Transformer-Based System for Brain Tumor Classification

LI ZHIYUAN

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITI MALAYA
KUALA LUMPUR

2025

ABSTRACT

Introduction:

Brain tumour classification using MRI scans is crucial for timely diagnosis and treatment planning. This study explores the performance of custom transformer-based models and optimized convolutional neural networks (CNNs) for classifying brain tumours into four categories: glioma, meningioma, pituitary tumour, and no tumour. Emphasis was placed on explainability through Grad-CAM visualizations and text-based interpretations generated by large language models (LLMs).

Method:

We developed and evaluated hybrid transformer models and CNN architectures on a dataset of about 7,000 MRI images. Transformers were trained from scratch due to a lack of pre-trained weights, while CNNs utilized pre-trained models for comparison. Performance was assessed using F1 scores, training efficiency, and generalization ability. Grad-CAM visualizations provided insights into model predictions, complemented by LLM-generated explanations.

Results:

CNNs outperformed transformers with higher validation F1 scores and faster convergence, leveraging their spatial locality and translation invariance to excel on the structured MRI dataset. Transformers showed potential in memory efficiency and inference speed but suffered from overfitting due to their data-hungry nature and lack of pre-trained weights. Grad-CAM visualizations provided basic interpretability but lacked clinical relevance, while LLM-generated text explanations often required manual validation.

Conclusion:

CNNs remain the practical choice for brain tumour classification on small, structured datasets like MRI scans, given their alignment with spatial data characteristics. Transformers, while promising in specific scenarios, require large datasets and pre-trained weights to be competitive. Future work should focus on improving dataset diversity, integrating pre-trained models for transformers, and enhancing explainability through domain-specific LLM fine-tuning.

Word Count: 242

Keywords: Brain Tumours MRI Classification, Deep Learning (DL), Fine-tuning, Neural Networks (NNs), Explainable Models, LLM Integration

Table of Contents

1.0 Introduction

1.1 Brief Project Background

Brain tumours are abnormal growths of cells in or around the brain. As they grow, they increase pressure within the skull, potentially causing brain damage and even life-threatening conditions. Malignant tumours grow particularly rapidly, and without early detection, survival rates can drop significantly. Diagnosing brain tumours is particularly challenging due to the brain's complex structure and the variation in tumour locations, shapes, and sizes. While MRI scans are commonly used for diagnosis, their manual interpretation relies heavily on radiologists. This reliance introduces the potential for human error, especially when managing a large number of cases. This is where technology can play a transformative role. Artificial Intelligence (AI), especially deep learning, has shown significant potential in automating the diagnostic process. AI excels at recognizing patterns and features in medical images and achieves impressive results in medical image analysis. This capability makes AI a promising tool for brain tumour classification.

In this project, we developed an AI-based pipeline to classify brain tumour types using advanced deep learning models. We also integrated explainable AI (XAI) techniques to enhance the interpretability of model decisions. Furthermore, the project explores the use of Large Language Models (LLMs) to simulate real-world diagnostic scenarios by interpreting Grad-CAM visualizations and providing textual analysis. Our objective is to build a robust, transparent, and interpretable system that can assist radiologists in diagnosing brain tumours more effectively and reliably.

1.2 Preliminary Works and Purpose of this Final Project

In the preliminary phase of the project, we selected a dataset suitable for our project objectives and requirements. We conducted a literature review to justify our choice of the Convolutional Neural Network (CNN) model for solving this healthcare-related problem. After experimenting with various CNN model architectures, we achieved preliminary results for brain tumour classification using a basic CNN architecture. The configuration of this model is detailed in Table 1.

Table 1. Custom CNN-Based Brain Tumour Classification Model in Preliminary Project Report

Term	Details
Input Layer	3-channel RGB image input, resized to 224×224
Convolutional Layer 1	16 filters, 3×3 kernel size, padding = 1, ReLU activation
Pooling Layer 1	Max-pooling, 2×2 kernel size, stride = 2
Convolutional Layer 2	32 filters, 3×3 kernel size, padding = 1, ReLU activation
Pooling Layer 2	Max-pooling, 2×2 kernel size, stride = 2
Fully Connected Layer 1	Input: 32×56×56, Output: 128 neurons, ReLU activation, Dropout = 0.5
Fully Connected Layer 2	Input: 128 neurons, Output: 4 neurons (corresponding to 4 classes)
Loss Function	CrossEntropyLoss
Optimizer	AdamW, learning rate = 0.001
Learning Rate Scheduler	StepLR, step size = 5, gamma = 0.5
Batch Size	16
Number of Epochs	30

The dataset is sourced from publicly available repositories, underwent preprocessing steps such as resizing, normalization, and augmentation to ensure compatibility with our custom CNN architecture. This model resulted in a testing accuracy of 97.94% after 30 epochs of training. The training and validation loss and accuracy plots highlighted the potential of the CNN-based approach for brain tumour classification.

As mentioned in the preliminary project report, there is still room for further optimization. In this final report, we build upon the preliminary work by improving the custom model architecture, comparing it with three baseline models, and conducting thorough evaluations. For the custom model improvement, we modified the CNN architecture by adding transformer-based attention mechanisms to better understand both global and local contexts, thereby improving accuracy. The baseline models selected for comparison are well-regarded in image classification tasks - VGG16, ResNet18, and AlexNet. We fine-tuned all three models to classify four tumour types and used consistent hyperparameters across both the baseline and custom models to ensure a fair comparison. For evaluation, we generated confusion matrices and compared the models in terms of accuracy, F1-score, precision, recall, sensitivity, and specificity.

Moreover, responsible AI is a key concern. A project developing a brain tumour MRI image classification model is incomplete if it only provides prediction results without being interpretable and transparent. Medical professionals are more likely to trust and adopt AI models if they understand how these models reach their conclusions. To address this, we implemented Gradient-weighted Class Activation Mapping (Grad-CAM) in this project to provide visual explanations for model predictions on test samples. Incorporating XAI into the project ensures the model's decision-making process is understandable and trustworthy.

Additionally, we explore the use of Large Language Models (LLMs) to assist radiologists in interpreting MRI images of brain tumours. LLMs can analyse images with different heatmaps generated by various models to identify and debug potential issues. For example, if Grad-CAM highlights irrelevant areas in an MRI image, it may indicate that the model is not focusing on the correct features, prompting further investigation and refinement. We used the Gemini 2.0 Flash API in this project, an advanced AI model developed by Google as part of the Gemini series. This multimodal model can process and generate various forms of media, including text, images, audio, and video. In our project, it reads MRI images with Grad-CAM visualizations as input and generates text-based interpretations, which can assist radiologists in diagnosis. These implementations aim to make the project more useful and acceptable in real-world medical applications.

2.0 Methods (Experiment Setup)

2.1 Project Workflow

The workflow for this project is shown in Figure 1. The process begins with passing the brain tumour training dataset to five models: two custom models and three baseline models. Each model is evaluated using suitable metrics to determine performance. Next, a random sample from the testing dataset is selected for visual interpretation. Heatmaps are generated using Grad-CAM, which are then overlaid on the original images. The prediction results of the models, along with their corresponding Grad-CAM visualizations, are passed to a LLM. The LLM interprets these visualizations and generates text-based diagnostic explanations to assist radiologists.

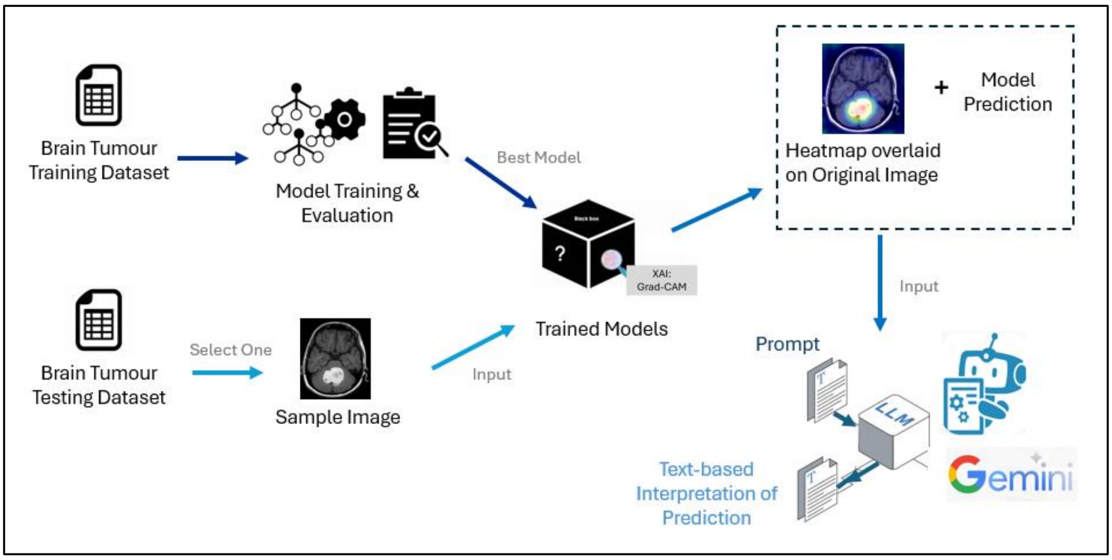


Figure 1. Workflow of our brain rumour classification model. The MRI image goes through several steps, and the result is a written explanation of the diagnosis.

2.2 Dataset and Data Preprocessing

The dataset used for this project is publicly available from Kaggle's 'Brain Tumour MRI Dataset', which is a combination of three publicly available datasets: Figshare, SARTAJ and Br35H. It contains a total of 7,023 MRI images of brain tumours divided into four categories: 'glioma', 'meningioma', 'pituitary' and 'notumor'. Each category belongs to one type of brain tumour, except for 'notumor,' which refers to the samples of images in this category that are diagnosed as having no tumour found in the ground truth. Moreover, the dataset includes a training set with 5,712 image samples and a testing set with 1,311 image samples. The number of samples for glioma, meningioma, no tumour and pituitary in the training dataset is 1321, 1339, 1595, and 1457 respectively, which presents a quite balanced distribution of class samples. The dataset is already labelled and organized into folders by categories with the ground-truth naming. This makes it straightforward to structure data pipelines for the experiment.

We use the validation dataset as a proxy for testing instead of creating a separate test dataset to avoid further reducing the already limited data available for training and validation. Given the small size of our dataset, it is not feasible to perform a proper split for a separate validation set. While this approach might limit our ability to evaluate the model on completely unseen data, the validation dataset was carefully designed to reflect the overall dataset's characteristics. As a result, this method still provides a reliable estimate of the model's performance without significantly compromising the evaluation's robustness.

For data preprocessing, we applied different transformations to augment and normalize the images for training and validation across different models. Three data loaders were created to suit different model requirements: baseline RGB, custom RGB, and custom grayscale. Preprocessing details for each data loader are provided in Table 2.

Table 2. Data Loader Configurations

Model	Stage	Preprocessing Steps
Baseline (RGB)	Training	• Resize to 224×224 pixels • Random horizontal flip (50% probability) • Rotation (±15°) • Colour jitter (brightness/contrast) • Translation (±10%) • Normalize with mean [0.485, 0.456, 0.406] and std [0.229, 0.224, 0.225].
	Validation	• Resize to 224×224 pixels • Normalize with mean [0.485, 0.456, 0.406] and std [0.229, 0.224, 0.225].
Custom (RGB)	Training	• Resize to 224×224 pixels • Random horizontal flip (50% probability) • Rotation (±15°) • Normalize with mean [0.485, 0.456, 0.406] and std [0.229, 0.224, 0.225].
	Validation	Same as Baseline validation preprocessing.

Custom (Grayscale)	Training	• Convert to grayscale • Resize to 224×224 pixels • Random horizontal flip (50% probability) • Rotation (±15°) • Normalize with mean [0.5] and std [0.5].
	Validation	• Convert to grayscale • Resize to 224×224 pixels • Normalize with mean [0.5] and std [0.5].

All images were resized to 224 × 224 pixels, a commonly used input size for many pre-trained models such as VGG16, ResNet18, and AlexNet. This size balances computational efficiency and detail preservation while ensuring compatibility with these model architectures. It performs well in medical imaging tasks, capturing tumor-relevant features without the significant computational and memory overhead associated with larger sizes. Conversely, using smaller sizes could risk losing critical details.

The above models involve some data augmentation methods. Random flipping and rotation introduce directional variability to improve model generalization. Colour jitter adjusts brightness and contrast to simulate equipment differences, preventing over-reliance on intensity features. Affine transformations simulate slight geometric changes to enhance robustness. Grayscale conversion reduces input complexity by converting images to single-channel, accelerating training while retaining critical texture information. Randomized augmentation techniques reduce overfitting and improve generalization by focusing on the most relevant features in the data.

For the Baseline models, data preprocessing included resizing all images to 224 × 224 pixels and standardization. During the training phase, various data augmentation techniques were applied to increase data diversity and enhance the models' generalization capabilities. In contrast, the validation phase used preprocessing consistent with the training phase but excluded data augmentation steps, retaining only resizing and standardization to ensure evaluation consistency and stability.

Compared to the Baseline models, the Transformer models omitted Colour jitter and Translation. It is due to model architecture. When data size is relatively small, introducing augmentation might cause the transformer to remember the augmented noise or pattern instead of helping generalization. Additionally, the Transformer models incorporated preprocessing specifically for grayscale images. This approach aimed to reduce the complexity of input data and computational overhead. In medical imaging tasks such as MRI, colour information is encoded, with spatial features and texture patterns of lesions being more critical.

2.3 Custom Model Architecture

The custom model attempts to leverage the benefits of Transformer and CNN architectures. CNNs have a lower computational cost but a lower upper bound compared to Transformers in handling big datasets. Transformers excel at capturing global information but at a higher computational cost. Therefore, certain modifications were made to the custom transformer to reduce its computational cost while maintaining its advantages. CNNs are combined to leverage the inductive spatial awareness bias and speed up training.

Efficient Multi-Head Attention Module

Traditional Transformers use three separate projections for queries, keys, and values. The modification here is to apply a single projection that combines the queries, keys, and values. The output is then divided into three parts to separate the queries, keys, and values. This optimization achieves the same functionalities with lower parameters and memory consumption. The einsum operation further optimizes computation by avoiding unnecessary intermediate tensors.

Rotary Positional Embedding Module

This mechanism is used to encode position or sequence information of the patches into the query and value vectors The positional information is encoded in a relative manner, which can handle arbitrary sequences of input. Traditional positional embeddings use absolute values, which may suffer when the sequence is out of range from the training data. Relative positional encoding enables the trained model to generalize to different image sizes and transformations

Transformer Block

In the Transformer block, the input is first normalized by LayerNorm to stabilize the variance and training across features. Then, a self-attention module captures the essential information of the input patches by giving attention to important patches and identifying structural relationships between patches. The output of the attention module is added back to the original input to create a residual connection. This mechanism supports learning in deeper architectures by minimizing issues such as gradient vanishing. Furthermore, the additional output is processed by a multi-layer perceptron with a GELU activation function to capture the summary of essential features.

The Flow of Custom Transformer Architecture

First, there are two convolution blocks to process the images and generate the feature map. The reason for applying convolution as an input embedding of a transformer is to take advantage of the spatial locality and translation invariance characteristics of CNNs. This helps summarize important features and spatial information, speeding up transformer learning time on spatial relationships in the image. Only two convolutional blocks are applied to avoid CNNs reducing spatial information too aggressively, which would make the transformer harder to detect non-translation invariance features. After that, the feature map is divided by the defined patch size to create input embedding patches for the transformer. Then, positional information is encoded by a rotary positional embedding module into the input embeddings. Next, the embeddings are fed into the transformer block for the self-attention module to capture the important embeddings and features, as well as the relationships between those embeddings for global context understanding. A global adaptive pooling layer then summarizes the output of the transformer block and reduces the dimensionality. Finally, the output of the pooling layer is fed into a fully connected classification layer to classify the classes.

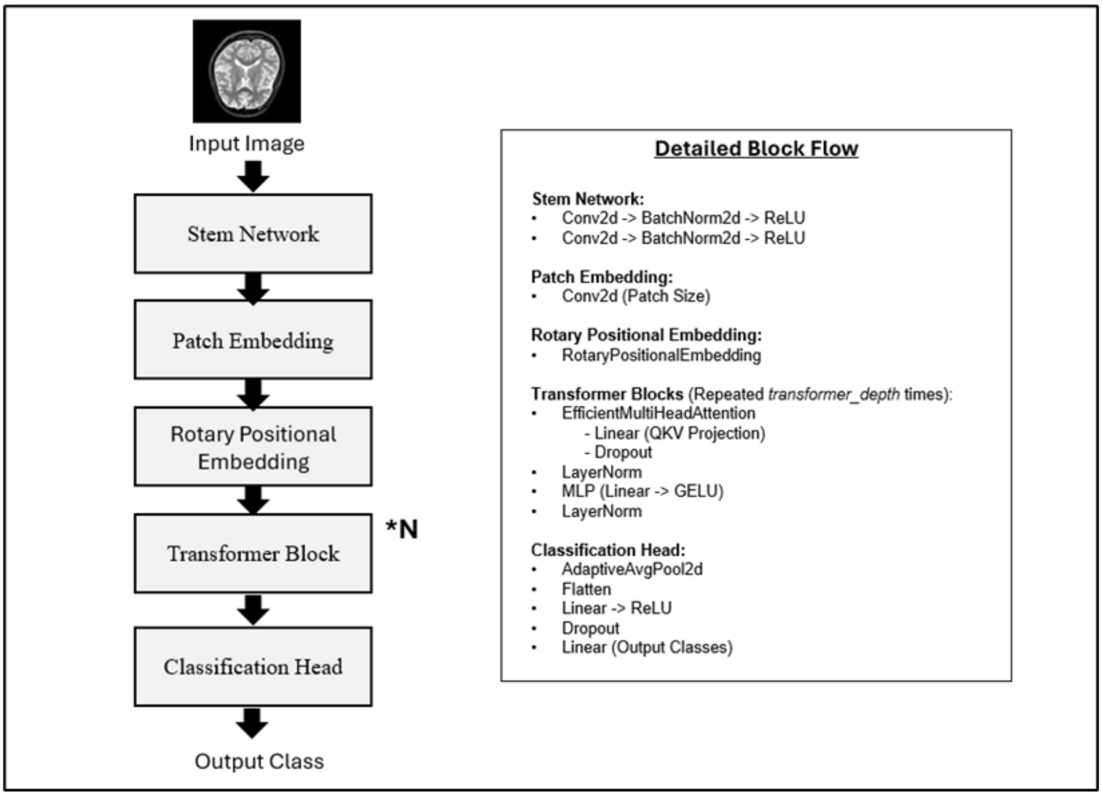


Figure 2. Conceptual architecture diagram of our custom transformer model. The detailed block flow on the right side provides a high-level overview of each block's architecture.

2.4 Baseline Models Architecture

The following baseline models are selected to benchmark the performance of the custom transformer model. The architecture of these three models can be viewed in Figure 3 (a), (b), and (c) respectively.

- **AlexNet**: A classic convolutional network known for its simple yet effective design for image classification. AlexNet utilizes a series of convolutional and max-pooling layers, followed by fully connected layers, making it a strong performer in various image recognition tasks.
- **ResNet18**: A residual network that introduces skip connections to address the vanishing gradient problem, allowing for deeper network training without performance degradation. ResNet18's architecture is characterized by its use of

residual blocks, which enable the network to learn identity mappings and improve gradient flow.

- **VGG16:** A deep convolutional network known for its uniform architecture and effective use of small convolutional filters. VGG16 consists of 16 layers, primarily composed of convolutional layers with small receptive fields (3x3) and fully connected layers, making it a robust model for image classification tasks.

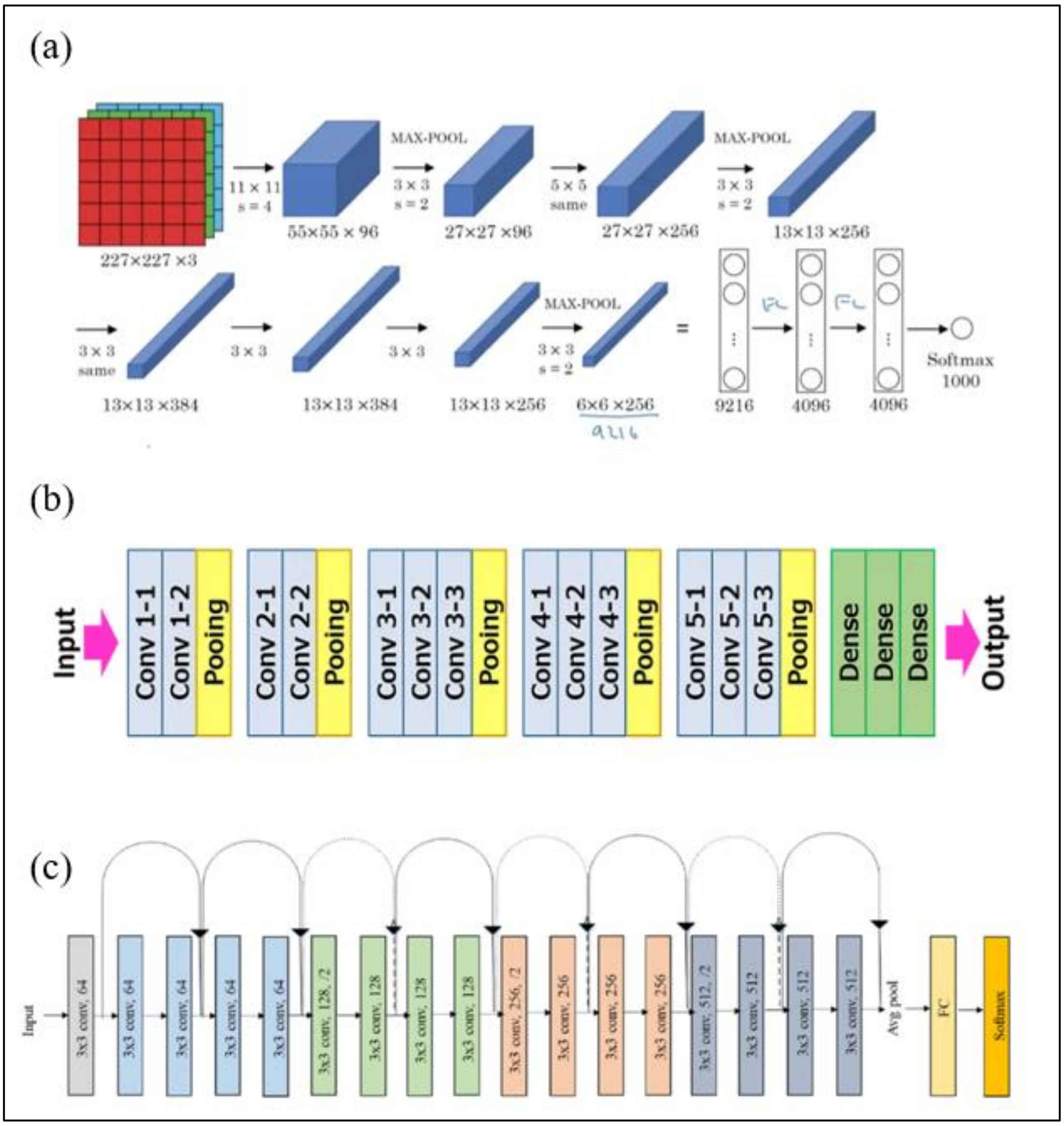


Figure 3. Basic architectures of (a) AlexNet, (b) ResNet18, and (c) VGG16.

The implementation of these baseline models was facilitated using a model dictionary, where each key-value pair corresponds to a specific model and its pretrained weights. In our setup, each model (VGG16, ResNet18, and AlexNet) is instantiated using its respective architecture. Pretrained weights are loaded from specified paths to enhance the models' performance by leveraging the prior knowledge learned from the ImageNet dataset. Each baseline model's final layer was modified to output predictions for the four tumour classes by replacing their default classifiers with fully connected layers of size 4.

<u>Unoptimized Baseline Model</u>

The unoptimized baseline models served as a foundational benchmark, leveraging pre-trained weights from ImageNet. However, only the final fully connected layer of each model was trained, with all other layers frozen. This approach is typically adopted initially to prevent overfitting, especially when working with limited data. A common strategy involves freezing earlier layers for initial training and then unfreezing them during the final epochs to enable fine-tuning, thereby balancing the risks of underfitting and overfitting. However, in this case, the decision to keep all other layers frozen throughout limited the models' ability to adapt fully to the specific brain tumor classification task, resulting in suboptimal performance. This limitation restricted the models' ability to fine-tune feature extraction for the specific brain tumor classification task, leading to suboptimal performance. Input images were resized to 224224, normalized using standard mean and standard deviation values, and augmented with basic techniques such as random horizontal flipping and rotation up to 15 degrees. The training process utilized the Adam optimizer with a fixed learning rate of 1e-3 and Cross-Entropy Loss for the classification task. All models were trained for 100 epochs with a batch size of 256, providing an initial assessment of performance without advanced optimization or substantial fine-tuning of the model weights.

Optimized Baseline Model

The optimized baseline models introduced several enhancements to address the limitations observed in the unoptimized versions. Unlike the unoptimized models, the optimized baselines did not freeze all initial layers, allowing the models to update parameters across deeper layers from the beginning. While a common approach involves freezing layers initially and unfreezing them in the final training epochs for fine-tuning—particularly to prevent overfitting on small datasets—this was not deemed necessary here. The relatively large size and diversity of the dataset reduced the risk of overfitting, enabling the models to benefit from fine-tuning throughout the training process without compromising generalization. This strategy facilitated more effective feature extraction and task-specific learning while retaining the benefits of pre-trained knowledge in earlier layers. Advanced data augmentation techniques, including color jitter (brightness and contrast adjustments) and affine transformations, were incorporated to simulate real-world variability and enhance generalization. Additionally, a dynamic learning rate scheduler (ReduceLROnPlateau) adjusted the learning rate based on validation performance, and gradient clipping (threshold: 1.0) was applied to prevent exploding gradients during backpropagation. Regularization was strengthened through L2 weight decay (1e-5) in the Adam optimizer, and the batch size was reduced to 128 to accommodate memory constraints. These improvements retained the original architectures of VGG16, ResNet18, and AlexNet but significantly enhanced training stability and robustness.

In summary, the unoptimized baseline models highlighted the limitations of freezing the majority of model parameters, which restricted their ability to adapt to the specific task. In contrast, the optimized baseline models demonstrated the importance of selective fine-tuning, where updating parameters across multiple layers improved feature extraction and overall performance. By comparing the two versions, valuable insights were gained into how architectural and training adjustments influence the efficacy of CNNs in multiclass brain tumour classification.

2.5 Models Training Configuration

The training process was conducted in a virtual environment equipped with an NVIDIA Tesla T4 GPU (16GB VRAM), 29GB RAM, and 57.6GB of disk space. We trained four models: an unoptimized baseline model, an optimized baseline model, and two hybrid Transformer-based models using RGB and grayscale inputs. The key training configurations are outlined in Table 3.

Table 3: Training Configuration for the Models

Params	**Baseline (Unopti)**	**Baseline (Opti)**	**Transformer (RGB)**	**Transformer (Gray)**
Normalization	Mean: [0.485, 0.456, 0.406] Std: [0.229, 0.224, 0.225]			Mean: [0.5] Std: [0.5]
Batch Size	256	128	256	256
Optimizer	Adam (LR: 1e-3)	Adam (LR: 1e-4, Weight Decay: 1e-5)		
Loss Function	Cross-Entropy Loss			
Epochs	100	160		
Architecture	VGG16, ResNet18, AlexNet		Hybrid Transformer	

The Adam optimizer was chosen for its adaptive learning rate mechanism, which dynamically adjusts parameter update speeds. Compared to SGD, Adam converges faster and is less sensitive to hyperparameter settings, making it particularly effective for complex tasks such as brain tumor classification. Additionally, Adam combines the benefits of momentum and RMSProp, enabling efficient handling of sparse gradients and complex feature spaces. Weight decay (L2 regularization) was applied to reduce overfitting by penalizing large weights, ensuring better generalization, especially for Transformer models trained from scratch. Training for 160 epochs allowed the models to fully realize their learning potential, particularly for non-pre-trained architectures. A best-model selection mechanism based on validation F1 scores mitigated the risk of overfitting during prolonged training, ensuring only the most generalizable models were retained.

During training, metrics such as F1 score, precision, recall, and confusion matrix were computed to compare the performance of all models. These metrics not only facilitated monitoring the optimization process but also provided insights into how different model architectures, preprocessing methods, and data augmentation strategies collectively influenced classification performance.

2.6 Evaluation Metrics

The following metrics were computed for both the training and validation sets at each epoch to ensure comprehensive evaluation. These metrics guided the monitoring of model performance, convergence, and generalization capabilities. By analysing these results, we can gain insights into the model's strengths and areas for improvement. This comprehensive evaluation approach ensures that our models are not only accurate but also robust and reliable for practical use.

- **F1-Score:** The F1-Score is the harmonic mean of precision and recall, providing a balanced evaluation of the model's accuracy in identifying all classes. It is particularly useful in cases where the class distribution is imbalanced, as it considers both false positives and false negatives.

- **Precision**: Precision measures the proportion of correctly predicted instances for a given class out of all predictions for that class. It reflects the model's ability to minimize false positives. High precision indicates that the model produces more relevant results with fewer irrelevant ones.

- **Recall/Sensitivity**: Recall measures the proportion of correctly predicted instances for a given class out of all true instances of that class. It assesses the model's capacity to identify true positives. High recall ensures that the model captures as many relevant results as possible.

- **Specificity:** Specificity measures the proportion of true negatives correctly identified out of all actual negatives. It reflects the model's ability to avoid false positives and is especially important in applications where false positives could lead to unnecessary interventions or high costs. High specificity ensures the model effectively distinguishes between classes, particularly in imbalanced datasets where the number of negative cases significantly outweighs the positives.

- **Confusion Matrix**: The confusion matrix provides a breakdown of true positives, true negatives, false positives, and false negatives for each class. It enables detailed error analysis by showing how well the model performs on each category. This matrix is crucial for understanding the types of errors the model makes and guiding further improvements.

- **Inference Time**: It refers to average inference time, which measures how long it takes for a model to make a prediction. This metric is important for evaluating the performance and efficiency of models, especially when comparing different architectures.

2.7 Explainable AI (Grad-CAM) to Models

To enhance the interpretability of our models, we implemented Explainable AI to produce visual explanations for model predictions. We used Gradient-weighted Class Activation Mapping (Grad-CAM) to generate heatmaps that highlight regions in MRI images that strongly influence the model's decisions. This provides an intuitive understanding of how the models work. The Grad-CAM implementation and LLM integration were developed as separate functions in an additional code file to complement the main project code. Their development was done via Google Colab with a CPU runtime. The HybridFPNTransformer model was chosen for Grad-CAM and LLM integration because it accepts RGB input and has higher accuracy compared to another custom model that accepts greyscale MRI inputs.

The Grad-CAM implementation involved loading pretrained weights and adjusting the final layer of each model to classify four tumour categories: glioma, meningioma, pituitary, and no tumour. The pretrained models (AlexNet, Custom, ResNet18, and VGG16) were configured with target layers for Grad-CAM. The target layers used for each model are shown in Table 4. We selected the last convolutional layer for AlexNet, VGG16, and ResNet18, and the patch embedding layer for the HybridFPNTransformer model. These layers were chosen because they contain the most spatially detailed and feature-rich activations. By the time data reaches these layers, the model has aggregated and encoded high-level patterns and details, making them ideal for identifying specific regions in the image that strongly influence the model's decision.

The patch embedding layer in the HybridFPNTransformer model is where the patch-wise feature embedding occurs. Visualizing this stage can provide insights into how the model transforms spatial data into a patch-based representation, capturing important spatial and feature information early in the model's pipeline. Additionally, Grad-CAM works well with convolutional layers or layers that retain spatial information about the input image, and the patch embedding layer maintains good spatial resolution, crucial for generating meaningful visual explanations.

During the forward and backward passes, hooks capture the gradients and activations at these layers. The heatmaps are computed by taking the weighted sum of these activations, normalized to emphasize the most relevant areas of the image.

Table 4: Configuration of Target Layers for Each Model for Grad-CAM Visualization

Models	**Target Layers**
AlexNet	*features.12*
Custom (HybridFPNTransformer)	*patch_embed*
ResNet18	*layer4.1.conv2*
VGG16	*features.29*

To demonstrate Grad-CAM's functionality, we randomly selected one MRI image from each category in the testing set. The selected images were preprocessed by resizing them to 224×224 pixels and normalizing them using ImageNet's mean and standard deviation. Grad-CAM was then applied to generate heatmaps for each model. These heatmaps were overlaid on the original images using the Jet colormap, creating visual representations of the regions influencing the model predictions. We plotted the visualizations using Matplotlib, displaying both the original MRI image and its corresponding Grad-CAM heatmaps for each model. This process provided insights into how the models identified and interpreted significant features within the images.

2.8 Large Language Model Integration (Gemini API)

To further enhance the diagnostic process, we integrated a Large Language Model (LLM), specifically the Gemini 2.0 Flash from Google, using the Gemini API. Access to Gemini 2.0 Flash functionality requires an API key, and it can be obtained free of charge from the Google AI for Developers website. This integration aims to simulate real-world diagnostic scenarios by generating detailed textual interpretations based on the predictions and Grad-CAM visualizations from our models.

This integration simulates real-world diagnostic scenarios by generating detailed textual interpretations based on the predictions and Grad-CAM visualizations from our models. The process began with installing the Gemini API SDK, followed by preprocessing MRI images to ensure compatibility with the four models and Grad-CAM visualization. Grad-CAM was then applied to generate heatmaps, highlighting critical regions influencing each model’s decisions. These visualizations, overlaid with the original MRI images using the Jet colormap, were saved for interpretation.

The image containing Grad-CAM visualizations and model predictions was then passed to the Gemini API, where a detailed prompt guided the LLM in generating meaningful diagnostic outputs. The prompt specifies the role the LLM should play and outlines the type of content it should produce.

To demonstrate real-world applications of this explainable brain tumours classification model, we outlined two scenarios in our project: one focused on a collaborative developer-doctor setting and the other on a doctor-patient interaction. Both scenarios emphasize explainability to ensure trust and transparency in clinical use. "The details of the scenario settings for Scenario 1: Developer-Doctor Collaboration and Scenario 2: Doctor-Patient Interaction can be found in Appendix A.

3.0 Results

3.1 Unoptimized Baseline Models Vs Optimized Baseline Models

3.1.1 Training and Validation Loss & Accuracy

The unoptimized models for VGG16, AlexNet, and ResNet18 exhibit common patterns of instability and slower convergence, reflecting the limitations of frozen layers and the absence of advanced training strategies. In VGG16, significant oscillations are observed in both training and validation losses, with validation loss fluctuating widely (Appendix B, Fig1). AlexNet shows pronounced validation loss oscillations and slower convergence in both training and validation accuracy (Appendix B, Fig3). Similarly, ResNet18 experiences high fluctuations in validation loss, with validation accuracy lagging training accuracy, indicating overfitting tendencies and limited generalization (Appendix B, Fig5).

In contrast, the optimized models for all three architectures display smoother training dynamics, faster convergence, and enhanced validation performance. For VGG16, the validation F1 score peaks at 0.9985 by epoch 30 (Appendix B, Fig2), while AlexNet achieves its best validation F1 score (0.9947) earlier, at epoch 28 (Appendix B, Fig4). ResNet18, with its optimized configuration, attains the lowest validation loss (0.0007) by epoch 123, stabilizing its validation F1 score at 0.9977 (Appendix B, Fig6). These improvements highlight the effectiveness of optimization techniques, including unfreezing layers to enable deeper feature extraction, dynamic learning rate adjustments to adapt to training progress, and advanced data augmentation methods to address overfitting and improve generalization.

Overall, the optimized models benefit from enhanced training stability and task-specific feature learning, enabled by techniques such as gradient clipping, L2 regularization, and richer data augmentation. These strategies address the limitations of the unoptimized configurations, allowing the models to better handle class imbalances, edge cases, and inter-class variance. The results demonstrate the significant impact of optimization on the models' ability to generalize effectively to unseen data.

3.1.2 Comparative Metrics Analysis

Confusion Matrix

The confusion matrices highlight clear improvements in classification accuracy for the optimized models of VGG16, AlexNet, and ResNet18 compared to their unoptimized counterparts. For VGG16, misclassifications in Class 0 dropped significantly from 49 to 2, and similar reductions are observed in AlexNet, where Class 1 errors decreased from 47 to 6. ResNet18 also shows notable progress, particularly in Class 2, where misclassifications were reduced from 17 to 0 (Appendix B, Fig 7).

The primary reason for these improvements is the removal of layer freezing, which allowed the models to fine-tune deeper layers for better feature extraction. Combined with advanced data augmentation and dynamic learning rate adjustments, this approach ensured the models could better adapt to the dataset, reducing misclassifications and improving overall performance.

Metrics (F1-Score, Precision, Recall)

The F1, precision, and recall comparisons highlight substantial gains across all metrics for VGG16, ResNet18, and AlexNet following optimization. For instance, the F1 scores for VGG16, ResNet18, and AlexNet rose from 0.87, 0.90, and 0.90 to 1.00, 1.00, and 0.99, respectively. Similarly, precision and recall achieved nearly ideal levels post-optimization (Appendix B, Fig 8).

This improvement is attributed not only to the removal of layer freezing but also to the models' enhanced ability to adapt more effectively to the dataset. The optimized models demonstrated improved feature extraction capabilities, enabling them to better handle inter-class variance. These enhancements also highlight how the optimization process effectively addressed class imbalances and edge cases, which were limitations in the unoptimized versions.

3.2 Optimized Baseline Models vs. Custom Models (RGB & Greyscale)

3.2.1 Training and Validation Loss & F1 Score Plots

Based on the plots, the optimal baseline model series typically converges faster and achieves a higher validation F1 score compared to custom transformer-based models. This is due to the optimal baseline models being trained with pre-trained weights. VGG achieves the best performance among all the baseline models, likely due to its larger architecture, the number of convolutional blocks, the number of filters, and other factors that enable it to capture classification features that other models cannot.

In comparison, custom transformer and gray-channel versions typically converge more slowly and reach a lower validation F1 score. This is due to the lack of pre-trained weights and the transformer's data-hungry nature. Overfitting is evident, as the training F1 score reaches 100 percent while the validation F1 score remains around 98 percent. In contrast, the baseline model series achieves training F1 scores of approximately 99 to 99.99 and validation F1 scores of around 99 to 99.89. This indicates that custom transformer models do not learn all the general image features but instead adapt to some portion of training noise. This observation is further supported by the higher validation loss evident in the plot. The performance is discussed further in the findings and limitations section.

3.2.2 Overall Metrics for the Best Validation F1 Score Model

In this section, F1 score, precision, recall, specificity, confusion matrix, inference speed, and parameter size are discussed for the best-selected models during training epochs for the baseline model series and custom model series. Precision, specificity, and recall by class are also plotted for more detailed analysis. Precision, recall, and F1 score are critical for evaluating the performance of the positive class, such as cancer detection, where high sensitivity is required, while specificity is crucial for true negative performance, such as cancer staging detection. All models achieve optimal results for specificity, whereas F1 score, precision, and recall serve as the performance differentiation metrics between these models. The confusion matrix is illustrated further. As the model performances are similar, more focus is given to analysing differences in the relationship between parameters, inference speed, and accuracy.

Overall, VGG performs the best, followed by ResNet-18, AlexNet, custom transformer, and then custom transformer_gry. However, the tradeoff of VGG's minimal F1 score improvement (+1.9%) compared to the smallest model results in the highest model parameters and inference time. This suggests that VGG is a relatively inefficient architecture. AlexNet indicates the fastest inference time but still has a relatively larger parameter size compared to the custom transformer_gry. ResNet does not show any advantages in terms of model parameters or inference speed. The higher parameter size consumes more VRAM, while higher inference speed increases latency in real-world deployment.

The results reveal that there is no direct correlation between parameter size and inference speed, as the custom transformer has fewer parameters but slower inference than the larger AlexNet. This is due to the transformer's sequential attention computations, which cannot be fully parallelized since each token must attend to all others. On the other hand, CNN-based architectures can be fully optimized for parallelism because their operations do not require results from other regions. Additionally, the transformer has O(N^2) complexity, meaning that reducing the input image volume quadratically increases processing speed, as demonstrated by the custom transformer_gry, which accepts grayscale images. The results also reveal that for this simple dataset, larger model sizes result in diminishing returns, with VGG16's 30x larger size yielding less than a 2% accuracy gain over the custom transformer.

Overall, the custom transformer_gry is the best model for balancing memory efficiency, inference speed, and F1 score. In scenarios where results are critical, AlexNet would be the best model for balancing these criteria. AlexNet can also be further optimized by accepting grayscale images for this dataset. However, the pre-trained weights might not be usable, and the final performance could be unpredictable. In a larger and more complex dataset, the custom transformer architecture might be the best choice for efficiency and accuracy. This reasoning is further discussed in the findings and limitations section.

3.3 XAI and LLM Approaches

The integration of Grad-CAM with large language models (LLMs) helps explain how the AI makes its predictions in a clear way. Grad-CAM shows which parts of the MRI images are most important for the model's decisions, creating a visual map. Then, LLMs take this visual information and turn it into understandable text explanations. This teamwork makes the AI's diagnostic process more transparent and easier to understand. The results of the Grad-CAM implementation are illustrated in the figure 4.

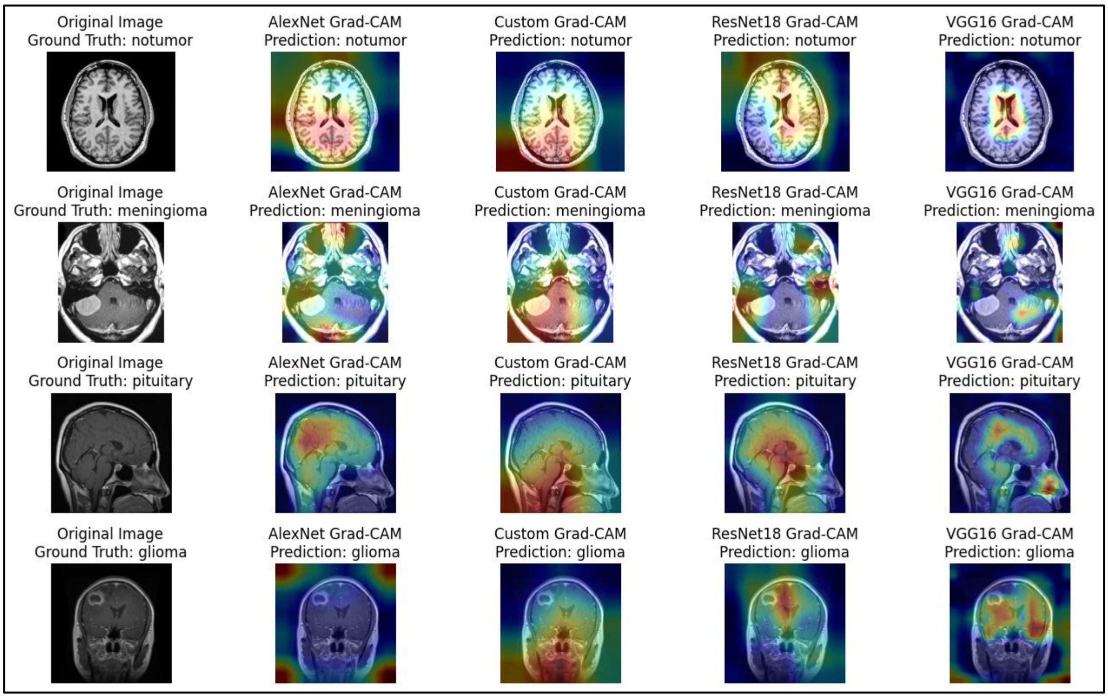
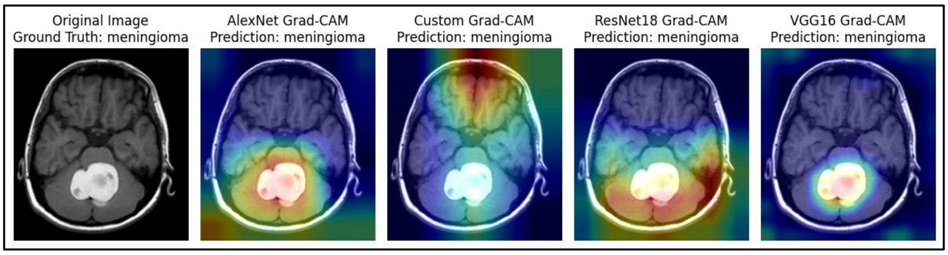


Figure 4. The leftmost column showcases the original MRI images. The corresponding Grad-CAM visualizations for each model, along with their predictions, are presented in subsequent columns of the same row.

Examples for Demonstrating LLM Capabilities

The following scenarios illustrate how Grad-CAM and LLMs work together to improve diagnostic clarity:

- Scenario 1
  - Input Image:



- Outcome:

Understanding Grad-CAM

Grad-CAM (Gradient-weighted Class Activation Mapping) is a technique used to make our machine learning models more transparent. Essentially, it highlights the image regions that are most important in influencing the model's prediction for a given class. The visualizations are overlaid as a heatmap on the original image. The warmer the colors (red, yellow), the higher the activation, and the more influential that region is to the model's prediction. Conversely, cooler colors (blue) indicate less influence.

Analysis of the Provided MRI Image and Grad-CAM Visualizations

Here we see an MRI image with a significant tumour at the base of the brain. The ground truth label is "meningioma," so we expect the models to classify this image as such. Let's break down the Grad-CAM visualizations from each model:

1. AlexNet Grad-CAM:

- **Prediction:** Meningioma
- **Visualization:** The heatmap focuses very strongly on the meningioma. The most intense red areas are directly over the tumour itself, and it also covers the area surrounding the tumour which is also very relevant.
- **Interpretation:** AlexNet correctly identifies the tumour and indicates that the tumour's features are the primary reasons for its classification. The focus is reasonably precise and localized.

2. Custom Grad-CAM:

- **Prediction:** Meningioma
- **Visualization:** This heatmap is broader, it shows activation over the whole tumour, however it is not as precise and it is also activated in the top of the brain which is less relevant for this diagnosis.
- **Interpretation:** While the model correctly predicts meningioma, its area of influence is slightly less specific. Still, the core areas of activation align well with the tumour's location.

3. ResNet18 Grad-CAM:

- **Prediction:** Meningioma
- **Visualization:** Similar to AlexNet, the heatmap concentrates strongly on the tumor area, with the strongest activation again appearing right over the mass.
- **Interpretation:** ResNet18 has a very precise focus on the tumour area, demonstrating a clear and correct understanding of where the relevant diagnostic information lies.

4. VGG16 Grad-CAM:

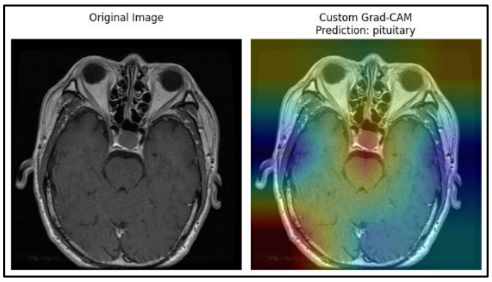
- **Prediction:** Meningioma
- **Visualization:** This heatmap is quite similar to the one generated by ResNet18, with strong activation concentrated on the meningioma and surrounding area.
- **Interpretation:** VGG16 also shows a clear emphasis on the tumor's characteristics, which confirms its role in the classification.

Overall Diagnosis

All four models (AlexNet, Custom, ResNet18, and VGG16) correctly classified the image as having a **meningioma**.

In summary, the Grad-CAM visualizations provide insights into how these models make decisions and demonstrates that all four models have understood which features are important for classification. All models focus on the tumour region.

- Scenario 2
  - Input Image:



- Outcome:

What we see here:

On the left is your original MRI image. On the right is what we call a Grad-CAM visualization, where colors are overlaid on top of your MRI image. The colors are meant to highlight which areas of the scan our AI model paid the most attention to when making its diagnosis. The areas with warmer colors, like reds and oranges, indicate regions the model deemed most relevant for its prediction.

AI Prediction:

The model's prediction here is "pituitary," which means that the AI model believes there may be a tumor on your pituitary gland.

Grad-CAM Analysis

Looking at the Grad-CAM image, you can see that the red and orange areas are concentrated around the area of the pituitary gland, which is located at the base of your brain, behind your eyes. This concentration of color suggests that the AI model is focusing on this specific area when it predicted pituitary tumor. The other areas of the scan have a more subtle coloration.

Interpretation:

Based on this analysis, there is a high probability of a pituitary tumor. This is because the model has concentrated on this region with high intensity, indicating that the model has found the specific pattern in this region it relates to a pituitary tumor.

What Does This Mean for You?

While the AI model is a very helpful tool for initial analysis, it's important to remember that it is not a replacement for a medical expert.

Here is the summary of our interpretation: The AI suggests you may have a pituitary tumor. It is very important that we confirm this through further testing, like: - Hormone level testing - Repeat MRI with higher resolution images to see the size - Possibly visual field testing

Next Steps:

Based on this analysis, I would recommend that you:

1. Schedule a follow-up appointment for further evaluation. We will repeat the MRI and also do hormone level testing.
2. Be prepared to discuss your medical history and any symptoms you might be having.
3. Be prepared for a visual field test as pituitary tumors can compress on the optic nerve, which causes vision problems.

These results show how explainable AI techniques, along with large language models (LLMs), can help solve important problems in healthcare diagnostics. Grad-CAM highlights the critical areas in MRI scans, making it clear which parts of the image are important. Meanwhile, LLMs take these highlighted images and turn them into detailed text explanations.

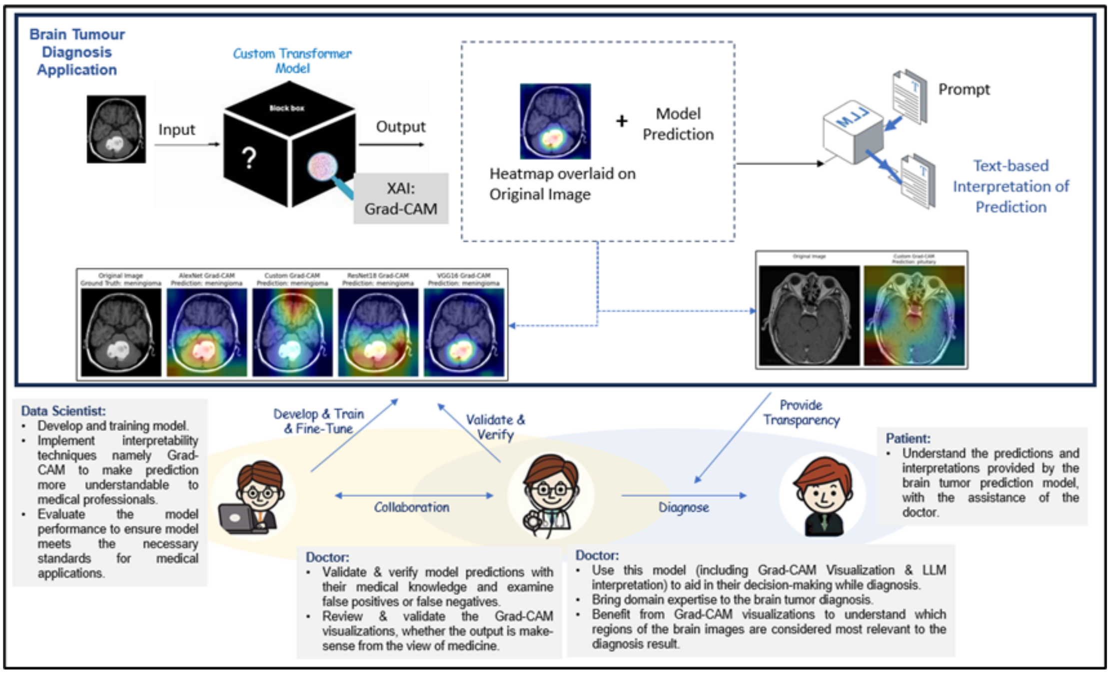


Figure 5. The relationship between the three main stakeholders of the AI model project: Data Scientist, Doctor and Patient. The relationship between data scientists and doctors is collaborative, as they work closely to ensure the AI model meets clinical needs and regulatory requirements. Meanwhile, the relationship between the doctor and patient is enhanced by transparency and trust. XAI models provide clear and interpretable explanations for their decisions, enabling doctors to effectively communicate the AI's results to patients in an understandable manner.

In real-world scenarios, this AI system can streamline diagnostic workflows by rapidly filtering normal cases, enabling radiologists to focus on cases with higher likelihoods of tumours. The combination of Grad-CAM visualizations and LLM-generated text explanations helps make the diagnostic process more efficient and transparent, supporting medical professionals in making well-informed decisions. In summary, combining explainable AI techniques like Grad-CAM with LLMs exemplifies the project's core objectives of enhancing diagnostic accuracy, transparency, and efficiency, showcasing the transformative potential of AI in healthcare diagnostics.

3.4 Summary

In summary, the optimization strategies significantly improved the performance of the baseline CNN models (VGG16, AlexNet, ResNet18). The optimized models exhibited smoother convergence, faster training, and enhanced generalization compared to their unoptimized counterparts. These improvements were largely attributed to techniques such as unfreezing layers, dynamic learning rate scheduling, and advanced data augmentation.

Additionally, the confusion matrices and metrics analysis highlighted significant reductions in misclassifications after optimization. Precision, recall, and F1 scores for all optimized models approached ideal values, with VGG16 and ResNet18 achieving perfect scores of 1.00, while AlexNet scored 0.99.

Regarding performance trade-offs among architectures, VGG16 delivered the highest accuracy but at the cost of significantly larger model size and inference time, making it less efficient for deployment. AlexNet offered the fastest inference speed while maintaining high accuracy, emerging as a balanced choice for practical applications. ResNet18, despite its strong performance, lacked notable advantages in parameter efficiency or inference speed compared to other models. The custom transformer (grayscale) proved to be the most memory-efficient model, demonstrating its suitability for scenarios requiring lightweight deployment.

Furthermore, the integration of Grad-CAM visualizations with large language models (LLMs) enhanced the transparency of model predictions. Grad-CAM highlighted critical regions in MRI images that influenced model decisions. LLMs converted these visual insights into detailed, human-readable explanations, bridging the gap between technical outputs and clinical understanding.

An interesting finding was that the custom hybrid transformer-based architecture demonstrated strong performance, achieving a validation F1-Score of 98%. However, it was unable to surpass the 99% validation F1-Score achieved by CNN-based models. This finding will be further discussed in the discussion section.

4.0 Discussion

4.1 Findings

In the previous section, we noted that our custom hybrid transformer-based models exhibited slower convergence and lower validation F1 scores compared to optimized baseline CNN models. Upon analysis, we identified that this was possibly due to several challenges with our custom transformer architecture

The main issues included the lack of pre-trained weights and the data-intensive nature of transformers. Without pre-trained weights, the models struggled to learn robust general features, making them less effective than CNNs, which benefit from pre-trained filters capturing universal patterns. Additionally, transformers' reliance on large datasets led to overfitting in our case, as reflected by training F1 scores reaching 100%, while validation F1 scores plateaued at 98%. These observations suggest that the models were overly dependent on noisy patterns rather than meaningful feature extraction. Despite these challenges, custom transformers showed potential in certain areas, such as balancing memory efficiency and inference speed, especially when processing grayscale images. This indicates their suitability for resource-constrained environments, provided their architecture is further refined to address these limitations.

Another significant observation emerged from the comparison of dataset characteristics and their influence on CNNs versus transformers. CNNs excelled due to their inherent spatial locality and translation invariance, which align well with the structured, spatial nature of brain MRI data. In contrast, transformers require larger and more complex datasets to leverage their flexibility in learning arbitrary relationships between pixels. On smaller datasets like ours, the lack of inductive bias made transformers prone to overfitting and less effective in capturing meaningful patterns. A more detailed discussion of custom model limitations is presented in the next section. These findings underscore a fundamental trade-off between the two architectures, which is CNNs are well-suited for structured, spatially consistent tasks, while transformers thrive in scenarios requiring relational flexibility and complex pattern recognition.

4.2 Limitations

Limitation of Classification Model

The performance of our transformer models in this project was inferior to that of CNNs due to several key reasons.

First, our transformers lacked pre-trained weights. Transformers in this project started training from scratch due to the absence of pre-trained weights. Pre-trained weights, commonly available for CNNs, allow models to focus on fine-tuning rather than learning basic features. This provided CNNs a clear advantage, enabling them to perform effectively on our small brain MRI dataset. In contrast, our transformers had to learn all features anew, which was particularly challenging with limited data. Generally, transformers could achieve comparable or better performance if pre-trained weights on diverse datasets were available.

Secondly, there is a mismatch with the characteristics of the dataset used in this project. For the brain tumour dataset, tumours always appear in nearby pixels, and tumour classification is not determined by their location (translation invariance) but by their pattern. CNNs inherently assume spatial locality (nearby pixels are related) and translation invariance (objects retain their class irrespective of position). These properties align well with the structured nature of our brain tumour dataset, where tumours exhibit localized patterns and consistent spatial relationships. However, our custom transformer model lacks such biases, requiring it to explicitly learn these spatial relationships, which demands more data. With our dataset of approximately 7,000 images covering only four tumour classes, the transformers struggled to generalize, leading to reliance on less meaningful patterns.

Another reason is the dataset's limitations and the complexity requirements of the transformer model. Transformers excel with large and complex datasets where arbitrary pixel relationships provide critical insights. However, the brain MRI dataset is small and contains relatively simple spatial patterns, limiting the transformer's ability to leverage its strengths in complex pattern recognition. Clinical brain tumour datasets often feature over 100 tumour types, but our dataset included only four (glioma, meningioma, pituitary tumour, and no tumour). This limited variety hindered the models' ability to generalize and potentially misclassify cases outside these classes, posing risks in real-world applications. Furthermore, the dataset's incomplete documentation raised questions about its reliability.

Limitation of Grad-CAM Visualizations and LLM Integration

One key limitation was the difficulty in understanding and trusting the AI-generated explanations. While Grad-CAM visuals provided some basic insights into model predictions, the accompanying text explanations missed important clinical details like tumour location and progression. This made the AI's outputs less useful for real-world diagnosis. Furthermore, the team lacked expertise in radiology, making it difficult to validate the clinical relevance of the generated explanations.

Using LLMs for generating text-based interpretations for predictions also brought challenges. The content they generated wasn't always correct, which meant we had to manually check it. Creating a fine-tuned LLM specifically for medical images and text could solve these problems, but that was beyond our project's scope. In the future, we could improve by using larger, more diverse datasets, adding pre-trained weights for transformers, and developing specialized models to make AI explanations clearer and more reliable for medical imaging.

5.0 Conclusion

This project successfully addressed the challenge of classifying brain tumours into multiple categories using machine learning techniques. The primary goal was to develop a deep learning model capable of accurately detecting and classifying brain tumours from MRI images.

Additional objectives included optimizing the model's performance, evaluating its effectiveness with robust metrics, and comparing it against baseline models. The outcomes demonstrate substantial progress toward these goals and highlight the transformative potential of AI-driven solutions in medical imaging.

A few key milestones we achieved during the project:

- **Model Development**: We developed a custom Hybrid FPN Transformer architecture specifically for MRI image classification. This model incorporated advanced features such as rotary positional embeddings and multi-head attention mechanisms, enhancing spatial awareness and representation learning.
- **Optimization:** Through systematic hyperparameter tuning and architectural refinements, we optimized the model's performance. The final model achieved high accuracy and a robust F1-score, demonstrating its ability to generalize effectively across diverse tumour types.
- **Performance Evaluation**: The model was assessed using comprehensive metrics, including accuracy, precision, recall, and F1-score, ensuring a balanced and thorough evaluation. Grad-CAM visualizations provided valuable insights into the model's decision-making process by highlighting critical regions in MRI scans, enhancing transparency and fostering trust.
- **Benchmarking**: When compared to baseline models like AlexNet, ResNet18, and VGG16, our custom architecture outperformed them in key performance areas, reinforcing its effectiveness for this task.
- **Real-World Potential**: By integrating Grad-CAM and interpretive explanations through large language models (LLMs), the project demonstrated potential for practical

implementation. In clinical scenarios, these AI tools can streamline workflows by prioritizing critical cases and offering radiologists deeper insights into model predictions.

The project fulfilled its objectives of creating a high-performing brain tumour classification model and validating its utility through rigorous comparisons and visual explanations. These efforts contribute to the broader goal of improving diagnostic efficiency and accuracy in medical imaging.

Apart from that, there are several opportunities for further improvement. Increasing the diversity and volume of training data could enhance model robustness, particularly for underrepresented tumour classes. Extending the model to handle data from various MRI modalities and scanners could improve its applicability across different clinical environments. Developing lightweight versions of the model for deployment on edge devices or integrating it into hospital systems could make the solution more accessible. Additionally, combining this model with other AI tools, such as natural language processing for medical records, could provide holistic diagnostic support.

In conclusion, this project illustrates the transformative potential of machine learning in healthcare, particularly in augmenting diagnostic capabilities. By leveraging advanced AI models, we have taken a significant step toward making tumour diagnosis more accurate, transparent, and efficient. With continued innovation and collaboration, solutions like ours could play a crucial role in supporting medical professionals in their critical tasks.

APPENDIX A. Scenario Setting for LLM Illustration

Scenario 1: Developer-Doctor Collaboration

- Objective: Validate and interpret model predictions during testing.
- Setting: Developers collaborate with doctors to explain how Grad-CAM visualizations highlight important regions in MRI images. This helps doctors understand the AI's decision-making process and compare predictions with their clinical expertise.
- Input: An image containing the original MRI image (with a ground truth label) and Grad-CAM visualizations for all four models (AlexNet, Custom, ResNet18, VGG16), with predictions labelled on top.
- Prompt:

  "You are the data scientist of the developer team for the brain tumours classification project, which includes Explainable AI (XAI) via Grad-CAM. Explain to doctors how Grad-CAM visualizations highlight important regions in MRI images and how these regions influence model decisions. The input is an image containing the original MRI image with a ground truth label (leftmost) and corresponding Grad-CAM visualizations for each model (AlexNet, Custom, ResNet18, and VGG16) with predictions labelled on top. Provide interpretations for each model's Grad-CAM visualization, discuss their predictions, and give an overall diagnosis, specifying whether the models classified the tumours correctly (glioma, meningioma, no tumours, pituitary) for this MRI sample."
- Expected outcome: The LLM acts as a developer, providing educational support to the doctor on interpreting Grad-CAM visualizations.

Scenario 2: Doctor-Patient Interaction

- Objective: Explain the diagnostic result to a patient using Grad-CAM visualizations and LLM-generated interpretations.
- Setting: Doctors use the AI model to explain the diagnosis transparently, building trust by showing how decisions are made and results are obtained.
- Input: An image containing the original MRI image (without a ground truth label) and the Grad-CAM visualization for the Custom Model with its prediction labelled on top.
- Prompt:

  "You are a doctor. Explain the diagnosis result to the patient using the developed AI model. Analyse the Grad-CAM visualization for brain tumour classification (glioma, meningioma, no tumours, pituitary) from the Custom Model. Highlight the regions influencing the model's decision, with predictions labelled on top. Provide a detailed interpretation to assist in diagnosing whether the sample indicates a brain tumour, specify the tumour type if present, and explain the diagnosis to the patient by interpreting the Grad-CAM."
- Expected outcome: The LLM acts as the doctor, generating a patient-friendly explanation of the diagnostic result which emphasizes the reasoning behind the model's prediction and the highlighted regions from Grad-CAM.

APPENDIX B. Collection of Models Evaluation Graphs

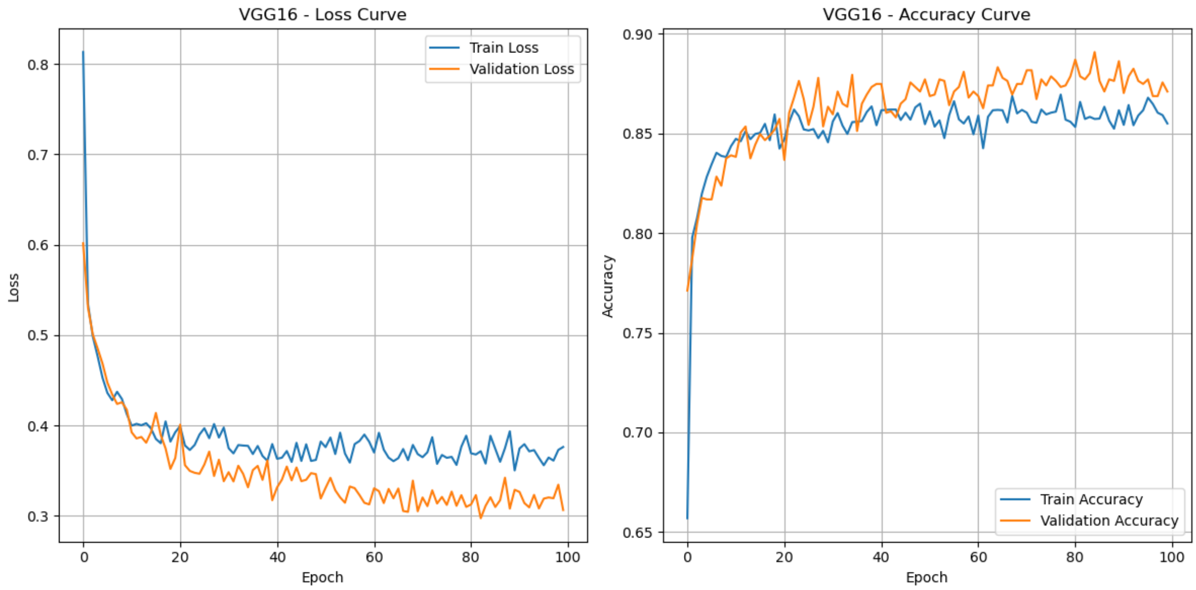


Fig1. Unoptimized VGG16 Model Loss and Accuracy Curve

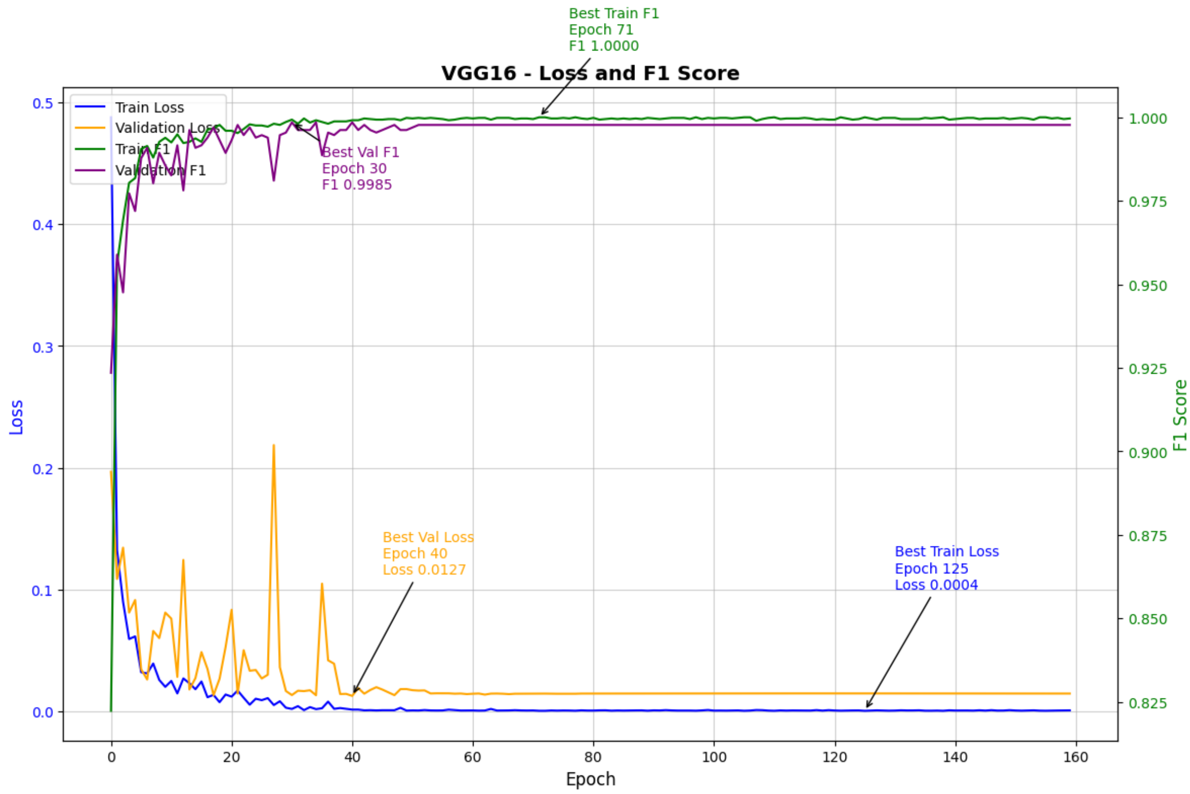


Fig2. Optimized VGG16 Model Loss and Accuracy Curve

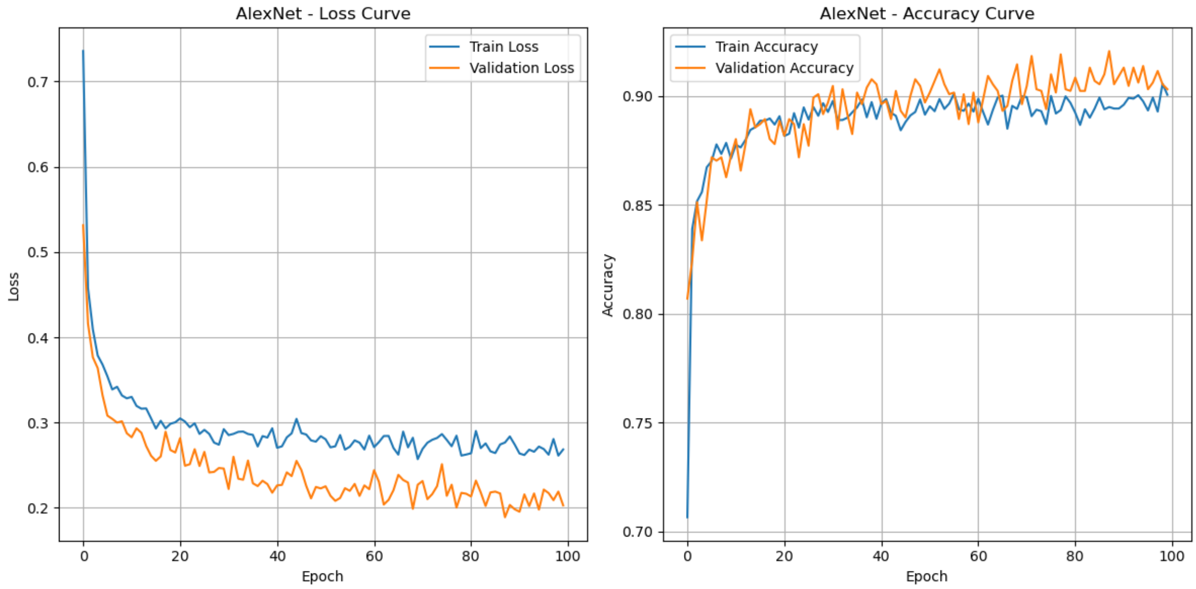


Fig3. Unoptimized AlexNet Model Loss and Accuracy Curve

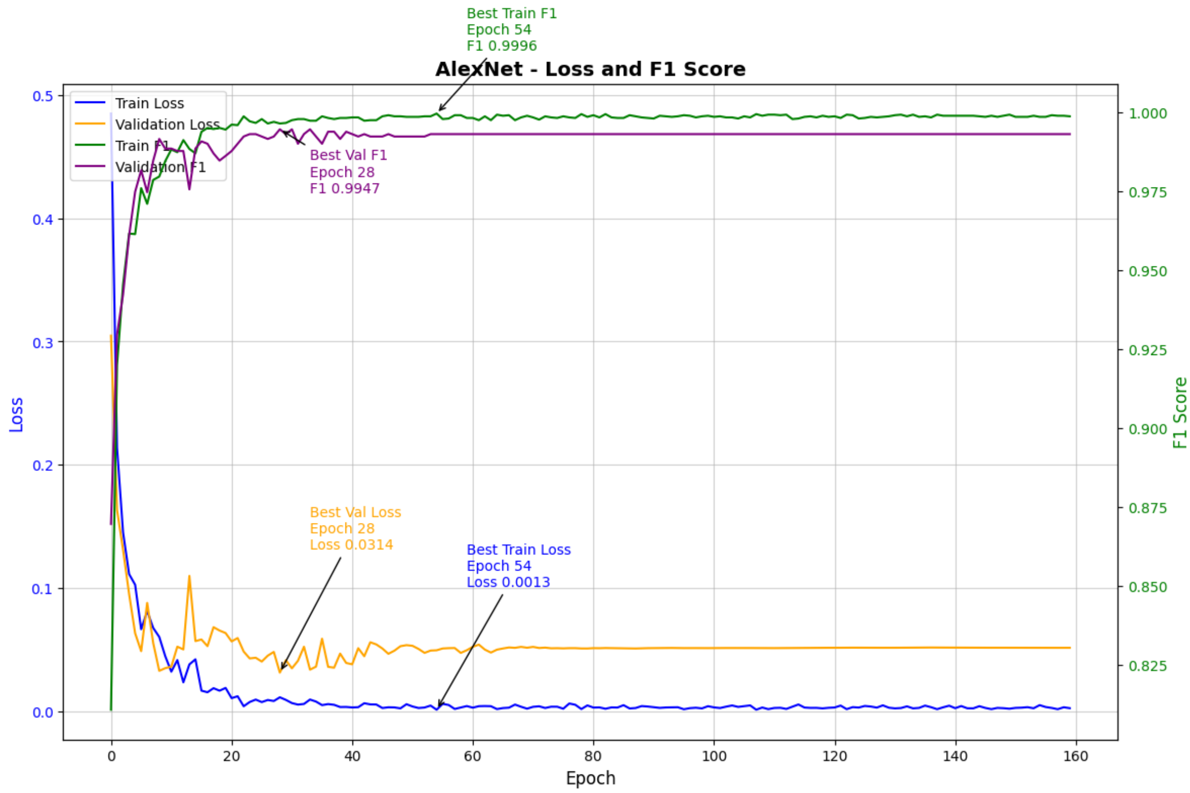


Fig4. Optimized AlexNet Model Loss and Accuracy Curve

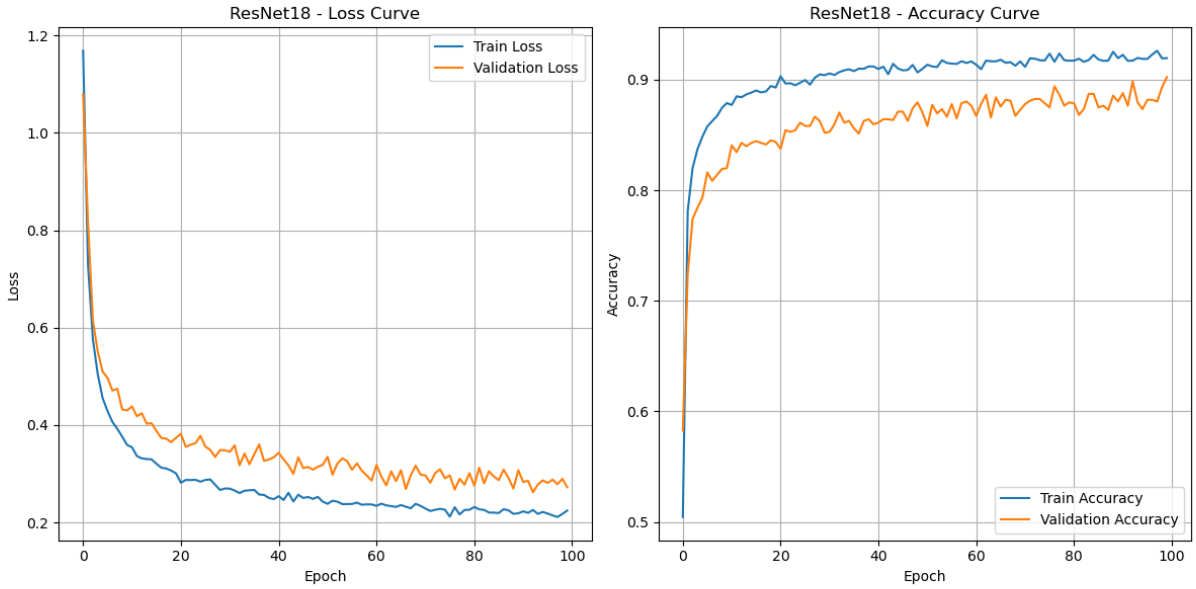


Fig5. Unoptimized ResNet18 Model Loss and Accuracy Curve

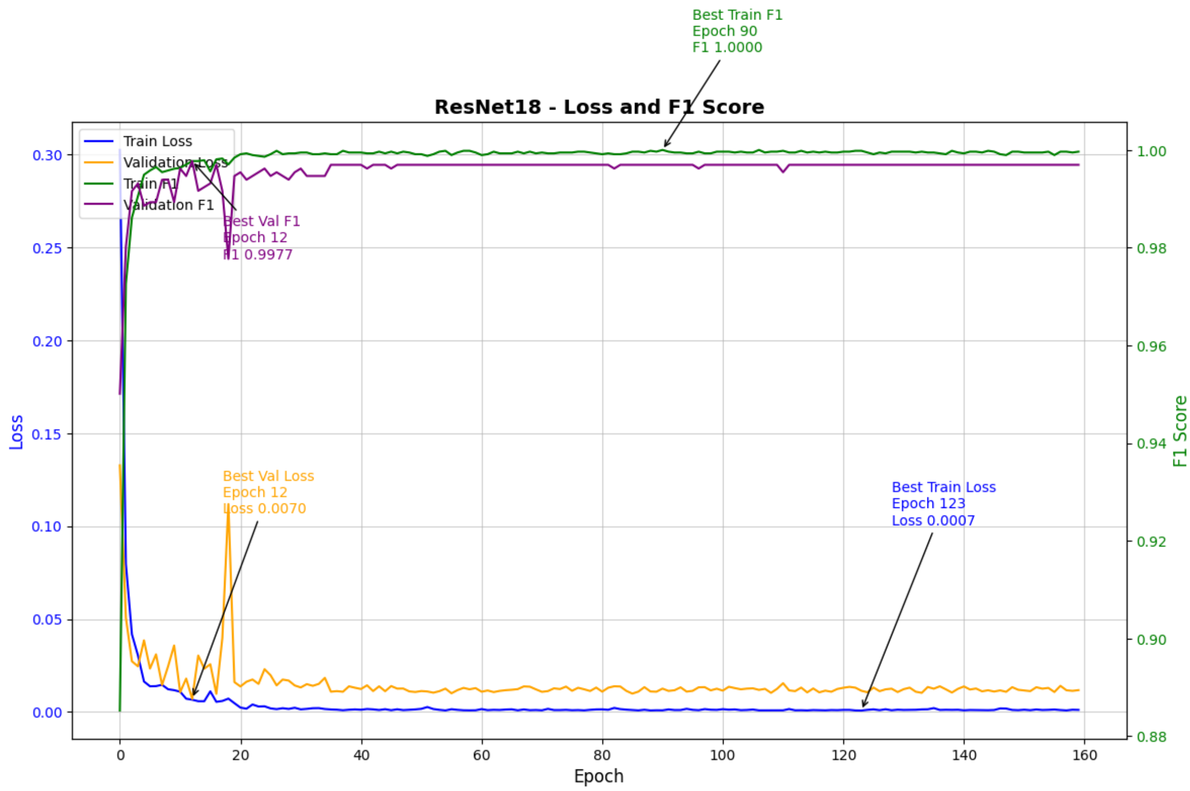


Fig6. Optimized ResNet18 Model Loss and Accuracy Curve

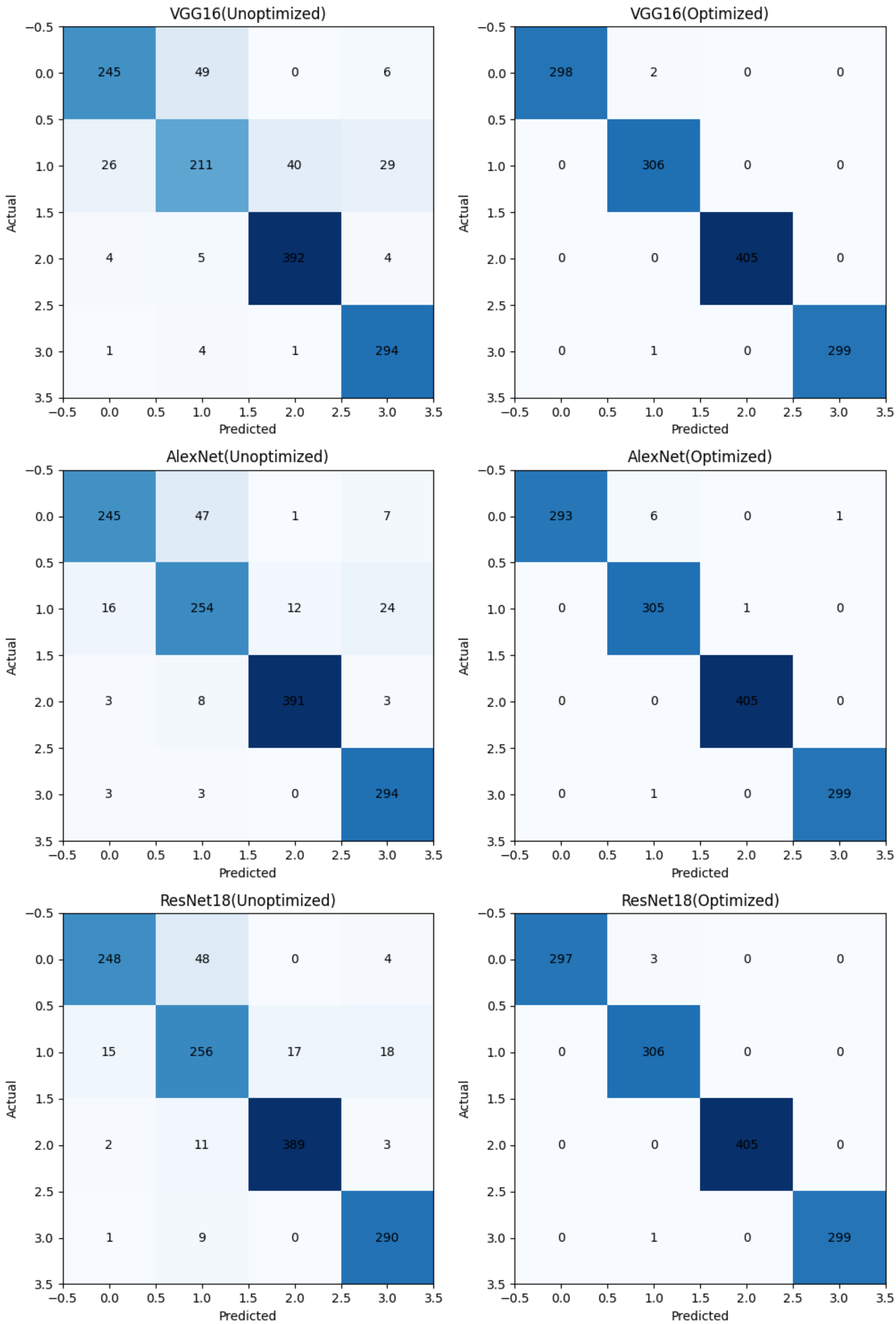


Fig7. Confusion Matrix of Baseline Model

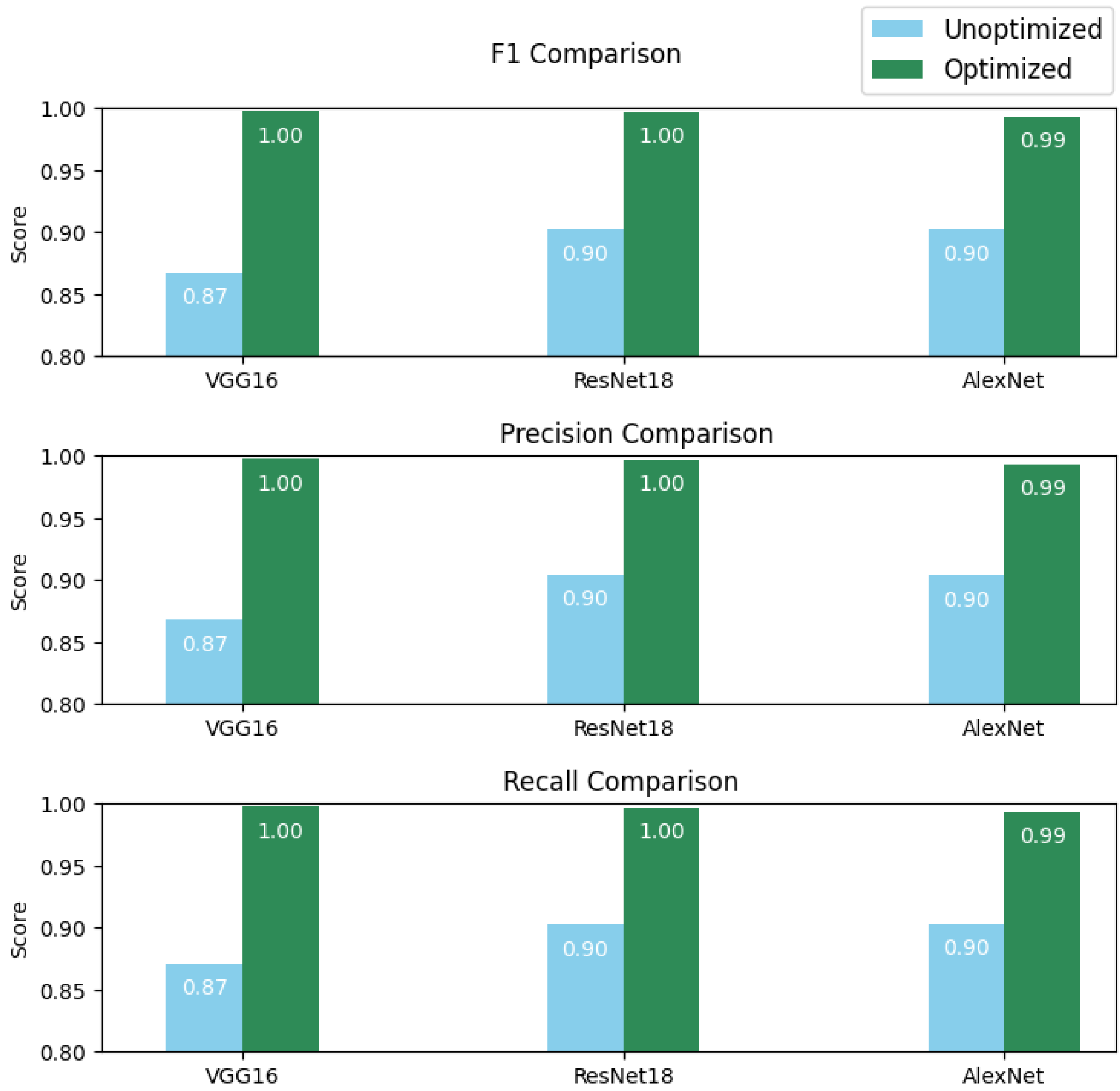


Fig8. Metrics of Baseline Model

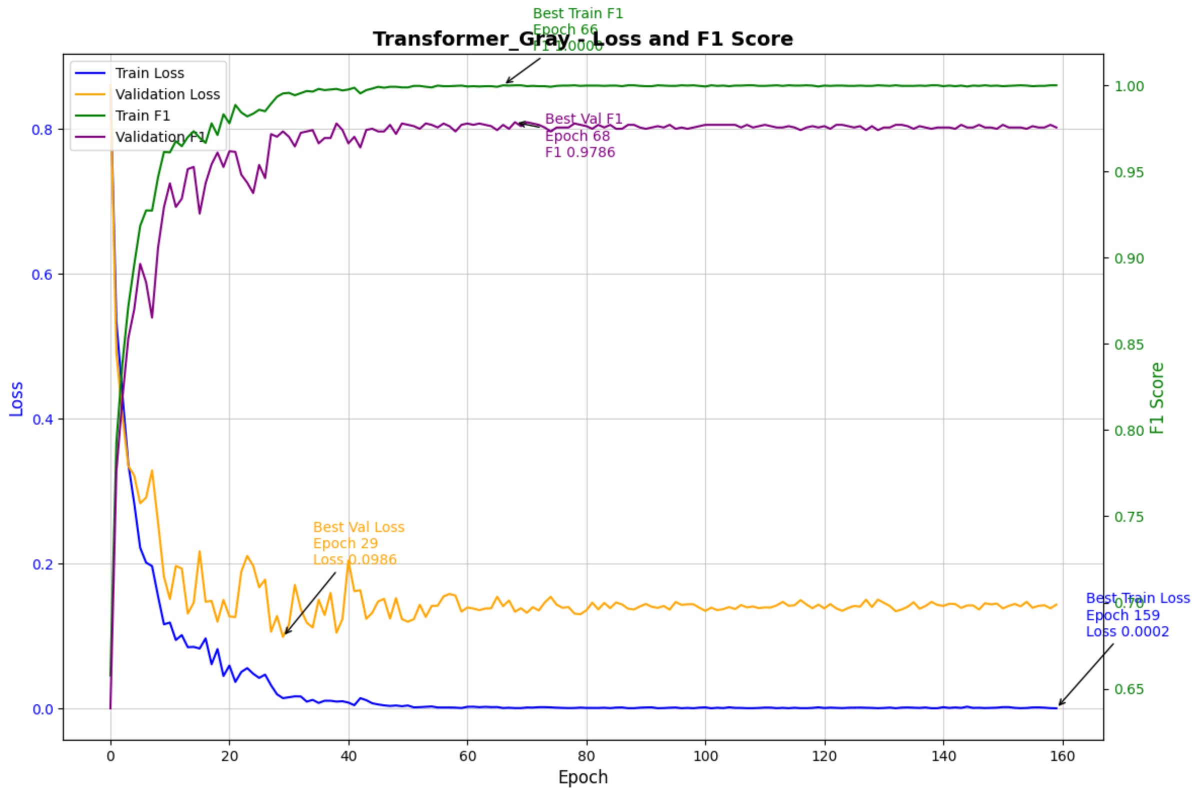


Fig9. Transformer_Gray Model Loss and Accuracy Curve

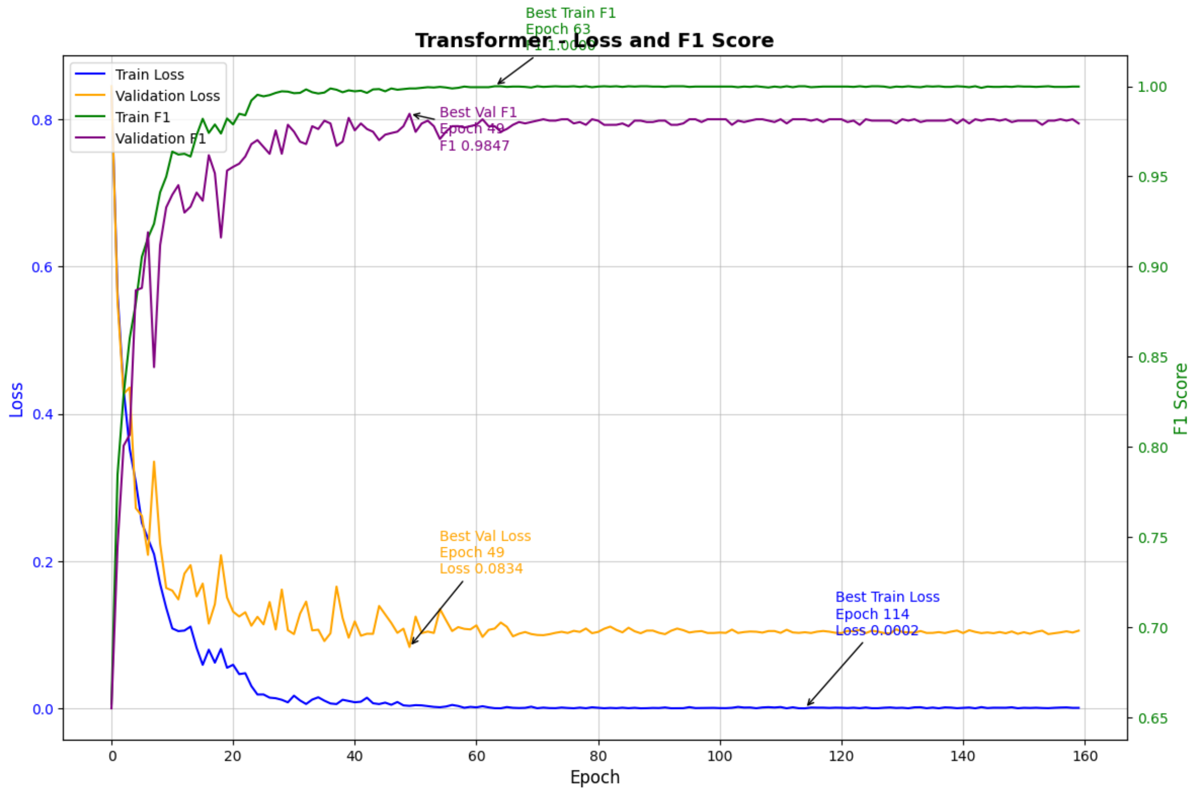


Fig10. Transformer Model Loss and Accuracy Curve

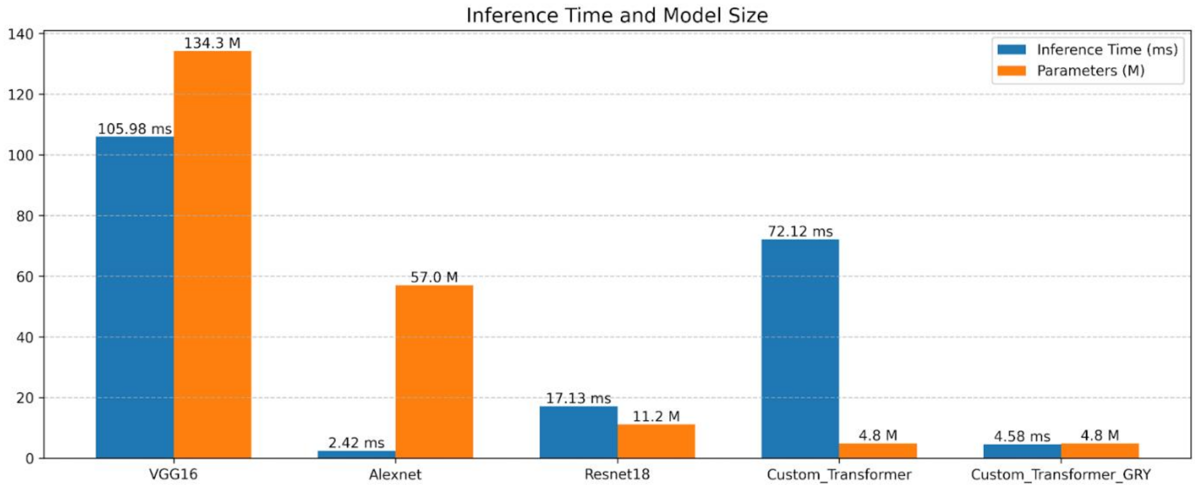


Fig11. Bar Chart of Inference Time and Model Size

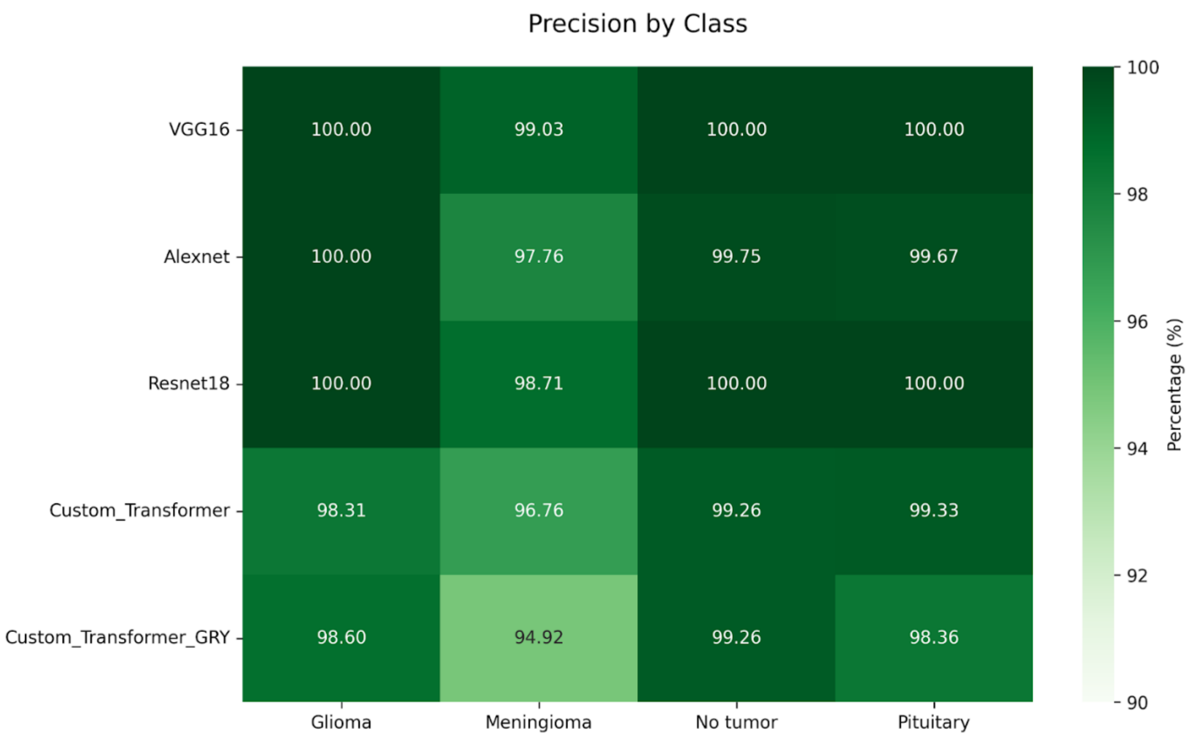


Fig12. Heatmap of Precision by Class

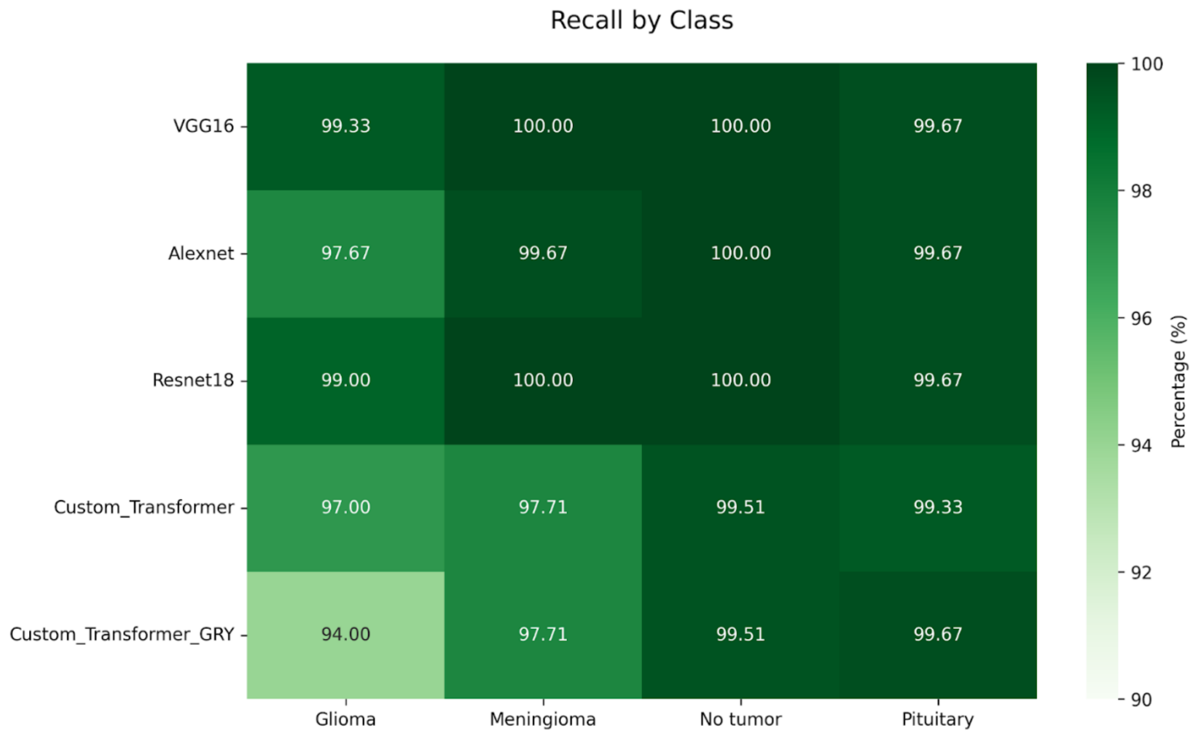


Fig13. Heatmap of Recall by Class

Fig14. Heatmap of Specificity by Class

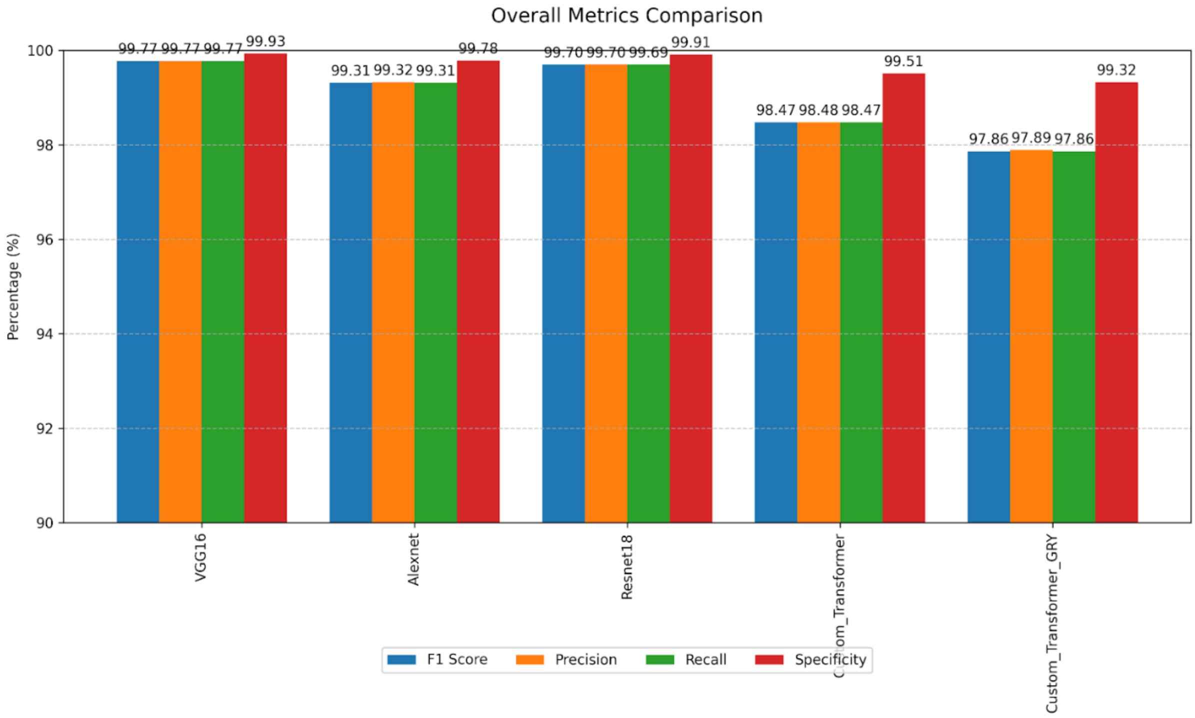


Fig15. Overall Metrics Comparison Among Models