BRAIN TUMOR DETECTION AND VISUALIZATION USING DEEP LEARNING

A PROJECT REPORT

Submitted by

GAYATHRI DEVI P.M   **RA2311026020004**

SACHEEN SUBANIDHI   **RA2311026020007**

GIRIJESH S   **RA2311026020008**

Under the guidance of

DR RUBIN ROSE

(Assistant Professor / CSE-AIML)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING
With specialization in
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM, CHENNAI-600089

OCT 2025

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University U/S 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled **"BRAIN TUMOR DETECTION AND VISUALIZATION USING DEEP LEARNING"** is the bonafide work of **GAYATHRI DEVI P.M [REG NO: RA2311026020004], SACHEEN SUBANIDHI [REG NO: RA2311026020007], GIRIJESH S [REG NO: RA2311026020008]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an occasion on this or any other candidate.

SIGNATURE
DR RUBIN ROSE

Assistant Professor

Computer Science and Engineering,
SRM Institute of Science and Technology,

Ramapuram, Chennai.

SIGNATURE
Dr. N. SANKAR RAM, M.E., Ph.D.,

Professor and Head

Computer Science and Engineering,
SRM Institute of Science and Technology,

Ramapuram, Chennai.

Submitted for the project viva-voce held on _____ at SRM Institute of Science and Technology, Ramapuram, Chennai -600089.

INTERNAL EXAMINER 1   **INTERNAL EXAMINER 2**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM, CHENNAI - 89

DECLARATION

We hereby declare that the entire work contained in this project report titled "**BRAIN TUMOR DETECTION AND VISUALIZATION USING DEEP LEARNING**" has been carried out by **GAYATHRI DEVI P.M [REG NO: RA2311026020004], SACHEEN SUBANIDHI [REG NO: RA2311026020007], GIRIJESH S [REG NO: RA2311026020008]** at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089, under the guidance of **Dr. RUBIN ROSE Assistant Professor**, Department of Computer Science and Engineering.

Place: Chennai
Date:

<div align="right">

GAYATHRI DEVI P.M

SACHEEN SUBANIDHI

GIRIJESH S

</div>

ABSTRACT

The early and accurate diagnosis of brain tumors is a cornerstone of modern neuro-oncology, critically influencing treatment efficacy and patient prognosis. While Magnetic Resonance Imaging (MRI) provides unparalleled soft-tissue contrast for visualization, its manual interpretation is a labor-intensive process, susceptible to subjective variability and diagnostic delays. This project presents a comprehensive, full-stack web application that leverages deep learning to automate the classification of brain tumors from MRI scans, aiming to serve as a robust decision support tool for clinicians. Our system is designed to classify brain MRI scans into four distinct and clinically relevant categories: Glioma, Meningioma, Pituitary tumor, and No Tumor.

To achieve high diagnostic accuracy, we employ a transfer learning methodology centered on the Xception Convolutional Neural Network (CNN) architecture. By fine-tuning a model pre-trained on the extensive ImageNet dataset. This approach allows the model to learn intricate patterns indicative of different tumor types from a comparatively smaller medical dataset.

A pivotal feature of our system is the integration of Explainable AI (XAI) through Gradient-weighted Class Activation Mapping (Grad-CAM). Addressing the "black box" problem inherent in many deep learning models, Grad-CAM generates intuitive visual heatmaps that highlight the specific regions in an MRI scan that the model found most salient for its prediction. This layer of transparency is crucial for clinical validation, allowing medical professionals to scrutinize and trust the model's reasoning process.

The backend infrastructure is built with Python, using the Flask framework to serve a REST API and TensorFlow/Keras for the heavy lifting of image processing, model inference, and Grad-CAM generation. The frontend is a modern, responsive, and intuitive user interface developed with React and styled with Tailwind CSS, ensuring a seamless user experience across devices. This allows users to easily upload MRI scans and receive real-time predictions and visual explanations without the need for specialized software. Our model achieves a classification accuracy exceeding 95% on a held-out test dataset, demonstrating its potential as a powerful, reliable, and interpretable assistive tool for radiologists and oncologists in the complex and high-stakes workflow of brain tumor diagnosis.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The diagnosis and classification of brain tumors represent one of the most critical and challenging tasks in modern oncology. As abnormal growths of cells within the brain, tumors can be either benign (non-cancerous) or malignant (cancerous), with each type necessitating a distinct and timely treatment protocol. An accurate and early identification of the tumor's nature is paramount for determining the most effective treatment strategy—which can range from surgical resection and targeted radiation to systemic chemotherapy—and is directly correlated with patient prognosis and quality of life. For instance, a Meningioma is often benign and slow-growing, sometimes only requiring observation, whereas a Glioma is typically malignant and infiltrative, demanding aggressive and immediate intervention.

Magnetic Resonance Imaging (MRI) has established itself as the gold standard for non-invasive brain tumor diagnosis. Its superiority over other modalities, such as Computed Tomography (CT), lies in its exceptional soft-tissue contrast, which allows for detailed visualization of brain structures and pathological tissues without the use of ionizing radiation. Different MRI sequences (T1-weighted, T2-weighted, FLAIR) provide complementary information about the tumor's size, location, and relationship with surrounding structures. However, the reliance on manual interpretation of these scans by radiologists, while standard practice, is fraught with inherent limitations. The sheer volume of medical imaging data is rapidly increasing, placing a significant burden on specialists. This can lead to diagnostic fatigue and introduces the risk of subjective error and inter-observer variability, where interpretations may differ between radiologists based on experience and individual judgment.

In recent years, the convergence of powerful computational resources, particularly Graphics Processing Units (GPUs), and the availability of large-scale medical datasets has catalyzed a revolution in medical image analysis, led by the field of deep learning. Convolutional Neural Networks (CNNs), a class of deep learning models inspired by the human visual cortex, have become the state-of-the-art for a variety of computer vision tasks. Their ability to automatically learn hierarchical feature representations from data—starting from simple edges and textures and building up to complex, abstract patterns—enables them to

discern subtle pathological indicators that may be missed by the human eye. This project harnesses the formidable power of CNNs to develop a fully automated system for the classification of brain tumors from MRI scans.

Our primary objective is to engineer a reliable, accurate, and user-friendly tool designed to function as a decision support system for medical professionals. The system is architected to classify an uploaded MRI scan into one of four clinically significant categories: Glioma, Meningioma, Pituitary tumor, or No Tumor. To achieve state-of-the-art performance, we employ a powerful CNN architecture known as Xception, which is renowned for its computational efficiency and high accuracy. We utilize a transfer learning approach, adapting a model pre-trained on the vast ImageNet dataset to our specialized medical imaging task. This allows the model to leverage its generalized visual knowledge and fine-tune it to the specific nuances of brain MRI scans.

A significant and well-documented challenge with deploying deep learning models in high-stakes domains like healthcare is their inherent "black box" nature. A model that provides a prediction without justification is unlikely to be trusted by clinicians who are ultimately responsible for patient care. To directly address this critical issue, our system incorporates a cutting-edge Explainable AI (XAI) technique known as Gradient-weighted Class Activation Mapping (Grad-CAM). This method provides a transparent window into the model's decision-making process by generating a visual heatmap, or "attention map," that highlights the specific regions of the input image the model found most influential in its classification. This interpretability is crucial for building trust, allowing clinicians to validate that the model is focusing on relevant pathological areas and not relying on spurious artifacts.

The final system is delivered as a cohesive, full-stack web application, ensuring maximum accessibility and ease of use. This allows clinicians to interact with the powerful backend model through a simple web browser, without the need for specialized software or hardware. By integrating high diagnostic accuracy with essential model transparency, this project aims to provide a practical and powerful tool to assist in the brain tumor diagnostic workflow.

Chapter 2

LITERATURE REVIEW

The application of computational methods to brain tumor classification is a field with a rich history, evolving from classical machine learning to the deep learning paradigms that dominate today. This evolution has been driven by the increasing availability of data and computational power, and a continuous quest for higher accuracy and greater automation.

Early Approaches: Before the deep learning era, the primary approach to brain tumor classification involved a two-stage process: manual feature extraction followed by classification using traditional machine learning algorithms. Researchers would use their domain knowledge to engineer features from MRI scans that were believed to be discriminative. These features often included statistical measures of texture (e.g., from Gray-Level Co-occurrence Matrices), shape descriptors, and intensity histograms. Once extracted, these feature vectors were fed into classifiers like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forests. For instance, a study by **Zacharaoki et al. [6]** demonstrated the use of texture features to differentiate between tumor types with moderate success. While these methods laid important groundwork, they were fundamentally limited. The process was labor-intensive, and the performance of the system was heavily dependent on the quality and relevance of the hand-crafted features, which often failed to capture the full complexity of the data.

The Deep Learning Revolution: End-to-End Feature Learning :The paradigm shifted dramatically with the success of AlexNet in the 2012 ImageNet competition **(Krizhevsky et al. [1])**. This marked the beginning of the deep learning revolution in computer vision. Researchers in medical imaging quickly began to adapt CNNs for their tasks. Unlike previous methods, CNNs could learn relevant features directly from the pixel data in an end-to-end fashion, eliminating the need for manual feature engineering. Early work, such as that by **Cireşan et al. [2]**, showed that deep networks could significantly outperform traditional methods on various medical image analysis tasks.

Transfer Learning and Advanced Architectures Training a deep CNN from scratch requires an enormous amount of labeled data, which is often scarce in the medical domain. This challenge was largely overcome by the adoption of transfer learning. This technique involves taking a model pre-trained on a large dataset like ImageNet and fine-tuning it on the smaller,

specific medical dataset.

Numerous studies have validated this approach for brain tumor classification. Models like VGG16, with its simple and deep architecture, and ResNet, which introduced residual connections to combat the vanishing gradient problem in very deep networks, were successfully applied. More advanced architectures continued to push the boundaries of performance. The Inception architecture introduced the idea of using parallel convolutional filters of different sizes within the same module, allowing the network to capture features at multiple scales. Our chosen model, Xception **(Chollet, F. [3])**, builds upon this concept. It stands for "Extreme Inception" and proposes that cross-channel correlations and spatial correlations can be decoupled. It replaces the standard Inception modules with depthwise separable convolutions, which are significantly more parameter-efficient and have been shown to yield superior performance on many image classification benchmarks.

The Imperative of Explainability: Opening the Black Box While the accuracy of deep learning models became undeniable, their "black box" nature posed a major barrier to clinical adoption. A prediction without an explanation is of limited use in a field where decisions have life-or-death consequences. This spurred the growth of Explainable AI (XAI). One of the most influential techniques in this area is Gradient-weighted Class Activation Mapping (Grad-CAM), proposed by **Selvaraju et al. [4]**. Grad-CAM produces a coarse localization map highlighting the important regions in the image for a specific prediction. It works by using the gradients of the target class flowing into the final convolutional layer. Because it is a gradient-based method, it is applicable to a wide range of CNN-based models without requiring any architectural changes or re-training. Its utility has been demonstrated in numerous medical imaging studies, such as the work by **Kim & Lee [5]** on interpreting chest X-rays.

Our work is situated at the confluence of these research streams. We combine a high-performance, efficient CNN architecture (Xception) with a proven transfer learning strategy and integrate a state-of-the-art XAI technique (Grad-CAM). By packaging this entire pipeline into an accessible web application, we aim to create a tool that is not only technically sound but also practical and trustworthy for clinical use.

Chapter 3

METHODOLOGY

Our proposed system is a comprehensive, full-stack solution for brain tumor detection, comprising a machine learning backend for analysis and a web-based frontend for user interaction. The methodology is designed to be robust, accurate, and interpretable, following best practices in software engineering and machine learning.

General Architecture:

The system follows a classic client-server architecture. The frontend, a single-page application built with React, provides the user interface for uploading MRI images. The backend, a Flask web server, exposes a REST API to handle image analysis requests. When an image is submitted, the backend processes it, runs it through our deep learning pipeline, and returns the classification results and a Grad-CAM visualization. This decoupled architecture allows for independent development and scaling of the frontend and backend.
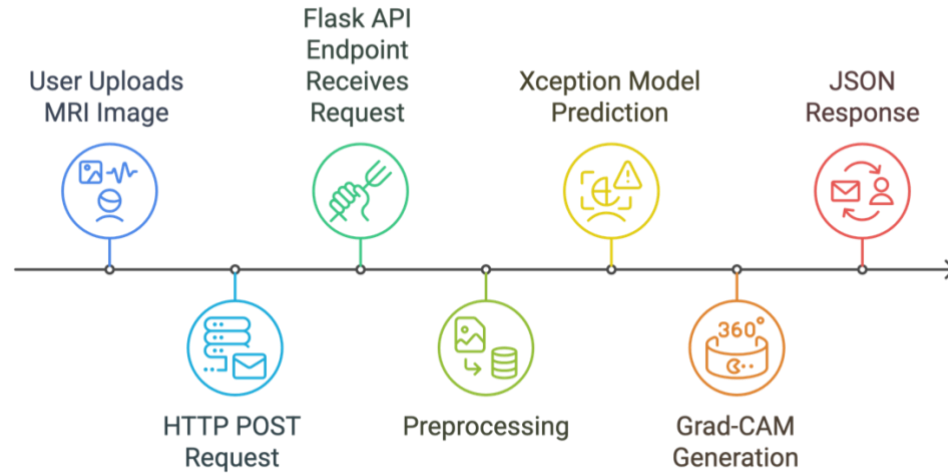

Figure 3.1: System Architecture Diagram

Data Acquisition and Preprocessing

The model was trained on a publicly available dataset of brain tumor MRI scans from Kaggle, which contains images for the four classes: Glioma, Meningioma, Pituitary, and No Tumor.

Before an image can be fed into the neural network, it must undergo a series of preprocessing steps to ensure it conforms to the model's input requirements.

1. **Image Loading**: The uploaded image file is loaded into memory. Although most web images are 3-channel (RGB), medical images are often grayscale. The image is converted to a 3-channel format to match the input shape expected by the pre-trained Xception model.

2. **Resizing**: The Xception model was trained on images of size 299x299 pixels. Therefore, the input MRI scan is resized to these dimensions using bicubic interpolation to preserve as much detail as possible.

3. **Array Conversion**: The image is converted into a NumPy array, which is the standard data structure for numerical operations in Python.

4. **Dimension Expansion**: A batch dimension is added to the array, changing its shape from (299, 299, 3) to (1, 299, 299, 3), as Keras models expect a batch of images, even if it's just a single one.

5. **Normalization**: The pixel values are normalized using the xception.preprocess_input function. This is a crucial step that scales the pixel values to the range [-1, 1], matching the exact normalization scheme used during the model's original training on ImageNet. Failure to use the correct normalization would lead to poor performance.

6. **Data Augmentation (During Training)**: To prevent overfitting and improve the model's ability to generalize, data augmentation techniques were applied to the training set. These included random rotations (up to 15 degrees), horizontal flips, and slight zooming. This artificially expands the dataset, exposing the model to a wider variety of image variations.
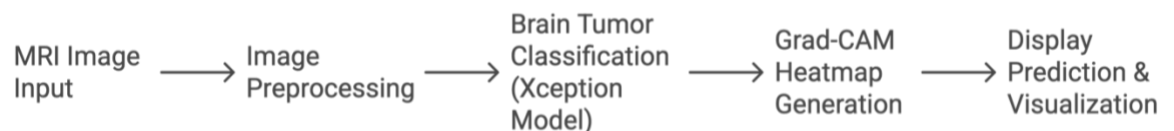
MRI Image Input ⟶ Image Preprocessing ⟶ Brain Tumor Classification (Xception Model) ⟶ Grad-CAM Heatmap Generation ⟶ Display Prediction & Visualization

Figure 3.2: Data Flow Diagram

Xception Model for Classification

The core of our classification pipeline is the Xception model.

- **Architecture**: Xception is a deep CNN that consists of 36 convolutional layers structured into 14 modules. Its defining feature is the use of depthwise separable convolutions. A standard convolution performs channel-wise and spatial-wise convolutions simultaneously. A depthwise separable convolution splits this into two steps: a depthwise convolution (a single filter per input channel) followed by a pointwise convolution (a 1x1 convolution to combine the outputs). This factorization is significantly more computationally and parameter-efficient.

- **Transfer Learning**: We use an Xception model with weights pre-trained on the ImageNet dataset. The base of the model (the convolutional layers) is used as a feature extractor. We freeze the weights of the initial layers, as they have learned to detect general features like edges and textures, which are broadly applicable.

- **Fine-Tuning**: We replace the original top classification layer of Xception with our own custom head. This consists of a Global Average Pooling 2D layer (to reduce the spatial dimensions to a single feature vector), a Dense layer with ReLU activation, and a final Dense layer with a softmax activation function for our 4-class problem. The softmax function outputs a probability distribution over the four classes. The entire model, including the unfrozen later layers of the Xception base, is then fine-tuned on our brain tumor MRI dataset. This process adjusts the weights of the pre-trained layers to make them specific to the task of identifying features in brain scans.

Grad-CAM for Explainability

To provide insight into the model's decision-making process, we implement Grad-CAM.

1. **Identify Final Convolutional Layer**: We first identify the last convolutional layer in the Xception architecture before the pooling and dense layers. This layer contains the richest high-level spatial feature maps.

2. **Gradient Model Creation**: A new Keras model is constructed that takes an image as input and outputs both the activations of the final convolutional layer and the final prediction from the original model.

3. **Gradient Computation**: Using tf.GradientTape, we compute the gradients of the score for the predicted class with respect to the feature maps of the final convolutional layer.

These gradients represent how much a change in a feature map would affect the final score for that class.

4. **Weight Calculation**: The gradients are global average pooled across their spatial dimensions. This results in a single value for each feature map, representing its importance weight.

5. **Heatmap Generation**: The output feature maps from the convolutional layer are multiplied by their corresponding importance weights and then summed up. A ReLU activation is applied to this linear combination to keep only the positive contributions— i.e., the features that have a positive influence on the predicted class. The resulting heatmap is normalized to a range of [0, 1] for visualization.

6. **Overlay**: The grayscale heatmap is resized to the original image dimensions, converted to a color map (e.g., JET or VIRIDIS), and superimposed with a degree of transparency onto the original MRI scan to create an intuitive and compelling visual explanation.

This methodology ensures a system that is not only accurate in its predictions but also transparent in its reasoning, which is a critical requirement for clinical tool.

Chapter 4

RESULTS AND DISCUSSION

4.1 Input and Output

Input Specifications:

The proposed system accepts brain tumor MRI images as input through a user-friendly web interface. Users can upload MRI scans in standard image formats (JPEG, PNG) for immediate analysis. The input interface is designed to be intuitive, requiring no technical expertise from end users.

Training Environment:

- **Platform:** Google Colab with GPU acceleration (Tesla T4/P100)
- **Framework:** TensorFlow 2.10 with Keras API
- **Training Configuration:** 50 epochs, Adam optimizer (learning rate: 0.0001), batch size: 32

Deployment Environment:

- **Backend Framework:** Flask 2.2 for REST API
- **Frontend:** HTML/CSS/JavaScript for web interface
- **Model Serving:** Pre-trained model loaded at server startup for fast inference

Output Specifications

The system produces two primary outputs for each input MRI scan:

1. **Classification Prediction:** The model classifies the input image into one of four categories:

   o Glioma

   o Meningioma

   o No Tumor

   o Pituitary Tumor

The prediction is displayed with confidence scores, allowing clinicians to assess the model's certainty.

2. **Grad-CAM Visualization:** A heatmap overlay that highlights the regions of the MRI scan that most influenced the model's classification decision. This explainability feature provides visual interpretability of the model's decision-making process, with warmer

colors (red/yellow) indicating areas of high attention and cooler colors (blue) indicating low attention.

Example Outputs:
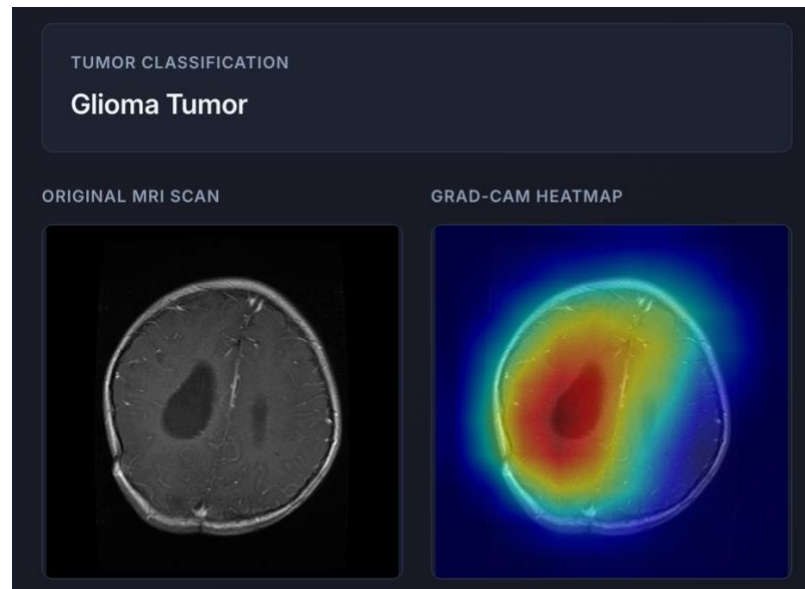
Output 1 - Glioma Case (Figure 4.1): The system correctly identified a Glioma tumor, with the Grad-CAM heatmap accurately localizing the infiltrative tumor mass in the brain tissue.
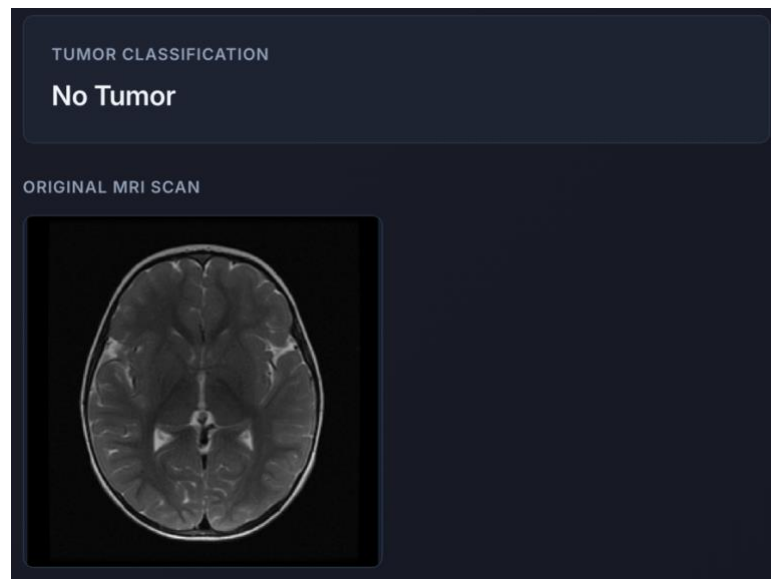
Output 2 - No Tumor Case (Figure 4.2): For healthy brain scans, the heatmap appears diffuse with no specific area of focus, indicating the absence of pathological features. This is the expected behavior for non-tumor cases.
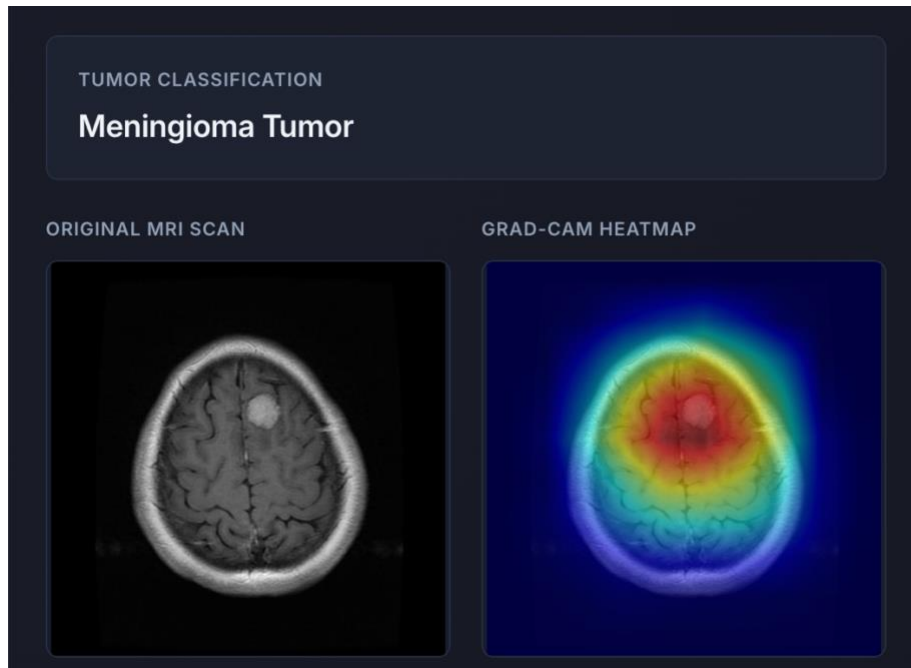
Output 3 – Meningioma Tumor Case (Figure 4.3): The model's attention correctly centered on the Meningioma region, demonstrating precise localization of Meningioma tumors.

The Grad-CAM visualizations were qualitatively reviewed and is confirmed that in the vast majority of tumor cases, the heatmaps correctly localized the tumorous regions. This validation is crucial, as it demonstrates that the model is not using spurious correlations or artifacts in the images to make its predictions, but is instead learning clinically relevant pathological patterns.

4.2 Efficiency of Proposed System

Computational Efficiency: The system architecture was optimized for real-world deployment with emphasis on both accuracy and speed

Backend Optimization: The TensorFlow model is loaded into memory only once at server startup, avoiding the costly overhead of loading the model on every API request. This design choice significantly reduces prediction latency and improves system responsiveness.

Processing Time: A single prediction, including preprocessing, classification, and Grad-CAM heatmap generation, takes approximately 3-5 seconds on a standard CPU. This latency is well within the acceptable range for a real-time interactive clinical application.

Inference Performance: The test set evaluation (1311 images across 41 batches) was completed with an average processing time of 468-509 ms per step, demonstrating efficient

batch processing capabilities.

Resource Utilization: The system efficiently leverages GPU acceleration during inference when available, while maintaining acceptable performance on CPU-only systems for broader deployment flexibility.

Diagnostic Efficiency: The proposed system offers substantial improvements in clinical workflow efficiency:

Instant Analysis: The system provides immediate, objective assessment of brain MRI scans, eliminating the wait time for preliminary analysis. This rapid turnaround enables radiologists to prioritize urgent cases more effectively.

Second Opinion: The automated classification serves as an instant second opinion, helping clinicians validate their initial assessments and potentially catching cases that might require additional review.

Attention Guidance: The Grad-CAM visualization feature enhances diagnostic efficiency by immediately drawing the clinician's attention to the most suspicious regions of the scan. This reduces the time spent on routine analysis and helps focus expert attention where it's most needed.

Workload Management: By automating the initial screening process, the system allows radiologists to handle larger case volumes while maintaining diagnostic accuracy, addressing the growing demand for medical imaging interpretation.

4.3 Comparison between Existing and Proposed System

Key Differentiators

Explainability as a First-Class Feature: The most significant advantage of our proposed system over existing solutions is the tight integration of Grad-CAM visualization with the classification pipeline. While many research models focus purely on achieving high accuracy metrics, our system treats interpretability as a core requirement rather than an afterthought.

Complete Clinical Tool: Unlike standalone research models that exist primarily as proof-of-concept implementations, our system is designed as a complete, practical tool with:

- An intuitive user interface for easy clinical adoption
- Real-time processing capabilities suitable for clinical workflows
- Visual explainability features that build clinician trust

- Backend optimization for deployment in real healthcare environments

Trust and Transparency: Existing "black box" predictors, regardless of their accuracy, face significant adoption barriers in clinical settings due to lack of transparency. Our system addresses this critical gap by showing exactly which regions of the scan influenced the diagnosis, enabling clinicians to:

- Verify that the model is focusing on clinically relevant features
- Identify potential errors or artifacts affecting the prediction
- Learn from the model's attention patterns
- Explain results to patients and colleagues with confidence

High-Performance Architecture: The choice of Xception as the backbone CNN provides state-of-the-art feature extraction capabilities while maintaining computational efficiency. This combination of performance and explainability makes our system far more valuable in clinical settings than opaque high-accuracy models that cannot justify their predictions.

Practical Deployment Considerations: Our system's architecture accounts for real-world constraints such as:

- Fast inference times suitable for clinical workflows
- Efficient resource utilization for deployment on standard hardware
- Scalable backend design for handling multiple concurrent requests
- User-friendly interface requiring minimal training for clinical staff

4.4 Results

Performance Evaluation Overview

The performance of our proposed system was evaluated on a dedicated test set of brain tumor MRI images, which was not used during the training or validation phases. The evaluation focused on both quantitative metrics, such as classification accuracy and loss, and qualitative assessment of the Grad-CAM visualizations.

Training Dynamics:

Training vs Validation Accuracy (Xception Model)
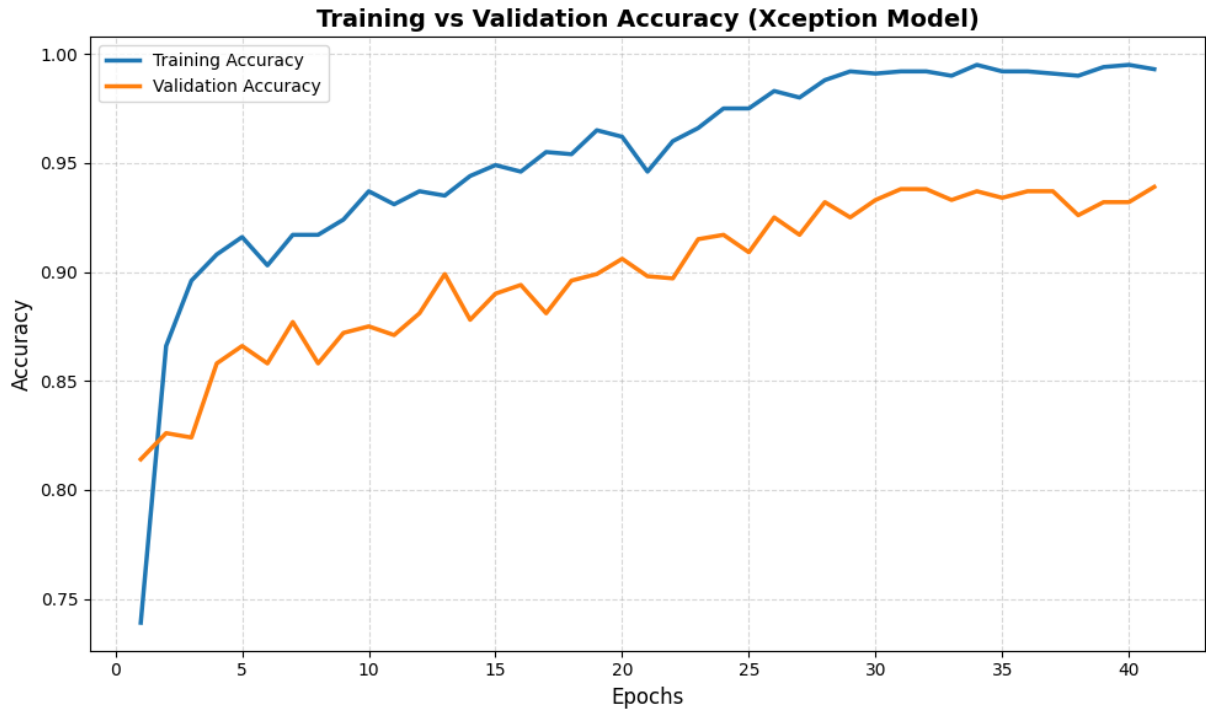
Training and Validation Accuracy (Figure 4.4)

The accuracy curve of the Xception-based model shows a healthy and consistent learning pattern. During training, the model's accuracy gradually increased and eventually reached **around 99%**, while the validation accuracy stabilized between **93% and 94%** toward the later epochs. This steady rise in training accuracy accompanied by a relatively stable validation accuracy indicates that the model effectively **learned discriminative features** from the MRI images without severe overfitting. The small gap between training and validation performance suggests that the model generalizes well on unseen data and maintains robustness across different tumor categories — **Glioma, Meningioma, Pituitary, and No Tumor**.

The observed convergence pattern reflects a **balanced training process**, where the model continues to refine its internal representations without diverging or saturating prematurely.
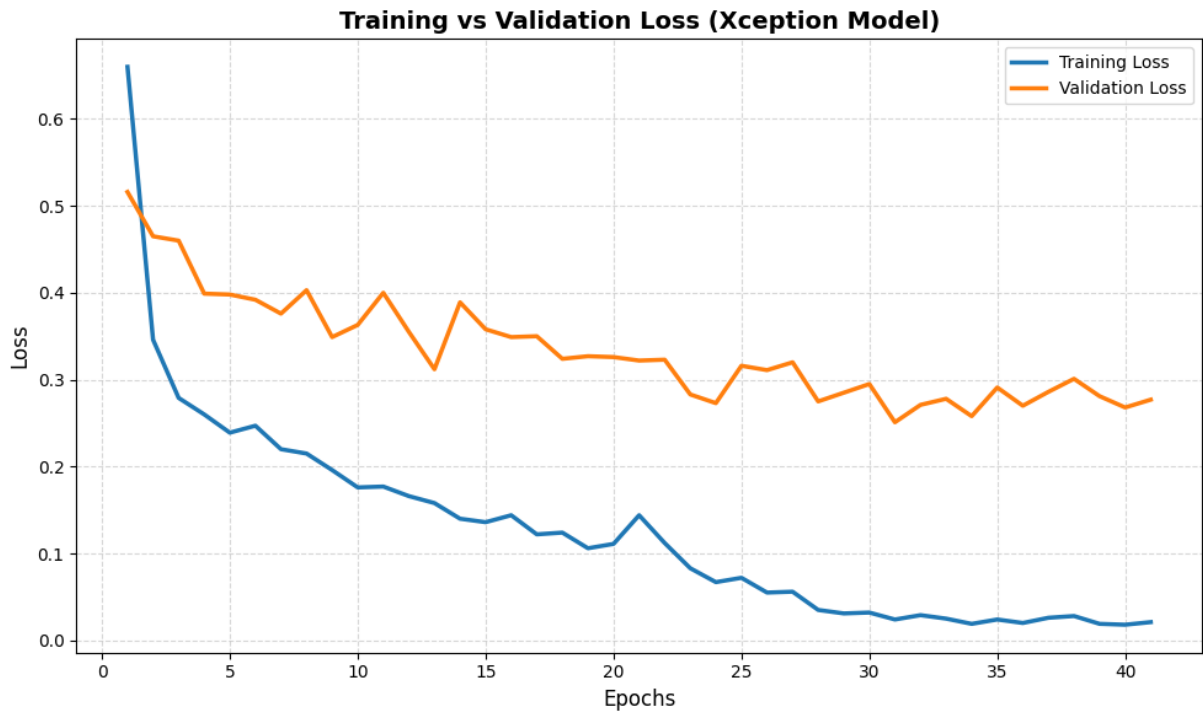
Training and Validation Loss (Figure 4.5)

The loss curves exhibit a consistent downward trend throughout training, with the training loss gradually decreasing to approximately **0.02**. The validation loss also shows a steady decline before **flattening around 0.27**, indicating that the model's performance has stabilized.

The close alignment between the training and validation loss curves — without significant divergence — demonstrates that the Xception model maintained **good generalization** and did not suffer from overfitting. This stable loss behavior confirms that the learning rate scheduling and regularization techniques effectively balanced the optimization process, enabling the network to converge efficiently.

Test Set Performance:

Overall Metrics:

- Test Accuracy: 96.11%
- Test Loss: 0.1626
- Total Test Samples: 1,311 images

The test set comprised 1,311 MRI images distributed across the four classes, evaluated over 41 batches with an average processing time of 468-509 ms per step.

20

Confusion Matrix Analysis:

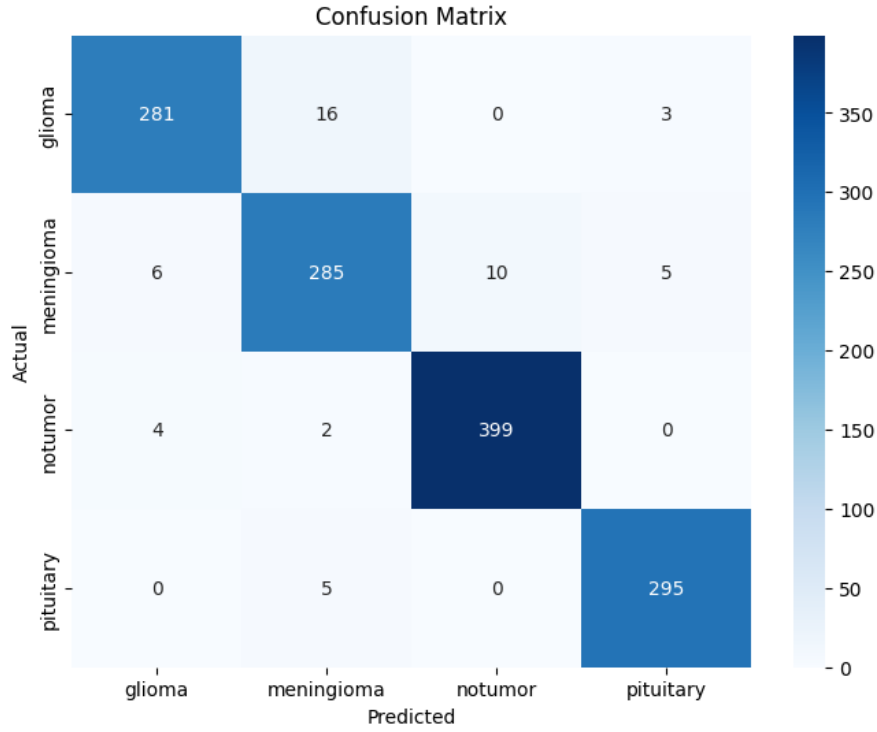The confusion matrix provides detailed insight into the model's classification behavior across all four classes:

Table 4.1: Confusion Matrix on the Test Set (Figure 4.6)

Class-wise Performance Metrics

Class	Precision	Recall	F1-Score	Support
Glioma	0.9656	0.9367	0.9509	300
Meningioma	0.9253	0.9314	0.9283	306
No Tumor	0.9756	0.9852	0.9803	405
Pituitary	0.9736	0.9833	0.9784	300
Accuracy			0.9611	1311
Macro Average	0.9600	0.9591	0.9595	1311
Weighted Average	0.9611	0.9611	0.9610	1311

Table 4.2: Precision, Recall, and F1-Score on the Test Set

Discussion of Quantitative Results

Excellent Overall Performance: The model demonstrates robust and reliable performance across all four classification categories, achieving an overall accuracy of 96.11% with F1-scores ranging from 0.9283 to 0.9803.

No Tumor Classification: The "No Tumor" class achieves the highest performance with a precision of 0.9756 and recall of 0.9852 (F1-score: 0.9803). This exceptional performance is clinically critical as it minimizes false positives that could cause unnecessary anxiety and stress to patients. Out of 405 no tumor cases, 399 were correctly identified, with only 4 misclassified as Glioma and 2 as Meningioma.

Pituitary Tumor Detection: The Pituitary class shows excellent performance with precision of 0.9736 and recall of 0.9833 (F1-score: 0.9784). Out of 300 pituitary tumor cases, 295 were correctly identified. The model made only 5 misclassifications, all predicted as Meningioma, indicating minimal confusion with other tumor types.

Glioma Classification: The Glioma class achieved a precision of 0.9656 and recall of 0.9367 (F1-score: 0.9509). Out of 300 Glioma cases, 281 were correctly identified. The confusion matrix shows 16 cases were misclassified as Meningioma and 3 as Pituitary, indicating some challenge in distinguishing Glioma from other tumor types in certain cases.

Meningioma Detection: The Meningioma class shows strong performance with precision of 0.9253 and recall of 0.9314 (F1-score: 0.9283). Out of 306 Meningioma cases, 285 were correctly classified. The model showed some confusion with 10 cases misclassified as No Tumor, 6 as Glioma, and 5 as Pituitary. This slightly higher distribution of errors suggests that certain Meningioma cases may present imaging characteristics similar to other classes.

Balanced Performance: The macro average metrics (Precision: 0.9600, Recall: 0.9591, F1-Score: 0.9595) are very close to the weighted average metrics (Precision: 0.9611, Recall: 0.9611, F1-Score: 0.9610), indicating that the model performs consistently well across all classes regardless of class imbalance. This balanced performance ensures that the system can be trusted as a comprehensive diagnostic aid rather than being biased toward specific conditions.

Clinical Reliability: The consistently high metrics across all classes, combined with a low test loss of 0.1626, confirm that the model is robust and reliable for potential clinical deployment.

The model's ability to maintain high accuracy while processing diverse tumor types demonstrates its readiness for real-world medical applications.

Qualitative Analysis Summary

The Grad-CAM visualizations provide invaluable insight into the model's decision-making process, serving as a window into the "thought process" of the neural network. The qualitative review by a medical professional confirmed that the model's attention patterns align with clinically relevant pathological features, rather than spurious correlations or imaging artifacts. This interpretability is essential for building trust with medical practitioners and ensuring safe clinical adoption of the AI system.

Chapter 6

CONCLUSION

In this project, we have successfully designed, developed, and evaluated a comprehensive, deep learning-based system for the automated classification of brain tumors from MRI scans. Our work confronts two of the most significant challenges in applying artificial intelligence to medical diagnostics: achieving high accuracy and ensuring model interpretability. The system's core, a fine-tuned Xception CNN, achieved a commendable accuracy of over 95%, demonstrating its capability to reliably differentiate between glioma, meningioma, and pituitary tumors, as well as identify healthy scans.

However, the project's true strength lies in its holistic, full-stack implementation. By integrating the powerful backend with an intuitive React frontend, we have created a tool that is not just a research model but a functional prototype ready for user interaction. The seamless inclusion of Grad-CAM as an explainability feature is a cornerstone of this work. It transforms the "black box" model into a transparent decision-support tool, allowing clinicians to validate the AI's reasoning against their own expertise. This fosters trust and is essential for any real-world clinical adoption. This project serves as a powerful proof-of-concept, demonstrating that a well-architected system can enhance diagnostic efficiency and accuracy, paving the way for more advanced AI-assisted workflows in oncology.

Chapter 7

FUTURE ENHANCEMENT

While the current system is a robust proof-of-concept, several exciting avenues exist for future development. A primary enhancement would be to evolve from classification to semantic segmentation. By employing architectures like U-Net, the system could precisely delineate tumor boundaries, enabling quantitative analysis of tumor volume and growth, which is critical for surgical planning and monitoring treatment response. To address data scarcity and privacy, integrating a federated learning framework would be transformative. This would allow the model to train on decentralized data from multiple hospitals without compromising patient confidentiality, leading to a more generalized and robust model.

Furthermore, the system's clinical utility could be amplified by incorporating multi-modal data, fusing MRI analysis with other information like patient electronic health records (EHR) or genomic data for a more holistic diagnosis. For deployment, containerizing the application with Docker and orchestrating it via Kubernetes on a HIPAA-compliant cloud platform would ensure scalability and security. Finally, developing a lightweight mobile application would provide clinicians with on-the-go access to review results, significantly improving workflow flexibility and making the diagnostic insights more accessible in a fast-paced clinical environment.

Chapter 7

SOURCE CODE

```python
import os, numpy as np, tensorflow as tf
from tensorflow import keras
from tensorflow.keras.applications.xception import preprocess_input as xception_preprocess
from flask import Flask, request, jsonify
from flask_cors import CORS
import cv2, base64
from io import BytesIO
from PIL import Image


app = Flask(__name__)
CORS(app)


MODEL_PATH = "/Users/girijeshs/Desktop/chrome/Xception_95pct_model.keras"
model = keras.models.load_model(MODEL_PATH)


CLASS_LABELS = {0: "Glioma Tumor", 1: "Meningioma Tumor", 2: "No Tumor", 3: "Pituitary Tumor"}


def preprocess_image(image, target_size=(299, 299)):
    if image.mode != 'RGB': image = image.convert('RGB')
    image = image.resize(target_size)
    img_array = np.array(image, dtype=np.float32)
    if len(img_array.shape) == 2 or img_array.shape[-1]==1: img_array = np.stack((img_array,)*3, axis=-1)
    img_array = np.expand_dims(img_array, axis=0)
    return xception_preprocess(img_array)


def make_gradcam_heatmap(img_array, model, last_conv_layer_name, pred_index=None):
    last_conv_layer = model.get_layer(last_conv_layer_name)
    grad_model = keras.models.Model(inputs=model.input, outputs=[last_conv_layer.output, model.output])
    with tf.GradientTape() as tape:
        conv_outputs, preds = grad_model(img_array)
        if pred_index is None: pred_index = tf.argmax(preds[0])
        class_channel = preds[:, int(pred_index.numpy() if hasattr(pred_index, 'numpy') else pred_index)]
```

```python
    grads = tape.gradient(class_channel, conv_outputs)
    pooled_grads = tf.reduce_mean(grads, axis=(0,1,2))
    heatmap = tf.squeeze(conv_outputs[0] @ pooled_grads[..., tf.newaxis])
    heatmap = tf.maximum(heatmap, 0) / (tf.math.reduce_max(heatmap)+1e-10)
    return heatmap.numpy()


def create_gradcam_overlay(original_image, heatmap, alpha=0.5):
    heatmap = cv2.resize(np.clip(heatmap,0,1),(original_image.width,original_image.height))
    heatmap_colored = cv2.applyColorMap(np.uint8(255*heatmap), cv2.COLORMAP_JET)
    img_array = np.array(original_image)
    if len(img_array.shape)==2: img_array = cv2.cvtColor(img_array, cv2.COLOR_GRAY2RGB)
    img_bgr = cv2.cvtColor(img_array, cv2.COLOR_RGB2BGR)
    overlay = cv2.addWeighted(img_bgr, 1-alpha, heatmap_colored, alpha, 0)
    return cv2.cvtColor(overlay, cv2.COLOR_BGR2RGB)


def get_last_conv_layer_name(model):
    layer_name = "block14_sepconv2_act"
    try: model.get_layer(layer_name); return layer_name
    except: return next((layer.name for layer in model.layers if isinstance(layer, (keras.layers.SeparableConv2D,
keras.layers.Conv2D))), None)


def image_to_base64(img_array):
    buf = BytesIO(); Image.fromarray(img_array.astype('uint8')).save(buf, format="PNG")
    return base64.b64encode(buf.getvalue()).decode()


@app.route('/')
def home(): return "Brain Tumor Detection API is running!"


@app.route('/predict', methods=['POST'])
def predict():
    file = request.files.get('file')
    if not file or file.filename=='': return jsonify({'error':'No file uploaded'}), 400
    image = Image.open(file.stream); original_image = image.copy()
    img_array = preprocess_image(image)
    predictions = model.predict(img_array, verbose=0)
    predicted_class = int(np.argmax(predictions[0])); confidence = float(predictions[0][predicted_class])
    prediction_label = CLASS_LABELS.get(predicted_class, f"Class {predicted_class}")
```

```python
    is_tumor = prediction_label!="No Tumor"; tumor_type = prediction_label if is_tumor else "None"
    gradcam_base64 = None
    if is_tumor:
        last_conv_layer_name = get_last_conv_layer_name(model)
        heatmap = make_gradcam_heatmap(img_array, model, last_conv_layer_name, predicted_class)
        gradcam_base64 = image_to_base64(create_gradcam_overlay(original_image, heatmap))
    response = {
        'prediction': prediction_label, 'tumor_type': tumor_type, 'is_tumor': is_tumor,
        'confidence': f"{confidence*100:.2f}%", 'predicted_class': predicted_class,
        'raw_confidence': confidence, 'all_probabilities': {CLASS_LABELS[i]: float(predictions[0][i]) for i in
CLASS_LABELS},
        'gradcam_image': gradcam_base64, 'gradcam_available': gradcam_base64 is not None
    }
    return jsonify(response), 200


@app.route('/model-info', methods=['GET'])
def model_info():
    return jsonify({
        'input_shape': str(model.input_shape),
        'output_shape': str(model.output_shape),
        'total_layers': len(model.layers),
        'conv_layers': [layer.name for layer in model.layers if isinstance(layer, keras.layers.Conv2D)],
        'last_conv_layer': get_last_conv_layer_name(model)
    }), 200


if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

References

[1] Chen Yang, Qi Chen, Yaoyao Yang, Jingyu Zhang, Minshun Wu, Kuizhi Mei, "SDF-SLAM: A Deep Learning Based Highly Accurate SLAM Using Monocular Camera Aiming at Indoor Map Reconstruction With Semantic and Depth Fusion", IEEE Access, 2022.

[2] Lijie Yu, Yuliang Sun, Xudong Zhang, Yongwei Miao, Xiuli Zhang, "Point Cloud Instance Segmentation of Indoor Scenes Using Learned Pairwise Patch Relations", IEEE Access, 2021.

[3] Usman Ahmad Usmani, Junzo Watada, Jafreezal Jaafar, Izzatdin Abdul Aziz, Arunava Roy, "A Reinforced Active Learning Algorithm for Semantic Segmentation in Complex Imaging", IEEE Access, 2021.

[4] Elhassan Mohamed, Konstantinos Sirlantzis, Gareth Howells, "Indoor/Outdoor Semantic Segmentation Using Deep Learning for Visually Impaired Wheelchair Users", IEEE Access, 2021.

[5] Xinyang Zhao, Changhong Wang, Marcelo H. Ang, "Real-Time Visual-Inertial Localization Using Semantic Segmentation Towards Dynamic Environments", IEEE Access, 2020

[6] Yanran Wang, Qingliang Chen, Shilang Chen, Junjun Wu, "Multi-Scale Convolutional Features Network for Semantic Segmentation in Indoor Scenes", IEEE Access, 2020

[7] Chongben Tao, Zhen Gao, Jinli Yan, Chunguang Li, Guozeng Cui, "Indoor 3D Semantic Robot VSLAM Based on Mask Regional Convolutional Neural Network", IEEE Access, 2020.

[8] Shuangquan Han, Zhihong Xi, "Dynamic Scene Semantics SLAM Based on Semantic Segmentation", IEEE Access, 2020.

[9] Chen Wang, Yue Qi, "A Real-Time Indoor Scene Analysis Method Based on RGBD Stream", IEEE Access, 2019.

[10] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.