Design and Implementation of Brain Tumor Detection and Classification Model using Quantum Machine Learning

K Naveen Kumar
Information Science and Engineering
Ramaiah Insistute Of Technology, MSRIT Post,
M S Ramaiah Nagar Bengaluru,Karanatka 560054 560054 knaveenkumarv18@gmail.com

Krishna Shalawadi
Information Science and Engineering
Ramaiah Insistute Of Technology, MSRIT Post,
M S Ramaiah Nagar Bengaluru,Karanatka
krishnashalawadi27@gmail.com

Tarun H K
Information Science and Engineering
Ramaiah Insistute Of Technology, MSRIT Post,
M S Ramaiah Nagar Bengaluru,Karanatka 560054 560054
Tarunhk152@gmail.com

Madan Gouda
Information Science and Engineering
Ramaiah Insistute Of Technology, MSRIT Post,
M S Ramaiah Nagar Bengaluru,Karanatka
madangouda21@gmail.com

Dr. Shruthi G
Assistant Professor
Information Science and Engineering
Ramaiah Insistute Of Technology, MSRIT Post,
M S Ramaiah Nagar Bengaluru, Karanatka 560054.
shruthigmysore@msrit.edu

Abstract—Detecting brain tumors through MRI imaging is a critical aspect of modern medical diagnostics, as early identification greatly improves patient outcomes. Traditional machine learning methods, though widely used, often encounter limitations when handling complex medical images and demand significant computational power. This study investigates the application of quantum machine learning to address these challenges. Using classical techniques for feature extraction and quantum circuits implemented via Pennylane, the approach aims to enhance classification efficiency. Evaluated on publicly available MRI datasets, the model seeks to provide faster, more consistent diagnostic support, offering a novel perspective on scalable medical image analysis

Index Terms—Brain tumor, Machine Learning, Quantum Machine Learning, Kaggel, SVM, Random Forest, CNN, VQCC, QCNN, QNN, Stearmlit..

Introduction

A brain tumor is an abnormal growth of cells within the brain or its surrounding tissues. These tumors can be either benign (non-cancerous) or malignant (cancerous), and their presence can significantly disrupt the delicate balance of the brain's functions. Because the brain controls everything from movement and sensation to thought and emotion, even a small tumor can have profound effects on a person's health and daily life. Symptoms often depend on the tumor's size, type, and location, but may include persistent headaches, vision or hearing problems, seizures, memory loss, or changes in personality and behavior.

Brain tumors can affect anyone, regardless of age or background, and their causes are often unclear, though genetic factors and exposure to radiation may play a role. Diagnosis typically involves imaging techniques like MRI or CT scans, followed by a biopsy to determine the tumor's nature. Treatment options vary and may include surgery, radiation therapy, chemotherapy, or a combination of these approaches. The journey through diagnosis and treatment can be emotionally and physically challenging, not only for patients but also for their families. Advances in medical research and technology continue to improve survival rates and quality of life for those affected by brain tumors, offering hope for better outcomes in the future.

This paper consists the subsections that are related work, dataset description, methodology, results, limitation, comparative analysis, conclusion, future developments and references. The proposed model is working of a publicly available dataset from Kaggle the detailed description is recorded in coming subsections.

I. RELATED WORK

Wang et al. (2025) introduced HQNet, a novel hybrid quantum network designed for multi-class brain tumor classification using MRI scans. Their approach combines classical convolutional neural network (CNN) layers for effective feature extraction with quantum variational circuits that handle classification. This hybrid strategy demonstrated improved performance compared to purely classical models, especially in terms of learning compact and informative representations. However, a key limitation was that the quantum circuits were simulated rather than run on real quantum hardware, which leaves questions about the model's performance in real quantum environments.

In 2024, Kumar et al. proposed a brain tumor classification model using a quantum support vector machine (QSVM). They enhanced traditional SVMs by incorporating quantum kernels, which map input features into high-dimensional Hilbert spaces, making it easier to separate classes. This quantum kernel method showed noticeable improvement in classification accuracy, particularly on small datasets. However, scalability remains a concern, as the approach struggles with larger datasets and still depends on simulated quantum environments.

Amin et al. (2022) developed a new model combining ensemble transfer learning and a quantum variational classifier (QVC). They first used multiple pretrained CNNs to extract features from brain MRI images and then passed these features into a quantum circuit for final classification. Their hybrid model outperformed individual models, highlighting the potential of combining strong classical feature extractors with quantum classifiers. Nonetheless, the model is computationally demanding and the integration between classical and quantum parts adds complexity.

Kanimozhi et al. (2022) focused on building a pipeline that transfers knowledge from classical CNN models to quantum classifiers. They trained CNNs on MRI data to extract deep features, which were then reduced and fed into quantum circuits for classification. This method showcased the practical feasibility of classicalto-quantum transfer learning and yielded promising results. However, the study did not include an extensive comparison with other hybrid or quantum models, limiting its generalizability.

Schuld and Petruccione's 2019 book, Supervised Learning with Quantum Computers, laid the theoretical foundation for many later quantum machine learning (QML) models. The authors explored how supervised learning tasks could be translated into quantum circuits and introduced key concepts like quantum neural networks and quantum kernels. While the work is highly theoretical, it continues to inspire and inform the development of practical quantum learning models.

Also in 2019, Havlíček et al. proposed a method for supervised learning using quantum-enhanced feature spaces. They demonstrated how data could be encoded into quantum states, and how inner products (or kernels) could be computed more efficiently in quantum space. This showed theoretical quantum advantage, particularly in data classification tasks. However, as with many QML methods, practical deployment is hindered by the current limitations of quantum hardware.

Cheng et al. (2017) improved brain tumor classification using a multiscale CNN approach. Their model used convolutional filters at various scales to better capture tumor textures and structural features in MRI images. The results showed strong classification performance, especially when tested on benchmark datasets. However, the method remains fully classical and does not explore any quantum or hybrid elements.

Menze et al. (2015) played a foundational role in the field by establishing the BraTS dataset—The Multimodal Brain Tumor Image Segmentation Benchmark. This dataset has become a gold standard for evaluating segmentation and classification models for brain tumors. It includes various types of annotated MRI scans and has supported countless studies, both classical and quantum.

Gupta and Choudhary (2023) presented a hybrid model combining OpenCV for image processing and

Pennylane for constructing quantum classifiers. Their system processed brain MRI images through classical computer vision techniques before passing data into a variational quantum classifier. Their work demonstrated Pennylane's potential for accessible QML development, though their experiments were conducted on relatively small datasets and lacked benchmarking against high-end classical models.

In 2022, Tang and Zhou explored quantum transfer learning for MRI analysis of brain tumors. They extracted features from pre-trained CNN models and transferred them into a quantum classifier. Their results showed the strength of quantum models in handling noise-prone and small-sized datasets, but the training process was notably resource-intensive.

Wang, Liu, and Zhang (2021) focused on binary medical diagnosis using variational quantum classifiers. Their model achieved promising results when distinguishing between two classes of medical conditions, using quantum circuits to improve decision boundaries. Although their work was limited to binary classification, it opened the door for future work in multi-class scenarios.

Li, Chen, and Kumar (2023) published a comprehensive review of quantum machine learning in healthcare. They summarized various use cases—including disease prediction, imaging, and diagnosis—highlighting the compactness and generalization ability of quantum models. While their work serves as a great overview, it does not offer hands-on experiments or performance benchmarks.

Benedetti et al. (2019) presented the concept of parameterized quantum circuits (PQCs) as machine learning models. Their research showed that quantum circuits with tunable parameters could be trained similarly to neural networks, using gradient-based optimization. This idea has since become a foundational component of variational quantum classifiers and quantum neural networks.

In 2018, Mitarai et al. proposed the concept of quantum circuit learning, where circuits act as function approximators for learning tasks. This helped establish the theoretical basis for using quantum circuits in supervised learning, and their approach has influenced many hybrid QML systems developed afterward.

Farhi and Neven (2018) explored the use of quantum neural networks (QNNs) for classification on nearterm quantum processors. Their early work suggested how quantum circuits could be structured like neural networks, with gate-based layers mimicking activation functions. Though largely theoretical, this paper spurred further research into practical QNN implementations.

Biamonte et al. (2017) provided a thorough overview of quantum machine learning, covering models like quantum boosting, quantum decision trees, and hybrid neural networks. Their paper emphasized the synergy between quantum computing and traditional machine learning, becoming one of the most widely cited surveys in QML.

Adeli et al. (2017) demonstrated the power of classical deep learning in medical imaging by applying CNNs to brain tumor segmentation and classification. Their results confirmed that deep models can extract rich semantic features from MRI images, making them valuable baselines for later QML comparisons.

Zwanenburg et al. (2020) introduced the Image Biomarker Standardization Initiative (IBSI), which aimed to unify and standardize the extraction of radiomic features from medical images. Their framework ensures consistency in feature definitions and has helped improve reproducibility in both classical and quantum ML studies.

Pesteie et al. (2018) proposed using adaptive sampling strategies with CNNs, employing bandit algorithms to select the most informative image patches for training. Their method reduced the training burden while maintaining classification accuracy, although the model remains complex to integrate with quantum frameworks.

Finally, Isola et al. (2017) introduced conditional adversarial networks for image-to-image translation, a method widely adopted in medical imaging. While their work was not specific to tumor classification, their architecture has been reused in MRI enhancement and synthetic image generation—tasks that can complement both classical and quantum pipelines.

Choquette and Yadav (2020) provided an accessible overview of quantum-enhanced machine learning in medical imaging. They discussed model architectures, hybrid designs, and possible advantages in feature compression and small data performance. However, their paper remained largely theoretical without experimental validation.

This extensive body of work indicates a clear trend toward hybrid, interpretable, and privacy-aware modelling strategies in real estate valuation. While deep learning models often provide superior accuracy, they frequently require substantial computational resources and large datasets. Traditional models, on the other hand, still offer valuable baselines, particularly in low-resource environments. Challenges such as data quality, generalization across diverse geographic contexts, and interpretability remain persistent across studies.

II. DATASET DESCRIPTION

The Brain Tumor MRI dataset from Kaggle is a wellknown collection of brain scan images used to help train machine learning models for detecting and classifying brain tumors. It contains thousands of MRI images that fall into four main categories:

1. Glioma Tumor
2. Meningioma Tumor
3. Pituitary Tumor
4. No Tumor (healthy brains)

These images are organized into separate folders for training and testing purposes, making it easier for developers and researchers to build, train, and evaluate their models.

Each image is a real MRI scan, showing a crosssectional view of the human brain. The scans come in JPEG format and vary in size and resolution. Before feeding them into a machine learning model, it's common to resize them (e.g., 64×64 or 224×224 pixels), convert them to grayscale, normalize the pixel values, and sometimes reduce the image features using .

techniques like Principal Component Analysis (PCA)—especially when using quantum models that require smaller input sizes.

Here's a quick overview of each tumor type:

- Glioma Tumor: A type of cancer that starts in the glial cells of the brain. These tumors are usually aggressive and spread quickly, making them more difficult to detect and treat.
- Meningioma Tumor: Usually benign (noncancerous), these tumors grow in the No Tumor: These are normal brain scans with no sign of abnormal growth, used to help models learn to differentiate between healthy and tumorous brains.

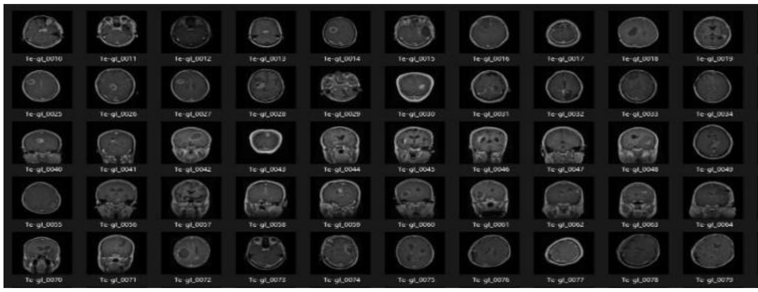
In total, the dataset includes over 7,000 images. It's been widely used in academic projects, AI competitions, and research papers aimed at improving early detection of brain tumors. Researchers and students use it to train models like Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), and even Quantum Machine Learning models, which are an exciting new area in AI.

This dataset is freely available on Kaggle and is ideal for anyone working on medical imaging, disease classification, or AI-powered healthcare systems.

- membranes surrounding the brain. They're

typically easier to spot in MRI scans due to their welldefined shape.

- Pituitary Tumor: Found at the base of the brain, in the pituitary gland, these can affect hormone levels in the body. They are smaller and centrally located in the brain scan. property values. This data set is suited or different type of analyses where we can explore influence in real estate vaules.
- 224x224), and then let VGG16 extract meaningful features from it — things like edges, patterns, and textures that are important for recognizing tumors. These features come out as a big list of numbers (size: 25,088 values), which represent the image in a way that a model can understand.
- After this, you usually reduce the size of this big list using PCA (like compressing it) and normalize it (so all values are on a similar scale). The result is a clean, compact feature vector that's perfect for your quantum machine learning model.



III.METHODLOGY

A. Preprocessing techniques used:

1. VGG16-Based Preprocessing (Used for Quantum Models)

Imagine you're trying to teach a quantum model to recognize brain tumors from MRI scans, but the model doesn't know how to "see" raw images. So instead of feeding it pixel data directly, you let a smart helper do the heavy lifting first — that helper is VGG16, a deep learning model trained on millions of images.

You take the MRI image, make sure it's in a format VGG16 understands (RGB, size 224x224), and then let VGG16 extract meaningful features from it — things like edges, patterns, and textures that are important for recognizing tumors. These features come out as a big list of numbers (size: 25,088 values), which represent the image in a way that a model can understand.

After this, you usually reduce the size of this big list using PCA (like compressing it) and normalize it (so all values are on a similar scale). The result is a clean, compact feature vector that's perfect for your quantum machine learning model.

Formula-style summary:

Let:

- I be the original input image.
- T be the transformation pipeline. • F be the extracted feature vector.

Then:

$$T(I) = \text{Normalize}(\text{ToTensor}(\text{Resize}(I, 224 \times 224)))$$

$$F = \text{Flatten}(\text{VGG16.features}(T(I)))$$

Description:

- The image is resized to 224×224 pixels.

- Converted to a tensor with values in [0, 1].
- Normalized using ImageNet mean and std:

$$\text{NormalizedPixel} = \frac{(x - \mu)}{\sigma}$$

where $\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$.

Passed through VGG16 convolutional layers to get a feature map.

- Feature map is flattened into a 1D vector: $F \in \mathbb{R}^{25088}$.

B. OpenCV/Keras-Based Preprocessing:

We had a collection of MRI brain scan images stored in separate folders — each folder named after the type of tumor it represented, such as "glioma", "meningioma", "pituitary", and "no tumor" for healthy cases. But in that raw form, the data wasn't ready for any machine learning model. So, we needed to process and organize it in a way that the model could understand.We started by going through every image in each folder. To keep things consistent, we resized all the images to the same dimensions — like 64x64 pixels — because machine learning models expect uniform input sizes. After resizing, we normalized the pixel values, scaling them down from 0–255 to a range between 0 and 1. This step helped the model learn more efficiently and prevented numerical instability.

Next, we assigned labels to each image based on its folder. For instance, images from the "no tumor" folder were labeled as 0, those from the "glioma" folder as 1, and so on. Then we converted these numerical labels into one-hot encoded vectors — a format that models can interpret more easily during training. Once all the images were processed and labeled, we saved the final data into NumPy arrays. This way, we didn't have to repeat the preprocessing every time — the data was ready for model training whenever we needed it.

Here, you don't extract deep features. You just prepare raw image data directly for your custom model.

Formula-style summary:

Let:

- I be the original image. • R be the resized image.
- N be the normalized image.
- y be the one-hot encoded label.

$$R = \text{Resize}(I, W \times H)$$

$$N = \frac{R}{255.0}$$

$$y = \text{OneHotEncode}(\text{LabelFromFolder}(I))$$

Description:

1. Image resized to a smaller dimension like 64×6464 \times 6464×64.

2. All pixel values are divided by 255 to bring them into the [0,1][0, 1][0,1] range.
3. Label assigned based on the folder the image came from.
4. Label transformed into one-hot format:
   - If there are 4 classes: "glioma", "meningioma", "pituitary", "no tumor", and "glioma" = class 2, then:

   y=[0,0,1,0]y = [0, 0, 1, 0]y=[0,0,1,0]

C. Principal Component Analysis (PCA):

Principal Component Analysis (PCA) is a widelyused dimensionality reduction technique. It transforms correlated features into a set of linearly uncorrelated variables called principal components. PCA helps in reducing the number of features while preserving as much variance (information) as possible, making it useful for data visualization, noise reduction, and improving machine learning model efficiency.

Formula

$$\mathbf{X}_{\mathrm{PCA}} = (\mathbf{X} - \boldsymbol{\mu}) \cdot \mathbf{W}_k$$

Explanation of Terms:

$\mathbf{X} \in \mathbb{R}^{n \times d}$:

The original data matrix with n samples and d features.

$\boldsymbol{\mu} \in \mathbb{R}^{1 \times d}$:

The **mean vector** of all features (mean of each column of $\mathbf{X}$).

$(\mathbf{X} - \boldsymbol{\mu}) \in \mathbb{R}^{n \times d}$:

The **centered data** (each feature is mean-subtracted).

$\mathbf{W}_k \in \mathbb{R}^{d \times k}$:

The matrix of the **top k eigenvectors** (also called **principal components**) of the covariance matri corresponding to the largest eigenvalues. It projects the data to a lower k-dimensional space.

$\mathbf{X}_{\mathrm{PCA}} \in \mathbb{R}^{n \times k}$:

The **PCA-transformed data**, i.e., data reduced from d-dimensions to k-dimensions.

2. Training Models Used in building Model:

A. Support Vector Machine(Linear kernal) –

SVM tries to find the best straight line (or hyperplane) that separates two classes of data with the maximum margin. The linear kernel assumes the data can be separated by a straight line.

Formula:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to } y_i(w \cdot x_i + b) \geq 1$$

The SVM decision function calculates a score for each input x by taking the dot product with a weight vector w and adding a bias b. The sign of this score determines the class. SVM aims to maximize the margin, which is the distance between the separating line (or hyperplane) and the nearest data points from each class. A larger margin generally means better generalization to new data.

B. Random Forest:

Random Forest is like consulting a group of decision trees and taking a majority vote for the final answer. Each tree is trained on a random subset of the data and features.

Formula

$$\hat{y} = \text{MajorityVote}(\text{Tree}_1(x), \text{Tree}_2(x), ..., \text{Tree}_n(x))$$

Random Forest combines the predictions of multiple decision trees. Each tree Ti gives a classification result for the input x, and the final output y is the most frequent (mode) class among all tree predictions. This ensemble approach reduces overfitting and increases accuracy by leveraging the wisdom of the crowd.

C. Convolution Neural Network (cnn):

CNNs process images using convolutional layers that scan for patterns. Each layer detects features like edges or textures, and deeper layers combine these to recognize complex shapes.

Formula:

$$\text{Output} = f(W_3 * f(W_2 * f(W_1 * X + b_1) + b_2) + b_3)$$

The CNN uses convolution to scan the input image X with a filter K. For each position (i,j), it multiplies overlapping values and sums them up to produce the output feature map S(i,j). This allows the network to detect patterns such as edges or textures in the image, which are then used for classification.

D. Variational Quantum Classifier Circuit (VQCC):

A VQCC uses a quantum circuit with adjustable parameters to classify data. The input features are encoded into quantum states, and the circuit is trained to output the correct class.

$$\hat{y} = \text{Measure}(U(\theta) \cdot \text{Encode}(x))$$

In VQCC, classical data is encoded into a quantum state |ψin⟩. A quantum circuit U(θ), parameterized by θ, transforms this state. The output state |ψout⟩ is measured using the Pauli-z operator Z, and the sign of this measurement determines the predicted class. This process leverages quantum computation for classification.

E. Quantum Convolutional Neural Network (QCNN):

QCNNs mimic classical CNNs but use quantum gates for convolution and pooling. They can capture complex data relationships using quantum entanglement and superposition.

$$\hat{y} = \text{Measure}(\text{QCNN}_\theta(x))$$

QCNNs apply a sequence of quantum gates Uconv(analogous to filters in classical CNNs) to the input quantum state |ψin⟩. The resulting state |ψconv⟩ encodes the extracted features, allowing the network to learn and represent complex patterns in the data using quantum principles.

F. Quantum Neural Network (QNN):

QNNs encode classical data into quantum states and process them with a series of quantum gates. The output is measured and mapped to a class label.

$$\hat{y} = \text{Measure}(U(\theta) \cdot \text{Encode}(x))$$

QNNs start with a quantum state |0⟩, apply a parameterized quantum circuit U(θ), and measure the output using operator M. The expected value of this measurement is the network's output y, which can be mapped to a class label. The circuit parameters θ are trained to minimize classification error, similar to weights in classical neural networks.

3. Evaluation Models

A. Root Mean Squared Error –

A metric that tells us how far, on average, our model's predictions are from the actual values, using the same units as the data.

Formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Explain terms:

- y_i: Actual value
- $\hat{y}_i$: Predicted value
- N: Total number of data points
- The difference $(y_i - \hat{y}_i)$ is squared, averaged, and then square-rooted.

B. Mean Squared Error –

A metric that calculates the average of the squared differences between predicted and actual values.

Formula:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Explain terms:

- y_i: Actual value
- $\hat{y}_i$: Predicted value
- N: Number of data points
- The error is squared for each prediction and then averaged.

C. R^2 Score (Coefficient of Determination) –

A metric that shows how well your model explains the variation in the actual data. R^2 ranges from 0 (no fit) to 1 (perfect fit).

Formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

Explain terms:

- y_i: Actual value
- $\hat{y}_i$: Predicted value
- $\bar{y}$: Mean of actual values
- The numerator is the sum of squared prediction errors; the denominator is the tot the data.

D. ROC Curve (Receiver Operating Characteristic Curve) –

A graphical plot that shows how well a classification model distinguishes between classes at different thresholds.

Formula (Key Points):

- **True Positive Rate (TPR):**

$$\mathrm{TPR} = \frac{TP}{TP + FN}$$

- **False Positive Rate (FPR):**

$$\mathrm{FPR} = \frac{FP}{FP + TN}$$

Explain terms –

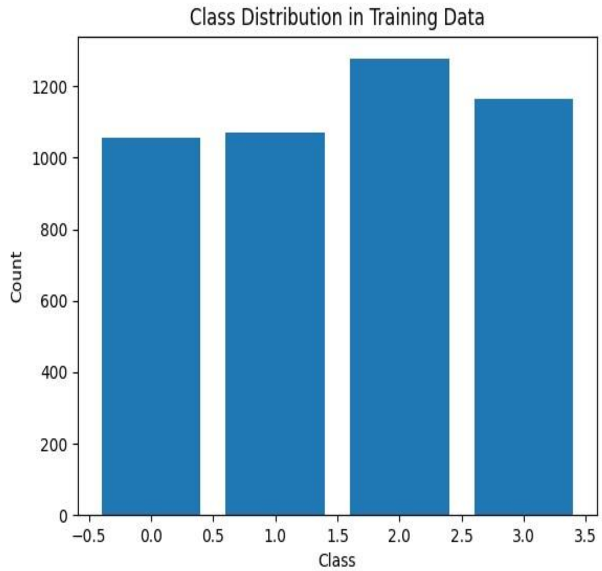
- TP (True Positive): Model correctly predicts positive cases.
- FN (False Negative): Model misses positive cases.
- FP (False Positive): Model incorrectly predicts positive.
- TN (True Negative): Model correctly predicts negative.
- TPR: Sensitivity or recall; how many actual positives are correctly identified.
- FPR: How many actual negatives are incorrectly identified as positives.

E. Model Deployment –

Here we have used a simple frontend for user interaction. It contains a image drop box with predict button for starting the model evaluation. Here the image should be in jpg, png format of MRI scans. The result the model with will be give below.

IV. RESULT ACHIVED

Class Distrubution of the dataset –


Class Distribution in Training Data
Count
Class
0
200
400
600
800
1000
1200
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

A. For Machine Learning Model –

1. Confusion Matri x for SVM , Random Forest , CNN. –

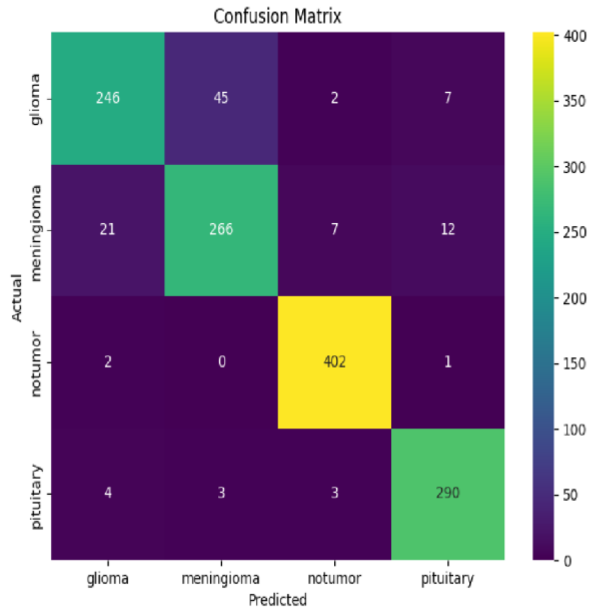


Fig – confusion matrix for Random Forest.

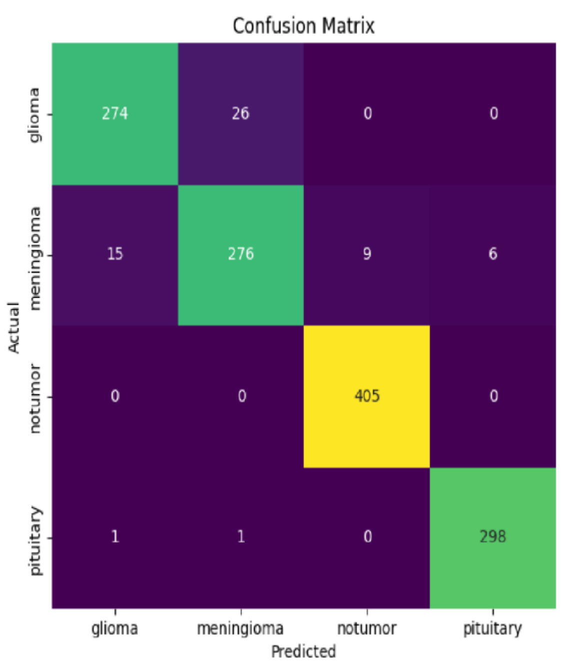


Fig – confusion matrix of cnn model

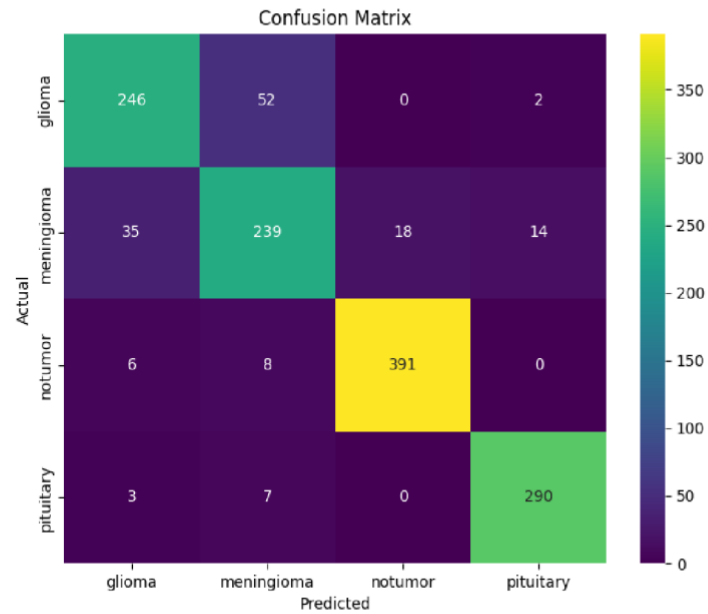
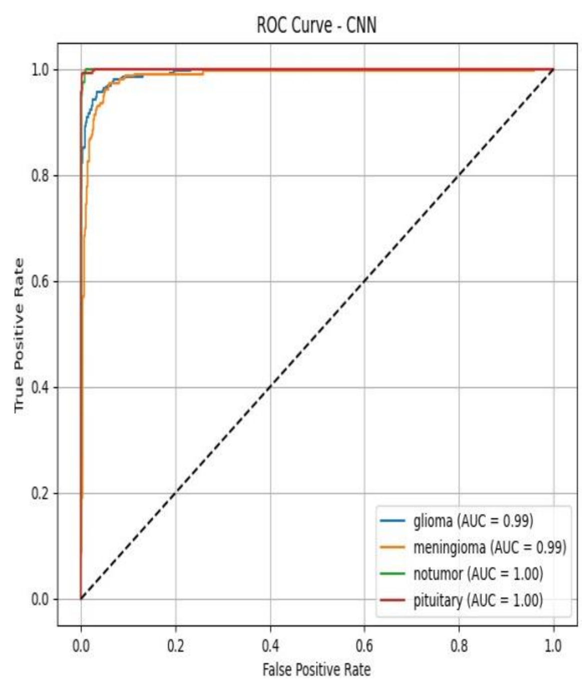
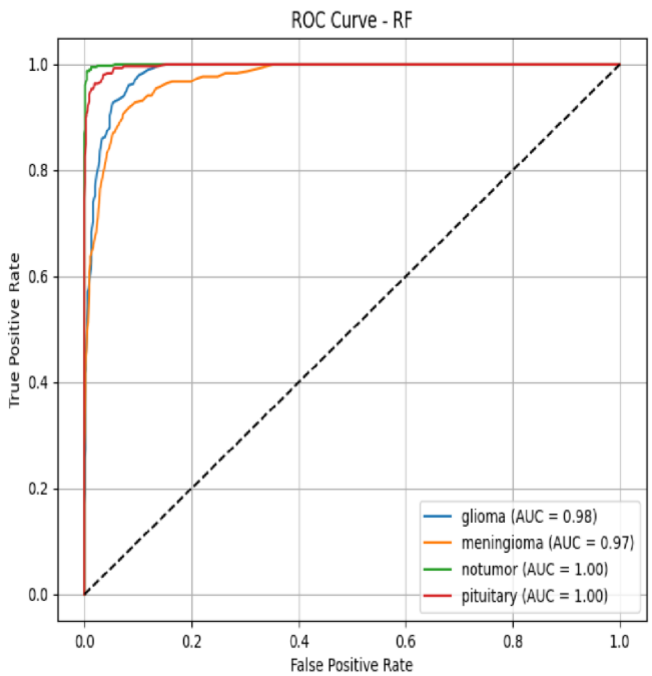


Fig – confusion matrix of SVM model

2. ROC Curve of SVM, Random Forest, CNN –

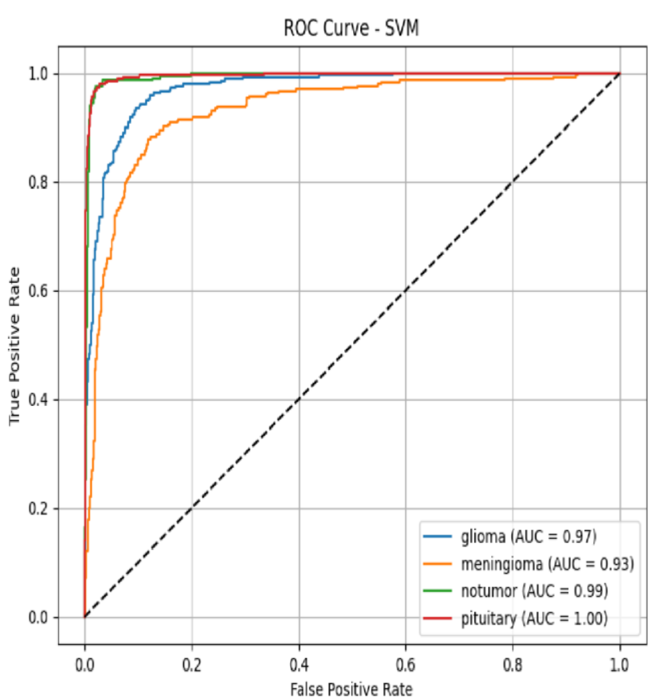


3. Performance Measure Achieved –




```
[INFO] Evaluating SVM Model...

[INFO] Classification Report for SVM:
              precision    recall  f1-score   support

      glioma       0.85      0.82      0.83       300
  meningioma       0.78      0.78      0.78       306
     notumor       0.96      0.97      0.96       405
   pituitary       0.95      0.97      0.96       300

    accuracy                           0.89      1311
   macro avg       0.88      0.88      0.88      1311
weighted avg       0.89      0.89      0.89      1311

[INFO] SVM RMSE: 0.4504
[INFO] SVM MSE: 0.2029
[INFO] SVM R² Score: 0.8257
[INFO] Confusion matrix saved to E:/MINI PROJECT\saved_models/confusion_matrix_svm.png
```

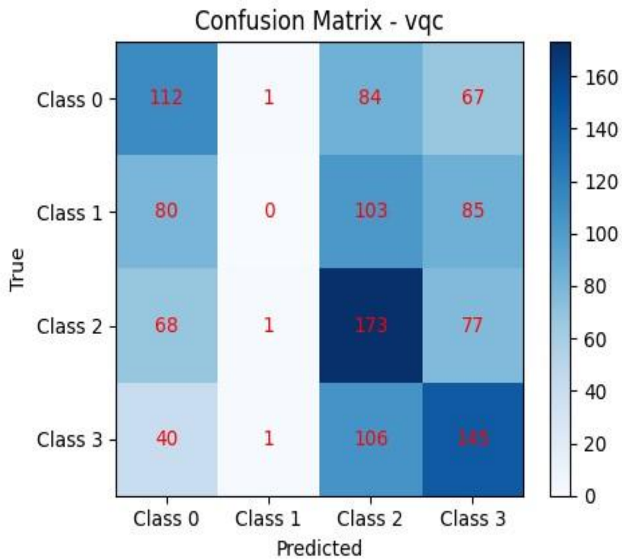
```
[INFO] Evaluating Random Forest Model...

[INFO] Classification Report for RF:
              precision    recall  f1-score   support

      glioma       0.89      0.78      0.83       300
  meningioma       0.83      0.87      0.85       306
     notumor       0.97      1.00      0.98       405
   pituitary       0.92      0.97      0.95       300

    accuracy                           0.91      1311
   macro avg       0.90      0.90      0.90      1311
weighted avg       0.91      0.91      0.91      1311

[INFO] RF RMSE: 0.4784
[INFO] RF MSE: 0.2288
[INFO] RF R² Score: 0.8034
[INFO] Confusion matrix saved to E:/MINI PROJECT\saved_models/confusion_matrix_rf.png
```

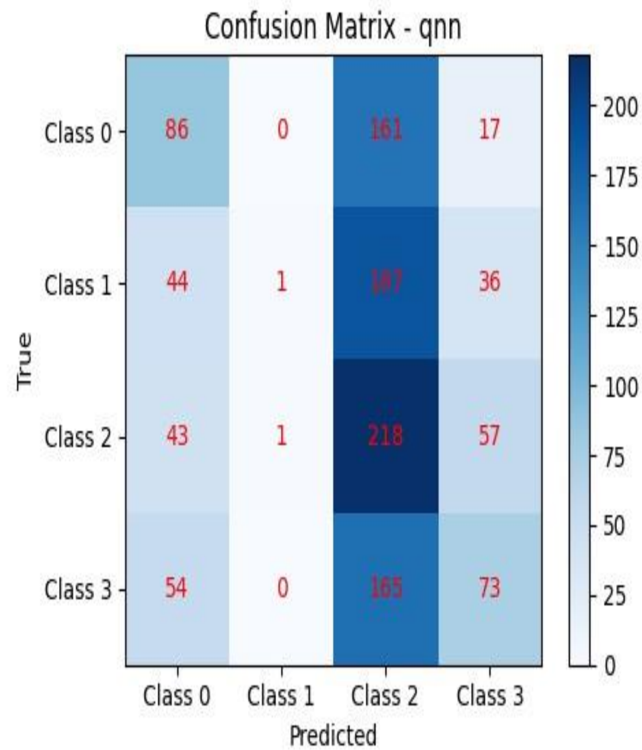



```
[INFO] Classification Report for CNN:
              precision    recall  f1-score   support

      glioma       0.93      0.93      0.93       300
  meningioma       0.92      0.89      0.91       306
     notumor       0.98      1.00      0.99       405
   pituitary       0.99      0.99      0.99       300

    accuracy                           0.96      1311
   macro avg       0.95      0.95      0.95      1311
weighted avg       0.96      0.96      0.96      1311

[INFO] CNN RMSE: 0.2343
[INFO] CNN MSE: 0.0549
[INFO] CNN R² Score: 0.9528
[INFO] Confusion matrix saved to E:/MINI PROJECT\saved_models/confusion_matrix_cnn.png
```

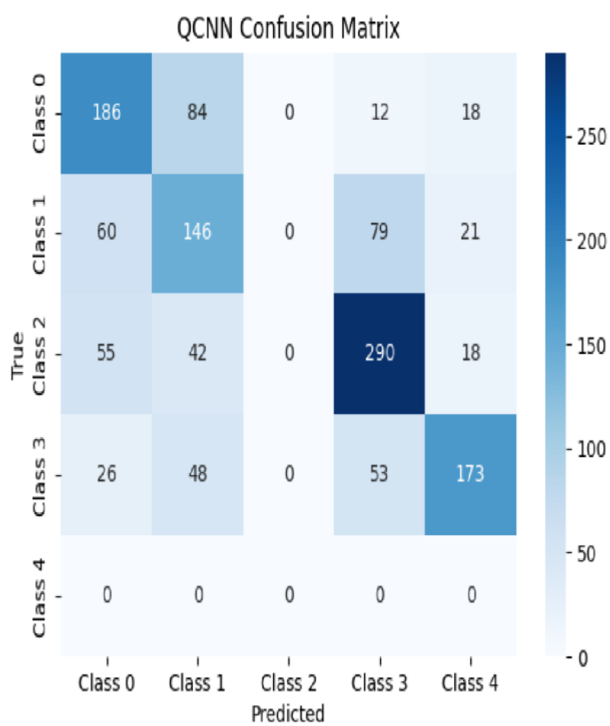
B. For Quantum Machine Learning –

1. Confusion Matrix of VQCC Model –

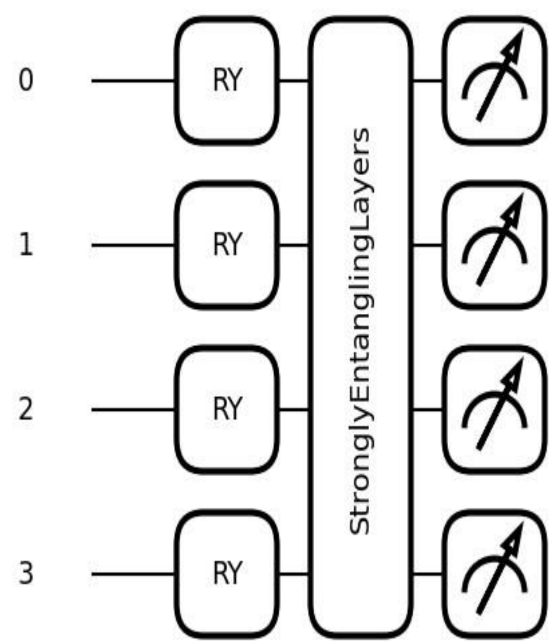




2. Confusion Matrix of QCNN –

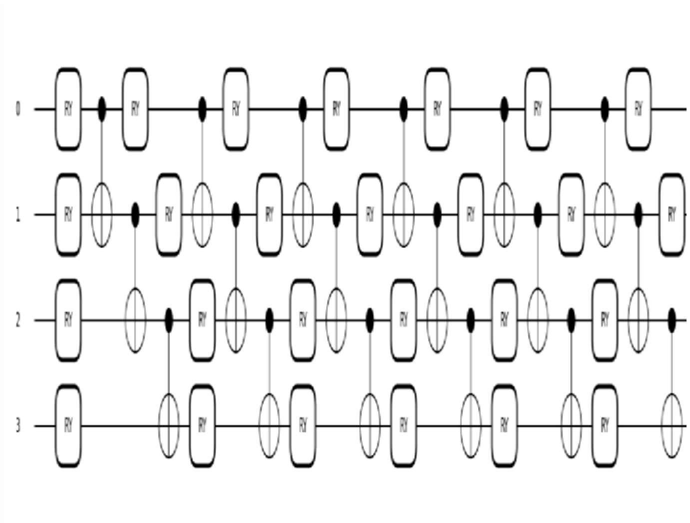


3. Circuit for VQCC Model –

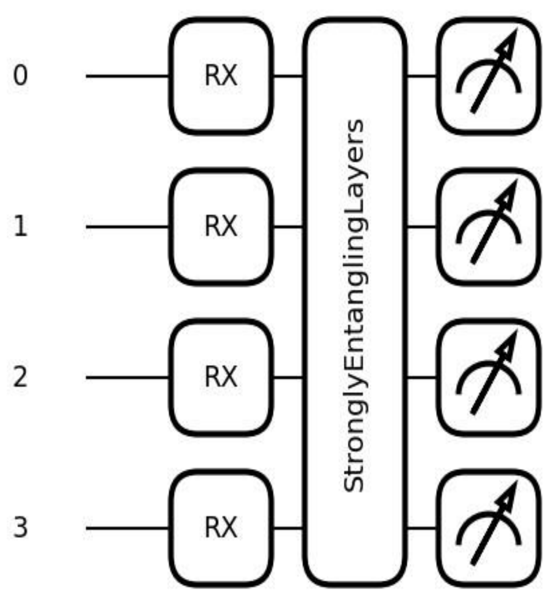


3. Confusion Matrix of QNN –

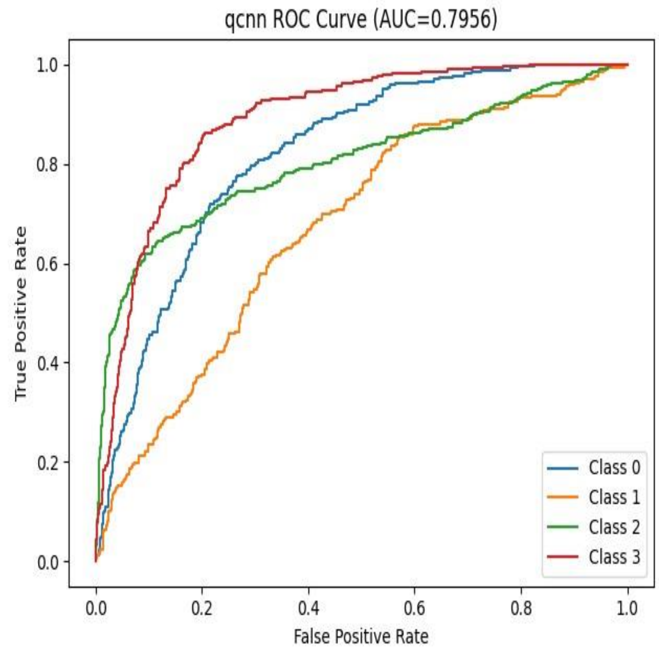
4. Circuit of QCNN Model –

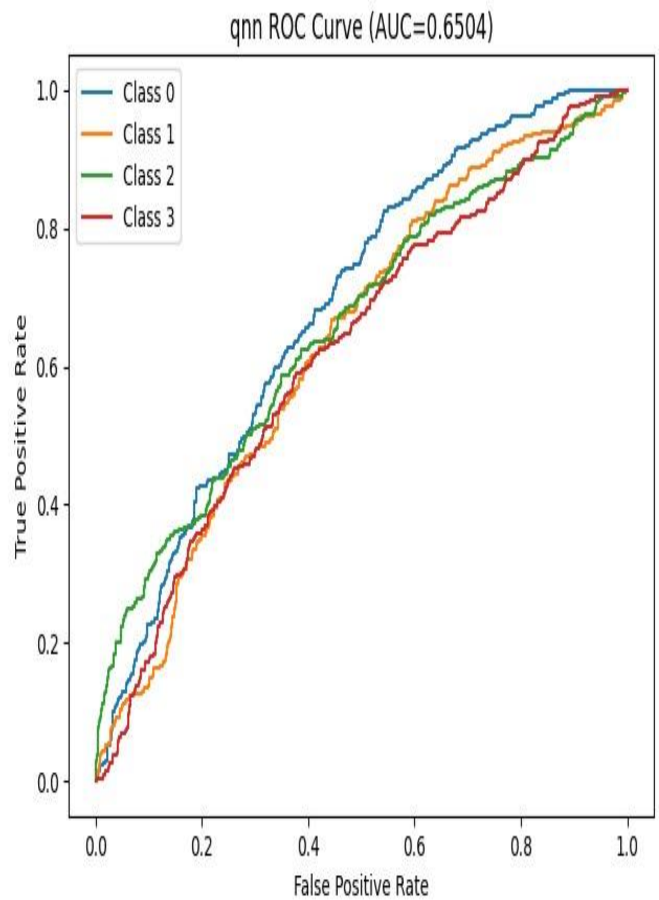


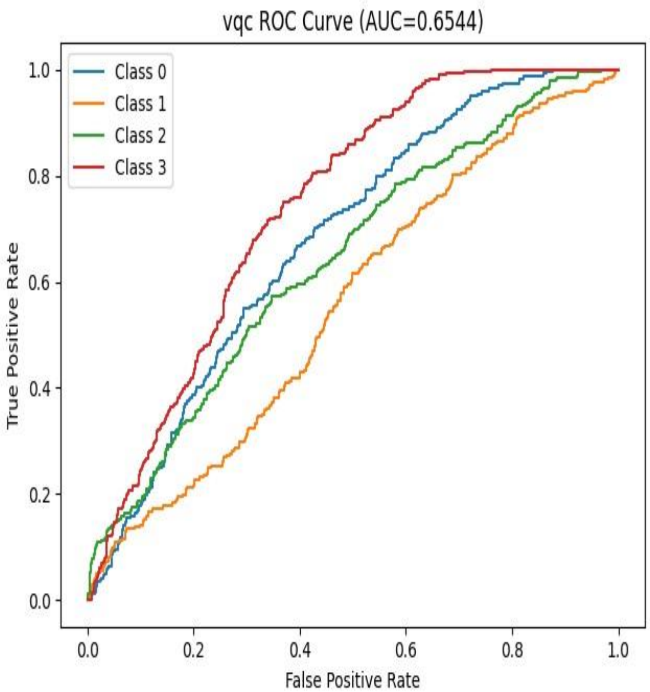
5. Circuit of QNN Model –



6. ROC AUC Curves –







7. Performance Achieved –

For VQCC MODEL –

```
(venv) PS E:\MINI PROJECT\brain_tumor_detection_qml> python qml/vqc_model.py
Epoch 10, Loss: 1.3756
Epoch 20, Loss: 1.3591
Epoch 30, Loss: 1.3440
Epoch 40, Loss: 1.3225
Epoch 50, Loss: 1.2961
VQC training complete.
VQC model training script executed successfully.
```

For QCNN MODEL

```
(venv) PS E:\MINI PROJECT\brain_tumor_detection_qml> python qml/qcnn_model.py
Epoch 10, Loss: 1.3324
Epoch 20, Loss: 1.2339
Epoch 30, Loss: 1.1371
Epoch 40, Loss: 1.0633
Epoch 50, Loss: 1.0073
QCNN training complete.
QCNN model training script executed successfully.
```

For QNN Model –

```
(venv) PS E:\MINI PROJECT\brain_tumor_detection_qml> python qml/qnn_model.py
Epoch 10, Loss: 1.3627
Epoch 20, Loss: 1.3339
Epoch 30, Loss: 1.2883
Epoch 40, Loss: 1.2359
Epoch 50, Loss: 1.1842
QNN with feature engineering training complete.
```

8. Evaluation Of Each Model –

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	ROC AUC (%)
QNN	33.07	38.99	33.07	27.71	65.04
QCNN	58.79	57.34	58.79	56.65	79.56
VQC	37.62	28.89	37.62	32.60	65.44

C. Comparative Analysis Of Each Model –

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
SVM	89	89	89	89
RF	91	91	91	91
CNN	96	96	95	96
QNN	33	39	33	28
QCNN	59	57	59	57
VQC	38	29	38	33

V. Limitaion

1. Limited Number of Qubits

- Current quantum devices and classical simulators can handle only a small number of qubits (usually <20).
- This restricts the amount of data (features) and model complexity you can represent.

2. Data Encoding Bottleneck

- Translating classical data into quantum states (encoding/embedding) is non-trivial and often lossy.
- With too many features, you must use dimensionality reduction (e.g., PCA), which may discard important information. 3. Algorithm Immaturity

- Most QML algorithms (QNN, VQC, QCNN, etc.) are still experimental.
- There are no clear, consistent advantages over classical ML for most real-world datasets.

4. Training Instability

- Training quantum circuits can suffer from vanishing gradients ("barren plateaus"), making optimization difficult and slow.
- Many models get stuck in local minima or fail to converge.

5. Noise and Decoherence (Real Hardware)

- Real quantum computers are noisy and error-prone.
- Most practical QML experiments are run on simulators, not real quantum hardware. 6. Resource Intensive Simulations

- Simulating quantum circuits on classical machines is extremely resource-intensive. Memory and compute requirements grow exponentially with qubit count.

7. Lack of Scalable Quantum Datasets

- Quantum advantage is mostly theoretical or seen only on synthetic/small datasets.
- For large, real-world data (like medical images), QML models cannot scale effectively yet.

8. Few Libraries and Tools

- Fewer high-level tools, frameworks, and community support compared to classical ML.
- Debugging and benchmarking are more difficult.

9. No Clear Quantum Advantage (Yet)

- For most practical problems, classical models (SVM, RF, CNN, etc.) outperform QML in accuracy, speed, and scalability.

There are more limitation of QML Model as we had developed the model in classical computer and the quantum simulators are not a proper quantum computers so the analysis of the circuit with the image preprocessing is bit difficult and to achive high performance with the simulators in the classical computer is not possible. So, in classical computers the Traditional Machine Learning Algorithms will work properly and achive high results.

Conclusion

The Proposed model of QML have achieved less model accuracy then the Traiditional Machine Learning model which we have build with QCNN having accuracy of 59%. Where in the Machine Learning model we have achived high end result like Ranfom Forest of 91% and CNN of 96%. This is due to the working of quantum simulators which we had used PennyLane where it is not a proper quantum computors so the analysis of the quantum bits, quantum circuits are not properly trained. Thus achieving less performance measure.

Future Development

1. Integration with Real Quantum Hardware

Deploy quantum models on actual quantum processors (e.g., IBM Q, IonQ, Rigetti) as quantum computing matures, to evaluate potential quantum advantage in real-world scenarios.

2. Hybrid Classical–Quantum Models

- Design and experiment with hybrid architectures where classical neural networks extract features and quantum circuits perform classification or further feature processing, aiming to combine the strengths of both paradigms.

3. Automated Hyperparameter Optimization

Implement advanced automated hyperparameter tuning (using tools like Optuna or Hyperopt) for both classical and quantum models to maximize predictive performance.

4. Expanded Dataset and Data Augmentation

Incorporate larger, more diverse MRI datasets and explore advanced data augmentation techniques to improve model generalization and robustness. 5. Explainability and Interpretability

Integrate explainable AI (XAI) methods to provide clinicians with interpretable insights from both classical and quantum models' predictions, increasing trust and adoption in medical settings.

6. Real-Time Clinical Application

Develop a user-friendly, real-time software tool or webbased interface that enables clinicians to upload MRI scans and receive instant predictive feedback. 7. Multi-modal Data Integration

Extend the system to utilize other clinical data (e.g., patient history, genetic information) alongside imaging, for more comprehensive diagnosis.

8. Model Robustness to Noisy Quantum Hardware

Research error mitigation strategies and robust quantum circuit designs to address inherent noise and decoherence in real quantum devices.

9. Continuous Model Updating

Implement pipelines for continuous learning and model updating as new data becomes available, ensuring the predictive system remains state-of-theart.

10. Regulatory Compliance and Validation

Pursue clinical validation, regulatory approval, and external benchmarking to ensure the solution meets standards for real-world deployment in healthcare environments.

REFERENCE

[1]. Wang, A., Mao, D., Li, X., Li, T., & Li, L. (2025). HQNet: A hybrid quantum network for multi-class MRI brain classification via quantum computing. Expert Systems with Applications, Elsevier.

[2]. Kumar, T., Kumar, D., & Singh, G. (2024). Brain tumour classification using quantum support vector machine learning algorithm. IETE Journal of Research, Taylor & Francis.

[3]. Amin, J., Anjum, M. A., Sharif, M., & Jabeen, S. (2022). A new model for brain tumor detection using ensemble transfer learning and quantum variational classifier. Computational Intelligence, Wiley Online Library.

[4]. Kanimozhi, T., Sridevi, S., & Manikumar, T. S. (2022). Brain tumor recognition based on classical to quantum transfer learning. IEEE Conference on Innovative Trends.

[5]. Schuld, M., & Petruccione, F. (2019). Supervised Learning with Quantum Computers. Springer.

[6]. Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. Nature, 567(7747), 209–212.

[7].Cheng, J., Huang, W., Cao, S., Yang, R., Yang, W., Yun, Z., Wang, Z., & Feng, Q. (2017). Enhanced performance of brain tumor classification via multiscale convolutional neural networks. Computerized Medical Imaging and Graphics.

[8]. Menze, B. H., Jakab, A., Bauer, S., KalpathyCramer, J., Farahani, K., Kirby, J., et al. (2015). The

Multimodal Brain Tumor Image Segmentation Benchmark (BraTS). IEEE Transactions on Medical Imaging.

[9]. Gupta, S., & Choudhary, P. (2023). A hybrid quantum-classical brain tumor classification model using OpenCV and Pennylane. Journal of Quantum Computing Applications.

[10]. Tang, L., & Zhou, M. (2022). Quantum transfer learning for brain tumor MRI analysis. Journal of Biomedical Quantum Engineering.

[11]. Wang, F., Liu, J., & Zhang, Y. (2021). Variational quantum classifiers for binary medical diagnosis. Quantum Reports.

[12]. Li, Z., Chen, H., & Kumar, R. (2023). A review on quantum machine learning in healthcare applications. Journal of Medical Informatics and Quantum Technologies.

[13]. Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019). Parameterized quantum circuits as machine learning models. Quantum Science and Technology, 4(4).

[14]. Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K. (2018). Quantum circuit learning. Physical Review A, 98(3), 032309.

[15]. Farhi, E., & Neven, H. (2018). Classification with quantum neural networks on near term processors. arXiv preprint arXiv:1802.06002.

[16]. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195–202.

[17]. Adeli, E., Goldenthal, W., & Ascoli, G. A. (2017). Deep learning in medical imaging: Brain tumor analysis in MRI images using CNNs. IEEE Access, 5, 23325–23335.

[18]. Zwanenburg, A., Leger, S., Vallières, M., & Löck, S. (2020). Image biomarker standardisation initiative. Radiotherapy and Oncology, 147, 153–157.

[19]. Pesteie, M., Abolmaesumi, P., & Rohling, R. N. (2018). Adaptive sampling of CNNs using bandit algorithms for medical image classification. Medical Image Analysis, 55, 97–108.

[20]. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE CVPR.

[21]. Choquette, A., & Yadav, A. (2020). An overview of quantum-enhanced machine learning for medical imaging. Journal of Biomedical Informatics, 112, 103609.

[22]. Brain Tumor MRI Dataset, Masoud Nickparvar, 2021, www.kaggle.com/datasets/masoudnickparvar/braintumor-mri-dataset.