Revolutionizing brain Cancer Detection Using Gen AI

Marwan Mostafa

Supervisor: Dr. Aya Salama

Faculty of Media Engineering and Technology
German University In Cairo
May 2025

Faculty of Media Engineering and Technology
German University in Cairo

Revolutionizing brain Cancer Detection Using Gen AI

Bachelor Thesis

Author:	Marwan Mostafa
Supervisors:	Dr. Aya Salama
Submission Date:	29 May, 2025

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowledgment has been made in the text to all other material used

Your Full Name Here
29 May, 2025

Acknowledgments

I would like to express my deepest gratitude to my thesis supervisor, Dr. Aya Salama, whose invaluable guidance, unwavering support, and continuous encouragement have been instrumental throughout this research. Her profound expertise in Generative AI and insightful direction in navigating its complexities were pivotal in shaping this work. I am especially thankful for her meticulous feedback, consistent availability, and willingness to tackle challenges with innovative thinking. It was truly a privilege to learn from a mentor who combines deep knowledge with approachability.

I also extend my sincere appreciation to my co-supervisor, Dr. Shereen Afifi, for her constructive critiques, insightful comments, and valuable perspectives that enriched this research. Her support and scholarly input added an essential layer of rigor to the project.

My gratitude further extends to the Enosh Science Center for their generous support and collaboration. I am especially thankful to Professor Dr. Ibrahim Elnoshokaty for his continuous encouragement, thoughtful contributions, and for fostering a research-conducive environment. I am equally grateful to Engineer Radwa Taha for her practical support, consistent engagement, and valuable perspectives throughout the various stages of this work.

Finally, I would like to thank the Faculty of Media Engineering and Technology at the German University in Cairo for providing a stimulating academic environment that encourages research, creativity, and innovation.

Abstract

Early brain tumor diagnosis is essential for effective treatment but remains challenging due to inter-observer variability and diagnostic delays in MRI interpretation. This thesis presents a novel pipeline integrating advanced Generative AI models — specifically Large Language Models (LLMs) and Vision-Language Models (VLMs) — with a hierarchical CNN-based feature engineering approach for brain tumor classification.

MRI scans are preprocessed and passed through three diverse CNNs (EfficientNetB0, InceptionV3, Xception), producing robust 1024-dimensional feature vectors. To adapt LLMs for visual input, we introduce a quantization technique that discretizes these continuous embeddings into token-like sequences, enabling frozen LLMs (Gemma 2B and RoBERTa Base) to classify tumor types with up to 94.30% accuracy. In parallel, we explore VLM-based classification using CLIP. A direct zero-shot approach fails (13.66% accuracy), but we propose a learned projection that maps CNN features into CLIP's image-text embedding space. Fine-tuning only the CLIP text encoder achieves 96.83% accuracy — outperforming direct CLIP fine-tuning on raw images and rivaling ensemble state-of-the-art models.

Our results demonstrate that when coupled with engineered domain-specific features and minimal adaptation, powerful Gen AI models can significantly improve medical image classification performance. The framework is efficient, interpretable via cosine similarity probing, and opens new paths for explainable and scalable medical AI systems.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
Gen AI	Generative Artificial Intelligence
LLM(s)	Large Language Model(s)
VLM(s)	Vision-Language Model(s)
CNN(s)	Convolutional Neural Network(s)
CLIP	Contrastive Language-Image Pre-training
ViT	Vision Transformer
MLP	Multi-Layer Perceptron
GAN(s)	Generative Adversarial Network(s)
cGAN(s)	Conditional Generative Adversarial Network(s)
ReLU	Rectified Linear Unit
MLM	Masked Language Model
NSP	Next Sentence Prediction
BPE	Byte-Pair Encoding
SE	Squeeze-and-Excitation
MBConv	Mobile Inverted Bottleneck Convolution
PCA	Principal Component Analysis
Grad-CAM	Gradient-weighted Class Activation Mapping
RF	Random Forest
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
MRI	Magnetic Resonance Imaging
ADF	Anisotropic Diffusion Filtering
CSF	Cerebrospinal Fluid
T1w	T1-weighted (images)
T2w	T2-weighted (images)
FLAIR	Fluid-Attenuated Inversion Recovery
DWI	Diffusion-Weighted Imaging

Continued on next page

Abbreviation	Full Form
PWI	Perfusion-Weighted Imaging
MRS	Magnetic Resonance Spectroscopy
fMRI	functional Magnetic Resonance Imaging
WHO	World Health Organization
DCTN	Dual Convolution Tumor Network
DSC	Dice Similarity Coefficient
M-CNN	Multi-Path CNN
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
GPU(s)	Graphics Processing Unit(s)
L2	L2 Regularization

Chapter 1

Introduction

1.0.1 Motivation

Brain tumors pose a significant challenge to global health, characterized by their complex and catastrophic effects on patients and healthcare systems. In 2022 alone, an estimated 321,731 new brain cancer cases were diagnosed, leading to 248,500 deaths globally, highlighting the pressing need for better treatment and diagnostic approaches [1]. Early and accurate diagnosis is paramount, dramatically improving treatment efficacy and patient survival rates [2]. However, conventional diagnostic workflows, which predominantly rely on radiologists' manual analysis of Magnetic Resonance Imaging (MRI) scans, are susceptible to notable inter-observer variability. This variability can lead to discrepancies that may affect overall consistency in the results. [3]. These limitations highlight a compelling need for innovative solutions, and Artificial Intelligence (AI) has emerged as a promising avenue to revolutionize brain tumor detection and classification, offering the potential for faster, more reliable, and objective diagnoses [4].

While Generative AI (Gen AI) encompasses a diverse array of techniques, this thesis specifically focuses on harnessing the advanced representational power of Large Language Models (LLMs) and Vision-Language Models (VLMs). A central premise guiding this research is that the full potential of these sophisticated AI architectures within specialized domains, such as medical imaging, is optimally realized when they operate on highly informative, domain-specific features. Standard Gen AI models, while powerful, often require such tailored feature conditioning to effectively navigate the complexities and nuanced data inherent in medical diagnostics. Therefore, this work proposes and rigorously evaluates a robust hierarchical feature engineering pipeline as a critical foundational step. This pipeline is designed to distill complex visual information from MRI scans into a rich, structured representation suitable for these advanced AI architectures, thereby paving the way for more efficient and effective brain cancer care.

1.0.2 Problem Statement

Brain tumors persist as a significant public health concern, with their increasing incidence rates imposing a substantial burden on healthcare systems worldwide, evidenced by over 300,000

reported cases annually [1,5]. As established, early and precise diagnosis is critical for effective therapeutic intervention and improved patient outcomes [2]. However, current diagnostic methods rely heavily on subjective visual assessment of MRI scans, a practice prone to human error, time inefficiencies, and diagnostic inconsistencies [3].

Although deep learning-based methods have demonstrated considerable promise in automating and enhancing brain tumor analysis, existing approaches frequently encounter significant limitations. These include challenges in achieving robust explainability and difficulties in handling the inherent variations in tumor location, shape, and size, which can impede accurate segmentation and classification, particularly when generalizing across diverse datasets or when data is scarce [6]. The unique capabilities of Gen AI, especially its potential for nuanced understanding and generation, offer a new paradigm to address these persistent issues.

Therefore, the central problem this thesis aims to address is the need for a more effective, efficient, and clinically relevant AI system for brain tumor diagnosis. It specifically explores how Generative AI, when thoughtfully integrated with domain-specific visual information, can surpass current diagnostic paradigms and existing AI limitations. By utilizing advanced generative AI techniques, this research aims to enhance the accuracy, efficiency, and potentially the interpretability of brain tumor diagnosis, ultimately contributing to improved patient outcomes and reducing delayed or incorrect diagnoses.

1.0.3 Objectives

The primary objectives of this thesis are:

- To investigate the effectiveness of novel Generative AI methods, particularly those involving LLMs and VLMs, for the detection and classification of brain cancer from MRI data.
- To develop and implement an integrated model that leverages the strengths of Generative AI for brain tumor diagnosis, operating on features derived from single MRI scans.
- To comprehensively train and evaluate the performance of the proposed model using relevant brain tumor MRI datasets, assessing its accuracy, efficiency, and potential for clinical application against established baselines.
- To explore and analyze the decision-making process or inherent interpretability of the integrated Gen AI diagnostic system, aiming to enhance trustworthiness.
- To critically analyze the results, discuss the limitations and potential benefits of the proposed architecture, and delineate clear directions for future research in the field of Generative AI for brain tumor diagnosis.

1.0.4 Thesis Outline

This thesis explores the application of Generative AI methods for the detection and classification of brain cancer using MRI images. The structure is as follows:

- **Chapter 1 (Introduction):** Discusses the motivation behind the project, articulates the problem statement, outlines the research objectives, and provides an overview of the thesis structure.
- **Chapter 2 (Background):** Provides an overview of fundamental concepts related to brain tumors, MRI, and the AI technologies employed (CNNs, LLMs, VLMs), followed by a comprehensive literature review of previous work in AI-driven brain tumor detection and relevant Gen AI applications.
- **Chapter 3 (Methodology):** Details the systematic workflow for the project, including data preprocessing, the hierarchical feature engineering pipeline, the specific Gen AI model integration strategies, and the experimental setup for replication.
- **Chapter 4 (Results):** Presents the empirical findings, including performance evaluation of the feature engineering pipeline, comparative analysis of the different Generative AI classification techniques, and detailed discussion of the outcomes, including comparisons with baseline methods and existing state-of-the-art.
- **Chapter 5 (Conclusion):** Summarizes the key achievements of this research, discusses the implications of the findings, and outlines promising avenues for future work to further advance the field and refine the proposed methods.

Chapter 2

Background

2.1 Concept Overview

2.1.1 Brain Tumors

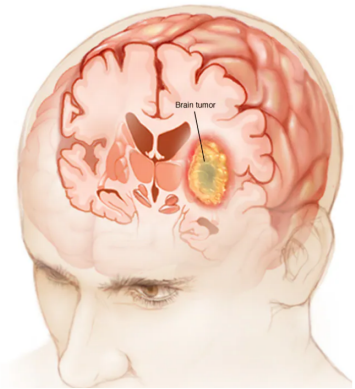


Figure 2.1: Brain tumor [7]

Brain tumors are abnormal growths of cells originating within the brain or its surrounding structures, including the cranial nerves, meninges (protective membranes), pituitary gland, or pineal gland. They represent a diverse group of neoplasms with varying degrees of malignancy and clinical behavior [8]. Brain tumors can be broadly categorized into two main types based on their origin:

1. **Primary Brain Tumors**: These tumors originate directly from cells within the brain or its immediate vicinity. They can be benign (non-cancerous) or malignant (cancerous).

    - **Benign tumors** are typically slow-growing, have relatively distinct borders, and rarely spread to other parts of the body. However, even benign tumors can be life-

threatening if they grow in critical areas and exert pressure on vital brain structures, disrupting normal function. Examples include many meningiomas and pituitary adenomas.

- **Malignant tumors** (brain cancer) are characterized by rapid, infiltrative growth, often invading surrounding brain tissue. They can be highly aggressive and difficult to treat. Gliomas, which arise from glial cells (support cells of the brain), are the most common type of malignant primary brain tumor and include subtypes like astrocytomas (e.g., glioblastoma), oligodendrogliomas, and ependymomas.

2. **Secondary (Metastatic) Brain Tumors**: These tumors originate from cancer cells that have spread (metastasized) to the brain from a primary cancer site elsewhere in the body, such as the lung, breast, colon, kidney, or skin (melanoma). Metastatic brain tumors are more common than primary brain tumors overall.

The symptoms of a brain tumor can vary widely depending on its type, size, location, and rate of growth. Common symptoms may include headaches (often worsening over time), seizures, nausea, vomiting, changes in vision or hearing, weakness or numbness in parts of the body, difficulty with balance or coordination, personality or behavioral changes, and cognitive decline. Early and accurate diagnosis, typically involving neurological examination and neuroimaging techniques like MRI, is crucial for determining the optimal treatment strategy, which may include surgery, radiation therapy, chemotherapy, targeted therapy, or a combination thereof [9]. The World Health Organization (WHO) classification system for tumors of the central nervous system is the standard for diagnosing and grading brain tumors, incorporating histological features and, increasingly, molecular genetic markers to guide prognosis and treatment decisions [8].

2.1.2 Magnetic Resonance Imaging (MRI)

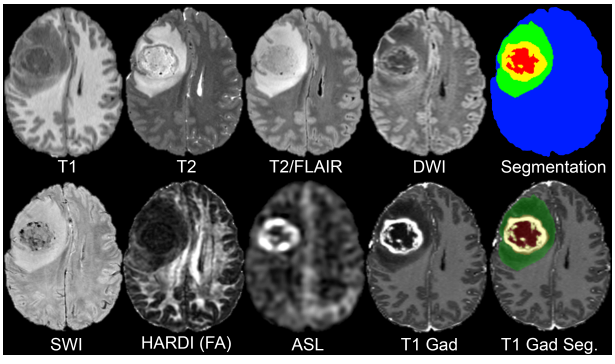


Figure 2.2: MRI [10]

Magnetic Resonance Imaging (MRI) is a sophisticated and versatile non-invasive medical imaging technique that has become indispensable in modern diagnostics, particularly in neu-

roradiology for the evaluation of brain pathologies, including tumors [11]. Unlike imaging modalities that use ionizing radiation (like X-rays or CT scans), MRI utilizes strong magnetic fields, radiofrequency (RF) pulses, and the principles of nuclear magnetic resonance to generate detailed cross-sectional images of the body.

The fundamental principle of MRI involves placing the patient within a powerful magnet, which aligns the protons (primarily in water molecules) within the body's tissues. A subsequent RF pulse temporarily perturbs this alignment. As the protons relax back to their equilibrium state, they emit RF signals that are detected by receiver coils. Sophisticated computer algorithms then process these signals to reconstruct high-resolution images. Different tissues have varying proton densities and relaxation properties (T1 and T2 relaxation times), which result in different signal intensities and thus contrast in the MR images. This allows for excellent soft tissue differentiation, making MRI particularly well-suited for visualizing brain anatomy and pathology.

Various MRI sequences can be employed to highlight different tissue characteristics:

- **T1-weighted (T1w) images**: Provide good anatomical detail, where fat appears bright and water (like CSF) appears dark. They are often used with contrast agents.
- **T2-weighted (T2w) images**: Are sensitive to water content, where fluid-filled structures and pathology (like edema or tumors) often appear bright.
- **FLAIR (Fluid-Attenuated Inversion Recovery)**: A T2-based sequence that suppresses the signal from cerebrospinal fluid (CSF), making abnormalities near CSF spaces (like periventricular lesions) more conspicuous.
- **Contrast-Enhanced MRI**: Involves the intravenous administration of a contrast agent (typically gadolinium-based). These agents shorten the T1 relaxation time of tissues they accumulate in, leading to signal enhancement. In brain tumors, contrast enhancement often indicates a breakdown of the blood-brain barrier or increased vascularity, helping to delineate tumor margins and assess activity.

Advanced MRI techniques, such as Diffusion-Weighted Imaging (DWI) to assess cellularity, Perfusion-Weighted Imaging (PWI) to evaluate tumor vascularity and blood flow, Magnetic Resonance Spectroscopy (MRS) to analyze tumor metabolism, and functional MRI (fMRI) to map brain activity, further extend the diagnostic capabilities of MRI. These techniques provide crucial information for tumor detection, characterization, grading, treatment planning, and monitoring response to therapy [12].

2.1.3 Convolutional Neural Networks (CNNs)

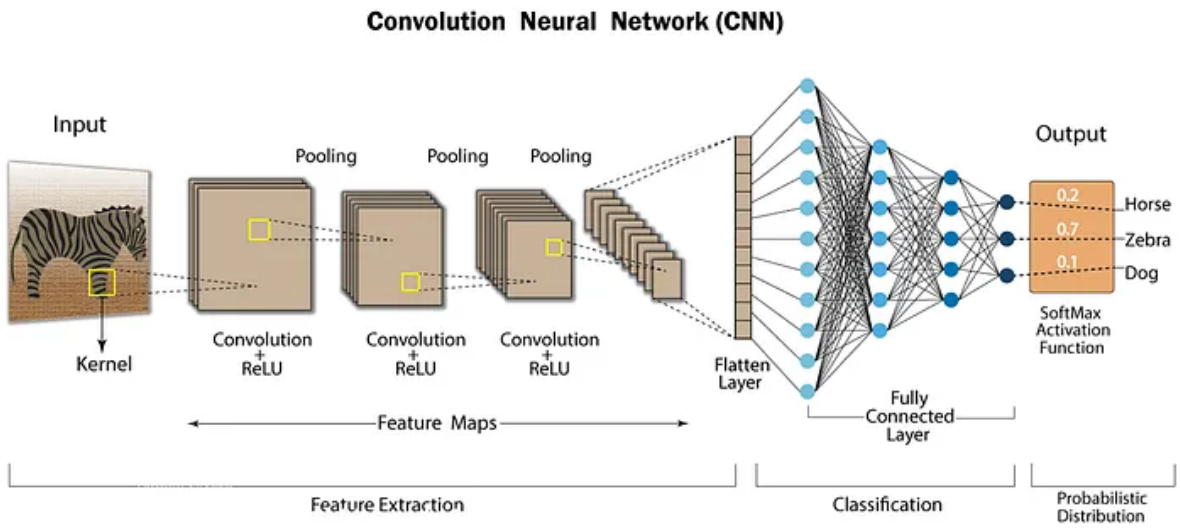


Figure 2.3: Convolutional Neural Networks (CNNs) [13]

Convolutional Neural Networks (CNNs or ConvNets) are a class of deep neural networks that have become the dominant approach for a wide array of visual recognition tasks, including image classification, object detection, and image segmentation [14]. Inspired by the organization of the animal visual cortex, CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input data, such as images.

The architecture of a typical CNN is characterized by several distinct types of layers:

1. **Convolutional Layer**: This is the core building block of a CNN. It employs a set of learnable filters (or kernels) that are convolved across the input volume (e.g., an image or the output of a previous layer). Each filter is small spatially (e.g., 3x3 or 5x5 pixels) but extends through the full depth of the input volume. During the convolution, the filter slides over the input, computing dot products between the filter entries and the input at any position. This process produces a 2D activation map (or feature map) of that filter, indicating the locations where the specific feature detected by the filter is present. The network learns multiple filters in each convolutional layer, each detecting different features (e.g., edges, textures, or more complex patterns in deeper layers).

2. **Activation Layer (Non-linearity)**: After each convolutional operation, an activation function is typically applied element-wise to introduce non-linearity into the model. The Rectified Linear Unit (ReLU), $f(x) = \max(0, x)$, is a commonly used activation function due to its simplicity and effectiveness in mitigating the vanishing gradient problem. Other activations like Sigmoid or Tanh were used in earlier networks.

3. **Pooling Layer (Subsampling)**: Pooling layers are commonly inserted between successive convolutional layers to progressively reduce the spatial dimensionality (width and height) of the representation. This helps to decrease the number of parameters and computation in the network, and also provides a form of translation invariance. Common

pooling operations include Max Pooling (taking the maximum value in a local region) and Average Pooling.

4. **Fully Connected Layer**: After several convolutional and pooling layers, the high-level reasoning in the neural network is typically done via fully connected layers. Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular Multi-Layer Perceptrons (MLPs). The output of the final fully connected layer is often fed to a softmax function for classification tasks to produce probability distributions over the classes.

Through the hierarchical stacking of these layers, CNNs can learn low-level features (like edges and corners) in the initial layers, which are then combined to form mid-level features (like simple shapes or parts of objects) in intermediate layers, and finally, high-level, more abstract representations (like entire objects) in deeper layers. This ability to learn feature hierarchies directly from data, without the need for manual feature engineering, is a key reason for their success.

Inception V3

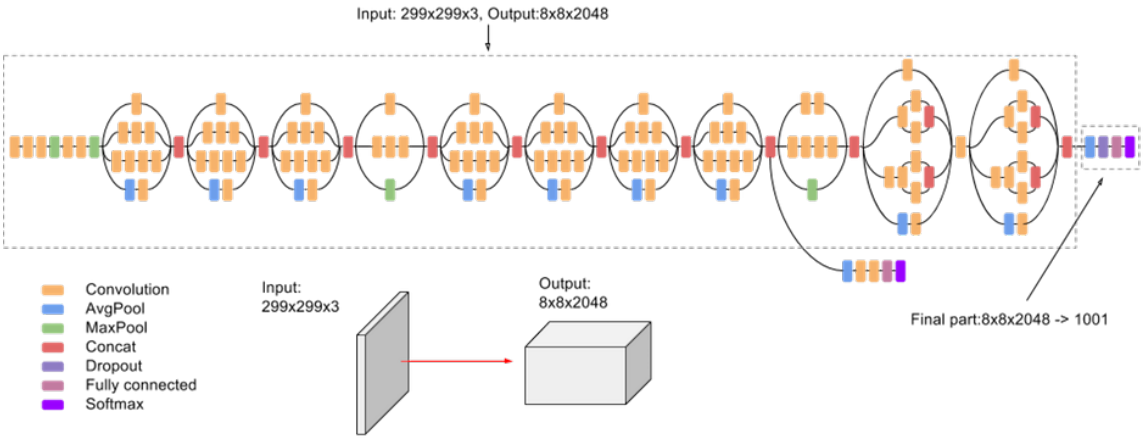


Figure 2.4: InceptionV3 [15]

InceptionV3, developed by researchers at Google, is a significant iteration in the Inception family of convolutional neural network architectures [16]. It builds upon the principles of its predecessors, primarily focusing on improving computational efficiency and accuracy while reducing the parameter count. Key design principles and improvements in InceptionV3 include:

- **Factorizing Convolutions**: Larger convolutions are factorized into smaller ones to reduce computational cost. For instance, a 5x5 convolution is replaced by two stacked 3x3 convolutions.

- **Asymmetric Convolutions**: Further factorization is achieved by replacing NxN convolutions with a 1xN followed by an Nx1 convolution (e.g., a 3x3 can be replaced by a 1x3 then a 3x1). This is particularly effective for medium grid sizes.
- **Auxiliary Classifiers**: Used during training as regularizers, particularly for deeper networks, by adding classifiers to intermediate layers to combat vanishing gradients. In InceptionV3, their role as regularizers is emphasized.
- **Efficient Grid Size Reduction**: Careful design of how feature map grid sizes are reduced to avoid representational bottlenecks.

These modifications allow InceptionV3 to achieve high accuracy on tasks like ImageNet classification with greater computational efficiency compared to earlier Inception versions.

Xception

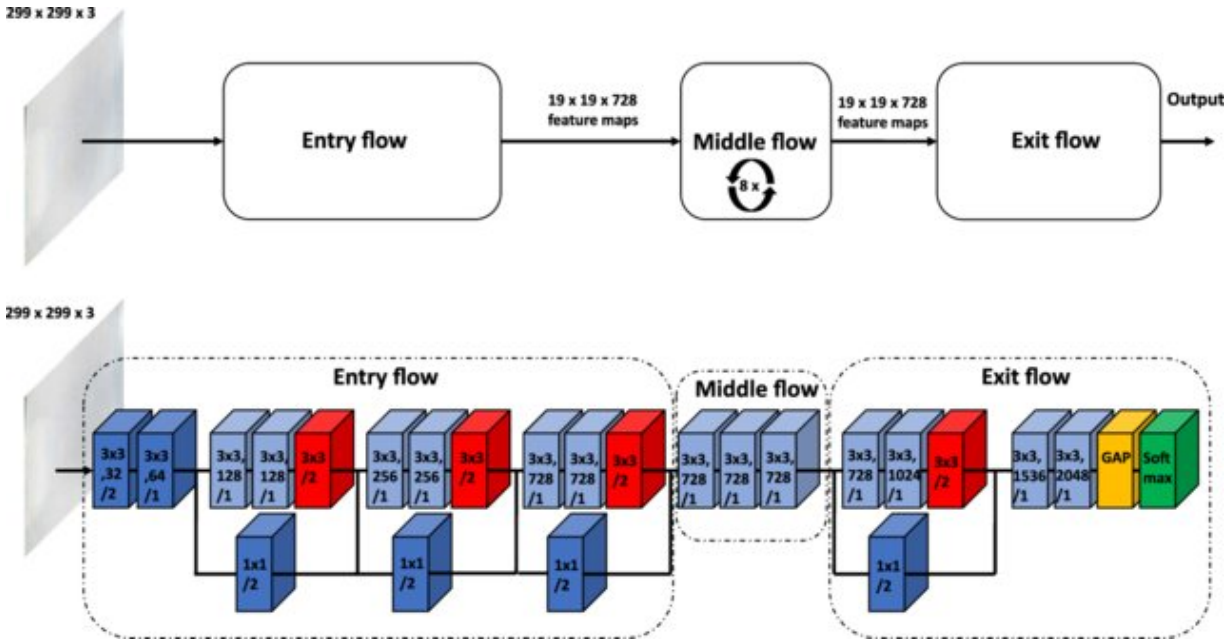


Figure 2.5: Xception [17]

Xception, which stands for "Extreme Inception," is a deep convolutional neural network architecture developed by François Chollet at Google. It builds upon the Inception hypothesis that cross-channel correlations and spatial correlations can be decoupled and mapped separately. Xception takes this idea to an extreme by proposing that these two types of correlations can be *entirely* disentangled.

The core of the Xception architecture lies in its extensive use of **depthwise separable convolution layers**, which replace the standard Inception modules. A depthwise separable convolution consists of two steps:

1. A **depthwise convolution**: This performs spatial convolution independently for each input channel (e.g., a 3x3 filter is applied to each channel separately).

2. A **pointwise convolution**: This is a 1x1 convolution that projects the channels output by the depthwise convolution onto a new channel space, effectively combining the information across channels.

This approach is significantly more parameter-efficient and computationally less expensive than standard convolutions. The Xception architecture is essentially a linear stack of depthwise separable convolution layers with residual connections (similar to ResNet), allowing for the creation of very deep and powerful models. Xception demonstrated strong performance on image classification tasks like ImageNet, often outperforming Inception V3 with a similar or even smaller number of parameters, underscoring the effectiveness and efficiency of depthwise separable convolutions [18].

EfficientNetB0

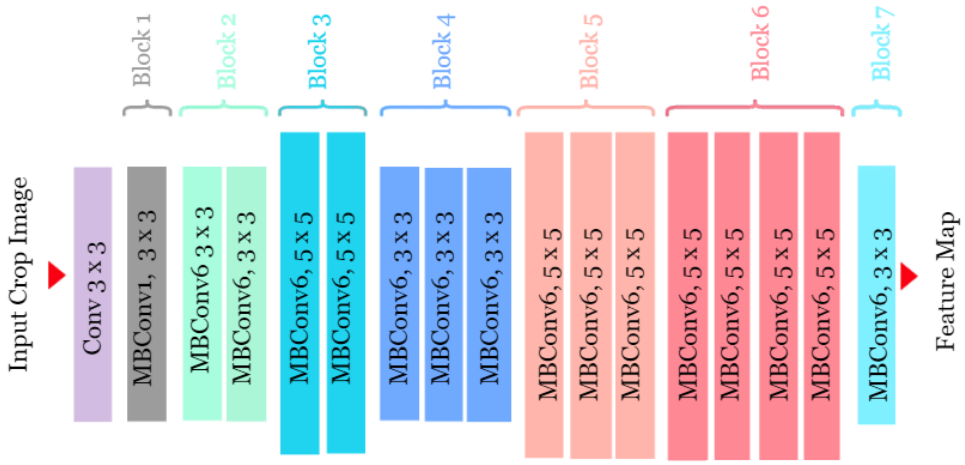


Figure 2.6: EfficientNetB0 [19]

EfficientNetB0 is a highly influential convolutional neural network architecture developed by Mingxing Tan and Quoc V. Le at Google, known for its systematic approach to model scaling . Addressing the limitations of traditional depth, width, or resolution scaling in isolation, EfficientNet introduces **compound scaling**. This method uniformly scales all three dimensions (depth, width, and resolution) using a fixed set of scaling coefficients, optimized through neural architecture search.

The architecture itself utilizes mobile inverted bottleneck convolution (MBConv) blocks, similar to those found in MobileNetV2, enhanced with **squeeze-and-excitation (SE) optimization** layers to improve feature representation by adaptively recalibrating channel-wise feature responses. EfficientNetB0 serves as the baseline model in the EfficientNet family. While larger

variants (like EfficientNet-B7) achieve state-of-the-art accuracy on ImageNet with significantly fewer parameters and FLOPs than previous models, EfficientNetB0 itself provides a strong and efficient foundation. The entire EfficientNet family demonstrates excellent transfer learning capabilities to various other datasets [20].

2.1.4 Large Language Models (LLMs)

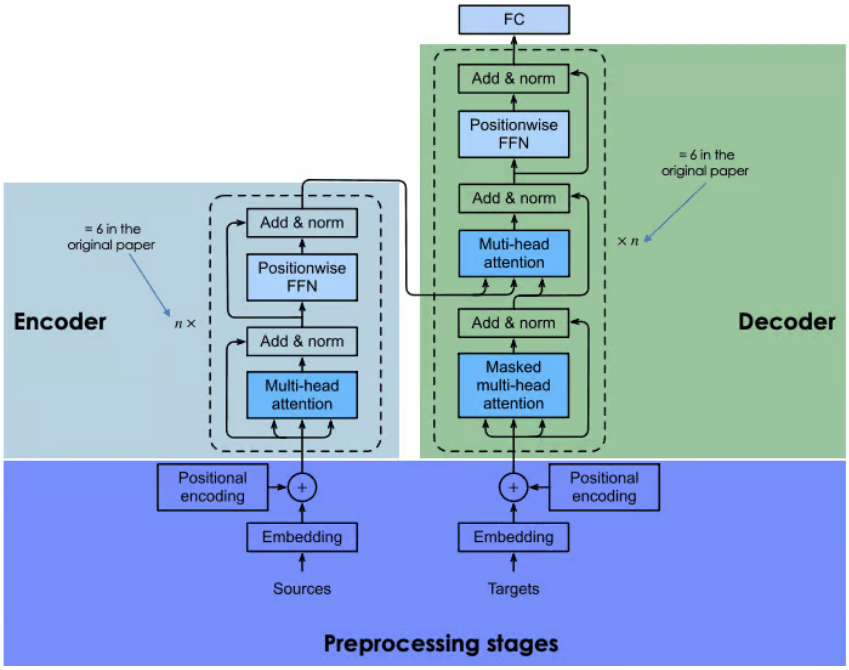


Figure 2.7: Large Language Models (LLMs) [21]

Large Language Models (LLMs) have rapidly become a cornerstone of the current artificial intelligence revolution, representing a significant leap forward in the ability to process, understand, and generate human language. As sophisticated AI systems, LLMs are characterized by their vast scale, often incorporating billions (or even trillions) of parameters, and are trained on massive datasets of text and code [22].

The predominant architecture underpinning most modern LLMs is the **Transformer**, which introduced the concept of the **self-attention mechanism**. This mechanism allows the model to weigh the importance of different words in an input sequence when processing any given word, enabling a rich contextual understanding of language. LLMs typically undergo a two-stage training process:

1. **Pre-training**: The model is trained on a very large, general-purpose corpus of text using self-supervised learning objectives, such as predicting masked words (as in BERT-like

models) or predicting the next word in a sequence (as in GPT-like models). During this phase, the model learns statistical relationships, grammar, common sense knowledge, and various linguistic patterns.

2. **Fine-tuning**: After pre-training, the model can be adapted to specific downstream tasks (e.g., sentiment analysis, question answering, summarization, translation) by training it further on smaller, task-specific datasets. This often involves techniques like instruction tuning or reinforcement learning from human feedback (RLHF) to better align the model's outputs with desired behaviors.

The impact of LLMs is already being felt across numerous domains, from powering advanced chatbots and virtual assistants to enabling high-quality machine translation, content creation, and complex reasoning. While their capabilities are immense, ongoing research addresses challenges related to bias, transparency, factual accuracy (hallucinations), and the ethical implications of their widespread deployment.

Gemma

Gemma is a family of lightweight, state-of-the-art open-weight language models developed by Google, built from the same research and technology used to create the Gemini models [23]. Released in early 2024, Gemma models are designed to be accessible and to promote responsible AI development. They are available in various sizes, such as 2 billion (2B) and 7 billion (7B) parameters, to suit different computational resources and application needs.

Key characteristics of Gemma include:

- **Open Weights**: Unlike many large proprietary models, Google has released the model weights for Gemma, allowing researchers and developers to use, modify, and build upon them more freely.
- **Pre-trained and Instruction-Tuned Variants**: Gemma models are offered in both pre-trained versions, which have learned general language understanding, and instruction-tuned versions, which are fine-tuned to follow user prompts and instructions more effectively.
- **Performance**: Despite their relatively smaller size compared to some larger models, Gemma models demonstrate strong performance across a range of text generation and understanding benchmarks.
- **Responsible AI Toolkit**: The release is accompanied by tools and guidance to encourage responsible use and mitigate potential harms, reflecting a focus on safety and ethics in AI development.

The availability of Gemma models provides a valuable resource for the research community and developers looking to leverage capable LLMs without the need for extensive computational resources typically required by much larger models.

RoBERTa

RoBERTa, which stands for "Robustly Optimized BERT Pretraining Approach," is a language model developed by researchers at Facebook AI (now Meta AI) that builds upon and significantly improves the original BERT (Bidirectional Encoder Representations from Transformers) architecture and pre-training strategy [24]. While BERT was a landmark model, RoBERTa demonstrated that careful modifications to the pre-training process could lead to substantially better performance on downstream natural language understanding tasks.

The key optimizations introduced in RoBERTa include:

- **More Extensive Training Data and Longer Training**: RoBERTa was trained on a significantly larger dataset (including CC-News, a new dataset collected for this work) for a longer duration.
- **Dynamic Masking**: In BERT, the masking pattern for the Masked Language Model (MLM) objective was fixed during data preprocessing. RoBERTa implements dynamic masking, where the masking pattern is generated every time a sequence is fed to the model, increasing the diversity of the training examples.
- **Removal of Next Sentence Prediction (NSP) Objective**: The authors found that removing the NSP loss from BERT's pre-training improved performance on several downstream tasks, suggesting it might have been detrimental or not as beneficial as initially thought.
- **Larger Batch Sizes**: Training with larger mini-batches was found to be beneficial for performance.
- **Byte-Level BPE**: Instead of character-level BPE (Byte-Pair Encoding) used in BERT, RoBERTa uses byte-level BPE, which handles unseen Unicode characters more gracefully and allows for a moderately sized vocabulary.

These modifications collectively allowed RoBERTa to achieve state-of-the-art results on various NLP benchmarks like GLUE, SQuAD, and RACE, outperforming BERT and other contemporary models at the time of its release.

2.1.5 Vision-Language Models (VLMs)

Vision-Language Models (VLMs) represent a significant advancement in artificial intelligence, aiming to bridge the gap between visual perception and natural language understanding. These models are designed to learn joint representations that connect information from both visual modalities (e.g., images, videos) and textual modalities. Unlike traditional computer vision models that operate solely on pixel data or language models that process only text, VLMs strive to understand the semantic relationship between visual content and its corresponding linguistic descriptions. This capability enables a wide range of cross-modal tasks, such as:

- **Image Captioning**: Generating textual descriptions for a given image.
- **Visual Question Answering (VQA)**: Answering natural language questions based on the content of an image.
- **Text-based Image Retrieval**: Finding images that match a given textual query.
- **Zero-Shot Image Classification**: Classifying images into categories defined by textual descriptions, even if the model has not been explicitly trained on those specific image-category pairs.

Architecturally, VLMs often employ separate encoders for visual and textual inputs. For instance, a Convolutional Neural Network (CNN) or a Vision Transformer (ViT) might process the image, while a Transformer-based text encoder processes the language. The core challenge lies in aligning these representations. A prominent approach, exemplified by models like CLIP (Contrastive Language-Image Pre-training) [25], involves training these encoders jointly using a **contrastive learning** objective. This objective aims to maximize the similarity (e.g., cosine similarity) between the embeddings of corresponding image-text pairs while minimizing the similarity for non-matching pairs from a large dataset of (image, text) examples, often scraped from the web. This pre-training on vast, noisy datasets allows VLMs to learn robust, generalizable visual concepts directly from natural language supervision, often exhibiting impressive zero-shot transfer capabilities to various downstream tasks.

CLIP (Contrastive Language-Image Pre-training)

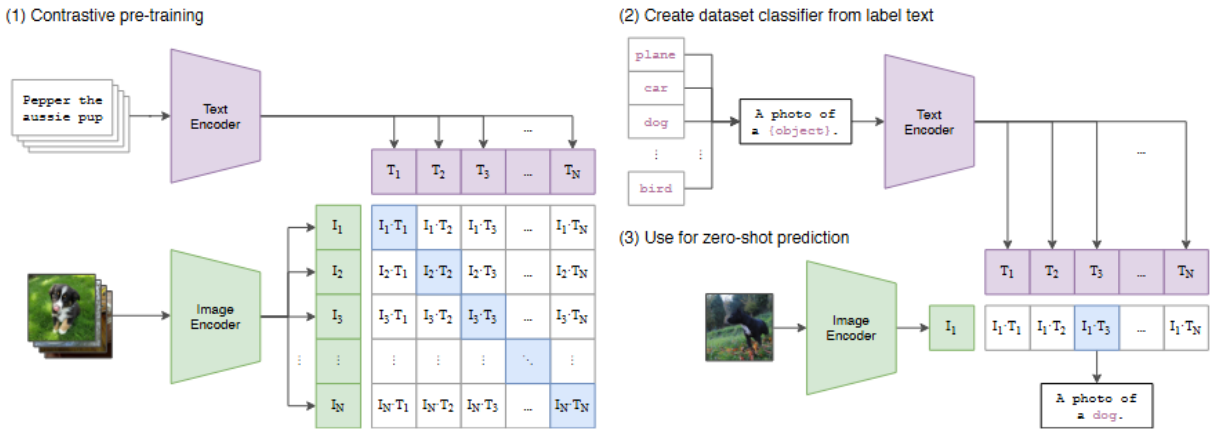


Figure 2.8: CLIP (Contrastive Language-Image Pre-training) [25]

CLIP, which stands for Contrastive Language-Image Pre-training, is a highly influential Vision-Language Model developed by OpenAI [25]. It is designed to learn visual concepts directly from natural language supervision by pre-training on a massive dataset of (image, text

caption) pairs collected from the internet. The core innovation of CLIP lies in its training methodology and its remarkable zero-shot transfer capabilities.

The CLIP architecture consists of two main components:

1. **Image Encoder**: This component processes images and can be implemented using various architectures, such as a ResNet or, more commonly in later versions, a Vision Transformer (ViT). It outputs a feature vector (embedding) representing the image.

2. **Text Encoder**: This component processes text and is typically a Transformer-based model. It takes a textual description (e.g., a class name like "a photo of a dog" or a more complex caption) and outputs a feature vector representing the text.

During pre-training, these two encoders are trained jointly using a **contrastive loss** function. For a batch of N (image, text) pairs, the model aims to predict which of the N×N possible (image, text) pairings in the batch are the actual correct pairs. This is achieved by learning to embed corresponding image-text pairs close together in a shared multimodal embedding space, while pushing non-corresponding pairs further apart. Specifically, the cosine similarity between the image embedding and the text embedding is maximized for correct pairs and minimized for incorrect pairs.

A key strength of CLIP is its **zero-shot classification** ability. Once trained, CLIP can classify an image into a new set of categories without requiring any fine-tuning on images from those specific categories. This is done by:

1. Computing the image embedding for the input image using the image encoder.

2. Creating text prompts for each target category (e.g., "a photo of a [category name]").

3. Computing text embeddings for these prompts using the text encoder.

4. Comparing the image embedding with all text embeddings using cosine similarity. The category whose text prompt yields the highest similarity is predicted as the class for the image.

This flexibility and strong generalization make CLIP a powerful tool for a wide array of vision and vision-language tasks, moving beyond traditional supervised learning paradigms that require extensive labeled data for each new task .

2.2 Literature Review

Brain Tumor Detection using CNNs

This section explores articles and discusses how CNNs are effective in processing MRI scans for brain tumor detection.

Al-Zoghby et al. [26], proposed the Dual Convolution Tumor Network (DCTN) for detecting brain tumor types. Their model, a dual-branch CNN combining a pre-trained VGG-16 and a custom CNN, achieved a 99% testing accuracy on a dataset that consisted of the following tumor types meningioma, glioma, and pituitary using T1-weighted contrast-enhanced MRI. This study shows the high accuracies that can be achieved with CNNs and how effective combining pre-trained and custom architectures for feature extraction from MRI images.

Huda and Ku-Mahamud [27] provided a review on CNN image segmentation approaches for brain tumor classification. Their work shows how effective various CNN architectures are especially U-Net and its variations for brain tumor segmentation by achieving high Dice Similarity Coefficient (DSC) scores. This review shows the strength of CNNs for medical image analysis.

Batool and Byun [28] introduced a lightweight Multi-Path CNN (M-CNN) architecture "A lightweight multi-path convolutional neural network architecture using optimal features selection for multiclass classification of brain tumor using Mri scans." Their M-CNN, designed for efficient multi-class brain tumor classification, utilizes parallel convolutional paths with different kernel sizes for multi-scale feature extraction. that achieved a 96.03% accuracy.

Mugdha and Uddin [29] compared a custom CNN, NeuroSight, against pre-trained models. While their custom CNN achieved a 92.59% test accuracy, pre-trained models like VGG-16 outperformed it, reaching 95.52%. This article supports the effectiveness of transfer learning with CNNs for brain tumor detection.

Rastogi et al. [30] compared fine-tuned InceptionResNetV2, VGG19, Xception, and MobileNetV2 models for binary brain tumor detection. Their findings indicated that fine-tuned Xception had the highest accuracy, with 96.11%, showing that Xception and VGG-16 have better architectures for this task.

GANs and cGANs for Medical Image Generation

This section explores studies that discuss Generative Adversarial Networks (GANs) and conditional GANs (cGANs) and their unique capabilities in medical image generation.

Ahmad et al. [31] developed a GAN-based architecture for medical image super-resolution. Their Multi-Path Progressive Upscaling GAN outperformed other GANS like SRGAN and bicubic interpolation, which shows the GANs' potential to enhance medical image quality. This work shows the generative power of GANs in medical imaging.

Mukherkjee et al. [32] introduced AGGrGAN for generating synthetic brain tumor MRI images. Their approach used multiple GAN models and incorporated style transfer to improve realism and increase classification performance when used for data augmentation. This study directly shows the ability of GANs to generate realistic brain tumor images.

Park et al. [33] showed the ability of GANs to generate synthetic post-contrast MRI images for glioblastoma assessment. The research they did demonstrated that GAN-generated synthetic images could accurately predict tumor progression, suggesting that GANs can produce realistic synthetic medical images.

Xia et al.'s [34] provided an overview of GAN-based anomaly detection methods. While focused on anomaly detection, this review highlights GANs' ability to learn features of normal data and identify deviations.

Hybrid Architectures and Ensemble Learning for Brain Tumor Diagnosis

This section explores studies that discuss hybrid and ensemble methods in deep learning for medical image analysis.

Shaikh et al. [35] proposed SEL-DenseNet201, a stacking ensemble learning approach using DenseNets. Their ensemble achieved a high accuracy of (99.65%) and a Dice coefficient of (97.43%) in brain tumor detection, demonstrating the performance benefits of combining CNN architectures. This work shows the potential of ensemble methods to improve accuracy.

Hosny et al. [36] developed an explainable ensemble model combining DenseNet121 and InceptionV3 with Grad-CAM for brain tumor diagnosis. Their ensemble achieved excellent accuracy of (99.02%) and enhanced explainability through Grad-CAM visualizations. This study not only reinforces the performance benefits of ensemble learning but also shows the importance of explainability in medical AI.

Noreen et al. [37] also explored ensemble techniques, demonstrating the effectiveness of combining features from fine-tuned CNN models (specifically Inception-v3 and Xception) with traditional machine learning classifiers (SVM, RF, KNN). Their ensemble method achieved a notable accuracy of 94.34% for a three-class brain tumor classification task, highlighting the synergy between deep features and established classification algorithms.

Aurna et al. [38] presented a sophisticated "two-stage feature level ensemble" of deep CNN models. This approach was applied to a four-class brain tumor problem, similar to the scope of the current thesis, including a 'no tumor' class. Their methodology involved an initial selection of optimal CNNs, followed by the creation of first-stage ensembles through feature concatenation. A final two-stage ensemble was then formed, with Principal Component Analysis (PCA) used for dimensionality reduction before classification with a Softmax layer. This intricate process yielded a remarkable average accuracy of 98.96% on a comprehensive merged dataset, providing a significant benchmark in the field. These studies by Noreen et al. and Aurna et al. collectively underscore the considerable potential and varied strategies for employing ensemble learning in enhancing brain tumor diagnosis.

Tehsin et al. [39] introduced GATransformer, a hybrid model combining Graph Attention Networks and Transformers for explainable brain tumor detection. Their model generates explainable attention maps, emphasizing the value of hybrid architectures and explainability in this domain.

Data Preprocessing and Augmentation Techniques in Brain Tumor MRI Analysis

This section explores studies that discuss effective data preprocessing and augmentation that are crucial for training deep learning models.

Abraham et al. [40] detailed preprocessing and augmentation steps, including intensity normalization, uniform input dimensions, denoising using Gaussian filters, and extensive data augmentation using Roboflow. Their preprocessing and augmentation pipeline shows the importance of these steps to achieve high performance.

Mugdha and Uddin [29] utilized Canny edge detection and image cropping as preprocessing steps in NeuroSight, while Rastogi et al. [30] and Hosny et al. [36] employed data augmentation techniques like rotation, flipping, and scaling.

Table 2.1: Summary of Brain Tumor Detection and Analysis Articles

Authors	Year	Method/Model	Task	Key Findings/Accuracy
2.2.1 Brain Tumor Detection using CNNs				
Al-Zoghby et al. [26]	2023	Dual Convolution Tumor Network (DCTN): Dual-branch CNN (VGG-16 + Custom CNN)	Brain Tumor Type Detection	99% testing accuracy (meningioma, glioma, pituitary) using T1-weighted contrast-enhanced MRI. Shows effectiveness of combining pre-trained and custom CNNs.
Huda and Ku-Mahamud [27]	2025	Review of CNN Image Segmentation Approaches (U-Net, variations)	Brain Tumor Classification (Segmentation)	Shows effectiveness of CNN architectures, especially U-Net and variations, for brain tumor segmentation with high Dice Similarity Coefficient (DSC) scores.
Batool and Byun [28]	2025	Multi-Path CNN (M-CNN)	Multi-class Brain Tumor Classification	Lightweight M-CNN using parallel convolutional paths for multi-scale feature extraction achieved 96.03% accuracy.
Mugdha and Uddin [29]	2025	NeuroSight (Custom CNN) vs. Pre-trained Models (VGG-16)	Brain Tumor Detection	Pre-trained VGG-16 (95.52%) outperformed custom CNN NeuroSight (92.59%) in test accuracy, supporting transfer learning.
Rastogi et al. [30]	2025	Fine-tuned Transfer Learning Models (InceptionResNetV2, VGG19, Xception, MobileNetV2)	Binary Brain Tumor Detection	Fine-tuned Xception achieved highest accuracy (96.11%) among compared models, indicating superior architecture for this task.
2.2.2 GANs and cGANs for Medical Image Generation				

Continued on next page

Table 2.1 *continued from previous page*

Authors	**Year**	**Method/Model**	**Task**	**Key Findings/Accuracy**
Ahmad et al. [31]	2022	Multi-Path Progressive Upscaling GAN	Medical Image Super-Resolution	GAN-based architecture outperformed SRGAN and bicubic interpolation, demonstrating GANs' potential to enhance medical image quality.
Mukherkjee et al. [32]	2022	AGGrGAN (Aggregated GAN models with style transfer)	Brain Tumor MRI Image Generation	AGGrGAN generated synthetic brain tumor MRI images with improved realism and classification performance when used for data augmentation.
Park et al. [33]	2022	GANs for Synthetic Post-Contrast MRI Images	Glioblastoma Assessment (Tumor Progression Prediction)	GAN-generated synthetic images accurately predicted tumor progression, showing GANs can produce realistic synthetic medical images for diagnosis.
Xia et al. [34]	2022	Review of GAN-based Anomaly Detection Methods	Anomaly Detection	Review highlights GANs' ability to learn normal data features and identify deviations for anomaly detection.
2.2.3 Hybrid Architectures and Ensemble Learning for Brain Tumor Diagnosis				
Shaikh et al. [35]	2025	SEL-DenseNet201 (Stacking Ensemble Learning with DenseNets)	Brain Tumor Detection and Segmentation	Ensemble approach achieved high accuracy (99.65%) and Dice coefficient (97.43%), demonstrating performance benefits of ensemble methods.

Continued on next page

Table 2.1 *continued from previous page*

Authors	**Year**	**Method/Model**	**Task**	**Key Findings/Accuracy**
Hosny et al. [36]	2025	Explainable Ensemble Model (DenseNet121 + InceptionV3 + Grad-CAM)	Brain Tumor Diagnosis	Explainable ensemble model achieved excellent accuracy (99.02%) and enhanced explainability using Grad-CAM visualizations.
Noreen et al. [37]	2021	Fine-Tuned Models (Inception-v3, Xception) and Ensemble Method (KNN+SVM+RF)	Brain Tumor Classification	Achieved 94.34% accuracy for 3-class tumor classification (glioma, meningioma, pituitary) using ensemble of fine-tuned CNN features.
Aurna et al. [38]	2022	Two-Stage Feature Level Ensemble of Deep CNNs (e.g., EfficientNet-B0, ResNet-50) with PCA	MRI Brain Tumor Classification (4-class)	Reported 98.96% accuracy on a merged dataset using a two-stage feature ensemble, PCA, and Softmax classifier for 4 classes.
Tehsin et al. [39]	2025	GATransformer (Graph Attention Network + Transformer)	Explainable Brain Tumor Detection	Hybrid GATransformer generates explainable attention maps, highlighting value of hybrid architectures and explainability.
2.2.4 Data Preprocessing and Augmentation Techniques in Brain Tumor MRI Analysis				
Abraham et al. [40]	2025	Preprocessing and Augmentation Pipeline (Intensity Normalization, Denoising, Augmentation with Roboflow)	Brain Tumor MRI Detection	Detailed preprocessing and augmentation pipeline (intensity normalization, denoising, Roboflow augmentation) is crucial for high performance.

2.2.1 Discussion of Literature Review Findings

The preceding literature review, summarized in Table 2.1, reveals several key trends, advancements, and remaining opportunities in the application of AI to brain tumor detection and classification from MRI data.

Dominance and Evolution of CNNs: Convolutional Neural Networks (CNNs) have unequivocally established themselves as a cornerstone technology. Early and ongoing research (e.g., [26, 29, 30]) demonstrates their power in feature extraction and direct classification, with various architectures like VGG, Inception, and custom designs achieving high accuracies. The utility of transfer learning from pre-trained models is a recurring theme, often providing a strong performance baseline or outperforming custom models trained from scratch [29]. Furthermore, specific CNN architectures like U-Net and its variants are shown to be highly effective for segmentation tasks [27], which, while not the primary focus of this thesis, underpins many diagnostic pipelines. The consistent success across different CNN approaches [28] highlights their robustness for processing medical image data.

Generative Models for Data Enhancement and Synthesis: Generative Adversarial Networks (GANs) and conditional GANs (cGANs) have emerged as powerful tools, primarily for enhancing the training data and image quality. Studies like [31] show their utility in super-resolution, while [32] and [33] demonstrate their capability in generating synthetic MRI images, which can be invaluable for data augmentation, especially in data-scarce medical domains, or for predicting tumor progression. The ability of GANs to learn normal data features also extends their use to anomaly detection [34].

The Rise of Hybrid and Ensemble Architectures for Peak Performance: A significant trend towards achieving state-of-the-art results involves the use of hybrid architectures and ensemble learning methods [35–38]. These approaches leverage the strengths of multiple models or techniques, such as stacking different CNNs [35], combining pre-trained models with explainability modules like Grad-CAM [36], or more complex multi-stage feature-level ensembles [38]. The work by Aurna et al. [38], in particular, demonstrates that meticulous feature engineering and ensemble strategies can yield exceptionally high accuracies, even on challenging multi-class problems. The integration of traditional machine learning classifiers with deep features extracted by CNNs, as shown by Noreen et al. [37], also remains a viable strategy. As model complexity increases with these approaches, the need for explainability, as explored by [36] and [39] with hybrid Transformer models, becomes more pertinent.

Foundational Importance of Data Preprocessing and Augmentation: Across all methodologies, the critical role of robust data preprocessing and augmentation is consistently highlighted [40]. Techniques such as intensity normalization, denoising, skull stripping, and diverse data augmentation strategies are fundamental to achieving reliable and high-performing models, addressing issues like class imbalance and improving model generalization.

Identified Gaps and Opportunities for Generative AI: While the reviewed literature showcases significant progress, particularly with CNN-based ensembles, several opportunities re-

main, especially concerning the application of the latest generation of large-scale Generative AI models (LLMs and VLMs) beyond data augmentation:

1. **Advanced Feature Representation for Gen AI:** Most existing works either use CNNs as end-to-end classifiers or extract features from a single (or few) CNNs for traditional ensembles. There is an opportunity to explore more sophisticated, *hierarchically engineered feature representations* derived from multiple diverse CNNs, specifically designed as rich input for more advanced AI architectures like LLMs and VLMs.

2. **Novel Integration of LLMs and VLMs for Classification:** The direct application of LLMs and VLMs for classification tasks in specialized medical imaging domains, particularly when operating on pre-extracted, complex visual features, is still an emerging area. Traditional methods often don't fully leverage the contextual understanding or multimodal capabilities of these large models.

3. **Bridging Modality Gaps with Engineered Features:** While VLMs like CLIP are powerful, their direct zero-shot application might be suboptimal for highly specific medical tasks without tailored adaptation. Investigating methods to effectively project domain-specific engineered features into the latent spaces of VLMs, and potentially fine-tuning these VLMs, presents a promising research direction.

This thesis aims to address these gaps by proposing a novel framework that first constructs a robust, multi-perspective feature set from MRI images using a hierarchical CNN pipeline. Subsequently, it systematically investigates and compares different strategies for integrating these engineered features with cutting-edge Generative AI models, including quantized LLMs and adapted VLMs, for the task of brain tumor classification. The goal is to explore whether these newer Gen AI paradigms, when provided with high-quality, specialized visual features, can offer new levels of performance or complementary insights compared to existing state-of-the-art ensemble methods.

Chapter 3

Methodology

This chapter details the systematic methodology employed to develop and evaluate advanced artificial intelligence systems for brain tumor classification from Magnetic Resonance Imaging (MRI) data. The core process involved foundational stages of image preprocessing, data augmentation, and hierarchical feature extraction using Convolutional Neural Networks (CNNs). Building upon these extracted features, several distinct approaches leveraging state-of-the-art AI architectures – namely Large Language Models (LLMs) and Vision-Language Models (VLMs) like CLIP – were implemented and evaluated as the final classification stage. Each step was designed to achieve high classification accuracy for differentiating Glioma, Meningioma, Pituitary tumors, and Tumor cases while rigorously exploring modern AI paradigms' capabilities and integration challenges in medical image analysis. The workflow of the foundational stages and the branching into different advanced classifiers is conceptually illustrated in Figure 3.1.

3.1 Overview of Approach

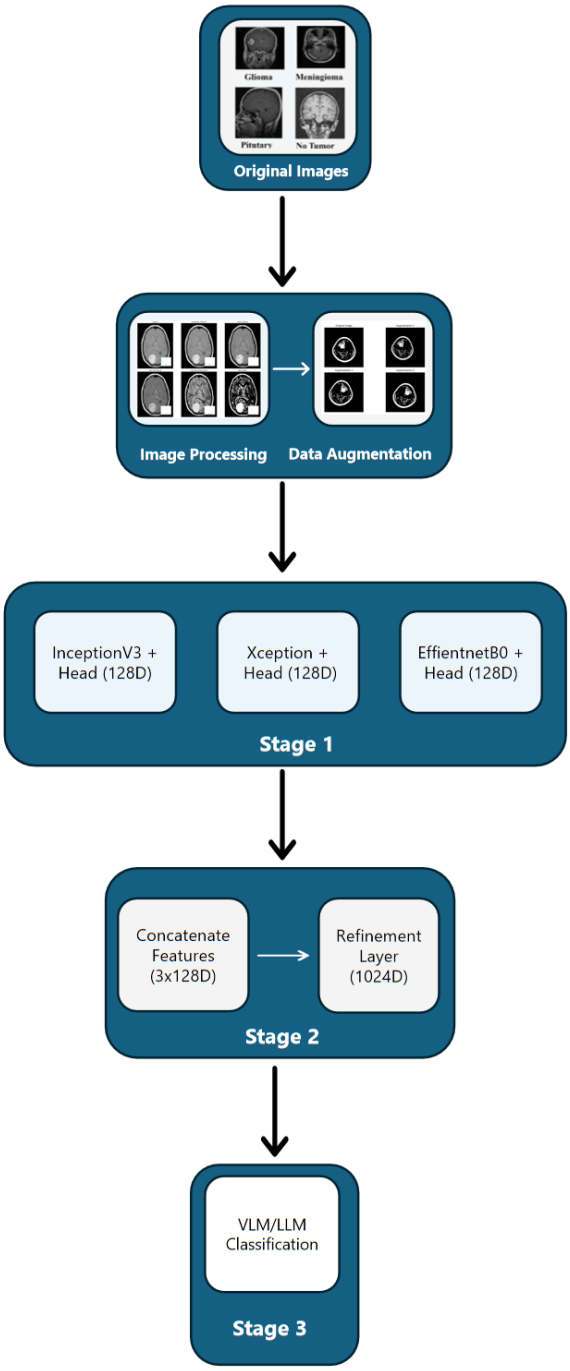


Figure 3.1: The process flows from MRI input, through preprocessing and augmentation, hierarchical feature extraction using parallel CNNs (Stage 1) followed by feature fusion (Stage 2), to the final advanced VLM/LLM classification stage (Stage 3).

The primary objective was to accurately classify brain MRI images into four clinically relevant categories. The methodology included key phases, utilizing a novel hierarchical feature engineering strategy before applying generative AI models.

1. **Foundational Data Preparation:** Application of a standardized image preprocessing pipeline (3.4.1) to enhance image quality and remove artifacts, followed by extensive data augmentation (3.4.2) applied to the training set to increase dataset size, diversity, and mitigate class imbalance.

2. **Hierarchical Feature Extraction (CNNs):** A multi-stage process to generate rich feature representations:

    - *Stage 1:* Parallel extraction of 128-dimensional features using three diverse, frozen pre-trained CNNs (EfficientNetB0, InceptionV3, Xception) adapted with small trainable heads (3.4.3).
    - *Stage 2:* Fusion and refinement of the Stage 1 features (concatenated to 384D) using a dedicated, regularized neural network to produce robust 1024-dimensional feature representations (3.4.4). These consolidated, high-dimensional features, derived from multiple diverse CNN perspectives, served as the primary input for the subsequent Gen AI classifiers, representing a departure from using raw images or single-source features directly.

3. **Gen AI Classification Techniques (3.5):** Exploration and comparison of different strategies using the refined 1024D features (or derived variants) as input:

    - *Attempted Direct LLM Injection (Informative):* Initial unsuccessful attempt to directly use 1024D features with an LLM (3.5.1).
    - *Quantized Feature LLM Classifiers:* Transforming 1024D features into discrete code sequences for input to frozen LLM backbones (Gemma 2B and RoBERTa Base) with trainable embedding/classification layers (3.5.2, 3.5.3).
    - *Direct Feature-Text Comparison (CLIP Zero-Shot):* Directly comparing specially prepared 512D image features against fixed CLIP text embeddings (3.5.4).
    - *Learned Feature Projection into CLIP Space (MLP Only):* Training a dedicated MLP projection layer to map 1024D features into CLIP's 512D space for comparison with fixed text embeddings (3.5.5).
    - *Learned Feature Projection with CLIP Text Fine-tuning:* Jointly training the MLP projection layer and fine-tuning the final layers of CLIP's text encoder (3.5.6).
    - *Baseline Direct Fine-Tuning of CLIP on Raw Images:* Fine-tuning the pre-trained CLIP ViT-B/32 model directly on raw MRI images, using cosine similarity between image and text embeddings for classification. This evaluates adapting the end-to-end CLIP architecture 3.5.7.

4. **Evaluation:** Rigorous assessment of all final classification models using standard metrics relevant to medical diagnosis (3.6).

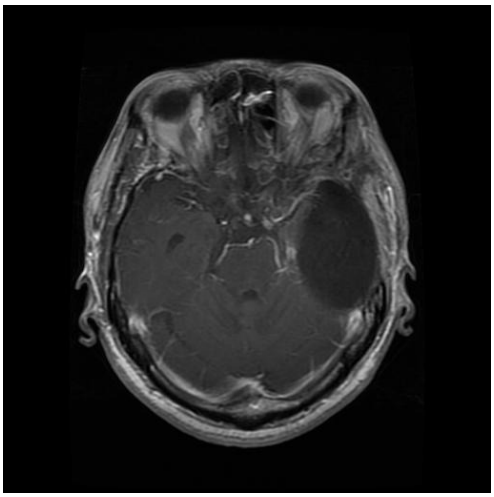
3.2 Dataset

3.2.1 Dataset Source and Characteristics

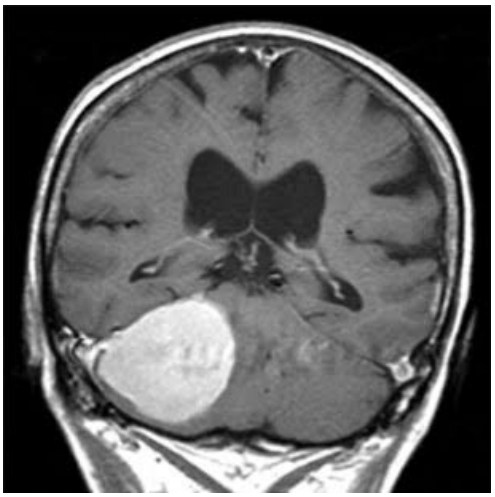
Accurate brain tumor classification from MRI is crucial for timely diagnosis and effective treatment planning. This study utilizes a curated dataset obtained from Kaggle [41], compiled explicitly for this task. The dataset comprises **7023 human brain MRI images**, categorized into four clinically relevant classes: **glioma**, **meningioma**, **pituitary** tumor, and **no tumor** (healthy). Figure 3.2 shows representative examples of each class.

The dataset aggregates images sourced initially from multiple public repositories, ensuring diversity. It includes images captured from various anatomical planes (axial, sagittal, and coronal), providing different perspectives of the brain structures.

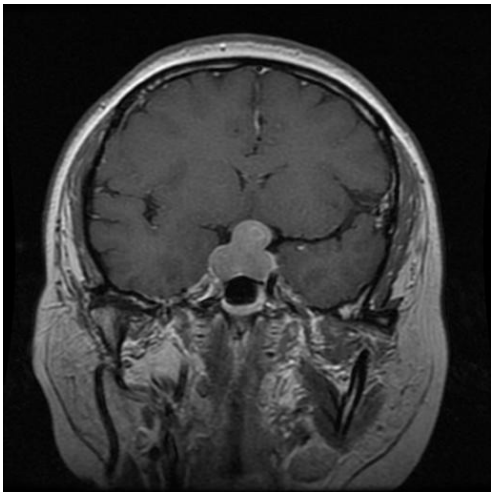
The images represent human brain MRIs. A key characteristic noted was the variability in original image dimensions and the potential presence of non-brain tissue (e.g., skull). Therefore, standardized preprocessing (3.4.1) was applied first, culminating in a **binarized (black and white) representation** of the brain region, before being fed into the CNNs in Stage 1 (3.4.3), the single channel of this **binary image** was replicated three times to match the required 3-channel input format (e.g., 224x224x3 or 299x299x3) of the ImageNet pre-trained models. The initial dataset exhibited some class imbalance, which was addressed through the data augmentation techniques described in Section 3.4.2.



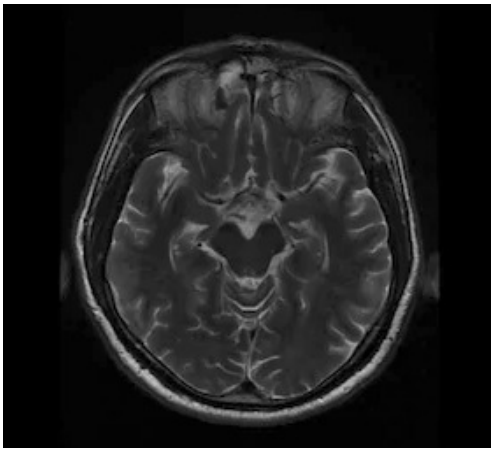
(a) Glioma Tumor



(b) Meningioma Tumor



(c) Pituitary Tumor



(d) No Tumor (Healthy)

Figure 3.2: Representative MRI examples for each of the four classes in the dataset: (a) Glioma, (b) Meningioma, (c) Pituitary tumor, and (d) No Tumor.

3.3 Experimental Environment

All experiments were conducted within the Kaggle Notebooks environment, providing access to GPU acceleration.

- **Hardware:** Computations were primarily accelerated using dual NVIDIA Tesla T4 GPUs (T4 x2 configuration).
- **Software:** The implementation stack was based on Python (`3.11`) and key libraries including: TensorFlow (`2.18.0`), Keras (`3.5.0`), KerasNLP (`0.18.1`), PyTorch (`2.6.0`).

- **Reproducibility:** A fixed random seed (`SEED = 42`) was employed wherever feasible (e.g., K-Means initialization, data splitting/shuffling, weight initializations in applicable frameworks, parameter search) to enhance the reproducibility of the results.

3.4 Foundational Feature Engineering Pipeline

This section details the initial stages focused on preparing the MRI data and extracting high-quality image features using CNNs. These features formed the basis for the subsequent advanced classification experiments.

3.4.1 Image Preprocessing Pipeline

Before model training, all raw MRI images underwent a standardized, multi-step preprocessing pipeline using the OpenCV library (`cv2`) in Python. This pipeline aimed to normalize the images, enhance diagnostically relevant features, and remove extraneous structures like the skull. The sequence of operations, based on `[42]`, was applied to each grayscale image, is conceptually illustrated in Figure 4.1 and detailed below:

1. **Anisotropic Diffusion Filtering (ADF):** The Perona-Malik model was applied (`num_iter=3`, `kappa=30`, `gamma=0.1`) to reduce noise while preserving important edge information.

2. **Skull Stripping:** A multi-step process isolated the brain parenchyma: Gaussian blurring (`kernel=(5,5)`), Otsu's thresholding, morphological closing (elliptical kernel `size=(5,5)`, `iterations=2`), connected components analysis (selecting the largest component), dilation (`MORPH_DILATE`, `kernel=(3,3)`, `iterations=2`), and contour-based hole filling. The resulting mask isolated the brain via a bitwise AND operation.

3. **Top-Hat Filtering:** Morphological top-hat filtering (using an elliptical structuring element, `kernel_size=15`) enhanced small, bright structures (potentially indicative of tumors) relative to their surroundings. The filtered result was added back to the skull-stripped image.

4. **Contrast Enhancement:** Global contrast was improved using standard Histogram Equalization (`cv2.equalizeHist`).

5. **Binarization:** The final step involved converting the contrast-enhanced grayscale image into a binary (black and white) format using a fixed threshold (`threshold_value=175`).

3.4.2 Data Augmentation

An offline data augmentation strategy was applied to both the training and test partitions after preprocessing to increase the effective size and diversity of the training data, mitigate overfitting, and address class imbalance. Implemented via a custom Python script using OpenCV and PIL:

- **Transformations Applied Randomly:** Rotation (-15 to +15 degrees), Width/Height Shift (up to 8%), Horizontal Flip (50% probability), Zoom (0.92x to 1.08x), Brightness Adjustment (0.85x to 1.15x). Fill mode used constant black padding (`cval=0`). Examples are shown in Figure 4.2.
- **Class Balancing:** Augmentations per image were scaled based on inverse class frequency (approx. 2x for tumor classes, 1x for no tumor), aiming for a more balanced training distribution.
- **Resulting Dataset:** The original 7023 images yielded an augmented dataset of approximately 15,695 training images and 3,528 test images. The final class distribution is visualized in Figure 4.3.

These augmented training and test sets were used for all subsequent model training and evaluation phases.

3.4.3 Stage 1: Parallel CNN Feature Extraction

This stage aimed to extract diverse, complementary features using three different pre-trained CNNs in parallel. Each network branch produced a 128-dimensional feature vector.

- **Common Architecture Head:** Each frozen CNN base (EfficientNetB0, InceptionV3, Xception) had a common trainable head attached: `GlobalAveragePooling2D` $\rightarrow$ Trainable `Dense`(128, activation='relu', kernel_regularizer=L2(1e-4), name='feature_dense_layer') $\rightarrow$ `Dropout`(0.4) $\rightarrow$ Temporary `Dense`(4, activation='softmax').
- **Training Goal:** The temporary classification layer provided a loss signal to train the weights of the `'feature_dense_layer'` specifically for this dataset.
- **Branches (Key Differences):**
  - **EfficientNetB0:** Input 224x224x3, `efficientnet_preprocess_input`. Trained 25 epochs (best epoch 21).
  - **InceptionV3:** Input 299x299x3, `inception_v3_preprocess_input`. Trained 23 epochs (best epoch 16).
  - **Xception:** Input 299x299x3, `xception_preprocess_input`. Trained 21 epochs (best epoch 14).
- **Common Training Parameters:** Adam optimizer (`lr=0.001`), batch size 32, EarlyStopping (`patience=7`).
- **Output:** 128-dimensional features extracted from the trained `'feature_dense_layer'` of each branch for both training and test sets.

3.4.4 Stage 2: Feature Combination and Refinement

This stage fused the parallel features from Stage 1 and trained a dedicated network to learn higher-level interactions, producing the final 1024-dimensional features used by subsequent advanced classifiers.

- **Input Features:** Concatenated 128D features from the three Stage 1 branches, resulting in 384D vectors per sample.
- **Model Architecture (`Combined_E0_I3_X_RegV2`):** Feed-forward network: Input(384) $\rightarrow$ Dense(1024, no bias, kernel_regularizer=L2(1.5e-3)) $\rightarrow$ BatchNormalization $\rightarrow$ ReLU Activation (name='combined_feature_layer') $\rightarrow$ Dropout(0.65) $\rightarrow$ Dense(4, softmax).
- **Training:** Trained on 384D features using Adam (`lr=0.0005`), sparse_categorical_crossentropy, batch size 64, up to 40 epochs. Used EarlyStopping(`patience=10`) and ReduceLROnPlateau(`patience=5`). Best weights from epoch 28 restored (approx. 96.7% validation accuracy achieved during training).
- **Output:** Final 1024-dimensional features extracted from the output of the 'combined_feature_layer' for train/test sets.

3.5 Gen AI Classification Techniques

Building upon the refined 1024-dimensional features from Stage 2 (3.4.4), which represent a distilled, multi-perspective visual summary, this section details the various Gen AI techniques explored. The central hypothesis is that these rich, hierarchically engineered features provide a more potent input for advanced models like LLMs and VLMs than raw or minimally processed images for this specific task. For LLMs, the quantized 1024D features form a 'pseudo-language' allowing them to leverage their sequential understanding. For VLMs like CLIP, these comprehensive features offer a robust visual representation to be aligned with semantic text prompts. The use of these rich, engineered features as input and the subsequent adaptation strategies represent key aspects of the methodology explored, aiming to bridge domain-specific visual information with the powerful reasoning capabilities of modern AI.

3.5.1 Attempted Direct LLM Feature Injection (Informative)

An initial attempt was made to directly input the 1024D numerical features from Stage 2 into a frozen Gemma 2B LLM backbone. This proved infeasible due to fundamental architectural incompatibilities between continuous features and the discrete, sequential token processing expected by the LLM's transformer architecture. This underscored the need for feature transformation strategies like quantization.

3.5.2 Quantized Feature LLM Classifier (Gemma 2B)

This method successfully integrated an LLM by converting the continuous, information-rich 1024D features into discrete sequences. This transformation aimed to create a 'feature language' where the LLM could apply its inherent strengths in modeling sequential dependencies and contextual relationships, analogous to how it processes natural language text.

- **Input:** 1024D features (3.4.4).
- **Quantization:** K-Means (`n_clusters=256`) applied to 32x32D sub-vectors (trained on training data only) to generate sequences of 32 integer codes (0-255) per image feature.
- **Model (`Quantized_gemma_2b_v6_final`):** Keras model: Input(32 codes) → Trainable Embedding(256 → 2048) → Frozen GemmaBackbone("gemma_2b") → Pooling → Dropout(0.2) → Trainable Dense(4, softmax).
- **Training:** Adam (`lr=1e-4`), batch 32, 30 epochs, EarlyStopping(`patience=10`), ReduceLROnPlateau(`patience=3`).
- **Performance:** Achieved **94.30%** test accuracy (4.3.1).

3.5.3 Quantized Feature LLM Classifier (RoBERTa Base - Comparative)

This experiment replicated the quantization approach using RoBERTa Base for comparison.

- **Input:** Identical quantized sequences as for Gemma (3.5.2).
- **Model (`Quantized_RoBERTa_base_v1`):** Keras model: Input(32 codes) → Trainable Embedding(256 → 768) → Frozen RoBERTaBackbone(" RoBERTa_base") → Pooling → Dropout(0.2) → Trainable Dense(4, softmax). Embedding dimension-matched RoBERTa.
- **Training:** Identical procedure to Gemma.
- **Performance:** Achieved **93.57%** test accuracy (see 4.3.1).

3.5.4 Direct Feature-Text Comparison (CLIP Zero-Shot)

This VLM approach tested zero-shot classification. The rationale was to explore if the engineered 512D image features (instead of the 1024D set) were already sufficiently aligned with CLIP's general semantic understanding to be classified directly against fixed CLIP text embeddings without further adaptation.

- **Input Image Features:** Utilized **512-dimensional** feature vectors. These were specifically generated for this experiment by replacing the trainable 1024 Dense layer with a trainable 512 Dense layer,which replaces the 1024D features with 512D features. The dimension matched CLIP's (`ViT-B-32`).

- **CLIP Model & Text Prompts:** `ViT-B-32` CLIP model (`openai` weights) via `open_clip_torch`. Standard class prompts used (e.g., "an MRI showing a glioma tumor").
- **Zero-Shot Classification:** Fixed, L2 normalized 512D text embeddings generated using the frozen CLIP text encoder. Input 512D image features were L2 normalized. Class predicted based on the highest cosine similarity. No parameters were trained during this specific classification step.
- **Implementation:** PyTorch, `open_clip_torch`.
- **Performance:** Resulted in poor test accuracy of **13.66%** (see 4.3.2).

3.5.5 Learned Feature Projection into CLIP Space (MLP Only)

This approach explicitly learned a mapping from the 1024D CNN features into CLIP's 512D space, using fixed text embeddings.

- **Input:** Original **1024D** features (3.4.4).
- **Targets:** Fixed, L2 normalized 512D text embeddings from frozen `ViT-B-32` CLIP.
- **Trainable Model:** Simple MLP: Input(1024D) $\rightarrow$ Dropout(0.4) $\rightarrow$ Trainable Linear(1024 $\rightarrow$ 512) $\rightarrow$ Output L2 Normalization.
- **Training:** Trained using CrossEntropyLoss on cosine similarities. AdamW optimizer (`lr=1e-4`, `weight_decay=2e-3`). Employed ReduceLROnPlateau(`patience=3`) and EarlyStopping(`patience=5`). Stopped early (epoch 28/30).
- **Classification:** Input 1024D features $\rightarrow$ Trained MLP $\rightarrow$ 512D projected feature $\rightarrow$ Compare via cosine similarity to fixed 512D text embeddings.
- **Implementation:** PyTorch, `open_clip_torch`, `scikit-learn`.
- **Performance:** Achieved **95.72%** test accuracy (see 4.3.2).

3.5.6 Learned Feature Projection with CLIP Text Fine-tuning

This advanced approach combined the learned projection with fine-tuning of the CLIP text encoder itself.

- **Input:** Original **1024D** features (3.4.4).
- **CLIP Model:** `ViT-B-32` CLIP (`openai` weights). Text embeddings generated dynamically during training.
- **Trainable Components (Jointly Optimized):**

  - **Projection MLP:** Input(1024D) → Linear(1024 → 768) → GELU → Dropout(0.4) → Linear(768 → 512) → Output L2 Normalization.
  - **CLIP Text Encoder:** Final transformer block (`CLIP_LAYERS_TO_TUNE=1`) and final Layer Normalization/projection layer unfrozen.

- **Training Procedure:**
  - **Loss:** CrossEntropyLoss on cosine similarities between MLP output and dynamically generated text embeddings.
  - **Optimizer:** AdamW with parameter groups: MLP (`lr=1e-4`), Unfrozen CLIP Text (`lr=1e-6`, i.e., `1e-4 * 0.01`). Weight decay `1e-3`.
  - **Regularization & Callbacks:** Dropout(0.4), weight decay, mixed-precision (`torch.amp`), ReduceLROnPlateau(`patience=3`), EarlyStopping(`patience=7`).
  - **Outcome:** Stopped early (epoch 20/30), restoring best weights (≈96.8% validation accuracy).

- **Classification:** Input 1024D features → Trained MLP → 512D projected feature → Compare via cosine similarity to the final fine-tuned 512D text embeddings.

- **Implementation:** PyTorch, `open_clip_torch`, `scikit-learn`.

- **Performance:** Achieved the highest test accuracy: **96.83%** (see 4.3.2).

3.5.7 Direct Fine-Tuning of CLIP on Raw Images (Benchmark)

This benchmark approach involves fine-tuning the pre-trained CLIP ViT-B/32 model directly on the raw MRI images to classify brain tumors. This method serves to evaluate the performance of adapting the end-to-end CLIP architecture to the specific task by updating its image and text encoder weights.

- **Input:** Raw MRI images from the training, validation, and test sets, preprocessed using CLIP's specific preprocessing transformations.

- **CLIP Model Used for Fine-Tuning:** Full `ViT-B/32` CLIP model (openai/clip-vit-base-patch32 weights from `open_clip_torch`).

- **Fine-Tuning Strategy:**
  - The final **1** transformer block of the vision encoder, along with its final layer normalization and image projection layer, were made trainable.
  - The final **1** transformer block of the text encoder, along with its final layer normalization and text projection layer, were made trainable.
  - All other CLIP parameters remained frozen.

- **Text Prompts:** Class-specific prompts were used during training and evaluation (e.g., "an MRI scan showing a glioma tumor").
- **Classification Mechanism:**
  - Input images are passed through the fine-tuned CLIP vision encoder to obtain 512D image embeddings (L2-normalized).
  - Text prompts for each class are passed through the fine-tuned CLIP text encoder to obtain 512D text embeddings (L2-normalized).
  - The **cosine similarity** is computed between each image embedding and all class text embeddings.
  - The class corresponding to the text prompt with the highest cosine similarity is the predicted class.
- **Training Procedure:**
  - **Optimizer:** AdamW (`lr=1e-5`, `weight_decay=1e-4`).
  - **Loss Function:** CrossEntropyLoss (applied to the cosine similarity scores).
  - **Epochs:** Up to 10, with early stopping (`patience=3`) based on validation loss.
  - **Scheduler:** ReduceLROnPlateau.
  - Best model weights based on validation loss were saved and used for final testing.
- **Achieved Performance on Test Set: 90.96%** accuracy (see Section 4.3.2 and Table 4.5, Figure 4.17).

3.6 Evaluation Metrics

The performance of the final classification models developed using the different Gen AI techniques (3.5.2 through 3.5.6), as well as intermediate models like the Stage 2 classifier and the baseline CLIP model (3.5.7), was rigorously evaluated on their respective independent test sets. A standard suite of metrics suitable for medical diagnosis tasks was employed.

The core components derived from the confusion matrix for each class are: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Based on these, the following metrics were calculated:

- **Accuracy:** Overall proportion of correct predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.1}$$

- **Precision:** Accuracy of positive predictions ($\frac{\text{TP}}{\text{TP+FP}}$). Calculated per class and as a weighted average.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

- **Recall (Sensitivity):** Ability to identify actual positives ($\frac{\text{TP}}{\text{TP+FN}}$). Calculated per class and as a weighted average.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

- **F1-Score:** Harmonic mean of Precision and Recall ($2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$). Calculated per class and as a weighted average.

$$\text{F1-Score} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{3.4}$$

- **Confusion Matrix:** A table visualizing performance, showing TP, TN, FP, and FN counts for all classes.

These metrics collectively provide a comprehensive assessment of model diagnostic capabilities. Detailed numerical results and comparative analysis are presented in Chapter 4.

Chapter 4

Results

4.1 Introduction

This chapter outlines the empirical results obtained from the experiments detailed in the Methodology (Chapter 3). The findings cover the outcomes of the data preparation stages, the performance of the intermediate feature extraction models, and, critically, a comparative evaluation of the various Gen AI classification techniques employed. The results are assessed using the standard evaluation metrics defined in 3.6, including accuracy, precision, recall, F1-score, specificity, and confusion matrices, to provide a comprehensive view of each model's ability to classify brain tumors from MRI data. This analysis highlights the effectiveness of the proposed hierarchical feature engineering pipeline feeding into advanced classifiers compared to baseline approaches.

4.2 Data Preparation Outcomes

The initial stages focused on preparing the MRI data for subsequent analysis.

4.2.1 Image Preprocessing Visualization

The standardized preprocessing pipeline (3.4.1) was applied to all images. Figure 4.1 illustrates the key steps of this pipeline on a representative sample MRI image, showcasing the effects of noise reduction, skull stripping, and contrast enhancement.

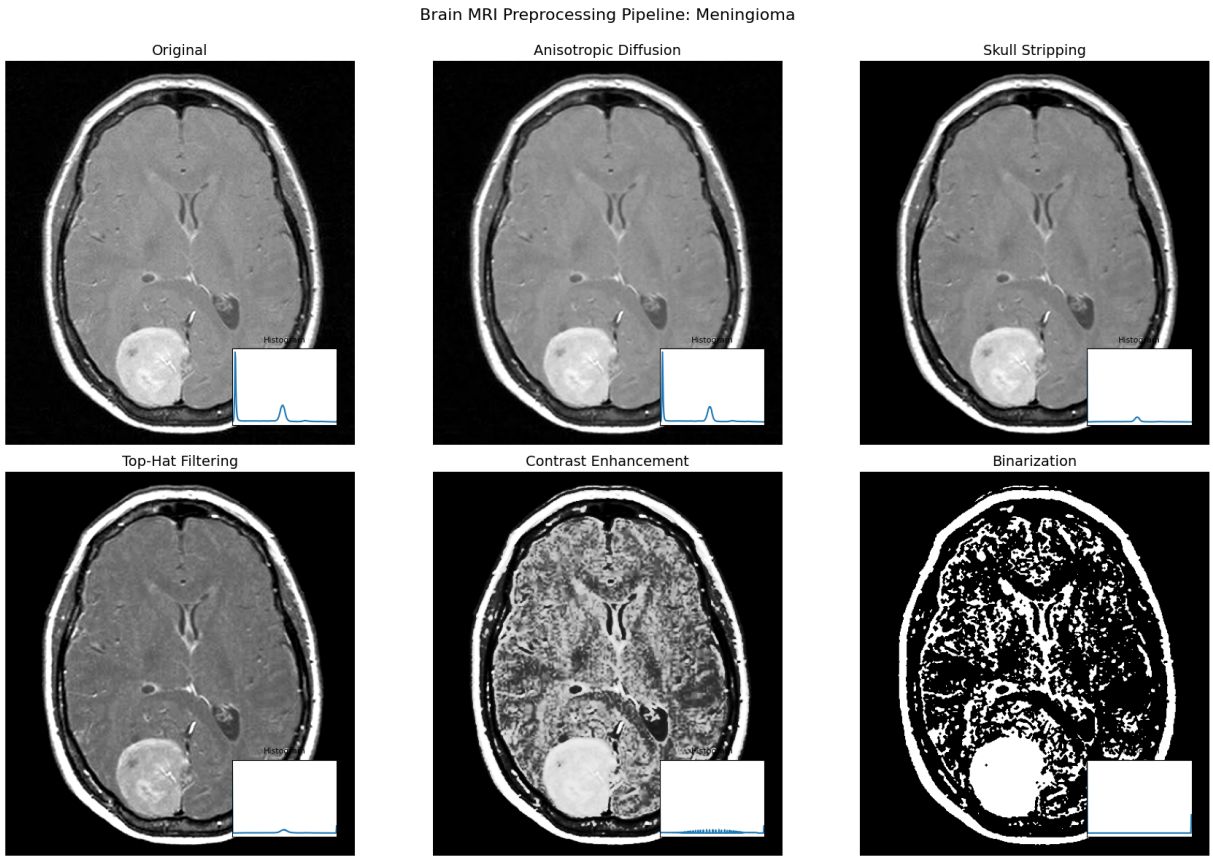


Figure 4.1: Example visualization of the image preprocessing pipeline steps applied to a sample MRI slice. (A) Raw Image, (B) After Anisotropic Diffusion, (C) After Skull Stripping, (D) After Top-Hat Enhancement, (E) Contrast Enhanced Grayscale Image, (F) After Binarization (Illustrative).

4.2.2 Data Augmentation Visualization and Distribution

Data augmentation (3.4.2) significantly expanded both the training and test datasets. Figure 4.2 displays examples of the transformations applied to a single image instance, demonstrating the increased diversity introduced.

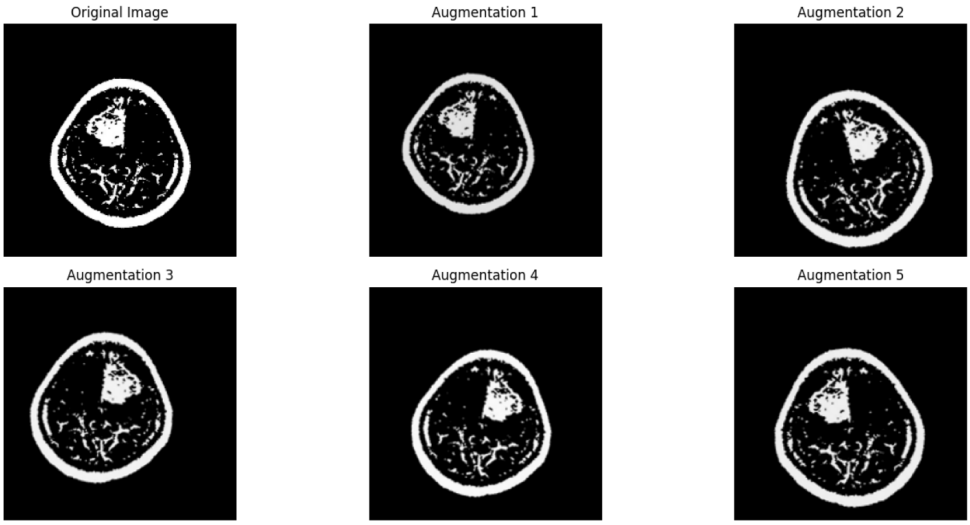
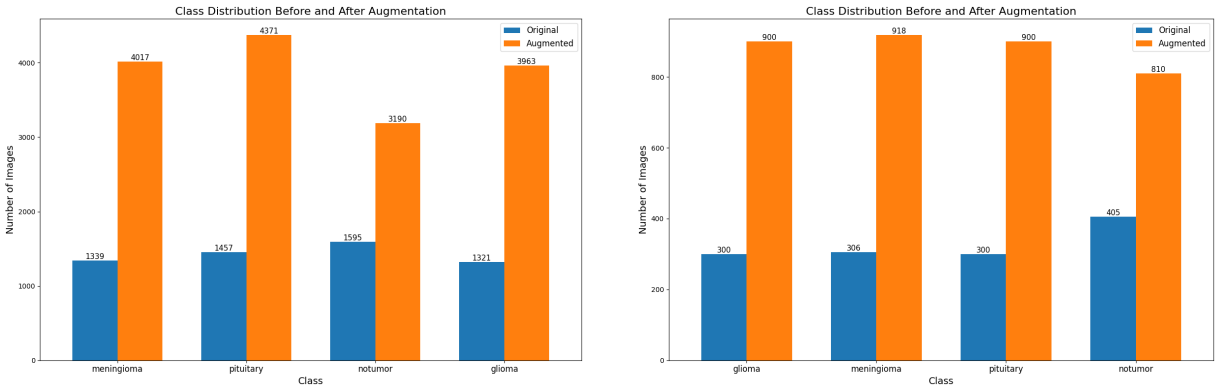


Figure 4.2: Examples of data augmentation techniques applied to a sample preprocessed MRI image, including rotation, shifting, zooming, flipping, and brightness adjustment.

Furthermore, the augmentation strategy aimed to mitigate the class imbalance present in the original dataset. Figure 4.3.



(a) Class distribution in the augmented Training dataset (Improved balance).

(b) Class distribution in the augmented Testing dataset (Improved balance).

Figure 4.3: Class distributions after augmentation for the Training dataset (left) and Testing dataset (right), both showing improved balance.

4.2.3 Stage 1: Parallel CNN Feature Extractor Performance

The three parallel CNN branches (EfficientNetB0, InceptionV3, Xception) were trained with a temporary classification head to optimize their respective 128-dimensional feature layers, as described in 3.4.3. While their primary role was feature extraction, their classification perfor-

mance provides insight into their individual capabilities on this task. Table 4.1 summarizes the performance metrics achieved by each branch's temporary classifier on the test set.

Table 4.1: Comparative Performance Metrics of Stage 1 Branch Classifiers (on Test Set)

	EfficientNetB0			**Xception**			**InceptionV3**		
Metric	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Glioma	0.93	0.91	0.92	0.90	0.86	0.88	0.88	0.86	0.87
Meningioma	0.90	0.92	0.91	0.84	0.83	0.83	0.83	0.83	0.83
No Tumor	0.98	1.00	0.99	0.94	0.99	0.96	0.94	0.98	0.96
Pituitary	0.97	0.97	0.97	0.94	0.95	0.94	0.95	0.94	0.94
Accuracy			0.95			0.90			0.90
Macro Avg	0.95	0.95	0.95	0.90	0.91	0.90	0.90	0.90	0.90
Weighted Avg	0.95	0.95	0.95	0.90	0.90	0.90	0.90	0.90	0.90

Note: These metrics reflect the performance of the temporary classification heads used to train the 128D feature layers, not the final system performance. Prec. = Precision, Rec. = Recall, F1 = F1-Score.

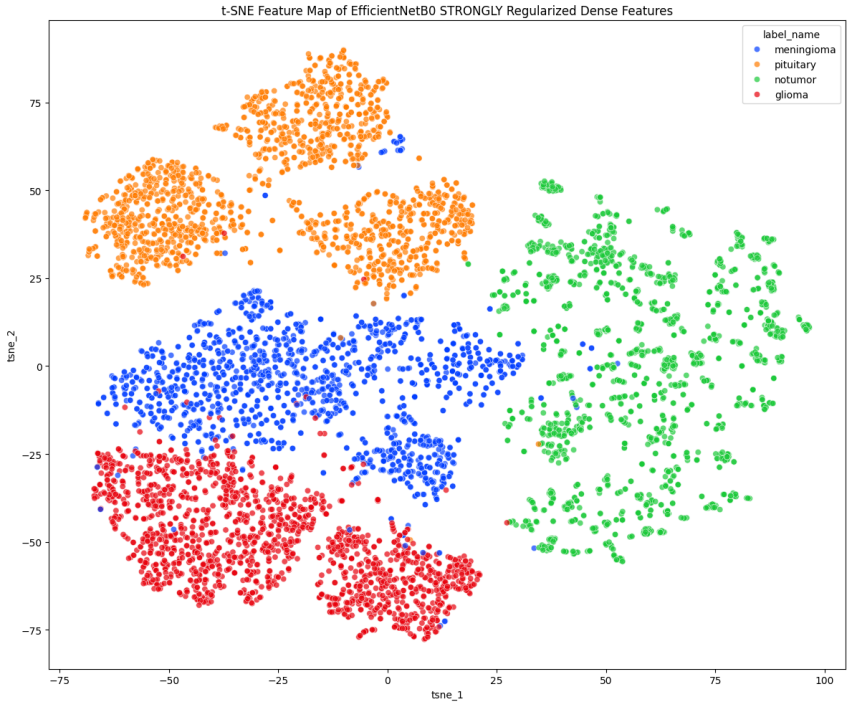


Figure 4.4: t-SNE visualization of the 128-dimensional features extracted from the Stage 1 **EfficientNetB0** branch. Colors represent the different brain tumor classes (glioma, meningioma, notumor, pituitary).

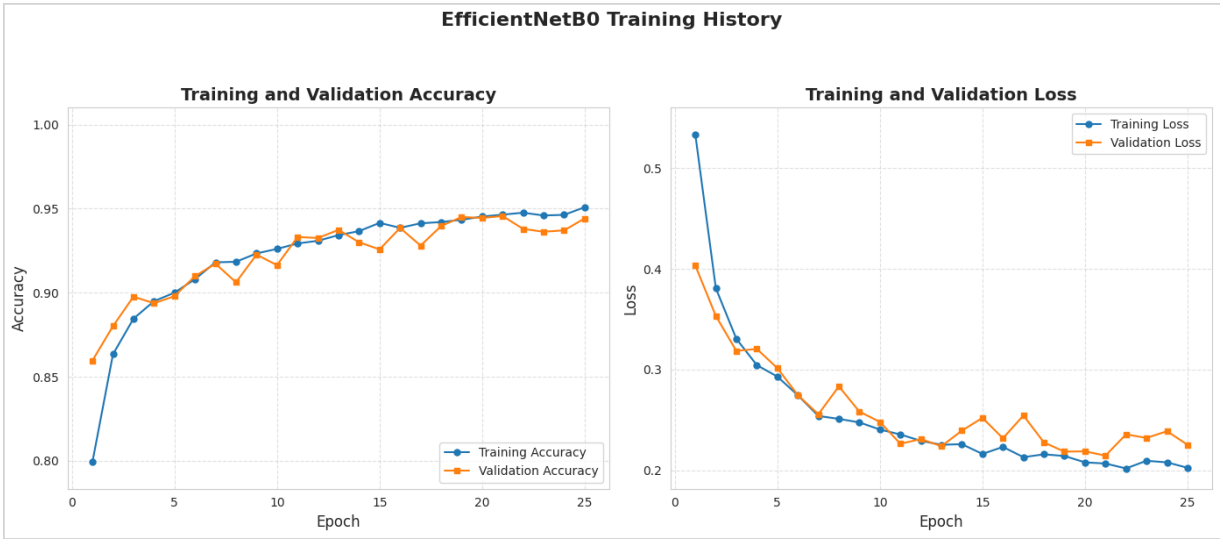


Figure 4.5: Training and validation history (accuracy and loss) for the Stage 1 **EfficientNetB0** branch classifier head used to train the feature extraction layer.

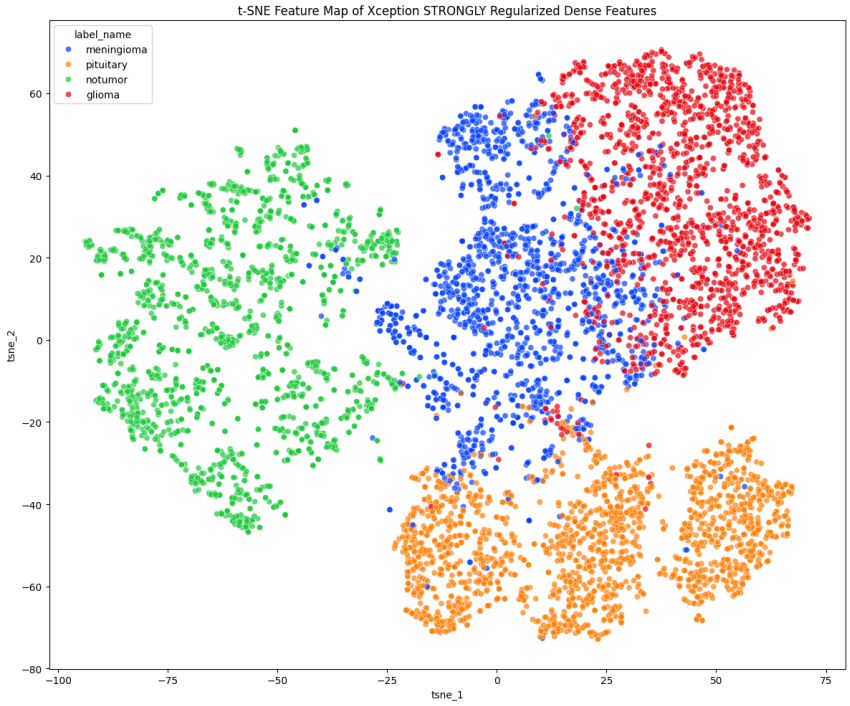


Figure 4.6: t-SNE visualization of the 128-dimensional features extracted from the Stage 1 **Xception** branch, showing the class clustering.

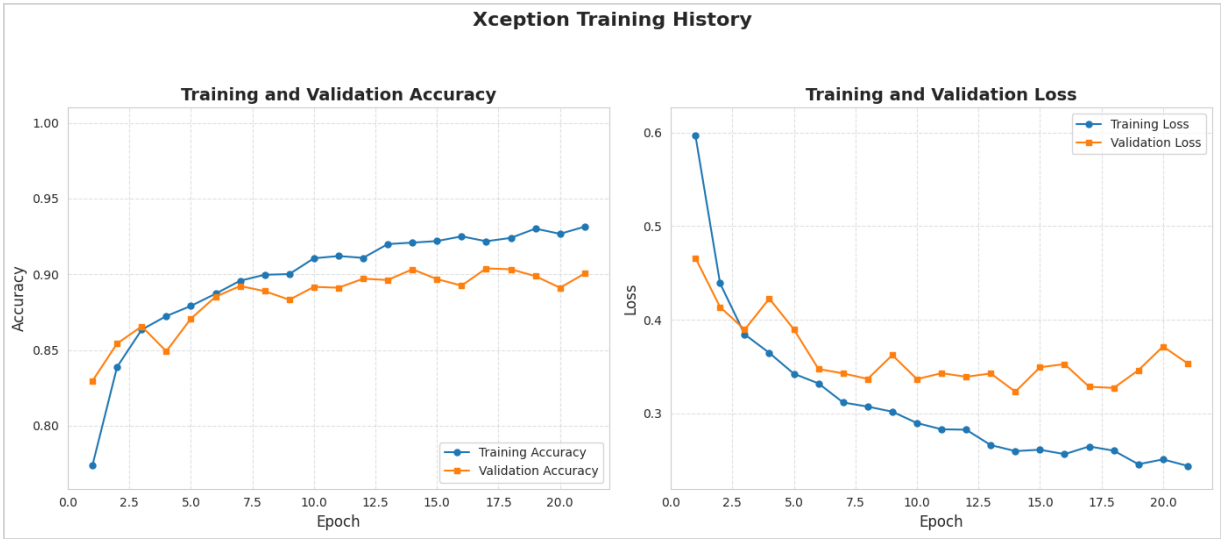


Figure 4.7: Training and validation history (accuracy and loss) for the Stage 1 **Xception** branch classifier head.

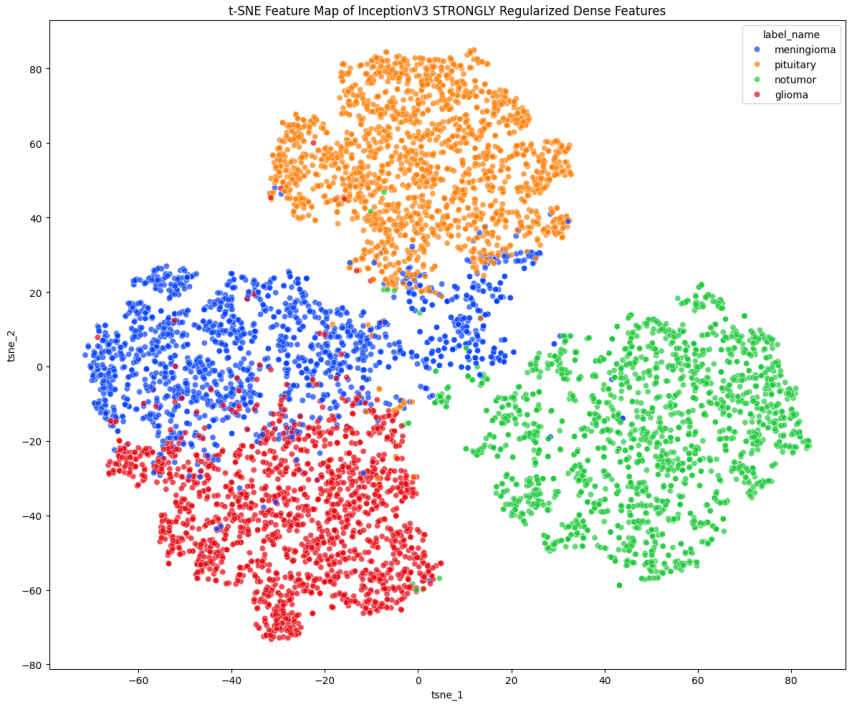


Figure 4.8: t-SNE visualization of the 128-dimensional features extracted from the Stage 1 **InceptionV3** branch, illustrating feature space separability.

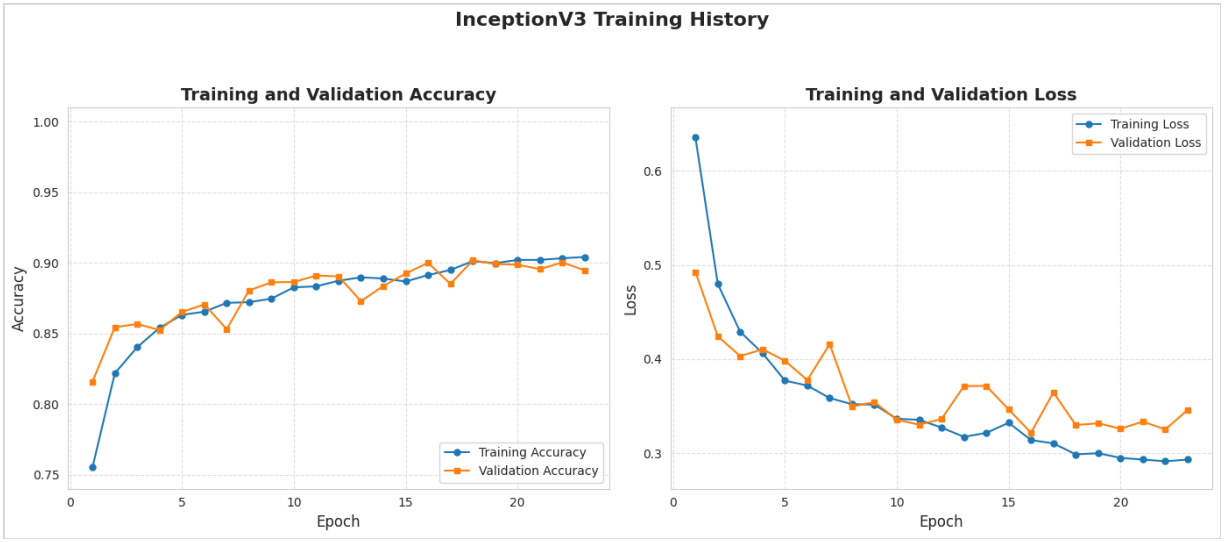


Figure 4.9: Training and validation history (accuracy and loss) for the Stage 1 **InceptionV3** branch classifier head.

4.2.4 Stage 2: Combined Feature Classifier Performance

The Stage 2 model (`Combined_E0_I3_X_RegV2`, 3.4.4) was trained on the concatenated 384-dimensional features from Stage 1. This model served both as a classifier itself and as the extractor for the final 1024-dimensional features. Its performance on the test set, achieving an accuracy of 97%, is detailed in Table 4.2. The corresponding confusion matrix is shown in Figure 4.10.

Table 4.2: Performance Metrics of the Stage 2 Classifier (`Combined_E0_I3_X_RegV2`) on the Test Set (1024D Feature Extractor)

Class	Precision	Recall	F1-Score
Glioma	0.97	0.93	0.95
Meningioma	0.93	0.96	0.94
No Tumor	0.99	1.00	0.99
Pituitary	0.98	0.98	0.98
Accuracy			0.97
Macro Avg	0.97	0.97	0.97
Weighted Avg	0.97	0.97	0.97

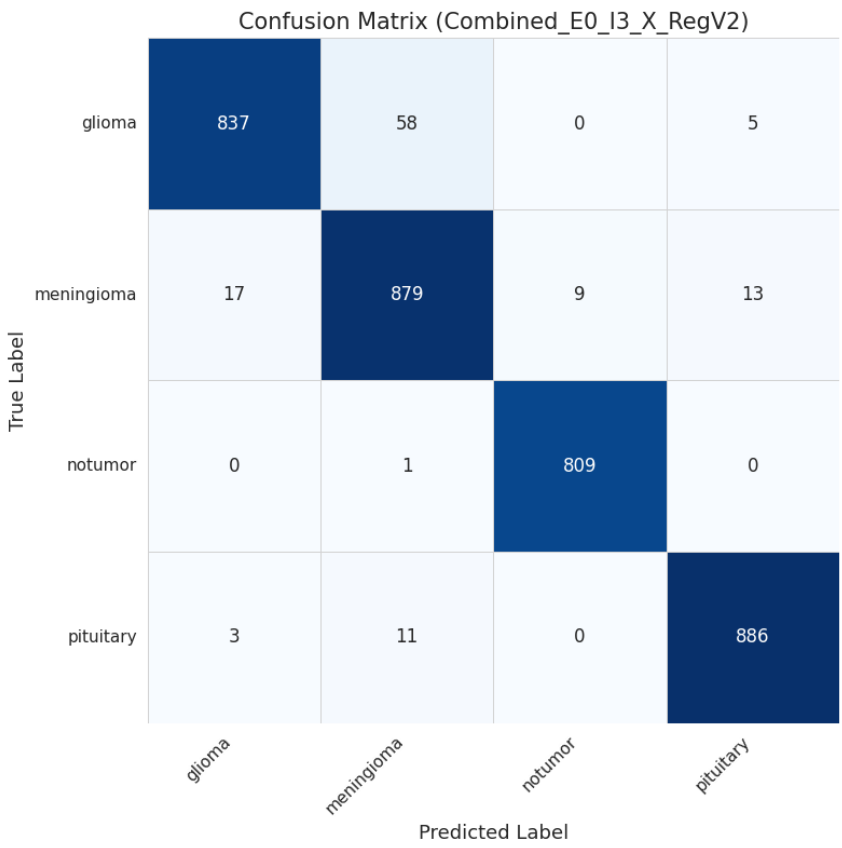


Figure 4.10: Confusion matrix for the Stage 2 feature combination model evaluated on the test set.

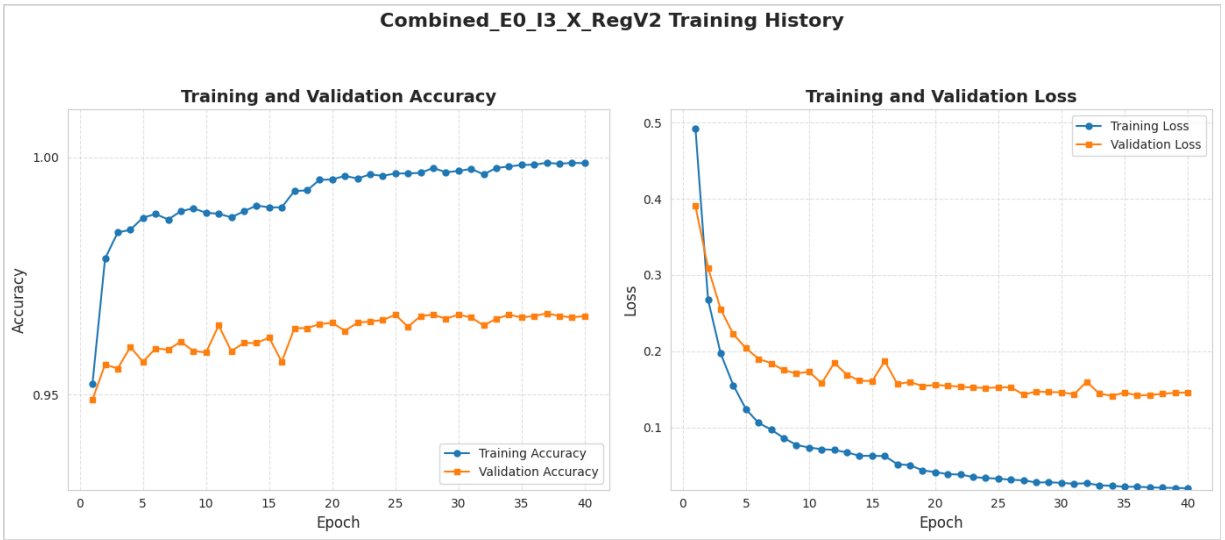


Figure 4.11: Training and validation history (accuracy and loss) for the Stage 2 feature combination model. This model takes the 384D features from Stage 1 as input.

This model demonstrates strong classification ability independently and provides the high-dimensional features utilized in the subsequent Gen AI experiments.

4.3 Gen AI Classifier Performance

This section presents the core results, evaluating the performance of the different Gen AI classification techniques detailed in 3.5, primarily using the 1024D features from Stage 2 as input (except where noted).

4.3.1 Quantized Feature LLM Classifier Results

The feature quantization approach enabled the use of LLMs for classification, demonstrating a viable path for integrating pre-trained language models with engineered visual features.

Gemma 2B Variant

Before successfully applying feature quantization, an initial exploratory attempt was made to directly integrate the 1024-dimensional continuous features from Stage 2 into the frozen Gemma 2B LLM backbone. This approach involved projecting the 1024D features to the LLM's hidden dimension and providing dummy token IDs, with the hypothesis that the LLM might directly process these projected continuous features.

However, this direct injection method yielded exceptionally poor performance. The model achieved a test accuracy of only 26.02%, comparable to random guessing for a 4-class problem. The confusion matrix for this attempt, presented in Figure 4.12, illustrates that the model predominantly classified all inputs as 'meningioma', irrespective of the true label. This outcome strongly suggested that the frozen Gemma backbone was unable to effectively process the projected image features in this direct manner, likely defaulting to processing the dummy token IDs. This experiment underscored the architectural incompatibility of directly feeding high-dimensional continuous features into standard pre-trained LLM backbones without specific feature transformation.

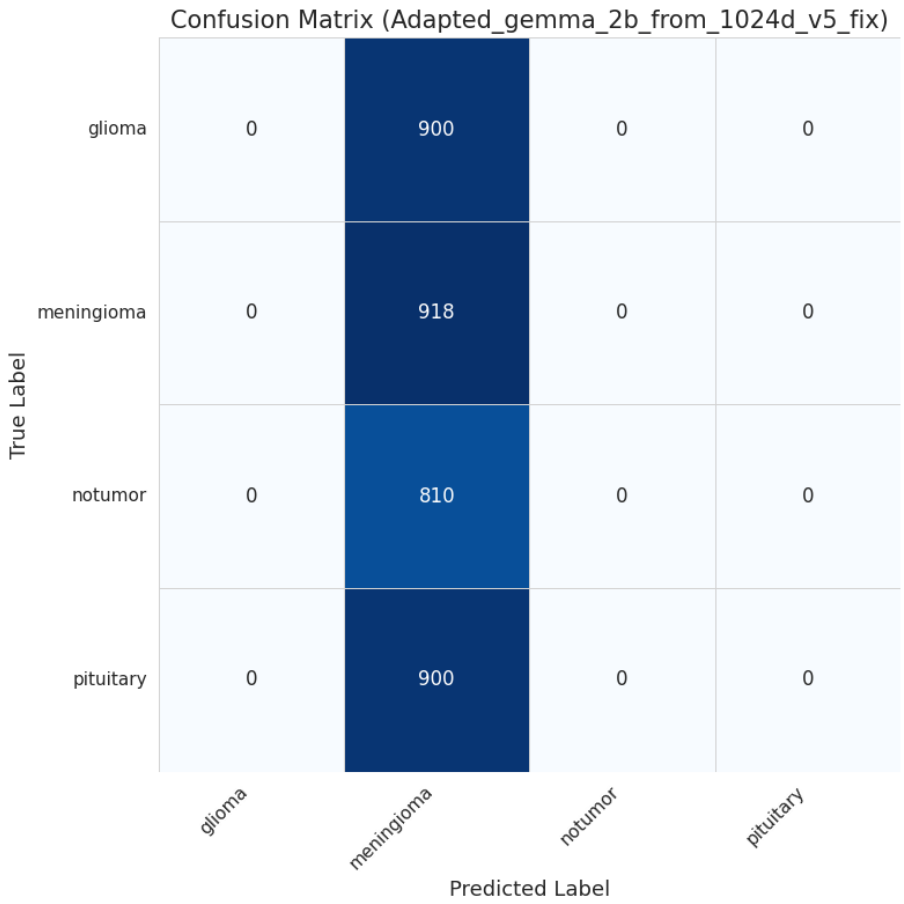


Figure 4.12: Confusion matrix for the Attempted Direct 1024D Feature LLM Injection (Gemma 2B) on the test set, illustrating the model's collapse. This attempt preceded the successful quantization approach.

The failure of this direct approach highlighted the necessity of transforming the continuous features into a format more amenable to the LLM's architecture. Consequently, feature quantization was employed. The performance of the classifier using the frozen Gemma 2B backbone with these quantized features (as described in Section 3.5.2) on the test set is presented in Table 4.3 and Figure 4.13. This quantized approach proved significantly more effective, with the model achieving an accuracy of 94.30%.

Table 4.3: Performance Metrics of the Quantized Feature LLM Classifier (Gemma 2B) on the Test Set

Class	Precision	Recall	F1-Score
Glioma	0.96	0.90	0.93
Meningioma	0.88	0.93	0.91
No Tumor	0.97	0.99	0.98
Pituitary	0.96	0.96	0.96
Accuracy			0.94
Macro Avg	0.95	0.94	0.94
Weighted Avg	0.94	0.94	0.94

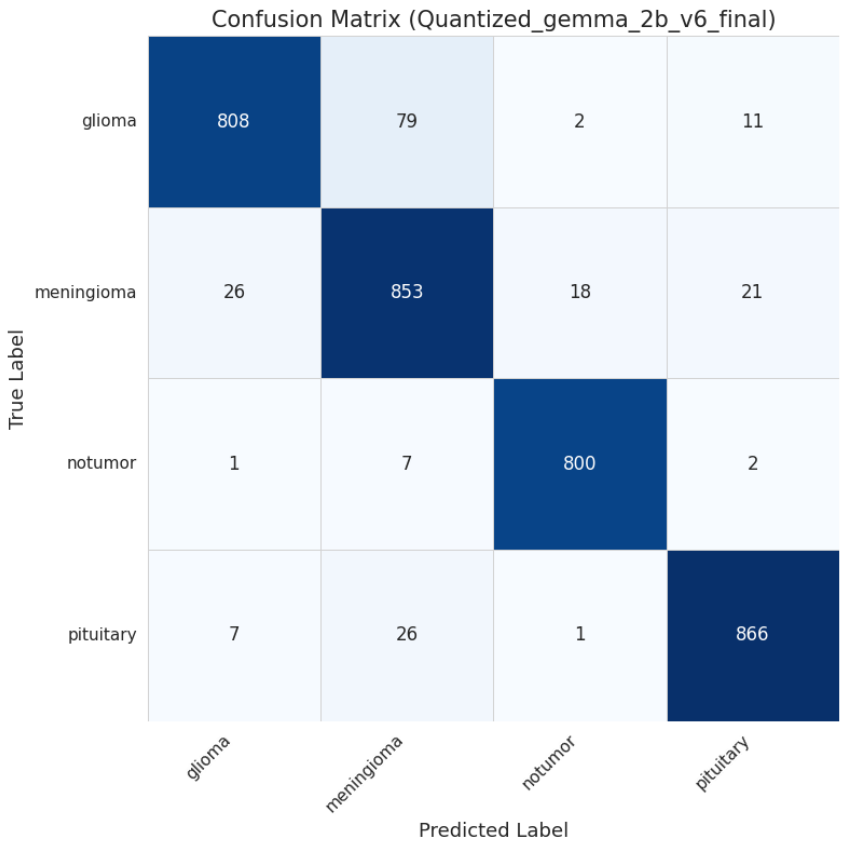


Figure 4.13: Confusion matrix for the Quantized Feature LLM Classifier (Gemma 2B) on the test set.

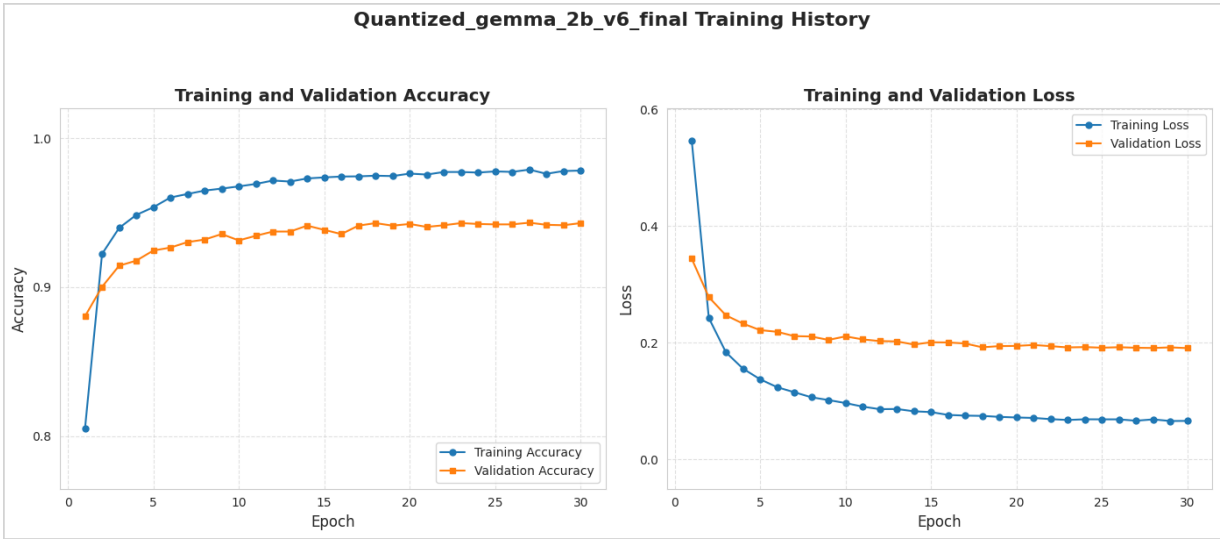


Figure 4.14: Training and validation history (accuracy and loss) for the Quantized Feature LLM Classifier using the **Gemma 2B** backbone

RoBERTa Base Variant

For comparison, the RoBERTa Base backbone was substituted for Gemma (3.5.3). Its performance, reaching 93.57% accuracy, is shown in Table 4.4 and Figure 4.15.

Table 4.4: Performance Metrics of the Quantized Feature LLM Classifier (RoBERTa Base) on the Test Set

Class	Precision	Recall	F1-Score
Glioma	0.9461	0.8778	0.9107
Meningioma	0.8698	0.9390	0.9031
No Tumor	0.9803	0.9827	0.9815
Pituitary	0.9584	0.9478	0.9531
Accuracy			0.9357
Macro Avg	0.9387	0.9368	0.9371
Weighted Avg	0.9373	0.9357	0.9358

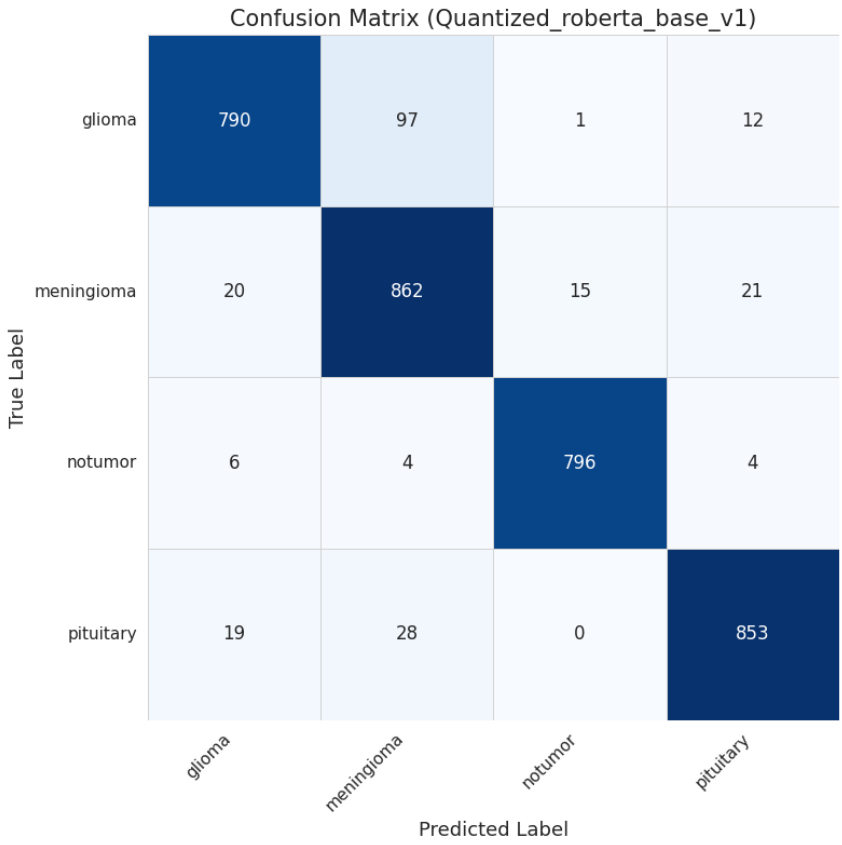


Figure 4.15: Confusion matrix for the Quantized Feature LLM Classifier (RoBERTa Base) on the test set.

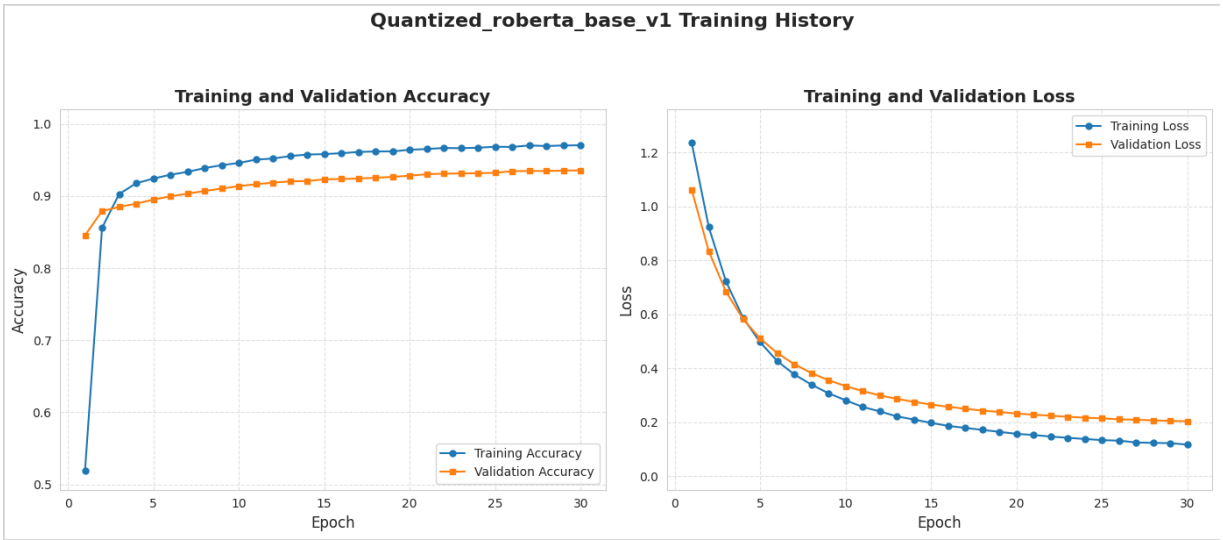


Figure 4.16: Training and validation history (accuracy and loss) for the Quantized Feature LLM Classifier using the **RoBERTa Base** backbone.

4.3.2 VLM-Based Classifier Results

Alternative approaches using the Vision-Language Model CLIP were also evaluated. These methods leverage CLIP's ability to understand relationships between images (or derived image features) and text descriptions. A key aspect of these techniques is their potential to provide richer output beyond a simple class label. By comparing the image representation against text embeddings for each class (e.g., "an MRI showing a glioma tumor"), these models naturally produce **similarity scores**. These scores can then be converted into **probabilities** (e.g., via softmax), offering insights into the model's confidence and potential ambiguities in its predictions. The following subsections detail the performance of different CLIP-based integration strategies explored in this work (3.5.4 to 3.5.6), followed by a qualitative example (4.3.2).

Benchmark: Direct Fine-Tuning of CLIP on Raw Images

To establish a strong benchmark utilizing the CLIP architecture more directly with the image data, the CLIP ViT-B/32 model was fine-tuned on the raw MRI images, as detailed in Section 3.5.7. This involved making the final layers of both the vision and text encoders trainable and performing classification based on the cosine similarity between the resulting image and text embeddings. This fine-tuned CLIP model achieved an overall test accuracy of **90.96%**. The detailed performance metrics are presented in Table 4.5, and the corresponding confusion matrix is shown in Figure 4.17. This result provides a valuable comparison point for evaluating the effectiveness of our primary methods, which employ hierarchically engineered features integrated with Vision-Language Models.

Table 4.5: Performance Metrics of the Fine-Tuned CLIP Classifier (Raw Image Input) on Test Set

Class	Precision	Recall	F1-Score
Glioma	0.9516	0.8300	0.8866
Meningioma	0.8678	0.8508	0.8592
No Tumor	0.9336	0.9901	0.9611
Pituitary	0.8933	0.9767	0.9331
Accuracy			0.9096
Macro Avg	0.9116	0.9119	0.9100
Weighted Avg	0.9108	0.9096	0.9084

Note: Metrics based on the CLIP ViT-B/32 model fine-tuned directly on raw MRI images, using cosine similarity for classification.

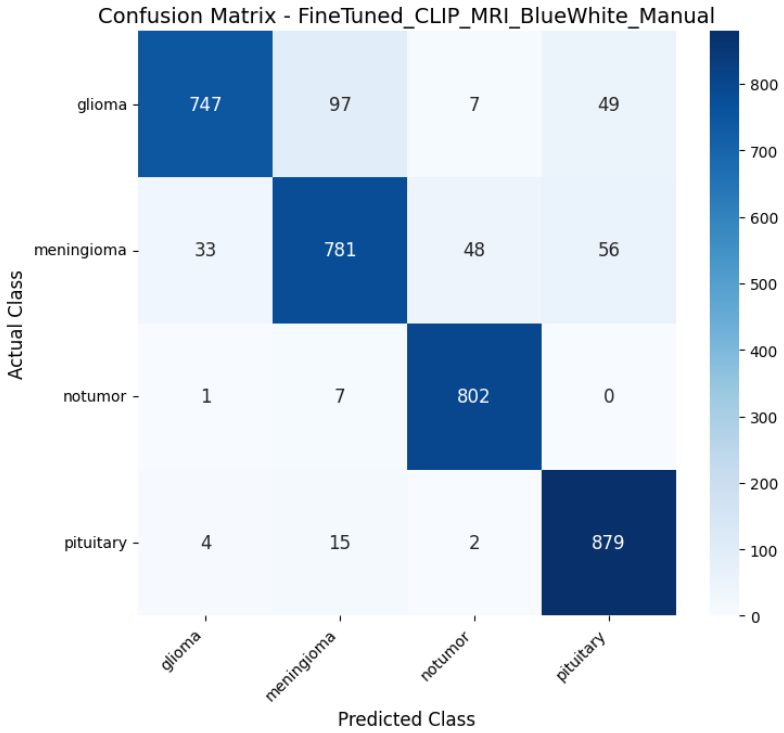


Figure 4.17: Confusion matrix for the fine-tuned CLIP model (ViT-B/32, raw image input, cosine similarity classification) on the test set.

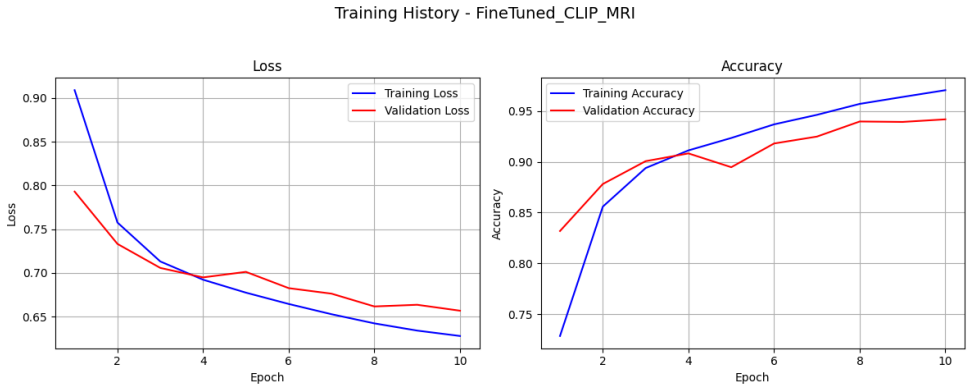


Figure 4.18: Training and validation loss and accuracy curves for the fine-tuned CLIP ViT-B/32 model over 10 epochs. The model was saved at the epoch with the best validation loss.

Direct Feature-Text Comparison (CLIP Zero-Shot)

This approach compared pre-computed 512D image features directly to fixed CLIP text embeddings (3.5.4). As shown in Table 4.6 and Figure 4.19, the zero-shot performance was poor (13.66% accuracy), indicating a significant misalignment between the derived image features and CLIP's native understanding without task-specific adaptation.

Table 4.6: Performance Metrics of the Direct CLIP Comparison (Zero-Shot) on the Test Set (using 512D features)

Class	Precision	Recall	F1-Score
Glioma	0.2850	0.0678	0.1095
Meningioma	0.3333	0.0054	0.0107
No Tumor	0.0308	0.0580	0.0402
Pituitary	0.2084	0.4100	0.2763
Accuracy			0.1366
Macro Avg	0.2144	0.1353	0.1092
Weighted Avg	0.2197	0.1366	0.1104

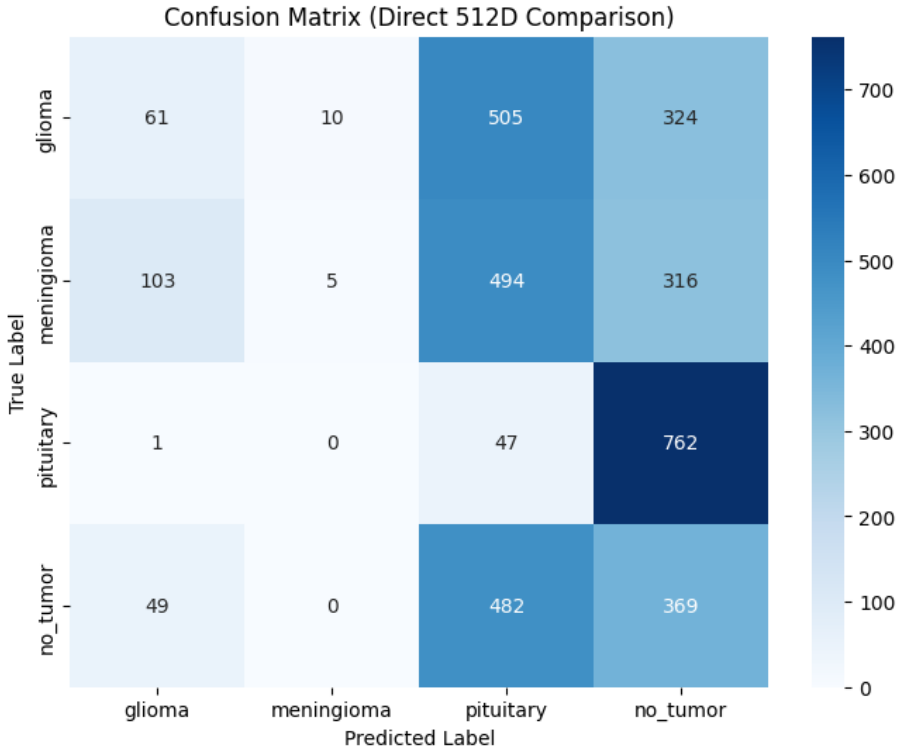


Figure 4.19: Confusion matrix for the Direct CLIP Comparison (Zero-Shot) on the test set.

Learned Feature Projection into CLIP Space (MLP Only)

This method involved training an MLP to project the 1024D features into CLIP space (3.5.5). This significantly improved performance over the zero-shot approach, achieving 95.72% accuracy. Results are shown in Table 4.7 and Figure 4.20.

Table 4.7: Performance Metrics of the Learned Projection Classifier (MLP Only) on the Test Set

Class	Precision	Recall	F1-Score
Glioma	0.99	0.89	0.94
Meningioma	0.90	0.96	0.93
No Tumor	0.97	0.98	0.98
Pituitary	0.98	1.00	0.99
Accuracy			0.96
Macro Avg	0.96	0.96	0.96
Weighted Avg	0.96	0.96	0.96

Note: Per-class metrics appear rounded in the source report.

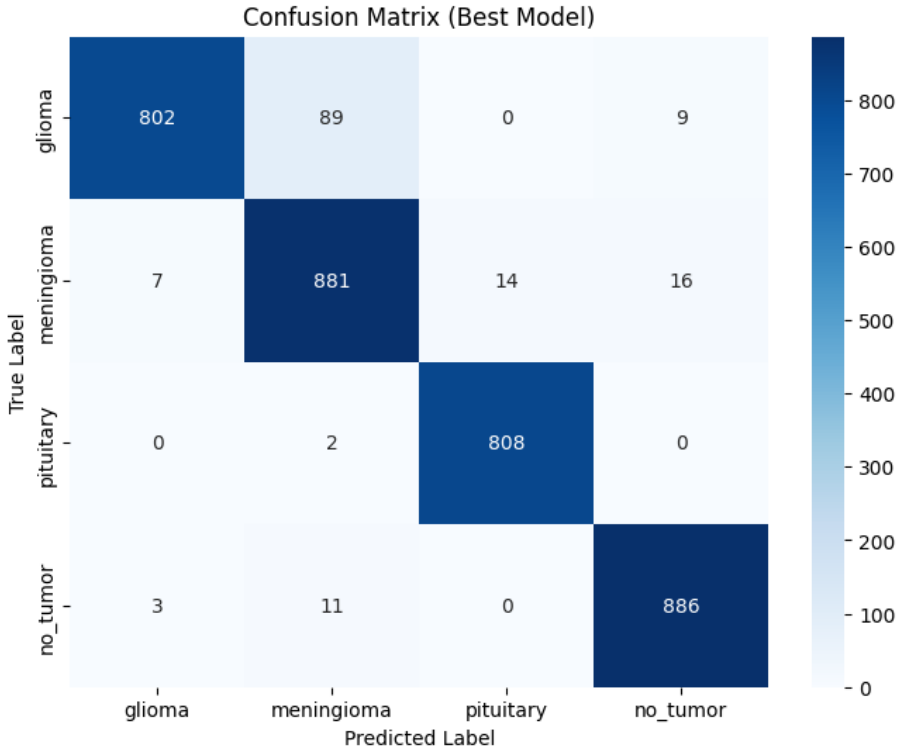


Figure 4.20: Confusion matrix for the Learned Feature Projection Classifier (MLP Only) on the test set.

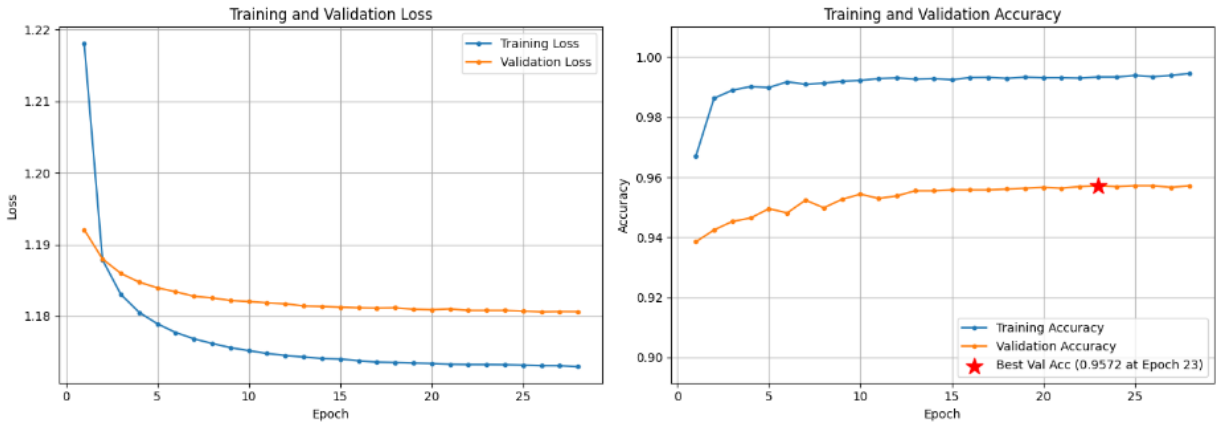


Figure 4.21: Training and validation history (loss and accuracy) for the **Learned Feature Projection model (MLP Only)**, as described in Section 3.5.5. This model projects the engineered 1024D features into CLIP's 512D embedding space for classification against fixed text embeddings.

Learned Feature Projection with CLIP Text Fine-tuning

This final approach combined the learned MLP projection with fine-tuning of CLIP's text encoder layers (3.5.6). This joint optimization yielded the best overall performance in this study, achieving 96.83% accuracy. Detailed results are presented in Table 4.8 and Figure 4.22.

Table 4.8: Performance Metrics of the Learned Projection Classifier with CLIP Text Fine-tuning on the Test Set

Class	Precision	Recall	F1-Score
Glioma	0.9801	0.9300	0.9544
Meningioma	0.9303	0.9597	0.9448
No Tumor	0.9780	0.9889	0.9834
Pituitary	0.9890	0.9975	0.9932
Accuracy			0.9683
Macro Avg	0.9694	0.9690	0.9690
Weighted Avg	0.9687	0.9683	0.9682

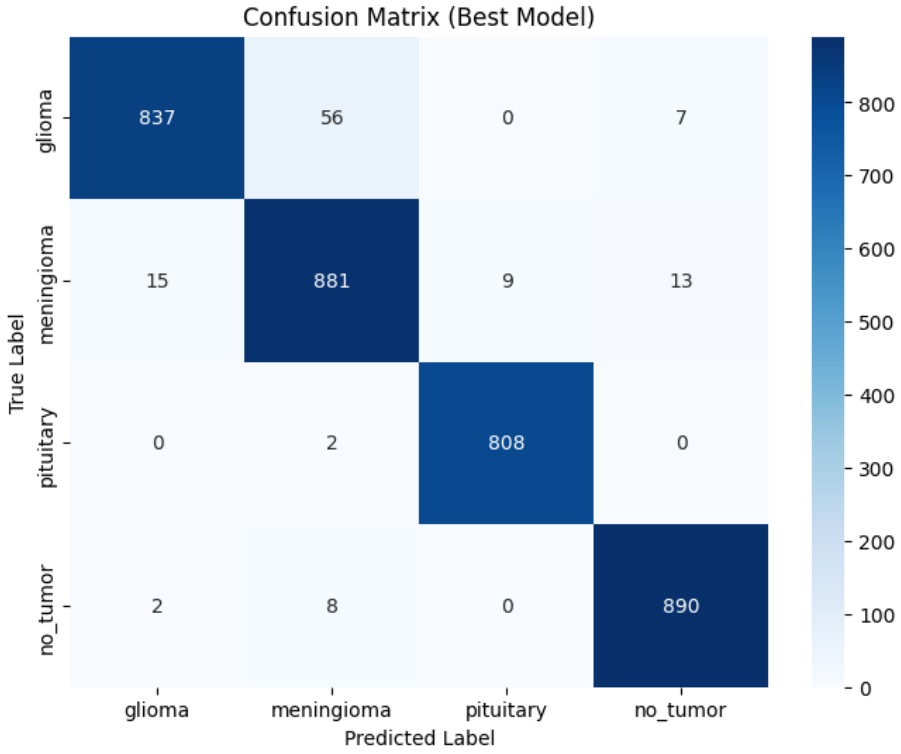


Figure 4.22: Confusion matrix for the Learned Feature Projection Classifier with CLIP Text Fine-tuning on the test set.

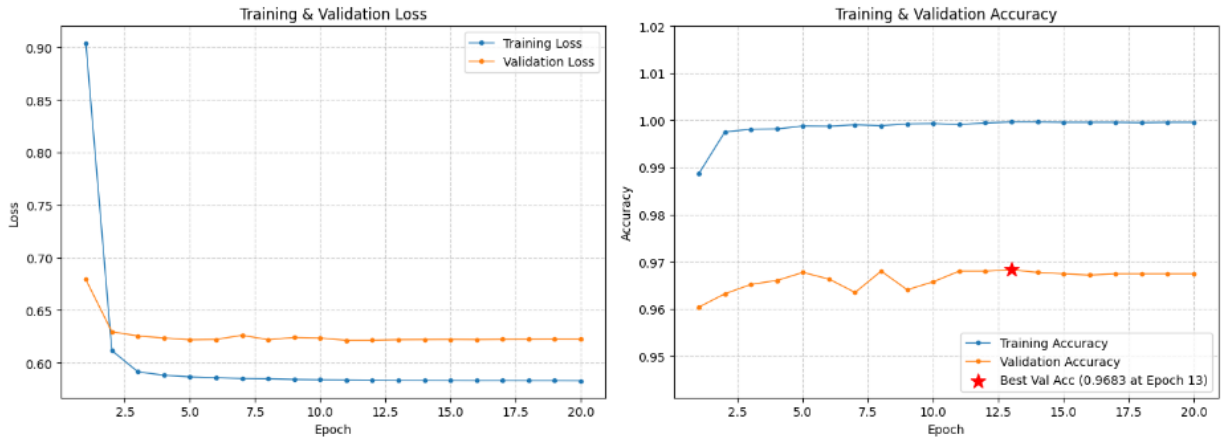


Figure 4.23: Training and validation history (accuracy and loss) for the Learned Feature Projection with CLIP Text Fine-tuning model (4.3.2).

Qualitative Analysis of CLIP-Based Prediction

Beyond aggregate metrics, the VLM approach utilizing learned projection and text fine-tuning (4.3.2) allows for inspection of individual predictions. Figure 4.24 illustrates this for a representative test sample (Index 927). The projected 1024D image feature is compared against the fine-tuned text embeddings for each class prompt via cosine similarity.

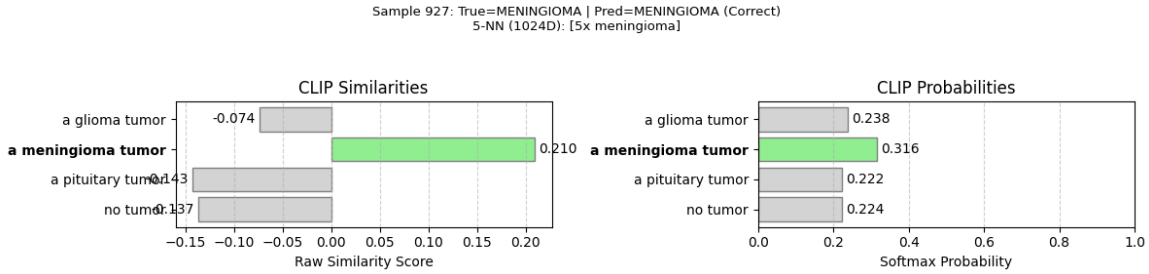


Figure 4.24: Example analysis of a single test sample (Index 927) using the Learned Projection + CLIP Text Fine-tuning model. The left plot shows the raw cosine similarity scores between the projected image feature and the fine-tuned text prompts for each class. The right plot shows the corresponding probabilities after applying a softmax function. The model correctly predicts 'meningioma' (highlighted green) based on the highest similarity/probability.

As shown in Figure 4.24, the model computes raw similarity scores (left panel) and converts these into probabilities (right panel). The class associated with the highest score/probability ('meningioma tumor' in this case, with a score of 0.228 and probability of 0.317) is selected as the prediction, which matches the true label for this sample. This type of output provides more granularity than a simple class prediction; the relative scores and probabilities can indicate the model's confidence and potentially highlight cases where the model finds similarities to multiple classes.

4.4 Comparative Analysis of Advanced Classifiers

The key performance metrics of the final successful advanced classification models are compared in Table 4.9 to provide a clear overview. This comparison highlights the performance differences between using LLMs (via quantization) and VLMs (via learned projection) on the engineered 1024D features, as well as the baseline using standard images using CLIP's image encoder.

Table 4.9: Comparative Performance Summary of Adapted AI Classification Techniques and Direct Feature Approaches on the Test Set

Method	Input Features	Test Accuracy
Adapted Models with Engineered 1024D Features:		
Learned Projection + CLIP Text Fine-tuning	1024D	**0.9683**
Learned Projection (MLP Only)	1024D	0.9572
Quantized LLM (Gemma 2B)	1024D (Quantized)	0.9430
Quantized LLM (RoBERTa Base)	1024D (Quantized)	0.9357
Benchmark Direct Fine-Tuning of CLIP on Raw Images:		
VLM (CLIP)	Images (Direct CLIP)	0.9096
Direct Feature Injection Attempts :		
LLM (Gemma 2B)	1024D	0.2602
VLM (CLIP)	512D (Derived)	0.1366

The results presented in Table 4.9 clearly demonstrate that the **Adapted Models utilizing Engineered 1024D Features**—whether through learned projection for Vision-Language Models (VLMs) like CLIP or feature quantization for Large Language Models (LLMs)—achieve significantly higher performance than other approaches. Specifically, methods like Learned Projection with CLIP Text Fine-tuning (96.83% accuracy), Learned Projection alone (95.72%), and Quantized LLMs (up to 94.30%) substantially outperform the benchmark fine-tuned CLIP model (90.96% using raw image input).

Furthermore, the stark underperformance of the **Direct Feature Injection Attempts**—namely the direct zero-shot CLIP comparison (13.66% accuracy, 4.6) and the attempted direct injection of 1024D features into an LLM (26.02% accuracy)—powerfully illustrates that 'off-the-shelf' application of these potent models is insufficient for this nuanced medical task. These outcomes highlight that it is the deliberate adaptation strategies, such as feature quantization for LLMs or learned feature projection combined with text encoder fine-tuning for VLMs, that are crucial for unlocking their classification prowess when applied to specialized, engineered visual features. The joint optimization of the projection MLP and CLIP's text encoder, representing the most sophisticated adaptation strategy explored, fittingly yielded the best overall accuracy, underscoring the value of tailored integration.

4.5 Chapter Summary

This chapter presented the quantitative and qualitative results of the brain tumor classification experiments. Visualizations confirmed the successful application of preprocessing and augmentation techniques (Figures 4.1, 4.2, 4.3). Performance metrics for the intermediate Stage

1 and Stage 2 models were reported (Tables 4.1, 4.2), establishing baseline feature quality and demonstrating the efficacy of the hierarchical feature engineering pipeline.

The core findings focused on the comparative performance of the Gen AI classifiers (4.3). The novel LLM quantization approach (using Gemma 2B and RoBERTa Base) yielded strong results (94.30% and 93.57% accuracy), demonstrating its feasibility for integrating LLMs with complex, pre-extracted medical imaging features. However, VLM-based approaches using learned projection of the 1024D features into CLIP space proved superior. While direct zero-shot comparison was ineffective (13.66% accuracy,4.6), learning an MLP projection significantly boosted performance (95.72% accuracy, Table 4.7). The combination of the projection MLP with CLIP text fine-tuning achieved the highest test accuracy of 96.83% (Table 4.8).

Crucially, all methods utilizing the engineered 1024D features substantially outperformed the benchmark approaches. Specifically, they surpassed the directly fine-tuned CLIP model operating on raw images (90.96% accuracy, Table 4.5) and the attempted direct zero-shot CLIP application (13.66% accuracy, 4.6). This underscores the value of the proposed hierarchical feature engineering pipeline. These findings provide a clear empirical basis for the conclusions drawn in Chapter 5.

4.5.1 Detailed Comparison with State-of-the-Art Ensemble Methods

The performance of the proposed Generative AI-based classification methods, particularly the Learned Projection with CLIP Text Fine-tuning approach, achieving 96.83% accuracy (Table 4.9), warrants a detailed comparison with recent state-of-the-art studies employing ensemble techniques on similar datasets. Two particularly relevant works are Shaikh et al. [35] and Aurna et al. [38].

Comparison with Shaikh et al. (2025)

Shaikh et al. [35] recently proposed an advanced "stacking ensemble learning (SEL)" methodology, termed SEL-DenseNet201, for enhancing brain tumor detection and segmentation. Of particular relevance to this thesis is their classification performance on a dataset they refer to as Dataset-I, which comprises 7023 MRI samples across four classes (glioma, meningioma, pituitary, and no tumor).

Their methodology is characterized by a two-level ensemble approach:

1. **Level-0 (Base Models):** Six diverse pre-trained Convolutional Neural Networks (CNNs) were employed as base learners. These included 3D-CNN, AlexNet, MobileNet-v3, VGG-16, VGG-19, and ResNet50. These models were individually trained on the brain tumor MRI data to capture distinct feature perspectives.

2. **Level-1 (Meta-Model):** The prediction outputs (interpreted as feature vectors) from all six base models were concatenated. This aggregated feature set was fed into a DenseNet201 architecture, which served as the meta-learner (SEL-DenseNet201). The DenseNet201

was trained to make the final classification based on the combined insights from the base models.

3. **Training Strategy:** The authors report training the SEL-DenseNet201 meta-model from scratch. The base models, being pre-trained on ImageNet, were likely fine-tuned on the brain tumor dataset to adapt their learned features for the specific medical imaging task, a common practice in transfer learning.

4. **Data Augmentation:** Shaikh et al. applied data augmentation techniques, including rotation, flipping, random cropping, brightness adjustments, and Gaussian noise to enhance dataset diversity and model robustness.

On their Dataset-I (7023 samples, four classes), Shaikh et al. reported an exceptionally high average accuracy of **99.65%** [35] for their SEL-DenseNet201 model.

Comparing this outstanding result with our best-performing model (Learned Projection with CLIP Text Fine-tuning), which achieved 96.83% accuracy, several critical distinctions in approach and objectives emerge:

- **Core Architectural Paradigm:** Shaikh et al. leverage the established and potent paradigm of stacking ensemble learning. By combining multiple, diverse CNNs, they mitigate individual model biases and exploit complementary strengths. The DenseNet201 meta-learner adeptly learns to weigh and combine these diverse inputs. This meticulous construction of an ensemble is a well-known strategy for pushing performance boundaries. Our thesis takes a different approach, focusing on pioneering and validating methodologies for integrating contemporary Generative AI models (LLMs and VLMs) with engineered visual features. The novelty lies in bridging the gap between sophisticated, pre-extracted image representations and the unique processing capabilities of these large-scale AI architectures.

- **Extent of Model Training and Fine-Tuning:** While Shaikh et al. state their meta-model was trained from scratch, the overall system benefits from the (presumed) fine-tuning of six substantial base CNNs plus the training of a large DenseNet201 meta-learner. This represents a considerable optimization effort across the entire ensemble. Our approach deliberately explored the efficacy of leveraging powerful, pre-trained Gen AI models with *minimal and targeted* fine-tuning. For instance, the base CNNs in our feature engineering pipeline (Stage 1) had essentially frozen weights, with only small classification heads being trained. Similarly, the LLM backbones were kept entirely frozen, and for our best VLM (CLIP) method, only the projection MLP and the final layers of the CLIP text encoder were made trainable. This strategy was intended to assess the adaptability of inherent Gen AI capabilities rather than undertaking exhaustive end-to-end fine-tuning of these massive models.

- **Feature Engineering and Representation:** Shaikh et al.'s meta-learner operates on features essentially the predictive outputs or high-level representations from the base CNNs.

The ensemble itself performs an implicit, learned feature fusion. Our methodology involved an explicit, hierarchical feature engineering process (Stages 1 and 2) to create a consolidated 1024-dimensional feature vector *prior* to its introduction to the Gen AI models. This was designed to provide a rich, multi-perspective, yet condensed visual summary tailored for these advanced AI systems.

- **Reinforcement of Foundational Concepts:** The remarkable success of Shaikh et al.'s multi-CNN ensemble strongly corroborates a foundational principle of our research: the significant benefit derived from integrating diverse feature perspectives. Our Stage 1 and Stage 2 feature engineering, which combines EfficientNetB0, InceptionV3, and Xception outputs, is built on this idea. Their results demonstrate the high level of performance achievable when such feature diversity is optimally exploited by a robust learning algorithm. This suggests that further performance gains in our framework are plausible with more extensive fine-tuning of our Gen AI components or even more elaborate feature engineering preceding the Gen AI stage.

- **Divergent Primary Contributions:** Shaikh et al.'s primary contribution is demonstrating state-of-the-art classification accuracy on this dataset through meticulous application and tuning of a stacking ensemble of well-established CNNs. Our thesis, on the other hand, contributes novel and validated strategies for integrating engineered features with LLMs (via feature quantization) and VLMs (via learned projection and targeted fine-tuning). We demonstrate that these Gen AI models, when appropriately adapted, can achieve high performance (e.g., 96.83% with our adapted CLIP model, significantly outperforming the 90.96% from direct CLIP fine-tuning on raw images, Table 4.5), offering a distinct and promising avenue for leveraging the semantic and contextual understanding of large pre-trained multimodal models in specialized fields like medical imaging.

In summary, while Shaikh et al. [35] achieve a higher classification accuracy through an expertly constructed ensemble of CNNs with extensive training, our research charts a different but equally valuable course. We establish the viability and effectiveness of integrating advanced Generative AI models with domain-specific engineered features, particularly highlighting the necessity of tailored adaptation strategies rather than off-the-shelf deployment. The success of ensemble methods like that of Shaikh et al. underscores the power of feature diversity, a principle also central to our feature engineering pipeline, and suggests that future work combining the strengths of advanced feature engineering with more comprehensively trained Gen AI models could yield further breakthroughs.

Comparison with Aurna et al. (2022)

Aurna et al. [38] presented a sophisticated "two-stage feature level ensemble" approach for brain tumor classification, also including the 4-class problem (glioma, meningioma, pituitary, no tumor). Critically, they utilized multiple datasets, including the Figshare dataset [43] and others,

creating a "Merged Dataset" that closely resembles the composition of the Kaggle dataset [41] used in our study. Therefore, their results on this Merged Dataset provide a highly relevant benchmark.

Their methodology involved:

1. Selecting the best individual CNN models (from a pool including EfficientNet-B0, ResNet-50, and a proposed custom CNN) based on initial performance.
2. Creating first-stage ensembles by concatenating features from pairs of these selected models (e.g., EfficientNet-B0 + ResNet-50).
3. Selecting the best two first-stage ensemble models.
4. Creating a final two-stage ensemble by concatenating the features from these two best first-stage models.
5. Applying Principal Component Analysis (PCA) for dimensionality reduction on the final concatenated features.
6. Classifying the PCA-reduced features using a Softmax classifier.

On their Merged Dataset, Aurna et al. reported a remarkable average accuracy of **98.96%** [38].

Comparing this with our best accuracy of 96.83%, several factors must be considered:

- **Feature Engineering Complexity:** Both studies employ advanced feature engineering. Aruna et al.'s two-stage concatenation potentially captures an even wider array of complementary features than our Stage 2 fusion, although our approach benefits from the distinct architectures of EfficientNet, Inception, and Xception. Their use of PCA also explicitly aims to reduce redundancy and noise in the high-dimensional concatenated feature space, which might contribute significantly to performance.
- **Classification Method:** While their feature engineering is complex, Aurna et al. ultimately rely on a standard Softmax classifier applied to the final PCA-reduced features. Our approach uses the more novel VLM integration. The fact that their intricate feature engineering paired with Softmax achieved higher accuracy suggests that, for this specific dataset and feature configuration, optimizing the feature representation itself (through multi-stage ensembling and PCA) might have been more impactful than the choice of a more advanced classifier like our adapted CLIP model.
- **Data Augmentation/Preprocessing:** While both studies likely used similar base datasets and standard augmentation, subtle differences in implementation could exist, potentially influencing the final performance. Aruna et al. mention specific augmentation techniques (flipping, rotation, zoom, shift, scaling) similar to ours.

The results from Aurna et al. [38] demonstrate that meticulous, multi-stage feature engineering combined with dimensionality reduction can achieve state-of-the-art performance on this dataset using conventional classifiers. While our VLM-based approach (96.83%) did not surpass their accuracy on the merged dataset, it significantly outperformed baseline benchmarks, including the directly fine-tuned CLIP model (90.96%, Table 4.5). Our contribution lies in demonstrating the viability and effectiveness of integrating engineered features with VLMs/LLMs, particularly the success of learned projection and text fine-tuning for CLIP, offering a distinct pathway for leveraging large pre-trained multimodal models in medical imaging. The comparison highlights that sophisticated feature engineering and advanced classifier integration are valuable avenues, and the optimal combination may be task- and dataset-dependent.

Chapter 5

Conclusion

This thesis presented a comprehensive investigation into developing high-performance brain tumor classification systems using advanced deep-learning methodologies on MRI data. A robust foundational pipeline involving preprocessing, augmentation, and hierarchical CNN-based feature extraction yielded high-quality 1024-dimensional representations (Stage 2). This deliberate, multi-stage feature engineering approach consolidates information from diverse CNN perspectives, was not merely a preliminary step but a cornerstone of the proposed methodology. It formed a critical and novel basis, providing a concentrated and highly informative feature set designed to empower the subsequent advanced classification experiments with LLMs and VLMs. This distinguishes the work from methods relying on single-source features or attempting end-to-end application of large models on limited medical data without such tailored feature conditioning.

Building on these rich 1024D features, several Gen AI classification techniques were implemented and critically compared. Integrating Large Language Models (LLMs) via feature quantization proved successful. Transforming continuous features into discrete sequences enabled the use of frozen LLM backbones (Gemma 2B: **94.30%** test accuracy; RoBERTa Base: **93.57%** test accuracy), validating quantization as a viable strategy for applying LLMs to complex medical feature data and revealing subtle performance differences between LLM architectures.

Exploration of Vision-Language Models (VLMs) using the engineered features yielded varied outcomes depending on the integration strategy. A direct zero-shot comparison using specially prepared 512D image features (derived from the 1024D features) and fixed CLIP text embeddings (3.5.4) performed poorly (**13.66%** accuracy, 4.6). This suggests that even derived features optimized for the target embedding dimension may not inherently align with CLIP's general semantic space without further task-specific adaptation.

In contrast, strategies involving learned adaptation between the engineered 1024D features and CLIP's space were highly effective. Training a dedicated MLP projection layer to map the 1024D features into CLIP's 512D space while keeping CLIP's text embeddings fixed (3.5.5) achieved strong performance (**95.72%** accuracy). This demonstrated the value of learning an explicit bridge between the distinct feature spaces.

The most successful approach further enhanced this by jointly optimizing the projection MLP and fine-tuning the final layers of the CLIP text encoder (3.5.6). This combined adaptation, concurrently refining the image feature projection and the text embedding semantics, achieved the highest accuracy in this study **96.83%**.

Crucially, all methods leveraging the engineered 1024D features and appropriate integration techniques (quantization or projection) significantly outperformed benchmark approaches, such as directly fine-tuning the CLIP model on raw images (90.96% accuracy). This highlights the substantial benefit derived from the initial hierarchical feature engineering stages compared to more direct CLIP applications or off-the-shelf CLIP features.

In conclusion, this thesis proposes a novel framework for leveraging pre-trained Generative AI models in medical image classification by integrating engineered CNN features with LLMs and VLMs. We demonstrate that we can achieve high diagnostic accuracy without extensive model retraining through feature quantization, embedding projection, and minimal model adaptation. Despite using fewer parameters and limited fine-tuning, our best model outperforms existing baselines and rivals ensemble CNNs.

Future work could focus on systematic hyperparameter optimization, extending the quantization framework to richer token vocabularies, and evaluating the pipeline on other medical domains such as CT scans or fundus images. Additionally, further development of interpretability tools — such as prompt sensitivity analysis or attention-based visual explanations — could enhance trust in real-world clinical deployment. This work lays the foundation for efficient, scalable, and explainable Gen AI-based diagnostic tools.

5.0.1 Future Work

The research presented in this thesis provides a robust foundation for leveraging Generative AI in brain tumor classification from MRI data. However, several exciting avenues remain for further exploration to enhance performance, interpretability, and clinical applicability. Key directions include:

Exploration of Advanced Generative AI Architectures

A primary area for future investigation involves harnessing the capabilities of larger, more recent, or specialized Generative AI models.

- **Vision-Language Models (VLMs):** Experimenting with newer VLM architectures beyond the foundational CLIP model, such as LLaVA, or exploring updated CLIP variants, could yield performance gains. Furthermore, developing and utilizing VLMs specifically pre-trained or fine-tuned on large-scale medical vision-language datasets would be highly beneficial. Such domain-specific models are anticipated to possess a more refined understanding of medical visual concepts and terminology, potentially leading to more accurate and nuanced classifications when integrated with the engineered features.

- **Large Language Models (LLMs):** For the quantized feature classification approach, investigating more advanced LLMs (e.g., newer iterations of Gemma, models from the Llama family, or Mixtral) could improve the model's ability to discern complex patterns within the "feature language." With their increased parameter counts and refined architectures, larger models often demonstrate superior contextual understanding and reasoning capabilities.

The rationale behind this exploration is that larger models can often capture more subtle and intricate patterns, while domain-specific pre-training can provide a crucial inductive bias, improving alignment with the specific characteristics of medical imaging data.

Integration and Fusion of Multi-Modal MRI Data

The current framework primarily processes MRI data from (implied) a single sequence. A significant extension would be to adapt the methodology to incorporate and fuse information from multiple MRI sequences (e.g., T1-weighted, T1-weighted contrast-enhanced, T2-weighted, FLAIR).

- **Adapting Feature Extraction:** The initial hierarchical CNN feature extraction pipeline (Stage 1 and Stage 2) would need to be modified to process features from each modality, perhaps with shared or separate branches, followed by an effective fusion strategy before input to the final GenAI classifier.

- **Benefits:** Different MRI sequences highlight tumors' distinct tissue properties and pathological characteristics. A multi-modal approach is expected to provide a more comprehensive and robust representation of the underlying anatomy and pathology, potentially leading to improved diagnostic accuracy, especially for challenging or ambiguous cases that may be difficult to resolve with a single sequence.

Advanced Feature Engineering and Refinement for GenAI Input

While the proposed hierarchical feature engineering pipeline (Stage 1 and Stage 2) effectively generated informative 1024-dimensional features, further enhancements could be explored.

- **Sophisticated Fusion Techniques:** Drawing inspiration from state-of-the-art ensemble methods, such as the multi-stage feature-level ensembling demonstrated by Aurna et al. [38], more complex fusion strategies could be implemented **before** the features are passed to the GenAI classifiers. This could involve, for instance, learning attention mechanisms to dynamically weigh the contributions of features extracted from the different parallel CNNs in Stage 1, thereby creating an even more discriminative and refined input representation.

- **Refinement of Feature Quantization for LLMs:** The K-Means-based quantization strategy successfully enabled using LLMs. Future work could involve a more systematic investigation into the quantization process itself. This includes exploring different clustering

algorithms, varying the number of clusters (i.e., the vocabulary size of the ”feature language”), and experimenting with quantifying the dimensionality of the sub-vectors. Fine-tuning these quantization parameters could potentially lead to a more optimal discrete representation of the visual features, thereby enhancing the performance of the subsequent LLM-based classifiers.

These refinements aim to further optimize the input provided to the advanced GenAI models, as their performance is critically dependent on the quality and representational power of the features they operate on.

These future directions, building upon the methodologies and findings of this thesis, hold considerable promise for advancing the capabilities of AI-driven brain tumor diagnosis.

Bibliography

[1] World Health Organization, "Brain, central nervous system - estimated incidence, mortality and prevalence worldwide in 2022," https://gco.iarc.who.int/media/globocan/factsheets/cancers/31-brain-central-nervous-system-fact-sheet.pdf, 2022, accessed: May 2025.

[2] DR vijay anand reddy director, apollo cancer centers, "How important is early detection in treating brain tumors?" https://drvijayanandreddy.com/how-important-is-early-detection-in-treating-brain-tumors/, accessed: May 2025.

[3] R. K. Loo and T. Bucho, "Bidimensional measurements in brain tumors: Assessment of interobserver variability," *American Journal of Roentgenology*, vol. 193, pp. W515–W522, 2009.

[4] S. Khalighi, K. Reddy, A. Midya, and et al., "Artificial intelligence in neuro-oncology: advances and challenges in brain tumor diagnosis, prognosis, and precision treatment," *npj Precision Oncology*, vol. 8, p. 80, 2024.

[5] Aaron Cohen-Gadol, MD, "Brain tumor statistics," https://www.aaroncohen-gadol.com/en/patients/brain-tumor/types/statistics, accessed: May 2025.

[6] H. Ahmed, D. Michael, and B. Samaila, "Current challenges of the state-of-the-art of AI techniques for diagnosing brain tumor," *Material Science & Engineering*, vol. 7, pp. 196–208, 2023.

[7] Mayo Clinic, "Brain tumor - symptoms and causes," https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084, accessed: May 2025.

[8] D. N. Louis, A. Perry, P. Wesseling, D. J. Brat, I. A. Cree, D. Figarella-Branger, C. Hawkins, H. K. Ng, S. M. Pfister, G. Reifenberger, R. Scollary, D. W. Ellison, and WHO Classification of Tumours Editorial Board, "The 2021 WHO classification of tumors of the central nervous system: a summary," *Neuro-Oncology*, vol. 23, no. 8, pp. 1231–1251, 2021.

[9] Q. T. Ostrom, M. Price, C. Neff, G. Cioffi, K. Waite, C. Kruchko, and J. S. Barnholtz-Sloan, "CBTRUS statistical report: Primary brain and other central nervous system tumors diag-

nosed in the United States in 2016–2020," *Neuro-Oncology*, vol. 25, no. Supplement_4, pp. iv1–iv99, 2023.

[10] The Cancer Imaging Archive (TCIA), "UCSF-PDGM," https://www.cancerimagingarchive.net/collection/ucsf-pdgm/, accessed: May 2025.

[11] W. E. Brant and C. A. Helms, *Fundamentals of Diagnostic Radiology*, 4th ed. Lippincott Williams & Wilkins, 2012.

[12] D. D. Stark and W. G. Bradley Jr., *Magnetic Resonance Imaging*, 3rd ed. Mosby, 1999.

[13] P. Modi, "Convolutional neural networks for dummies: A step-by-step cnn tutorial," Dec 2023. [Online]. Available: https://medium.com/@prathammodi001/convolutional-neural-networks-for-dummies-a-step-by-step-cnn-tutorial-e68f464d608f

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[16] ——, "Rethinking the Inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[17] V. Chauhan and N. R. Killi, "A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks," *International Journal of Advanced Manufacturing Technology*, vol. 115, pp. 115–128, Mar 2021, accessed: May 2025. [Online]. Available: https://www.researchgate.net/publication/350319854_A_Machine_Learning_Method_for_Defect_Detection_and_Visualization_in_Selective_Laser_Sintering_based_on_Convolutional_Neural_Networks

[18] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807.

[19] D. Das, T. Ashik, M. M. Islam, M. J. Hossain, and M. Hasan, "Optimized crop disease identification in bangladesh: A deep learning and svm hybrid model for rice, potato, and corn," Jul 2024, accessed: May 2025. [Online]. Available: https://www.researchgate.net/publication/382660620_Optimized_Crop_Disease_Identification_in_Bangladesh_A_Deep_Learning_and_SVM_Hybrid_Model_for_Rice_Potato_and_Corn

[20] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*,

ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 6105–6114.

[21] Javier Canales Luna, "What is an LLM? A guide on large language models and how they work," https://shorturl.at/LgKzT, accessed: May 2025.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017, pp. 5998–6008.

[23] Gemma Team and et al., "Gemma: Open models based on Gemini research and technology," *arXiv preprint arXiv:2403.08295*, 2024.

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 8748–8763.

[26] A. Al-Zoghby, E. Al-Awadly, A. Moawad, N. Yehia, and A. Ebada, "Dual deep CNN for tumor brain classification," *Diagnostics*, vol. 13, p. 2127, 2023.

[27] N. Huda and K. Ku-Mahamud, "CNN-based image segmentation approach in brain tumor classification: A review," *Engineering Proceedings*, vol. 84, p. 66, 2025.

[28] A. Batool and Y.-C. Byun, "A lightweight multi-path convolutional neural network architecture using optimal features selection for multiclass classification of brain tumor using magnetic resonance images," *Results in Engineering*, vol. 25, p. 104327, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590123025004086

[29] S. Mugdha and M. Uddin, "NeuroSight: A deep-learning integrated efficient approach to brain tumor detection," *Engineering Reports*, vol. 7, p. e13100, 2025.

[30] D. Rastogi, P. Johri, M. Donelli, L. Kumar, S. Bindewari, A. Raghav, and S. Khatri, "Brain tumor detection and prediction in MRI images utilizing a fine-tuned transfer learning model integrated within deep learning frameworks," *Life*, vol. 15, p. 327, 2025.

[31] W. Ahmad, H. Ali, Z. Shah, and S. Azmat, "A new generative adversarial network for medical images super resolution," *Scientific Reports*, vol. 12, no. 1, p. 9533, 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-13658-4

[32] D. Mukherkjee, P. Saha, D. Kaplun, A. Sinitca, and R. Sarkar, "Brain tumor image generation using an aggregation of gan models with style transfer," *Scientific Reports*, vol. 12, no. 1, p. 9141, 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-12646-y

[33] J. E. Park, P. Vollmuth, N. Kim, and H. S. Kim, "Research highlight: Use of generative images created with artificial intelligence for brain tumor imaging," *Korean Journal of Radiology*, vol. 23, no. 5, pp. 500–504, May 2022, epub 2022 Apr 4. [Online]. Available: https://doi.org/10.3348/kjr.2022.0033

[34] X. Xia, X. Pan, N. Li, X. He, L. Ma, X. Zhang, and N. Ding, "Gan-based anomaly detection: A review," *Neurocomputing*, vol. 493, pp. 497–535, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221019482

[35] A. Shaikh, S. Amin, M. A. Zeb, A. Sulaiman, M. S. Al Reshan, and H. Alshahrani, "Enhanced brain tumor detection and segmentation using densely connected convolutional networks with stacking ensemble learning," *Computers in Biology and Medicine*, vol. 186, p. 109703, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482525000538

[36] K. Hosny, M. Mohammed, R. Salama, and et al., "Explainable ensemble deep learning-based model for brain tumor detection and classification," *Neural Computing and Applications*, vol. 37, pp. 1289–1306, 2025.

[37] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, and M. O. Alassafi, "Brain tumor classification based on fine-tuned models and the ensemble method," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3967–3982, 2021. [Online]. Available: https://doi.org/10.32604/cmc.2021.014158

[38] N. F. Aurna, M. A. Yousuf, K. A. Taher, A. Azad, and M. A. Moni, "A classification of mri brain tumor based on two stage feature level ensemble of deep cnn models," *Computers in Biology and Medicine*, vol. 146, p. 105539, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482522003316

[39] S. Tehsin, I. M. Nasir, and R. Damaševičius, "Gatransformer: A graph attention network-based transformer model to generate explainable attentions for brain tumor detection," *Algorithms*, vol. 18, no. 2, 2025. [Online]. Available: https://www.mdpi.com/1999-4893/18/2/89

[40] L. Annet Abraham, G. Palanisamy, and V. Goutham, "Dilated convolution and yolov8 feature extraction network: An improved method for mri-based brain tumor detection," *IEEE Access*, vol. 13, pp. 27 238–27 256, 2025.

[41] M. Nickparvar, "Brain tumor MRI dataset," Kaggle Dataset, 2020, available: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.

[42] W. A. A. Wanabilini, "Brain tumor detection using image processing," Medium, 2022, available: https://medium.com/wanabilini/brain-tumor-detection-using-image-processing-a26b1c927d5d.

[43] J. Cheng, "Brain tumor dataset. Figshare MRI dataset version 5," 2017, dataset.