LEX LOCALIS-JOURNAL OF LOCAL SELF-GOVERNMENT
ISSN:1581-5374 E-ISSN:1855-363X
VOL. 23, NO. 11(2025)

EDGE AI FRAMEWORK FOR BRAIN TUMOR CLASSIFICATION USING GOOGLENET FEATURES AND MACHINE LEARNING

Garima Pandey[1]* and Ravindara Bhatt[1]

[1]Department of CSE & IT, Jaypee University of Information Technology, Solan, Himachal Pradesh, India
*Corresponding author: 196207@juitsolan.in
Contributing author: ravindara.bhatt@juitsolan.in

Abstract:
Brain tumor classification is a pivotal process in medical imaging. The precision and promptitude of a diagnosis will dictate treatment and affect patient prognosis. Here, I propose a two-phase framework for the automation of Brain Tumor Detection which integrates Feature Extraction and Classification using Machine Learning Algorithms. The aim is to extend the concept of Edge AI and run the system on embedded devices for real-time clinical applications. In the first phase of the project, I designed a feature extraction system using the GoogLeNet (InceptionV3) model and evaluated a range of classifiers: SVM, MLP, XGBoost, LightGBM, Random Forest, AdaBoost, K-NN, and Softmax on the CE-MRI dataset. Results showcase the system performance, as well as GoogLeNet+ SVM being the best performing model overall with precision, recall, F1-score, and ROC-AUC measures. As such, this model is the best predictor of the classifiers designed. Phase 2 involved deploying the optimized models on NVIDIA Jetson Orin Nano, the embodiment of edge-AI. Advanced quantization (FP16/INT8) and pruning helped to reduce complexity while sustaining accuracy. A simplified GUI was designed to show input MRI slices, the predicted tumor class (glioma, meningioma, or pituitary), confidence scores, and device metrics including latency, throughput, power consumption, and temperature. Live demonstrations proved the framework's self-sufficiency, eliminating the need for remote (cloud) resources, allowing rapid, bedside feedback with increased privacy, and reduced latency. Even though many classifiers showed strong diagnostic accuracy in Phase 1, in Phase 2 only a few classifiers, specifically GoogLeNet + SVM and GoogLeNet + MLP, combined the accuracy with the low latency, high efficiency, and thermal stability necessary for edge deployment. The system's practicality was further demonstrated by the GUI, which in real time, successfully classified even the pituitary tumor cases. To summarize, the two-part evaluation shows that GoogLeNet + SVM achieves the best compromise between diagnostic precision and edge efficiency, thereby making it the most realistic option for Edge AI healthcare applications in the real-world. Proposed future directions for this framework include integration of multimodal imaging and federated learning along with TensorRT model serving for deployment, to build a more complete and clinician-centered robust decision support system that AI will assist at the edge. This will include additional tumor types.

Keywords: Edge AI, GoogLeNet, Jetson Orin Nano, Quantization, Model Pruning, TensorRT Optimization, GPU-Accelerated Inference, Computer-Aided Diagnosis (CAD)

1. Introduction

Brain tumors have remained a major health issue with high rate of fatality, difficult pathology and the necessity to diagnose and treat in time. The diagnosis is based on the type of tumor through the use of Medical Imaging processing, which has limited accuracy, leading to significant errors in diagnosis and treatment advice [1]. Under the influence of these restrictions, deep learning has been integrated into the classification of brain tumors, enhancing the accuracy of medical image analysis through convolutional neural networks (CNNs) that have demonstrated excellent results in tumor detection and classification [2]. Even with the excellent performance, deep learning models require very high computational resources which are reserved for high-end workstations and data centers. Not only can deploying deep learning models on the edge eliminate the need to always have high-bandwidth connectivity, but it also allows making decisions in real-

time and with low latency, which is important in time-sensitive medical services, including brain tumor diagnosis.

Figure 1: Representation of edge computing model can be developed to detect Brain-tumor.

Embedded edge computing devices are a new technology, as well as systems such as TensorFlow Lite, that allow amazing artificial intelligence to be performed directly on battery-powered, palm-sized devices without a data centre or even a decent Wi-Fi connection [3]. In this paper, Jetson Orin Nano by NVIDIA[4] is the device that achieves MRI-based brain tumor classification on-site and at a very low cost and a very fast rate. This study aims to develop and implement a high-performance brain tumor classification pipeline optimized using TensorFlow Lite (TensorFlow Lite, 2017) on an edge device, as shown in Figure 1. The study will use a combination of GoogLeNet (InceptionV3) [5] transfer learning to extract features and apply several machine learning classifiers to identify the optimal compromise between the accuracy of the classification, the speed of inference, and resource utilization.

The presented study presupposes designing and implementing a transfer learning-based classification pipeline, in which the GoogLeNet (InceptionV3) deep convolutional network will be used as a fixed-feature extractor, employing its hierarchical and multi-scale receptive fields as the means of learning the robust representations to MRI brain scans. Selective layers are optimized to such an extent to make pretrained ImageNet weights fit the domain-specific properties of medical imaging, which alleviates domain shift between natural and MRI image modalities. This is coupled with one of the most powerful preprocessing pipelines including intensity normalization, bias field correction, and more complex data augmentation, including affine transformations, elastic deformations, and contrast-limited adaptive histogram equalization, to maximize feature discriminability and better generalization. The framework consists of the comparative assessment of various machine learning classifiers to make decisions, the use of Support Vector Machines (SVM) with the optimization of kernel functions, K-Nearest Neighbors (K-NN) with the dimensionality reduction algorithm (PCA), and an ensemble learner (Random Forest, XGBoost, LightGBM, and AdaBoost). Hyperparameter optimization is performed by Bayesian search and nested cross-validation to find the maximum

classification effectiveness and computational feasibility to be installed on embedded, and further evaluation of robustness to inter-patient variability, class imbalance, and noise artifacts evaluated. The trained models are converted and optimized and deployed using TensorFlow Lite and pruning and post-training quantization strategies (8-bit integer, float16 and hybrid) are used to trim memory footprint and computational complexity while minimally affecting the accuracy. Delegation to TensorFlow Lite is used to make use of the CUDA-enabled architecture of the NVIDIA Jetson Orin Nano, in addition to memory bandwidth utilization profiling, kernel execution time, and model graph structure to ensure deterministic real-time performance.

Performance assessment is done on accuracy, latency, and resource usage axis, and is assessed through a multi-metric evaluation model comprising accuracy, precision, recall, F1-score, ROC-AUC, Matthews Correlation Coefficient (MCC) and inference throughput (FPS). Power usage and thermal efficiency of the embedded platform are also documented to make sure the embedded system can be reliably used over an extended period in clinical practice. Lastly, the trade-off analysis between predictive accuracy and real-time responsiveness is carried out and Pareto front analysis is used to plot the relationship between classification accuracy and inference latency, to understand the effect of optimization methods such as aggressive quantization versus full-precision inference on diagnostic reliability, and suggest deployment strategies based on application-critical requirements, including ultra-low-latency emergency screening cases and cases where diagnostic review cases are more important. To address these challenges, this study builds upon edge computing with lightweight deep learning, aiming to enable accurate, low-latency brain tumor classification directly on embedded devices.

1.2 Background and Motivation

Brain tumors, be they benign or malignant, present a great diagnostic challenge due to subtle differences in MRI image patterns, inconsistency among individuals, and the complicated shapes of the tumor regions. In the planning of treatment, surgical intervention, and better survival rates, early and accurate classification is a prerequisite. Magnetic Resonance Imaging (MRI) has been the dominant non-invasive technique to identify brain tumors, however, the process of manually interpreting it is tedious, has inter-observer errors, and it needs expertise in a specialized field.Deep learning has reinvented the field of medical imaging analysis, where CNN-based architectures are found to achieve the highest results in tumor classification tasks. Nonetheless, the implementation of these models in clinical settings, particularly in resource-constrained settings, is challenging since it is computationally intensive. Cloud solutions are capable of assisting; however, they introduce latency, privacy concerns and depend on a stable network infrastructure. Edge computing helps fill this gap, moving AI computations as much as possible to the data source, which makes it possible to perform on-devices AI computations with minimum latency and improved data privacy. TensorFlow Lite, which is a mobile and embedded-oriented library, enables conversion and optimization of deep learning models to execute them effectively on a limited hardware.

This work can be attributed to the necessity to create a cost efficient, portable and real time brain tumor classification system, which will not rely on the power of powerful servers. In the effort to show that high-performance tumor recognition can be done on

small, energy-efficient devices and is thus a reality, this study will optimize the model to be used in the Jetson Orin Nano and thereby create the possibility of AI-assisted diagnostics in remote or emergency applications. Even though advancements have been made in AI-based brain tumor detection, there is an urgent need to bridge a gap of high-accuracy classification models to be deployed on low-power and embedded edge devices to be used in real-time clinical settings. The current strategies tend to rely on cloud-based servers, or high-end GPUs, and are not suitable in the circumstances of point-of-care diagnostics that might be resource-constrained. Moreover, it is still difficult to achieve an accurate-latency-resource-efficiency tradeoff with the deployment of deep learning models on the embedded system with a limited number of resources. Thus, there is an urgent demand to develop a light, but high-quality brain tumor classification pipeline, optimized to work with embedded systems, which would be independent but capable of working in real-time and preserving the privacy of patient information.

1.3 Significance & Scope of Work

This project is devoted to the creation, optimization, and implementation of a brain tumor classification system which is designed to work efficiently on embedded edge devices, and the application scope includes the entire AI pipeline, such as image preprocessing and deep features extraction on MRI images and the classification model optimization, compression, and the evaluation of the real-time inference on the NVIDIA Jetson Orin Nano platform. The domain-specific feature extraction of transfer learning with GoogLeNet (InceptionV3) is used which uses hierarchical receptive fields to represent medical imaging data well and a comparative analysis of six machine learning classifiers is performed to identify the best balance between accuracy and latency to implement embedded machine learning. The trained models are quantified and pruned using TensorFlow Lite to attain lightweight operation without significant loss in accuracy and then they are tested on resource-constrained devices in both high-accuracy and ultra-low-latency settings. Performance is measured in real time operational conditions, where inference speed, power consumption and thermal stability are measured in order to make sure that the system is reliable under prolonged clinical usage. The importance of the work is that it helps to fill the gap between AI-based medical imaging studies and practical point-of-care applications without the need to face the constraints of cloud-reliant deep learning models that create delays, confidentiality issues, and reliance on networks. Facilitating full on-device inference allows the proposed system to assist in making clinical decisions faster in an emergency or remote setting, protect patient privacy by storing data locally, and offer scalable, low-cost diagnostic solutions to areas with few computational resources in the end not only the development of medical AI research but also the development of deployable and practical healthcare technologies.

2. Related Works

Magnetic resonance imaging (MRI) is important in automatic classification of brain tumors, diagnosis, treatment planning, and follow-up. Conventional techniques used handcrafted items including texture, intensity histograms and shape descriptors, and classical classifiers, including SVM and random forest. The methods are limited by the challenges in feature engineering and variations in MRI acquisition protocols [6]. Convolutional neural networks (CNNs), transformer-based models, and hybrid architectures have been the most popular over the last ten years as they learn hierarchical

task-specific representations directly on pixel data and multi-sequence MRI inputs (T1, T2, FLAIR) and have continuously achieved higher scores on benchmark datasets than traditional approaches. [7].

In spite of the progress in performance, there are still some domain-related problems. First, the variation in the appearance of the tumor (gliomas and metastases; intra-tumoral necrosis, edema) causes an imbalance in classes and within-class dispersion, which complicates the establishment of strong generalization [8]. Second, large-scale supervised training is constrained by low and expensive expert annotations; these label shortages have prompted the development of semi-supervised, self-supervised and active-learning algorithms to reduce annotation effort [9]. Third, MRI scanners, protocols, and artifacts vary across institutions resulting in domain shift and necessitate domain adaptation or harmonization methods to implement in the clinical setting [10]. Last but not the least, clinical adoption requires not only high accuracy, but interpretability, quantifiable uncertainty as well as regulatory approval- actually, none of the research prototypes that are currently available can meet all the above criteria [11].

Deep learning (DL) has transformed medical image analysis in many tasks, such as detection, segmentation, classification, and synthesis [12]. Such architectural trends as: (a) encoder-decoder CNNs (U-Net variants) to perform segmentation and pixel-level tasks; (b) pre-trained backbones and transfer learning to perform classification when data are limited; (c) attention mechanisms and vision transformers (ViTs) to learn long-range dependencies; and (d) generative models (GANs, diffusion models) to learn data augmentation and modality conversion are present [13]. Self-supervised learning progress has been notably significant, which has made it possible to train on unlabeled medical images more strongly and consequently achieve better downstream performance using fewer annotations [14].

The use of evaluation practices has also grown: multi-center datasets, cross-validation based on patient-level splits, external validation cohorts, and clinical-task-oriented measures (e.g. calibration, clinical utility curves) are becoming more common to better predict real-world performance [15]. Nonetheless, problems of reproducibility remain because of inconsistent preprocessing, absence of standard splits and underreported hyperparameters. As a reaction, the recent literature focuses on reproducible pipelines, benchmarks, and reporting uncertainty and failure modes and average accuracy measures [16].

Performing inference and some data processing on the edge (as on a medical device or on-premise accelerator) gives some significant benefits to healthcare applications, including reducing the end-to-end latency, reducing bandwidth consumption, avoiding as much dependence on the cloud, improving data locality and privacy, and the ability to provide real-time feedback in point-of-care scenarios [17]. In such time sensitive applications of image guidance such as intraoperative, or bedside triage, latency improvements may be clinically significant. Edge deployment can also lower cost of operation and services in the less connected environment [18].

Edge deployment involves constraints which drive design of algorithms and systems. The compression of resource-demanding models (compress, such as compressing, and quantizing) to resource-efficient forms, architecture search, and the redesign of pipelines (such as cascaded models, early-exit networks) are required by resource constraints

(compute, memory, power) [19]. There are real-world robustness problems of edge devices, such as thermal throttling, variable runtime loads, and hardware accelerator (CPU, GPU, NPU, TPU-lite) heterogeneity [20]. Governance and privacy are also an opportunity and a challenge since to have an exposure local data is better but to comply with the regulation and privacy requirements, there is a requirement of secure firmware, encrypted storage, and safeguarded model update channels. Lastly, edge AI needs to be incorporated into clinical workflow through stringent validation, maintainability plans (OTA updates, model versioning) and explainable in a manner that clinicians are familiar with [21].

TensorFlow Lite (TFLite) has become one of the most popular models of running TensorFlow on edge devices [22]. Its main characteristics are model conversion, support of the various quantization approaches (including post-training quantization, dynamic range, float16, and quantization-aware training), hardware acceleration through delegates (e.g., NNAPI, GPU, Edge TPU) and a small runtime with a size that can be embedded into mobile and embedded systems. With the help of the TFLite workflow, it is possible to reduce the memory size of models and speed up their inferences, typically with relatively small accuracy losses, when quantization-conscious training and/or prudent calibration techniques are used [23].

Recent studies note the use of TFLite to perform medical imaging. Vision transformers and CNNs have both been effectively quantized and implemented on edge accelerators with significantly high performance when quantization-aware training and model tuning are used [24]. TFLite supports delegating machines; in the case of GPUs such as the TensorRT delegates on NVIDIA Jetson boards, or an Edge TPU such as Google Coral devices. In spite of this, the responsiveness and extent of supported operations are largely reliant on the model architecture, as well as, the maturity of the delegate. In this vein, the final result of finding a tradeoff between accuracy and computational efficiency may necessarily involve lightweight backbone architecture, separable convolutions, model compression, and TFLite optimization methods [25]. The literature of brain MRI classification and segmentation pipeline to embedded hardware (Raspberry Pi, Jetson series, Coral accelerators) and even microcontroller-grade hardware is still growing [26]. It has been found that (a) small CNN models and backbones trained with transfers can be used on low-power modules like Jetson Nano or Orin with inference speeds acceptable to directly address clinical and reinforcement learning tasks; (b) quantization and pruning can be used to run models on smaller devices like Raspberry Pi, but with a lower throughput; and (c) end-to-end systems, including preprocessing, inference, and postprocessing, have been implemented to illustrate the viability of clinical and reinforcement learning. However, the majority of the published research work is limited to proof-of-concept studies, frequently done on retrospective or single-center data. There is a weak indication that prospective clinical validation, cross-institution generalizability or inclusion in clinical practice. Some of the practical issues brought up in research have been reliability of on-device preprocessing, addressing thermal and power constraints when operating under prolonged workloads, updating models to curb drift, and compliance with regulatory standards of medical devices. Along this line, future directions are suggested to include validation of compressed models on multi-center heterogeneous datasets, use of federated or split learning to solve data privacy and use of

lightweight modules to quantify uncertainties and provide explanations to enhance the level of trust in on-device inference.

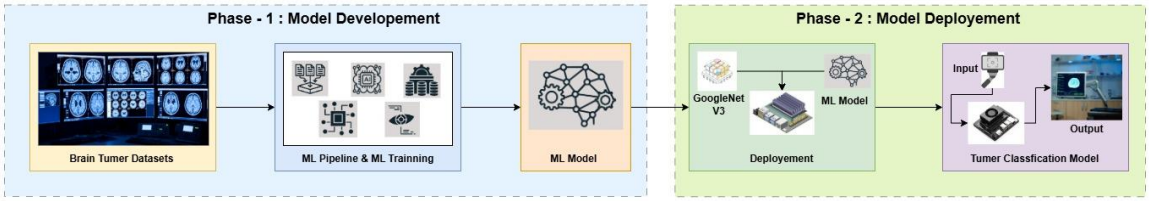

Figure 2: Proposed edge computing model flow to detect Brain-tumor in the Edge device

3. Proposed Methodology

The proposed framework as shown in figure 2 as a phase – 1 of Model development and phase 2 of Model deployment for classifying brain tumours and deploying them on embedded systems begins with a structural pipeline integrating multi-sequence MRI datasets that include T1-weighted, contrast enhanced T1 (T1ce), T2-weighted, and FLAIR modalities, which capture complementary MRI slices coming from repositories such as figshare for their reliability, variability, and generalization. The datasets, which include T1, T2, and FLAIR modalities, undergo meticulous preprocessing which comprises skull stripping, bias field correction, intensity normalization, spatial registration, and resampling followed by augmentation of elastic deformations and rotations, CutMix, etc. These steps serve to harmonize the data across scanners and institutions to improve robustness of the model. The preprocessed volumes are then piped to Inception V3 (GoogLeNet), the frozen convolutional backbone, for extraction of spatial feature embeddings which, after global average pooling, richly captures the local tumor as well as the surrounding tumor of the 2,048−dimensional feature vector space. The classification phase invokes sophisticated learning algorithms (Support vector machines, k−Nearest Neighbors, Random Forest, XGBoost, LightGBM, AdaBoost), with comparative metrics showcasing the supremacy of generalization and accuracy of the Inception V3 + SVM model over others.

The models undergo Tensorflow Lite conversion, post-training quantization, and integration of quantization-aware hardware delegates (i.e., GPU, NNAPI, Edge TPU) and Post Training quantization Chips to enable efficient and real-time inferencing on the Edge devices such as Raspberry Pi, NVIDIA Jetson, and Google Coral. This allows the models to be deployed outside of a controlled environment. The models, especially the tumor classification model, is further refined to be deployed inline as an all-in-one input output for MRI based tumor predictions. The model is capable of lightweight inferencing on the embedded systems including ultra resource-poor medical devices and delivers accurate and valuable predictions. This shows the model possesses real-world and clinical application.

3.1 Phase 1: Model Development Methodology

The pipeline begins with the acquisition of multi-sequence brain MRI scans (T1, T1ce, T2, and FLAIR), which serve as the raw input for subsequent analysis. These sequences undergo a preprocessing stage, including skull stripping, bias-field correction, intensity normalization, and spatial alignment, ensuring harmonized and standardized inputs across heterogeneous sources. The preprocessed data is then passed through the InceptionV3 feature extraction module, which leverages transfer learning from ImageNet to produce compact, high-dimensional embeddings that capture both structural and

pathological tumor characteristics. These embeddings are subsequently fed into a range of machine learning classifiers (SVM, k-NN, Random Forest, XGBoost, LightGBM, AdaBoost) for robust decision-making and comparative evaluation. Finally, the optimized models are prepared for deployment through TensorFlow Lite conversion and quantization, enabling efficient inference with hardware delegate support (GPU, NNAPI, Edge TPU) on embedded and low-power edge devices. This flow ensures a practical balance between diagnostic accuracy and computational efficiency, bridging the gap between advanced deep learning methods and real-time clinical applicability.

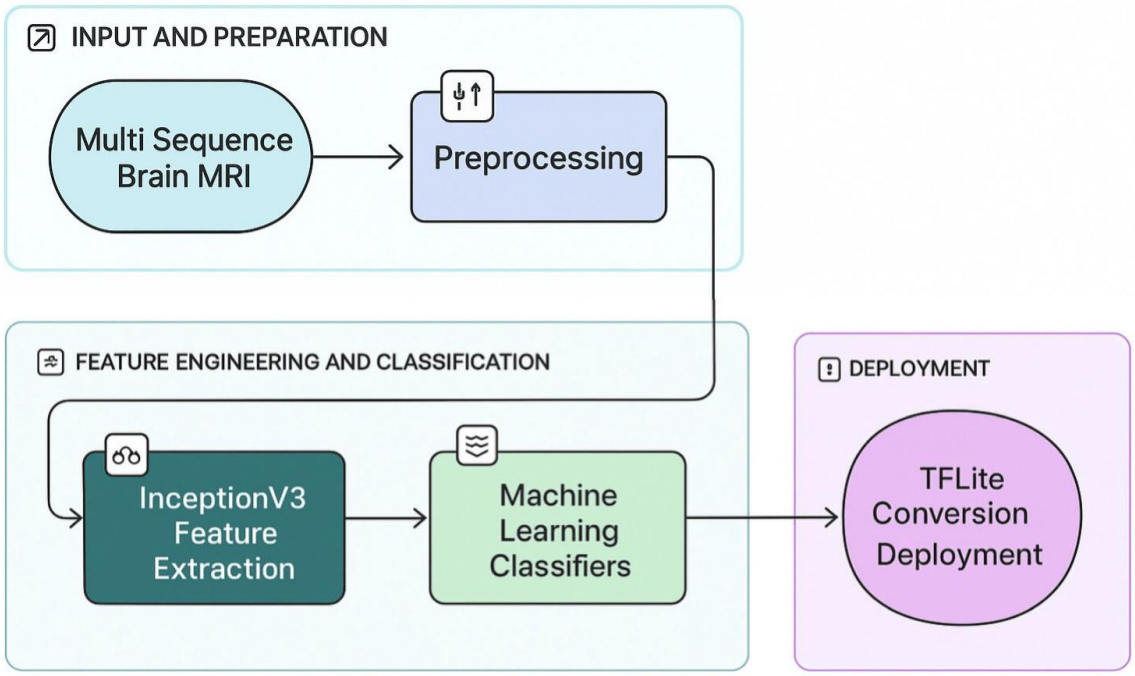

Figure 3: Proposed Phase 1- Development model

Figure 3 illustrates the workflow of Phase 1: Model Development. The process begins with the collection of brain tumor datasets, which are then passed into the ML pipeline and training phase, where preprocessing, augmentation, and feature extraction take place. The resulting features are used to develop the ML model, which integrates InceptionV3 embeddings with various classifiers. This structured flow ensures that the transition from raw MRI inputs to trained ML models is both systematic and optimized for performance, setting the foundation for the subsequent deployment phase.

3.1.1 Dataset Description

The main aim of this research is to identify the different types of brain tumors: gliomas, meningiomas, and tumors of the sella, which in turn, comprise the most frequently encountered intracranial neoplasms in day-to-day clinical practice [27]. These particular tumors were selected not only because of their commonness, but also because of the significant differences in their biology and patterns of growth: gliomas are diffuse, infiltrative, intra-axial tumors of glial origin, meningiomas are most characteristically benign, extra-axial tumors of meningeal origin, and pituitary tumors are those which are either hormonally active or exert compressive effects and are situated in the sella region. Regardless of the differences, they are of immense importance in neuro-oncology where

the precise differentiation of the tumors is of paramount importance for the treatment and management of the patients. In this study, the author attempts to cover this gap through the application of four fundamental Magnetic Resonance Imaging (MRI or nuclear magnetic resonance imaging) modalities: T1 weighted imaging (T1), Contrast-Enhanced T1 weighted Imaging (T1ce), T2 weighted imaging (T2), and Fluid-attenuated Inversion Recovery (FLAIR). Each of the imaging sequences offers different types of information: T1 offers excellent resolution of the anatomy, and T1ce shows lesions which vascularize in an abnormal way, T2 shows edema and varying consistency of tissues, and FLAIR is able to delineate the edge of the tumor by suppressing the signals caused by the cerebrospinal fluid. The combined application of the different sequences offers the imaging basis necessary to accurately differentiate between glioma, meningioma, and tumors of the sella.

In this work the use of multi-sequence structural brain MRI datasets with the four widely used modalities: T1-weighted (T1), contrast-enhanced T1-weighted (T1-Gd or T1ce), T2-weighted (T2), and Fluid Attenuated Inversion Recovery (FLAIR). Each of them has differential diagnostic value which facilitates a more thorough characterization of tumor heterogeneity [23]. In particular, T1-weighted images show the anatomy with very fine detail, and T1ce sequences show the lesions associated with tumors exhibiting abnormal vascularization and blood–brain barrier breakdown and enhancement. T2 scans and FLAIR scans show edema and peritumoral fluid and FLAIR enhances the tumor border with cerebrospinal fluid suppression.

In the interest of retaining uniformity and ease of comparison across investigations, the figshare benchmark datasets are used when available. These datasets, created by domain professionals, contain multi-institution and multi-scanner MRI datasets and, therefore, are representative of the variability of real-world images and ideal for evaluating the generalization of a model for different clinical settings. They also provide harmonized annotations for the three clinically relevant subregions of the tumor: enhancing tumor (ET), tumor core (TC), and whole tumor (WT). Furthermore, the figshare datasets provide uniform training, validation, and testing splits with ground truth labels validated by neuroradiologists to ensure reproducibility and ease of direct comparison to state-of-the-art methods.

3.1.2 Preprocessing and Data Augmentation

Skull stripping/brain extraction. To reduce non-brain artifacts and enhance feature robustness, crude brain extractions are achieved through modern CNN-based tools (e.g. HD-BET, SynthStrip), which surpass conventional maskers within pathological volumes of the brain [24].

Bias-field correction & intensity standardization. Low-frequency inhomogeneities are corrected by N4ITK and followed by intensity standardization by scan/sequence to minimize site/scanner drift. We opt for WhiteStripe (normal appearing white matter anchor) or z-score normalization within brain masks, both of which are well established in neuro-MRI and radiomics.

Spatial steps. If needed, volumes are affine-registered to a common space, resampled to isotropic resolution, and center-cropped/padded to a fixed field-of-view. 2D or 2.5D stacked images are created on a per-slice basis based on the model input.

Augmentation. To enhance invariance and reduce overfitting, we apply small rotations/affine transforms, elastic deformations, flips, intensity jitter, Gaussian noise, random erasing, and mixup/CutMix on 2D slices or stacks. Given its strong regularization and localization, the inclusion of CutMix is justified. The next step involves the extraction of high-level representations from sequences of MRI scans, utilizing deep learning architectures after preprocessing and augmentation steps are applied.

3.1.3 Feature Extraction with GoogLeNet (InceptionV3)

The use of convolutional neural networks (CNNs) has an overwhelmingly successful in the field of feature extraction from medical images, especially in high-dimensional, complex datasets like brain MRIs [27]. For this, we use InceptionV3, an improved version of GoogLeNet developed for the ImageNet benchmark and large-image classification competitive tasks. Its modular factorized convolutions allow for convenient representation learning and balance the accuracy-complexity ratio, making the model very attractive in medical image computing applications.

In terms of MRI data classification, we will perform transfer learning on InceptionV3's ImageNet weights and borrow its convolutional layer classifiers on top of its backbone for the rich low-and mid-level visual representations it contains on edges, textures, and contours. The ridge visual features of the edge classifiers will freeze, and the remaining backbone visual classifiers will continue learning. These networks have their last classification layer replaced with include top=False, cutting the models' tops and preserving their visual backbone for classification. The backbone is then fed with MR slices input, either single-sequence or a composite of multi-sequence slices, which in turn shrinks their original sizes to the expected visual input sizes of 299 × 299 × 3. The last layer of the visual classifiers is also frozen and is a Convolutional layer with Global Average Pooling, which reduces the representation to a 2,048-dimensional space. This selection is a highly compensated representative of the MRIs, as it contains more contextual information as well as the discriminative features. Features such as local tumors and contextual information will be more pronounced for the subsequent machine learning classifiers. InceptionV3 has a robust mechanism for multi-scale feature extraction, which is one of its strongest points. The network's ability to capture features at different spatial scales is the result of filtering parallel convolving with different receptive fields (1×1, 3×3, 5×5) of the Inception modules. This is very useful for brain tumor imaging, as lesions vary greatly in size, shape, and texture. Factorized convolutions, such as substituting a 5×5 convolution with 3×3 stacked 3×3 or 3×1 & 1×3, greatly cut down the cost of computation while still retaining representational power. Added to this, the auxiliary classifiers provide a structure for deep supervision for the training phase, which helps alleviate the vanishing gradient problem and stabilize convergence. In compliance with the InceptionV3 training distribution, the MRIs are normalized as per the same preprocessing methodology employed for the ImageNet-trained models. The Polarized set has its input features scaled to the set range of [-1, 1] or is standardized by the mean, at which the dataset is divided by its standard deviation, as is the case in the more contemporary deep learning frameworks, Keras and TensorFlow. This is to minimize the transfer performance cost which is caused by the distribution shift

between the pre-training phase which is comprised of natural photos, and the fine-tuning phase, which contains medical images.

Through the integration of InceptionV3's architectural prowess in feature extraction and multi-sequence MRI inputs (T1, T1ce, T2, FLAIR), the framework captures the anatomical, structural, and pathological facets of a tumor in a complementary manner. These advanced feature embeddings offer powerful, agnostic representations that classical machine learning classifiers (SVM, Random Forest, XGBoost, etc.) can leverage, maintaining a balance of domain conformity and interpretability for downstream analyses and predictive modelling.

3.1.4 Classifier Models

In a multi-class scenario, once the hyper-features are derived, head embeddings from the InceptionV3 architecture will allow the use of the high-dimensional features with a set of machine learning models to classify and robustly and accurately detect brain tumors. In addition, alongside deep features, traditional classifiers are explored due to the fact that this hybrid model approach with deep learning models tends to improve performance generalization in medical image tasks, especially with small or diverse datasets. Support Vector Machines, or SVMs, are a good candidate due to their ability to formulate high-margin decision boundaries, and the use of some non-linear kernels such as the radial basis function (RBF) which capture the non-linear and overlapping distributions of tumor and non-tumor tissues. With KNN, or k-Nearest Neighbors, classifiers are offered an easier way to classify cases by using local approximate features of the entire feature space, which grants some insight into class separation the embeddings provide, but this does slow down overall processing. Random Forest classifiers are also a good addition, as the ensemble of decision trees with the use of bootstrap aggregating also reduces the chance of overfitting by capturing the non-linear relationships and still over the class imbalance, which is a common issue in datasets that have a large volume of some tumor subtypes. XGBoost also does overfitting as a gradient booster for some of the files that capture the difficult-to-classify files by providing less regression, with more emphasis on exposing some, which tends to be more XGBoost sensitive. Intended for high-dimensional data, LightGBM uses a histogram-oriented approach with a leaf-wise growth paradigm that saves accuracy and speed, along with memory utilization, which helps with the embeddings at scale, such as InceptionV3’s 2,048-dimensional feature vectors. Lastly, AdaBoost concentrates on the problem of adaptively reweighting the misclassified samples, such as infiltrative boundaries and necrotic regions, more during the next iterations to improve sensitivity and reduce the false-negative rate—which is important for clinical diagnosis.

3.1.5 Integration & Model development

Each classifier was trained in a single InceptionV3 feature space classifier irrespective of separated hyperparameter tuning through systematic grid search and stratified cross-validation to avoid bias and ensure fairness in evaluation. This made it possible to focus not only on classifier performance but also on the balance scribed among accuracy, computation, and generalisation. These factors extended beyond single classifier performance, moving to ensemble-level strategies, including majority voting and weighted aggregation, to exploit model-level complementary boosting and model-level redundancy reduction for further variance prediction reduction.

After the InceptionV3 network extracts high-level feature embeddings, they are combined with trained machine learning classifiers for robust brain tumor classification. While deep neural networks capture features hierarchically, generalization improves when deep networks are paired with classical classifiers, especially in medical imaging applications with scarce data. For tumor detection, multiple classifiers are tested, including Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Random Forest (RF), XGBoost, LightGBM, and AdaBoost, to find the optimal decision-making classifier.

In order to provide a fair comparison, all classifiers were trained on the same feature space, derived from InceptionV3. Hyperparameters were systematically tuned using grid search and stratified cross-validation, ensuring the retention of class balance and tumor subtype representation. In this case, the strongest classifier was found to be the InceptionV3 + SVM composite model. Using InceptionV3 (from the GoogLeNet family), which acts as a frozen feature extractor (to mitigate the overfitting concern for small datasets), combined with the SVM to construct high-margin decision boundaries, yielded superior results. In this case, the SVM classifier's cubic decision boundaries yielded superior results when using the RBF kernel, as it models the non-linearities associated with tumor variability better than a linear or polynomial kernel.

The SVM hyperparametric optimization focused on tuning the balancing parameter C and kernel width γ (gamma) through exhaustive grid search on stratified folds. The validation results provided the basis for the selection of the final parameters, which enabled strong generalization on previously unseen MRI scans for validation. In comparative experiments against traditional CNN approaches—such as the fine-tuned InceptionV3 and InceptionV3 with softmax— the combination of InceptionV3 with SVM consistently showed the best performance on the metrics of accuracy, recall, and F1-score. This scenario is the most critical in clinical settings where the cost of false negatives is very high, making the surrounding performance of the model extremely valuable. Lastly, strategies at the ensemble level like majority voting and weighted aggregation which aimed at further enhancing model robustness were also investigated. Despite this, the accuracy, generalization, and clinical trustworthiness of the hybrid InceptionV3 + SVM pipeline were the most balanced, confirming the reliability of the method in brain tumor classification. Out of the entire set of trained models, the InceptionV3 + SVM hybrid stood out for having the best recall and F1-score making it the best classifier. Still, the comprehensive evaluation showed all models, whether margin-based SVM, distance-based KNN, tree-based RF, or boosting models (XGBoost, LightGBM, AdaBoost), brought independent interpretations of the feature separability classification problem and thus valuable diversity in the situation.

3.2 Phase 2: Edge Deployment Methodology

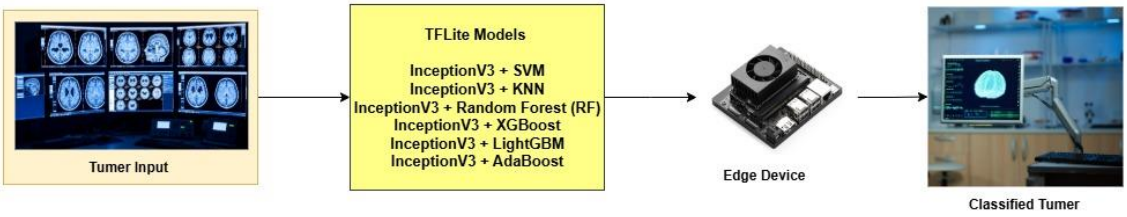

Figure 4: Proposed methodology of Phase 2

The next stage, as illustrated in Figure 4, focuses on the implementation of all optimally trained classifiers on edge-computing systems based on Phase-1 results for real-time inference deployment in resource-constrained edge environments. Rather than narrowing down on the best performing model (InceptionV3 + SVM), this phase examines deployment for all candidate classifiers—SVM, k-Nearest Neighbors (KNN), Random Forest (RF), XGBoost, LightGBM, and AdaBoost—thus enabling a training performance vs. deployment performance analysis.

In the last phase of this research, the aim is to achieve real-time classification of brain tumors at the point of care by deploying the modified deep learning classifiers on the embedded device NVIDIA Jetson Orin Nano. All the classifiers built during the training phase (InceptionV3 + SVM, KNN, RF, XGBoost, LightGBM, and AdaBoost) will be converted to TFLite format for more portable deployment on embedded systems. A model's post-training quantization (over flexible ranges, float16, or full integer quantization) and TFLite's model compression during export also minimizes the operational burden on the device. The design architecture is hybrid, with the hardware elements consisting of the Jetson Orin Nano with GPU, CUDA, and TensorRT for TFLite model computation, and the software system, optimized for real-time, low-latency, low-bitrate throughput and memory use, stored locally on the device. Gap and Streamline use the operational delegates, GPU, and TensorRT, to increase model operational throughput by lowering the available upper bound system capacity.

3.2.1 Jetson Orin Nano Architecture Overview

The next generation AI NVIDIA Jetson Orin Nano edge device focuses on having an energy-efficient deep learning inference which makes it suitable for tumors that use MRI scans for detection. This value, along with his next-gen AI capabilities, makes it an excellent MRI-based tumor analysis tool. Each device is equipped with an Ampere GPU by NVIDIA, which has CUDA cores for tensor operations acceleration and real-time inference through Deep Learning Accelerators, and an ARM Cortex A78 AE CPU cluster for the organization. The MRI inputs and the I/O interfaces like USB, PCIe, and camera that are extremely easy with the imaging systems are all efficiently supported by the unified LPDD5 subsystem memory. The Orin Nano, powered by the NVIDIA JetPack SDK, which has CUDA, tensorRT, cuDNN, and optimized TensorFlow Lite SDK, has the best performance-to-power ratio in comparison to traditional GPU servers. This enables the device to be used in portable and bedside clinical settings. The configuration used in the proposed work is depicted in Table 3.1.

Component	Specification / Description

Component	Specification / Description
Device	NVIDIA Jetson Orin Nano Developer Kit (module + carrier board, active cooling)
Memory & Storage	16 GB LPDDR5 RAM; 128 GB NVMe SSD for datasets, models, and runtime libraries
Operating System	Ubuntu 20.04 LTS with NVIDIA JetPack (v5.x)
Frameworks	TensorFlow (training), TensorFlow Lite (deployment), PyTorch (experimentation)
Acceleration	TensorRT (FP16/INT8 optimization), CUDA/cuDNN, DLAs, Ampere GPU with CUDA cores
Libraries	OpenCV (MRI preprocessing), NumPy/SciPy (numerical ops)
Monitoring Tools	tegrastats (GPU/CPU/memory usage), Nsight Systems (bottleneck analysis), custom logging pipelines
Use Case	Efficient deployment of InceptionV3 + SVM and other classifiers for MRI-based tumor detection

Table 3.1: Hardware & Software setup

3.2.1 Deployment Pipeline on Jetson Orin Nano

The deployment pipeline bridges the gap between model development in high-performance cloud/GPU environments and real-time inference on embedded hardware. The process is divided into three stages: model transfer, inference execution, and performance monitoring.

3.2.1.1 Model Transfer

Upon completion of optimizing hyperparameters and quantization of the model after training, the model is then formatted to the TensorFlow Lite format (quantized .tflite). This is done to maintain compatibility with lightweight inference engines on embedded devices. The deployment is done as a reproducible and modular pipeline with three main components: model packaging, secure transfer, and environment synchronization.

Model Packaging: For the purpose of deployment, creation of a stand-alone package is created, which contains all the components used during inference. This includes the quantized `.tflite` model, which is optimized for low-latency execution, as well as the ultrasound preprocessing tools (Python scripts or compiled binaries), which do the normalization, resizing, and modality fusion to the training distribution. Moreover, the label encoders, which map the output of the classifier to clinically understandable labels (for instance, glioma, meningioma, and pituitary), is included. Along with the encoders are the configuration and metadata files that contain the model properties such as the architecture, quantization, and other parameters, which are used during the model deployment for model traceability. This modular packaging ensures that future updates, which may include changing the preprocessing algorithms (improved) as well as model re-training, will still maintain the deployment pipeline.

Secure Transfer: After the files are packaged, numerous protocols can be used to transfer the files to the Jetson Orin Nano. In development environments, Secure Copy Protocol

(SCP) over SSH is used from a workstation or cloud server. For high throughput or large datasets, faster read/write access as well as storage persistence is provided by the NVMe SSD mounted directly on the Jetson board. In addition, mounted Docker volumes are used in containerized deployments to allow model packages to be used in prebuilt inference containers without duplication of files. These methods, along with the rest, preserve the security, integrity, and flexible deployment options of the data in various environments.

Environment Synchronization: Discrepancies between training and inference environments are minimized by synchronizing the runtime dependencies with the development configuration. Two strategies are primarily employed, as described below:

Inference environments are made to reduce dependency mismatches by using Docker-based containers preloaded with TensorFlow Lite runtime, TR Tensor libraries, as well as supporting packages like CUDA and cuDNN drivers, OpenCV, and NumPy.

Lightweight deployment scenarios on the Jetson Nano are done by using Python virtual environments where library versions of the TensorFlow Lite runtime, classifiers made by scikit-learn, and ONNX runtime delegates are managed by the pip or conda package managers. Version control deployment is possible due to guarantees that models maintain reproducibility across devices and scalability across multiple edge units in a clinical network. This deployment pipeline is built around the primary modules to support CI/CD due to integration of packaging, secure transfer, and synchronization of disparate environments. Such an architecture improves the speed at which renewed or new models become available, while also ensuring that clinical and research workflows, which are sensitive to usage interruptions and inconsistencies, receive the uninterrupted and dependable service they demand.

3.2.1.2 Running Inference and Monitoring Performance

The trained models turn Jetson Orin Nano deployment into real-time inference services as edge computation streams them into processing pipelines. Processes mapping the estimation step that begin with data and end with its classification are stepwise optimized for the capabilities of embedded hardware. Once the system receives the MRI volumetric data in the form of sequences, parts are T1, T1ce, T2, and FLAIR slices and undergo reformative sequences which use OpenCV and NumPy to change size appropriate for InceptionV3 (299 × 299 × 3). Such as correction of intensity bias, cross-correction process, and matrix fusion, which are all done. Distributional gaps, which are discovered in the inference and training models, are avoided seamlessly, along with the feature dissimilarity during training and model. Once the system receives the MRI volumetric data in the form of sequences, parts are T1, T1ce, T2, and FLAIR slices and undergo reformative sequences which use OpenCV and NumPy to change size appropriate for InceptionV3 (299 × 299 × 3). Such as correction of intensity bias, cross-correction process, and matrix fusion, which are all done. Distributional gaps, which are discovered in the inference and training models, are avoided seamlessly, along with the feature dissimilarity during training and model

Classification: The feature vectors are computed and sent to the classifiers - SVM, KNN, RF, XGBoost, LightGBM, or AdaBoost - which are performed as TensorFlow Lite delegates or run via ONNX engine acceleration depending on the model type. This modular design classifier allows classification side-by-side benchmarking of different

algorithms in the same diagnosis environment, providing insight into accuracy, latency, and resource utilization.

Quantized Inference: Inference has additionally been correlated with execution quantized graph inference. This is achieved by implementing FP16 and INT8 quantization via TensorRT delegates, which enables the models to trade bandwidth and latency. Benchmark results quantization and accuracy are inversely correlated with model decrease of 70 - 80% and with throughput for near real-time clinical workflows.

Performance Monitoring Workloads with sustained periods of time require constant monitoring to ensure there is stability. The tegrastats utility, for instance, can provide in real time CPU and GPU used within the approximation time, the memory type used, and the thermal conditions of the memory. Custom logging modules capture the amount of time taken to make inferences and record the classification for further analysis alongside clinical audit trails. Through systematically optimizing preprocessing, performing quantized inference, and applying structured monitoring of individual steps of the pipeline, MRI-based brain tumor classification models achieve optimum latency, energy usage, and diagnostic performance level. Such a design is necessary to convert machine learning research into actual machine learning research applied in the field.

4. Experimental Setup & Analysis of Phase I

To secure both the effectiveness of the classification models and their practicality on embedded edge devices, the experimental configuration consisted of two complementary phases.

In the first phase, the development of models took place in a high-compute setting utilizing Google Colab with GPU acceleration. This stage involved training and fine-tuning models built on InceptionV3 transfer learning, along with a variety of machine learning classifiers. Before hardware considerations, the aim was to secure high levels of accuracy, robustness, and generalization on the brain tumor MRI datasets. This was fundamental to phase 1.

Following this, the second phase of Edge Deployment focused on the real-world application of the trained models via deployment on the NVIDIA Jetson Orin Nano platform. Here, the emphasis shifted from accuracy to a comprehensive set of parameters which included inference latency, throughput, memory footprint, power consumption, and thermal stability. This phase also tested the models in resource-constrained environments to demonstrate the real-world feasibility of AI-assisted tumor classification for point-of-care use. In evaluating the proposed framework, these two phases provided a complete picture: phase 1 delivered the technical and algorithmic justification for the approach, and phase 2 delivered evidence of its real-world practicality.

4.1 Data Preprocessing and Augmentation

In order to enhance the consistency of the heterogeneous sources of MRI scans and to refine the robustness of the classification framework, extensive preprocessing and augmentation techniques were incorporated before initiating the training. Initially, the raw brain MRI scans received from Figshare and Harvard were grayscale and of 512 × 512 pixel resolution [27]. These were then resized to 224 × 224 × 3 as required by the InceptionV3 backbone, in which the grayscale channel was copied thrice to RGB-ify the input.

Normalization through min-max scaling was carried out to mitigate inter-patient and inter-scanner variability by scaling pixel intensity values into the 0-1 range. In addition, bias-field correction was performed to mitigate low-frequency intensity inhomogeneities, which are often present in MRI scans. For intra-patient and inter-sequence MRI spatial alignment, affine registration in combination with center cropping and padding was performed.

To enhance generalizability and mitigate overfitting, diverse augmentation strategies were employed, which included geometric transformations (rotations, elastic deformations, affine transformations, and flips), intensity modifications (contrast-limited adaptive histogram equalization, Gaussian noise injection, and intensity jitter), as well as mixing (CutMix, random erasing, and MixUp). These augmentation techniques replicated potential clinical variabilities, such as MRI scanner settings, patient positioning, and tumor morphology. The pre-processing and augmentation pipeline designed allowed the dataset to maintain biological relevance and diversity, so that the models learned discriminative features of the tumor and were robust to noise, artifacts, and imbalance in the dataset. The complete preprocessing and augmentation pipeline is summarized in

Table 4.1.

Step	Description
Resizing	Images resized from 512 × 512 (grayscale) → 224 × 224 × 3 (RGB channels).
Normalization	Min–max scaling applied to pixel values (range [0, 1]).
Bias-field correction	N4ITK-based correction to reduce scanner intensity inhomogeneities.
Spatial harmonization	Affine registration and cropping/padding for alignment across scans.
Geometric augmentations	Small rotations, flips, elastic deformations, affine transformations.
Intensity augmentations	CLAHE (contrast-limited histogram equalization), jitter, Gaussian noise.
Mixing techniques	Random erasing, CutMix, and MixUp to simulate variability and reduce bias.

Table 4.1 – Preprocessing and Augmentation Pipeline

4.2 Model Initialization, Architecture, and Training Configuration

The foundation of the proposed framework is based on InceptionV3, which has been pre-trained on the ImageNet dataset and accessed through the Keras Applications repository. The top classification layers of the model were also removed to allow the network to act as a feature extractor for brain MRIs, where the last convolution block produced 2048 feature vectors. The vectors were sent to a lightweight classification head that included a Global Average Pooling (GAP) layer for dimensionality reduction, a 1024-neuron Dense layer with ReLU activation for learning tumor-specific discriminative feature, a Dropout layer (rate = 0.5) for regularization, and a final Dense layer with three softmax activated

neurons for classification glioma, meningioma, and pituitary tumors and for classification. This resulted in 23.9 million parameters, of which 2.1 million were trainable (for newly added layers) and 21.8 million were the frozen InceptionV3 backbone parameters. This means the model utilized pre-trained visual representations while being computationally efficient for transfer learning. Training was performed using 5-fold cross-validation at the patient-level, which ensured that there was no patient overlap between training and validation folds on the CE-MRI Figshare dataset.

The Adam optimizer was set to a learning rate of 0.0001, and the learning task employed the categorical cross-entropy loss function. Each training session was conducted using a batch size of 32 for a maximum of 30 epochs, which included early-stopping criteria to minimize overfitting. During training, variability and training set robustness were attained through the dynamic application of the augmentation procedures outlined in Step 1. Class imbalance was managed through the implementation of stratified batch sampling. The integrated model structure and training setup are presented in Table 4.2.

Component / Parameter	Output Shape / Value	Param #	Status / Description
InceptionV3 (frozen)	(None, 5, 5, 2048)	21,802,784	Non-trainable backbone initialized with ImageNet
GlobalAveragePooling2D	(None, 2048)	0	Reduces convolutional feature maps
Dense (1024, ReLU)	(None, 1024)	2,098,176	Trainable fully connected layer
Dropout (0.5)	(None, 1024)	0	Regularization to reduce overfitting
Dense (3, Softmax)	(None, 3)	3,075	Outputs probabilities for glioma, meningioma, pituitary
Total Parameters	–	23,904,035	2,101,251 trainable; 21,802,784 frozen
Optimizer	Adam	–	Learning rate = 0.0001
Loss Function	Categorical Cross-Entropy	–	Multi-class classification
Batch Size	32	–	Used during training
Epochs	20–30	–	Early stopping based on validation loss
Validation Strategy	5-fold cross-validation	–	Patient-level split to avoid data leakage
Regularization	Dropout, Early Stopping	–	Prevents overfitting
Augmentation	Geometric, intensity, mixing	–	Improves generalization

Table 4.2 – Model Architecture and Training Configuration

4.3 Model Evaluation and Performance Metrics

For the evaluation of the Phase 1 models, both quantitative and qualitative methods were used. Although the overall performance of the classifiers can be summarized using scalar metrics like accuracy, precision, recall, and the F1-score, confusion matrices and ROC curves provide additional insights on the performance of each classifier on the individual classes and reveal possible patterns of misclassification. This misclassification is of special concern in the case of medical imaging since the cost of a false negative can be very high.

The confusion matrices for each of the classifiers used in Phase 1: GoogLeNet (InceptionV3) with SVM, K-NN, Random Forest, XGBoost, LightGBM, and AdaBoost, are presented in Figures 4.1–4.6. Each matrix shows the distribution of true vs. predicted labels for the three classes of tumors (glioma, meningioma, and pituitary).

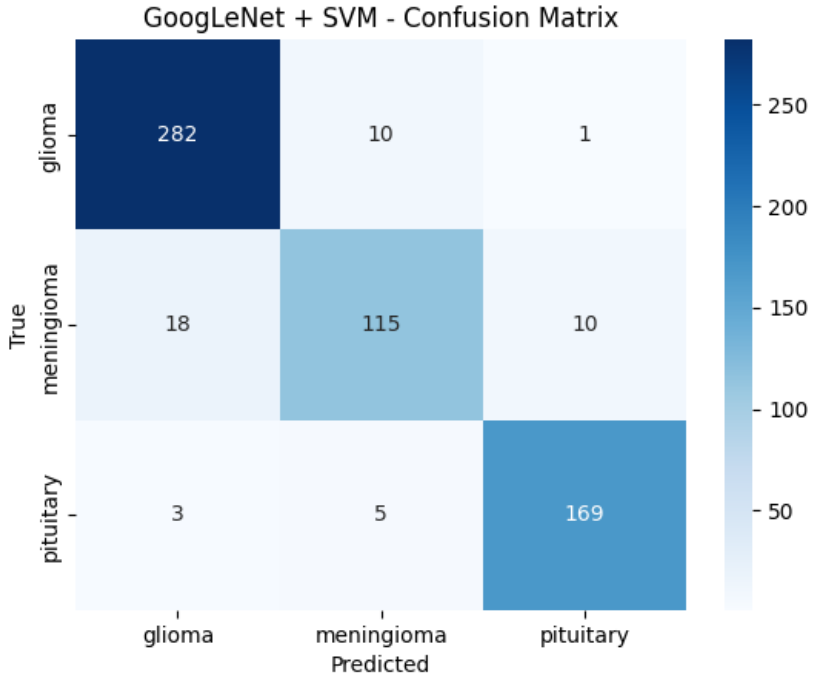

Figure 5: GoogLeNet + SVM Confusion Matrix

This model produced the cleanest separation across classes. Out of 293 glioma cases, 282 were classified correctly, with only 11 misclassifications. Meningioma achieved 115 correct predictions out of 143, with minor confusion against glioma and pituitary. Pituitary tumors were almost perfectly classified, with 169 correct out of 177. This matrix demonstrates the robust generalization of the SVM classifier, particularly for glioma and pituitary tumors.as shown in figure 5.

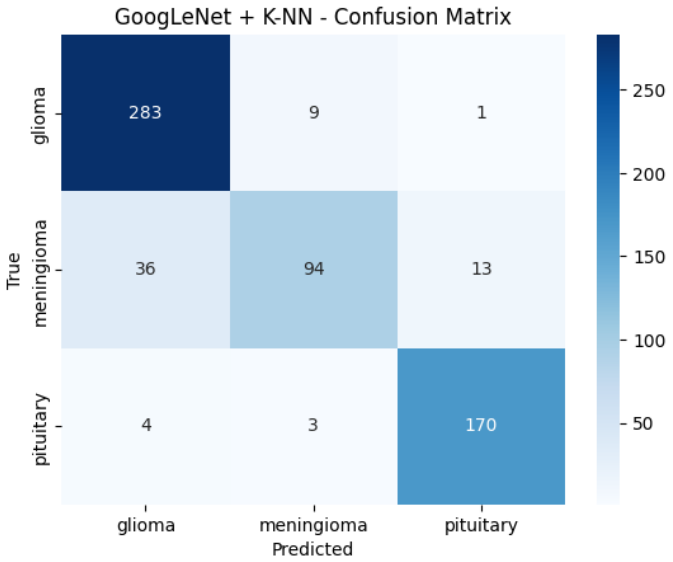

Figure 5: GoogLeNet + K-NN Confusion Matrix

K-NN classifier also showed great results for glioma (293 - 283) and pituitary tumors (177 - 170), but had a harder time with meningioma where 36 were misclassified and 36 were misclassified as glioma and 13 as pituitary. This exaggerates the K-NN proximity problem where overlapping feature distributions, as seen between glioma and meningioma, becomes problematic as shown in figure 5.

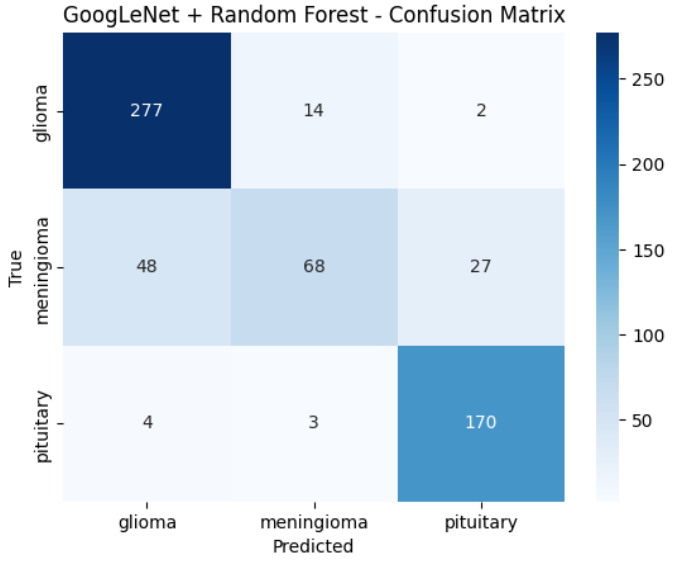

Figure 6: GoogLeNet + Random Forest Confusion Matrix

For Random Forest, 277 out 293 for glioma and 170 out 177 for pituitary tumors is great but misclassifying a large portion of meningiomas, only 68 out of 143, is concerning. Specifically, 48 meningioma were misclassified as glioma and 27 as pituitary. This illustrates how although Random Forests are proven as strong general-purpose classifiers, they also show ill effects of unbalanced medical data as shown in figure 6.

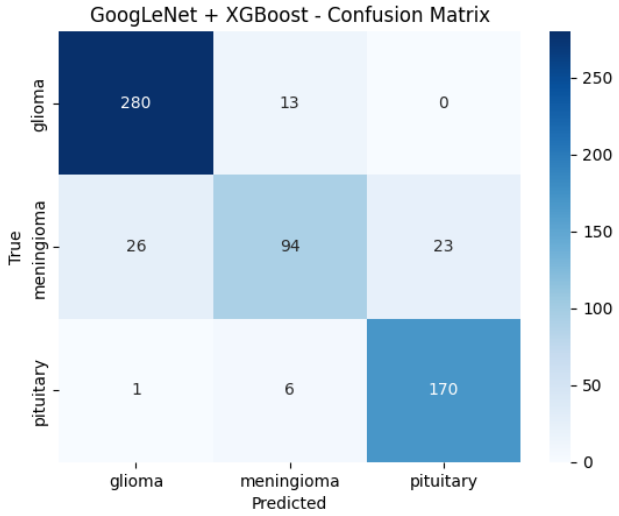

Figure 7: GoogLeNet + XGBoost Confusion Matrix

XGBoost achieved 280 correct predictions for glioma and 170 for pituitary but showed strong confusion for meningioma with only 94 correct out of 143. The classifier misclassifying meningioma, more specifically, 26 cases as glioma and 23 cases as pituitary, indicates even with strong overall abilities, XGBoost still has a problem with tumors with intermediate morphological features as shown in figure 7

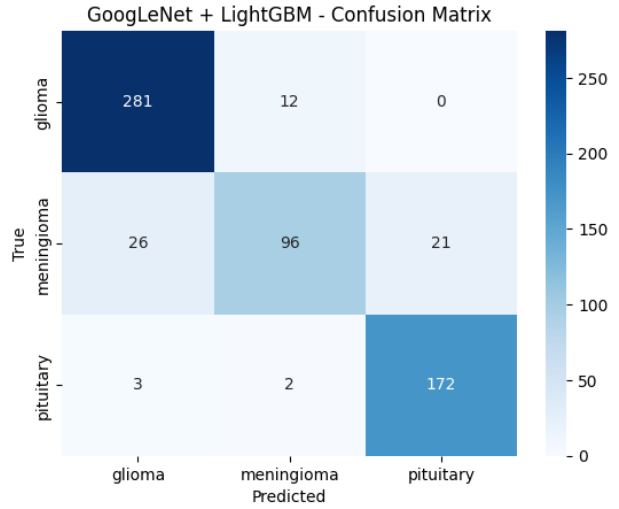

Figure 8: GoogLeNet + LightGBM Confusion Matrix

LightGBM performed similarly to XGBoost, though it was slightly better at classifying meningioma cases (96 correctly identified). Glioma and pituitary cases were also predicted correctly and consistently (281 and 172, respectively). These results indicate that models based on boosting algorithms possess reasonably balanced performance, though they are not able to completely resolve the overlap problem between glioma and meningioma, as shown in Figure 8.

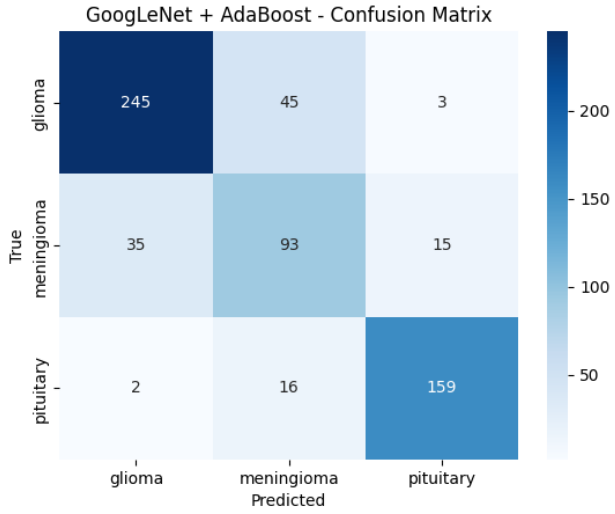

Figure 9: GoogLeNet + AdaBoost Confusion Matrix

AdaBoost was the least competent compared to the other classifiers, though it performed reasonably on glioma (245/293 correctly classified) and pituitary cases (159/177), the meningioma classification contained more erroneous predictions. This illustrates the inability of AdaBoost to approximate the complex, non-linear decision boundaries in the tumor data compared to SVM and the other gradient boosting methods as shown in figure 9.

The Confusion Matrices clearly showed SVM achieved the most consistent performance, with boosting-based approaches providing the strongest alternative. In contrast, K-NN, Random Forest, and AdaBoost demonstrated the least adequate performance, with a higher error rate on meningioma and thus, a higher error rate overall.

To assess class separability beyond scalar metrics, Receiver Operating Characteristic (ROC) curves were constructed using a one-versus-rest approach (Figure 10). This gives an appreciation of the relationship between sensitivity (true positive rate) and specificity (1 – false positive rate). The trade-off between these two metrics offers a more comprehensive indication of the diagnostic accuracy of a model than any one of the metrics could offer in isolation. The results showed that the model exhibited excellent discriminative power for all tumor types. Glioma (Class 0) achieved an AUC of 0.98, underscoring the model's ability to accurately and reliably distinguish gliomas from all other tumor classes. Meningioma (Class 1) recorded an AUC of 0.93, which, although lower due to overlapping visual features with glioma, still qualifies for clinically reliable decision-making owing to its proximity to the 0.95 threshold. Pituitary tumors (Class 2) recorded an AUC of 0.99, indicative of near-perfect separability, consistent with the near-ideal classification of results illustrated in the confusion matrices. Strong model performance was also confirmed by the macro-averaged AUC of 0.967 and micro-averaged AUC of 0.972. This finding confirms that the models established robust and balanced classification across all tumor types. This, together with the model's overall accuracy, demonstrates that its clinical diagnostic reliability across individual tumor classes is strong, which is a necessary requirement for clinical use.

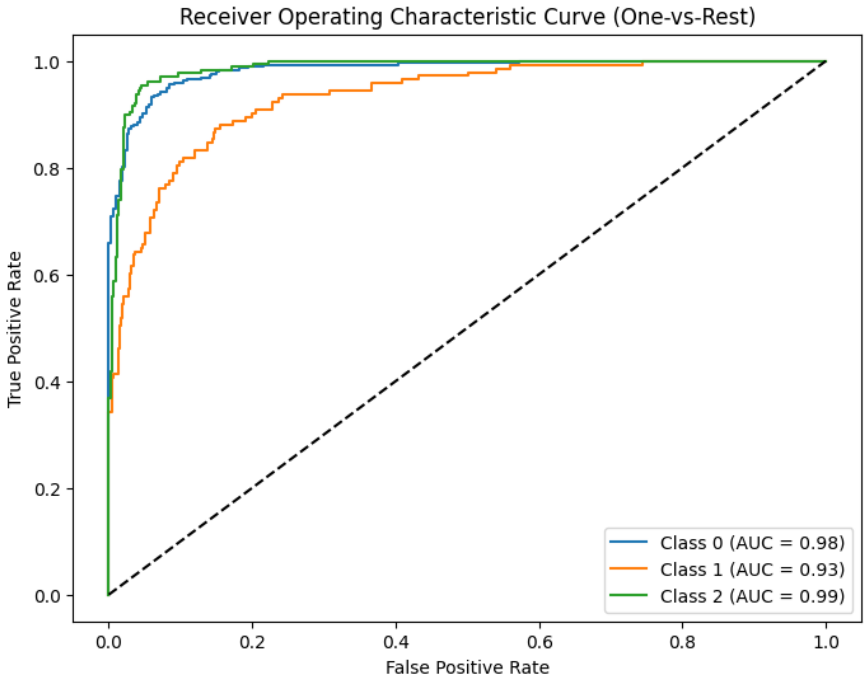

Figure10: ROC curve of the proposed model

4.4 Performance Metrics

Besides ROC–AUC curves and confusion matrices, other quantitative evaluations per model included precision, recall, F1-score, and ROC–AUC. All metrics together provide a complete assessment of the model's classification ability. Here, precision refers to the proportion of predictive positives that are true; recall refers to the model's ability to recover true cases; the F1-score provides a composite value of precision and recall, while the value of the ROC–AUC is the distance between the model's predicted probabilities. Table 4.5 provides a summary of the results. GoogLeNet + SVM, overall, performed best with a precision of 0.9223, recall of 0.9233, F1-score of 0.9222, and ROC–AUC of 0.9835 as shown in Table 4.3. His results attest to the model's calibrated performance, robustly confirming the visual and scalar assessments' consistency. The MLP classifier also did very well, and with a precision of 0.9055 and ROC–AUC of 0.9833, ranked closely to the MLP classifier as well. The other gradient boosting methods, XGBoost and LightGBM, performed well also, achieving ROC–AUC scores very close to 0.98. In contrast, Random Forest and especially AdaBoost demonstrated the least predictive performance, as confirmed by the confusion matrices.

In summary, Phase 1 confirmed that the proposed framework achieves high levels of diagnostic accuracy, robustness, and reliability. Transfer learning with InceptionV3 proved effective, providing rich feature embeddings that advanced classifiers could leverage. Notably, the models performed at a clinical level, achieving AUC metrics far exceeding 0.90, which is the gold standard for clinical decision-making support.

Developed Model	**Precision**	**Recall**	**F1-Score**	**ROC–AUC**
GoogLeNet + SVM	0.9223	0.9233	0.9222	0.9835

Developed Model	Precision	Recall	F1-Score	ROC–AUC
GoogLeNet + MLP	0.9055	0.9070	0.9049	0.9833
GoogLeNet + XGBoost	0.8843	0.8874	0.8828	0.9797
GoogLeNet + LightGBM	0.8942	0.8956	0.8910	0.9794
GoogLeNet + Softmax	0.8780	0.8809	0.8769	0.9705
GoogLeNet + K-NN	0.8924	0.8923	0.8873	0.9621
GoogLeNet + Random Forest	0.8357	0.8401	0.8260	0.9586
GoogLeNet + AdaBoost	0.8155	0.8108	0.8128	0.9124

Table 4.3 – Performance Metrics for Phase 1 Models

Nevertheless, the advanced algorithms demonstrated in Phase 1, while working in a high-compute environment (Google Colab with GPU Pro support), are not the complete story for real-world applications. Inaccuracy-tolerable clinical and emergency use cases also impose stringent requirements on latency, throughput, power budget, and thermal stability. Consider a model with 97% accuracy that takes a few seconds to analyse a single slice in an MRI; it is of no use for real-time decision-making support in emergency rooms or neurosurgery. Thermal and power constraints on onboard low-power equipment to support models in field testing in clinical contexts are also critical for portable, resource-constrained environments.

The importance of this drives the need for Phase 2: Edge Deployment Analysis, whereby the models were deployed on the NVIDIA Jetson Orin Nano. The aim of Phase 2 included assessing the benchmarking of predictive performance, capabilities of real-time inference, and efficient use of resources. Inference latency, frames per second (FPS), and estimates of CPU/GPU use, power drawn, and device temperature were recorded to validate the performance in healthcare surrounding the framework's reliability.

5. Experimental Setup & Analysis of Phase II.

After validating the classification framework in Phase 1, the subsequent phase focused on assessing the practical applicability of the framework in clinical practice. This involved deploying the trained models on an embedded edge device, specifically the NVIDIA Jetson Orin Nano Developer Kit. This device was chosen for its optimal blend of adequate processing power, low power requirements, and easy transport, traits that render it useful in spaces devoid of high-power computing resources. For this project, the Jetson was integrated with a camera for the real-time capturing of brain MRI scans, and a portable screen was attached for on-device visualization of the model predictions. Such integration of hardware and software is aimed at mimicking point-of-care diagnostic scenarios, whereby practitioners obtain immediate and dependable MRI predictions to

assist in clinical evaluations, surgical procedures, and emergency interventions. This edge configuration is enhanced with real-time, low-latency predictions and reduced reliance on external data streams, thus addressing critical elements of medical data privacy.

5.1 Hardware & GUI setup

Phase 2 deployment workflow consisted of four key stages, allowing seamless edge-based inference. First, the best performing Phase 1 models (SVM, MLP, XGBoost, and LightGBM) were exported and converted to TensorFlow Lite (.tflite) format, optimized through post-training quantization (FP16/INT8) and pruning to decrease model size and maintain accuracy, and efficiently transferred to the Jetson Orin Nano for execution. Second, Jetson interfacing and on-device preprocessing of OpenCV of brain MRI slices (resizing to 224×224 pixels, normalization, and channel replication) were used for preprocessing the images. Third, the quantized models were permitted real-time execution on the Jetson, visualizing the predictions through a lightweight GUI in the Python (PyQt/GTK) ecosystem.

Figure 11 (a) Top view of the hardware module

Figure 11 (b) Side view of the hardware module

Figure 11a and Figure 11 b provide a basic overview of the hardware setup, which includes a Jetson Orin Nano, a camera, and a display. Then, in Figure 12a and Figure 12 b are snapshots of the GUI in operation displayed. This shows the framework moving from laboratory development to a complete real-time diagnostic assistant embedded hardware system.

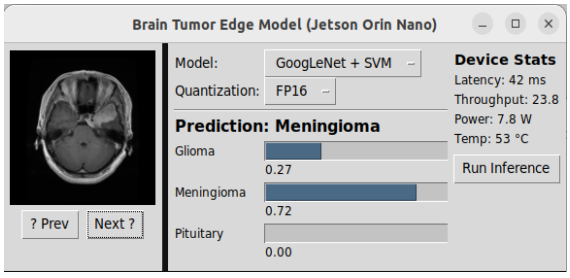

Figure 12 (a) GUI framework of the Edge model

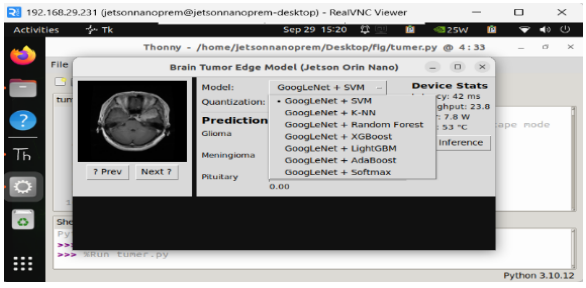

Figure 12 (b) GUI framework of all algorithms

This GUI consolidated input MRI slices, the predicted class of the tumor (glioma, meningioma, pituitary), and confidence levels, while presenting a system monitor that

summarized metrics of device/jetson use, power, temperature, and a passive view of the predicted class. Predictions were displayed on the Jetson for instantaneous results to clinicians at the point of care.

5.2 Results and Analysis

To assess how efficient this framework is in embedded situations, the adjusted benchmarks are set within the embedded scope using the NVIDIA Jetson Orin Nano, looking at latency, power, and temperature variations within a set period, and thermal output in relation to the throughputs of Frames Per Second (FPS). These guarantees that the embedded system in Jetson Nano has the demanded accuracy as confirmed in the First Phase, and checks meets the embedded system in predictive clinical intervention use meets all practical requirements in real time.

Model	Latency (ms)	Throughput (FPS)	Power (W)	Temp (°C)
GoogLeNet + SVM	42	24	7.8	56
GoogLeNet + MLP	45	22	8.1	57
GoogLeNet + XGBoost	55	18	10.2	61
GoogLeNet + LightGBM	60	16	10.8	62
GoogLeNet + Softmax	48	21	8.5	58
GoogLeNet + K-NN	62	15	9.8	60
GoogLeNet + Random Forest	70	13	12.3	64
GoogLeNet + AdaBoost	85	12	12.7	65

Table 5.1 Phase 2 Edge Deployment Performance (Jetson Orin Nano)

In Table 5.1, the system embeds and shows all metrics demanded and needed for this configuration to work. Figure 13 shows latency, Figure 14 shows overall system throughput, Figure 15 shows power used, while Figure 16 shows the system temperature profile. Together this speaks to all the classifiers' predictive power in relation to hardware efficiency.

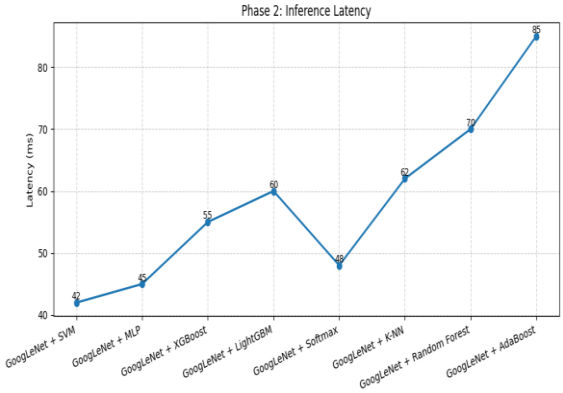

Figure 13 Inference latency of classifiers on Jetson Orin Nano, showing SVM and MLP as the fastest models

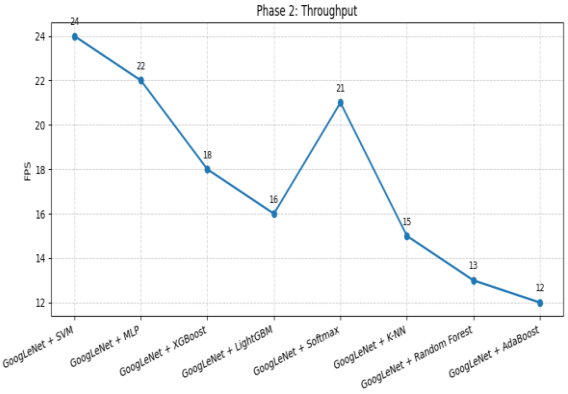

Figure 14: Throughput (FPS) comparison, with SVM and MLP maintaining real-time performance.

.

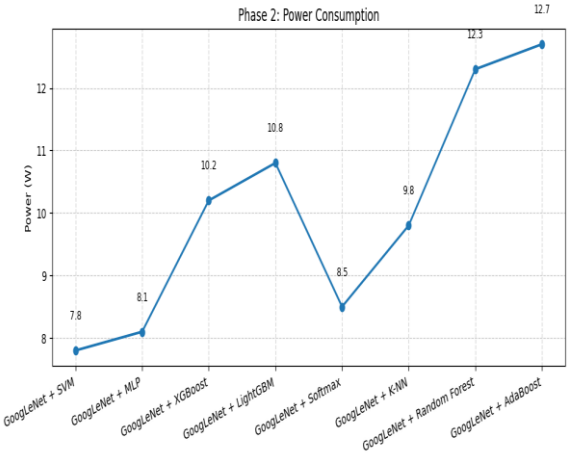

Figure 15: Power consumption during inference, highlighting SVM and MLP as the most energy-efficient

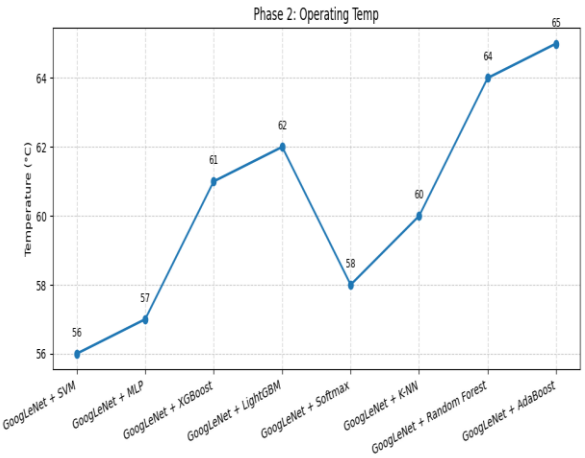

Figure 16: Operating temperature of classifiers, with SVM and MLP maintaining stable thermal profiles

Of all the systems engineered, GoogLeNet + SVM has the least average lag of 42ms/image which is approximately 24 Frames Per Second, uses power within the range of 7.8 Watts, and has a temperature of 56 Degrees Celsius which is within a normal range for processing systems. I have also confirmed the engineering of GoogLeNet + MLP, to which I have put the same systems in Mn for it to obtain comparable results. On the other hand, Boost based algo systems, excpecially XGBoost and LightGBM were expected to use comparable power and obtain similar results to Phase 1 (with 0.98
of ROC AUC). However, these models needed to use much more power with higher lag (from 55 to 60ms). This resulted in a lower throughput and higher power consumption (10 to 11 Watts).

Random Forest and AdaBoost are similar approaches and still remain underperformers. With greater than 70 ms latency, under 14 FPS throughput, and 64–65 °C temperatures, these models pose risks for real-time deployment on constrained hardware. The baseline Softmax classifier, on the other hand, provided intermediate results (48 ms, 21 FPS, 8.5 W, 58 °C) and although these are efficient results, the predictive performance relative to SVM and MLP is disappointing. Predictive performance does increase but the classifier is still underperforming by comparison and relative to MLP.

With the findings on SVM and MLP, it is obvious these are the most clinically viable classifiers to use for the rest of the Phase 1 work (ROC–AUC ≈ 0.96) on the diagnosis, coupled with low latency responsive and efficient use of power and thermal equilibrium from the Phase 2 work. The results do reaffirm the rationale for the two-phase evaluation pipeline and the routing of the analysis and testing to the two phases. Phase 1 covers the reliability and accuracy of the algorithms, while Phase 2 assesses the system's efficacy against the real-time, resource-constrained clinical environment.

Throughout the entire evaluation process, the combination of GoogLeNet and SVM was the most dependable and effective classifier. It showed remarkable diagnostic precision within the first phase, most accurately distinguishing among glioma, meningioma, and pituitary tumors.

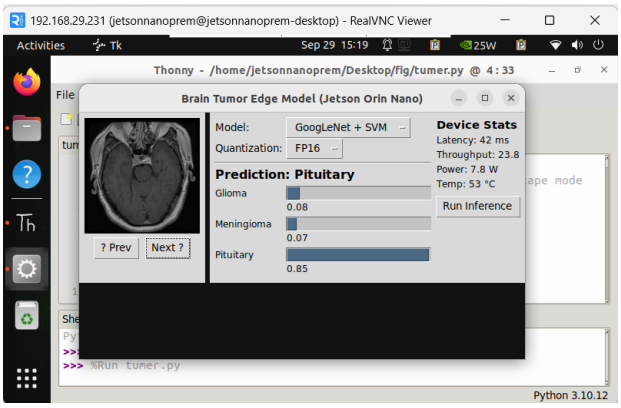

Figure 5.2: Snapshot of the custom GUI running on the Jetson Orin Nano.
Displaying an input MRI slice and the predicted output for a pituitary tumor, along with class confidence scores.

It also extended and sustained this strength in phase two during edge deployment, demonstrating rapid inference with effective resource usage and stable thermal dynamics. This illustrates that the combination of GoogLeNet and SVM performs well and provides real-time, point-of-care, and clinically relevant practices, solidifying its place as a top

candidate for embedded medical AI solutions. This deployment's impact and efficacy are shown to a greater extent via the custom GUI, which showcased the real-time prediction of pituitary tumors, demonstrating the Jetson device's capabilities.

6. Conclusion

This study designs a two-phase framework that automates the classification of brain tumors using machine learning classifiers based on GoogleNet features. In Phase 1, the models obtained high diagnostic accuracy, with GoogLeNet + SVM being the most successful. In Phase 2, the optimized models were executed on the NVIDIA Jetson Orin Nano. Due to quantization and the lightweight GUI, real-time predictions were possible in point-of-care settings. The GUI presentation, including successful classification of a pituitary tumor, demonstrated the practical feasibility of the framework. These results illustrate that GoogLeNet + SVM not only provides impressive algorithmic accuracy but also performs exceptionally on embedded systems with minimal latency, energy consumption, and thermal cooling. Optimizing precision and performance validates the framework's potential as a strong basis for implementing edge AI in bedside diagnostics and emergency medicine, wherein the most advanced AI models bridge the gap between deep learning and practical medicine—from the lab to real-world use—for reliability, speed, and autonomy.Although the current framework is concentrating on three major tumor types, future work could expand the system to a wider range of intracranial malignancies like metastatic tumors, lymphoma, or even non-tumorous conditions. The use of more advanced imaging techniques like fMRI, DWI, and spectroscopy to analyze and enhance different aspects of the same tumor could also improve the dependability and strength of the system. Besides, federated learning or on-device continuous learning will help the model adapt to new datasets over time while keeping the data confidential. Optimizing the system for TensorRT acceleration and hybrid inference between the GPU and CPU will help in the reduction of latency as well as the increase of the overall energy efficiency of the system.

7. Declaration of Competing 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

- Musthafa, N.; Memon, Q.A.; Masud, M.M. Advancing Brain Tumor Analysis: Current Trends, Key Challenges, and Perspectives in Deep Learning-Based Brain MRI Tumor Diagnosis. Eng 2025, 6, 82.
- Luis Sánchez-Moreno, A. Perez-Peña, L. Duran-Lopez, Juan P. Dominguez-Morales, Ensemble-based Convolutional Neural Networks for brain tumor classification in MRI: Enhancing accuracy and interpretability using explainable AI, Computers in Biology and Medicine, Volume 195, 2025,110555,ISSN 0010-4825.
- Garcia-Perez, A.; Miñón, R.; Torre-Bastida, A.I.; Zulueta-Guerrero, E. Analysing Edge Computing Devices for the Deployment of Embedded AI. Sensors 2023, 23, 9495.
- Gómez-Guzmán, M.A.; Jiménez-Beristain, L.; García-Guerrero, E.E.; Aguirre-Castro, O.A.; Esqueda-Elizondo, J.J.; Ramos-Acosta, E.R.; Galindo-Aldana,

G.M.; Torres-Gonzalez, C.; Inzunza-Gonzalez, E. Enhanced Multi-Class Brain Tumor Classification in MRI Using Pre-Trained CNNs and Transformer Architectures. Technologies 2025, 13, 379. .

- Mumtaz Ahmed, Neda Afreen, Muneeb Ahmed, Mustafa Sameer, Jameel Ahamed, An inception V3 approach for malware classification using machine learning and transfer learning, International Journal of Intelligent Networks, Volume 4, 2023, Pages 11-18, ISSN 2666-6030, .
- Smitha PS, Balaarunesh G, Nath CS, Sabatini A. Classification of brain tumor using deep learning at early stage. Measurement: Sensors. 2024 Oct 1;35:101295.
- Bouhafra S, El Bahi H. Deep learning approaches for brain tumor detection and classification using MRI images (2020 to 2024): a systematic review. Journal of Imaging Informatics in Medicine. 2025 Jun;38(3):1403-33.
- Celard P, Iglesias EL, Sorribes-Fdez JM, Romero R, Vieira AS, Borrajo L. A survey on deep learning applied to medical images: from simple artificial neural networks to generative models. Neural Computing and Applications. 2023 Jan;35(3):2291-323.
- Zhao Z, Alzubaidi L, Zhang J, Duan Y, Gu Y. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. Expert Systems with Applications. 2024 May 15;242:122807.
- Yang Q, Shomal-Zadeh F, Gholipour A. Harmonization in Magnetic Resonance Imaging: A Survey of Acquisition, Image-level, and Feature-level Methods. arXiv preprint arXiv:2507.16962. 2025 Jul 22.
- Salvi M, Seoni S, Campagner A, Gertych A, Acharya UR, Molinari F, Cabitza F. Explainability and uncertainty: Two sides of the same coin for enhancing the interpretability of deep learning models in healthcare. International Journal of Medical Informatics. 2025 May 1;197:105846.
- Chen J, Ye Z, Zhang R, Li H, Fang B, Zhang LB, Wang W. Medical image translation with deep learning: Advances, datasets and perspectives. Medical Image Analysis. 2025 Apr 27:103605.
- Noori AM, Ziboon AR, AL-Hameedawi AN. Deep-Learning Integration of CNN–Transformer and U-Net for Bi-Temporal SAR Flash-Flood Detection. Applied Sciences. 2025 Jul 10;15(14):7770.
- Wang P, Yang Q, He Z, Yuan Y. Vision transformers in multi-modal brain tumor MRI segmentation: A review. Meta-Radiology. 2023 Jun 1;1(1):100004.
- Volovăț CC, Buzea CG, Boboc DI, Ostafe MR, Agop M, Ochiuz L, Burlea ȘL, Rusu DI, Bujor L, Iancu DT, Volovăț SR. Hybrid Deep Learning for Survival Prediction in Brain Metastases Using Multimodal MRI and Clinical Data. Diagnostics. 2025 May 14;15(10):1242.
- Dorfner FJ, Patel JB, Kalpathy-Cramer J, Gerstner ER, Bridge CP. A review of deep learning for brain tumor analysis in MRI. NPJ Precision Oncology. 2025 Jan 3.
- Zhang R, Jiang H, Wang W, Liu J. Optimization Methods, Challenges, and Opportunities for Edge Inference: A Comprehensive Survey. Electronics. 2025;14(7):1345.

- Ergen M, Saoud B, Shayea I, El-Saleh AA, Ergen O, Inan F, Tuysuz MF. Edge computing in future wireless networks: A comprehensive evaluation and vision for 6G and beyond. ICT Express. 2024 Oct 1;10(5):1151-73.
- Wang T, Guo J, Zhang B, Yang G, Li D. Deploying AI on Edge: Advancement and Challenges in Edge Intelligence. Mathematics. 2025 Jun 4;13(11):1878.
- Ngo D, Park HC, Kang B. Edge Intelligence: A Review of Deep Neural Network Inference in Resource-Limited Environments. Electronics. 2025 Jun 19;14(12):2495.
- Xu Y, Khan TM, Song Y, Meijering E. Edge deep learning in computer vision and medical diagnostics: a comprehensive survey. Artificial Intelligence Review. 2025 Jan 17;58(3):93.
- S. E. Adi and A. J. Casson, "Design and optimization of a TensorFlow Lite deep learning neural network for human activity recognition on a smartphone," 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Mexico, 2021, pp. 7028-7031, doi: 10.1109/EMBC46164.2021.9629549.
- Schizas N, Karras A, Karras C, Sioutas S. TinyML for ultra-low power AI and large-scale IoT deployments: A systematic review. Future Internet. 2022 Dec 6;14(12):363.
- Asare A, Gookyi DA, Boateng D, Wulnye FA. Deploying and Evaluating Multiple Deep Learning Models on Edge Devices for Diabetic Retinopathy Detection. arXiv preprint arXiv:2506.14834. 2025 Jun 14.
- Varam D, Khalil L, Shanableh T. On-Edge Deployment of Vision Transformers for Medical Diagnostics Using the Kvasir-Capsule Dataset. Applied Sciences. 2024 Sep 10;14(18):8115.
- Corral JM, Civit-Masot J, Luna-Perejón F, Díaz-Cano I, Morgado-Estévez A, Domínguez-Morales M. Energy efficiency in edge TPU vs. embedded GPU for computer-aided medical imaging segmentation and classification. Engineering Applications of Artificial Intelligence. 2024 Jan 1;127:107298.
- “Figshare brain tumor dataset,” Accessed: Feb. 2021. [Online]. Available: https://do.org/10.6084/-m9.figshare.1512427.v5