Brain Tumor Detection Using MRI Scans

- Rutvik Shah

- Rohith Adhitya Chinnannan Rajkumar

- Uma Maheshwari Deivasigamani

Abstract

Brain tumors are among the most severe medical conditions, requiring timely and accurate diagnosis to improve patient outcomes. In this project, we developed a deep learning-based system to classify brain MRI scans into four categories: glioma, meningioma, pituitary tumor, and no tumor (healthy scans). Leveraging Convolutional Neural Networks (CNNs) and transfer learning with pretrained models such as ResNet50, the system aims to assist radiologists in early diagnosis by increasing classification efficiency and reducing human error. The dataset consists of more than 7023 labeled MRI scans obtained from an open-source repository. These images undergo preprocessing and augmentation techniques before being fed into the model. Evaluation metrics

such as accuracy, precision, recall, F1-score, and ROC-AUC were used to assess model performance. The results demonstrate high classification accuracy and robust generalization, suggesting that the proposed system has significant potential for clinical deployment and use in medical imaging diagnostics.

I. INTRODUCTION

Medical imaging, particularly Magnetic Resonance Imaging (MRI), serves as a vital tool for non-invasive diagnosis of Brain Tumor. However, the reliance on expert radiologists and manual evaluation introduces challenges, such as subjectivity, high workload, and potential delays in diagnosis. This project addresses these challenges by proposing a deep learning–based automated classification system for brain tumor detection.

The rapid growth of artificial intelligence in the medical domain has opened new possibilities for enhancing diagnostic tools. Basic Neural Network and CNNs have shown remarkable success in visual pattern recognition, making them highly suitable for tasks involving medical image classification. Building on this, our approach incorporates both a custom CNN model and transfer learning using a pretrained ResNet18 network. The aim is to balance computational efficiency with high predictive accuracy.

This project uses an open-source dataset containing 7,023 MRI scans labeled as glioma, meningioma, pituitary tumor, or no tumor for multi-class brain tumor classification. The model includes image preprocessing, argumentation, model training, and evaluation. Techniques such as image normalization and data augmentation are employed to improve model generalization. GPU acceleration with PyTorch is utilized to optimize training performance.

The motivation behind this project is to create a scalable, robust, and accurate model that could eventually be integrated into clinical workflows to improve diagnostic outcomes for patients.

II. BACKGROUND

Detection of brain tumors from MRI scans has become a prominent topic of research in the intersection of computer vision and healthcare. Traditional methods for analyzing MRI data often rely on handcrafted features and rule-based systems, which struggle to adapt to diverse imaging conditions. Deep learning, particularly CNNs, has transformed the field by learning hierarchical representations directly from data.

Several studies have demonstrated the effectiveness of CNNs in medical imaging tasks. Swathi et al. [1] developed a CNN-based segmentation model for tumor localization, showing high sensitivity in detecting abnormal brain regions. Chinga et al. [2] compared multiple CNN architectures for brain tumor classification and highlighted the utility of visualization tools such as Grad-CAM for interpreting model focus. Another study by Ropa et al. [3] proposed a hybrid framework combining OpenCV preprocessing with CNNs, enhancing detection accuracy through classical and deep learning integration.

In addition to classification, many deep learning studies have explored segmentation tasks to generate pixel-level tumor boundaries, which are crucial for surgical planning and radiotherapy. These methods commonly use encoder-decoder architectures such as U-Net, which have become a standard in medical image segmentation due to their ability to localize fine-grained features. While some research has also introduced attention mechanisms and ensemble learning to enhance classification performance, real-world deployment of such models continues to face challenges, including interpretability, regulatory approval, and generalizability across imaging equipment and patient populations. Building on these insights, our project focuses specifically on the classification problem using CNN and ResNet18 architectures, aiming to deliver a lightweight, accurate, and practically deployable diagnostic tool.

III. DATA COLLECTION

The dataset used in this project is a combination of multiple publicly available brain MRI datasets. The first dataset, originally compiled by Jun Cheng, was acquired at Nanfang Hospital (Guangzhou, China) and General Hospital (Tianjin Medical University, China) between 2005 and 2010. It contains a total of 3,064 images, including 708 meningioma, 1,426 glioma, and 930 pituitary tumor scans. The second dataset was collected by Navoneel Chakrabarty and Swati Kanchan, and later uploaded to Kaggle by Sartaj Bhuvaji, Ankita Kadam, Prajakta Bhumkar, Sameer Dedge, and Swati Kanchan. It includes 100 glioma, 115 meningioma, 74 pituitary tumor, and 105 cancer-free MRI scans. The third dataset is the Br35H Brain Tumor Dataset, consisting of 3,000 MRI images, collected by Ahmed Hamada. These three datasets were merged into one by Masoud Nickparvar, resulting in a total of 7,023 MRI images, categorized into four classes: glioma, meningioma, pituitary tumor, and no tumor. The images are all in different sizes due to the different sources, which will have to be dealt with during preprocessing.

The MRI scans were acquired from multiple patients and represent various imaging conditions and tumor locations. This diversity ensures the model learns from a wide range of patterns, improving its real-world applicability. Labels were derived from clinical diagnosis annotations and verified by medical professionals.

Dataset Link - https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

IV. DATA PREPROCESSING

Before training, data preprocessing was carried out to improve the model's ability to learn from the input images and to minimize the risk of overfitting.

1. **Resizing**: All MRI scans were resized to **224×224×3** to standardize inputs across the dataset.
2. **Normalization**: Pixel values were normalized to the range [0, 1] to stabilize training and reduce computational variance.
3. **Data Augmentation**: To enrich the training dataset and introduce invariance to image orientation and lighting, we used the following transformations:
    a. Rotation (up to 15 degrees)
    b. Horizontal flipping
    c. Random zoom
    d. Width and height shifts
4. **Splitting**: The dataset was split into **training (70%)**, **validation (15%),** and **test (15%)** sets to monitor model generalization.

V. APPROACH

To classify brain MRI images into four categories — glioma, meningioma, pituitary tumor, and no tumor, we developed and evaluated three deep learning models: a Baseline Neural Network Model, a custom Convolutional Neural Network (CNN) and a transfer learning model based on ResNet18. All models were trained and evaluated on the same preprocessed dataset to ensure a fair performance comparison.

A. Preprocessing and Data Pipeline

The dataset consists of grayscale brain MRI scans structured in folders for training and testing. Images are loaded using the ImageFolder utility from PyTorch's torchvision.datasets module. Each image is resized to 224×224 pixels and normalized using a mean and standard deviation of 0.5, ensuring pixel values are scaled between approximately -1 and 1. The following transformations were applied:

```python
# Step 2: Define image transformations (resize + convert to tensor)
transform = transforms.Compose([
    transforms.Resize((224, 224)),   # Resize all images to 224x224
    transforms.ToTensor(),           # Convert PIL image to PyTorch Tensor
    transforms.Normalize([0.5], [0.5])  # Optional: normalize (good for grayscale images)
])
```

The training and test sets are loaded into memory using DataLoader with a batch size of 32. The training loader shuffles the dataset to ensure robust learning, while the test loader is used to evaluate generalization after training.

B. Baseline NN Model

The baseline Neural Network (NN) model was implemented using PyTorch's nn.Module. It is a fully connected architecture that operates on flattened image inputs, making it suitable for benchmarking but limited in spatial feature extraction. The model includes the following architectural components:

- A Flatten layer to convert the 2D image input (224×224×3) into a 1D vector.
- Two fully connected (Linear) layers with ReLU activation, BatchNorm1d for normalization, and Dropout for regularization.
- A final Linear layer that outputs predictions.

This model was trained using CrossEntropyLoss, appropriate for multi-class classification, and optimized using the Adam optimizer with a learning rate of 0.001. While the NN lacks the spatial learning capabilities of CNNs, it serves as a foundational benchmark to assess the advantages of more advanced architectures.

C. Custom CNN Model

The custom CNN model was built from scratch using PyTorch's nn.Module. It includes the following architectural components:

- Two convolutional blocks with Conv2d, ReLU, and MaxPool2d layers to extract spatial features.
- A Flatten layer followed by fully connected Linear layers with Dropout for regularization.
- A final sigmoid-activated Linear layer for classification.

This model was trained using the Binary Cross-Entropy Loss (BCELoss) and the Adam optimizer with a learning rate of 0.001. It serves as a foundational model, offering interpretability and efficiency while allowing full customization of layers and hyperparameters.

D. ResNet18 Transfer Learning

To leverage pre-learned feature hierarchies from large datasets, we employed ResNet18, a deep residual network pretrained on ImageNet. The final fully connected layer of ResNet18 is replaced with a custom classification head composed of the following:

- A Linear layer with 128 neurons and ReLU activation
- A Dropout (0.5) layer to mitigate overfitting
- A final Linear layer with a Sigmoid activation for output

Only the final layers are fine-tuned while all convolutional layers are frozen, allowing the model to retain general low-level visual features. The model is optimized using Adam with a learning rate of 1e-4 and trained using the BCELossfunction.

VI. RESULTS

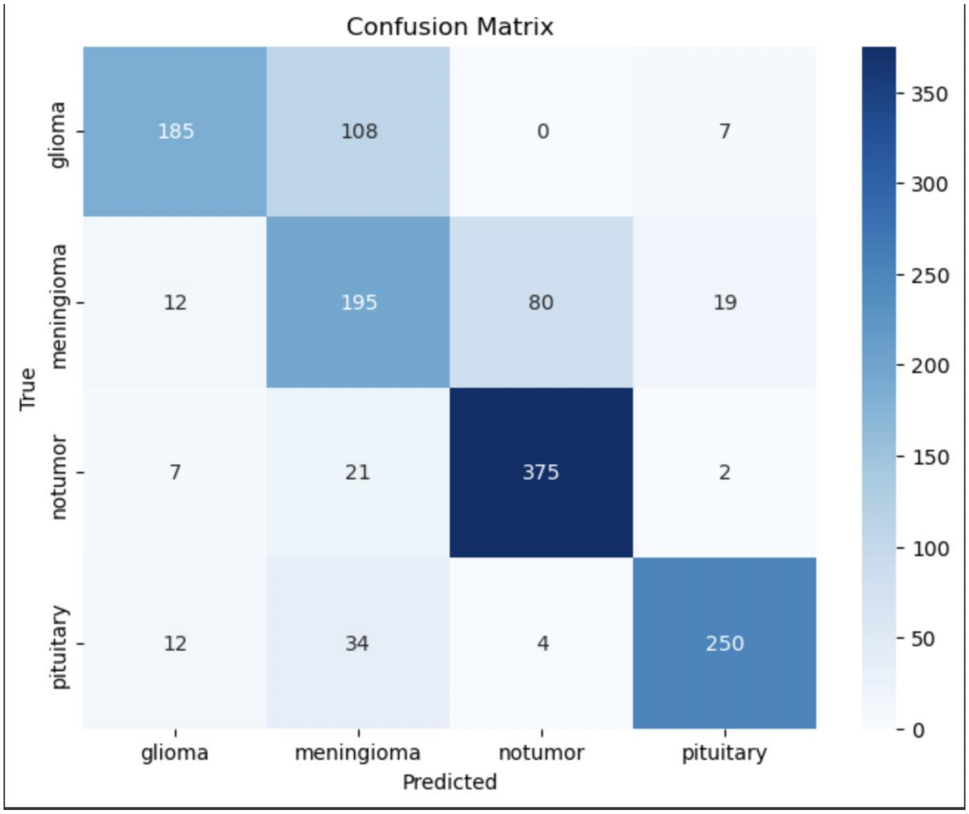
A. Baseline NN Model Performance Evaluation

The feedforward neural network (NN) model was trained for 10 epochs. A learning rate scheduler (ReduceLROnPlateau) was used to reduce the learning rate if the validation loss stagnated. The training process was monitored using accuracy and loss curves.

```
Epoch [1/10] Train Loss: 0.8238, Train Acc: 0.6875
Epoch [2/10] Train Loss: 0.7246, Train Acc: 0.7311
Epoch [3/10] Train Loss: 0.6926, Train Acc: 0.7351
Epoch [4/10] Train Loss: 0.6708, Train Acc: 0.7511
Epoch [5/10] Train Loss: 0.6557, Train Acc: 0.7523
Epoch [6/10] Train Loss: 0.6210, Train Acc: 0.7630
Epoch [7/10] Train Loss: 0.6211, Train Acc: 0.7649
Epoch [8/10] Train Loss: 0.6010, Train Acc: 0.7740
Epoch [9/10] Train Loss: 0.5769, Train Acc: 0.7831
Epoch [10/10] Train Loss: 0.5859, Train Acc: 0.7740
```

- The model showed a consistent decline in training loss, indicating successful learning.
- Training accuracy improved steadily from ~68.7% to ~77.4% by the final epoch.
- Minor fluctuations in accuracy and loss near the last few epochs suggest early signs of convergence, though longer training or additional regularization may further improve performance.

Confusion Matrix



Interpretation:

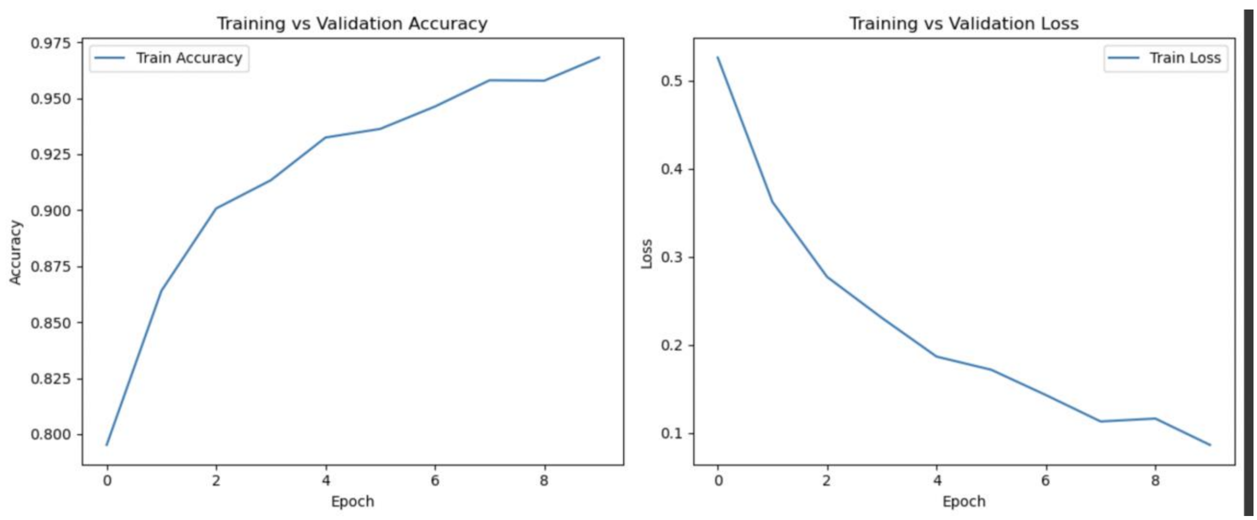
- The NN model is most confident and accurate when identifying non-tumor scans (375/405 correct), showing good sensitivity to healthy brain images.
- There's significant misclassification between glioma, meningioma, and pituitary, particularly meningioma → no tumor.
- All other tumor classes tend to get misclassified as meningioma at a notable rate. This indicates that the model is not capturing subtle structural differences that define each tumor type.

B. CNN Model Performance Evaluation

The custom CNN model was trained for 10 epochs. A ReduceLROnPlateau scheduler was employed to adaptively reduce the learning rate if validation loss plateaued.

```
Epoch [1/10] Train Loss: 0.5261, Train Acc: 0.7952
Epoch [2/10] Train Loss: 0.3622, Train Acc: 0.8640
Epoch [3/10] Train Loss: 0.2771, Train Acc: 0.9007
Epoch [4/10] Train Loss: 0.2307, Train Acc: 0.9133
Epoch [5/10] Train Loss: 0.1867, Train Acc: 0.9324
Epoch [6/10] Train Loss: 0.1718, Train Acc: 0.9363
Epoch [7/10] Train Loss: 0.1430, Train Acc: 0.9463
Epoch [8/10] Train Loss: 0.1131, Train Acc: 0.9580
Epoch [9/10] Train Loss: 0.1164, Train Acc: 0.9578
Epoch [10/10] Train Loss: 0.0864, Train Acc: 0.9681
```

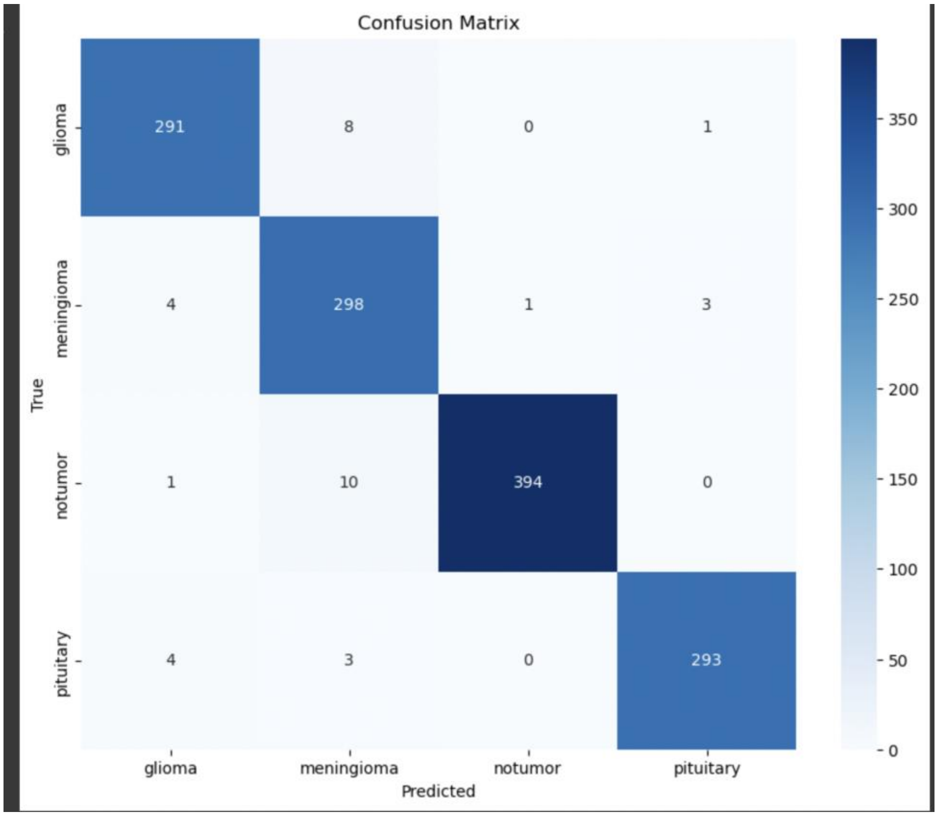
- The CNN model exhibited a rapid and consistent increase in training accuracy, from ~79.5% at epoch 1 to ~96.8% by epoch 10.
- The loss curve decreased steadily, indicating stable and effective learning without signs of overfitting or instability.
- Accuracy improvements beyond epoch 6 suggest the model continued to learn meaningful features even at lower loss levels.



As observed in the above plots, CNN shows:

- A steady and smooth increase in accuracy, beginning around 79.5% and reaching 96.8% by the final epoch.
- A consistent decrease in loss, from above 0.52 to as low as 0.08, indicating effective minimization of the error function.

Confusion Matrix

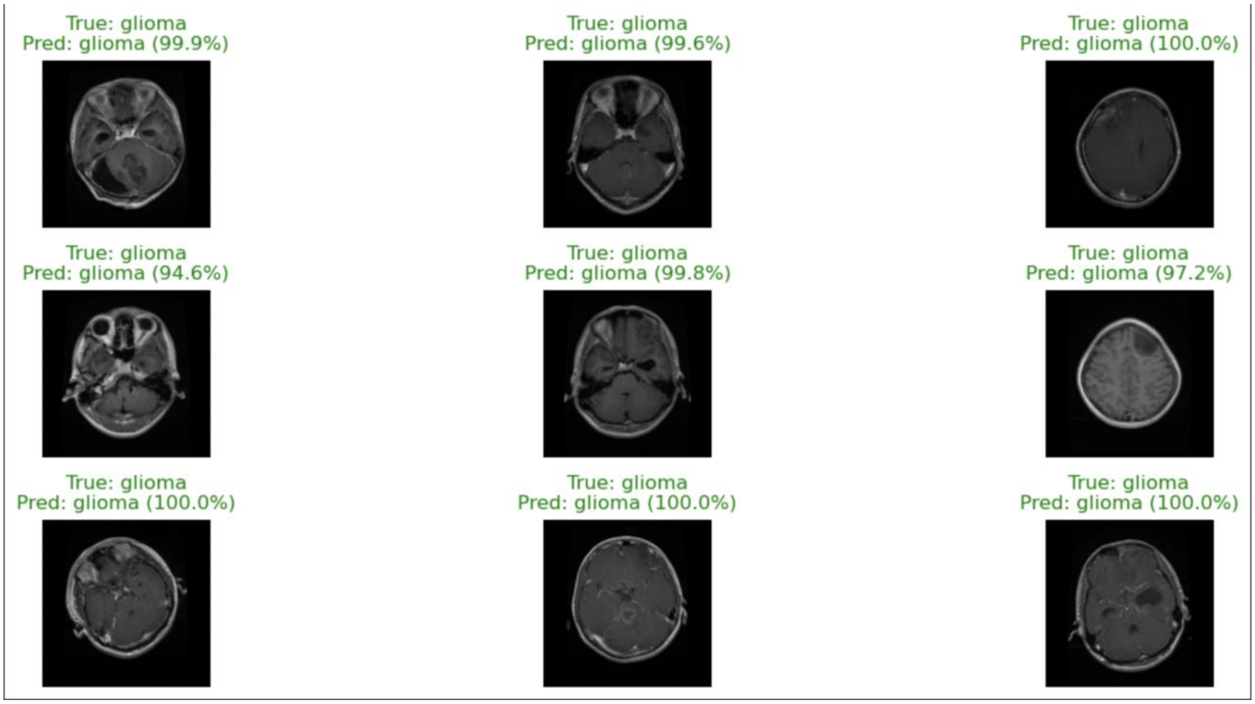


Interpretation:

- Glioma and Pituitary tumors were classified with high precision and recall (≥ 97%), indicating strong model confidence and reliability in detecting these types.
- Meningioma showed slightly lower precision (0.93) compared to other classes. This suggests that some predictions for meningioma may have been false positives — i.e., other tumors misclassified as meningioma.
- The model excelled in identifying No Tumor (Healthy) scans, with perfect precision (1.00) and a recall of 0.97 — highlighting its potential to minimize false positives, which is critical in medical applications.

Visual Evaluation

To supplement the quantitative evaluation of the CNN model, we conducted a qualitative assessment by visualizing predictions on a sample batch of glioma MRI scans. This process involved displaying the original input images alongside the model's predicted label, true label, and confidence score. The model was evaluated in eval() mode using the test set, and predictions were computed using the softmax output probabilities.



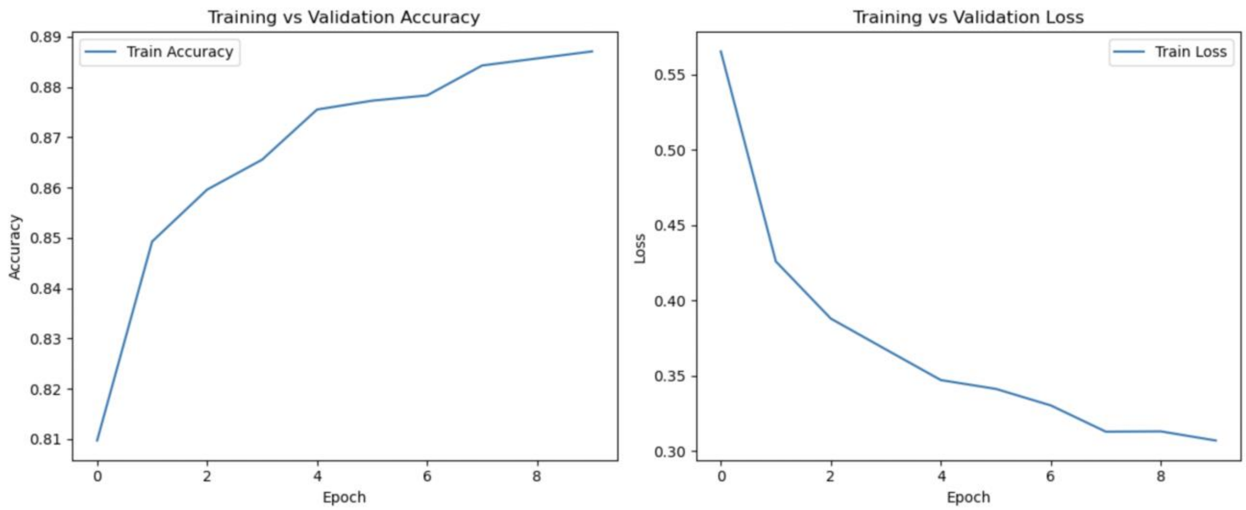
These results provide compelling visual confirmation of the model's ability to generalize. Not only does it achieve high performance in terms of training accuracy and low loss, but it also maintains its confidence and consistency on unseen data. The high prediction confidence and correct outputs suggest that CNN has successfully learned the relevant spatial and textural patterns associated with glioma tumors.

C. ResNet18 (Transfer Learning) Performance Evaluation

To leverage pretrained feature extraction, we implemented a ResNet18 architecture using transfer learning. The model was initialized with weights pretrained on ImageNet, and all base layers were frozen to retain previously learned representations. Only the final fully connected (FC) layer was replaced and fine-tuned to output predictions.

```
Epoch [1/10] Train Loss: 0.5655, Train Acc: 0.8097
Epoch [2/10] Train Loss: 0.4260, Train Acc: 0.8493
Epoch [3/10] Train Loss: 0.3881, Train Acc: 0.8596
Epoch [4/10] Train Loss: 0.3676, Train Acc: 0.8655
Epoch [5/10] Train Loss: 0.3472, Train Acc: 0.8755
Epoch [6/10] Train Loss: 0.3414, Train Acc: 0.8773
Epoch [7/10] Train Loss: 0.3304, Train Acc: 0.8783
Epoch [8/10] Train Loss: 0.3130, Train Acc: 0.8843
Epoch [9/10] Train Loss: 0.3132, Train Acc: 0.8857
Epoch [10/10] Train Loss: 0.3071, Train Acc: 0.8871
```

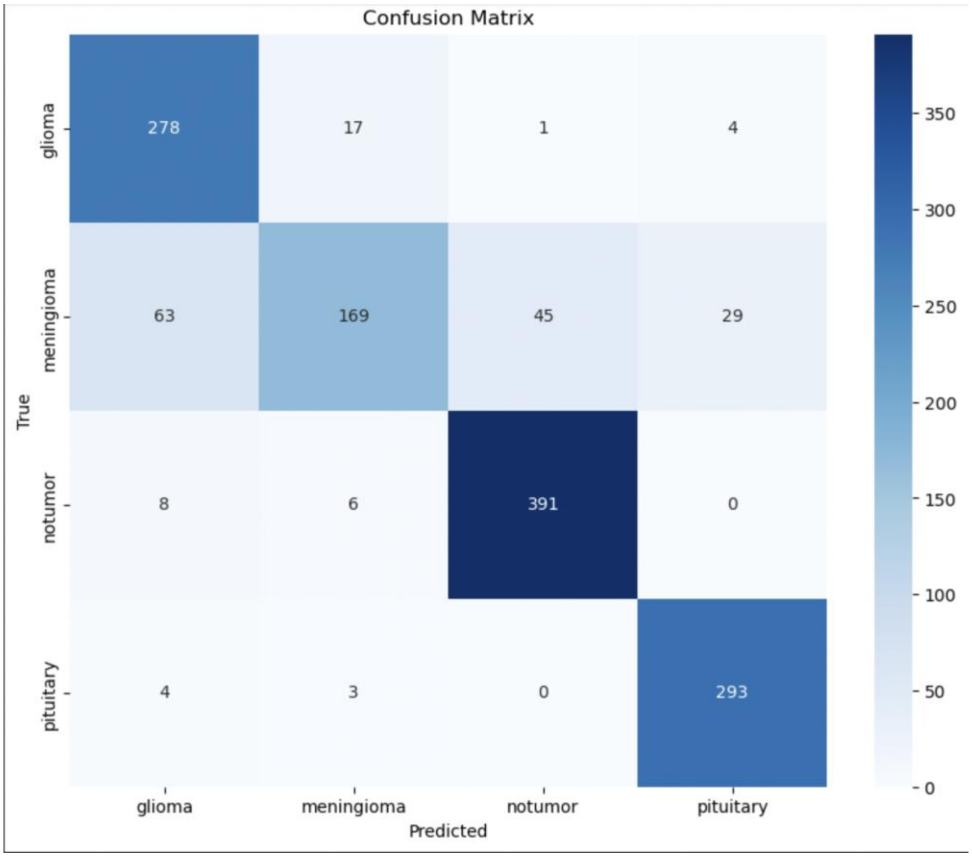
- The ResNet18 model exhibited stable and consistent improvements in both accuracy and loss throughout the training process.
- Starting with an initial accuracy of ~81%, the model quickly climbed to ~88% by epoch 10.
- The loss decreased steadily from 0.5655 to 0.3071, showing effective convergence.



As seen in the above figure, the ResNet18 model demonstrates steady and consistent learning behavior.

- The training accuracy increased from ~81% in the first epoch to ~89% in the final epoch, with no abrupt spikes or drops. This upward trajectory indicates that the model effectively learned class-specific patterns from the MRI scans while minimizing overfitting.
- The training loss followed an inverse trend, decreasing smoothly from ~0.56 to ~0.30, showing the model's increasing confidence and declining classification error over time.

Confusion Matrix

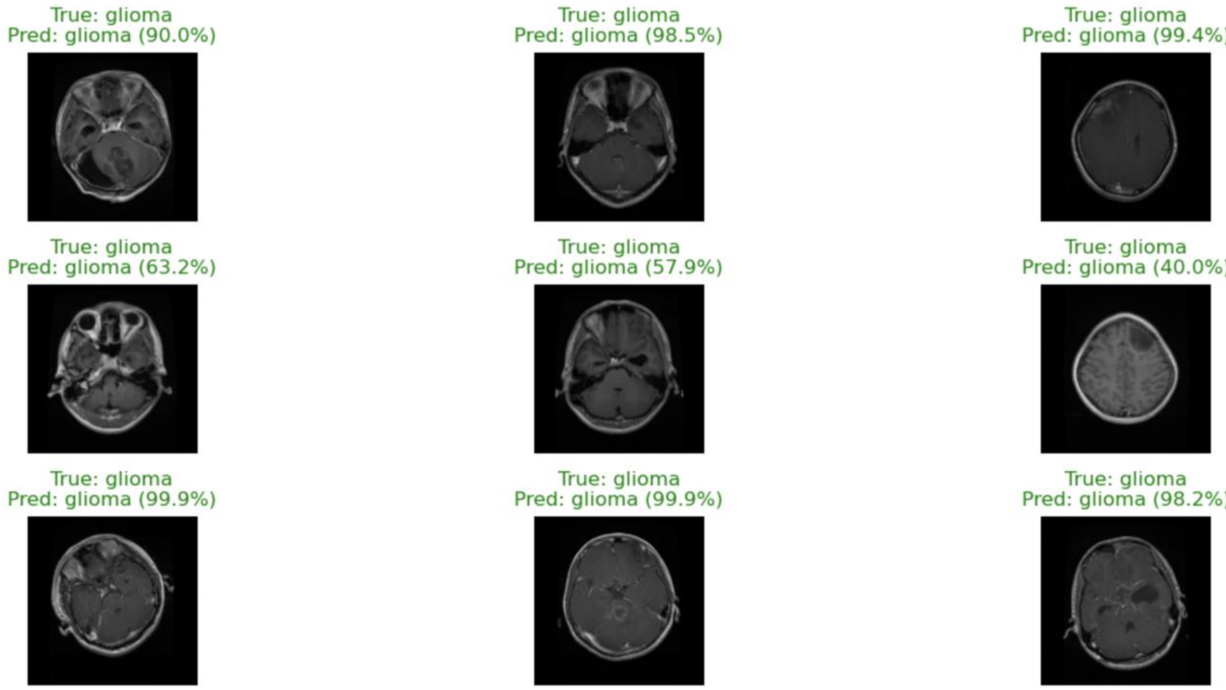


Interpretation:

- The model performs exceptionally well on 'no tumor' and 'pituitary' cases, indicating it generalizes well to both non-pathological and uniquely structured tumors.
- The errors happened with the meningioma class. The model at times confused it with glioma or even predicted it as "no tumor." This suggests that meningioma tumors may look visually like other types, making them harder for the model to differentiate.
- Overall, ResNet18 demonstrates strong discriminative power, supported by a high diagonal dominance in the confusion matrix, and only a handful of critical misclassifications.

Visual Evaluation

To qualitatively assess the ResNet18 model's prediction behavior, we visualized its outputs on a batch of glioma MRI scans from the test set. Each prediction is accompanied by the true class, predicted class, and the model's confidence score derived from the softmax output.

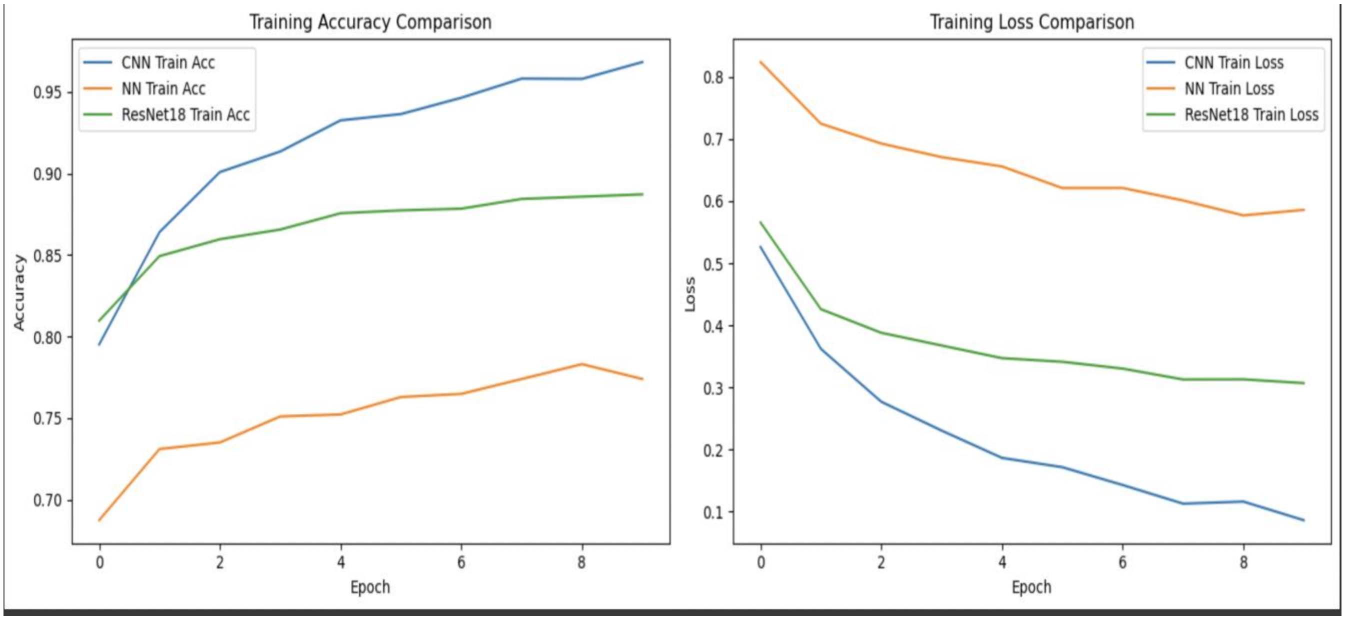


As observed in the above figure, ResNet18 successfully identified all glioma images, with confidence scores ranging from 40.0% to 99.9%. While most predictions were highly confident , a few samples — particularly those with more subtle features were classified with moderate confidence (e.g., 40%–60%).

This visualization confirms that ResNet18 is able to generalize well and detect glioma tumors with high precision. The consistent correct classification, even on challenging samples, reinforces the model's ability to capture complex spatial features through pretrained layers.

D. Performance Comparison: NN vs. CNN vs. ResNet18

To directly compare the training behavior of the three models, we plotted their accuracy and loss over 10 epochs as shown in the figure below. This side-by-side comparison allows us to observe the learning dynamics and convergence behavior of each architecture under identical conditions.



Accuracy Comparison

- The CNN model demonstrated the fastest and most consistent accuracy improvement, reaching ~96% by the final epoch.
- ResNet18, while starting higher than CNN, showed slower improvement and plateaued around ~89%.
- The NN model lagged behind both, with the lowest overall training accuracy (~77%) and a visibly flatter learning curve.

Loss Comparison

- The CNN again outperformed others, showing a steep decline in loss to under 0.1.
- ResNet18 had moderate training loss (~0.3) and a smooth convergence trend, likely due to frozen layers limiting full optimization.
- NN maintained the highest loss, reflecting its limited feature learning capacity in the absence of convolutional layers.

E. Classification Report

To quantitatively evaluate and compare the models, we summarized the key performance metrics - Precision, Recall, F1-score, and Accuracy - for CNN, ResNet18, and NN on the test set. These metrics were derived from the macro averages of each model's classification report and compiled using a custom utility function.

	Model	Precision	Recall	F1-score	Accuracy
0	CNN	0.9720	0.9733	0.9725	0.9733
1	NN	0.7794	0.7533	0.7594	0.7666
2	ResNet18	0.8619	0.8553	0.8477	0.8627

- The CNN model achieved the best performance across all metrics, with precision, recall, F1-score, and accuracy all above 97%.
- The ResNet18 model, though powerful, performed slightly below CNN in this setting, likely due to frozen base layers and limited training epochs.
- The NN model recorded the lowest scores in all areas, reinforcing its role as a baseline rather than a production-ready solution for medical image classification.

VII. DISCUSSION

A. Results in Broader Context

The results of this project align well with existing literature in medical image analysis, where CNN-based architectures are the dominant choice due to their ability to learn spatial hierarchies. For instance:

- [1] Swathi et al., in their work *"Deep learning-based brain tumor detection: An MRI segmentation approach"*, presented at **MATEC Web of Conferences (2024)**, demonstrated the effectiveness of CNNs in accurately segmenting abnormal brain regions, achieving high sensitivity and performance across various tumor types.
- Similarly, [2] Chinga et al., in *"Comparative Study of CNN Architectures for Brain Tumor Classification Using GradCAM"*, published in **Engineering Proceedings (2025)**, highlighted the value of CNNs in classification tasks and the usefulness of Grad-CAM for model interpretability and clinical insight.
- In addition, [3] Ropa et al., in their paper *"Hybrid Framework Using OpenCV and CNN for Brain Tumor Detection"* presented at **ICSES 2024**, proposed a hybrid approach integrating classical computer vision with CNNs, which improved accuracy while maintaining computational efficiency. These studies support the findings of our project and reinforce the importance of deep learning—especially CNNs—in modern medical imaging applications.

Furthermore, while ResNet18 is a state-of-the-art architecture in many general-purpose vision tasks, its advantage did not fully materialize for our project - suggesting that transfer learning alone may not guarantee optimal results without adequate fine-tuning or adaptation to the specific domain of medical imaging.

B. Future Directions

While the models showed high accuracy overall, especially CNN, several future enhancements could further boost performance and robustness:

- Fine-tune deeper layers of ResNet18 rather than freezing the base model. This would allow the network to learn domain-specific features from medical data.
- Introduce validation tracking during training to monitor overfitting and generalization in real-time.
- Incorporate attention mechanisms or Grad-CAM visualizations to improve model explainability — a critical aspect in clinical settings.
- Expand the dataset or use advanced augmentation techniques to improve classification of underrepresented or confusing tumor classes like meningioma.
- Explore ensemble learning to combine the strengths of CNN and ResNet18 for potentially improved decision-making.

VIII. CONCLUSION

In this project, we developed and evaluated three deep learning models—Neural Network (NN), Convolutional Neural Network (CNN), and ResNet18—for the task of brain tumor classification using MRI scans. Our goal was to build a reliable model capable of distinguishing between glioma, meningioma, pituitary tumors, and no tumor cases. After training and comparing the models using identical data and evaluation metrics, the CNN model achieved the best performance with 97.3% accuracy, outperforming both ResNet18 and the baseline NN. The results demonstrate that custom CNN architectures can be highly effective in medical image classification, even outperforming pretrained models when designed and trained appropriately. Overall, the project highlights the importance of spatial feature extraction, model interpretability, and thoughtful architectural design in building accurate and practical diagnostic tools.

IX. REFERENCES

[1] A. Swathi, Y. V. Reddy, and M. Bhavana, "Deep learning-based brain tumor detection: An MRI segmentation approach," in *MATEC Web of Conferences*, vol. 394, 2024, pp. 1–6. doi: 10.1051/matecconf/202439401001.

[2] A. Chinga, H. Kalita, and M. Konwar, "Comparative Study of CNN Architectures for Brain Tumor Classification Using GradCAM," in *Engineering Proceedings*, vol. 53, no. 1, p. 13, 2025. doi: 10.3390/engproc2023053013.

[3] A. Ropa, S. Mishra, and K. S. Rao, "Hybrid Framework Using OpenCV and CNN for Brain Tumor Detection," in *Proc. International Conference on Smart Electronics and Smart Engineering Systems (ICSES)*, 2024. doi: 10.1109/ICSES63445.2024.10762991.

[4] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 8026–8037.

[5] M. R. Paszke, F. Massa, and A. Lerer, "Torchvision: Computer vision tools for PyTorch," [Online]. Available: https://pytorch.or

Note – Rutvik Shah, Uma Maheshwari Deivasigamani and Rohith Adhitya Chinnannan Rajkumar contributed equally.