rasool et al. (2022) proposed a hybrid deep learning model that combines the feature extraction capabilities of a fine-tuned GoogleNet CNN with the classification strength of a Support Vector Machine (SVM). In this approach, GoogleNet is first trained on MRI images to extract high-level features, after which the final classification is performed by an SVM rather than the CNN's native softmax layer. This hybrid model achieved an accuracy of 98.1%, outperforming several baseline and state-of-the-art methods, including standalone CNNs and traditional feature-based classifiers. The study also demonstrated that the hybrid model reduced diagnostic errors and improved early detection rates, which are critical for effective treatment planning. Notably, the hybrid approach was particularly effective in distinguishing between glioma, meningioma, pituitary tumours, and non-tumour cases, thereby enhancing clinical utility [9].

B. Transfer Learning with CNN Architectures: AlexNet, GoogLeNet, and VGG16

Rehman et al. (2020) presented a comprehensive framework for automatic brain tumour classification using transfer learning with three prominent CNN architectures: AlexNet, GoogLeNet, and VGG16. The study used the Figshare brain tumour MRI dataset, which includes meningioma, glioma, and pituitary tumour images. Two transfer learning strategies were explored: fine-tuning (adjusting pretrained weights on the target dataset) and freezing (using pretrained features with a separate classifier such as SVM) [10]. Key contributions and findings include:

- **Data Augmentation**: Techniques such as rotation and flipping were used to expand the dataset and reduce overfitting.
- **Fine-Tuning vs. Freezing**: Fine-tuning VGG16 achieved the highest accuracy (98.69%), outperforming both AlexNet and GoogLeNet. The freeze approach with SVM also yielded strong results, with GoogLeNet features achieving up to 97% accuracy.
- **Comparative Performance**: All three CNN architectures outperformed traditional machine learning methods and earlier deep learning models that did not use transfer learning or data augmentation.

C. Hierarchical Deep Learning Neural Network (HieDNN) Classifier

Shajin et al. (2023) presented an efficient framework using a Hierarchical Deep Learning Neural Network (HieDNN) classifier for brain tumour classification. Their pipeline included:

- **Preprocessing**: Savitzky-Golay (SG) denoising to enhance MRI image quality.
- **Feature Extraction**: Texture features using the grey-level co-occurrence matrix (GLCM).
- **Classification**: The HieDNN classifier, which incorporates both physics-based and experimental neural subnets, reduces dimensionality and extracts relevant features for robust classification.

The system was evaluated on the BRATS dataset, achieving substantial improvements in accuracy, precision, and F-score compared to conventional CNN and 3D CNN models. Notably, the HieDNN approach also reduced computational complexity, making it suitable for real-time clinical applications. The method was particularly effective in classifying benign, malignant, and normal cases, outperforming existing deep learning frameworks in both accuracy and efficiency [11].

IV. Methodology

In this section we discuss the pretrained models used, the evaluation metrics to determine the performance of each model, and the custom CNN architecture to act as a baseline. To enable comparisons between the state of the art methods, we use the Adam optimizer with a learning rate of 0.0001 for 10 epochs. Categorical cross entropy loss was utilized for multiclass classification. In addition, we follow a train-test-val split of 70-20-10, where 70% of the data is used for training, 20% used for testing, and 10% used for validation.

For the CustomCNN however, the number of epochs was bumped to 50, with the addition of Kaiming weight intialization and the incorporation of weight decay to handle overfitting. Early stopping was initially used, but due to training stopping too early, despite the improvement in validation accuracy, we decided to forego its use in later experiments. The winning combination that gave us the best results is explained in the Results and Discussion section.

A. Pretrained Models

We use the following pretrained models: Alexnet, Resnet50, Vision Transformer (ViT), and Densenet-169. We utilize transfer learning, a machine learning technique that allows us to exploit the knowledge the model gained on one task to improve performance on another related task [12].

Alexnet: AlexNet is a pioneering convolutional neural network that won the ImageNet competition in 2012, sparking the deep learning revolution in computer vision. The architecture consists of eight layers: five convolutional layers (some followed by max-pooling) and three fully connected layers [13]. Key innovations include the use of ReLU activation functions for faster training, local response normalization, overlapping max pooling, and dropout for regularization. AlexNet demonstrated that deep CNNs could achieve outstanding performance

on large-scale image classification tasks and laid the foundation for modern deep learning models.

Resnet50: ResNet50 is a deep convolutional neural network with 50 layers, designed to tackle the vanishing gradient problem in very deep networks by introducing residual blocks with skip connections [14]. These connections allow the network to learn residual functions, making it easier to train deeper architectures. Each residual block in ResNet50 uses a bottleneck design with three convolutional layers (1x1, 3x3, 1x1), enabling efficient feature extraction and information flow throughout the network. ResNet50 is widely used for image classification and has demonstrated high accuracy in various computer vision applications, including medical image analysis.

Vision Transformer (ViT): The Vision Transformer (ViT) is a transformer-based model adapted for image classification [15]. Unlike traditional convolutional neural networks, ViT splits an image into fixed-size patches, flattens them, and embeds each patch as a vector. These vectors, with positional encodings, are processed by standard transformer encoder layers that use self-attention mechanisms to capture global relationships across the image. ViT models have shown strong performance, especially when trained on large datasets, and are notable for their flexibility and efficiency in learning complex visual patterns without relying on convolutional layers.

Densenet-169: DenseNet169 is a convolutional neural network from the DenseNet family, characterized by dense connectivity between layers [16]. In DenseNet architectures, each layer receives inputs from all preceding layers, promoting feature reuse and improving gradient flow. DenseNet169 consists of four dense blocks, with the number of layers in each block arranged as [6, 12,32]. This dense connectivity leads to efficient parameter usage and strong performance in image classification. DenseNet169 is particularly effective for tasks requiring detailed feature extraction, such as medical image analysis.

B. Custom Architecture

Along with comparing state of the art methods with each other, we also implemented a custom CNN architecture to determine the feasibility of training a model from scratch and achieving acceptable performance. The custom architecture consisted of convolution layers, max pooling layers, dropout layers, and batch normalization layers. For activations, we used the ReLU activation function. Weights were initalised with Kaiming weight intialisation.

Convolutional Layer: Fundamental to Convolutional Neural Networks (CNNs), they enable the model to extract spatial hierarchies of features from input images. Each convolutional layer applies a set of learnable filters (kernels) that convolve across the input's spatial dimensions, producing feature maps that capture local patterns such as edges, textures, and shapes.

Max Pooling Layer: These layers are used to downsample feature maps, reducing their spatial dimensions while retaining the most salient information. By sliding a window (typically 2×2) over the input and selecting the maximum value within each region, max pooling provides a form of translation invariance and reduces computational complexity.

Dropout Layer: Dropout is a regularization technique that mitigates overfitting in neural networks. During training, dropout layers randomly deactivate a fraction of neurons in the network, preventing the network from using the same neurons too much [17]. This encourages the network to develop redundant representations, enhancing its generalization capabilities.

Batch Normalization Layer: Batch normalization layers normalize the output of each layer to have zero mean and unit variance. By stabilizing the layer outputs, batch normalization not only accelerates and stabilizes training, but also handles the exploding or vanishing gradient problem.

ReLU Activation Function : The Rectified Linear Unit (ReLU) is a widely used activation function in deep learning models. It introduces non-linearity by outputting zero for negative inputs and the input itself for positive inputs. ReLU is computationally efficient and helps mitigate the vanishing gradient problem, facilitating the training of deep networks.

Kaiming Weight Initialization: The idea behind it is to initialize the weights using a random number so that each neuron learns a different feature. The weights should be initialized with numbers that aren't too large or too small, which would otherwise lead to vanishing or exploding gradients. There are different ways to initialize the weights, one such way that is optimized with the ReLU activation function is the Kaiming (He) initialization method. This method ensures that the variance of the outputs of each layer is roughly the same as the variance of the inputs, which helps to prevent the aforementioned gradient problems.

Below is a high level overview of the custom architecture's implementation:

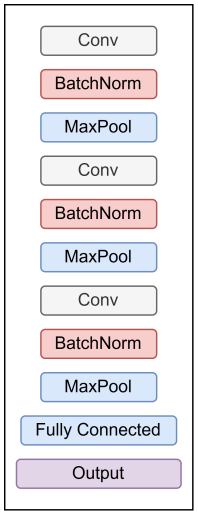

Fig. 1. Model Architecture

The model is made up of three convolutional blocks, with each block having a convolutional layer, a batch normalization layer, a ReLU activation function, a max pooling layer, and a dropout layer, in that specific order. After the input passes

through the three convolutional blocks, the output is flattened and fed into two fully connected layers, ultimately producing a prediction across one of four classes. Additionally, the weights were intialized using the Kaiming weight initialization. A detailed table of the model architecture can be found in the appendix.

C. Evaluation Metrics

To evaluate the performance of the state of the art methods as well as the custom CNN architecture, we consider the following metrics: Accuracy, Precision, Recall, F1score, and the ROC curve. We also look at the confusion table to get a visual overview of the model performance.

Confusion Matrix: This matrix is a tabular visualization that summarizes the performance of a classification model by displaying the number of correct and incorrect predictions for each class. Each row of the matrix represents the instances of an actual class, while each column represents the instances of a predicted class. This diagonal of the confusion matrix shows the number of correct prediction that the model makes, and every other cell are incorrect predictions made by the model. The figure of the classic confusion matrix is given below:

	Predicted Label 1	Predicted Label 2
Actual Label 1	TP	FP
Actual Label 2	FN	TN

Fig. 2. Example of a Confusion Matrix.

Accuracy: It is a straightforward metrics defined as the ratio the correct predictions over the total number of observations. This metric is suitable for balanced datasets, which applies in our case. It is given by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision: This metric measures the ratio of correctly predicted positive observations to the total predicted positive observations. It becomes particularly important in applications where the cost of false positives is high, such as in brain tumor classification, where incorrectly diagnosing a healthy patient as having a tumor could lead to unnecessary anxiety, further invasive testing, and unwarranted treatments. The formula for precision is given by:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall: Also called sensitivity, this metric measures the ratio of correctly predicted positive observations to all actual positive observations. Recall is critical in scenarios where minimizing false negatives is crucial, such as in medical diagnoses, where incorrectly identifying a patient with a specific tumor as having no tumor or some other tumor could prove to be fatal. The formula for recall is given by:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-Score: This metric is the harmonic mean of precision and recall and is calculated as 2 * (Precision * Recall) / (Precision + Recall). It is a robust metric that shows the trade-off between precision and recall. The formula for F1-Score is given by:

$$F_1 = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

ROC: Known as the Receiver Operating Characteristic, it is a graphical representation that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold varies. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings, effectively showcasing the trade-off between sensitivity (TPR) and specificity (TNR). The Area Under the ROC Curve (AUC) quantifies the overall ability of the model to distinguish between positive and negative classes. An AUC of 1.0 indicates perfect discrimination, whereas an AUC of 0.5 suggests no discriminative power, equivalent to random guessing. In the context of brain tumor classification, a high AUC value implies that the model is proficient at ranking tumor cases higher than non-tumor cases, regardless of the specific threshold applied. However, it's important to note that while AUC provides a measure of separability, it doesn't convey information about the actual classification thresholds, which are critical when considering the balance between false positives and false negatives in medical diagnoses. Below is sample diagram of an ROC curve from one of our models:

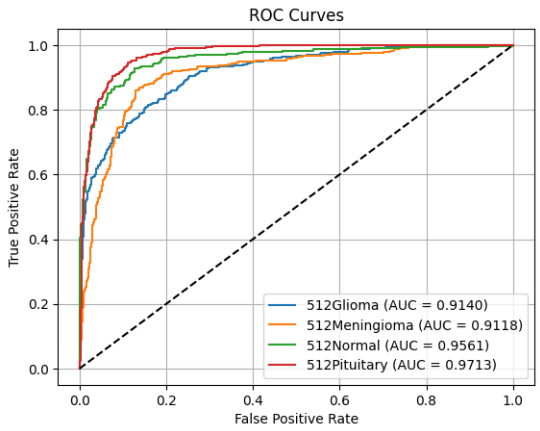

Fig. 3. Sample ROC Curve

We can see that the AUC values are relatively high for all classes, indicating that the model is proficient as differentiating between the positive class and the negative classes. For brevity, we summarize the AUC values in the main text. Full ROC curve plots for all models are provided in the Appendix.

V. Results and Discussion

Below we present a table that contains the results from the state of the art methods and our baseline Convolutional Neural Network (CNN). In order to fit the table properly we abbreviate the classes as shown below:

- Glioma (G)
- Meningioma (M)
- Normal (N)
- Pituitary (P)

The results in Table 1 demonstrate that all evaluated pre-trained models achieved high performance across multiple metrics on the brain tumor classification task. Among the models, Densenet-169 and Resnet-50 stand out, both attaining perfect or near-perfect scores in several categories. Densenet-169, in particular, achieved 100% accuracy, precision, recall, F1-score, and AUC across all tumor classes, indicating flawless classification on this dataset. Resnet-50 also performed exceptionally well, with 98% accuracy and similarly high values for all metrics.

Alexnet and ViT also demonstrated strong performance, though with slightly lower metrics compared to Densenet-169 and Resnet-50. Alexnet achieved 94% accuracy, with precision, recall, and F1-scores in the range of 85% to 98% across different tumor types. ViT showed variability, with precision for Meningioma dropping to 87% and recall for Glioma dropping to 84%. However, high scores were still maintained for the rest of the classes and their associated metrics, scoring 90% or more.

The consistently high AUC values across all models and classes suggest that the models are highly effective at distinguishing between the different tumor types and normal cases. The superior performance of Densenet-169 and Resnet-50 may be attributed to their deeper architectures and ability to capture complex features. Resnet-50, in particular utilizes residual (shortcut) connections, which allow the network to be trained with many more layers than traditional CNNs without suffering from vanishing gradients or degradation problems [18]. On the other hand, Densenet-169 distinguishes itself through its densely connected layers, where each layer receives input from all preceding layers, promoting overall depth and feature extraction capabilities [19]. This architecture also employs global average pooling, improving generalization [19].

The CustomCNN achieved an overall accuracy of 82.03%. While its performance was lower compared to the pre-trained models, it still managed to achieve reasonable precision, recall, and F1-score values across the four classes, particularly with regards to the Normal and Pituitary classes but had more difficulty with the Glioma and Meningioma classes. Despite the lower precision and recall, the AUC values were relatively high for all classes, indicating that the model was still able to distinguish between the different tumor types effectively.

The results for this CustomCNN was achieved after re-running the model at different weight decays, starting from 0.0001. A weight decay of 0.01 gave us the best results in this case. Initially, step scheduling was used where the learning rate of the model was reduced by a factor of 0.1 every 10 epochs for a total of 50 epochs. However, this led to lackluster results, with the validation accuracy unable to exceed 50%. Modifying the gamma value of the step scheduler and increasing the number of epochs, along with adjusting the architecture of the model could lead to better performance.

This contrast underscores the value of leveraging pre-trained models for brain tumor classification tasks. Pre-trained models, having been trained on extensive and diverse datasets, can capture complex features that might be challenging to learn from scratch. By fine-tuning these models on specific tasks, we can achieve improved performance and generalization, making them particularly beneficial in medical applications.

VI. Conclusion

In this project, we implemented and evaluated several state-of-the-art deep learning architectures for the task of brain tumor classification using MRI images. Leveraging a diverse and well-augmented dataset, we demonstrated that pretrained models, particularly Densenet-169 and Resnet-50, significantly outperformed our Custom CNN baseline, achieving near-perfect classification metrics across all tumor classes. These findings underscore the effectiveness of transfer learning when dealing with medical imaging tasks that have limited training data, allowing models to generalize better and achieve higher performance with fewer computational resources. Our results also highlight the critical role of architectural choices—such as residual connections and dense connectivity—in improving classification accuracy and robustness. Moving forward, further improvements could be explored through model ensembling, more advanced data augmentation techniques, or integrating attention mechanisms to enhance the model's focus on tumor-specific regions. The complete source code and trained models used in this project are available at: GitHub

Acknowledgement

This final project was required for our Computer Vision Course at the American University of Sharjah. We utilized Generative AI tools such as ChatGPT for optimization of the code and to enhance the efficiency of our development workflow.

References

[1] I. Ilic and M. Ilic, "International Patterns and trends in the brain cancer incidence and mortality: An observational study based on the global burden of disease," Heliyon, vol. 9, no. 7, Jul. 2023. doi:10.1016/j.heliyon.2023.e18222

[2] C. Zhang, "Cancer types: Brain cancer - national foundation for cancer research," NFCR, https://www.nfcr.org/cancer-types/cancer-types-brain-cancer/ (accessed Apr. 26, 2025).

[3] R. Kaifi, "A review of recent advances in brain tumor diagnosis based on AI-based classification," *Diagnostics*, vol. 13, no. 18, p. 3007, Sep. 2023. doi:10.3390/diagnostics13183007.

TABLE I
PERFORMANCE OF PRE-TRAINED MODELS ON BRAIN TUMOR CLASSIFICATION

Model	Accuracy (%)	Precision (%)				Recall (%)				F1-Score (%)				AUC (%)			
		G	M	N	P	G	M	N	P	G	M	N	P	G	M	N	P
Alexnet	94.0	91.0	97.0	94.0	93.0	96.0	85.0	95.0	98.0	94.0	91.0	95.0	96.0	99.54	98.98	99.35	99.70
Resnet-50	98.0	100.0	98.0	98.0	97.0	93.0	99.0	100.0	100.0	96.0	99.0	99.0	99.0	99.90	99.98	100.0	99.98
ViT	94.0	98.0	87.0	99.0	94.0	84.0	99.0	96.0	99.0	90.0	93.0	97.0	97.0	99.19	99.53	99.91	99.88
Densenet-169	100.0	100.0	99.0	100.0	1000.0	99.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
CustomCNN	82.03	79.0	74.0	87.0	87.0	71.0	73.0	88.0	96.0	75.0	74.0	87.0	91.0	93.40	94.10	97.57	98.82

[4] Admin, "The importance of early detection- brain tumor screening and diagnosis - spine & brain clinic," Spine & Brain Clinic - Best Spine & Brain Specialist in Pune, https://www.spinenbrain.com/uncategorized/the-importance-of-early-detection-brain-tumor-screening-and-diagnosis/ (accessed Apr. 26, 2025).

[5] J. Cheng, "Brain tumor dataset," *figshare*, Apr. 2, 2017. [Online]. Available: https://figshare.com/articles/dataset/brain_tumor_dataset/1512427/8

[6] Y. Gusev *et al.*, "The Rembrandt Study, a large collection of genomic data from Brain Cancer Patients," *Scientific Data*, vol. 5, no. 1, Aug. 2018. doi:10.1038/sdata.2018.158.

[7] Md. Nasif Safwan, Souhardo Rahman, Mahamodul Hasan Mahadi, Taharat Muhammad Jabir, Iftekharul Mobin, October 6, 2024, "Brain MRI ND-5 Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/q2vt-tf46.

[8] Orvile, "PMRAM: Bangladeshi brain cancer - MRI dataset," Kaggle, https://www.kaggle.com/datasets/orvile/pmram-bangladeshi-brain-cancer-mri-dataset/data (accessed Apr. 26, 2025).

[9] M. Rasool et al., "A Hybrid Deep Learning Model for Brain Tumour Classification," Entropy, vol. 24, no. 6, p. 799, Jun. 2022, doi: https://doi.org/10.3390/e24060799.

[10] A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran, "A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning," Circuits, Systems, and Signal Processing, vol. 39, no. 2, pp. 757–775, Sep. 2019, doi: https://doi.org/10.1007/s00034-019-01246-3.

[11] F. H. Shajin, P. Rajesh, Venu Gopal Rao, and P. Salini, "Efficient Framework for Brain Tumour Classification using Hierarchical Deep Learning Neural Network Classifier," Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, vol. 11, no. 3, pp. 1–8, Aug. 2022, doi: https://doi.org/10.1080/21681163.2022.2111719.

[12] Ibm, "What is transfer learning?," IBM, https://www.ibm.com/think/topics/transfer-learning (accessed Apr. 26, 2025).

[13] N. Klingler, "AlexNet: A Revolutionary Deep Learning Architecture," viso.ai, Apr. 29, 2024. https://viso.ai/deep-learning/alexnet/

[14] "Understanding ResNet50: A Comprehensive Guide to the Architecture," thinkingstack, Feb. 06, 2025. https://www.thinkingstack.ai/blog/business-use-cases-11/a-comprehensive-guide-to-resnet50-architecture-and-implementation-56

[15] D. Shah, "Vision Transformer: What It Is & How It Works [2023 Guide]," www.v7labs.com, Dec. 15, 2022. https://www.v7labs.com/blog/vision-transformer-guide.

[16] B. A. S. Al-rimy, F. Saeed, M. Al-Sarem, A. M. Albarrak, and S. N. Qasem, "An Adaptive Early Stopping Technique for DenseNet169-Based Knee Osteoarthritis Detection Model," Diagnostics, vol. 13, no. 11, p. 1903, May 2023, doi: https://doi.org/10.3390/diagnostics13111903.

[17] W. by: G. D. Luca, "How relu and dropout layers work in cnns," Baeldung on Computer Science, https://www.baeldung.com/cs/ml-relu-dropout-layers. (accessed Apr. 28, 2025).

[18] A. Sanapala, A. Sanapala, and P. Naveena, "A Review On Image Based Diagnosis Using ResNet-50," Issue 1 International Journal of Research and Analytical Reviews, vol. 11, no. 1, p. 796, 2024, Accessed: Apr. 27, 2025. [Online]. Available: https://www.ijrar.org/papers/IJRAR24A3421.pdf

[19] S. K. Mathivanan et al., "Employing deep learning and transfer learning for accurate brain tumor detection," Scientific Reports, vol. 14, no. 1, Mar. 2024. doi:10.1038/s41598-024-57970-7

APPENDIX

TABLE II
ARCHITECTURE DETAILS OF CUSTOMCNN

Layer Type	Output Shape	Parameters	Description
Input	(3, 224, 224)	0	Three-channel grayscale input
Conv2D (3×3)	(32, 224, 224)	320	32 filters, stride 1, padding 1
BatchNorm2D	(32, 224, 224)	64	Normalizes activations
ReLU	(32, 224, 224)	0	Activation function
MaxPool2D (2×2)	(32, 112, 112)	0	Downsamples feature maps
Dropout (p=0.25)	(32, 112, 112)	0	Regularization
Conv2D (3×3)	(64, 112, 112)	18,496	64 filters, stride 1, padding 1
BatchNorm2D	(64, 112, 112)	128	Normalizes activations
ReLU	(64, 112, 112)	0	Activation function
MaxPool2D (2×2)	(64, 56, 56)	0	Downsamples feature maps
Dropout (p=0.25)	(64, 56, 56)	0	Regularization
Conv2D (3×3)	(128, 56, 56)	73,856	128 filters, stride 1, padding 1
BatchNorm2D	(128, 56, 56)	256	Normalizes activations
ReLU	(128, 56, 56)	0	Activation function
MaxPool2D (2×2)	(128, 28, 28)	0	Downsamples feature maps
Dropout (p=0.25)	(128, 28, 28)	0	Regularization
Flatten	(100352,)	0	Flattens feature maps
Linear	(512,)	51,384,576	Fully connected layer
ReLU	(512,)	0	Activation function
Dropout (p=0.5)	(512,)	0	Regularization
Linear	(4,)	2,052	Output layer with 4 classes

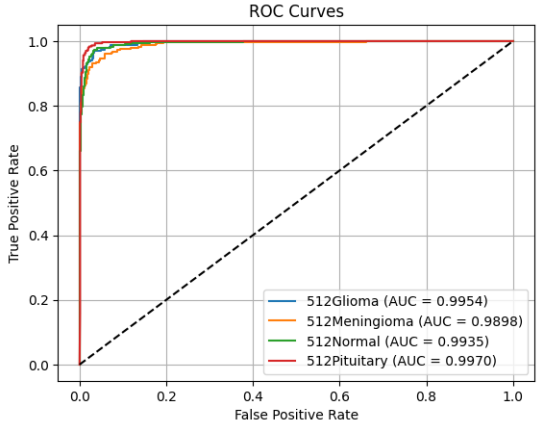

Fig. 4. ROC Curve of Alexnet

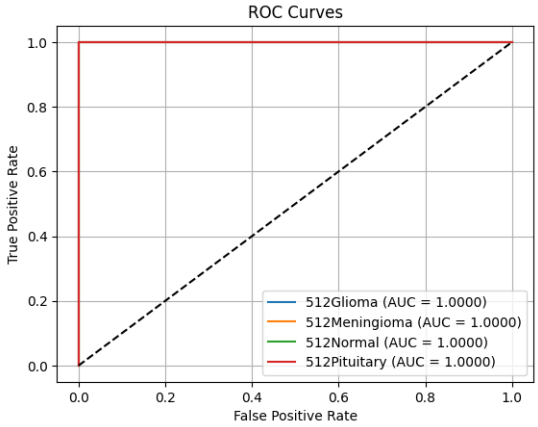

Fig. 5. ROC Curve of Densenet-169

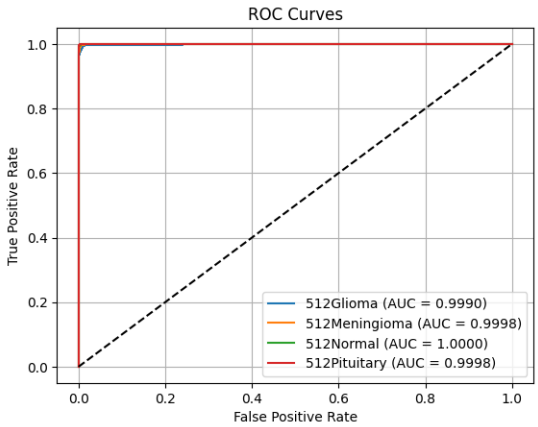

Fig. 6. ROC Curve of Resnet-50

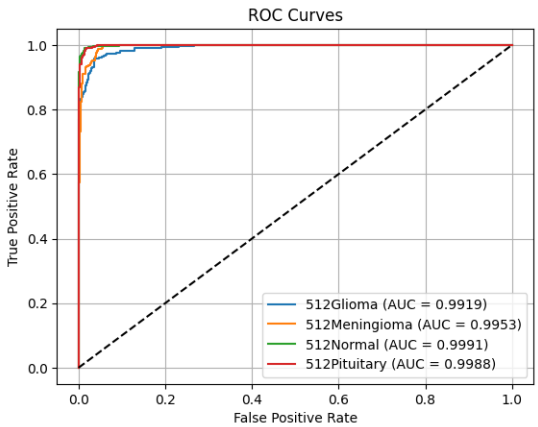

Fig. 7. ROC Curve of ViT