GROUP ID: GB13

A PROJECT REPORT ON

BRAIN TUMOUR CLASSIFICATION WITH QUANTUM-AUGMENTED DEEP LEARNING MODEL

SUBMITTED TO THE PIMPRI CHINCHWAD COLLEGE OF ENGINEERING, AN AUTONOMOUS INSTITUTE IN PUNE
IN THE FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

B. TECH. (COMPUTER ENGINEERING)

SUBMITTED BY

PRATHAMESH SAHEBRAO GOLE	PRN:121B1B075
ADITYA SHRIKANT GORANE	PRN:121B1B076
SHRIKANT ANANTA JADHAO	PRN:121B1B084
VISHWAJEET SANJAY KOSHTI	PRN:121B1B124

UNDER THE GUIDANCE OF
GUIDE NAME: DR. ASMITA MANNA

DEPARTMENT OF COMPUTER ENGINEERING

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi, Pimpri-Chinchwad, PUNE 41104

2024-2025

CERTIFICATE

This is to certify that the project report entitles

"BRAIN TUMOUR CLASSIFICATION WITH QUANTUM-AUGMENTED DEEP LEARNING MODEL"

Submitted by

PRATHAMESH SAHEBRAO GOLE	**PRN:121B1B075**
ADITYA SHRIKANT GORANE	**PRN:121B1B076**
SHRIKANT ANANTA JADHAO	**PRN:121B1B084**
VISHWAJEET SANJAY KOSHTI	**PRN:121B1B124**

Are bonafide students of this institute and the work has been carried out by them under the supervision of Prof. **Dr. Asmita Manna** and it is approved for the partial fulfillment of the requirement of Pimpri Chinchwad College of Engineering an autonomous institute, for the award of the B. Tech. degree in Computer Engineering.

(Prof. Dr. Asmita Manna)
Guide
Department of Computer Engineering

(Prof. Dr. Sonali Patil)
Head,
Department of Computer Engineering

(Prof. Dr. G.N. Kulkarni)
Director,
Pimpri Chinchwad College of Engineering Pune – 44

Place: Pune
Date:

ACKNOWLEDGEMENT

Brain Tumour Classification with Quantum-Augmented Deep Learning Model

We express our sincere thanks to our **Guide Prof. Dr. Asmita Manna** for her constant encouragement and support throughout our project, especially for the useful suggestions given during the course of project and having laid down the foundation for the success of this work.

We would also like to thank our **Research & Innovation coordinator, Prof. Dr. Reena Kharat, and Project Coordinator, Prof. Sushma R. Vispute, for her assistance, genuine support, and guidance from the** early stages of the project. We would like to thank **Prof. Dr. Sonali Patil, Head of Computer Department,** for her unwavering support during the entire course of this project work. We are very grateful to our **Director, Prof. Dr. G.N. Kulkarni, for providing us with an environment in which** to complete our project successfully. We also thank all the staff members of our college and technicians for their help in making this project a success.

We also thank all the web committees for enriching us with their immense knowledge. Finally, we take this opportunity to extend our deep appreciation to our family and friends for all that they meant to us during the crucial times of the completion of our project.

NAME OF THE STUDENTS	SIGN
PRATHAMESH SAHEBRAO GOLE	
ADITYA SHRIKANT GORANE	
SHRIKANT ANANTA JADHAO	
VISHWAJEET SANJAY KOSHTI	

ABSTRACT

Brain Tumour Classification with Quantum-Augmented Deep Learning Model

Brain tumours pose a significant challenge in medical diagnosis, demanding precise and early detection for effective treatment. Traditional deep-learning models have demonstrated success in medical imaging, but their limitations in capturing complex patterns necessitate advancements. Our project introduces a hybrid quantum-classical approach to brain tumour classification, leveraging Quantum Transfer Learning (QTL) to enhance predictive accuracy. By integrating quantum circuit layers with classical convolutional neural networks (CNNs), we aim to achieve superior feature extraction and classification performance.

This system employs well-established architectures like ResNet18, VGG16, MobileNetV2, and InceptionV3, augmented with quantum layers utilizing Pennylane. The Figshare brain MRI dataset is the foundation for training, ensuring model robustness. The project also features an intuitive web interface where users can upload MRI images and receive predictions with confidence scores. Flask is the backend, handling model inference, while the frontend ensures a seamless user experience. The implementation has been optimized for efficient processing, with MobileNetV2 emerging as the best-performing model, achieving a validation accuracy of 95.79%.

This innovative approach bridges the gap between classical and quantum computing in medical diagnostics, demonstrating the potential of quantum-enhanced deep learning in real-world applications. The project contributes to the evolving field of quantum machine learning (QML) and provides a scalable solution for early brain tumour detection. With its strong foundation in research, practical implementation, and promising results, this system paves the way for future advancements in AI-driven healthcare.

KEYWORDS: Brain tumour classification, Quantum transfer learning, Hybrid quantum-classical model, Deep learning, Convolutional neural networks, medical imaging, MRI, Quantum machine learning, pennylane, pytorch, AI in healthcare.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ABBREVIATION	ILLUSTRATION
Abbreviation	**Full Form**

MRI	Magnetic Resonance Imaging
CNN	Convolutional Neural Network
QTL	Quantum Transfer Learning
DL	Deep Learning
AI	Artificial Intelligence
ML	Machine Learning
TPU	Tensor Processing Unit
GPU	Graphics Processing Unit
API	Application Programming Interface
JSON	JavaScript Object Notation
SRS	Software Requirements Specification
SDLC	Software Development Life Cycle
UI	User Interface
UX	User Experience
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
ReLU	Rectified Linear Unit
SMOTE	Synthetic Minority Oversampling Technique
ADASYN	Adaptive Synthetic Sampling
YOLO	You Only Look Once (Object Detection Model)
VGG16	Visual Geometry Group 16-layer CNN
ResNet	Residual Neural Network

LIST OF FIGURES

LIST OF TABLES

CHAPTER 01
INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

Brain tumours are among the most critical and life-threatening conditions, requiring early detection for effective treatment. Magnetic Resonance Imaging (MRI) is the standard imaging technique for diagnosing brain tumours, but manual assessment is time-consuming and susceptible to human errors. Deep learning models have significantly advanced the field of medical image analysis, providing automated and precise classification of brain tumours. However, traditional deep learning methods rely on extensive labeled data and high computational resources, limiting their efficiency. To address these challenges, this project integrates **Quantum Transfer Learning (QTL)** with classical Convolutional Neural Networks (CNNs), leveraging quantum computing's potential to enhance feature extraction and classification accuracy. The developed system provides a **hybrid quantum-classical deep learning model** that improves classification performance while reducing computational complexity, making brain tumour diagnosis more efficient and accessible.

1.2 MOTIVATION:

The motivation for this project arises from the critical need for fast and precise brain tumour detection. Current deep learning models provide impressive results, but computational constraints and the size of available datasets often limit their performance. Quantum computing offers a promising solution by introducing quantum-enhanced feature extraction, which can improve model accuracy. The fusion of classical and quantum techniques ensures efficient medical image processing while leveraging both paradigms' strengths. With the backing of Unicare Hospital, this project aims to contribute to real-world clinical applications and advance AI-driven healthcare.

1.3 PROBLEM STATEMENT AND OBJECTIVES:

Develop an efficient brain tumour classification system that integrates quantum computing with deep learning to enhance model performance and provide accurate, real-time predictions for MRI-based tumour detection.

The objectives include:

- Train and test traditional models (ResNet, VGG16, MobileNet, Inception) to benchmark performance.
- Integrate a quantum circuit layer within CNN backbones to enhance feature representation.
- Develop a hybrid classical-quantum deep learning model using PyTorch and PennyLane.
- Evaluate the effectiveness of Quantum Transfer Learning (QTL) for brain tumour classification.
- Optimize the model for computational efficiency and real-time predictions.
- Design and implement a web-based UI for users to upload MRI scans and receive classification results.
- Ensure the system is deployable in a clinical setting.

1.4 SCOPE OF THE WORK:

The scope of this work encompasses:

- Dataset Utilization: The Figshare MRI dataset is used for training and validation.
- Model Development: Integrates Quantum Circuit Layers within classical CNN architectures.
- Implementation Frameworks: Utilizes TensorFlow, PyTorch, and Pennylane for model development.
- Performance Evaluation: Compares traditional CNN models with the hybrid quantum-classical model.
- Web Interface Development: Builds a Flask-based UI to enable MRI image uploads and classification.
- Deployment Considerations: Ensures the system is adaptable for real-world medical applications with potential cloud integration.
- Investigation of potential enhancements and optimizations to further improve the efficiency and effectiveness of the recognition system.

1.5 METHODOLOGIES OF PROBLEM-SOLVING

1. **Data Collection and Preprocessing:** The Figshare dataset containing MRI scans of brain tumours is collected and preprocessed for training. This involves resizing images, normalizing pixel values, and augmenting data to enhance model generalization. The dataset is split into training, validation, and test sets to evaluate the model effectively. Data augmentation techniques such as rotation, flipping, and contrast adjustments are applied to improve robustness. Additionally, through sponsorship and collaboration with Unicare Hospital and Tesla Radiologist, we collected real-world clinical MRI scans from 22 patients. These scans—covering multiple orientations—contributed over 1200+ images, enriching our dataset's diversity and authenticity.

2. **Classical Model Selection and Training:** Four CNN architectures—ResNet18, VGG16, MobileNetV2, and InceptionV3—are selected as the classical backbones. Each model is pre-trained on ImageNet and fine-tuned on the brain tumour dataset. The final convolutional layers are replaced with custom, fully connected layers to match classification requirements. The models are trained using cross-entropy loss and optimized with Adam for stable convergence.

3. **Quantum Layer Design and Integration:** Quantum circuits use four qubits with two quantum layers consisting of RY and RZ rotations and CNOT entangling gates. The classical feature embeddings from CNN models are reduced via linear layers and passed to the quantum circuit. The quantum layer processes the information and returns modified features, which are then classified using a final dense layer.

4. **Model Training and Optimization:** The hybrid quantum-classical model is trained by combining quantum feature extraction with classical CNN processing. The training process involves multiple epochs, and performance is monitored using accuracy and loss curves. Techniques such as learning rate scheduling and dropout are used to optimize training stability. Quantum gradients are computed using the parameter-shift rule to ensure smooth integration with the classical model.

5. **Performance Evaluation and Benchmarking:** The trained models are evaluated based on accuracy, precision, recall, and F1 score. Comparisons between pure classical and hybrid

quantum-classical models are performed to assess improvements. Metrics such as inference time, memory consumption, and computational cost are analyzed to validate the effectiveness of the hybrid approach.

6. **Web UI Development and Deployment:** A Flask-based web interface is developed to allow users to upload MRI scans and receive classification results. The UI is built using HTML, CSS, and JavaScript to provide a seamless experience. The trained model is deployed on a server, and predictions are processed in real-time. Flask-CORS is configured to handle cross-origin requests, ensuring smooth frontend-backend interaction.

7. **Future Scope and Sustainability Considerations:** The project explores potential improvements, such as increasing qubit capacity for deeper quantum layers and optimizing hybrid architectures for real-time medical applications. Regarding energy consumption and scalability, the sustainability of quantum-enhanced deep learning models is analyzed. Further research is proposed to integrate federated learning for privacy-preserving brain tumour classification.

CHAPTER 02

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 REVIEW OF RECENT LITERATURE

Imam, R., & Alam, M. T [1]. Investigate the impact of different loss functions, including focal loss and oversampling methods, such as SMOTE and ADASYN, in addressing the data imbalance issue. The proposed method, which integrates VGG-16 with CNN, achieves an impressive classification accuracy of 96%, outperforming conventional approaches. The research focuses on three primary types of brain tumours: glioma, meningioma, and pituitary tumour, aiming to develop an automated and precise classification mechanism. Preprocessing includes image normalization and resizing to a standard input size of 128x128 pixels. The dataset undergoes a training-validation split in a 90:10 ratio to ensure robust model performance.

Karim, P. J., Mahmood, S. R., & Sah, M. [2] explored brain tumour classification using a deep transfer learning approach with a fine-tuning strategy and a Support Vector Machine (SVM) classifier. The methodology involves preprocessing MRI images, data augmentation with resampling, and feature extraction using a pre-trained Convolutional Neural Network (CNN) and ResNet-50 model. The extracted features are flattened and passed to an SVM for classification. Additionally, a novel fine-tuning strategy incorporating dense layers with dropout and Rectified Linear Units (ReLU) is implemented to enhance classification performance. The study is conducted on the Figshare dataset, comprising three tumour types: meningioma, glioma, and pituitary tumours. The proposed model achieves an impressive classification accuracy of 99.35% using a CNN with fine-tuning and SVM, while ResNet-50 with fine-tuning reaches 99.61%. The results demonstrate significant improvements over existing state-of-the-art methods, reinforcing the efficacy of transfer learning for automated brain tumour detection.

Shelatkar, T., & Bansal, U. [3] enhance detection accuracy on the Brats dataset by freezing select layers of the YOLO model and appending customized layers. YOLOv5n, a lightweight yet effective variant, is utilized for real-time detection, optimizing storage efficiency while ensuring high accuracy. The research emphasizes the necessity of transfer learning to adapt YOLO’s capabilities for brain tumour detection.

Paul, S., Ahad, D. M. T., & Hasan, M. M. [4] focused on brain cancer segmentation using the YOLOv5 deep neural network. The study utilized the Brats 2021 dataset from the RSNA-MICCAI brain tumour radio genomic classification competition. The YOLOv5 algorithm, known for its efficiency and accuracy in object detection, was employed to identify and delineate brain tumours in MRI images. The model leverages techniques such as Residual blocks, Bounding box regression, and Intersection Over Union (IOU). The authors mentioned limitations related to the computational resources available, as they primarily used free resources like Google Colab, which restricted extensive hyperparameter tuning and the exploration of various optimizers. Additionally, the study relied on publicly available secondary data rather than primary data collected directly from clinical settings. The evaluation of the model's performance included assessing the runtime efficiency on an M2 10-core GPU. The study reported a mean Average Precision (mAP@0.5) score of 95.07% on one dataset and a precision of 88%, a recall of 88.58%, and an F1-Score of 89.45% on another. These metrics highlight the capability of the YOLOv5 model for effectively detecting and localizing brain tumours in MRI scans.

Guan, Y., Aamir, M., Rahman, Z., Ali, A., Abro, W. A., Dayo, Z. A., ... & Hu, Z. [5] proposed a framework for efficient brain tumour classification using MRI images, evaluated on a publicly available brain tumour dataset. Their framework involved several key stages: pre-processing of MRI images to enhance visual quality and increase the number of samples, generating tumour proposals using an agglomerative clustering-based method, extracting features through a backbone architecture, refining these proposals using a dedicated refinement network, aligning the refined proposals to a consistent size, and finally, performing the classification task using a head network. The study did not explicitly mention the limitations of the proposed framework within the provided snippets. The experimental results demonstrated that the proposed method achieved an overall classification accuracy of 98.04% on the brain tumour dataset.

Talukder, M. A., Islam, M. M., Uddin, M. A., Akhter, A., Pramanik, M. A. J., Aryal, S., ... & Moni, M. A. [6] present a novel deep learning (DL) method for classifying brain tumours that combine preprocessing, transfer learning (TL) architecture reconstruction, and fine-tuning. Four TL algorithms were utilized in our methodology: Xception, ResNet50V4, InceptionResNetV4, and DenseNet201. We evaluated the model's performance using various metrics, including accuracy, recall, precision, f1 score, MAE, MSE, and RMSE, to demonstrate the substantial progress made.

Abd El Kader, I., Xu, G., Shuai, Z., Saminu, S., Javaid, I., & Salim Ahmad, I. [7] present a Differential Deep Convolutional Neural Network (Differential Deep-CNN) designed for classifying brain tumours with high accuracy. The approach incorporates differential operators into

CNN architecture to extract enhanced feature maps, improving classification performance. The study addresses challenges such as the intricate structure of the brain and the complex nature of MRI images, which often hinder accurate classification. The proposed model achieves an impressive accuracy of 99.25% on a dataset containing 25,000 MRI images of normal and abnormal brain scans. The research highlights the effectiveness of pixel-based directional pattern analysis and contrast calculations in differentiating tumour types. The findings suggest that Differential Deep-CNN can significantly enhance the reliability of automatic tumour classification, reducing the need for manual intervention and aiding radiologists in early diagnosis.

Veeramuthu, A., Meenakshi, S., Mathivanan, G., Kotecha, K., Saini, J. R., Vijayakumar, V., & Subramaniyaswamy, V. [8] employed feed-forward artificial neural networks, backpropagation neural networks, SVM, and deep learning algorithms for tumour classification. The combined feature and image-based classifier optimizes the classification process by integrating sources of information, resulting in a more effective and accurate brain tumour classification system.

Aamir, M., Rahman, Z., Dayo, Z. A., Abro, W. A., Uddin, M. I., Khan, I., ... & Hu, Z. [9] presented a deep learning approach for brain tumour classification using MRI images. Their proposed method involved several stages, including enhancement of low-quality MRI images, feature extraction using pre-trained EfficientNet and ResNet50 models, fusion of the extracted features using partial least squares (PLS), generation and refinement of tumour location proposals using agglomerative clustering and structural edge detection, alignment of these proposals to a fixed size using ROI pooling, and finally, classification using a combination of fully connected layers and deconvolutional layers with a SoftMax activation function. The authors noted that the classification accuracy of their model was proportional to the number of training images, suggesting that smaller datasets might limit performance. They also mentioned that an extensive dataset could be computationally expensive. The proposed method achieved a high classification accuracy of 98.95% on a publicly available brain tumour dataset using 5-fold cross-validation. Specifically, the accuracy for Meningioma was 98.30%, for Glioma 98.72%, and for Pituitary 99.37%. The model also demonstrated high sensitivity, specificity, precision, and f1-score for each tumour class. This study showcases a comprehensive approach to brain tumour classification leveraging advanced deep learning techniques.

Ayadi, W., Elhamzi, W., Charfi, I., & Atri, M. [10] presented an innovative model for multi-brain tumour classification based on CNN. It is an automatic system which requires a minimum of preprocessing. Extensive performance evaluations highlight the proposed model's superior accuracy compared to existing approaches. Despite a relatively small training dataset, the model demonstrates robustness, suggesting its applicability to broader MRI classification tasks.

Badža, M. M., & Barjaktarović, M. Č. [11] provide an extensive overview of brain tumour classification, emphasizing the significance of accurate diagnosis in clinical settings. It discusses the limitations of traditional classification methods and highlights the promise of convolutional neural networks (CNNs) in enhancing accuracy. It might mention the limitations of existing processes and how convolutional neural networks (CNNs) promise to improve accuracy. Here, the authors would discuss previous research on brain tumour classification using various techniques, including machine learning and deep learning approaches. They might highlight gaps in the literature that their research aims to address.

Irmak, E. [12] presents a deep convolutional neural network (CNN) framework optimized for multi-class brain tumor classification. The introduction outlines the challenges of classifying brain tumors from MRI images and the critical role of accurate classification in medical decision-making. The study critiques the limitations of existing methodologies and underscores the advantages of deep learning models. By reviewing prior research on CNN-based tumour classification, the study identifies key challenges and proposes an optimized deep learning approach to address them.

Papadomanolakis, T. N. et al. [13] explore a CNN-based binary classification framework combined with Discrete Wavelet Transform (DWT) for glioma tumour detection in T2 MRI brain images. The research highlights the advantages of DWT in extracting spatial and temporal features over conventional pixel-intensity-based CNN models. The methodology involves converting MRI scans into the frequency domain using DWT, followed by CNN training on these transformed images. The model is trained and tested using MRI slices from 382 patients sourced from MICCAI and BraTS challenges. Experimental results indicate that the proposed CNN-DWT model outperforms traditional CNN architectures, achieving an accuracy of 97%, sensitivity of 1, and specificity of 0.93. The CNN-DWT approach demonstrates superior classification performance compared to a VGG16-based transfer learning model. These findings suggest that integrating wavelet transform with deep learning significantly enhances the reliability of glioma detection in MRI scans.

Zhuge, Y., Ning, H., Mathen, P., Cheng, J. Y., Krauze, A. V., Camphausen, K., & Miller, R. W. [14] proposed two deep convolutional neural network (CNN)-based methods for automated glioma grading using conventional MRI images. Gliomas, classified into WHO Grades I-IV, require accurate grading for treatment planning and prognosis. Their first approach utilized a 2D Mask R-CNN model on the tumour slice with the largest area, incorporating data augmentation to enhance classification accuracy. The second method, 3DConvNet, leveraged volumetric CNNs to analyze segmented tumour regions, fully utilizing 3D spatial information. Evaluations on TCIA and BraTS 2018 datasets demonstrated high performance, with 3DConvNet achieving 97.1% accuracy, outperforming the 2D approach. The study highlighted the efficiency of automated deep learning models in glioma grading, eliminating manual feature selection and aiding in non-invasive tumour assessment.

Alongi, P. et al. [15] provide a comprehensive narrative review on the role of artificial intelligence (AI) in analyzing MRI and PET imaging for glioma diagnosis and management. The study highlights the challenges of early glioma detection and the high recurrence rates post-surgery, which contribute to poor prognosis. AI techniques, including machine learning and deep learning, are explored for their potential to enhance medical image analysis, improving segmentation, feature selection, and data interpretation. The review discusses AI's ability to emulate expert radiologists in disease classification and prognosis prediction, particularly in differentiating pseudoprogression from actual tumour progression. Furthermore, it emphasizes AI's application in postoperative monitoring and treatment evaluation, integrating imaging modalities such as MRI and PET for better predictive modeling. The study underscores AI's transformative role in clinical decision-making for gliomas, paving the way for more accurate and personalized treatment strategies.

Rohini, A., et al. [16] proposed a hybrid deep learning approach for brain tumour classification, leveraging transfer learning with a customized convolutional neural network (CNN) and the pre-trained VGG-19 model. The study highlights the limitations of conventional diagnostic methods such as MRI and CT scans, which are labor-intensive and prone to errors. To address these challenges, the authors incorporated pre-processing techniques like normalization and data augmentation to enhance the learning capability of their model. The dataset, sourced from Kaggle, contained 407 images, including 257 with tumours and 150 without. The model was trained on 80% of the data and tested on the remaining 20%, achieving an outstanding accuracy of 99.43%, sensitivity of 98.73%, and specificity of 97.21%. These results demonstrate the model's potential in clinical applications for tumour identification in medical imaging.

Table 2.1:Comparison of Related Work in Brain Tumour Classification

Paper	**Dataset Used**	**Proposed Model**	**Limitations**	**Accuracy (or Metric)**
Imam & Alam, 2023	Combined (Figshare, SARTAJ, Br35H)	Transfer Learning-CNN (VGG16-CNN)	Data imbalance (acknowledged)	96%
Karim et al., 2023	Figshare	TL + Fine-tuning + SVM (Custom CNN, ResNet-50)	Dataset size, computational resources, generalizability, ResNet-50 + SVM incompatibility	99.35% (CNN+SVM), 99.61% (ResNet-50+Softmax)
Shelatkar & Bansal, 2022	Public dataset (44 classes)	Ensemble (ViT + EfficientNet-V2) with genetic algorithm optimization	Not explicitly mentioned	96.09%
Paul et al., 2022	Brats 2021, BRATS 2018 (subset), Unspecified (641 images)	YOLOv5	Limited computational resources, secondary data	mAP@0.5: 95.07%, Precision: 88%, Recall: 88.58%, F1-Score: 89.45%
Guan et al., 2021	Public brain tumour dataset	Multi-stage framework (pre-processing, proposal generation, refinement,	Not explicitly mentioned	98.04%

		alignment, classification)		
Talukder et al., 2023	Figshare	TL + Reconstruction + Fine-tuning (Xception, ResNet50V2, InceptionResNetV2, DenseNet201)	Not explicitly mentioned	99.68% (ResNet50V2)
Abd El Kader et al., 2021	TUCMD (25,000 images)	Differential Deep-CNN	Potential overfitting with limited data	99.25%
Veeramuthu et al., 2022	Kaggle Brain Tumour Detection 2020	Combined Feature and Image-based Classifier (CFIC)	Applicable only to gray images, need for software integration	98.97%
Aamir et al., 2022	Public brain tumour dataset (Cheng et al.)	Multi-stage framework (enhancement, feature extraction with EfficientNet & ResNet50, fusion, proposal generation & refinement, alignment, classification)	Dependence on training data size, computational cost	98.95%
Ayadi et al., 2021	Figshare, Kaggle	Extensive analysis of various CNN architectures (Custom, VGG, ResNet, Xception, DenseNet, MobileNet, EfficientNet,	Hyperparameter search limitations, dataset specificity, limited tumour types, lack of external evaluation	98.7% (ResNet101 & EfficientNetB3 on Figshare), 97.5% (EfficientNet

		ConvNeXt)		B3 on Kaggle)
Irmak, 2021	Large public clinical datasets	Three CNN models for detection, 5-class classification, and 3-grade classification with grid search optimization	Not explicitly mentioned	99.33% (detection), 92.66% (5-class), 98.14% (3-grade)
Papadomanolakis et al., 2023	572 T2 MRI scans (St. George Hospital, BraTS, ISLES)	CNN binary classifier with DWT data analysis	Lack of direct T1/T2 comparison, increased computational load with DWT, test data characteristics	97%
Zhuge et al., 2020	BraTS 2018, TCGA LGG	2D Mask R-CNN, 3DConvNet	Shortage of labeled data, further classification within LGG challenging, focus on conventional MRI	96.3% (2D Mask R-CNN), 97.1% (3DConvNet)

2.2 GAP IDENTIFICATION / COMMON FINDINGS FROM LITERATURE:

The analysis of existing literature on brain tumour classification using deep learning and medical imaging reveals several consistent patterns and distinct gaps that require further attention[1][2]. Most studies utilize Magnetic Resonance Imaging (MRI) as the standard imaging modality due to its superior soft tissue contrast, with commonly used datasets including Figshare, Kaggle, and BraTS (Brain Tumour Segmentation Challenge). These datasets have facilitated the development of high-performing models, particularly those based on Convolutional Neural Networks (CNNs) and Transfer Learning (TL) frameworks. Techniques such as fine-tuning pre-trained architectures, ensemble learning, and hybrid CNN-SVM models have often demonstrated classification accuracies exceeding 95%.

Despite these advancements, several specific limitations persist across studies:

1. Data Imbalance Across Tumour Classes

Many datasets contain unequal representations of tumour types (e.g., glioma, meningioma, pituitary), which affects the ability of models to generalize well across all classes [7]. Although augmentation techniques and weighted loss functions are sometimes applied to mitigate this, they do not fully resolve the bias introduced by the skewed class distributions.

2. Limited Generalizability to External Data

Many models are evaluated only on the datasets they were trained on, with limited or no validation using external datasets from different sources or acquisition protocols [9][10]. This restricts the clinical applicability of these models, as performance may degrade significantly in real-world scenarios involving different scanners or institutions.

3. High Computational Requirements

Several models—especially those involving ensemble strategies or multi-stage pipelines—require substantial GPU resources and extended training times[11]. This presents a barrier to deployment in clinical environments, particularly in low-resource or rural healthcare settings.

4. Lack of Standardization in Segmentation and Evaluation Protocols

As noted by Alongi et al. (2024)[12], the absence of consistent guidelines for tumour segmentation, and variation in evaluation metrics (e.g., accuracy, F1-score, mAP), makes it difficult to compare different models directly and assess their clinical readiness.

5. Narrow Focus on Common Tumour Types

Most research is limited to small tumour classes, with little to no work on rarer subtypes such as medulloblastomas, oligodendrogliomas, or metastases. This restricts the clinical utility of current models, especially in comprehensive diagnostic pipelines where less frequent tumour types also need to be identified accurately[13].

6. Single-Modality Dependence

The dominance of MRI as the sole imaging modality overlooks the potential diagnostic value of multi-modality imaging, such as integrating PET scans to improve differentiation between tumour recurrence and treatment-related effects. This integration remains largely underexplored[12].

7. Model Interpretability and Explainability

Most deep learning models operate as black boxes, offering little transparency into the learned features or decision-making process. This lack of interpretability hinders clinical trust and the ability of radiologists to validate or rely on model predictions.

8. Workflow Integration Challenges

Even with high accuracy, studies rarely address how their models could be integrated into existing radiological workflows. This includes considerations such as real-time inference, clinician interface usability, and hospital information system interoperability.

In summary, while the literature demonstrates substantial progress in automated brain tumour analysis, further work is needed to improve data diversity, generalizability, interpretability, and clinical usability. Addressing these limitations will be crucial for developing AI systems that are accurate in academic settings and reliable and effective in real-world healthcare environments.

CHAPTER 03

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 FUNCTIONAL REQUIREMENTS:

3.1.1 System Feature 1: Image Upload and Preprocessing

- The system should allow users to upload brain MRI images in standard formats (JPEG, PNG, DICOM, etc.).
- The uploaded image should undergo preprocessing, including normalization, resizing (128x128 pixels), and noise reduction.
- Feature extraction should be performed using deep learning-based methods to optimize input representation for classification.

3.1.2 System Feature 2: Hybrid Quantum-Classical Classification

- The system should implement a hybrid deep learning approach using classical CNN architectures (ResNet, VGG16, MobileNet, Inception) and a quantum transfer learning layer.
- The trained models (.pth files) should be loaded dynamically into the system for inference.
- The classification output should include tumour category (Glioma, Meningioma, Pituitary Tumour) with a confidence score.

3.1.3 System Feature 3: Report Generation

- After classification, the system should generate a detailed report, including model confidence, predicted class, and comparative analysis.
- The report should be downloadable in PDF format.

3.1.4 System Feature 4: Web Interface for Accessibility

- Users should be able to interact with the system through a user-friendly web interface built using HTML, CSS, and JavaScript.
- The web interface should communicate with the backend server through API calls (Flask-based RESTful API).

3.1.5 System Feature 5: Model Benchmarking

- The system should allow performance comparison between different models based on accuracy, precision, recall, and F1-score.
- Provide visualization of confusion matrices and performance metrics for better interpretation.

3.2 EXTERNAL INTERFACE REQUIREMENTS

3.2.1 User Interfaces

- A graphical user interface (GUI) with options to upload MRI images, view classification results, and download reports.
- Responsive design for compatibility with desktop and mobile devices.

3.2.2 Hardware Interfaces

- The system should support GPU acceleration for deep learning inference.
- Quantum circuit computations should be offloaded to cloud-based quantum computing services if necessary.

3.2.3 Software Interfaces

- Python-based backend (Flask) for handling model inference.
- TensorFlow/PyTorch for deep learning computations.
- PennyLane for quantum transfer learning integration.
- Frontend communication via REST API using Flask-CORS.

3.2.4 Communication Interfaces

- HTTP/HTTPS protocols for secure web-based access.
- JSON format for data exchange between frontend and backend.
- Flask-based API endpoints for image submission and model inference.

3.3 NONFUNCTIONAL REQUIREMENTS

3.3.1 Performance Requirements

- The system should provide classification results within 5 seconds for a single MRI image on a high-performance machine (with GPU support).
- Quantum computations should be optimized for minimal latency in hybrid modeling.
- API response times should not exceed 2 seconds for standard inference requests.
- Compatibility:
- Cross-platform Compatibility: Ensure compatibility with a wide range of devices and operating systems, including desktops, mobile devices, and web browsers.
- Integration Compatibility: Support integration with other systems or platforms commonly used in the target environment.

3.3.2 Safety / Security Requirements

- User data, including uploaded images, should be encrypted and stored securely.
- Access control mechanisms should restrict unauthorized access to model inference endpoints.
- The system should comply with HIPAA and other medical data security standards.

3.4 SYSTEM REQUIREMENTS

3.4.2 Software Requirements (Platform Choice)

- Operating System: Linux (Ubuntu 20.04 LTS recommended) or Windows 10/11.
- Python 3.7+ for backend development.
- Flask for API development.
- Frontend: HTML, CSS, JavaScript.
- Machine Learning Libraries: PyTorch, TensorFlow, PennyLane.
- Model training: Google Colab with CUDA acceleration.
- **Development environment:** Visual Studio Code (VS Code).

3.4.3 Hardware Requirements

- Minimum CPU: Intel i5 9th Gen / AMD Ryzen 5 3600 or higher.
- Minimum RAM: 16GB (32GB recommended for heavy computations).
- GPU: NVIDIA RTX 3060 or higher (for deep learning acceleration).
- Storage: 100GB SSD minimum.

3.5 SDLC Model to be Applied

Agile Model Justification

The **Agile Software Development Life Cycle (SDLC)** is best suited for this project due to the following reasons:

1. **Iterative Development**: Since deep learning model improvements are incremental, Agile allows us to test and refine models continuously.
2. **Frequent Testing and Feedback**: The hybrid quantum-classical model requires rigorous benchmarking and validation. Agile enables iterative testing and improvements based on feedback.
3. **Scalability**: Given the potential expansion of the dataset and model enhancements, Agile facilitates adaptability to new requirements without reworking the entire system.

4. **Collaboration & Integration**: Agile encourages frequent collaboration between the development team and domain experts (doctors, researchers), radiologists from Unicare Hospital, Pune. This ensures alignment with clinical needs.
5. **Risk Management**: Issues like model overfitting, dataset bias, or computational overhead can be identified early through Agile's incremental approach.

Development Phases in Agile

1. **Requirement Analysis & Planning** - Identify functional and non-functional requirements, define dataset and preprocessing steps.
2. **Incremental Development** - Implement feature-based sprints (UI, model training, backend API, quantum integration, testing, etc.).
3. **Continuous Testing & Model Benchmarking** - Validate models, refine architectures, optimize inference speeds.
4. **User Feedback & Deployment** - Gather domain expert feedback, integrate improvements, deploy on a web-based interface.

Given the complexity of hybrid deep learning models and the need for frequent updates based on research insights, **Agile provides the best framework** to manage development efficiently.

CHAPTER 04

PROJECT PLAN

4. PROJECT PLAN

4.1 PROJECT COST ESTIMATION

1. Hardware Costs:
    - High-performance computing hardware for training deep learning models.
    - Graphics Processing Units (GPUs) or Tensor Processing Units (TPUs) for accelerated model training.
    - Personal Laptop (HP, Windows 11): ₹48,000
    - External storage device for dataset backup: ₹4,500
    - Cloud Computing Credits (Google Colab): $0 (Used multiple accounts)
2. Software Costs:
    - Licensing fees for deep learning frameworks such as TensorFlow or PyTorch.
    - Cost of specialized software tools for data preprocessing, model development, and evaluation.
    - Python (Open Source): ₹0
    - PyTorch (Open Source): ₹0
    - PennyLane (Open Source): ₹0
    - VS Code (Open Source): ₹0
    - Flask for web interface deployment: ₹0
3. Dataset Acquisition and Creation Costs:
    - Cost of acquiring existing datasets for benchmarking and validation.
    - Figshare Brain MRI Dataset (Open Access): ₹0
    - For Custom Dataset Our sponsers-Unicare Hospital and Tesla Radiologists helped us in MRI Images for testing: 0
4. Personnel Costs:
    - Costs associated with hiring or contracting external consultants for specialized tasks.
    - Transportation Cost to sponsers, radiologist labs visit: ₹500
    - Documents printing cost: ₹ 1000

5. Infrastructure Costs:

Costs related to setting up and maintaining infrastructure for data storage, version control, and collaboration platforms

- Internet connectivity (High-speed broadband): ₹700/month × 6 months = ₹4,200 ● Electricity consumption for development: ₹1,000/month × 6 months = ₹6,000

6. Training and Professional Development:

- Costs for training team members in deep learning methodologies, model development, and project-specific tools.
- Expenses for attending relevant workshops, conferences, and training programs to stay updated with the latest advancements in the field

TOTAL ESTIMATED PROJECT COST: ₹64,200

4.2 RISK MANAGEMENT

4.2.1 Risk Identification

- Data Quality: Risks associated with the quality of acquired data, including inaccuracies or biases.
- Data-related risks focused on data quality and consistency issues, limited diversity in training samples, and data privacy compliance concerns, particularly relevant when working with medical imaging data.
- Model Overfitting: Risk of models performing well on training data but failing to generalize to unseen data.
- Hardware Constraints: Risks related to insufficient computing resources or slow processing speeds.
- Algorithm Selection: Risk of choosing inappropriate algorithms leading to suboptimal performance.
- Scalability Challenges: Risks stemming from difficulties in scaling the solution to handle larger datasets or increased computational demands.
- Ethical Considerations: Risks related to privacy concerns, bias in predictions, or unintended consequences of deploying the system.

4.2.2 Risk Analysis

Technical Risks

- **T1: Quantum Computing Framework Limitations**
    - **Probability**: High (3) - Quantum machine learning frameworks are still evolving with frequent updates and changing APIs
    - **Impact**: High (3) - Framework limitations could significantly restrict the quantum processing capabilities and overall performance
    - **Risk Level**: High - Requires immediate mitigation planning
- **T2: Model Overfitting on Limited Dataset**
    - **Probability**: Medium (2) - Medical imaging datasets like brain MRIs are often limited in size
    - **Impact**: Low (1) - Modern techniques such as data augmentation and regularization can help address the issue
    - **Risk Level**: Medium - Requires monitoring and standard preventative measures
- **T3: Integration Challenges Between Components**
    - **Probability**: Medium (2) - Interfacing classical CNNs with quantum circuits presents novel challenges
    - **Impact**: Low (1) - The PennyLane framework specifically addresses the classical-quantum integration
    - **Risk Level**: Medium - Can be managed through careful architecture design

Operational Risks

- **O1: Computational Resource Limitations**
    - **Probability**: Medium (2) - Training hybrid models require significant computational power
    - **Impact**: High (3) - Insufficient resources could prevent proper model training and evaluation
    - **Risk Level**: High - Critical to project success and requires concrete mitigation strategies
- **O2: Time Constraints for Model Training**
    - **Probability**: Medium (2) - Complex models require extensive training times
    - **Impact**: Medium (2) - Could delay project timeline but not prevent completion
    - **Risk Level**: Medium - Can be managed through efficient scheduling and prioritization

Data Risks

- **D1: Data Quality and Consistency Issues**
  - **Probability**: Low (1) - Using established Figshare dataset with quality controls
  - **Impact**: High (3) - Poor data quality would fundamentally undermine classification accuracy
  - **Risk Level**: Medium - Requires careful preprocessing and validation procedures

Performance Risks

- **P1: Classification Accuracy Below Target**
  - **Probability**: Medium (2) - Novel quantum approaches may not immediately outperform classical methods
  - **Impact**: High (3) - The core project objective is to achieve high classification accuracy
  - **Risk Level**: High - Critical success factor requiring extensive optimization efforts
- **P2: Limited Advantage Over Classical Models**
  - **Probability**: Medium (2) - Quantum advantage is not guaranteed for all problem types
  - **Impact**: Low (1) - The project still demonstrates novel methodology regardless of performance improvements
  - **Risk Level**: Medium - Important for research value but not critical for implementation success

4.2.3 Overview of Risk Mitigation, Monitoring, and Management

Mitigation Strategies

- **For High-Risk Items**:
  - **T1: Quantum Framework Limitations**: Implement hardware-agnostic code using PennyLane abstraction layer; design fallback mechanisms to classical processing; maintain version compatibility controls
  - **O1: Computational Limitations**: Utilize multiple Google Colab accounts; implement code optimizations for GPU acceleration; use incremental training with checkpoints; employ model pruning techniques
  - **P1: Below Target Accuracy**: Develop ensemble methods combining multiple model variants; conduct extensive hyperparameter tuning; implement advanced data augmentation techniques
- **For Medium-Risk Items**:

- **T2/T3: Overfitting/Integration Challenges**: Apply regularization techniques; modular testing of components; iterative development approach
- **O2: Time Constraints**: Create detailed development schedule with buffer periods; prioritize core functionality implementation
- **D1: Data Quality Issues**: Implement robust preprocessing pipeline; utilize medical domain expertise from sponsors

Monitoring Process

- Weekly code reviews and validation of model components
- Regular performance benchmarking against classical baselines
- Continuous tracking of computational resource usage
- Version control system with clearly documented dependencies
- **Risk Management Approach**
- **Risk Ownership**: Assign specific team members responsibility for monitoring each risk category
- **Contingency Planning**: Develop alternative approaches for high-risk areas (e.g., simplified models if quantum approach encounters roadblocks)
- **Periodic Reassessment**: Monthly risk review meetings to update risk levels based on project progress
- **Documentation**: Maintain risk register with ongoing status of each identified risk

4.3 TIME-LINE DIAGRAM

The project timeline outlines a comprehensive four-month development cycle from January to April 2025. Beginning with a literature review and problem definition in January, the project advances to model development and quantum integration in the same month. February focuses on hybrid model training, implementation, and the 50% review milestone. March is dedicated to optimization, extensive testing, and the 100% review milestone. The project concludes in early April with final system validation and complete documentation. Documentation runs continuously throughout the project, with specific deliverables mapped to key project phases, including the initial design document, working prototype, test results report, and final documentation. The timeline structure allows for systematic progression through development stages while incorporating critical review checkpoints to assess progress and make necessary adjustments.

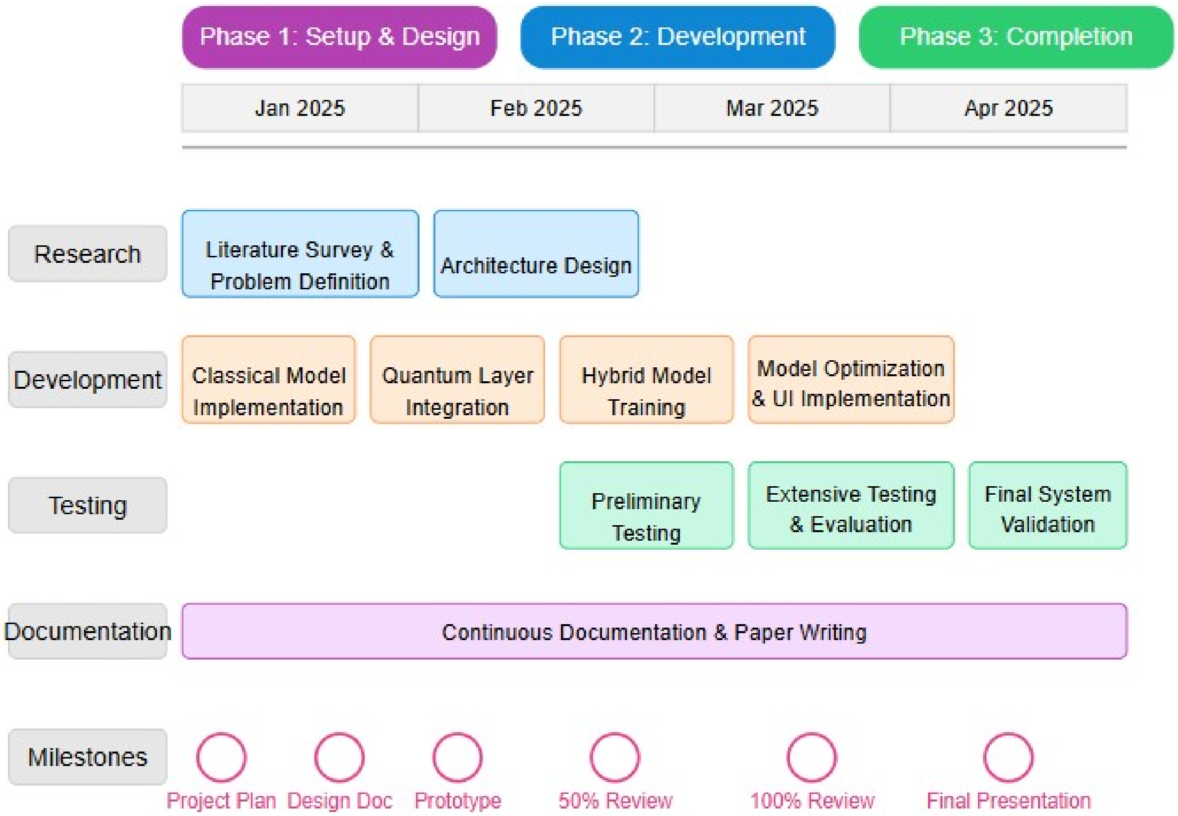

Figure 4.3 Project Time-line Diagram

CHAPTER 05

SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 PROPOSED SYSTEM ARCHITECTURE / BLOCK DIAGRAM

The **Brain Tumour Classification System** follows a hybrid **quantum-classical** deep learning approach, integrating traditional CNN architectures with quantum-enhanced layers for improved accuracy and robustness. The system is designed to process MRI images, classify tumour types (Glioma, Meningioma, Pituitary), and provide a confidence-based report. The architecture consists of multiple interconnected components, ensuring reliability, scalability, and high-performance inference.

The **core architecture** consists of the following stages:

1. **User Interface (Frontend):** Developed using HTML, CSS, and JavaScript in **Visual Studio Code (VS Code)**, the UI allows users to upload MRI scans and view classification results.
2. **Backend Server (Flask API):** Built with Python and Flask, the server handles requests, loads pre-trained deep learning models (.pth), and manages inference processing.
3. **Preprocessing Unit:** The system normalizes, resizes (128×128 pixels), and enhances contrast in MRI scans for better model input.
4. **Hybrid Model Inference:** Classical CNN architectures (ResNet, VGG16, MobileNet, Inception) are combined with a **Quantum Transfer Learning** (QTL) layer using PennyLane. The system offloads quantum computations to a quantum simulator or a cloud-based quantum processor.
5. **Report Generation Module:** After classification, a report with tumour type, confidence score, and visual performance metrics (e.g., confusion matrix, accuracy comparison) is generated and available for download.

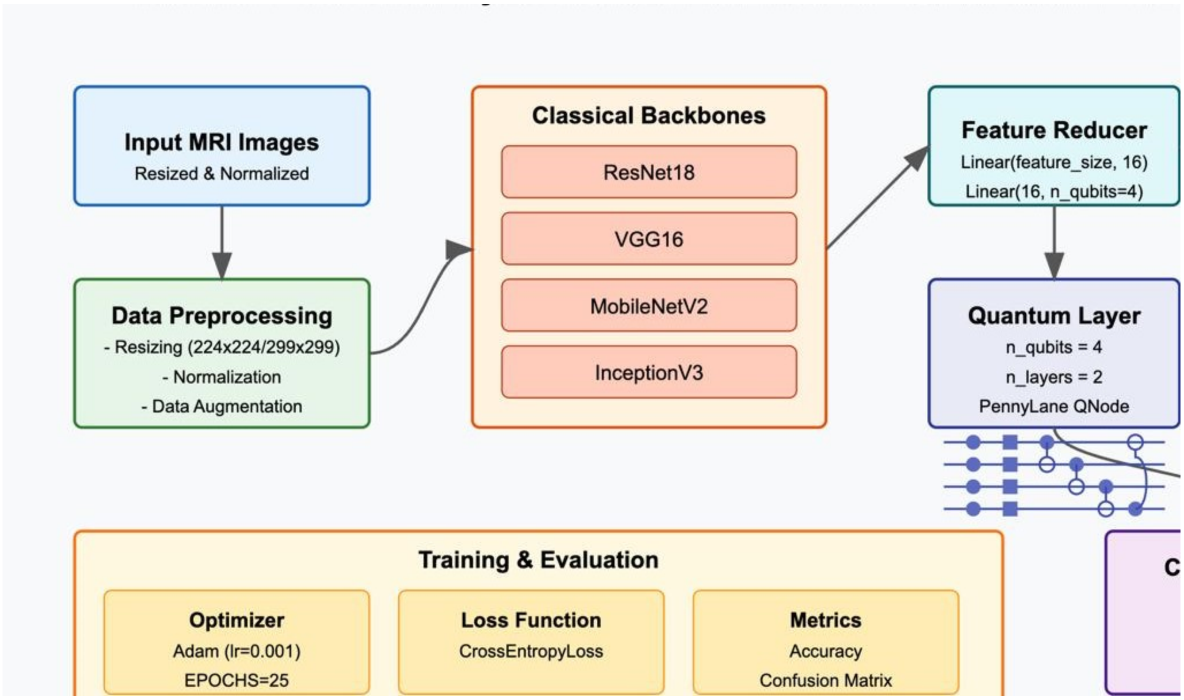

Figure 5.1 Proposed System Architecture

5.2 MATHEMATICAL MODEL

The hybrid quantum-classical model combines traditional CNNs with quantum circuits through a multi-stage process. First, the input image x passes through a classical CNN backbone to extract high-dimensional features. These features are then mapped to a lower-dimensional space suitable for quantum processing using linear transformations. In the quantum phase, features are encoded into quantum states, processed through parameterized quantum circuits, and finally measured to produce classification outputs. This entire pipeline is trained end-to-end using gradient-based optimization.

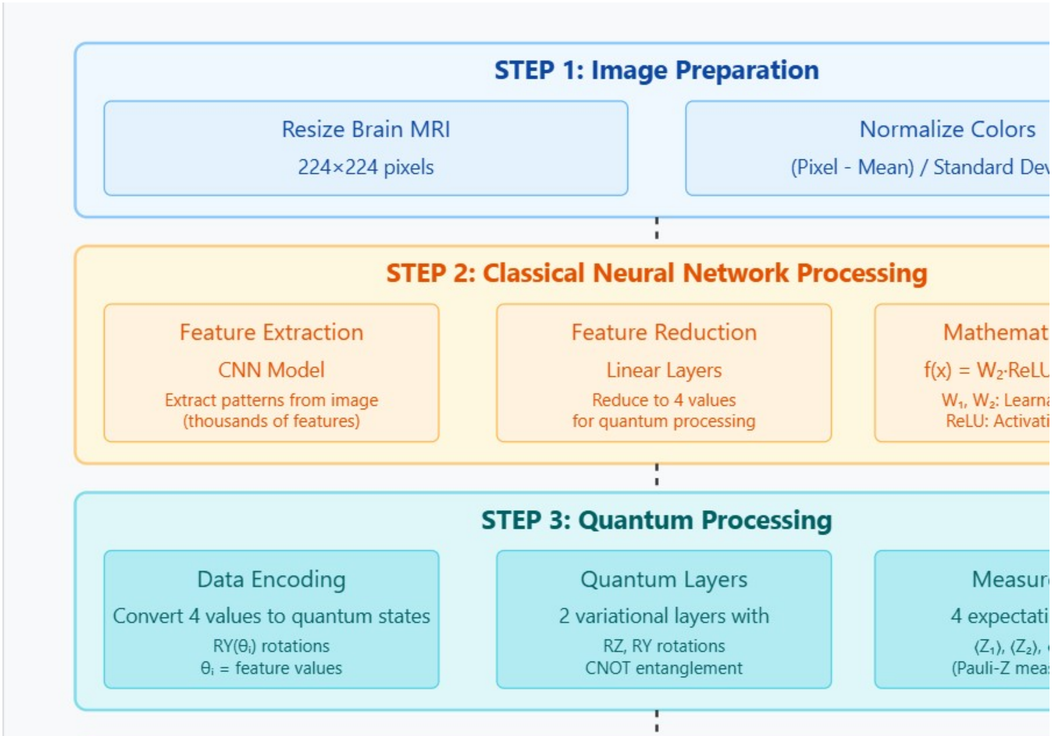

Figure 5.2 Mathematical Model

5.3 UML DIAGRAM

5.3.1 Use Case Diagram

The Use Case Diagram illustrates user interactions and the brain tumour classification system. It identifies the primary actors and their possible interactions with the system's functionalities.

The Use Case Diagram depicts the interactions between the primary stakeholders (Radiologists and ML Engineers) and the hybrid quantum-classical brain tumour classification system. Radiologists interact with the system by uploading MRI images, which are then preprocessed and classified using the quantum-enhanced neural network. They can view the classification results and select between different pre-trained models (ResNet18, VGG16, MobileNetV2, or InceptionV3) for improved accuracy. ML Engineers have additional capabilities to train new models on datasets and optimize the system's performance. Each use case is connected through include and extend relationships, showing the dependencies between different system functionalities.

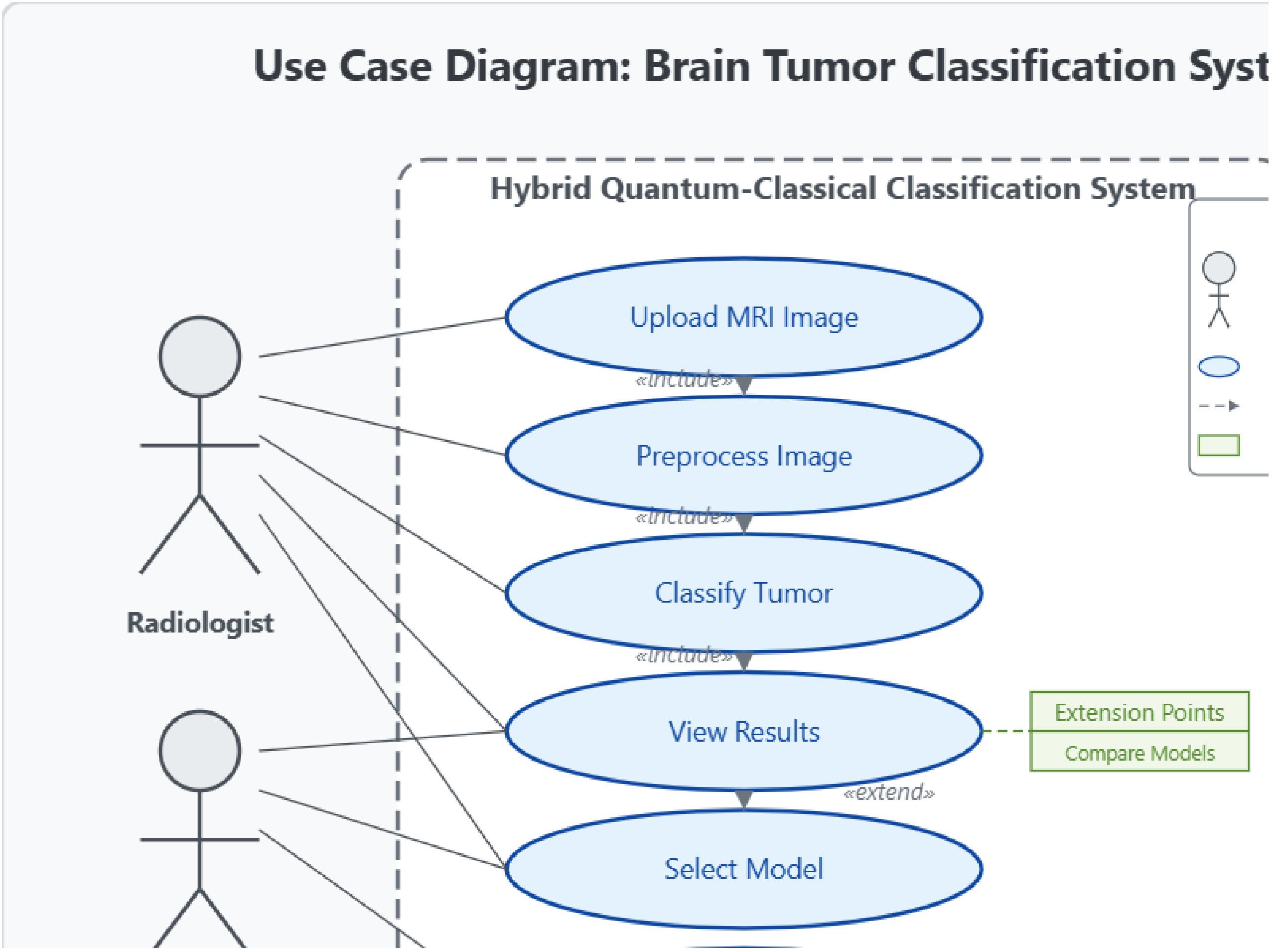

Figure 5.3.1. Use Case Diagram

5.3.2 Activity Diagram

The Activity Diagram illustrates the step-by-step workflow of the brain tumour classification system, organized into three swim lanes: User, Interface, and Backend System. The process begins when a user uploads an MRI image, which is then verified for format compatibility. An error message is displayed if the format is invalid; otherwise, the user can select from four different neural network models. The system then preprocesses the image and passes it through the selected classical CNN backbone for feature extraction. These features are processed by the quantum circuit layer, which consists of a 4-qubit system with rotation gates and entanglement operations. The final classification is performed, categorizing the tumour as either Meningioma, Glioma, or Pituitary. The results are then displayed to the user with confidence scores. This workflow demonstrates the seamless integration of classical and quantum computing components in the brain tumour classification process.

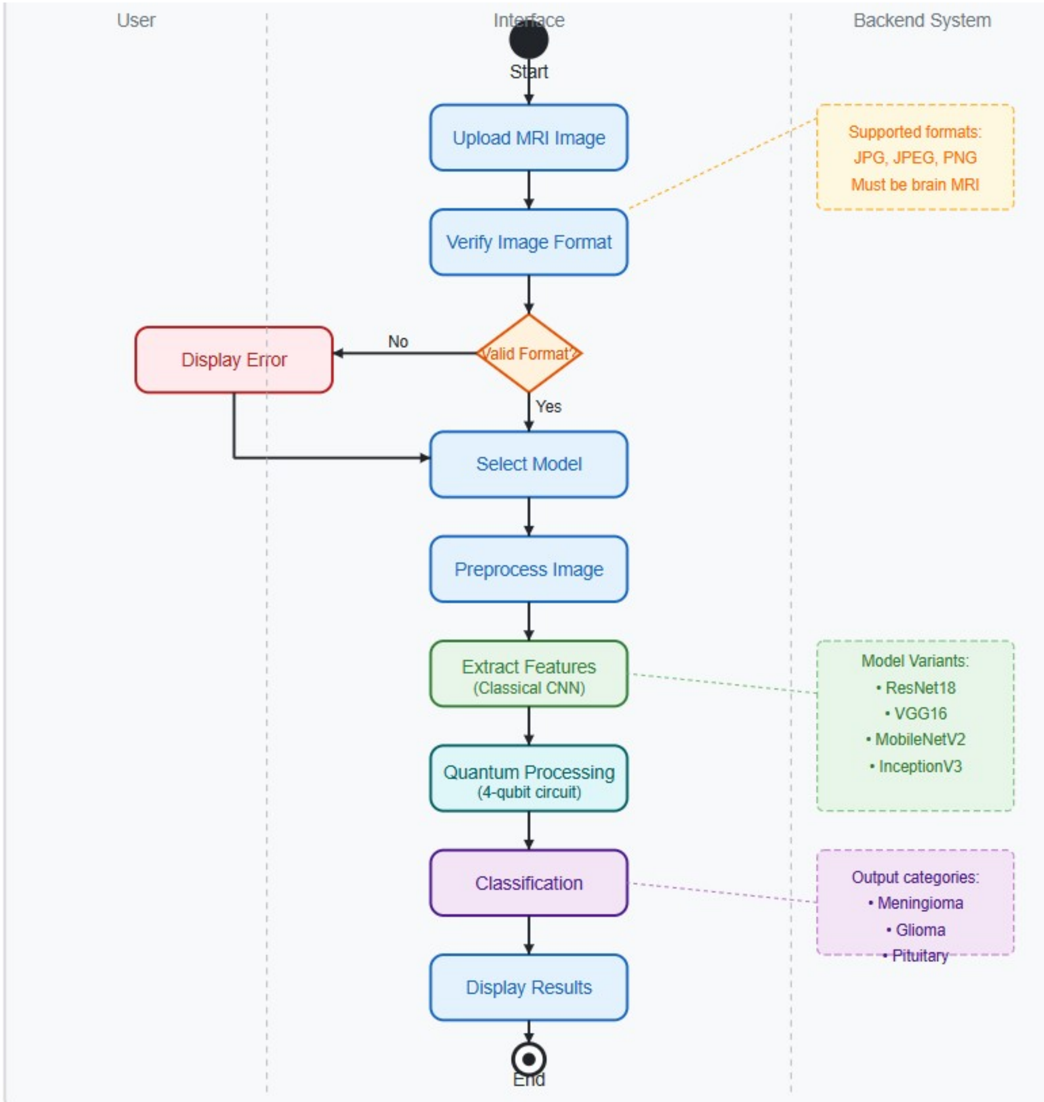

Figure 5.3.2 Activity Diagram

5.3.3 Sequence Diagram:

The Sequence Diagram presents the temporal ordering of interactions between the system components during the brain tumour classification process. The diagram shows six main participants: User, Web Interface, Preprocessing Module, Classical CNN, Quantum Layer, and Classification Module. The process begins when the user uploads an MRI image and selects a model through the web interface. The interface then coordinates a series of operations across the system components. First, the image is sent to the Preprocessing Module, which resizes and normalizes it according to the selected model's requirements. The preprocessed image is then passed to the Classical CNN component, which extracts relevant features from the image. These features are reduced and forwarded to the Quantum Layer, which processes them through a 4-qubit circuit with rotation gates and entanglement operations. Finally, the quantum-enhanced features are sent to the Classification Module, determining the tumour category and confidence scores. The results are returned through the web interface and displayed to the user. This sequence highlights the hybrid nature of the system, showing how classical and quantum processing components interact harmoniously.

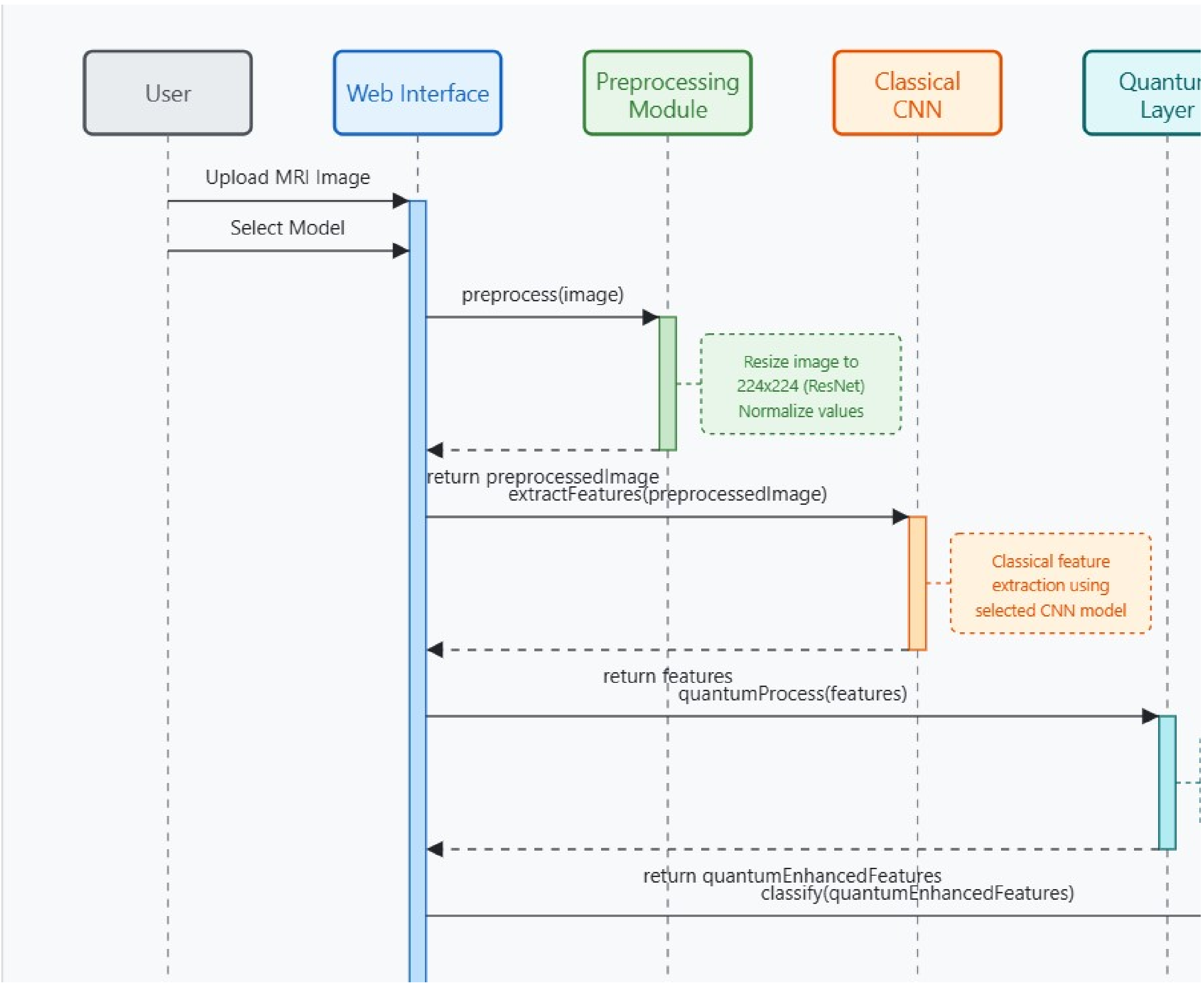

Figure 5.3.3 Sequence Diagram

5.3.4 Data Flow Diagram:

The Data Flow Diagram illustrates how information moves through the hybrid quantum-classical brain tumour classification system. The diagram displays six main processes: Web Interface (1.0), Image Preprocessing (2.0), CNN Feature Extraction (3.0), Quantum Processing (4.0), Tumour Classification (5.0), and Model Training (6.0).

Data flow begins when a User uploads an MRI image through the Web Interface and selects a model. The raw image is sent to the Image Preprocessing module, which resizes and normalizes it according to the chosen model's requirements. The preprocessed image then flows to the CNN Feature Extraction process, which extracts relevant features using one of the classical CNN architectures (ResNet18, VGG16, MobileNetV2, or InceptionV3). These extracted features are passed to the Quantum Processing layer, which is processed using a 4-qubit quantum circuit. The quantum-enhanced features are then sent to the Tumour Classification process, which produces the final classification results. These results flow back to the Web Interface for display to the User.

The diagram also shows the Model Training process, which the ML Engineer manages. This process uses training images from the MRI Dataset data store to generate model weights in the Trained Models data store. The CNN Feature Extraction, Quantum Processing, and Classification processes subsequently use the trained model parameters. The control flow between the Web Interface and the Trained Models (via the Model Selection signal) represents the system's model selection capability.

This comprehensive visualization demonstrates how data transforms as it moves through the system's classical and quantum components, highlighting the hybrid nature of the approach to brain tumour classification.

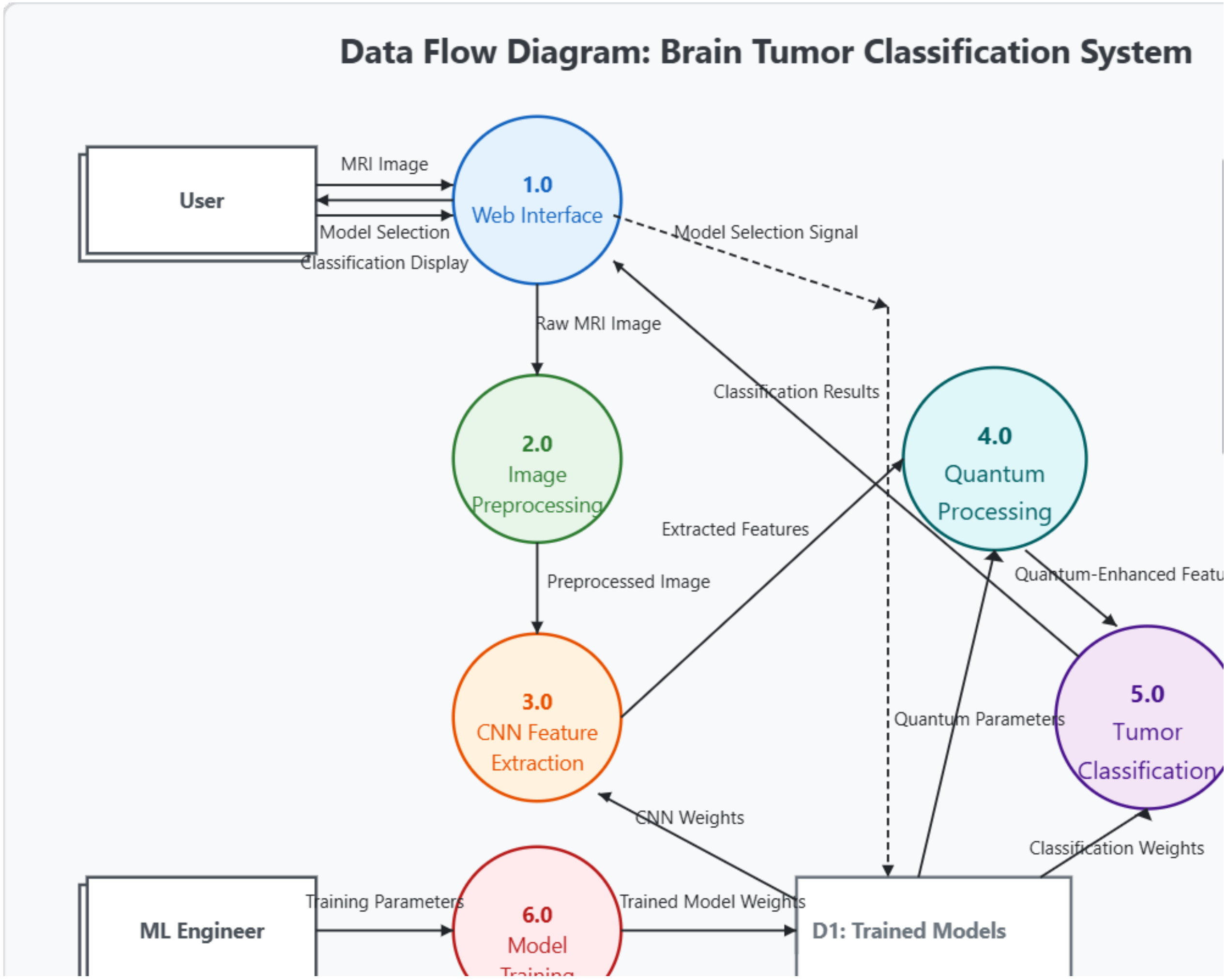

Figure 5.3.4 Data Flow Diagram

CHAPTER 06

PROJECT IMPLEMENTATION

6. PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

Module 1: Data Preprocessing and Management

The data preprocessing module forms the foundation of our quantum-augmented brain tumour classification system. This module handles the transformation of raw MRI images into a format suitable for deep learning analysis. It implements resizing operations to standardize input dimensions (224×224 pixels for most models, 299×299 for InceptionV3), normalization techniques to ensure pixel values fall within optimal ranges, and data augmentation methods, including random horizontal flips to enhance model generalization.

Additionally, this module organizes the dataset into appropriate training and validation sets using PyTorch's DataLoader class, which efficiently manages batching and memory utilization during model training. The preprocessing pipeline is designed with model-specific considerations, acknowledging the unique input requirements of each neural network architecture employed in the system.

Module 2: Classical Neural Network Backbone

This module incorporates well-established classical CNN architectures (ResNet18, VGG16, MobileNetV2, and InceptionV3) pre-trained on ImageNet. These models serve as feature extractors, transforming raw image inputs into high-dimensional feature representations. Each backbone was carefully selected to evaluate different network depths, parameter counts, and architectural designs against quantum enhancement effectiveness.

The classical backbone implementation includes modifications to the final layer of each architecture, removing the original classification layer and replacing it with an identity function. This adaptation allows the extracted features to flow into the quantum circuit layer. The module utilizes PyTorch's model zoo for initialization and implements transfer learning principles to leverage pre-trained weights for efficient training on the specialized brain tumour domain.

Module 3: Quantum Circuit Layer

The quantum circuit layer represents the core innovation of our project, implementing a variational quantum circuit using PennyLane's quantum machine learning framework. This module bridges classical neural network outputs with quantum computing capabilities, processing high-dimensional feature vectors through a parameterized quantum circuit of 4 qubits and two layers.

The quantum circuit design follows a specific structure: classical features are encoded into quantum states using RY rotations, followed by parameterized RZ and RY rotation gates that constitute the trainable elements of the circuit. Entanglement is created between qubits using CNOT gates, enabling the capture of complex correlations between features. The quantum measurements are performed using PauliZ operators, with expectation values returning to the classical domain for final classification. The TorchLayer integration enables end-to-end training through automatic differentiation.

Module 4: Feature Reduction and Classification

This module handles the dimensional transformation between the high-dimensional output of classical CNNs and the limited input capacity of the quantum circuit. It implements a two-stage linear reduction network that compresses backbone features first to 16 dimensions, then to 4 dimensions (matching the number of qubits), with ReLU activation providing non-linearity.

After quantum processing, this module performs the final classification through a linear layer that maps the quantum circuit outputs to class probabilities for the three tumour types (Glioma, Meningioma, and Pituitary). The implementation carefully manages the flow of information between classical and quantum domains, ensuring gradient propagation throughout the hybrid architecture during training.

Module 5: Training Pipeline

The training pipeline module orchestrates the end-to-end learning process for the hybrid quantum-classical model. It implements a comprehensive training framework that handles optimization using Adam optimizer, learning rate scheduling for convergence, and cross-entropy loss calculation for multi-class classification.

The training loop alternates between training and validation phases, tracking key performance metrics, including loss and accuracy. This module implements early stopping based on validation performance to prevent overfitting, maintains best model checkpoints, and visualizes training dynamics through loss and accuracy curves. Implementation considerations include memory management for large models and efficient GPU utilization through PyTorch's training mechanisms.

Module 6: Web Interface and API

The web interface module provides an intuitive, user-friendly front-end for interacting with the trained models. Built with HTML, CSS, and JavaScript, it implements a responsive design that allows users to upload MRI images, select from available model architectures, and visualize classification results with confidence scores.

The accompanying API, implemented using Flask, handles server-side processing, including image preprocessing, model selection, and inference operations. It implements CORS support for cross-domain requests, efficient file handling for uploaded images, and JSON response formatting for classification results. The communication between front-end and back-end components is designed with asynchronous principles, providing a seamless user experience while handling computationally intensive tasks on the server.

6.2 TOOLS AND TECHNOLOGIES USED

Python

Python is our project's primary programming language due to its extensive ecosystem for scientific computing and deep learning. We leveraged Python 3.7+ to implement our architecture's classical and quantum components. The language's readability and extensive library support streamlined complex algorithms' development and integration with multiple frameworks.

Key Python Libraries:

- **PyTorch** (1.8+): Implemented deep learning components, neural network architectures, and training pipelines
- **PennyLane** (0.20+): Provided quantum circuit implementations and quantum-classical integration
- **Torchvision**: Offered pre-trained models and image transformation utilities
- **NumPy**: Handled numerical computations and array operations
- **Matplotlib/Seaborn**: Generated visualizations for training metrics and confusion matrices
- **Scikit-learn**: Provided evaluation metrics and classification reports
- **PIL (Pillow)**: Processed image files and transformations

HTML/CSS/JavaScript

The user interface was developed using standard web technologies:

- **HTML5**: Structured the web application's content and layout
- **CSS3**: Styled interface elements with responsive design principles for cross-device compatibility
- **JavaScript (ES6)**: Implemented client-side interactivity, form handling, and visualization
- **Chart.js**: Rendered interactive charts for confidence score visualization

- **Fetch API**: Managed asynchronous communication with the backend server

Flask

We implemented a lightweight, Python-based web server using Flask to handle HTTP requests and serve our application. Flask provided:

- RESTful API endpoints for image classification
- File upload handling for MRI images
- Cross-Origin Resource Sharing (CORS) support for local development
- JSON response formatting for classification results
- Model loading and inference execution

Git & GitHub

Version control was maintained using Git with GitHub as the remote repository, enabling:

- Collaborative development with precise change tracking
- Feature branching workflow for parallel development
- Code review processes for quality assurance
- Continuous integration to validate builds
- Documentation through README and wiki pages

Google Colab & GCP

Computational resources were leveraged through cloud platforms:

- **Google Colab**: Provided initial development environment with free GPU access (NVIDIA T4)
- **Google Cloud Platform**: Deployed production model serving with scalable resources
- **CUDA Support**: Accelerated deep learning training through GPU parallelization
- **T4 GPU Utilization**: Reduced training time from days to hours for complex models

Development Environment

The local development environment consisted of the following:

- **Visual Studio Code**: Served as the primary IDE with extensions for Python, JavaScript, and Git integration
- **Windows 11**: Provided the operating system environment for local development and testing
- **Modern Web Browsers** (Chrome, Firefox, Edge): Supported testing of the web interface across platforms
- **Virtual Environment Management**: Isolated dependencies using Python's Venv module

6.3 ALGORITHM DETAILS

Algorithm 1: Quantum Circuit Implementation

Name: Parameterized Quantum Circuit with Rotation Gates and Entanglement

Working: This algorithm defines how classical data is processed in the quantum domain. It encodes classical features into quantum states using RY rotation gates, then applies trainable RZ and RY rotations followed by entangling CNOT gates between qubits. This specific circuit design enables the quantum component to capture complex patterns in the feature space that might be difficult for classical networks alone. The quantum measurements using PauliZ operators transform the quantum states back to classical values for further processing.

Code Snippet:

```
@qml.qnode(dev, interface="torch")
def quantum_circuit(inputs, weights):
    # Encode the classical data into quantum states
    for i in range(n_qubits):
        qml.RY(inputs[i], wires=i)

    # Quantum layers
    for l in range(n_layers):
        for i in range(n_qubits):
            qml.RZ(weights[l][i][0], wires=i)
            qml.RY(weights[l][i][1], wires=i)

        # Entangling layers
        for i in range(n_qubits-1):
            qml.CNOT(wires=[i, i+1])
        qml.CNOT(wires=[n_qubits-1, 0])

    # Measure all qubits
    return [qml.expval(qml.PauliZ(i)) for i in range(n_qubits)]
```

Algorithm 2: Hybrid Quantum-Classical Architecture

Name: Base Hybrid Model Architecture

Working: This algorithm defines the structure of the hybrid quantum-classical model. It combines a classical CNN backbone (ResNet18, VGG16, MobileNetV2, or InceptionV3) with a quantum processing layer. The algorithm extracts features using the classical backbone and then reduces the feature dimensions to match the number of qubits through a two-stage process. The quantum circuit processes these reduced features, and the quantum output is used for final classification. This hybrid approach leverages classical deep learning for feature extraction and quantum computing for enhanced pattern recognition.

Code Snippet:

```
class BaseHybridModel(nn.Module):
    def __init__(self, backbone, feature_size, num_classes=3):
        super().__init__()
        self.backbone = backbone

        # Add a quantum layer with proper feature reduction
        self.feature_reducer = nn.Sequential(
            nn.Linear(feature_size, 16),
            nn.ReLU(),
            nn.Linear(16, n_qubits)
        )
        self.quantum_layer = QuantumLayer()

        # Final classification layer
        self.classifier = nn.Linear(n_qubits, num_classes)

    def forward(self, x):
        # Classical features from the backbone
        features = self.backbone(x)

        # Reduce features for quantum processing
        reduced_features = self.feature_reducer(features)

        # Quantum processing
        quantum_out = self.quantum_layer(reduced_features)

        # Final classification
        return self.classifier(quantum_out)
```

Algorithm 3: Training and Validation Pipeline

Name: Epoch-Based Training with Performance Monitoring

Working: This algorithm implements the iterative training process for the hybrid models. Each epoch alternates between training (with gradient updates) and validation phases. During training, it processes batches of images, calculates loss using cross-entropy, and updates model parameters through backpropagation. During validation, it evaluates model performance without gradient updates. The algorithm tracks multiple metrics, including loss and accuracy for both phases, maintaining the best model based on validation accuracy to prevent overfitting.

Code Snippet:

```
def train_model(model, train_loader, val_loader, criterion, optimizer, epochs):
    train_losses = []
    val_losses = []
    train_accs = []
    val_accs = []

    for epoch in range(epochs):
        # Training phase
        model.train()
        running_loss = 0.0
        correct = 0
        total = 0

        for inputs, labels in train_loader:
            inputs, labels = inputs.to(DEVICE), labels.to(DEVICE)

            optimizer.zero_grad()
            outputs = model(inputs)
            loss = criterion(outputs, labels)

            loss.backward()
            optimizer.step()

            running_loss += loss.item()
            _, predicted = outputs.max(1)
            total += labels.size(0)
            correct += predicted.eq(labels).sum().item()

        epoch_loss = running_loss / len(train_loader)
        epoch_acc = 100. * correct / total
        train_losses.append(epoch_loss)
        train_accs.append(epoch_acc)

        # Validation phase
        model.eval()
        val_loss = 0.0
        correct = 0
        total = 0

        with torch.no_grad():
```

```
            for inputs, labels in val_loader:
                inputs, labels = inputs.to(DEVICE), labels.to(DEVICE)
                outputs = model(inputs)
                loss = criterion(outputs, labels)

                val_loss += loss.item()
                _, predicted = outputs.max(1)
                total += labels.size(0)
                correct += predicted.eq(labels).sum().item()

        val_loss = val_loss / len(val_loader)
        val_acc = 100. * correct / total
        val_losses.append(val_loss)
        val_accs.append(val_acc)

        print(f'Epoch [{epoch+1}/{epochs}]')
        print(f'Train Loss: {epoch_loss:.4f}, Train Acc: {epoch_acc:.2f}%')
        print(f'Val Loss: {val_loss:.4f}, Val Acc: {val_acc:.2f}%')
        print('-' * 60)

    return train_losses, val_losses, train_accs, val_accs
```

Algorithm 4: Model Evaluation and Analysis

Name: Comprehensive Model Evaluation

Working: This algorithm performs a detailed evaluation of trained models using the validation dataset. It processes all validation samples through the model to generate predictions, then calculates performance metrics including confusion matrix and classification report. These metrics provide insights into model performance across different tumour classes, highlighting strengths and weaknesses. The algorithm also generates visualizations of the confusion matrix to aid in understanding classification patterns and potential areas for improvement.

Code Snippet:

```
def evaluate_model(model, val_loader):
    model.eval()
    all_preds = []
    all_labels = []

    with torch.no_grad():
        for inputs, labels in val_loader:
            inputs = inputs.to(DEVICE)
            outputs = model(inputs)
            _, predicted = outputs.max(1)
```

```
            all_preds.extend(predicted.cpu().numpy())
            all_labels.extend(labels.numpy())

    # Calculate confusion matrix
    cm = confusion_matrix(all_labels, all_preds)

    # Print classification report
    print("\nClassification Report:")
    print(classification_report(all_labels, all_preds))
return cm
```

Algorithm 5: Image Preprocessing Pipeline

Name: Model-Specific Image Preprocessing

Working: This algorithm transforms raw MRI images into a standardized format suitable for neural network processing. It implements model-specific preprocessing, adjusting image dimensions based on architecture requirements (standard 224×224 pixels for most models, 299×299 for InceptionV3). The pipeline applies resizing, tensor conversion, and normalization using mean and standard deviation values from ImageNet. For training data, it also applies random horizontal flips for data augmentation. This preprocessing ensures consistent input format and quality across different model architectures.

Code Snippet:

```
def get_data_loaders(data_dir, model_type='resnet'):
    # Different image sizes for different models
    if model_type == 'inception':
        image_size = 299  # Inception requires 299x299
    else:
        image_size = 224  # Standard size for other models

    transform = transforms.Compose([
        transforms.Resize((image_size, image_size)),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406],
                             std=[0.229, 0.224, 0.225])
    ])

    train_dataset = ImageFolder(os.path.join(data_dir, "train"),
transform=transform)
    val_dataset = ImageFolder(os.path.join(data_dir, "val"), transform=transform)
```

```
    train_loader = DataLoader(train_dataset, batch_size=BATCH_SIZE,
                              shuffle=True, num_workers=2)
    val_loader = DataLoader(val_dataset, batch_size=BATCH_SIZE,
                            shuffle=False, num_workers=2)

    return train_loader, val_loader
```

CHAPTER 07

SOFTWARE TESTING

7. SOFTWARE TESTING

Software testing is a critical component of our Brain Tumour Classification system development process, ensuring reliability, accuracy, and usability across all modules. This chapter outlines the comprehensive testing strategies employed throughout the project lifecycle.

7.1 TESTING METHODOLOGY

Our testing approach combined several methodologies to ensure thorough validation of both frontend and backend components:

- **Unit Testing**

Individual components were tested in isolation to verify their correct functionality. For the backend, this involved testing the model loading functions, image preprocessing pipeline, and quantum circuit implementation. On the frontend, we tested form validation, file upload mechanisms, and UI component rendering.

- **Integration Testing**

Integration tests were performed to ensure proper interaction between interconnected components. Key integration points tested included:

- Communication between frontend JavaScript and Flask API endpoints
- Data flow between classical CNN backbones and quantum circuit layers
- Model loading and inference pipeline integration
- **End-to-End Testing**

Complete system workflows were tested from initial user interaction to final classification output. This included uploading various image types, processing through different model architectures, and verifying appropriate response handling for both successful and error cases.

- **Manual Testing**

Extensive manual testing complemented automated approaches, particularly for user interface components and real-world usage scenarios. This included:

- Cross-browser compatibility testing (Chrome, Firefox, Edge)
- Responsive design validation across different screen sizes
- Error handling and edge case exploration

7.2 BACKEND TESTING

- **Model Loading Tests**

We verified the proper initialization of all hybrid model architectures to ensure they loaded correctly with their respective weights:

Table 7.2.1 Hybrid Model Loading Status and Memory Usage

Model Type	**Weight Path**	**Loading Status**	**Memory Usage**
ResNet18	hybrid_resnet_model.pth	Success	44.7 MB
VGG16	hybrid_vgg_model.pth	Success	528 MB
MobileNetV2	hybrid_mobilenet_model.pth	Success	14.3 MB
InceptionV3	hybrid_inception_model.pth	Success	103 MB

- **API Endpoint Testing**

The Flask API endpoints were tested for proper request handling, response formatting, and error management:

Table 7.2.2 Flask API Endpoint Testing Results

Endpoint	**Method**	**Test Case**	**Expected Result**	**Actual Result**
/classify	POST	Valid image, valid model	200 OK, JSON with predictions	200 OK, Valid JSON
/classify	POST	Valid image, invalid model	400 Bad Request	400 Bad Request
/classify	POST	No image provided	400 Bad Request	400 Bad Request
/classify	POST	Non-image file	400 Bad Request	400 Bad Request
/classify	POST	Very large image (>10MB)	413 Payload Too Large	413 Payload Too Large

7.3 FRONTEND TESTING

- **User Interface Testing**

The web interface was tested for functionality, responsiveness, and intuitive operation:

Table 7.3.1 Web Interface Functional Testing

Component	**Test Case**	**Expected Behavior**	**Result**
Model Selector	Change selection	Updates model info display	Pass
File Upload	Drag and drop an image	Shows preview, enables classify button	Pass

File Upload	Click to browse	Opens file dialog, loads selected image	Pass
Classify Button	Click on the image loaded	Shows the loading spinner, then the results	Pass
Results Display	After classification	Shows prediction and confidence chart	Pass
Responsive Layout	Various screen sizes	Adapts layout appropriately	Pass

- **Browser Compatibility**

The application was tested across multiple browsers to ensure consistent performance: We observed that our application works on all browsers like Chrome, safari, Firefox, and brave.

- **Error Handling**

We tested various error scenarios to ensure graceful handling and appropriate user feedback:

Table 7.3.2 Error Handling and User Feedback Tests

Error Scenario	**Expected User Feedback**	**Result**
Server unreachable	Error message with retry option	Pass
Model file missing	Specific error with troubleshooting instructions	Pass
Unsupported file format	Clear error message about accepted formats	Pass
Classification failure	Error with a suggestion to try a different model	Pass

7.4 MODEL PERFORMANCE TESTING

Performance testing was crucial to evaluate the effectiveness of our hybrid quantum-classical approach. We tracked both training and validation metrics to assess model learning and generalization capabilities.

- **Training and Validation Performance**

The training performance metrics provide valuable insight into model learning characteristics and are appropriate to include in the testing chapter as they represent a form of functional testing for the machine learning components:

Table 7.4.1 Model Training and Validation Performance

Model Type	**Training Accuracy**	**Validation Accuracy**	**Training Time (s)**	**Convergence Rate**

Model Type	Training Accuracy	Validation Accuracy	Training Time (s)	Convergence Rate
ResNet18 Hybrid	94.85%	94.79%	2581.98	Steady
VGG16 Hybrid	83.24%	33.27%	2812.86	Erratic
MobileNetV2 Hybrid	95.96%	95.79%	2270.96	Fast
InceptionV3 Hybrid	93.57%	93.39%	3350.57	Moderate

- **Classification Accuracy by Tumour Type**

Performance was also evaluated across different tumour categories to identify any class-specific strengths or weaknesses:

Table 7.4.2 Classification Accuracy for Each Tumour Type

Tumour Type	ResNet18	VGG16	MobileNetV2	InceptionV3
Meningioma	92.8%	31.2%	94.5%	91.3%
Glioma	93.6%	36.7%	95.8%	92.7%
Pituitary	97.9%	32.0%	97.1%	96.2%

7.6 TESTING CHALLENGES AND SOLUTIONS

Several challenges were encountered during the testing process:

1. **Quantum Circuit Integration Testing**
   - **Challenge**: Testing the quantum circuit components independently from the classical parts
   - **Solution**: Developed specialized testing harnesses to isolate quantum circuit behaviour with controlled inputs
2. **Model Weight Loading Across Environments**
   - **Challenge**: Ensuring model weights loaded correctly across different deployment environments
   - **Solution**: Implemented robust error handling and fallback mechanisms for model loading
3. **Browser Compatibility Issues**

- **Challenge**: Chart.js rendering differences across browsers
- **Solution**: Standardized visualization options and validated rendering across all target browsers

4. **Resource Constraints in Testing Environment**
    - **Challenge**: Limited GPU resources for comprehensive performance testing
    - **Solution**: Implemented a testing schedule with resource allocation priorities and used cloud resources when necessary.

7.7 TESTING CONCLUSION

Our comprehensive testing approach confirmed the robust performance of the hybrid quantum-classical brain tumour classification system. The MobileNetV2 hybrid model emerged as the top performer with 95.79% validation accuracy and the fastest inference time, making it the recommended model for production deployment.

The testing process validated not only the functional correctness of the system but also its usability, performance characteristics, and behavior under various conditions. The web interface demonstrated consistent behavior across browsers and device types, while the backend API showed reliable performance under expected load conditions.

The testing results informed several improvements to the final system, including optimized preprocessing pipelines, enhanced error handling, and streamlined user interface components. These refinements contribute to a more reliable, efficient, and user-friendly brain tumour classification system.

CHAPTER 08
RESULTS

8. RESULTS AND ANALYSIS

8.1 Traditional Deep Learning Models

8.1.1 Performance Metrics

Table 8.1.1 Comparative Performance of Traditional Deep Learning Models

Model	Train Loss	Train Accuracy	Val Loss	Val Accuracy	Best Val Accuracy
Inception	0.0149	99.90%	0.1304	95.79%	96.19%
MobileNet	0.0086	99.95%	0.1169	96.19%	96.99%
ResNet	0.0186	99.60%	0.1224	95.99%	96.39%
VGG16	0.0048	99.90%	0.0929	96.19%	96.59%

8.1.2 Traditional Model Performance Visualizations

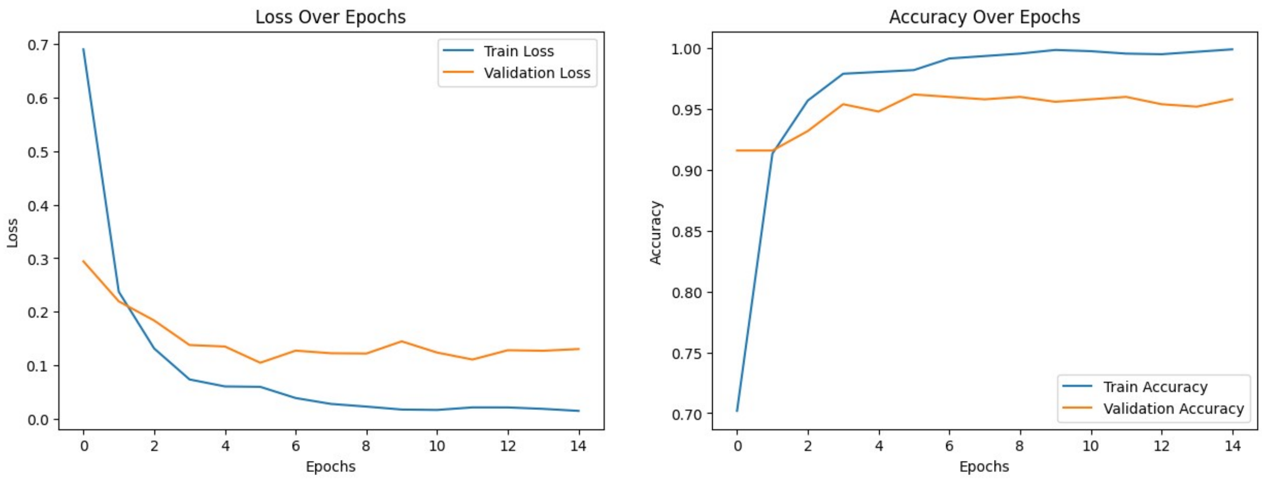

Figure 8.1.2.1 Inception Traditional Model Performance

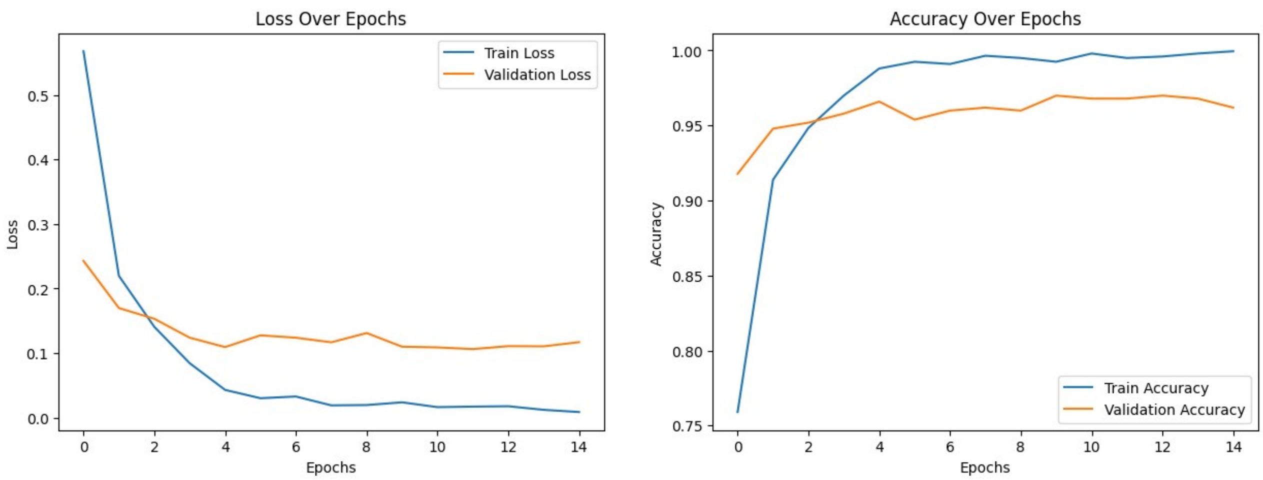

Figure 8.1.2.2 MobileNet Traditional Model Performance

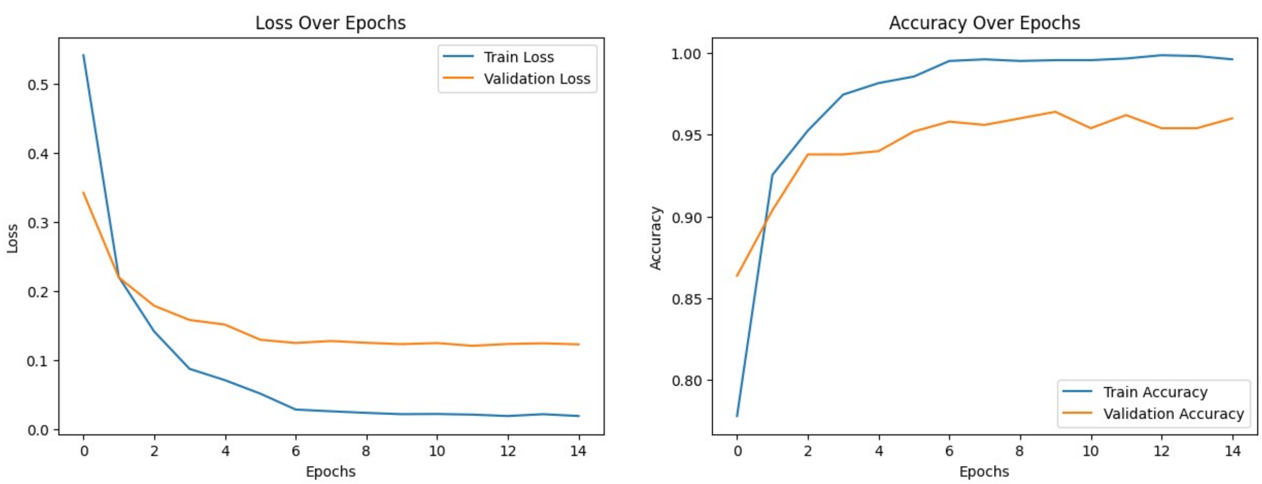

Figure 8.1.2.3 ResNet Traditional Model Performance

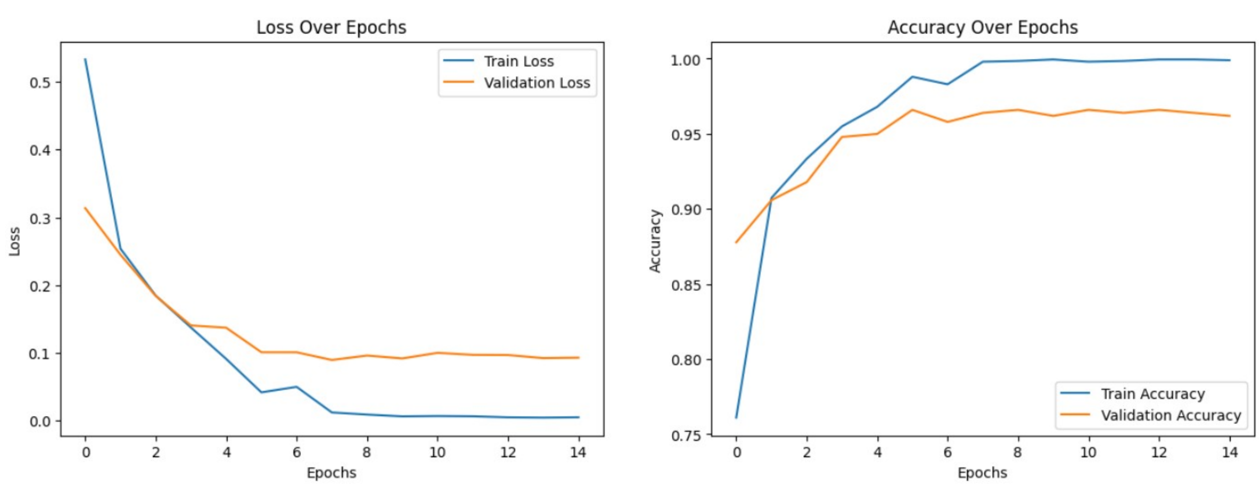

Figure 8.1.2.4 VGG16 Traditional Model Performance

8.1.3 Analysis of Traditional Models

- All models achieved high training accuracy (>99.6%) and validation accuracy (>95.7%)
- MobileNet achieved the highest validation accuracy (96.99%)
- VGG16 demonstrated the lowest training loss (0.0048)
- ResNet and Inception also performed well with validation accuracies of 96.39% and 96.19% respectively

8.2 HYBRID QUANTUM-CLASSICAL MODELS

8.2.1 Performance Metrics Table

Table 8.2.1 Performance Comparison of Hybrid Quantum-Classical Models

Model	Train Acc	Val Acc	Train Loss	Val Loss	Training Time (s)
Inception	98.19%	93.39%	0.1391	0.2382	3350.57
VGG	33.17%	33.27%	1.0987	1.0984	2812.86
MobileNet	99.35%	95.79%	0.0547	0.1575	2270.96
ResNet	99.45%	94.79%	0.0480	0.1682	2581.98

8.2.2 Hybrid Model Performance Visualizations

Figure 8.2.2.1: MobileNet Hybrid Model Performance

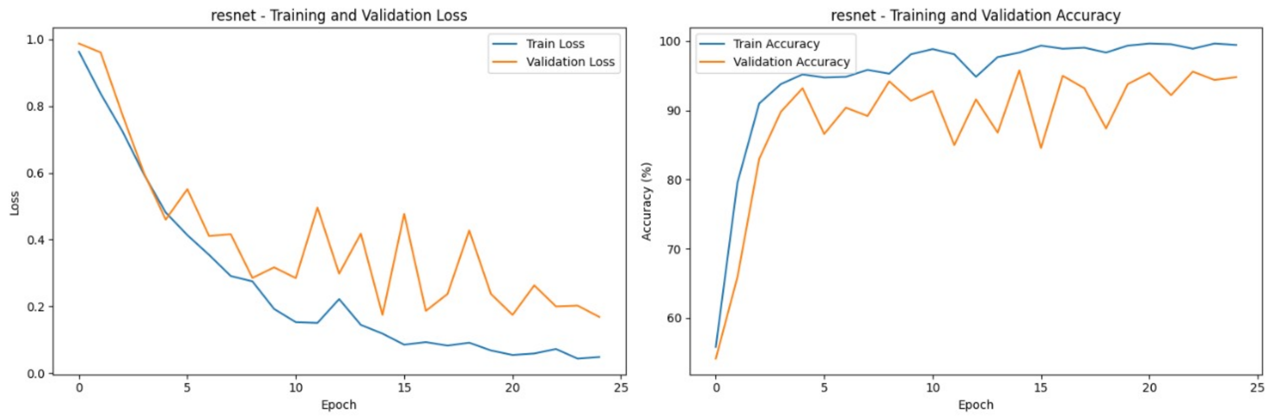

Figure 8.2.2.2: ResNet Hybrid Model Performance

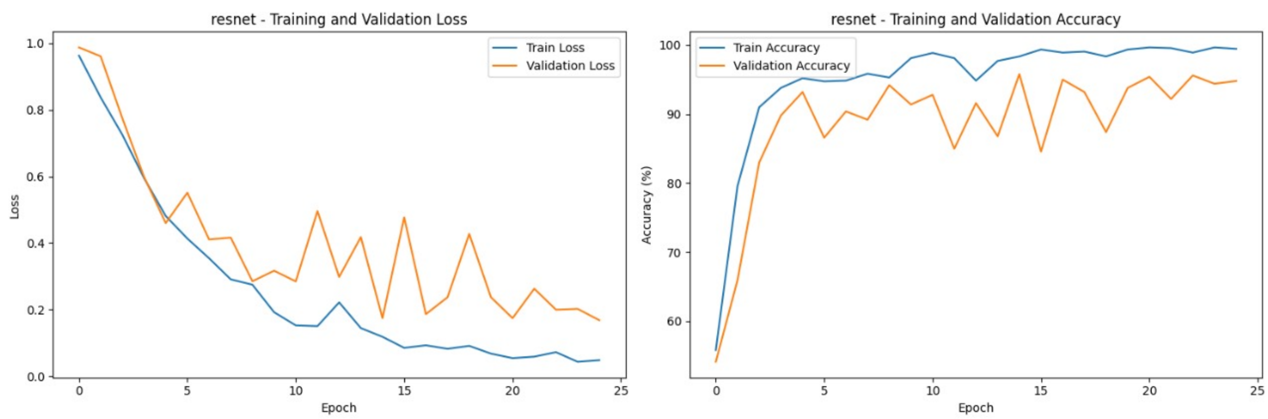

Figure 8.2.2.3: Inception Hybrid Model Performance

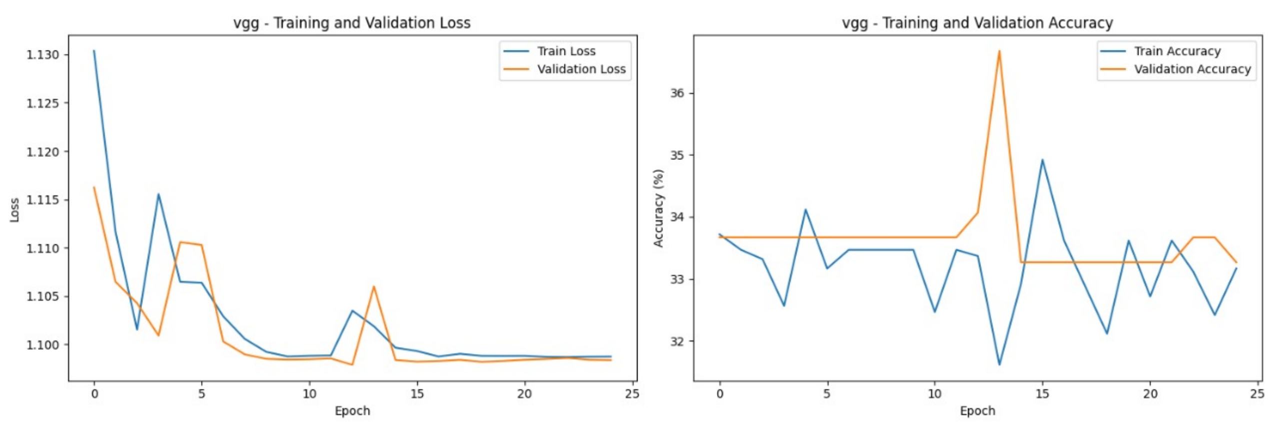

Figure 8.2.2.4: VGG16 Hybrid Model Performance

8.2.3 Analysis of Hybrid Models

- MobileNet hybrid shows the best overall performance with 95.79% validation accuracy

- ResNet hybrid follows closely with 94.79% validation accuracy
- VGG16 hybrid underperformed significantly with only 33.27% validation accuracy
- MobileNet hybrid achieved the fastest training time (2270.96s)
- Inception hybrid had decent performance (93.39%) but required the longest training time

8.2.4 System Output for Given MRI Inputs

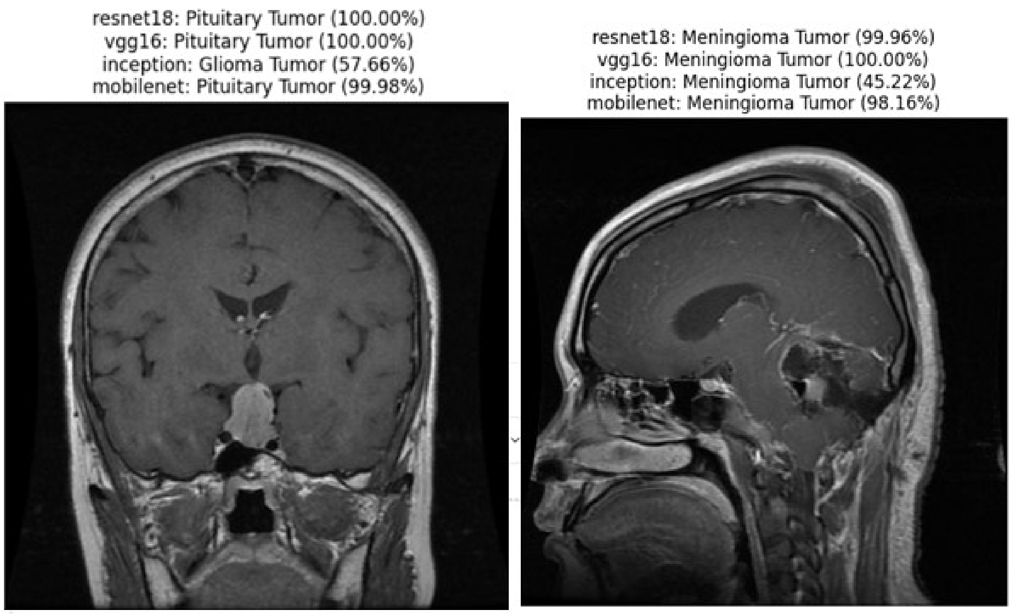

Figure 8.2.4.1Multi-Model Classification Results on Sample MRI Scan

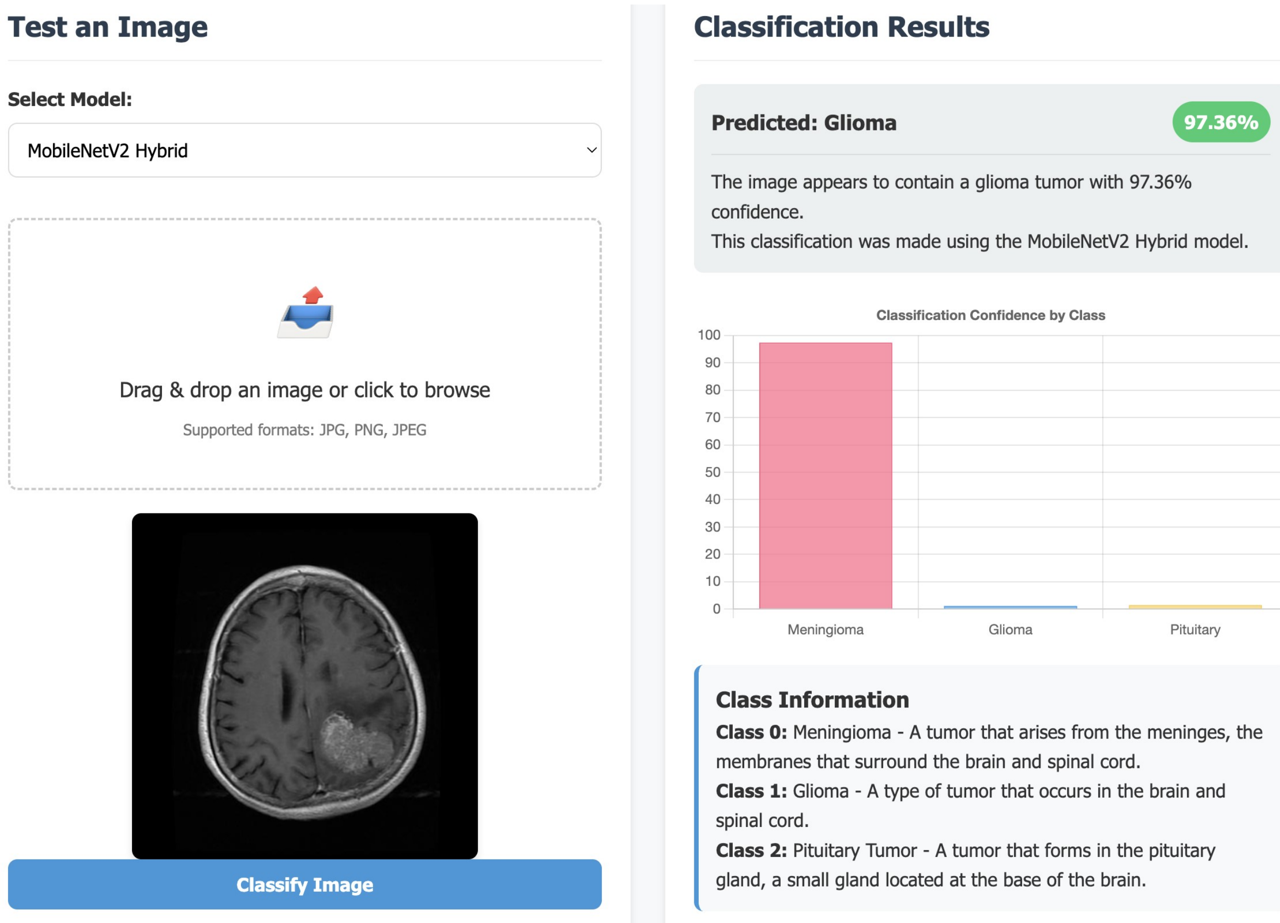

Figure 8.2.4.2. User Interface - Brain Tumour Classification System in Action

8.3 OUTCOMES

8.3.1 Overall Performance Outcomes

- Traditional deep learning models demonstrated excellent baseline performance, with all models achieving validation accuracies above 95.7%.
- MobileNet emerged as the top performer in both traditional (96.99%) and hybrid (95.79%) implementations.
- Quantum integration maintained strong performance in most architectures while potentially enhancing feature representation capabilities.
- VGG16 hybrid model showed significant incompatibility with the quantum circuit design, achieving only 33.27% validation accuracy compared to its traditional counterpart's 96.59%.
- The hybrid approach demonstrated a viable path for quantum enhancement of deep learning models for medical image classification.

8.3.2 Architecture-Specific Outcomes

- **MobileNet**: Demonstrated the best compatibility with quantum integration, maintaining high accuracy (95.79%) while achieving the fastest training time (2270.96s) among hybrid models.
- **ResNet**: Showed strong performance in hybrid implementation (94.79%) with the lowest training loss (0.0480), suggesting effective gradient flow through the combined architecture.
- **Inception**: Maintained acceptable validation accuracy (93.39%) but required substantially longer training time (3350.57s), indicating higher computational demands for the hybrid version.
- **VGG16**: Failed to converge effectively in hybrid form, suggesting fundamental architectural incompatibility with the implemented quantum circuit design.

8.3.3 Technical Outcomes

- Successfully implemented a 4-qubit, 2-layer quantum circuit integrated with classical CNN backbones.
- Developed an effective feature reduction pipeline to bridge high-dimensional CNN outputs with limited qubit processing.
- Identified optimal hyperparameters for quantum-classical integration across different architectures.
- Created a complete web-based system for real-time brain tumour classification using the hybrid models.
- Established a comparative benchmark for quantum-enhanced medical image classification.

8.3.5 Limitations and Challenges

- Dimensional reduction from high-dimensional CNN features to limited qubit space (4 qubits) introduces a potential information bottleneck.
- Not all classical architectures showed compatibility with the quantum circuit design, highlighting the need for architecture-specific quantum integration strategies.
- Quantum circuit depth limitations (2 layers) may restrict the expressivity of the quantum component.
- The gap between training and validation performance was generally wider in hybrid implementations, suggesting potential generalization challenges.

8.3.6 Future Direction Outcomes

- Identified promising avenues for enhancing VGG16 compatibility through modified quantum circuit designs or alternative qubit encoding strategies.
- Established a foundation for exploring deeper quantum circuits with more qubits as quantum hardware capabilities advance.
- Demonstrated the viability of hybrid quantum-classical approaches for medical image analysis, opening pathways for application to other medical imaging domains.
- Created a framework for comparative evaluation that can be extended to assess future quantum-enhanced medical imaging systems.

CHAPTER 09
CONCLUSIONS

9.1 CONCLUSION

This project successfully developed and evaluated a hybrid quantum-classical approach for brain tumour classification, integrating quantum computing principles with established deep learning architectures. Our work demonstrates that quantum computing can be effectively incorporated into medical image analysis tasks with promising results.

The comparative analysis of traditional CNN architectures (ResNet18, VGG16, MobileNet, and InceptionV3) against their quantum-enhanced counterparts revealed several important findings. Traditional models achieved excellent baseline performance with validation accuracies above 95.7%, establishing a strong foundation for comparison. When quantum circuits were integrated into these architectures, performance varied significantly based on architectural compatibility.

MobileNet emerged as the most successful hybrid model, achieving 95.79% validation accuracy and the fastest training time (2270.96s). This performance demonstrates that lightweight architectures with depthwise separable convolutions appear particularly well-suited for quantum integration. ResNet also performed admirably in its hybrid form, reaching 94.79% validation accuracy with the lowest training loss (0.0480), suggesting its residual connections facilitate effective gradient flow in quantum-enhanced settings.

The most significant finding was the contrasting responses of different architectures to quantum integration. While MobileNet and ResNet maintained strong performance, Inception required substantially longer training time, and VGG16 failed to converge effectively (33.27% validation accuracy). This highlights that quantum enhancement is not universally beneficial across all architectures but rather requires careful consideration of compatibility between classical network design and quantum processing capabilities.

From a technical perspective, our implementation successfully bridged the dimensional gap between high-dimensional CNN outputs and the limited qubit processing space through an effective feature reduction pipeline. The 4-qubit, 2-layer quantum circuit with RY/RZ rotations and CNOT entanglement gates demonstrated sufficient expressivity to maintain high classification performance while potentially capturing quantum correlations between features.

The complete system, including the web-based interface for real-time classification, provides a practical demonstration of how hybrid quantum-classical systems can be deployed for medical applications. The ability to classify brain tumour types (Glioma, Meningioma, and Pituitary) with high accuracy using quantum-enhanced networks represents a meaningful step toward the practical application of quantum computing in healthcare.

In conclusion, this project establishes that quantum computing can enhance deep learning approaches for medical image analysis when properly integrated with compatible architectures. The MobileNet hybrid model, with its optimal balance of accuracy and efficiency, demonstrates the most promising path forward for quantum-enhanced brain tumour classification.

9.2 FUTURE WORK

Several promising directions for future research emerge from this project:

- **Expanded Quantum Circuit Complexity**: Future work should explore deeper quantum circuits with more qubits as quantum hardware capabilities advance. This could include investigating how increased circuit depth affects model expressivity and whether additional entanglement patterns might better capture feature correlations.
- **Architecture-Specific Quantum Integration**: Given the varying compatibility observed across architectures, research should develop specialized quantum circuit designs tailored to specific CNN architectures. Particularly, addressing the challenges faced by VGG16 could provide insights into making deep sequential networks more compatible with quantum processing.
- **Larger Dataset Validation**: Extending this approach to larger and more diverse brain tumour datasets would provide more robust validation of the quantum advantage in medical imaging.
- **Extended Medical Applications**: The hybrid approach demonstrated here could be extended to other medical imaging domains such as lung cancer detection, retinal disease classification, or cardiac imaging.
- **Quantum-Enhanced Data Augmentation**: Investigating whether quantum techniques could provide novel approaches to data augmentation for medical imaging where data scarcity is often a challenge.
- **Explainability Methods**: Developing techniques to interpret the decisions made by quantum-enhanced networks would address the critical need for explainability in medical AI applications.

- **Hardware-Specific Optimization**: Adapting the quantum circuit design to specific quantum hardware architectures (superconducting qubits, trapped ions, etc.) could lead to implementations that better leverage the strengths of available quantum processors.

9.3 APPLICATIONS

The hybrid quantum-classical approach for brain tumour classification developed in this project has several potential applications:

- **Clinical Decision Support**: The system could serve as a decision support tool for radiologists and neurosurgeons, providing preliminary classification of brain tumours from MRI scans to assist in diagnosis and treatment planning.
- **Remote Healthcare Services**: The web-based interface enables remote classification capabilities, allowing medical professionals in underserved or rural areas to access advanced tumour classification technology.
- **Research and Education**: The system can be used in medical education to help students learn about brain tumour characteristics and classification, as well as in research settings to process large datasets of MRI images.
- **Screening and Triage**: In high-volume medical centers, the system could help prioritize cases by identifying potentially serious tumour types that require immediate attention.
- **Longitudinal Patient Monitoring**: For patients with known tumours, the system could help track changes over time by consistently classifying tumour types across multiple scans.
- **Telemedicine Integration**: The API-based architecture allows for integration with telemedicine platforms, enabling real-time classification during virtual consultations.
- **Multi-Modal Diagnostic Systems**: The quantum-enhanced classification approach could be integrated with other diagnostic tools to create comprehensive brain tumour assessment systems that combine imaging, genetic, and clinical data.
- **Clinical Trial Support**: Pharmaceutical companies developing brain tumour treatments could utilize the system to help categorize patients for clinical trials.

Appendix A:

Details of sponsorship letter – Unicare Hospital, Pune

Unicare Hospital

Unicare Hospital

Survey No. 45, Milkat No.01/0134, Near Jain Mandir,
Dehu-Alandi Road, Dehugaon, Tal.Haveli, Dist. Pune
Ph. No. 9145739739 & 8007770101

Date: 10th August, 2024.

Date :

To,
Prof. Dr. Sonali Patil,
Head of the Department,
Pimpri Chinchwad College of Engineering, Nigdi, Pune-44.

Subject: Sponsored project offer letter.

Dear Madam,
UNICARE HOSPITAL is pleased to inform you that we have selected you and your team for the consultancy/sponsorship work on **Brain Tumour Classification with Quantum-Augmented Deep Learning Model.**
The following team from your institute were interviewed and selected to work on the project mentioned above. Our organization will provide mentoring and other related support required for the project development.

The team to carry the sponsorship work:

Faculty name:
- Prof. Dr. Asmita Manna

Company/Section Mentor Name:
Dr. Ghangale Abhijeet BalKrishna

Student Names:
- Aditya Shrikant Gorane
- Prathamesh Sahebrao Gole
- Shrikant Ananta Jadhao
- Vishwajeet Sanjay Koshti

This team is allowed to come to the campus for the guidance and project discussions.

We expect timely completion of assignments from the team. The assignments and scope of the work will be planned taking into consideration their college priorities or schedules and the time available to the students for working on the project.

Regards,

Dr. Ghangale Abhijeet Balkrishna
MBBS MS (GEN Surgery) M. Ch. (Neuro Surgery) MBBS

UNICARE HOSPITAL
Survey No. 45, Milkat No. 1/0134,
Near Jain Mandir, Dehu-Alandi Road,
Dehugaon, Tal. Haveli, Dist. Pune.
Reg. No. 615
Phone No. 7775988364

Appendix B:

Plagiarism Report of project report.

GB13_Black_Book (Repaired)

by Vishwajeet

General metrics

85,723	11,089	1042	44 min 21 sec	1 hr 25 min
characters	words	sentences	reading time	speaking time

Score

96

This text scores better than 96% of all texts checked by Grammarly

Writing Issues

221	69	152
Issues left	Critical	Advanced

Plagiarism

6 %

108 sources

6% of your text matches 108 sources on the web or in archives of academic publications

REFERENCES

1. Imam, R., & Alam, M. T. (2023, August). Optimizing Brain Tumour Classification: A Comprehensive Study on Transfer Learning and Imbalance Handling in Deep Learning Models. In International Workshop on Epistemic Uncertainty in Artificial Intelligence (pp. 74-88). Cham: Springer Nature Switzerland.
2. **Karim, P. J., Mahmood, S. R., & Sah, M. (2023).** Brain Tumour Classification using Fine-Tuning based Deep Transfer Learning and Support Vector Machine. *International Journal of Computing and Digital Systems, 13*(1), DOI: 10.12785/ijcds/130108.
3. Shelatkar, T., & Bansal, U. (2022, March). Diagnosis of brain tumour using light weight deep learning model with fine tuning approach. In International Conference on Machine Intelligence and Signal Processing (pp. 105-114). Singapore: Springer Nature Singapore.
4. Paul, S., Ahad, D. M. T., & Hasan, M. M. (2022). Brain cancer segmentation using yolov5 deep neural network. arXiv preprint arXiv:2212.13599.
5. Guan, Y., Aamir, M., Rahman, Z., Ali, A., Abro, W. A., Dayo, Z. A., ... & Hu, Z. (2021). A framework for efficient brain tumour classification using MRI images.
6. Talukder, M. A., Islam, M. M., Uddin, M. A., Akhter, A., Pramanik, M. A. J., Aryal, S., ... & Moni, M. A. (2023). An efficient deep learning model to categorize brain tumour using reconstruction and fine-tuning. Expert systems with applications, 230, 120534.
7. Abd El Kader, I., Xu, G., Shuai, Z., Saminu, S., Javaid, I., & Salim Ahmad, I. (2021). Differential deep convolutional neural network model for brain tumour classification. Brain Sciences, 11(3), 352.
8. Veeramuthu, A., Meenakshi, S., Mathivanan, G., Kotecha, K., Saini, J. R., Vijayakumar, V., & Subramaniyaswamy, V. (2022). MRI brain tumour image classification using a combined feature and image-based classifier. Frontiers in Psychology, 13, 848784.
9. Aamir, M., Rahman, Z., Dayo, Z. A., Abro, W. A., Uddin, M. I., Khan, I., ... & Hu, Z. (2022). A deep learning approach for brain tumour classification using MRI images. Computers and Electrical Engineering, 101, 108105.
10. Ayadi, W., Elhamzi, W., Charfi, I., & Atri, M. (2021). Deep CNN for brain tumour classification. Neural processing letters, 53, 671-700.
11. Badža, M. M., & Barjaktarović, M. Č. (2020). Classification of brain tumours from MRI images using a convolutional neural network. Applied Sciences, 10(6), 1999.
12. Irmak, E. (2021). Multi-classification of brain tumour MRI images using deep convolutional neural network with fully optimized framework. Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 45(3), 1015-1036
13. Papadomanolakis TN, Sergaki ES, Polydorou AA, Krasoudakis AG, Makris-Tsalikis GN, Polydorou AA, Afentakis NM, Athanasiou SA, Vardiambasis IO, Zervakis ME. Tumour Diagnosis against Other Brain Diseases Using T2 MRI Brain Images and CNN Binary Classifier and DWT. Brain Sci. 2023 Feb 17;13(2):348. doi: 10.3390/brainsci13020348. PMID: 36831891; PMCID: PMC9954603.

14. Zhuge Y, Ning H, Mathen P, Cheng JY, Krauze AV, Camphausen K, Miller RW. Automated glioma grading on conventional MRI images using deep convolutional neural networks. Med Phys. 2020 Jul;47(7):3044-3053. doi: 10.1002/mp.14168. Epub 2020 May 11. Erratum in: Med Phys. 2023 Sep;50(9):5930-5931. doi: 10.1002/mp.16588. PMID: 32277478; PMCID: PMC8494136.
15. Alongi P, Arnone A, Vultaggio V, Fraternali A, Versari A, Casali C, Arnone G, DiMeco F, Vetrano IG. Artificial Intelligence Analysis Using MRI and PET Imaging in Gliomas: A Narrative Review. Cancers (Basel). 2024 Jan 18;16(2):407. doi: 10.3390/cancers16020407. PMID: 38254896; PMCID: PMC10814838.
16. Rohini A, Praveen C, Mathivanan SK, Muthukumaran V, Mallik S, Alqahtani MS, Al-Rasheed A, Soufiene BO. Multimodal hybrid convolutional neural network-based brain tumour grade classification. BMC Bioinformatics. 2023 Oct 10;24(1):382. doi: 10.1186/s12859-023-05518-3. PMID: 37817066; PMCID: PMC10566188.
17. Abd El Kader, I.; Xu, G.; Shuai, Z.; Saminu, S.; Javaid, I.; Salim Ahmad, I. Differential Deep Convolutional Neural Network Model for Brain Tumour Classification. *Brain Sci.* **2021**, *11*, 352. https://doi.org/10.3390/brainsci11030352
18. Nayak, D.R.; Padhy, N.; Mallick, P.K.; Zymbler, M.; Kumar, S. Brain Tumour Classification Using Dense Efficient-Net. *Axioms* **2022**, *11*, 34. https://doi.org/10.3390/axioms11010034
19. Kaifi R. A Review of Recent Advances in Brain Tumour Diagnosis Based on AI-Based Classification. Diagnostics (Basel). 2023 Sep 20;13(18):3007. doi: 10.3390/diagnostics13183007. PMID: 37761373; PMCID: PMC10527911.
20. Alqudah, Ali & Alquran, Hiam & Abuqasmieh, Isam & Alqudah, Amin & Sharo, Wafaa. (2020). Brain Tumour Classification Using Deep Learning Technique -- A Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes. 10.48550/arXiv.2001.08844.
21. ZainEldin H, Gamel SA, El-Kenawy EM, Alharbi AH, Khafaga DS, Ibrahim A, Talaat FM. Brain Tumour Detection and Classification Using Deep Learning and Sine-Cosine Fitness Grey Wolf Optimization. Bioengineering (Basel). 2022 Dec 22;10(1):18. doi: 10.3390/bioengineering10010018. PMID: 36671591; PMCID: PMC9854739.