mathematics

MDPI

Article

Hierarchical Deep Feature Fusion and Ensemble Learning for Enhanced Brain Tumor MRI Classification

Zahid Ullah and Jihie Kim *

Department of Computer Science and Artificial Intelligence, Dongguk University, Seoul 04620, Republic of Korea; zahid1989@dongguk.edu

* Correspondence: jihie.kim@dgu.edu; Tel.: +82-2-2260-4973

Abstract

Accurate classification of brain tumors in medical imaging is crucial to ensure reliable diagnoses and effective treatment planning. This study introduces a novel double ensemble framework that synergistically combines pre-trained Deep Learning (DL) models for feature extraction with optimized Machine Learning (ML) classifiers for robust classification. The framework incorporates comprehensive preprocessing and data augmentation of brain Magnetic Resonance Images (MRIs), followed by deep feature extraction based on transfer learning using pre-trained Vision Transformer (ViT) networks. The novelty of our approach lies in its dual-level ensemble strategy: we employ a feature-level ensemble, which integrates deep features from the top-performing ViT models, and a classifier-level ensemble, which aggregates predictions from various hyperparameter-optimized ML classifiers. Experiments on two public MRI brain tumor datasets from Kaggle demonstrate that this approach significantly surpasses state-of-the-art methods, underscoring the importance of feature and classifier fusion. The proposed methodology also highlights the critical roles that hyperparameter optimization and advanced preprocessing techniques can play in improving the diagnostic accuracy and reliability of medical image analysis, advancing the integration of DL and ML in this vital, clinically relevant task.

Keywords: brain tumor classification; Deep Learning; Machine Learning; ensemble learning; transfer learning

MSC: 68T07; 37-04

check for updates

Academic Editor: Jonathan Blackledge

Received: 27 July 2025
Revised: 21 August 2025
Accepted: 22 August 2025
Published: 29 August 2025

Citation: Ullah, Z.; Kim, J. Hierarchical Deep Feature Fusion and Ensemble Learning for Enhanced Brain Tumor MRI Classification. *Mathematics* **2025**, *13*, 2787. https://doi.org/10.3390/math13172787

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Magnetic Resonance Imaging (MRI) is a cornerstone of modern medical diagnostics. It is known for its exceptional ability to non-invasively visualize complex anatomical structures with high-resolution, multi-planar imaging so pathological conditions can be identified and treated [1]. This capability makes MRI invaluable for detecting and characterizing a wide range of diseases; particularly brain tumors, where early detection is often a matter of life and death [2]. MRI's sensitivity to subtle variations in tissue composition and physiological processes allows for precise differentiation between normal and abnormal tissues. By leveraging the unique magnetic properties of various tissue types, MRI provides detailed insights into the size, shape, and location of tumors, facilitating accurate diagnosis and treatment planning [3]. Even after initial detection, MRI continues to play a critical role in monitoring treatment responses and identifying tumor recurrence, directly influencing patient outcomes [4]. Despite these strengths, the interpretation of MRI images often relies

on the expertise of human specialists, introducing subjectivity and potential for error when analyzing the images produced [5]. This dependence underscores the need for advanced analytical tools to enhance diagnostic precision and consistency, addressing the challenges of human variability in image interpretation.

Accurate and reliable classification of brain tumors is essential for effective treatment planning and improved patient outcomes, yet it remains a significant challenge in medical image analysis [6,7]. This complexity stems from the substantial heterogeneity in tumor morphology, texture, and contrast; in fact, tumors vary not only between patients but also within different regions of the same tumor [8]. Traditional ML techniques, which rely heavily on handcrafted features, often fail to capture this complexity, limiting their generalizability and robustness [3]. In contrast, Deep Learning (DL) methodologies, particularly Convolutional Neural Networks (CNNs), have demonstrated remarkable success in automatically extracting intricate features directly from medical images, achieving State-Of-The-Art (SOTA) performance in brain tumor classification [9]. However, DL models typically require large, accurately labeled datasets for effective training, a challenge in the medical domain due to the time-intensive and costly nature of expert annotation. Brain tumors significantly impact patients' lifespan and quality of life, making early and accurate diagnosis critical. ML and image processing techniques have the potential to automate diagnostic workflows, improving both accuracy and reliability [10]. Furthermore, DL has revolutionized healthcare by enhancing recognition, prediction, and diagnosis across various medical domains, including brain tumors, lung cancer, cardiovascular conditions, and retinal diseases [11]. This progress underscores the transformative potential of advanced AI methodologies in addressing complex medical imaging challenges.

Classifying brain tumors from MRIs is a challenging task due to the heterogeneity of tumor types and subtle variations in their imaging characteristics. ML models, such as Support Vector Machines (SVMs), Random Forests (RFs), and Deep Neural Networks (DNNs), have been employed for this purpose, however, these models have achieved quite varying levels of success depending on the quality of the extracted features and the classification algorithm used [12–14]. Pre-training techniques are particularly valuable for enhancing the performance of Vision Transformers (ViTs), especially when working with limited medical image datasets [15]. Transfer learning, which involves fine-tuning a model pre-trained on a large general dataset for a more specific task, has been shown to improve generalization and reduce overfitting in medical imaging applications [16]. Recent research on ViTs for brain tumor classification has reported promising results, highlighting their ability to capture long-range dependencies and extract meaningful features with high accuracy and robustness [17]. These attributes make ViTs well-suited for tackling the complexities of brain tumor classification.

To overcome the limitations of traditional ML or purely DL approaches, we propose a new hybrid methodology that combines the strengths of both paradigms. ViTs models, pre-trained on large datasets, are used as robust feature extractors to capture salient and discriminative attributes from brain MRI scans. ViTs offer a compelling alternative to CNNs for feature extraction, particularly in medical imaging [18]. By leveraging self-attention mechanisms, ViTs more effectively capture global relationships within an image, identifying intricate dependencies that may be overlooked by traditional convolutional methods [15]. Unlike CNNs, which rely on localized receptive fields, ViTs process images as sequences of patches, transforming image analysis into a sequence-to-sequence learning problem [19]. This approach enables the model to learn contextual dependencies across the entire image, fostering a holistic understanding of inter-regional relationships.

ViTs have demonstrated promising results in various computer vision tasks and are particularly valuable in medical image analysis, especially when labeled data is limited.

The features extracted by these pre-trained models are subsequently fed into ML classifiers, (e.g., SVM or RF); to distinguish between tumorous and normal MRI scans. This hybrid strategy capitalizes on the representational power of pre-trained models while maintaining the flexibility and efficiency of ML classifiers [20].

By providing effective tools for analyzing high-dimensional and multimodal data, the integration of DL and ML methodologies has become increasingly vital in biomedical sciences [21]. These techniques have been extensively applied in brain image analysis, aiding the development of diagnostic and classification systems for various conditions including strokes, psychiatric disorders, epilepsy, neurodegenerative diseases, and demyelinating disorders [22]. Furthermore, ML and computer vision techniques have the potential to revolutionize radiology workflows by automating the prioritization of imaging studies, such as those involving suspected intracranial hemorrhage, thereby enhancing diagnostic efficiency and accuracy.

MRI is essential for the detection and characterization of brain tumors, but manual interpretation of the resulting images remains subject to variability and time constraints. Multi-reader studies report only moderate inter-rater agreement for glioma grade predictions (e.g., $\kappa \approx 0.56$ in recent multi-center evaluations), illustrating substantial diagnostic variability across clinicians [23]. Radiology workflow studies indicate that experienced neuroradiologists spend on average around 10–12 min interpreting a brain MRI, with longer times for trainees, which contributes both to reporting delays and potential fatigue-related errors in high-volume clinics [24]. Emerging evidence shows that AI assistance can materially reduce reading times (for example, reductions of 2.8 min were observed in an AI-assisted MRI reading study) and may streamline workflows while preserving or improving diagnostic consistency [25]. At the same time, contemporary deep-learning systems underscore their technical promise by demonstrating high accuracy on curated public benchmarks; however, many published models still show limited generalization across multi-center datasets and acquisition protocols [26,27]. Taken together, these clinical and technical realities motivate our dual-ensemble framework: by combining complementary DL with classical Machine Learning (ML) classifiers, we aim to (i) reduce inter-observer variability, (ii) improve robustness across diverse data sources, and (iii) accelerate reliable MRI triage in routine practice.

To harness the complementary strengths of diverse pre-trained DL architectures, we propose a novel feature ensemble approach for brain tumor classification. Our methodology includes multiple variations: a base version, versions incorporating normalization and Principal Component Analysis (PCA) (i.e., PCA has been used to reduce the dimensionality of the extracted deep features, retaining the most informative components), versions using the Synthetic Minority Over-sampling TEchnique (SMOTE) i.e., to addresses class imbalance in the datasets by generating synthetic samples for underrepresented tumor classes), and combinations of normalization, PCA, and SMOTE. The core of this approach lies in an innovative feature evaluation and selection mechanism, where deep features extracted from thirteen pre-trained DL models are rigorously assessed using nine distinct ML classifiers. A custom-designed criterion ensures the retention of only the most salient and discriminative features for subsequent analysis. The selected features are combined into a synthetic feature vector by concatenating outputs from the top-performing two or three feature extractor models. This fusion enables the integration of complementary information from multiple DL architectures, resulting in a more robust and discriminative feature representation compared to relying on individual models. These enhanced feature vectors are then input into various ML classifiers to yield the final classification prediction. By integrating features from diverse DL models, our approach surpasses conventional methods that rely solely on traditional feature extraction techniques. This innovative

ensemble strategy not only improves classification performance but also establishes a more comprehensive and effective framework for brain tumor diagnosis.

Specifically, our approach employs a double ensemble strategy to enhance brain tumor classification accuracy and robustness. First, we combine features extracted from the top-2 and top-3 performing DL models into aggregated feature representations. These combined features are then input into nine distinct ML classifiers for binary classification. Additionally, features extracted individually from the top-5 pre-trained DL models are presented to an ensemble of the top-2 or top-3 performing ML classifiers to generate the final prediction.

This ensemble strategy highlights the significance of combining multiple models to improve predictive accuracy and generalization. By amalgamating outputs from diverse models, our approach effectively mitigates individual biases and variance, resulting in enhanced performance metrics and more reliable classifications. The development of this robust framework addresses key challenges, including dataset variability, preprocessing requirements, and architectural complexities, establishing a scalable and effective system for brain tumor diagnosis.

In summary, the major contributions of this study are listed as follows:

- We introduced a hybrid approach combining a feature-level ensemble of pre-trained ViTs models with classifier-level ensembling of fine-tuned ML classifiers, significantly enhancing brain tumor classification accuracy.
- We implemented extensive preprocessing techniques and data augmentation strategies to improve data quality and address challenges such as noise and variability in MRI datasets.
- We conducted systematic hyperparameter tuning for multiple ML classifiers, demonstrating the pivotal role of this process in achieving superior performance and diagnostic reliability.
- We validated the proposed framework on two publicly available MRI brain tumor datasets from Kaggle, achieving SOTA performance and showcasing the effectiveness of integrating DL and ML models for medical image analysis.

Code is available at Link: https://github.com/Zahid672/ViT_Ensembling_Brain_Tumor_Classification (accessed on 10 August 2025).

The rest of the paper is structured as follows: Section 2 reviews the related work. Section 3 introduces the proposed methodology. Section 4 explains and analyzes the experimental setup. Section 5 presents the results of our experiments, while Section 6 contains a discussion of those results and points to future research directions; Section 7 concludes the paper.

2. Related Work

DL methodologies have revolutionized numerous healthcare domains, demonstrating unparalleled efficacy in recognition, prediction, and diagnosis within areas such as pathology, brain tumor analysis, lung cancer detection, abdominal imaging, cardiac assessments, and retinal evaluations [11]. The complexity and scale of healthcare services mean efficient computer-aided diagnostic techniques are becoming vital. This is especially true for the timely and precise detection of brain tumors using MRI scans. We have categorized the related work into two main groups: classical ML-based techniques and DL-based techniques, as summarized in Table 1.

Table 1. Comparison of related studies for brain tumor classification with dataset context for improved interpretability.

Ref.	Solution	AI Model	Objective	Dataset Size	Feature Extraction/Preprocessing	Accuracy (%)	Remarks
[28]	Classical ML	PNN	Brain tumor detection	25 MRI samples	k-means with fuzzy c-means	90.0	Small dataset; limited generalizability
[29]	Classical ML	Feed-forward NN	Classification into normal and abnormal	71 MRI samples	DWT	95.8	Dataset different from PNN, not directly comparable
[30]	Classical ML	Probabilistic NN	Classification into normal and abnormal	650 MRI samples	GLCM	95.0	Larger dataset; preprocessing details not fully reported
[31]	Hybrid ML	Genetic Algorithm + SVM	Classification into normal and abnormal	83 MRI samples	Wavelet-based features	98.14	Dataset and preprocessing differ from others
[32]	Classical ML	SVM	Tumor detection	41 MRI samples	Adaptive GLCM	98.0	Different dataset; comparison indicative only
[33]	Deep Learning	CNN models	Detection and classification	253 MRI samples	CNN	97.2	Dataset relatively small; different from others
[34]	Deep Learning	ResNet	Classification	3064 MRI samples	CNN	98.69	Large dataset; results comparable with other CNN-based methods
[35]	Deep Learning	Transfer Learning	Classification	3064 MRI samples	GoogleNet	99.4	Same dataset as ResNet; fair comparison
[36]	Deep Learning	CNN	Detection and classification	3064 MRI samples	CNN	97.8	Same dataset as ResNet; comparable
[37]	Deep Learning	CNN	Detection and classification	3064 MRI samples	CNN	97.9	Same dataset as ResNet; comparable
[38]	Deep Learning	CNN	Detection and classification	3064 MRI samples	CNN	91.43	Same dataset as ResNet; comparable
[39]	Deep Learning	CNN	Classification	220 MRI samples	CNN	94.5	Dataset much smaller; comparison indicative only

2.1. Brain Tumor Classification Using Traditional ML

The integration of ML methodologies into biomedical sciences has facilitated the analysis of complex, high-dimensional data, offering previously unavailable avenues for enhanced disease detection and treatment [21]. For instance, Ural et al. [28] introduced a computer-based approach for brain tumor detection using a Probabilistic Neural Network (PNN). This method employs k-means clustering for image segmentation and fuzzy c-means for feature extraction, enabling effective tumor identification. The approach was tested on 25 MRI samples, achieving a classification accuracy of 90%. While the method demonstrated promising results, its primary limitation lies in the small dataset size, which limits generalizability and robustness across diverse clinical scenarios. Additionally, its reliance on handcrafted features and conventional clustering techniques means it may struggle with complex, high-dimensional data, highlighting the need for more scalable and automated DL solutions in this domain.

Zahid et al. [29] proposed a framework combining traditional image preprocessing with DL for classifying brain MRI scans. This method employs median filtering for noise reduction, histogram equalization for contrast enhancement, Discrete Wavelet Transform (DWT) for feature extraction, and color moments (mean, standard deviation, skewness) for dimensionality reduction, followed by a DNN classifier. While achieved 95.8% accuracy, this approach has key limitations: (1) its performance heavily depends on the sequence of enhancement steps, which may not generalize across diverse MRI datasets with varying noise and contrast profiles; (2) aggressive preprocessing (e.g., histogram equalization) risks distorting subtle pathological features critical for diagnosis; and (3) the evaluation used limited datasets, raising concerns about scalability to real-world clinical settings with heterogeneous imaging protocols. These constraints highlight the need for adaptive enhancement strategies and larger, multi-institutional validation to improve robustness.

Shree and Kumar [30] proposed a methodology combining DWT for feature extraction and a PNN for classification, which obtained 95.0% accuracy on a dataset of 650 MRI images. The use of DWT helps to extract detailed frequency domain features, making the later classification process more effective. However, the study's reliance on handcrafted features through DWT limits its adaptability to more complex and varied datasets. Additionally,

the PNN classifier, while efficient for smaller datasets, may not scale well or handle the high-dimensional features extracted from larger datasets. The model also lacks robustness against noise and variability in MRI quality, which could impact its practical application in diverse clinical scenarios.

Kharrat et al. [31] proposed a methodology that combines a Genetic Algorithm (GA) for feature selection with an SVM for classification. The study reports significant accuracy improvements, demonstrating the effectiveness of GAs in optimizing the feature space for SVM classification. However, the primary limitation of this approach lies in its computational intensity; GA is resource-intensive and time-consuming, especially when dealing with larger datasets. Furthermore, the reliance on handcrafted features limits the model's ability to adapt to varying imaging conditions and complex data structures, highlighting the need for more automated DL methods to address these challenges.

Looking at these traditional ML techniques for brain tumor classification, such as those relying on handcrafted feature extraction [32] and algorithms like SVM or GA, we observed several notable limitations. These methods often depend heavily on manual feature engineering, which may not adequately capture the complex and heterogeneous patterns present in brain MRIs, leading to restricted accuracy and poor generalizability across diverse datasets. The high variability in tumor appearance, intensity, and shape can further challenge these models, as they struggle to adapt to the high-dimensional nature of MRI data. Additionally, traditional approaches are sensitive to noise and variations in the imaging protocols that produce the input data, while their performance is often constrained by the quality and relevance of the selected features. As a result, these models may underperform when confronted with real-world clinical data, highlighting the need for more automated, robust, and adaptive methods such as DL-based frameworks that can learn hierarchical features directly from raw images. Hence, researchers are increasingly adopting DL methodologies as a cornerstone for advancing brain tumor classification techniques.

2.2. Brain Tumor Classification Using DL

Ahmet and Muhammad [33] aimed to improve diagnostic accuracy and support computer-aided diagnosis. The authors based their approach on the ResNet50 architecture, modifying it by removing the last five layers and adding eight new layers tailored for tumor classification. Their hybrid model achieved a high accuracy of 97.2% and outperformed other popular CNN models such as AlexNet, DenseNet201, InceptionV3, and GoogLeNet, in benchmarking experiments. However, the study has several limitations: the evaluation was conducted on relatively limited datasets, which may not capture the full heterogeneity of real-world clinical images; the model's generalizability to unseen data and different MRI protocols was not thoroughly assessed; and the approach focuses primarily on classification without addressing tumor localization or segmentation, which are critical for clinical applications. These factors suggest the need for more diverse datasets and integration with localization methods to enhance clinical utility.

Hossein et al. [34] introduced a deep residual learning framework for categorizing brain tumors, leveraging an optimization-driven ResNet architecture enhanced by evolutionary algorithms such as ant colony optimization and differential evolution. This approach automates the design and hyperparameter tuning of the deep residual network, aiming to improve classification accuracy and reduce manual intervention. The proposed framework demonstrated strong performance, achieving an average accuracy of approximately 98.7% on benchmark datasets, highlighting its potential for robust and scalable tumor categorization in clinical imaging workflows. However, the method has notable limitations: its reliance on evolutionary optimization increases computational complexity and resource requirements, potentially hindering real-time or large-scale clinical deployment.

Deepak et al. [35] presented a brain tumor classification system that leverages deep CNN features via transfer learning to distinguish among brain tumors. The authors utilized a pre-trained GoogLeNet model to extract features from MRIs, followed by the integration of proven classifier models to perform the final classification. Their approach employed a patient-level five-fold cross-validation on a publicly available dataset to achieve a mean classification accuracy of 98%. The study further demonstrates that transfer learning is particularly effective when training data is limited, it also provides analytical insights into misclassification cases. Despite its strengths, notable limitations of this study remain. The research does not explore ensemble strategies or fine-tuning of the pre-trained models, potentially limiting the model's performance. The lack of detailed comparisons with traditional ML methods or other transfer learning frameworks also restricts insights into the method's relative advantages.

Francisco et al. [36] presented a fully automated system that leverages a multiscale deep CNN to identify brain tumors in MRIs. The proposed model processes 2D MRI slices using a sliding window approach and employs three parallel pathways with different convolutional kernel sizes to capture image features at multiple spatial resolutions, mimicking the human visual system. This architecture enables the model to distinguish between normal and abnormal MRIs. However, the approach has its limitations. The computational demands of multiscale CNNs are significantly higher than other methods, making real-time application or deployment on resource-constrained systems challenging. Additionally, the study does not thoroughly address the role of preprocessing and data augmentation, which could affect performance on noisy or imbalanced datasets. Furthermore, the evaluation primarily focused on classification and segmentation accuracy, without delving into robustness against diverse imaging conditions or comparisons with SOTA ensemble or hybrid models.

Khan et al. [37] proposed two deep CNN models for accurate brain tumor detection and classification using MRIs. The proposed framework leverages advanced CNN architectures to automatically extract discriminative features, achieving high classification accuracy. By adopting transfer learning, the model benefits from pre-trained weights learnt from extensive datasets, allowing it to perform effectively even on smaller, domain-specific datasets. The study demonstrates promising results in terms of classification precision and computational efficiency, showcasing the potential of CNNs in clinical applications. However, the work has certain limitations. The reliance on a single type of CNN architecture without exploring ensemble or hybrid approaches may restrict the model's robustness to varied data distributions. Additionally, the study does not address issues like imbalanced datasets or noisy MRI scans, which can significantly impact performance in real-world scenarios. Furthermore, the lack of comparative analysis with other SOTA methods means the proposed approach lacks validation limiting the broader applicability.

Paul et al. [38] investigated the potential of DL techniques for classifying brain tumors from MRI scans. The study utilized CNNs to automatically learn hierarchical features from MRI datasets, aiming to improve classification accuracy over traditional ML approaches. The authors demonstrated the ability of CNNs to achieve significant performance gains in tumor classification tasks, particularly through the use of architectures designed to handle high-dimensional medical imaging data. This work underscores the transformative potential of DL in medical imaging applications. However, the study has some limitations. The dataset used for training and evaluation is not explicitly detailed, raising questions about its diversity and size, which are critical for generalizability. Also, the study does not consider the impact of imbalanced datasets or preprocessing steps, such as augmentation, which are often required to enhance model robustness. Finally, the lack of a compara-

tive analysis with other contemporary methods limits the assessment of the proposed approach's effectiveness relative to existing techniques [39].

Recent research has focused on employing DL models, such as CNNs and ViTs, to automate the analysis of brain MRIs, achieving promising results in terms of accuracy and robustness [6].

The synergy between the long-range dependencies captured by transformers and the local representations learned by CNNs has led to advanced hybrid architectures that have improved performance in various medical image analysis tasks [40]. The efficacy of ViTs in feature extraction stems from their ability to model the correlation between image patches, which is particularly relevant for capturing subtle variations in tumor morphology and texture [18]. ViTs represent a paradigm shift in image analysis, diverging from traditional CNN-based approaches by leveraging self-attention mechanisms to capture long-range dependencies within images [5]. Transformers have demonstrated remarkable success in various computer vision tasks, including image classification, object detection, and semantic segmentation, owing to their ability to model contextual relationships effectively [41]. The foundational architecture of a ViTs involves partitioning an input image into a sequence of patches, which are then linearly embedded into a high-dimensional space. These embedded patches are processed through a series of transformer layers, each comprising a multi-head self-attention mechanism and a feed-forward network. This process; facilitates the learning of intricate relationships between image regions. Within the domain of medical image analysis, ViTs have garnered increasing attention due to their capacity to handle the inherent complexity and variability of medical images, offering potential advantages over conventional CNNs. The applicability of transformers in computer vision has prompted researchers to explore their utility in medical imaging, particularly for tasks such as classification, segmentation, registration, and reconstruction, achieving SOTA performance on standard medical datasets [42].

Specifically, compared with recent ensemble-based approaches in medical imaging, such as in Kang et al. [43], which applies homogeneous CNN backbones with decision-level voting, or in Ahmed et al. [44], which combines handcrafted features with CNN outputs, our framework introduces two major advancements. First, we employ heterogeneous backbones to capture complementary global and local features, followed by a feature-level fusion of the top-2/top-3 performing models that are selected through empirical validation; (rather than fixed or arbitrary combinations). Second, we integrate these fused representations into multiple hyperparameter-optimized ML classifiers before using a classifier-level ensemble to further improve predictive robustness. This dual ensembling strategy, combined with extensive preprocessing and augmentation, delivers substantial gains with an accuracy improvement of up to 5.2% on the Brain-Tumor-large with 2 classes (BT-large-2c) [45] dataset and of up to 4.8% on Brain-Tumor-small with 2 classes (BT-small-2c) [46] over the best single-model baseline, while maintaining robustness under cross-dataset variability and class imbalance. These results demonstrate the methodological and performance advantages of our approach over prior ensemble schemes.

2.3. Transformer-Based Models

ViTs have rapidly gained traction in medical imaging due to their ability to capture long-range dependencies and global context. Recent surveys comprehensively chart this shift and its implications across classification, detection, and segmentation tasks [47]. Foundational hybrid designs such as TransUNet and UNETR integrate transformer encoders with U-Net-style decoders, have demonstrated strong performance across multi-organ and volumetric segmentation. Subsequent variants (e.g., Swin-UNETR, stagewise/v2 improve-

ments, and self-supervised pretraining) further strengthened on 3D medical segmentation and BraTS-style tumor tasks [48,49].

2.3.1. Hybrid CNN–ViT Models

Beyond the canonical hybrids, recent works have incorporated attention modules and flexible encoders to blend local convolutional texture features with transformer-level global reasoning (e.g., FDR-TransUNet, DA-TransUNet, Swin-Unet3D) [50,51].

2.3.2. Brain tumor MRI (Classification/Segmentation)

Multiple recent studies have applied ViT or hybrid variants directly to brain tumor MRI, reporting competitive or improved accuracy vs. CNN baselines and highlighting robustness/efficiency advantages; examples include fine-tuned ViT classifiers, rotation-invariant ViTs, randomized/token-merging ViTs, and hybrid ViT-GRU/ensemble schemes [52,53].

2.4. Summary

In summary, the majority of existing research indicates that DL techniques achieve significantly higher accuracy in brain MRI classification compared to traditional ML methods. However, DL systems require substantial amounts of training data to outperform conventional ML approaches. Recent studies have established DL as a leading method in medical image analysis, though these techniques also present specific challenges that must be addressed in brain tumor classification and segmentation tasks. In this study, we propose a fully automated hybrid methodology for brain tumor classification, integrating two main components: pre-trained DL models for extracting deep features from brain MRIs and ML classifiers for precise brain MRI classification. Our approach is distinguished by four unique steps. First, deep features are extracted from MRIs using pre-trained DL models. Second, these features are rigorously analyzed using nine different ML classifiers. Third, the two and three top-performing DL feature extractor models, based on accuracy across various classifiers, are combined into a deep feature ensemble, which is subsequently input into different ML classifiers for final predictions. Finally, we explored constructing an ensemble of the top two and top three ML classifiers by incorporating features derived from the top five individual pre-trained DL models.

3. Proposed Methodology

3.1. Overview

As illustrated in Figure 1, the proposed architecture for brain tumor classification utilizes a multi-faceted approach, integrating DL techniques with ML classifiers to achieve enhanced accuracy and robustness. As illustrated in Figure 2, the process initiates with the meticulous preprocessing of MRIs, a crucial step that encompasses cropping to isolate the region of interest, resizing to standardize the input dimensions for subsequent processing, and augmentation to artificially expand the dataset and improve the model's generalization capability. These preprocessed images then serve as the input for pre-trained ViTs models, chosen for their powerful feature extraction capabilities, which capitalize on transfer learning to adapt knowledge gained from large-scale image datasets to the specific task of brain tumor classification. The extracted features, representing high-level abstractions of the input images, are then subjected to a rigorous evaluation process using a diverse array of ML classifiers; this allows for a comparative analysis of their performance in distinguishing between normal and abnormal brain tumors.

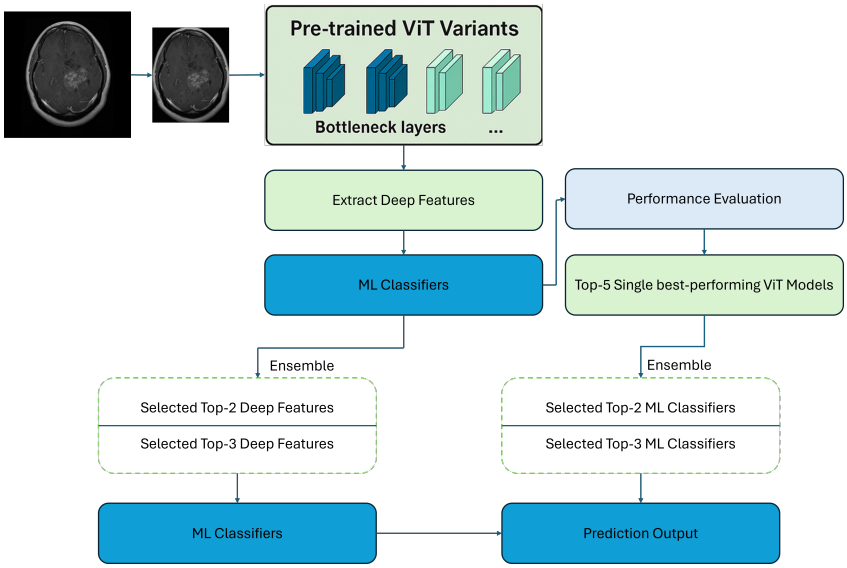

Figure 1. Proposed double ensemble architecture for brain tumor MRI classification. The framework integrates hierarchical ensembles of deep feature representations extracted from multiple pre-trained ViTs and ML classifiers. Combining the outputs of top-performing feature extractors and classifiers enhances predictive accuracy and robustness, improving the differentiation between normal and abnormal brain MRIs.

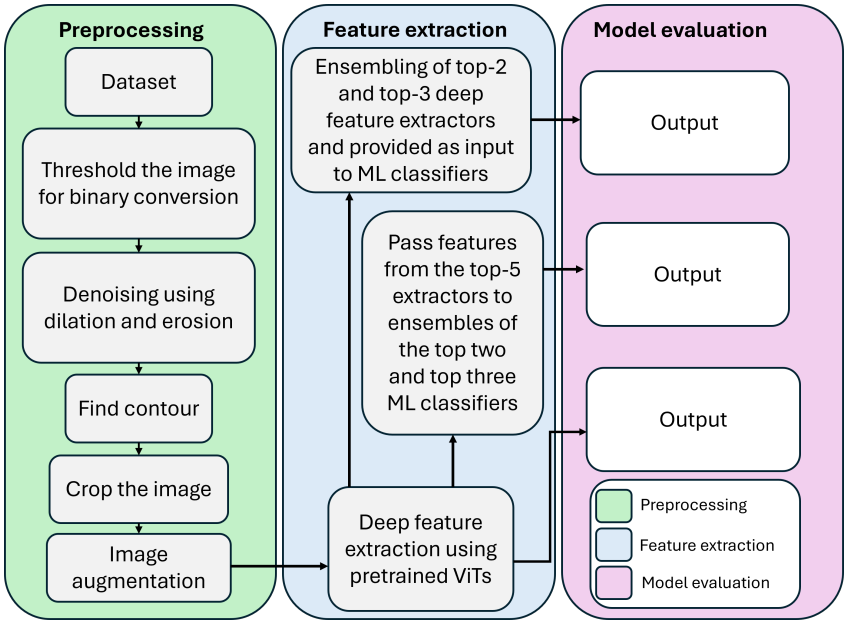

Figure 2. Workflow of the proposed double ensembling framework for brain tumor MRI classification. The pipeline includes preprocessing, binary conversion, denoising, contour detection, cropping, and image augmentation, followed by deep feature extraction using pre-trained ViTs. Features are combined through ensemble strategies that exploit the top-performing models and classifiers, with final classification accuracy obtained from both the ensemble and fine-tuned single classifiers.

Based on the empirical performance of the ML classifiers, a strategic selection process is undertaken to identify the top two or three deep feature sets, which are then combined within an ensemble module. This ensemble approach leverages the complementary strengths of different feature representations, potentially leading to a more comprehensive and discriminative feature space for classification. The concatenated features, representing a fusion of the most salient information extracted by the ViT models, are then fed as input to another set of ML classifiers, which are tasked with making the final prediction regarding

the type of brain tumor present in the input image. This cascaded approach, involving both deep feature extraction and ensemble learning, aims to maximize the accuracy and reliability of brain tumor classification, offering a potentially valuable tool for assisting medical professionals in diagnosis and treatment planning.

3.2. Model Selection and Fusion Rationale

The rationale for selecting specific ViT variants and ML classifiers was guided by both empirical evidence and practical considerations. For the ViT variants, we selected a diverse set of pre-trained models differing in patch sizes (e.g., 8, 16, 32), resolutions (224 vs. 384), and model scales (tiny, small, base, large). This ensured that the feature representations captured multiple levels of spatial granularity and model capacity, enabling the ensemble to leverage complementary strengths of diverse models rather than redundant features of similar ones.

For the ML classifiers, we chose a representative mix of widely used families, including linear models (SVM with linear kernel), non-linear kernel-based models (SVM with RBF and sigmoid), probabilistic models (GaussianNB), distance-based learning (KNN), tree-based ensembles (Random Forest, XGBoost, AdaBoost), and neural-based models (MLP). This diversity allowed us to compare classical learners with strong baselines and modern ensemble methods, while also evaluating how well they complemented ViT-derived deep features.

The criteria for selecting the top-2 and top-3 models for fusion were based on the performance of individual models across validation experiments. Specifically, we ranked models/classifiers by their balanced accuracy and consistency across multiple runs. The top-2 and top-3 performers were then ensembled to examine whether combining high-performing but complementary models could further enhance robustness and generalization.

3.3. Datasets

In this study, we performed a series of experiments using two publicly available brain MRI datasets to classify brain tumors. The first dataset, termed BT-small-2c, was obtained from Kaggle [46] and includes 253 images, 155 with tumors and 98 without. The second dataset, labeled BT-large-2c, was also sourced from Kaggle and contains 3000 images evenly distributed between 1500 tumor-present and 1500 normal cases [45]. These datasets provide a comprehensive basis for assessing brain tumor classification methods. Figure 3 presents representative samples from each dataset, illustrating the diversity in imaging modalities and tumor characteristics.

Both datasets were divided into a training set, representing 80% of the data, and a test set, accounting for the remaining 20%. Table 2 summarizes the datasets used in our experiments, while Figure 3 showcases sample brain MRIs from the BT-small-2c and BT-large-2c datasets. These datasets collectively contribute to a comprehensive and diverse resource for assessing the performance of brain tumor classification methodologies, allowing for more robust and generalizable models to be developed.

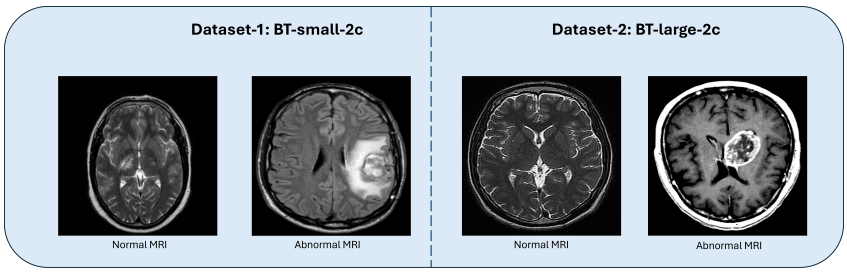

Figure 3. Sample MRI scans (normal and abnormal) from Dataset-1 (BT-small-2c) and Dataset-2 (BT-large-2c).

Table 2. Details of each dataset.

Types	Number of Classes	Training Set	Test Set
BT-small-2c	2	202	51
BT-large-2c	2	2400	600

3.4. Pre-Processing

In the selected brain MRI datasets, many images contain extraneous areas that negatively impact classification accuracy. To enhance performance, it is crucial to crop these images, removing irrelevant regions while retaining only the essential data. We utilize the cropping technique outlined in [54], which relies on calculating extreme points.

The cropping process for MRI images, illustrated in Figure 4, begins by applying thresholding to convert the images into a binary format, which is followed by dilation and erosion operations to reduce noise. Subsequently, the largest contour is identified in the binary images, and the four extreme points (topmost, bottommost, leftmost, and rightmost) are located. Using these points as a reference, the images are cropped and then resized using bicubic interpolation, which produces smoother curves and better manages the prominent edge noise often present in MRI images.

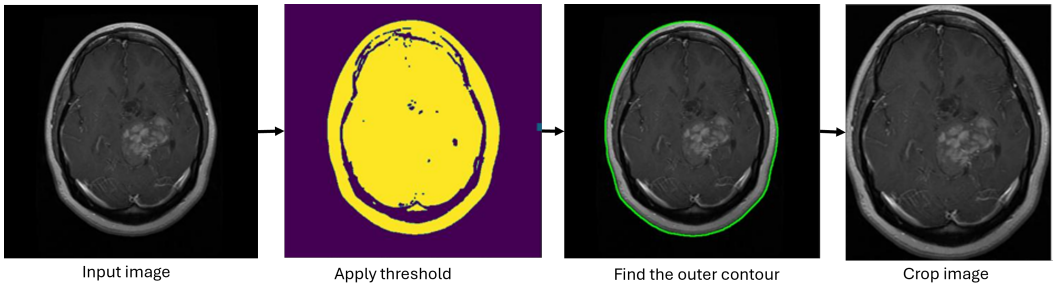

Figure 4. Preprocessing steps for cropping MRI images using binary thresholding and contour detection.

Furthermore, to address the limited size of our MRI dataset, we employed image augmentation, a technique that artificially increases the entries in a dataset by altering the original images. This approach creates multiple variations of each image by adjusting parameters such as scale, rotation, position, and brightness; among other attributes. Studies have shown [55,56] that augmenting datasets can enhance model classification accuracy more effectively than collecting additional data.

For our image augmentation process, we applied two specific methods: rotation and horizontal flipping. The rotation involved randomly rotating the input images by 90 degrees one or more times. Afterward, horizontal flipping was performed on each rotated image, further enriching the dataset with additional training samples.

3.5. Deep Feature Extraction Using Pre-Trained Visions Transformers

We employed many variants of ViT-based models as DL feature extractors, utilizing their capacity to autonomously recognize and capture essential features without manual intervention. To address the limitations posed by the small size of our MRI dataset, we adopted a transfer learning approach as shown in Figure 5, for developing our feature extractor. Training and fine-tuning a DNN from scratch would not be feasible in this scenario. Instead, we leveraged the fixed weights of each ViT model, which was pre-trained on the large-scale ImageNet dataset, to efficiently extract deep features from brain MRI images. This strategy enhances the efficiency and reliability of our feature extraction process.

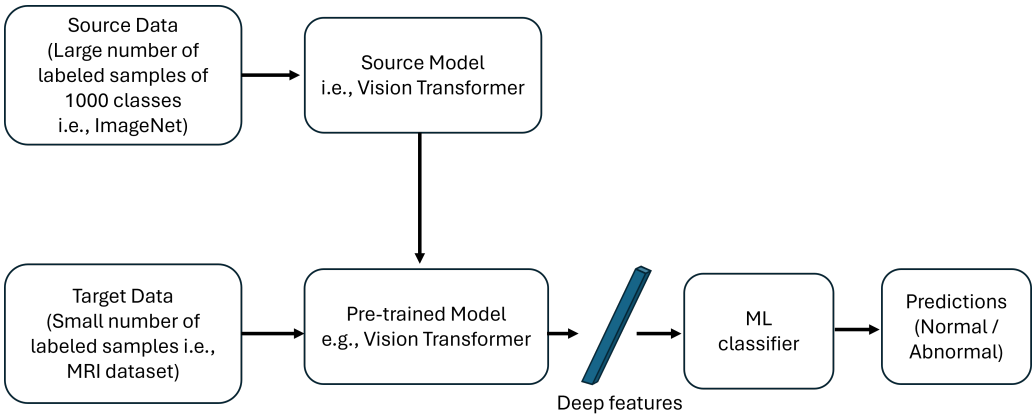

Figure 5. Methodology of transfer learning using pre-trained models as feature extractors for brain tumor MRI classification.

In this study, we utilized the following ViT variants (vit_base_patch16_224, vit_base_patch32_224, vit_large_patch16_224, vit_small_patch32_224, deit3_small_patch16_224, vit_base_patch8_224, vit_tiny_patch16_224, vit_small_patch16_224, vit_base_patch16_384, vit_tiny_patch16_384, vit_small_patch32_384, vit_small_patch16_384, vit_base_patch32_384 [57–59]) to extract the deep features from MRIs as shown in Figure 6.

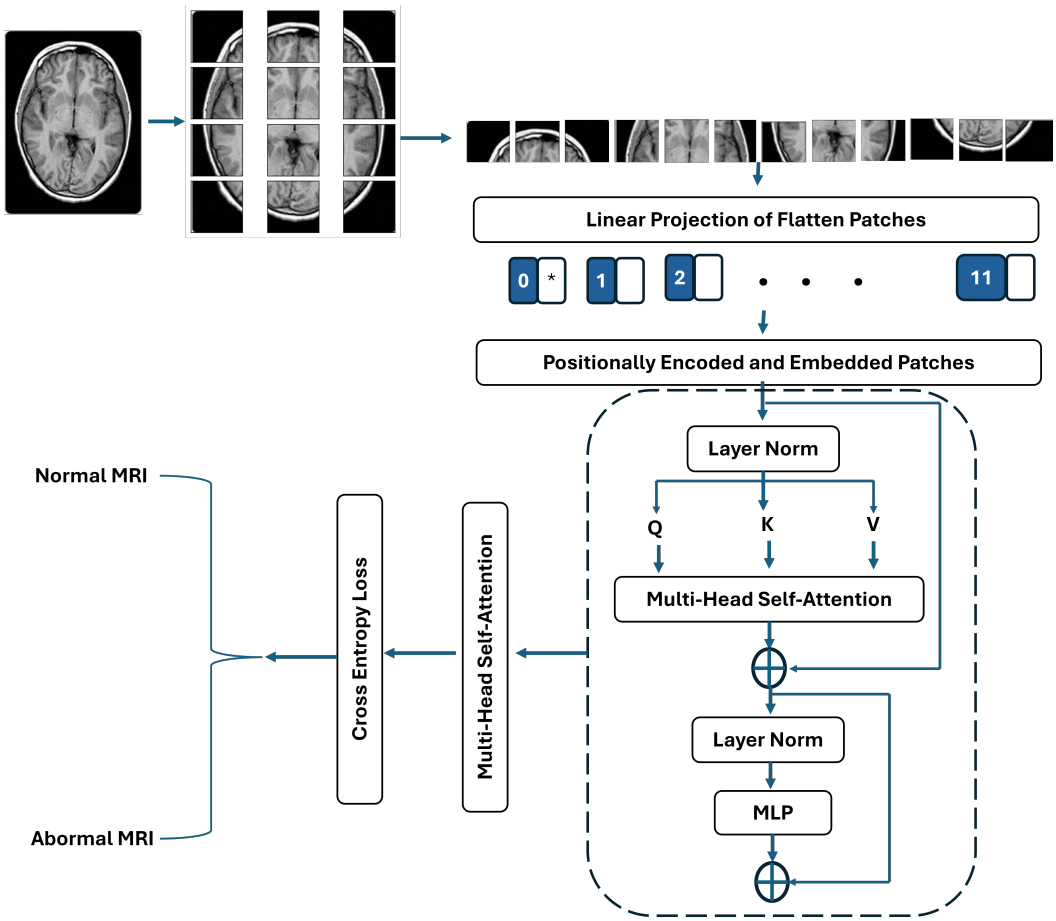

Figure 6. Schematic overview of the ViT architecture for brain MRI classification, illustrating patch embedding, Transformer encoding with multi-head self-attention, and final classification into normal and abnormal categories.

3.6. Brain Tumor Classification Using ML Classifiers

Over the past few years, many studies have focused on traditional or classical ML techniques for brain tumor diagnosis. ML algorithms can aid physicians in interpreting medical imaging findings and reduce interpretation times [60].

To address the complexities inherent in brain tumor classification, this paper embarks on a comparative analysis of several ML classifiers that leverage deep features extracted via pre-trained DL models. The integration of DL for feature extraction with traditional ML classifiers represents a powerful paradigm for medical image analysis that capitalizes on the strengths of both approaches. This strategy allows for the automated extraction of relevant image features, which are then fed into ML classifiers for categorization. By utilizing deep features extracted from pre-trained models, the system can leverage the knowledge learned from vast datasets, even when the available brain tumor dataset is limited in size. This approach is particularly advantageous in medical imaging, where acquiring large, annotated datasets can be challenging. The performance of these classifiers is rigorously evaluated and compared to determine their suitability for brain tumor classification. This process itself provides insights into the most effective approaches for this critical task. These classifiers are based on various systems, including neural networks with an MLP architecture, XGBoost, Gaussian Naïve Bayes, Adaptive Boosting, k-Nearest Neighbors (KNN), RF, and SVMs with linear, sigmoid, and RBF kernels. The selection of these classifiers was motivated by their widespread use and proven effectiveness in various classification tasks, as well as their suitability for handling the types of data encountered in medical image analysis. The implementation of these classifiers is facilitated by the scikit-learn ML library, a widely used and well-documented tool that provides a comprehensive suite of ML algorithms and utilities. Details of the ML classifiers used in our brain tumor classification experiments, and of their hyper-parameter configurations, are given in the subsequent sections to provide a comprehensive overview of the experimental setup and methodology.

3.6.1. MultiLayer Perceptron

We implemented a MultiLayer Perceptron (MLP) as one of the ML classifiers for the extracted ViT features. The network architecture and hyperparameters were selected through grid search as shown in Table 3, using ReLU activations and the Adam optimizer. Unlike the full architecture description, we focus here on the fact that MLP performed well due to its ability to model non-linear decision boundaries in high-dimensional fused feature spaces, achieving competitive accuracy in both datasets. During forward propagation, outputs for each layer are computed based on the inputs, as well as the weights and biases associated with the neurons. Mathematically, the output $h^{(l)}$ of the l-th layer is computed as:

$$y^{(l)} = w^{(l)} h^{(l-1)} + b^{(l)}, \tag{1}$$

$$h^{(l)} = ReLU(z^{(l)}), \tag{2}$$

where $w^{(l)}$ and $b^{(l)}$ denote the weight matrix and bias vector of the $l-$th layer, $y^{(l)}$ denotes the linear transformation of the input, and $h^{(l-1)}$ represent the output from the previous layer ReLU $(.)$ is utilized as the activation function and is defined as follows:

$$ReLU(x) = max(0, x) \tag{3}$$

Table 3. Selected hyperparameter with search space for ML classifiers.

Model	Hyperparameter	Search Space	Type
XGBoost	max_depth	[3, 5, 7],	Discrete
	learning_rate	[0.1, 0.01, 0.001]	Continuous
	subsample	[0.5, 0.7, 1]	Continuous
	n_estimators	[100, 200, 300]	Discrete
MLP	hidden_layer_sizes	[(50,), (100, 22), (100, 100, 50), (100, 50, 36, 30), (100, 100, 200, 150, 100)]	Discrete
	activation	[relu, tanh, logistic]	
	solver	[adam, sgd, lbfgs]	Categorical
	max_iter	[1000]	Discrete
	momentum	[0.9, 0.95, 0.99]	Continuous
Gaussian NB	var_smoothing	$[1 \times 10^{-9}, 1 \times 10^{-8}, 1 \times 10^{-7}, 1 \times 10^{-6}, 1 \times 10^{-5}]$	Continuous
	priors	[None, [0.3, 0.7], [0.4, 0.6], [0.5, 0.5]]	Continuous
Adaboost	n_estimators	[50, 70, 90, 120, 180, 200]	
	learning_rate	[0.001, 0.01, 0.1, 1, 10]	
KNN	n_neighbors	list(range(1, 31))	Discrete
	weights	[uniform, distance]	Categorical
	algorithm	[autom ball_tree, kd_tree, brute]	
	leaf_size	list(range(10, 51, 5))	Discrete
	p	[1, 2]	Discrete
	metric	[euclidean, manhattan, minkowski]	Categorical
	n_jobs	$[-1]$	Discrete
RF	n_estimators	[100, 200, 300, 400, 500]	
	max_depth	[None, 10, 20, 30, 40, 50]	
	min_samples_split	[2, 5, 10]	Discrete
	min_samples_leaf	[1, 2, 4]	Discrete
	max_features	[auto, sqrt, log2]	Discrete
	bootstrap	[True, False]	Discrete
	criterion	[gini, entropy]	Categorical
	oob_score	[True, False]	
	random_state	[42]	
SVM_linear	C	[0.1, 1, 10, 100, 1000]	Continuous
	kernel	[linear]	Categorical
	tol	$[1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}]$	
	class_weight	[None, balanced]	
	random_state	[42]	Discrete
SVM_sigmoid	kernel	sigmoid	
	C	[0.1, 1, 10, 100]	
	gamma	[scale, auto]	
	coef0	[0.0, 0.1, 0.5, 1.0]	
	tol	$[1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}]$	Continuous
	class_weight	[None, balanced]	Categorical
	shrinking	[True, False]	Continuous
	probability	[True, False]	
	cache_size	[200.0, 500.0, 100.0]	
	random_state	[42]	
SVM_RBF	C	[0.1, 1, 10, 100]	Discrete
	gamma	[scale, auto, 0.1, 1, 10]	Categorical
	kernel	[rbf]	
	class_weight	[None, balanced]	
	shrinking	[True, False]	
	probability	[True, False]	
	tol	$[1 \times 10^{-3}, 1 \times 10^{-4}]$	
	cache_size	[200, 500, 1000]	
	max_iter	$[-1, 1000, 5000]$	Discrete

The weights and biases are updated iteratively using the following rules:

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1)\nabla L_t, \tag{4}$$

$$V_t = \beta_2 V_{t-1} + (1 - \beta_2)(\nabla L_t)^2, \tag{5}$$

$$\hat{M}_t = \frac{M_t}{1 - \beta_1^t}, \quad \hat{V}_t = \frac{V_t}{1 - \beta_2^t}, \tag{6}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{M}_t}{\sqrt{\hat{V}_t} + \epsilon}, \tag{7}$$

where M_t and V_t represent the first and second moment estimates, respectively, ∇L_t denotes the gradient at time t, η is the learning rate (set to 0.001) and β_1, β_2, and ϵ are hyperparameters of the optimizer.

Learning Process: The learning process of the MLP involves minimizing a loss function L, which quantifies the difference between predicted outputs $\hat{y}$ and actual outputs y. For a dataset with N samples, the loss is typically computed as the Mean Squared Error (MSE) for regression tasks:

$$L = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{8}$$

The MLP progressively enhances its generalization and prediction accuracy by iteratively updating the weights and biases using the Adam optimizer. The integration of the ReLU activation function with the Adam solver facilitates efficient training and helps address challenges like vanishing gradients.

3.6.2. Gaussian Naive Bayes

This classifier is a ML model that assumes features are conditionally independent given the class label. In this study, we utilize the Gaussian Naïve Bayes (NB) classifier as one of our approaches for brain tumor classification. With this classifier, the conditional probability *P(y | X)* is calculated by multiplying the individual conditional probabilities, relying on the naïve assumption of independence among the features.

$$P(y|X) = \frac{P(y)P(X|y)}{P(X)} = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(X)} \tag{9}$$

Here, X represents a data instance derived from the deep features of a brain MR image, expressed as a feature vector $(x_1, \ldots, x_n)$. The variable y denotes the target class, that is, the type of brain tumor—with two classes for the BT-small-2c and BT-large-2c MRI datasets. Since $P(X)$ is the same for all classes, classification of a given instance is based on the remaining terms as follows:

$$\hat{Y} = \arg\max_{Y} \; P(Y) \left(\prod_{i=1}^{n} P(x_i \mid Y) \right) \tag{10}$$

where $(x_i|Y)$ is computed under the assumption that the feature likelihood follows a Gaussian distribution, as expressed below:

$$p(x_i|Y) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} exp(\frac{(x_i - \mu_Y)^2}{2\sigma_Y^2}) \tag{11}$$

where the parameters μ_Y and σ_Y are determined via maximum likelihood estimation.

Here, the smoothing parameter, which denotes the fraction of the highest variance among all features added to the variances to ensure computational stability, is set to 10^{-9}; in alignment with the default setting in the scikit-learn ML library.

3.6.3. AdaBoost

AdaBoost, introduced in [61], is an ensemble learning algorithm designed to enhance overall performance by combining multiple classifiers. It constructs a strong classifier through an iterative process that assigns weights to individual weak classifiers, updating them during each boosting iteration. This approach trains on weighted data samples to enable the combined model to accurately predict class labels, such as distinguishing between binary datasets (BT-small-2c and BT-large-2c). Since any ML classifier supporting sample weights can serve as a base model, we chose the decision tree classifier due to its widespread use with AdaBoost. Additionally, we configured the algorithm to use 100 estimators.

3.6.4. K-Nearest Neighbors

The KNN algorithm was applied according to Euclidean distance with the optimal k determined via grid search as depicted in Table 3. Its performance was influenced by the high-dimensional ViT feature space, where smaller k values better captured local decision boundaries. While simple, KNN benefited from the complementary nature of the fused features, particularly in the BT-small-2c dataset. However, this classifier showed reduced scalability compared to more complex models. The Euclidean distance, d, between two points x and y is calculated as follows:

$$d(X, Y) = \sqrt{(\sum_{i=1}^{N} (X_i - Y_i)^2)} \tag{12}$$

$$P = (Y_i), i, \ldots, n \tag{13}$$

where Y_i represents a specific training sample in the training dataset, n denotes the overall number of training samples, and c_i corresponds to the class label associated with Y_i.

During the testing phase, the distances between the new feature vector and the stored feature vectors from the training data are calculated. The new example is then classified based on the majority vote of its k-nearest neighbors. The accuracy of the algorithm was evaluated during the testing phase using the correct classifications. If the performance is unsatisfactory, the value of k can be adjusted to achieve a more acceptable level of accuracy. In this study, the number of neighbors is varied from 1 to 4, and the value that resulted in the highest accuracy was selected.

3.6.5. Random Forest

RF introduced by Breiman [62], is an ensemble learning algorithm that generates multiple decision trees using the bagging technique. It classifies new data points, such as the deep features extracted from brain MRI images, into specific target categories. For the BT-small-2c and BT-large-2c datasets, RF differentiates between two classes. During the construction of each tree, RF randomly selects n features to identify the optimal split point, using the Gini index as the cost function. This random feature selection reduces correlations between trees, thereby improving the overall accuracy of the ensemble. To make predictions, the algorithm passes the input data through all the decision trees, with each tree producing a class prediction. The final class label is determined through majority voting, where the class with the highest number of votes is selected as the predicted label.

In this study, the square root of the total number of features was used to determine the optimal split criteria. Tree sizes ranging from 1 to 150 were evaluated, and the configuration that achieved the highest accuracy was selected.

3.6.6. Support Vector Machine

We evaluated SVMs with linear, sigmoid, and RBF kernels, optimizing C and Gamma through grid search as shown in Table 3. The RBF kernel consistently yielded the best performance on fused ViT features, likely due to its flexibility in handling non-linear relationships in high-dimensional data. SVM's strong performance across both datasets underscores its effectiveness when combined with discriminative feature-level fusion.

$$f(x_i) = \sum_{n=1}^{N} \alpha_n y_n K(x_n, x_i) + b \tag{14}$$

Here, the support vectors, denoted as x_n, represent the deep features extracted from brain MR images. The Lagrange multipliers, α_n, are coefficients assigned to each support vector during the optimization process. The target classes, y_n, correspond to the classification labels in the datasets used in this study, two binary-class datasets (normal and abnormal).

We employed the most widely utilized SVM kernel functions: (1) the linear kernel, (2) the sigmoid kernel, and (3) the RBF kernel as outlined in Table 4. Additionally, SVMs rely on two critical hyperparameters: C and Gamma. The C hyperparameter governs the soft margin cost function, determining the impact of each support vector, while Gamma influences the degree of curvature in the decision boundary. We tested 'scale' and 'auto' Gamma values of [0.1, 1, and 10], and C values of [0.1, 1, 10, 100], ultimately choosing the Gamma and C combination that yielded the highest accuracy.

Table 4. Kernel types and their required parameters.

Kernel	Equation	Parameters
Linear	$K(x_n, x_i) = (x_n, x_i)$	-
Sigmoid	$K(x_n, x_i) = tanh(\gamma(x_n, x_i) + C)$	γ, C
RBF	$K(x_n, x_i) = exp(-\gamma\|x_n - x_i\|^2 + C)$	γ, C

3.7. HyperParameter Tuning for ML Models

Hyperparameter Optimization (HPO) involves identifying the most effective set of hyperparameter values and the optimal arrangement for categorical hyperparameters. This process aims to improve model performance by reducing a predefined loss function, leading to more accurate results with fewer mistakes. Hyperparameter tuning [63] refers to the method of crafting the perfect model structure with an ideal hyperparameter setup. Since each ML algorithm comes with its own unique hyperparameters, manually adjusting them demands a thorough knowledge of the models and their corresponding hyperparameter settings. Some proposed [64] automated hyperparameter tuning techniques, such as random search, grid search, and Bayesian optimization, offer greater adaptability compared to conventional approaches for choosing the best hyperparameters, ultimately boosting model performance more efficiently.

ML tasks can be described as developing a model M that reduces a predefined loss function $L(X_{Ts}; M)$ on a specific test set X_{Ts}, where the loss function (L) represents the error rate. A learning algorithm A utilizes a training set X_{Tr} to construct the model M; frequently, this is a nonconvex optimization challenge. The learning algorithm A incorporates certain hyperparameters lambda (λ), and the model M is then defined as $M = A(X_{Tr}); \lambda$. The goal

of HPO is to identify the optimal settings (λ^{ast}) that produce an ideal model (M^{ast}), which minimizes the loss function $L(X_{Ts}; M)$.

$$\lambda^* = \underset{\lambda}{\operatorname{argmin}}\, L(X_{Ts}; A(X_{Tr}; \lambda)) \qquad = \underset{\lambda}{\operatorname{argmin}}\, F(X_{Ts}, X_{Tr}, A, \lambda, L) \tag{15}$$

Here, F represents the model's objective function, which takes λ, a set of hyperparameters and returns the associated loss. The loss function L and learning algorithm are selected, and the datasets X_{Ts} (test set) and X_{Tr} (training set) are provided [65]. These elements vary based on the chosen model, the hyperparameter search space, and the selected ML classifiers.

In this study, hyperparameter tuning was uniformly applied to nine ML models, with each model undergoing the same optimization process using grid search. This approach aims to determine the optimal hyperparameters for each model, striking a balance between accuracy and computational efficiency. By maintaining consistency in the tuning process, the models can be evaluated under comparable conditions, ensuring a fair assessment of their predictive performance and generalization capabilities.

As illustrated in Table 3, to enhance the effectiveness of our ML models, we perform HPO through grid search methods [66]. The hyperparameter settings that produce the best outcomes on the validation set are chosen for each model. Tables 5–7, depict the results obtained from our ML models following this hyperparameter tuning process.

Table 5. Accuracies of pre-trained ViT models using fine-tuned hyperparameters of ML classifiers on the non-preprocessed BT-large-2c dataset. The best performance is indicated in bold.

Deep Feature from the Pre-Trained ViT Models	ML Classifier Accuracy									
	XGBoost	**MLP**	**GaussianNB**	**Adaboost**	**KNN**	**RFClassifier**	**SVM_linear**	**SVM_sigmoid**	**SVM_RBF**	**Average**
vit_base_patch16_224	0.9001	0.9800	0.8515	0.9731	0.9652	0.9752	0.9705	0.9200	0.9712	0.9452
vit_base_patch32_224	0.8832	0.9771	0.8600	0.9785	0.9532	0.9652	0.9800	0.9441	0.9804	0.9469
vit_large_patch16_224	0.9221	0.9851	0.8536	0.9855	0.9702	0.9632	0.9831	0.9591	0.9732	**0.9550**
vit_small_patch32_224	0.8911	0.9728	0.8752	0.9623	0.9723	0.9501	0.9506	0.9199	0.9800	0.9416
deit3_small_patch16_224	0.8544	0.9900	0.7857	0.9513	0.9602	0.9451	0.9401	0.8408	0.9766	0.9160
vit_base_patch8_224	0.9121	0.9917	0.8469	0.9641	0.9532	0.9502	0.9700	0.8600	0.9802	0.9365
vit_tiny_patch16_224	0.9016	0.9850	0.8321	0.9607	0.9700	0.9500	0.9415	0.8722	0.9612	0.9305
vit_small_patch16_224	0.9024	0.9900	0.8400	0.9802	0.9739	0.9700	0.9602	0.9192	0.9709	0.9452
vit_base_patch16_384	0.9132	0.9680	0.8768	0.9700	0.9621	0.9699	0.9631	0.9504	0.9700	0.9493
vit_tiny_patch16_384	0.9056	0.9666	0.7854	0.9703	0.9700	0.9708	0.9523	0.8700	0.9904	0.9313
vit_small_patch32_384	0.9214	0.9835	0.8899	0.9733	0.9833	0.9667	0.9612	0.9212	0.9808	0.9535
vit_small_patch16_384	0.8932	0.9627	0.8244	0.9621	0.9801	0.9700	0.9510	0.9219	0.9803	0.9384
vit_base_patch32_384	0.9021	0.9732	0.8241	0.9800	0.9800	0.9623	0.9701	0.9505	0.9612	0.9448
Average	0.9002	**0.9789**	0.8420	0.9701	0.9687	0.9622	0.9611	0.9115	0.9751	

The heatmap in Figure 7 provides a comparative overview of the classification accuracies obtained by combining different pre-trained ViT feature extractors with various ML classifiers on the BT-large-2c dataset without preprocessing. Each row corresponds to a ViT model and each column to a classifier, with the color gradient indicating accuracy values. The results show that most ViT features, when paired with classifiers such as SVM (RBF), MLP, and KNN, consistently achieve high accuracies, often exceeding 0.97. In contrast, the GaussianNB performance is comparatively weak, with accuracies dropping below 0.80 for certain models. Among the ViT variants, models like vit_small_patch16_224 and vit_base_patch8_224 deliver the strongest results, reaching up to 0.99 accuracy, while deit3_small_patch16_224 demonstrates slightly lower performance across the various classifiers. Overall, the Figure 7 highlights that classifier selection has a significant impact on performance, with SVM (RBF), MLP, and KNN emerging as the most reliable choices when leveraging ViT-based deep features.

Table 6. Accuracies of pre-trained ViT models using fine-tuned hyperparameters of ML classifiers on preprocessed BT-large-2c dataset. The top-3 deep features are represented using ⋆. The best performance is indicated in bold.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy									
	XGBoost	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF	Average
vit_base_patch16_224	0.9483	0.9917	0.865	0.9817	0.9867	0.98	0.9917	0.9383	0.99	0.9637
vit_base_patch32_224 ⋆	0.9517	0.995	0.8767	0.9883	0.9883	0.9783	0.9917	0.9633	0.995	**0.9698**
vit_large_patch16_224 ⋆	0.96	0.995	0.8683	0.9933	0.985	0.9833	0.9967	0.9833	0.9967	**0.9735**
vit_small_patch32_224	0.93	0.9867	0.8917	0.9833	0.9933	0.9683	0.9717	0.9367	0.995	0.9619
deit3_small_patch16_224	0.8717	0.9867	0.7983	0.96	0.97	0.9517	0.955	0.855	0.99	0.9265
vit_base_patch8_224	0.94	0.995	0.855	0.9833	0.985	0.9733	0.995	0.8817	0.9933	0.9557
vit_tiny_patch16_224	0.91	0.9867	0.865	0.9767	0.985	0.9717	0.9533	0.895	0.9867	0.9478
vit_small_patch16_224	0.94	0.9917	0.8583	0.9917	0.985	0.98	0.975	0.9383	0.9933	0.9615
vit_base_patch16_384	0.94	0.9883	0.895	0.9867	0.9867	0.9783	0.985	0.9633	0.985	0.9676
vit_tiny_patch16_384	0.9167	0.9917	0.8083	0.98	0.9883	0.9833	0.9733	0.8983	0.9933	0.9481
vit_small_patch32_384	0.9433	0.99	0.9067	0.9883	0.99	0.9867	0.975	0.9483	0.9967	0.9694
vit_small_patch16_384	0.9367	0.9883	0.8333	0.9817	0.99	0.99	0.975	0.945	0.9917	0.9591
vit_base_patch32_384 ⋆	0.9517	0.9933	0.8833	0.9917	0.985	0.9883	0.99	0.965	0.995	**0.9715**
Average	0.9338	**0.9908**	0.8619	0.9836	0.986	0.9779	0.9791	0.9317	0.9924	

Table 7. Accuracies of pre-trained ViT models using fine-tuned hyperparameters of ML classifiers on BT-small-2c dataset. The top-3 deep features were represented using ⋆. The best performance is indicated in bold.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy								
	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF	Average
vit_base_patch16_224 ⋆	0.9750	0.8944	0.9500	0.9105	0.9250	0.9750	0.9339	0.9750	**0.9423**
vit_base_patch32_224	0.9000	0.8460	0.9000	0.9355	0.9589	0.9016	0.9339	0.9339	0.9137
vit_large_patch16_224	0.9000	0.8460	0.8089	0.8194	0.9000	0.9750	0.9750	0.9750	0.8999
vit_small_patch32_224 ⋆	0.9500	0.9105	0.9339	0.9105	0.9339	0.9500	0.9589	0.9750	**0.9403**
deit3_small_patch16_224	0.8516	0.7032	0.8250	0.8371	0.8500	0.9427	0.7427	0.9016	0.8318
vit_base_patch8_224	0.9500	0.8782	0.9089	0.8855	0.9000	0.9339	0.9339	0.9500	0.9175
vit_tiny_patch16_224	0.9589	0.8944	0.8839	0.9032	0.9250	0.8516	0.9427	0.9750	0.9168
vit_small_patch16_224 ⋆	0.9750	0.8282	0.9750	0.9016	0.9500	0.9339	0.9750	0.9750	**0.9392**
vit_base_patch16_384	0.9339	0.9032	0.9500	0.8855	0.9500	0.9750	0.9339	0.9589	0.9363
vit_tiny_patch16_384	0.9266	0.8032	0.8839	0.8121	0.9250	0.8194	0.9339	0.9589	0.8829
vit_small_patch32_384	0.9427	0.9089	0.9339	0.9427	0.9339	0.9355	0.9339	0.9266	0.9323
vit_small_patch16_384	0.9500	0.8766	0.9500	0.8121	0.9177	0.9177	0.9589	0.9750	0.9198
vit_base_patch32_384	0.8750	0.8137	0.9089	0.9016	0.9250	0.9750	0.9339	0.9750	0.9135
Average	0.9299	0.8543	0.9086	0.8813	0.9226	0.9297	0.9300	**0.9581**	

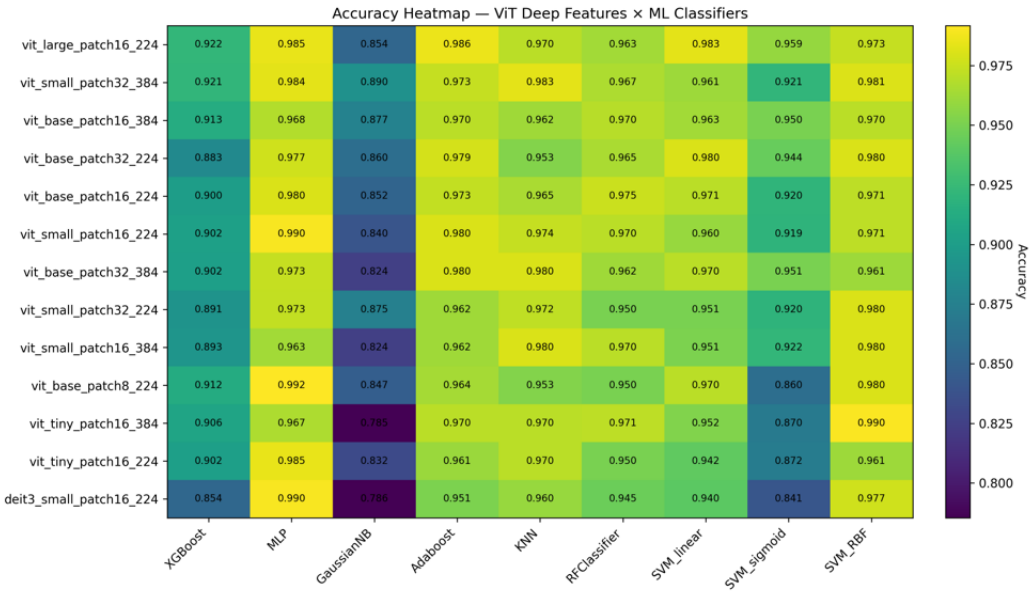

Figure 7. Accuracy heatmap illustrating the performance of different pre-trained ViT feature extractors combined with various ML classifiers on the BT-large-2c dataset without preprocessing.

In Figure 8, each cell represents the classification accuracy for a specific ViT-classifier pair, with values color-coded from lower (dark shades) to higher (bright shades) accuracy.

The visualization highlights performance trends across models, making it easier to identify top-performing combinations such as vit_large_patch16_224 with MLP and SVM-RBF.

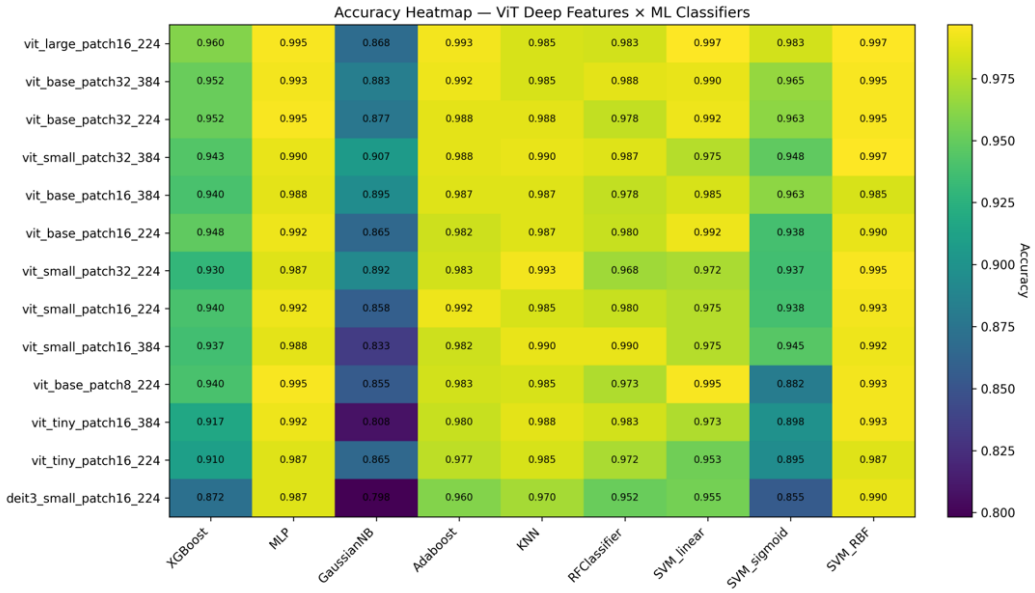

Figure 8. Accuracy heatmap showing the performance of different pre-trained ViT deep features combined with various ML classifiers on the preprocessed BT-large-2c dataset.

As shown in Tables 8–10, the precision, recall, and F1-scores are uniformly high, with SVMs (RBF and linear), XGBoost, KNN, and Random Forest consistently leading. GaussianNB is the clear underperformer across nearly all backbones. Performance varies only modestly across ViT variants (tiny/small/base, patch sizes, and input resolutions), indicating that classifier choice exerts a greater influence on these metrics than the specific ViT feature extractor.

Table 8. Precision (%) of pre-trained ViT variants evaluated on the preprocessed BT-large-2c dataset using ML classifiers with fine-tuned hyperparameters.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Precision								
	XGBoost	**MLP**	**GaussianNB**	**Adaboost**	**KNN**	**RFClassifier**	**SVM_linear**	**SVM_sigmoid**	**SVM_RBF**
vit_base_patch16_224	0.9769	0.9900	0.8590	0.9801	0.9899	0.9800	0.9900	0.9550	0.9868
vit_base_patch32_224	0.9932	0.9933	0.9556	0.9966	0.9933	0.9965	0.9868	0.9728	0.9934
vit_large_patch16_224	0.9900	0.9933	0.9623	0.9901	0.9866	0.9866	0.9967	0.9966	1.0000
vit_small_patch32_224	0.9833	0.9834	0.9304	0.9801	0.9966	0.9668	0.9732	0.9226	0.9967
deit3_small_patch16_224	0.9655	0.9933	0.7788	0.9631	0.9829	0.9562	0.9627	0.8403	0.9933
vit_base_patch8_224	0.9832	0.9967	0.9112	0.9833	0.9899	0.9830	0.9967	0.8908	0.9966
vit_tiny_patch16_224	0.9732	0.9868	0.8687	0.9704	0.9834	0.9732	0.9416	0.8911	0.9834
vit_small_patch16_224	0.9770	0.9933	0.8746	0.9900	0.9834	0.9800	0.9831	0.9550	0.9966
vit_base_patch16_384	0.9737	0.9833	0.9187	0.9771	0.9834	0.9799	0.9866	0.9760	0.9866
vit_tiny_patch16_384	0.9867	0.9868	0.8201	0.9768	0.9835	0.9770	0.9733	0.9193	0.9966
vit_small_patch32_384	0.9867	0.9900	0.8789	0.9867	0.9933	0.9803	0.9766	0.9381	1.0000
vit_small_patch16_384	0.9834	0.9967	0.8425	0.9898	0.9868	0.9933	0.9766	0.9435	0.9966
vit_base_patch32_384	0.9899	0.9901	0.9197	0.9900	0.9866	0.9835	0.9836	0.9697	0.9934

The heatmap in Figure 9 illustrates the classification accuracies obtained using ViT-derived deep features in combination with various ML classifiers. Overall, most ViT–classifier pairs achieve consistently high performance, with SVM (RBF) and MLP emerging as the most effective classifiers across different feature sets. Among the feature extractors, models such as vit_base_patch16_224 and vit_small_patch16_224 consistently deliver the best results, often exceeding 0.97 in accuracy, while deit3_small_patch16_224 performs comparatively weakly, with accuracies falling below 0.80 in some cases. These results demonstrate that both the choice of ViT architecture and classifier significantly

influence performance, with deeper ViT models and robust classifiers providing superior discriminability for the datasets tested.

Table 9. Recall (%) of pre-trained ViT variants evaluated on the preprocessed BT-large-2c dataset using ML classifiers with fine-tuned hyperparameters.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Recall								
	XGBoost	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF
vit_base_patch16_224	0.9867	0.9900	0.8733	0.9833	0.9833	0.9800	0.9933	0.9200	0.9933
vit_base_patch32_224	0.9733	0.9933	0.7900	0.9800	0.9833	0.9600	0.9967	0.9533	0.9967
vit_large_patch16_224	0.9933	0.9933	0.7667	0.9967	0.9833	0.9800	0.9967	0.9700	0.9933
vit_small_patch32_224	0.9800	0.9867	0.8467	0.9867	0.9900	0.9700	0.9700	0.9533	0.9933
deit3_small_patch16_224	0.9333	0.9900	0.8333	0.9567	0.9567	0.9467	0.9467	0.8767	0.9867
vit_base_patch8_224	0.9733	0.9933	0.7867	0.9833	0.9800	0.9633	0.9933	0.8700	0.9900
vit_tiny_patch16_224	0.9700	0.9933	0.8600	0.9833	0.9867	0.9700	0.9667	0.9000	0.9900
vit_small_patch16_224	0.9900	0.9900	0.8367	0.9933	0.9867	0.9800	0.9667	0.9200	0.9900
vit_base_patch16_384	0.9867	0.9833	0.8667	0.9967	0.9900	0.9767	0.9833	0.9500	0.9833
vit_tiny_patch16_384	0.9900	0.9967	0.7900	0.9833	0.9933	0.9900	0.9733	0.8733	0.9900
vit_small_patch32_384	0.9867	0.9900	0.9433	0.9900	0.9867	0.9933	0.9733	0.9600	0.9933
vit_small_patch16_384	0.9867	0.9967	0.8200	0.9733	0.9933	0.9867	0.9733	0.9467	0.9867
vit_base_patch32_384	0.9800	1.0000	0.8400	0.9933	0.9833	0.9933	0.9967	0.9600	0.9967

Table 10. F1-Score (%) of pre-trained ViT variants evaluated on the preprocessed BT-large-2c dataset using ML classifiers with fine-tuned hyperparameters.

Deep Feature from the Pre-Trained ViT Model	ML Classifier F1-Score								
	XGBoost	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF
vit_base_patch16_224	0.9818	0.9900	0.8661	0.9817	0.9866	0.9800	0.9917	0.9372	0.9900
vit_base_patch32_224	0.9832	0.9933	0.8650	0.9882	0.9883	0.9779	0.9917	0.9630	0.9950
vit_large_patch16_224	0.9917	0.9933	0.8534	0.9934	0.9850	0.9833	0.9967	0.9831	0.9967
vit_small_patch32_224	0.9816	0.9850	0.8866	0.9834	0.9933	0.9684	0.9716	0.9377	0.9950
deit3_small_patch16_224	0.9492	0.9917	0.8052	0.9599	0.9696	0.9514	0.9546	0.8581	0.9900
vit_base_patch8_224	0.9782	0.9950	0.8444	0.9833	0.9849	0.9731	0.9950	0.8803	0.9933
vit_tiny_patch16_224	0.9716	0.9900	0.8643	0.9768	0.9850	0.9716	0.9539	0.8955	0.9867
vit_small_patch16_224	0.9834	0.9917	0.8552	0.9917	0.9850	0.9800	0.9748	0.9372	0.9933
vit_base_patch16_384	0.9801	0.9833	0.8919	0.9868	0.9867	0.9783	0.9850	0.9628	0.9850
vit_tiny_patch16_384	0.9884	0.9917	0.8048	0.9801	0.9884	0.9834	0.9733	0.8957	0.9933
vit_small_patch32_384	0.9867	0.9900	0.9100	0.9884	0.9900	0.9868	0.9750	0.9489	0.9967
vit_small_patch16_384	0.9850	0.9967	0.8311	0.9815	0.9900	0.9900	0.9750	0.9451	0.9916
vit_base_patch32_384	0.9849	0.9950	0.8780	0.9917	0.9850	0.9884	0.9901	0.9648	0.9950

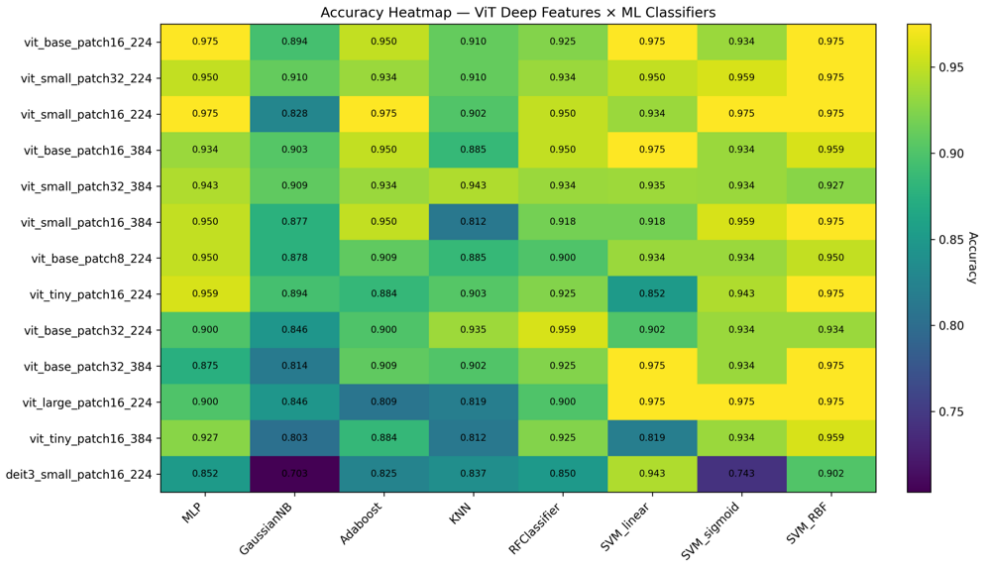

Figure 9. Heatmap of classification accuracies obtained from different ViT feature extractors combined with various ML classifiers. Overall, vit_base_patch16_224 and vit_small_patch16_224 paired with SVM (RBF) and MLP yielded the highest accuracies ($\geq$0.97), while deit3_small_patch16_224 showed comparatively weaker performance.

4. Experimental Setup

The experiments in this study were designed to evaluate the performance of the proposed hybrid approach for brain tumor classification. Details of the experimental setup are provided in the respective subsections.

4.1. Implementation Details

In this study, we utilized 13 pre-trained ViT-based models as feature extractors, as described in Section 3.5. These models, pre-trained on the ImageNet dataset [67], were adapted for our task by freezing the weights of their bottleneck layers. This approach preserves the learned features while preventing overfitting to our comparatively smaller dataset. Additionally, we incorporated nine distinct ML classifiers, detailed in Section 3.6, to evaluate the extracted features. The combined use of diverse feature extraction techniques and classifiers ensures robust and comprehensive analysis of the MRI datasets. Before training, all input images underwent preprocessing steps as outlined in Section 3.4. These steps included image cropping, resizing, and augmentation to enhance the dataset's quality and diversity, ultimately improving model performance.

All experiments were conducted using an NVIDIA RTX 3090 GPU, ensuring efficient processing and the ability to handle the computational demands of training DL models and performing extensive experimentation.

4.1.1. Deep Feature Evaluation and Selection

In this study, we evaluated the deep features extracted from 13 pre-trained ViT models using various ML classifiers, as outlined in Section 3.6. The goal was to identify the top-performing deep features for each of the two MRI datasets based on their classification accuracy.

4.1.2. Evaluation Criteria

The evaluation process involved calculating the average Accuracy, Precision, Recall, and F1-score achieved by each ViT model across nine different ML classifiers. This approach ensured a comprehensive assessment of the models' performance across a diverse set of classifiers, providing a robust basis for selection. To resolve ties where two or more models achieved identical accuracy, we prioritized models with lower standard deviation. A lower standard deviation indicates more consistent performance across different subsets of the data, which is critical for reliable classification.

4.1.3. Rationale for Selection

The primary reason for selecting only the top three deep features is to minimize redundancy and enhance diversity within the feature ensemble. Deep features extracted from closely related models often occupy overlapping feature spaces, which can lead to diminished ensemble performance due to a lack of variability. By focusing on the top three models with distinct and reliable performance, we ensure the ensemble leverages complementary feature representations. The top three deep features are then utilized in our ensemble module, which is detailed in the subsequent subsection.

4.1.4. Ensemble of Deep Features

Ensemble learning is a powerful technique aimed at improving model performance while reducing the risks associated with relying on a single feature or model. By combining multiple features from various models into a unified predictive feature set, ensemble learning enhances the robustness and accuracy of classification tasks. This technique can be broadly classified into two categories, feature ensembles and classifier ensembles, depending on the level at which the integration occurs.

4.2. Feature Ensemble vs. Classifier Ensemble

In the feature ensemble approach, feature sets extracted from different models are combined into a single unified feature sequence. This integrated feature set is then fed into a classifier to produce the final prediction. The feature ensemble approach leverages the rich, diverse information embedded in the feature sets, offering a more detailed representation of the input data (e.g., MR images) compared to individual classifier outputs. Whereas, in the classifier ensemble approach, it combines the predictions from multiple classifiers, using techniques such as majority voting or weighted voting to determine the final output.

4.3. Confidence Interval Estimation

To account for variability across runs and ensure statistical reliability, we repeated all experiments using five different random seeds. For each model classifier configuration, we computed the mean accuracy and the margin of error (MoE) at a 95% confidence level. The MoE was calculated based on the standard error of the mean using the Student's t-distribution, following:

$$\text{MoE} = t_{\alpha/2,\, n-1} \times \frac{s}{\sqrt{n}} \tag{16}$$

where s is the sample standard deviation across n runs, and $t_{\alpha/2,\, n-1}$ is the critical value from the t-distribution. Final results are reported as $mean \pm MoE$, which provides a confidence interval indicating that the true mean accuracy lies within this range with 95% confidence.

Table 11 presents the accuracies (mean ± Margin of Error (MoE)) of different pre-trained ViT backbones combined with multiple ML classifiers on the BT-large-2c dataset. The reported values include both the mean accuracy and the margin of error at a 95% confidence level, indicating performance stability across repeated runs. Overall, SVM with RBF and linear kernels, MLP, KNN, and Random Forest yield the highest and most consistent accuracies, often above 0.98, while GaussianNB performs noticeably worse across all backbones.

Table 11. Accuracies (mean ± MoE) of pre-trained ViT models with ML classifiers on the BT-large-2c dataset.

ViT Feature	Adaboost	GaussianNB	KNN	MLP	RFClassifier	SVM_RBF	SVM_linear	SVM_sigmoid
deit3_small_patch16_224	0.9600 ± 0.0158	0.7983 ± 0.0308	0.9700 ± 0.0142	0.9917 ± 0.0067	0.9517 ± 0.0167	0.9900 ± 0.0083	0.9550 ± 0.0167	0.8550 ± 0.0275
vit_base_patch16_224	0.9817 ± 0.0108	0.8650 ± 0.0250	0.9867 ± 0.0092	0.9900 ± 0.0075	0.9800 ± 0.0100	0.9900 ± 0.0075	0.9917 ± 0.0067	0.9383 ± 0.0192
vit_base_patch16_384	0.9867 ± 0.0092	0.8950 ± 0.0250	0.9867 ± 0.0092	0.9833 ± 0.0108	0.9783 ± 0.0117	0.9850 ± 0.0092	0.9850 ± 0.0100	0.9633 ± 0.0158
vit_base_patch32_224	0.9883 ± 0.0100	0.8767 ± 0.0267	0.9783 ± 0.0142	0.9950 ± 0.0058	0.9917 ± 0.0067	0.9950 ± 0.0050	0.9917 ± 0.0067	0.9633 ± 0.0158
vit_base_patch32_384	0.9917 ± 0.0067	0.8833 ± 0.0283	0.9850 ± 0.0108	0.9933 ± 0.0058	0.9883 ± 0.0083	0.9950 ± 0.0050	0.9900 ± 0.0083	0.9650 ± 0.0175
vit_large_patch16_224	0.9933 ± 0.0067	0.8683 ± 0.0250	0.9850 ± 0.0125	0.9950 ± 0.0058	0.9833 ± 0.0125	0.9967 ± 0.0042	0.9967 ± 0.0042	0.9833 ± 0.0158
vit_small_patch16_224	0.9917 ± 0.0067	0.8583 ± 0.0275	0.9850 ± 0.0125	0.9917 ± 0.0067	0.9800 ± 0.0100	0.9933 ± 0.0058	0.9750 ± 0.0125	0.9833 ± 0.0158
vit_small_patch16_384	0.9817 ± 0.0125	0.8333 ± 0.0292	0.9900 ± 0.0083	0.9883 ± 0.0083	0.9900 ± 0.0083	0.9917 ± 0.0067	0.9750 ± 0.0125	0.9483 ± 0.0208
vit_small_patch32_224	0.9833 ± 0.0117	0.8917 ± 0.0292	0.9933 ± 0.0067	0.9867 ± 0.0092	0.9683 ± 0.0167	0.9950 ± 0.0050	0.9717 ± 0.0142	0.9367 ± 0.0217
vit_small_patch32_384	0.9883 ± 0.0100	0.9067 ± 0.0258	0.9900 ± 0.0083	0.9900 ± 0.0100	0.9867 ± 0.0108	0.9967 ± 0.0042	0.9750 ± 0.0125	0.9483 ± 0.0208
vit_tiny_patch16_224	0.9767 ± 0.0108	0.8650 ± 0.0250	0.9850 ± 0.0125	0.9867 ± 0.0092	0.9717 ± 0.0158	0.9867 ± 0.0092	0.9533 ± 0.0183	0.8950 ± 0.0225
vit_tiny_patch16_384	0.9800 ± 0.0125	0.8083 ± 0.0317	0.9883 ± 0.0092	0.9917 ± 0.0067	0.9833 ± 0.0108	0.9933 ± 0.0058	0.9733 ± 0.0167	0.8983 ± 0.0250

4.4. Feature Ensemble methodology

In the feature ensemble approach, we integrated deep features extracted from the top-2 and top-3 distinct pre-trained ViT models. This integration was achieved by concatenating the feature vectors into a single unified feature sequence. For instance, if vit_base_patch_16_224, vit_small_patch_32_224, and vit_small_patch_16_224 are identified as the top-performing models, their respective feature sets are concatenated to form a comprehensive feature vector. This combined feature set is then input into various ML classifiers to predict the final output.

To assess the impact of combining multiple feature sets, we conducted a comparative analysis, such as two-feature Combinations. We explored all possible pairwise combinations of deep features extracted from the top three models. Each pair was fed into the classifiers to evaluate performance. As well as, three-Feature Ensemble. We integrated deep features from all three top-performing models and input the resulting concatenated feature vector into the classifiers.

This comparative analysis allowed us to evaluate the relative effectiveness of the three-feature ensemble against pairwise combinations, providing insights into the optimal configuration for robust classification.

4.4.1. Classifier Ensemble Methodology

In addition to feature-level integration, we also explored classifier ensembles by combining the predictions from multiple classifiers. Using techniques such as majority voting, the outputs of individual classifiers were aggregated to determine the final prediction. This approach provides an additional layer of robustness by leveraging the strengths of multiple classifiers.

4.4.2. Insights and Advantages

The feature ensemble approach ensures that the classifier receives a comprehensive and detailed input, enhancing its ability to make accurate predictions. By combining features from distinct models, the ensemble minimizes redundancy and captures diverse aspects of the input data.The combined use of feature and classifier ensembles enhances the overall reliability of the system, addressing potential weaknesses in individual models or classifiers.

The ensemble strategies employed in this study were integral to achieving superior performance in brain tumor classification tasks, as detailed in the experimental results.

5. Experimental Results

The experimental results were derived from two publicly available datasets for brain tumor classification tasks, referred to as BT-small-2c and BT-large-2c. The experiments were conducted to evaluate the effectiveness of the proposed hybrid approach using a systematic methodology. The study was divided into three distinct experiments, each targeting a specific aspect of the classification task.

5.1. Experiment 1: Hyperparameter Tuning of ML Classifiers

The first experiment focused on optimizing the performance of the nine ML classifiers employed in this study as illustrated in Table 3. Hyperparameter tuning was conducted for each classifier to enhance its ability to classify brain tumors accurately. This process involved adjusting key parameters for each ML model, such as the number of neighbors in KNN, the number of estimators in certain ensemble methods (e.g., RF and AdaBoost), and the learning rate in gradient-boosting classifiers. The tuning hyperparameters significantly improved the classifiers' performance. For instance, in KNN, the optimal value of k was determined to achieve the best balance between accuracy and computational efficiency. Whereas, in RF, the number of estimators and the maximum depth of trees were adjusted to maximize classification accuracy. In AdaBoost, the learning rate and the number of weak classifiers were fine-tuned for optimal results. The accuracy for each classifier after tuning was recorded. These results formed the baseline for subsequent experiments.

5.2. Experiment 2: Using an Ensemble of Deep Features with ML Classifiers

The second experiment aimed to demonstrate the advantages of combining deep features extracted from the top two or three pre-trained ViT models with various ML classifiers. The top-performing deep features were identified in the feature evaluation stage based on their average accuracy across the tuned ML classifiers.

5.2.1. Feature Ensemble Creation

Deep features from the top two ViT models were concatenated into a single feature vector and fed into each ML classifier. Similarly, deep features from the top three models were combined and input into the classifiers.

5.2.2. Performance Comparison

The classifiers were evaluated using the combined feature sets, and their performance was compared to that achieved using individual deep features. Results showed that the feature ensembles provided a richer representation of the input data, leading to improved classification accuracy across most ML classifiers.

5.3. Experiment 3: Using an Ensemble of ML Classifiers with Various Preprocessing Strategies

The final experiment explored using an ensemble of ML classifiers to further enhance classification performance. This involved combining the predictions of the top two and top three ML classifiers using various preprocessing strategies.

1. Enhanced Ensemble: Predictions from the top classifiers were combined using a simple majority voting mechanism. The optimal DL feature extractors and ML classifiers demonstrate dataset-specific performance characteristics. As illustrated in Table 12, on the BT-small-2c dataset, the ensemble of "vit base patch16 224 + vit small patch32 224" in conjunction with an MLP classifier achieves a notable accuracy of 0.9589. In contrast, as shown in Table 13, the BT-large-2c dataset benefits most from the combination of "vit large patch16 224 + vit base patch32 384" also paired with an MLP classifier, resulting in a superior accuracy of 0.9983. These results underscore the importance of tailoring the feature extraction and classification approach to the specific characteristics of the dataset to maximize performance.

Table 12. Accuracies of the top three pre-trained ViT models using a simple version on the BT-small-2c dataset. The table compares different deep features from the pre-trained ViT models in combination with various classification algorithms.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy							
	MLP	**GaussianNB**	**Adaboost**	**KNN**	**RFClassifier**	**SVM_linear**	**SVM_sigmoid**	**SVM_RBF**
vit_base_patch16_224 + vit_small_patch32_224	0.9589	0.6129	0.9250	0.9177	0.9250	0.9750	0.9589	0.9750
vit_base_patch16_224 + vit_small_patch16_224	0.9750	0.5968	0.8266	0.8694	0.8589	0.9750	0.9750	0.9750
vit_small_patch32_224 + vit_small_patch16_224	0.9750	0.7427	0.9177	0.8782	0.9500	0.9750	0.9589	0.9750
vit_base_patch16_224 + vit_small_patch32_224 + vit_small_patch16_224	0.9750	0.5427	0.9589	0.9266	0.8782	0.9750	0.9750	0.9750

Table 13. Accuracies of top three pre-trained ViT models using a simple version on the BT-large-2c dataset.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy							
	MLP	**GaussianNB**	**Adaboost**	**KNN**	**RFClassifier**	**SVM_linear**	**SVM_sigmoid**	**SVM_RBF**
vit_large_patch16_224 + vit_base_patch32_384	0.9983	0.6700	0.9900	0.9883	0.9900	0.9967	0.9700	0.9967
vit_large_patch16_224 + vit_base_patch32_224	0.9983	0.6550	0.9883	0.9917	0.9717	0.9983	0.9800	0.9967
vit_base_patch32_384 + vit_base_patch32_224	0.9950	0.6983	0.9917	0.9900	0.9750	0.9950	0.9800	0.9967
vit_large_patch16_224 + vit_base_patch32_384 + vit_base_patch32_224	0.9950	0.6683	0.9883	0.9900	0.9833	0.9983	0.9867	0.9967

2. Ensemble with Enhanced Processing: The following preprocessing techniques were applied before creating the ensemble.

Normalization: Feature scaling was applied to ensure that all input features contributed equally to the classification task.

Principal Component Analysis (PCA): Dimensionality reduction was performed to remove redundancy and highlight the most critical features.

The performance of DL feature extractors and ML classifiers varies significantly depending on the dataset and preprocessing techniques employed. When normalization and PCA are applied, specific combinations of ViT models and classifiers yield high accuracies. For instance, on the BT-small-2c dataset as depicted in Table 14, combining 'vit_base_patch16_224 + vit_small_patch32_224' with an MLP classifier achieves an accuracy of 0.9339. On the other hand, as shown in Table 15, the combination of "vit_large_patch16_224 + vit_base_patch32_384" paired with an MLP classifier achieves an accuracy of 0.9967, and 0.9983 with an SVM linear classifier. In comparison, when using the simple version of features, as shown in Table 12, similar combinations also perform well, but without normalization and PCA preprocessing, the "vit_base_patch16_224 + vit_small_patch32_224" paired with an MLP classifier on the BT-small-2c dataset achieves 0.9589 accuracy, and the combination of "vit_large_patch16_224 + vit_base_patch32_384" paired with an MLP classifier on the BT-large-2c dataset reaches 0.9983 as shown in Table 13. These results highlight the sensitivity of model performance to dataset characteristics and the importance of appropriate preprocessing strategies.

Table 14. Classification accuracies of the top three pre-trained ViT models using normalization and PCA for deep feature extraction, evaluated with various ML classifiers on the BT-small-2c dataset.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy							
	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF
vit_base_patch16_224 + vit_small_patch32_224	0.9339	0.6129	0.9589	0.9177	0.8177	0.9750	0.9589	0.9750
vit_base_patch16_224 + vit_small_patch16_224	0.9750	0.5645	0.8516	0.8694	0.7750	0.9750	0.9750	0.9750
vit_small_patch32_224 + vit_small_patch16_224	0.9500	0.7105	0.9177	0.8782	0.9339	0.9750	0.9589	0.9750
vit_base_patch16_224 + vit_small_patch32_224 + vit_small_patch16_224	0.9750	0.5427	0.9589	0.9266	0.8782	0.9750	0.9750	0.9750

Table 15. Classification accuracies of the top three pre-trained ViT models using normalization and PCA for deep feature extraction, evaluated with various ML classifiers on BT-large-2c dataset.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy							
	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF
vit_large_patch16_224 + vit_base_patch32_384	0.9967	0.6667	0.9917	0.9900	0.9883	0.9967	0.9717	0.9967
vit_large_patch16_224 + vit_base_patch32_224	0.9983	0.6550	0.9850	0.9917	0.9867	0.9983	0.9783	0.9967
vit_base_patch32_384 + vit_base_patch32_224	0.9983	0.6983	0.9933	0.9900	0.9833	0.9933	0.9817	0.9967
vit_large_patch16_224 + vit_base_patch32_384 + vit_base_patch32_224	0.9933	0.6600	0.9900	0.9900	0.9817	0.9983	0.9867	0.9967

Synthetic Minority Oversampling TEchnique (SMOTE): To address class imbalance, synthetic samples were generated for underrepresented classes. When normalization and PCA are applied, specific combinations of ViT models and classifiers yield high accuracies, though the use of SMOTE may achieve even greater accuracies. For instance, as we can observe from Table 15 on the BT-large-2c dataset, the combination of "vit_large_patch16_224 + vit_base_patch32_384" paired with an MLP classifier reaches an accuracy of 0.9967, and 0.9983 accuracy with SVM linear. When using SMOTE, as shown in Table 16, the combination of 'vit_base_patch16_224 + vit_small_patch32_224' combined with MLP reaches an accuracy of 0.9750 on the BT-small-2c dataset, while 'vit_large_patch16_224 + vit_base_patch32_224' combined with MLP reaches 0.9983 on the BT-large-2c dataset as depicted in Table 17. These results underscore the importance of carefully selecting preprocessing methods and model combinations to maximize performance for specific datasets.

Table 16. Accuracies of the top three pre-trained DL models using smote data, evaluated with various ML classifiers, on BT-small-2c dataset.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy							
	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF
vit_base_patch16_224 + vit_small_patch32_224	0.9750	0.6129	0.9089	0.9177	0.9500	0.9750	0.9589	0.9750
vit_base_patch16_224 + vit_small_patch16_224	0.9750	0.5645	0.8427	0.8694	0.8000	0.9750	0.9750	0.9750
vit_small_patch32_224 + vit_small_patch16_224	0.9750	0.7427	0.8927	0.8782	0.8839	0.9750	0.9589	0.9750
vit_base_patch16_224 + vit_small_patch32_224 + vit_small_patch16_224	0.9750	0.5427	0.9589	0.9266	0.8782	0.9750	0.9750	0.9750

Table 17. Accuracies of the top three pre-trained DL models using SMOTE only for BT-large 2c dataset.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy							
	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF
vit_large_patch16_224 + vit_base_patch32_384	0.9967	0.6717	0.9883	0.9883	0.9883	0.9967	0.9750	0.9967
vit_large_patch16_224 + vit_base_patch32_224	0.9983	0.6483	0.9850	0.9917	0.9833	0.9983	0.9783	0.9967
vit_base_patch32_384 + vit_base_patch32_224	0.9967	0.6933	0.9900	0.9900	0.9767	0.9950	0.9817	0.9967
vit_large_patch16_224 + vit_base_patch32_384 + vit_base_patch32_224	0.9983	0.6633	0.9900	0.9900	0.9867	0.9983	0.9833	0.9967

3. Combinations of Preprocessing: The classifiers were ensembled under different combinations of preprocessing techniques, such as Normalization and PCA, SMOTE only, and a combination of normalization, PCA, and SMOTE. The ensemble with normalization, PCA, and SMOTE provided the best results by balancing feature scaling, dimensionality reduction, and class balance.

These experiments demonstrate that combining deep feature ensembles with ML classifier ensembles, supported by effective preprocessing, significantly improves the accuracy and robustness of brain tumor classification. The results highlight the value of systematic feature integration and classifier combination in addressing complex medical image analysis tasks.

Overall, the accuracies achieved on the BT-large-2c dataset are generally higher than those on the BT-small-2c dataset, suggesting that the models benefit from the larger dataset or the inherent characteristics of the BT-large-2c dataset. For instance, looking at Table 18 'vit base patch16 224 + vit small patch32 224' with an MLP classifier achieves 0.9750 accuracy on the BT-small-2c dataset. In comparison, in Table 19, several combinations on the BT-large-2c dataset reach accuracies above 0.99, indicating superior performance on the latter.

We also created ensembles of ML classifiers as shown in Tables 20–27 to investigate their combined performance in brain tumor classification. Ensemble methods, such as combining the top-performing classifiers based on fine-tuned hyperparameters, leverage the strengths of individual models to enhance overall accuracy and robustness. By aggregating predictions from multiple classifiers, ensembles mitigate the limitations of any single model, leading to improved generalization and reliability. Our results demonstrate that an ensemble of classifiers, particularly those incorporating SVM with RBF kernel, RF, and MLP, consistently outperformed individual classifiers in terms of accuracy and stability. This approach further validates the efficacy of combining complementary decision-making processes for precise brain tumor classification. Various combinations of classifiers like MLP, SVM (with sigmoid and RBF kernels), and KNN were evaluated. The tables reveal that ML classifier ensembles often achieve higher accuracy than individual classifiers.

Table 18. Classification accuracies of the top three pre-trained DL models using normalization, PCA, and SMOTE on the BT-small-2c dataset, with various ML classifiers.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy							
	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF
vit_base_patch16_224 + vit_small_patch32_224	0.9750	0.6129	0.9250	0.9016	0.8750	0.9750	0.9589	0.9750
vit_base_patch16_224 + vit_small_patch16_224	0.9500	0.5806	0.8589	0.8694	0.8000	0.9750	0.9750	0.9750
vit_small_patch32_224 + vit_small_patch16_224	0.9339	0.7927	0.9000	0.8782	0.8839	0.9750	0.9589	0.9750
vit_base_patch16_224 + vit_small_patch32_224 + vit_small_patch16_224	0.9589	0.5427	0.9589	0.9266	0.8782	0.9750	0.9750	0.9750

Table 19. Classification accuracies of the top three pre-trained DL models using normalization, PCA, and SMOTE on the BT-large-2c dataset, with various ML classifiers.

Deep Feature from the Pre-Trained ViT Model	ML Classifier Accuracy							
	MLP	GaussianNB	Adaboost	KNN	RFClassifier	SVM_linear	SVM_sigmoid	SVM_RBF
vit_large_patch16_224 + vit_base_patch32_384	0.9967	0.6750	0.9883	0.9883	0.9817	0.9967	0.9750	0.9967
vit_large_patch16_224 + vit_base_patch32_224	0.9967	0.6550	0.9817	0.9917	0.9817	0.9983	0.9817	0.9967
vit_base_patch32_384 + vit_base_patch32_224	0.9967	0.6933	0.9850	0.9900	0.9817	0.9950	0.9817	0.9967
vit_large_patch16_224 + vit_base_patch32_384 + vit_base_patch32_224	0.9917	0.6700	0.9900	0.9900	0.9817	1.0000	0.9867	0.9967

Table 20. Accuracies of the top three ensembled ML classifiers combined with different pre-trained ViT models using the simple version on the BT-small-2c dataset.

ML Ensembling	Pre-Trained DL Models Ensembling				
	vit_base_patch16_224	vit_small_patch32_224	vit_small_patch16_224	vit_base_patch16_384	vit_small_patch32_384
MLP + SVM_sigmoid	0.9800	0.9483	0.9550	0.9783	0.9417
MLP + SVM_RBF	0.9883	0.9883	0.9950	0.9850	0.9950
SVM_sigmoid + SVM_RBF	0.9783	0.9533	0.9550	0.9767	0.9433
MLP + SVM_sigmoid + SVM_RBF	0.9883	0.9883	0.9900	0.9883	0.9900

Table 21. Accuracies of top 3 ensembled ML classifiers using simple version on BT-large-2c dataset.

ML Ensembling	Pre-Trained DL Models Ensembling				
	vit_large_patch16_224	vit_base_patch32_384	vit_base_patch32_224	vit_small_patch32_384	vit_base_patch16_384
KNN + MLP	0.9900	0.9917	0.9917	0.9900	0.9833
KNN + SVM_RBF	0.9917	0.9933	0.9933	0.9950	0.9850
MLP + SVM_RBF	0.9950	0.9900	0.9950	0.9900	0.9883
KNN + MLP + SVM_RBF	0.9950	0.9917	0.9950	0.9950	0.9900

Table 22. Accuracies of top 3 ensembled ML classifiers with normalization and PCA on BT-small-2c dataset.

ML Ensembling	Pre-Trained DL Models Ensembling				
	vit_base_patch16_224	vit_small_patch32_224	vit_small_patch16_224	vit_base_patch16_384	vit_small_patch32_384
MLP + SVM_sigmoid	0.9817	0.9467	0.9617	0.9767	0.9367
MLP + SVM_RBF	0.9883	0.995	0.9933	0.9883	0.9917
SVM_sigmoid + SVM_RBF	0.9817	0.9517	0.9617	0.9767	0.94
MLP + SVM_sigmoid + SVM_RBF	0.99	0.9917	0.995	0.9883	0.995

Table 23. Accuracies of top 3 ensembled ML classifiers with normalization and PCA on BT-large-2c dataset.

ML Ensembling	Pre-Trained DL Models Ensembling				
	vit_large_patch16_224	vit_base_patch32_384	vit_base_patch32_224	vit_small_patch32_384	vit_base_patch16_384
KNN + MLP	0.9917	0.9950	0.9917	0.9883	0.9850
KNN + SVM_RBF	0.9917	0.9933	0.9933	0.9950	0.9850
MLP + SVM_RBF	0.9950	0.9917	0.9950	0.9967	0.9900
KNN + MLP + SVM_RBF	0.9967	0.9933	0.9933	0.9950	0.9900

Table 24. Accuracies of top 3 ensembled ML classifiers with SMOTE only on BT-small-2c dataset.

ML Ensembling	Pre-Trained DL Models Ensembling				
	vit_base_patch16_224	vit_small_patch32_224	vit_small_patch16_224	vit_base_patch16_384	vit_small_patch32_384
MLP + SVM_sigmoid	0.9817	0.945	0.955	0.9733	0.94
MLP + SVM_RBF	0.9883	0.9917	0.9933	0.99	0.9883
SVM_sigmoid + SVM_RBF	0.9817	0.95	0.9567	0.975	0.9467
MLP + SVM_sigmoid + SVM_RBF	0.9883	0.99	0.9917	0.9883	0.9917

Table 25. Accuracies of top 3 ensembled ML classifiers using SMOTE only on BT-large-2c dataset.

ML Ensembling	Pre-Trained DL Models Ensembling				
	vit_large_patch16_224	vit_base_patch32_384	vit_base_patch32_224	vit_small_patch32_384	vit_base_patch16_384
KNN + MLP	0.9900	0.9950	0.9950	0.9883	0.9883
KNN + SVM_RBF	0.9900	0.9933	0.9933	0.9950	0.9850
MLP + SVM_RBF	0.9933	0.9950	0.9933	0.9950	0.9900
KNN + MLP + SVM_RBF	0.9967	0.9917	0.9950	0.9950	0.9883

Table 26. Accuracies of top 3 ensembled ML classifiers with normalization, PCA, and SMOTE on BT-small-2c dataset.

ML Ensembling	Pre-Trained DL Models Ensembling				
	vit_base_patch16_224	vit_small_patch32_224	vit_small_patch16_224	vit_base_patch16_384	vit_small_patch32_384
MLP + SVM_sigmoid	0.9817	0.9517	0.9550	0.9733	0.9417
MLP + SVM_RBF	0.9883	0.9950	0.9950	0.9883	0.9933
SVM_sigmoid + SVM_RBF	0.9817	0.9517	0.9567	0.9750	0.9433
MLP + SVM_sigmoid + SVM_RBF	0.9900	0.9917	0.9950	0.9867	0.9900

Table 27. Accuracies of top 3 ensembled ML classifiers with normalization, PCA, and SMOTE on BT-large-2c dataset.

ML Ensembling	Pre-Trained DL Models Ensembling				
	vit_large_patch16_224	vit_base_patch32_384	vit_base_patch32_224	vit_small_patch32_384	vit_base_patch16_384
KNN + MLP	0.9917	0.9950	0.9967	0.9917	0.9883
KNN + SVM_RBF	0.9917	0.9933	0.9933	0.9950	0.9850
MLP + SVM_RBF	0.9950	0.9917	0.9933	0.9950	0.9883
KNN + MLP + SVM_RBF	0.9967	0.9933	0.9967	0.9950	0.9883

Notably, the use of ML classifier ensembles frequently leads to improved accuracy compared to relying on single classifiers. For example, the combination of 'MLP + SVM sigmoid + SVM RBF' often appears among the top performers, indicating that combining the strengths of multiple classifiers can yield more robust and accurate results than relying on a single classifier alone. Specifically, this ensemble achieves 0.9950 accuracy on the BT-large-2c dataset with 'vit_large_patch16_224', and it reaches 0.9917 accuracy on the BT-small-2c dataset with 'vit_small_patch32_224'.

5.4. Impact of Preprocessing on Classification Performance

Tables 5 and 6 present a comparison of the classification accuracies achieved by pre-trained ViT models with fine-tuned hyperparameters of ML classifiers on preprocessed and non-preprocessed BT-large-2c datasets. A detailed analysis reveals that preprocessing the dataset significantly improves classification performance across all ML classifiers.

In Table 5, which shows results for the non-preprocessed dataset, the average highest classification accuracy is 0.9789, with specific classifiers like MLP achieving higher accuracies. However, the overall consistency and peak performance are limited by the lack of preprocessing, which may lead to noise and irrelevant features influencing the results. Conversely, in Table 6, the results for the preprocessed dataset show an average highest accuracy of 0.9908. Preprocessing eliminates irrelevant features and normalizes the input data, allowing classifiers to focus on the most discriminative features. This is particularly

evident in classifiers such as MLP, SVM (RBF), and KNN, where preprocessing enhances performance significantly, ensuring higher accuracy and more stable results.

The advantage of preprocessing lies in its ability to refine the dataset, remove redundant information, and improve feature representation. This leads to better alignment with the capabilities of the DL and ML models, resulting in a marked improvement in classification accuracy. These results underscore the critical role of preprocessing in achieving optimal performance for brain tumor classification tasks.

6. Discussion

This study presents a comprehensive methodology for brain tumor classification, integrating advanced techniques in image preprocessing, augmentation, feature extraction, feature selection, and ML. The obtained results highlight the importance and efficacy of the proposed approach in addressing the complexities inherent in medical image analysis.

Image preprocessing, as described in Section 3.4, plays a pivotal role in enhancing the quality of brain MRI data. By removing noise and irrelevant regions, the preprocessing step ensures that only critical features are retained for subsequent analysis. This improvement in image quality directly contributes to the robustness and reliability of the classification outcomes. To evaluate the significance of preprocessing, an experiment was conducted using the BT-large-2c dataset without applying any preprocessing steps. The results, presented in Tables 5 and 6, clearly demonstrate a significant decline in classification accuracy when preprocessing is omitted. These findings underscore the necessity of preprocessing for achieving dependable and precise results.

The incorporation of image augmentation techniques further addresses the challenges posed by the small size of publicly available MRI datasets. By generating diverse training samples through techniques such as rotation and flipping, augmentation expands the dataset and mitigates overfitting. This extra diversity ensures that the ML models generalize well, even to unseen data, ultimately contributing to better classification performance.

Sophisticated feature extraction using pre-trained ViT models enables the capture of deep, high-level representations of brain MRIs. The evaluation and selection of these features ensure that only the most informative and consistent features are utilized. The focus on minimizing redundancy through careful feature selection further enhances the effectiveness of the models.

HPO was instrumental in refining the performance of each ML classifier. Fine-tuning parameters such as the number of neighbors in KNN, the number of estimators in ensemble models, and learning rates ensured that the classifiers were operating at their optimal capacity. This optimization process significantly improved the accuracy and efficiency of the models, as evidenced by the experimental results.

This study also highlights the advantages of ensemble strategies. Both feature-level ensembling and ML classifier ensembles contributed to substantial improvements in accuracy. Combining deep features from the top-performing ViT models provided a richer and more diverse representation of the input data, resulting in superior classification outcomes.

Integrating the predictions of multiple ML classifiers through majority voting and preprocessing techniques, such as normalization, PCA, and SMOTE, further improved the robustness and accuracy of the results.

The outcomes of this study have significant implications for clinical applications. The high accuracy and reliability of the proposed methodology underscores the potential of ML in assisting clinicians with brain tumor diagnosis. By providing automated and trustworthy classifications, the proposed approach can reduce the workload of radiologists and improve diagnostic precision, ultimately leading to better patient care and outcomes.

To further evaluate the practicality of our framework for clinical use, we analyzed the computational efficiency of both training and inference phases. While hyperparameter optimization via grid search is resource-intensive, it is a one-time process performed exclusively during training. Once optimal hyperparameters are determined, grid search is no longer required at inference.

Table 28 summarizes the computational requirements observed on an NVIDIA RTX 3090 GPU. As shown, the training phase with grid search required approximately 18–22 h for the entire dataset with a peak memory usage of ∼22 GB. In contrast, inference was highly efficient, requiring less than one second per scan (0.45–0.60 s) with a memory footprint of ∼3.5 GB. Even with a batch of 16 scans, inference completed in under 8 seconds, highlighting the scalability of the approach.

Table 28. Computational efficiency analysis of the proposed framework. Training with grid search, while computationally demanding, is performed only once. Inference remains efficient and suitable for clinical use.

Stage	Average Time per Scan	GPU Memory Usage	Notes
Training with grid search	∼18–22 h (entire dataset)	∼22 GB	One-time process; hyperparameter optimization only
Inference (single scan)	0.45–0.60 s	∼3.5 GB	Grid search not required; real-time feasible
Inference (batch of 16)	6–8 s	∼5.2 GB	Scales linearly with batch size

These results demonstrate that although training with grid search is computationally demanding, the deployment stage remains efficient and feasible for clinical workflows. Moreover, the framework can be adapted to resource-limited settings by selecting smaller ViT variants (e.g., ViT-Tiny, ViT-Small), reducing ensemble size, or applying model compression techniques such as pruning, quantization, or knowledge distillation. This scalability ensures that the proposed framework is suitable not only for high-performance research environments but also for real-world clinical practice where computational resources may be constrained.

Limitations and Future Work

While the proposed approach demonstrates exceptional performance, it is not without limitations. Publicly available datasets may not adequately represent the full range of variability observed in real-world clinical data. Future work could focus on validating the methodology using larger, more diverse datasets and exploring the integration of additional data modalities, such as clinical and genetic information, to further enhance classification accuracy and clinical relevance. Extending the architecture for other medical imaging tasks and exploring semi-supervised learning approaches could broaden its applicability while reducing reliance on fully labeled datasets. Lastly, testing the model in clinical settings and integrating it into cloud or edge computing platforms could facilitate its real-world implementation, particularly in resource-limited environments.

7. Conclusions

This study presents a novel double ensemble framework that combines the strengths of pre-trained DL models and fine-tuned ML classifiers for accurate brain tumor classification. By leveraging extensive preprocessing, data augmentation, and transfer learning with ViT networks, the proposed method effectively extracts and optimizes deep features from MRI scans. The use of feature-level and classifier-level ensembles further enhances classification accuracy, demonstrating the superiority of the proposed approach over SOTA methods. The results emphasize the critical role of HPO and preprocessing in improving diagnostic reliability and model performance. The integration of DL and ML techniques in this framework provides a robust, efficient, and scalable solution for brain tumor classification.

This work not only advances the field of medical image analysis but also highlights the potential of hybrid methodologies to deliver precise and trustworthy diagnostic outcomes, paving the way for practical clinical applications.

Author Contributions: Z.U.: Conceptualization, Data curation, Methodology, Software, Formal analysis, Investigation, Writing—original draft, Writing—review & editing. J.K.: Conceptualization, Writing—review and editing, Formal analysis, Investigation, Supervision, Project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT) of Korea under the ITRC (Information Technology Research Center) support program (IITP-2025-RS-2020-II201789), and the Global Research Support Program in the Digital Field (RS-2024-00426860), supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Zeng, L.; Zhang, H.H. Robust brain MRI image classification with SIBOW-SVM. *Comput. Med. Imaging Graph.* **2024**, *118*, 102451. [CrossRef]
2. da Costa Nascimento, J.J.; Marques, A.G.; do Nascimento Souza, L.; de Mattos Dourado, C.M.J.; da Silva Barros, A.C.; de Albuquerque, V.H.C.; de Freitas Sousa, L.F. A novel generative model for brain tumor detection using magnetic resonance imaging. *Comput. Med. Imaging Graph.* **2025**, *121*, 102498. [CrossRef]
3. Musthafa, M.M.; Kumar V, V.; Guluwadi, S. Enhancing brain tumor detection in MRI images through explainable AI using Grad-CAM with Resnet 50. *BMC Med. Imaging* **2024**, *24*, 107.
4. Lei, J.; Dai, L.; Jiang, H.; Wu, C.; Zhang, X.; Zhang, Y.; Yao, J.; Xie, W.; Zhang, Y.; Li, Y.; et al. Unibrain: Universal brain mri diagnosis with hierarchical knowledge-enhanced pre-training. *Comput. Med. Imaging Graph.* **2025**, *122*, 102516. [CrossRef] [PubMed]
5. Aygün, M.; Şahin, Y.H.; Ünal, G. Multi modal convolutional neural networks for brain tumor segmentation. *arXiv* **2018**, arXiv:1809.06191. [CrossRef]
6. Dehkordi, A.A.; Hashemi, M.; Neshat, M.; Mirjalili, S.; Sadiq, A.S. Brain tumor detection and classification using a new evolutionary convolutional neural network. *arXiv* **2022**, arXiv:2204.12297. [CrossRef]
7. Gundogan, E. A Novel Hybrid Deep Learning Model Enhanced with Explainable AI for Brain Tumor Multi-Classification from MRI Images. *Appl. Sci.* **2025**, *15*, 5412. [CrossRef]
8. Abd-Ellah, M.K.; Awad, A.I.; Khalaf, A.A.; Hamed, H.F. A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned. *Magn. Reson. Imaging* **2019**, *61*, 300–318. [CrossRef]
9. Magadza, T.; Viriri, S. Deep learning for brain tumor segmentation: A survey of state-of-the-art. *J. Imaging* **2021**, *7*, 19. [CrossRef]
10. Madgi, M.; Giraddi, S.; Bharamagoudar, G.; Madhur, M. Brain tumor classification and segmentation using deep learning. In *Smart Computing Techniques and Applications: Proceedings of the Fourth International Conference on Smart Computing and Informatics*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 2, pp. 201–208.
11. Nadeem, M.W.; Ghamdi, M.A.A.; Hussain, M.; Khan, M.A.; Khan, K.M.; Almotiri, S.H.; Butt, S.A. Brain tumor analysis empowered with deep learning: A review, taxonomy, and future challenges. *Brain Sci.* **2020**, *10*, 118. [CrossRef]
12. Faradibah, A.; Widyawati, D.; Syahar, A.U.T.; Jabir, S.R.; Belluano, P.L.L. Comparison analysis of random forest classifier, support vector machine, and artificial neural network performance in multiclass brain tumor classification. *Indones. J. Data Sci.* **2023**, *4*, 55–63. [CrossRef]
13. Latif, G.; Ben Brahim, G.; Iskandar, D.A.; Bashar, A.; Alghazo, J. Glioma Tumors' classification using deep-neural-network-based features with SVM classifier. *Diagnostics* **2022**, *12*, 1018. [CrossRef]
14. Ahmad, S.; Choudhury, P.K. On the performance of deep transfer learning networks for brain tumor detection using MR images. *IEEE Access* **2022**, *10*, 59099–59114. [CrossRef]

15. Takahashi, S.; Sakaguchi, Y.; Kouno, N.; Takasawa, K.; Ishizu, K.; Akagi, Y.; Aoyama, R.; Teraya, N.; Bolatkan, A.; Shinkai, N.; et al. Comparison of vision transformers and convolutional neural networks in medical image analysis: A systematic review. *J. Med. Syst.* **2024**, *48*, 84. [CrossRef] [PubMed]
16. Matsoukas, C.; Haslum, J.F.; Söderberg, M.; Smith, K. Pretrained vits yield versatile representations for medical images. *arXiv* **2023**, arXiv:2303.07034. [CrossRef]
17. Dahan, S.; Fawaz, A.; Williams, L.Z.; Yang, C.; Coalson, T.S.; Glasser, M.F.; Edwards, A.D.; Rueckert, D.; Robinson, E.C. Surface vision transformers: Attention-based modelling applied to cortical analysis. In Proceedings of the International Conference on Medical Imaging with Deep Learning, PMLR, Zurich, Switzerland, 6–8 July 2022; pp. 282–303.
18. Feng, C.M.; Yan, Y.; Chen, G.; Xu, Y.; Hu, Y.; Shao, L.; Fu, H. Multimodal transformer for accelerated MR imaging. *IEEE Trans. Med. Imaging* **2022**, *42*, 2804–2816. [CrossRef]
19. Dahan, S.; Williams, L.Z.; Rueckert, D.; Robinson, E.C. The multiscale surface vision transformer. *arXiv* **2024**, arXiv:2303.11909v3.
20. Thakur, G.K.; Thakur, A.; Kulkarni, S.; Khan, N.; Khan, S. Deep learning approaches for medical image analysis and diagnosis. *Cureus* **2024**, *16*, e59507. [CrossRef]
21. Babayomi, M.; Olagbaju, O.A.; Kadiri, A.A. Convolutional xgboost (c-xgboost) model for brain tumor detection. *arXiv* **2023**, arXiv:2301.02317. [CrossRef]
22. Zhu, G.; Jiang, B.; Tong, L.; Xie, Y.; Zaharchuk, G.; Wintermark, M. Applications of deep learning to neuro-imaging techniques. *Front. Neurol.* **2019**, *10*, 869. [CrossRef]
23. Azizova, A.; Prysiazhniuk, Y.; Wamelink, I.J.; Cakmak, M.; Kaya, E.; Wesseling, P.; de Witt Hamer, P.C.; Verburg, N.; Petr, J.; Barkhof, F.; et al. Preoperative prediction of diffuse glioma type and grade in adults: A gadolinium-free MRI-based decision tree. *Eur. Radiol.* **2025**, *35*, 1242–1254. [CrossRef]
24. Al Yassin, A.; Sadaghiani, M.S.; Mohan, S.; Bryan, R.N.; Nasrallah, I. It is About "Time": Academic Neuroradiologist Time Distribution for Interpreting Brain MRIs. *Acad. Radiol.* **2018**, *25*, 1521–1525. [CrossRef]
25. Sieber, V.; Rusche, T.; Yang, S.; Stieltjes, B.; Fischer, U.; Trebeschi, S.; Cattin, P.; Nguyen-Kim, D.L.; Psychogios, M.N.; Lieb, J.M.; et al. Automated assessment of brain MRIs in multiple sclerosis patients significantly reduces reading time. *Neuroradiology* **2024**, *66* , 2171–2176. [CrossRef]
26. Aamir, M.; Rahman, Z.; Bhatti, U.A.; Abro, W.A.; Bhutto, J.A.; He, Z. An automated deep learning framework for brain tumor classification using MRI imagery. *Sci. Rep.* **2025**, *15*, 17593. [CrossRef] [PubMed]
27. Kong, C.; Yan, D.; Liu, K.; Yin, Y.; Ma, C. Multiple deep learning models based on MRI images in discriminating glioblastoma from solitary brain metastases: A multicentre study. *BMC Med. Imaging* **2025**, *25*, 171. [CrossRef] [PubMed]
28. Ural, B. A computer-based brain tumor detection approach with advanced image processing and probabilistic neural network methods. *J. Med. Biol. Eng.* **2018**, *38*, 867–879. [CrossRef]
29. Ullah, Z.; Farooq, M.U.; Lee, S.H.; An, D. A hybrid image enhancement based brain MRI images classification technique. *Med. Hypotheses* **2020**, *143*, 109922. [CrossRef]
30. Varuna Shree, N.; Kumar, T. Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network. *Brain Inform.* **2018**, *5*, 23–30. [CrossRef]
31. Kharrat, A.; Gasmi, K.; Messaoud, M.B.; Benamrane, N.; Abid, M. A hybrid approach for automatic classification of brain MRI using genetic algorithm and support vector machine. *Leonardo J. Sci.* **2010**, *17*, 71–82.
32. Rajan, P.; Sundar, C. Brain tumor detection and segmentation by intensity adjustment. *J. Med. Syst.* **2019**, *43*, 282. [CrossRef]
33. Çinar, A.; Yildirim, M. Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture. *Med. Hypotheses* **2020**, *139*, 109684. [CrossRef]
34. Mehnatkesh, H.; Jalali, S.M.J.; Khosravi, A.; Nahavandi, S. An intelligent driven deep residual learning framework for brain tumor classification using MRI images. *Expert Syst. Appl.* **2023**, *213*, 119087. [CrossRef]
35. Deepak, S.; Ameer, P. Brain tumor classification using deep CNN features via transfer learning. *Comput. Biol. Med.* **2019**, *111*, 103345. [CrossRef] [PubMed]
36. Díaz-Pernas, F.J.; Martínez-Zarzuela, M.; Antón-Rodríguez, M.; González-Ortega, D. A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network. *Healthcare* **2021**, *9*, 153. [CrossRef] [PubMed]
37. Khan, M.S.I.; Rahman, A.; Debnath, T.; Karim, M.R.; Nasir, M.K.; Band, S.S.; Mosavi, A.; Dehzangi, I. Accurate brain tumor detection using deep convolutional neural network. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 4733–4745. [CrossRef] [PubMed]
38. Paul, J.S.; Plassard, A.J.; Landman, B.A.; Fabbri, D. Deep learning for brain tumor classification. In *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging*; SPIE: Bellingham, WA, USA, 2017; Volume 10137, pp. 253–268.
39. Hemanth, D.J.; Anitha, J.; Naaji, A.; Geman, O.; Popescu, D.E.; Son, L.H. A modified deep convolutional neural network for abnormal brain image classification. *IEEE Access* **2018**, *7*, 4275–4283. [CrossRef]
40. Shen, Y.; Guo, P.; Wu, J.; Huang, Q.; Le, N.; Zhou, J.; Jiang, S.; Unberath, M. Movit: Memorizing vision transformers for medical image analysis. In Proceedings of the International Workshop on Machine Learning in Medical Imaging, Vancouver, BC, Canada, 8 October 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 205–213.

41. Xia, K.; Wang, J. Recent advances of transformers in medical image analysis: A comprehensive review. *MedComm–Future Med.* **2023**, *2*, e38. [CrossRef]
42. Henry, E.U.; Emebob, O.; Omonhinmin, C.A. Vision transformers in medical imaging: A review. *arXiv* **2022**, arXiv:2211.10043. [CrossRef]
43. Kang, J.; Ullah, Z.; Gwak, J. MRI-based brain tumor classification using ensemble of deep features and machine learning classifiers. *Sensors* **2021**, *21*, 2222. [CrossRef]
44. Ahmed, M.M.; Hossain, M.M.; Islam, M.R.; Ali, M.S.; Nafi, A.A.N.; Ahmed, M.F.; Ahmed, K.M.; Miah, M.S.; Rahman, M.M.; Niu, M.; et al. Brain tumor detection and classification in MRI using hybrid ViT and GRU model with explainable AI in Southern Bangladesh. *Sci. Rep.* **2024**, *14*, 22797. [CrossRef]
45. Hamada, A. Br35H Brain Tumor Detection 2020 Dataset. 2020. Available online: https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection (accessed on 1 August 2020).
46. Chakrabarty, N. Brain MRI Images for Brain Tumor Detection. 2019. Available online: https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection (accessed on 1 August 2025).
47. Shamshad, F.; Khan, S.; Zamir, S.W.; Khan, M.H.; Hayat, M.; Khan, F.S.; Fu, H. Transformers in medical imaging: A survey. *Med. Image Anal.* **2023**, *88*, 102802. [CrossRef]
48. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**, arXiv:2102.04306. [CrossRef]
49. Hatamizadeh, A.; Tang, Y.; Nath, V.; Yang, D.; Myronenko, A.; Landman, B.; Roth, H.R.; Xu, D. Unetr: Transformers for 3d medical image segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 574–584.
50. Chaoyang, Z.; Shibao, S.; Wenmao, H.; Pengcheng, Z. FDR-TransUNet: A novel encoder-decoder architecture with vision transformer for improved medical image segmentation. *Comput. Biol. Med.* **2024**, *169*, 107858. [CrossRef] [PubMed]
51. Sun, G.; Pan, Y.; Kong, W.; Xu, Z.; Ma, J.; Racharak, T.; Nguyen, L.M.; Xin, J. DA-TransUNet: Integrating spatial and channel dual attention with transformer U-net for medical image segmentation. *Front. Bioeng. Biotechnol.* **2024**, *12*, 1398237. [CrossRef] [PubMed]
52. Asiri, A.A.; Shaf, A.; Ali, T.; Shakeel, U.; Irfan, M.; Mehdar, K.M.; Halawani, H.T.; Alghamdi, A.H.; Alshamrani, A.F.A.; Alqhtani, S.M. Exploring the power of deep learning: Fine-tuned vision transformer for accurate and efficient brain tumor detection in MRI scans. *Diagnostics* **2023**, *13*, 2094. [CrossRef] [PubMed]
53. Wang, J.; Lu, S.Y.; Wang, S.H.; Zhang, Y.D. RanMerFormer: Randomized vision transformer with token merging for brain tumor classification. *Neurocomputing* **2024**, *573*, 127216. [CrossRef]
54. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Finding Extreme Points in Contours with OpenCV. PyImageSearch. 2020. Available online: https://www.pyimagesearch.com/2016/04/11/finding-extreme-points-in-contours-with-opencv (accessed on 10 August 2020).
55. Perez, L.; Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621. [CrossRef]
56. Yang, S.; Xiao, W.; Zhang, M.; Guo, S.; Zhao, J.; Shen, F. Image data augmentation for deep learning: A survey. *arXiv* **2022**, arXiv:2204.08610.
57. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
58. Wu, B.; Xu, C.; Dai, X.; Wan, A.; Zhang, P.; Yan, Z.; Tomizuka, M.; Gonzalez, J.; Keutzer, K.; Vajda, P. Visual transformers: Token-based image representation and processing for computer vision. *arXiv* **2020**, arXiv:2006.03677. [CrossRef]
59. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
60. Erickson, B.J.; Korfiatis, P.; Akkus, Z.; Kline, T.L. Machine learning for medical imaging. *Radiographics* **2017**, *37*, 505–515. [CrossRef]
61. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
62. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
63. Yu, T.; Zhu, H. Hyper-parameter optimization: A review of algorithms and applications. *arXiv* **2020**, arXiv:2003.05689. [CrossRef]
64. Tran, N.; Schneider, J.G.; Weber, I.; Qin, A.K. Hyper-parameter optimization in classification: To-do or not-to-do. *Pattern Recognit.* **2020**, *103*, 107245. [CrossRef]
65. Claesen, M.; De Moor, B. Hyperparameter search in machine learning. *arXiv* **2015**, arXiv:1502.02127. [CrossRef]

66. Belete, D.M.; Huchaiah, M.D. Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *Int. J. Comput. Appl.* **2022**, *44*, 875–886. [CrossRef]
67. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.