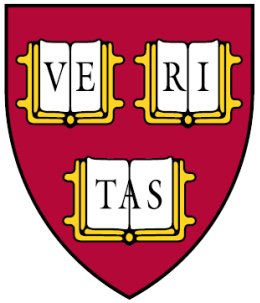
Final Project
Unified CNN Approach for Multi-Class Brain Tumor Classification

Math & Physics Fun With Gus



CSCI S-89 Introduction to Deep Learning
Summer 2023
Harvard Summer School

Introduction

- Welcome to my study on MRI Brain Tumor Classification using Convolutional Neural Networks (CNNs).
  - Explore the application of deep learning in medical diagnostics.
  - Develop a unified CNN for multi-class brain tumor classification from MRI scans.
  - Enhance brain tumor detection and accuracy through CNNs.

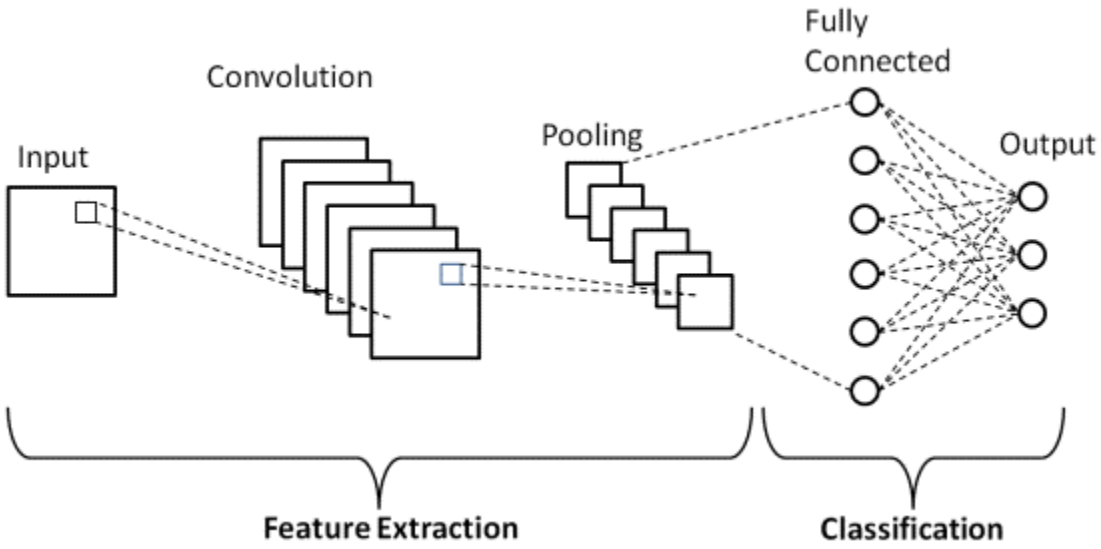


Figure 1: CNN general architecture [1]

Methodology Part 1 – Problem Statement

- **Problem Statement:** Develop a CNN-based deep learning model that can accurately classify MRI brain tumor images into: Glioma, Meningioma, No Tumor, and Pituitary categories.

- **Objective:** Achieve high-test accuracy for accurate brain tumor classification to help properly diagnosis people and help with earlier and proper treatment for patients

- **CNN Architecture:** Design multi-layered CNN with convolutional, pooling, etc. as well as fully connected layers for feature extraction and tumor classification.

- **Significance:**
  - Early and accurate diagnosis of brain tumors leads to improved medical treatment and patient outcomes.
  - Efficient brain tumor classification assists medical professionals in making informed decisions.
  - Leveraging the potential of CNNs automates classification and can lead to more personalized treatment plans.

Methodology Part 2 – About the Data

- About Brain Tumors
  - Brain tumors are abnormal collections of cells in the brain, that cause nerve damage or even life-threatening risks.
- About the Dataset
  - Contains 7,023 brain MRI images categorized into four classes:
  - Glioma: Cancerous brain tumors in glial cells.
  - Meningioma: Non-cancerous tumors originating from the meninges.
  - No Tumor: Normal brain scans without detectable tumors.
  - Pituitary: Tumors affecting the pituitary gland (cancerous or non-cancerous).

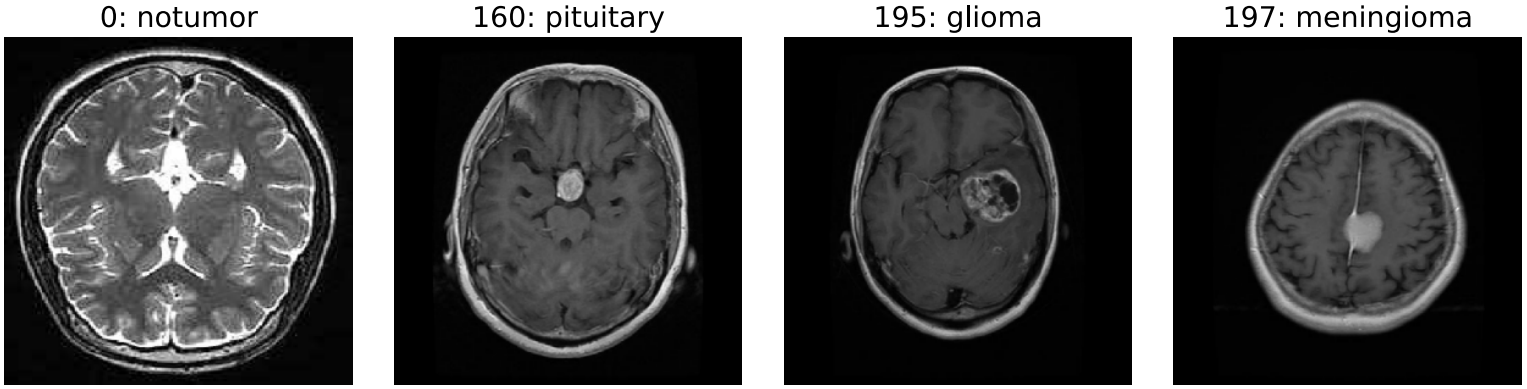


Figure 2: Each type of class image from test data

From Kaggle Brain Tumor MRI Dataset

Methodology Part 3 – Data Distributions

- Data Split & Distribution of Categories
  - The dataset was nicely split into a training and testing sets
    - Training set has 5712 images, about 80% of the total data.
    - Testing sed has 1311 images, about 20% of the total data.
  - The distribution of categories in the training set is well-balanced to ensure representation from all classes.

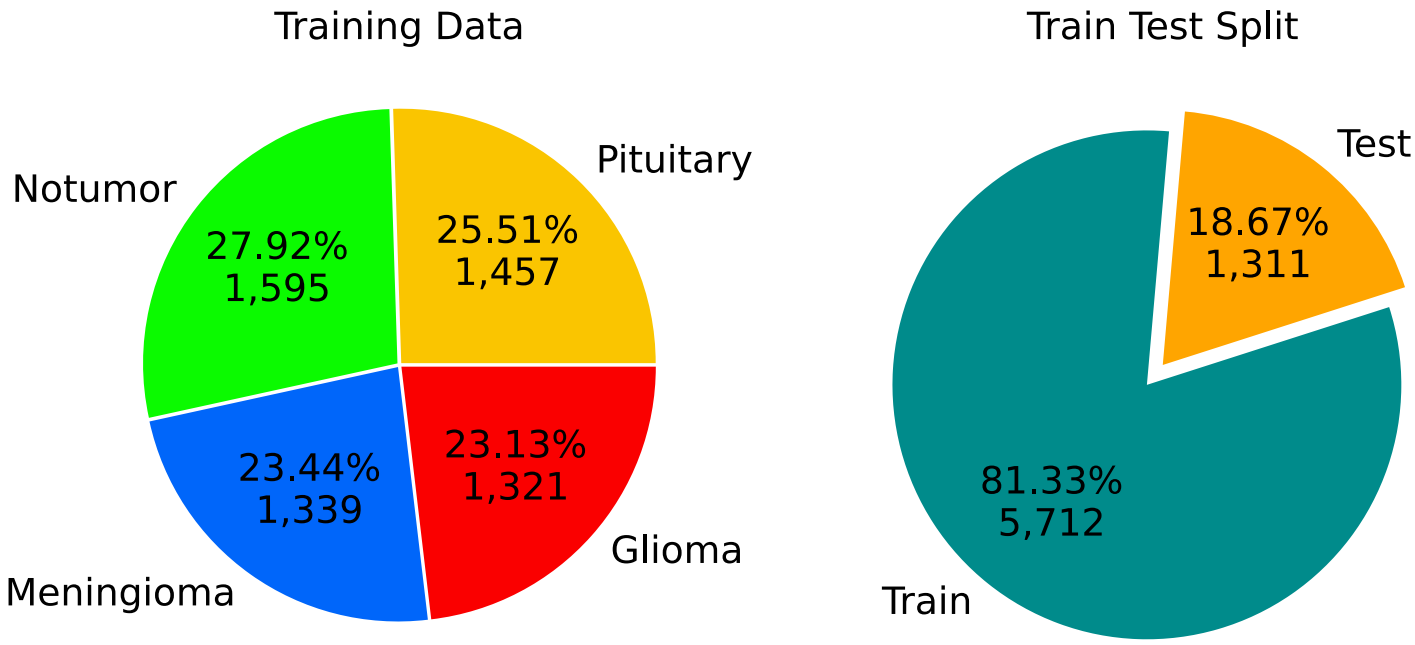


Figure 3: Data classes percentages and percent of training and testing data

Augmentation

- Image Shapes: (150, 150, 3)
  - The input images are resized to a shape of 150x150 pixels with three color channels (RGB).
- Data augmentation
  - Function: ImageDataGenerator from TensorFlow.Keras.
  - Applied augmentations: Rescaling, rotation, brightness shifts, width and height shifts, shearing, and horizontal flip.
- For testing data, only rescaling was applied to maintain consistency with real-world scenarios.

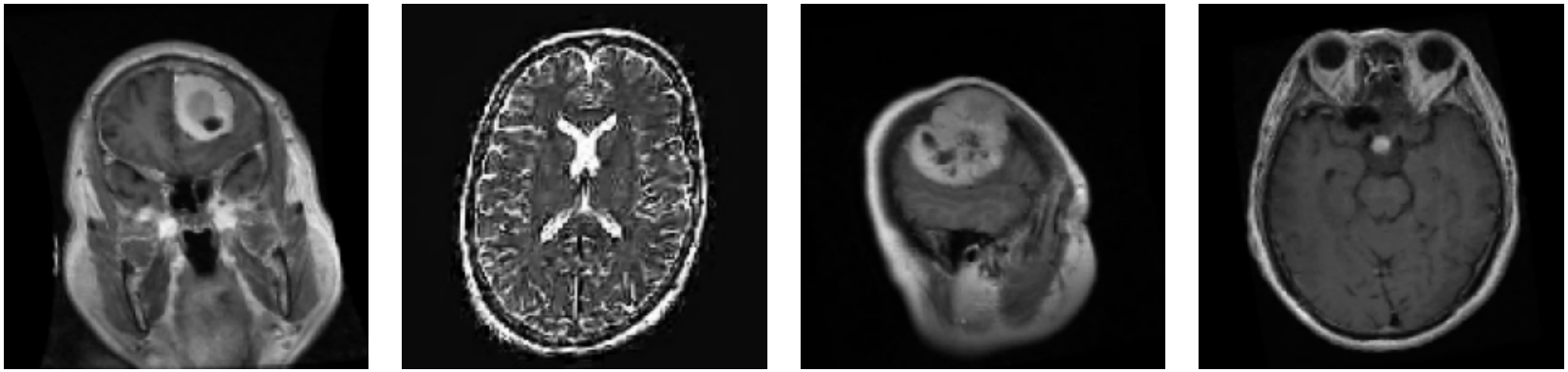


Figure 4: Display of training data augmentation

General Model Training Parameters

- **Epochs**: 40
  - The model was trained for 40 epochs to allow for sufficient training andalso preventing overfitting. Experimentation demonstrated that convergence was achieved within this epoch range.
- **Batch Size**: 32
  - A batch size of 32 was chosen to ensure efficient memory usage during training, allowing the model to process multiple samples simultaneously and update weights more frequently.
- **Steps Per Epoch**: 178
  - With 5712 training images and a batch size of 32, 178 steps per epoch were computed to cover the entire training dataset during each epoch.
- **Validation Steps**: 40
  - Like the training steps, the validation steps were set to 40, corresponding to the 1311 test images, to evaluate the model's performance during validation.

Initial Model - Testing & Architecture for Initial Model

- Base Model General Architecture:
  - 4 Convolutional layers with ReLU activation.
  - 2 MaxPooling2D layers with pool sizes (3, 3) and (3, 3) respectively.
  - 1 Flatten layer to reshape the output for the fully connected layers.
  - 1 Dense layer with 512 units and ReLU activation.
  - 1 Dropout layer with a dropout rate of 0.5 to reduce overfitting.
  - Output layer with softmax activation for 4 classification categories.
- Summary of Initial Model:
  - Total Parameters: 495,972
  - 0 non-trainable Parameters
- Initial Model Hyperparameter Experiments:
  1. Filter size: (4, 4), Pool size: (2, 2) - Test Accuracy: 0.96
  2. Filter size: (3, 3), Pool size: (3, 3) - Test Accuracy: 0.96
  3. **Filter size: (4, 4), Pool size: (3, 3) - Test Accuracy: 0.98**
  4. Filter size: (3, 3), Pool size: (2, 2) - Test Accuracy: 0.97
  - Average Pooling was tested but did not significantly improve performance.
- Initial Model Optimizer Experiments:
  1. **Adam - Test Accuracy: 0.98**
  2. RMSprop - Test Accuracy: 0.97
  3. Nadam - Test Accuracy: 0.96

Initial Model – Training Plots

- **Training Process For Initial Model**:
  - This achieved about a 98% accuracy on the testing dataset
  - Improvement Goal for Final Model:
    - Less randomness on the test accuracy epochs
    - Increased accuracy

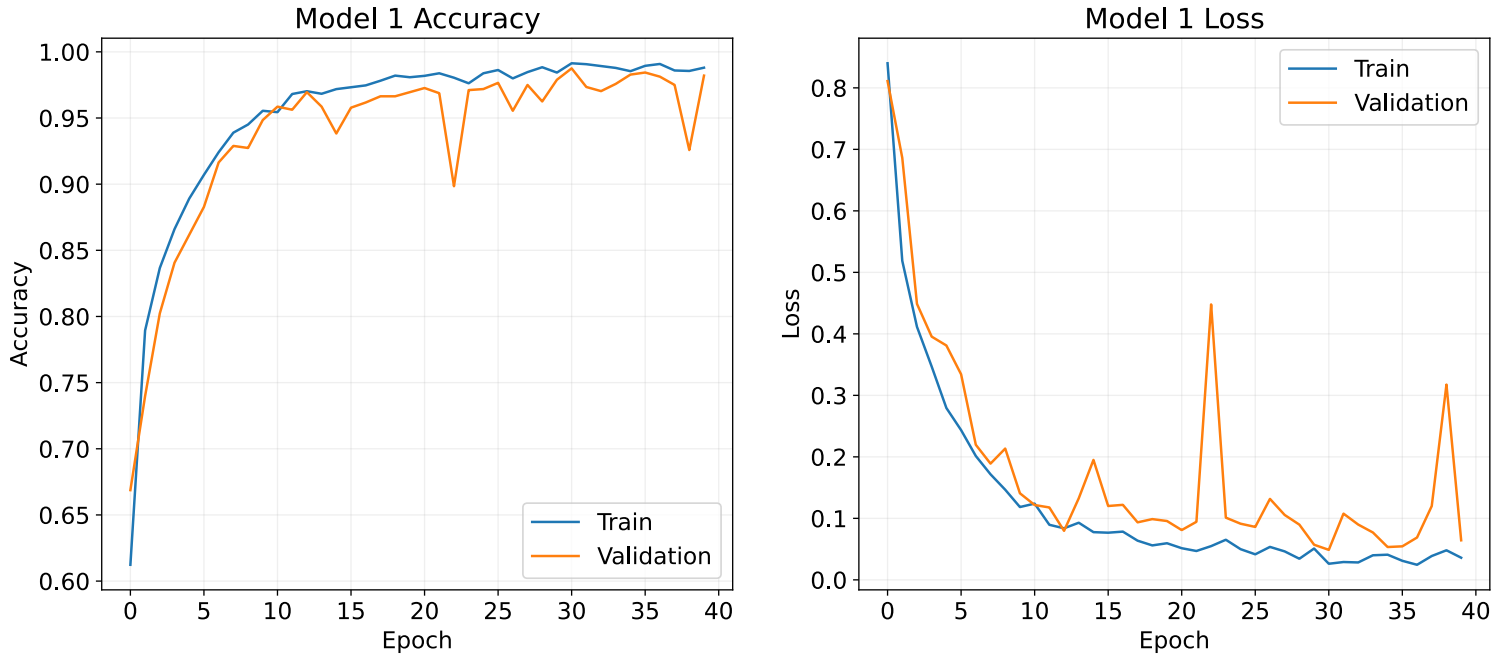
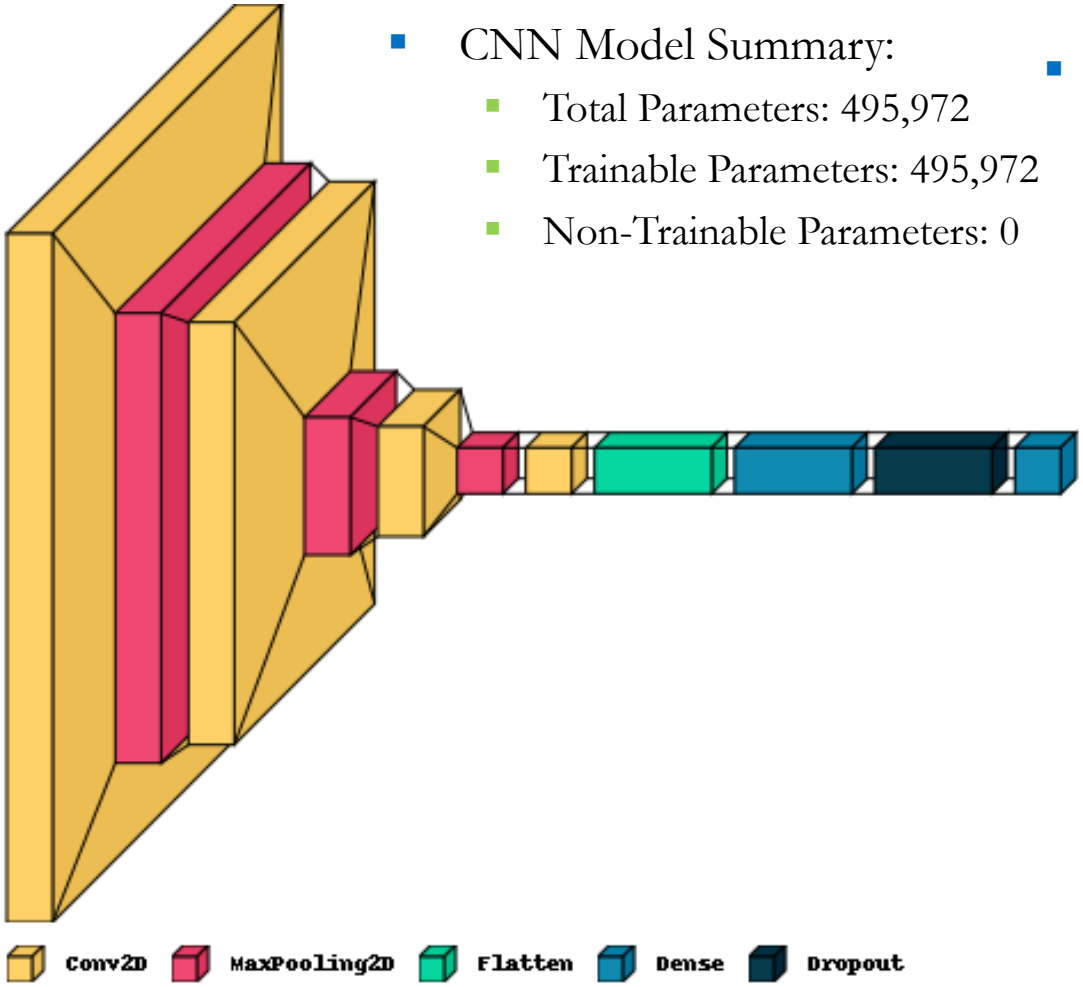


Figure 5: Initial model history metrics

Final Model - Architecture



- CNN Model Summary:
  - Total Parameters: 495,972
  - Trainable Parameters: 495,972
  - Non-Trainable Parameters: 0

- Layers:
  1. Conv2D (32 filters, (4,4))
  2. MaxPooling2D (3,3)
  3. Conv2D (64 filters, (4,4))
  4. MaxPooling2D (3,3)
  5. Conv2D (128 filters, (4,4))
  6. MaxPooling2D (3,3)
  7. Conv2D (128 filters, (3,3))
  8. Flatten
  9. Dense (512 neurons)
  10. Dropout (0.5)
  11. Dense (4 neurons, softmax)

Figure 6: Final Model Architecture Visualization

Final Model - Training Structure

- **The final model** uses the same architecture as the initial model but was trained differently with the following improvements:
  - The beta_1 parameter for the Adam optimizer was varied within the range [0.7, 0.99], and the beta_2 parameter was varied within the range [0.9, 0.9995], to find optimal values for faster convergence.
- **Optimizer**: Adam optimizer is utilized with a learning rate of 0.001, beta_1 set to 0.869, and beta_2 set to 0.995 for faster convergence during training.
- **Callbacks**: EarlyStopping is employed to halt training if the loss does not show significant improvement. ReduceLROnPlateau is used to reduce the learning rate if the validation loss plateaus, improving training efficiency.
- **Loss Function**: Categorical crossentropy is employed as the loss function, which is suitable for multiclass classification tasks.
- **Metrics**: The model's performance is evaluated using accuracy, measuring the proportion of correctly classified samples.

Final Model – Accuracy

- **Final Model:** Built from initial model
  - Parameter changes: beta_1=0.869, beta_2=0.995
  - Added model callbacks
  - Achieve an accuracy rate of 99.4%
  - It is fascinating to see such an improvement just from training adjustments!

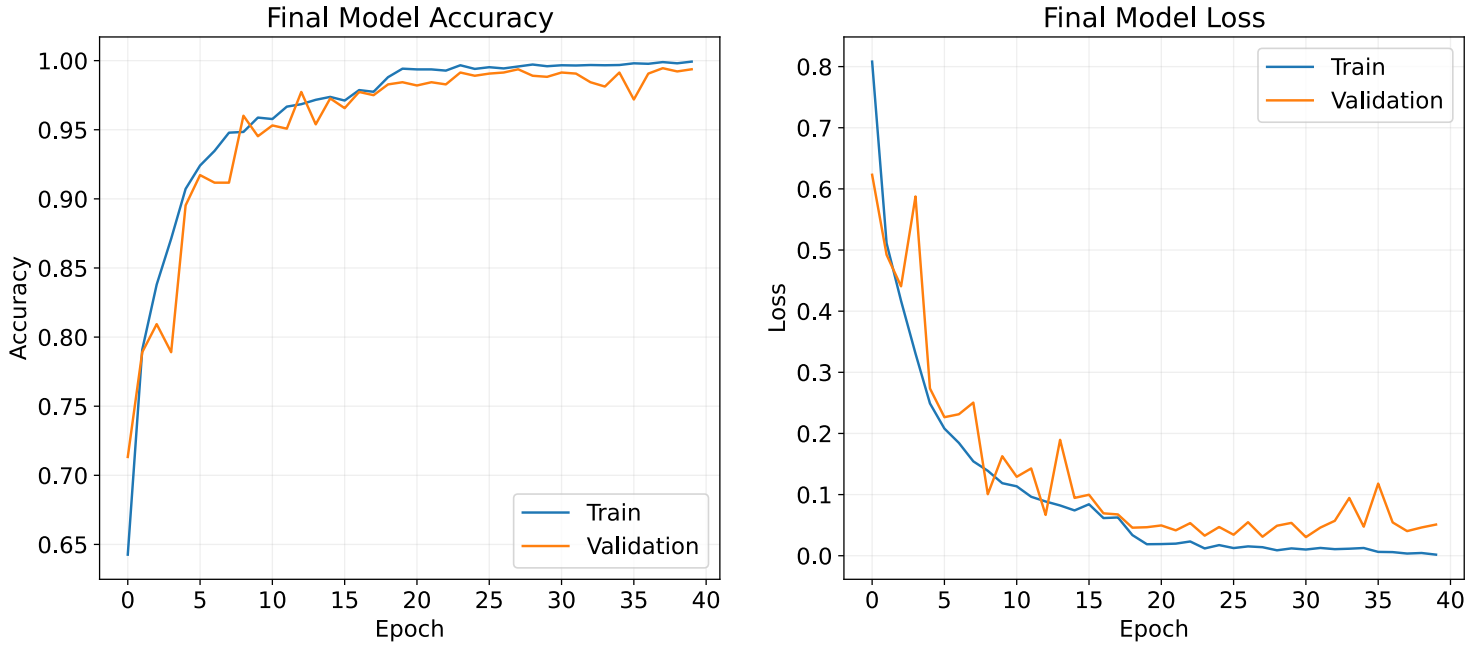


Figure 7: Final model history metrics

Final Model - Layers Visualization

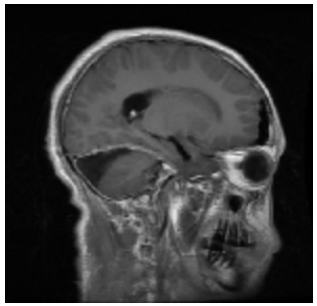


Figure 8: Input Image The original input image is shown here, serving as the starting point for the convolutional neural network (CNN) model.

Conv Layer 1 and Under a Max Pooling layer

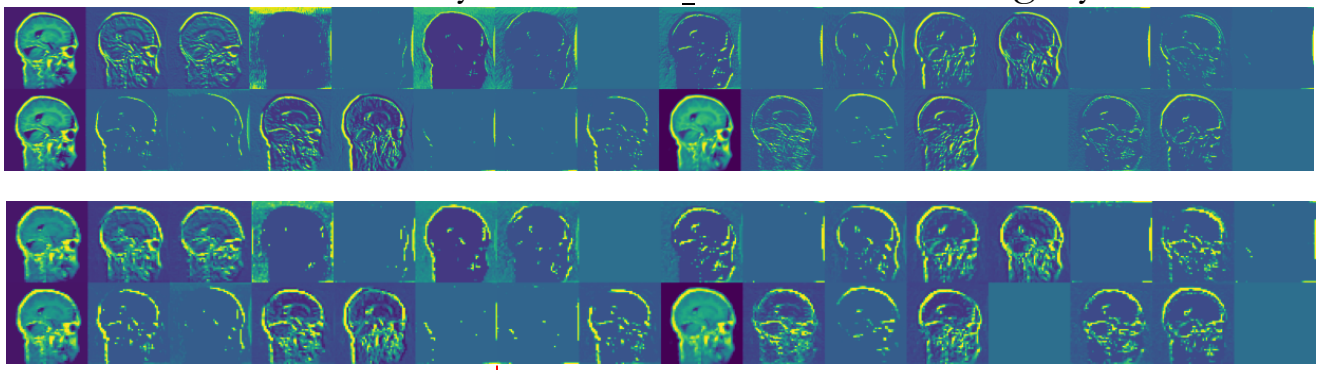


Figure 9: Conv Layer 1 + Max Pooling After passing through the first convolutional layer with 32 filters of size 4x4 and ReLU activation, the image undergoes max pooling (3x3). This extracts essential features and reduces spatial dimensions.

Conv Layer 2

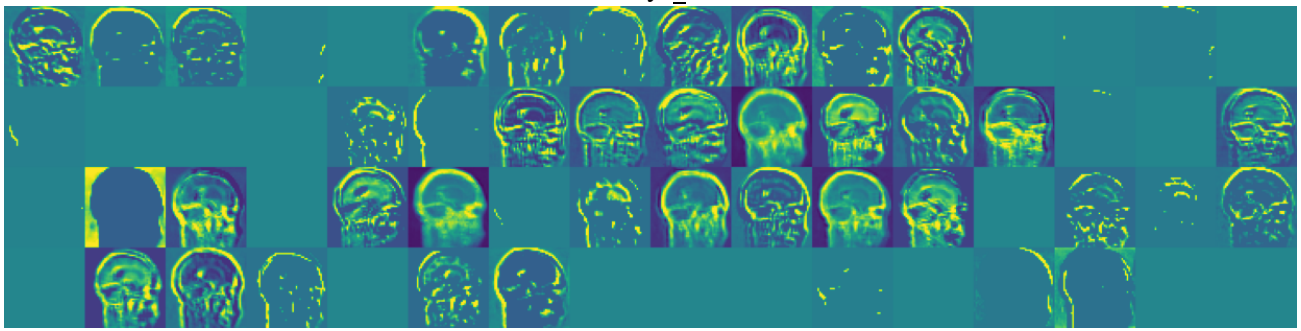


Figure 10: Conv Layer 2 The image displays the result after applying the second convolutional layer, which utilizes 64 filters of size 4x4 with ReLU activation. This layer refines learned features, contributing to the CNN model's hierarchical learning.

Performance Evaluation – Confusion Matrix

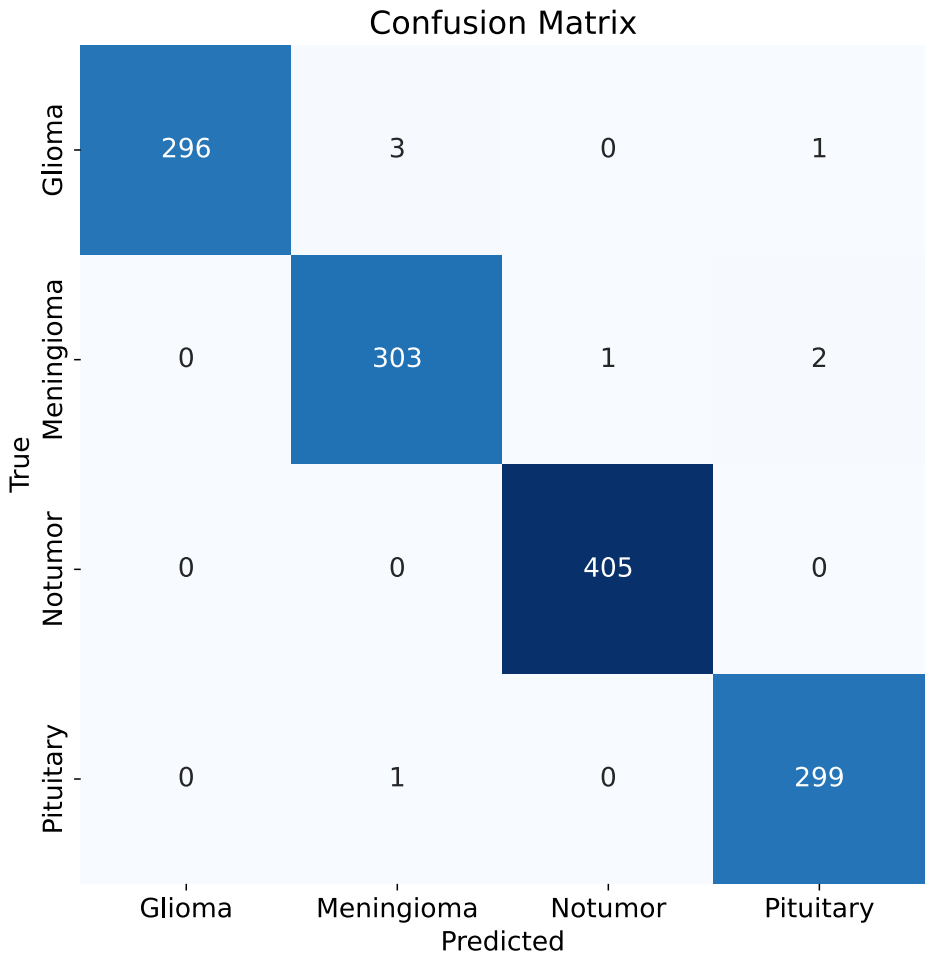


Figure 11: Confusion matrix of final model on test data. A total of eight images have been miss-classified.

Performance Evaluation – Metrics

- **Metrics**
  - TP (True Positives): Number of instances correctly classified as a specific class.
  - FP (False Positives): Number of instances incorrectly classified as a specific class, which do not actually belong to it.
  - FN (False Negatives): Number of instances belonging to a specific class but incorrectly classified as other classes.
- **Performance Evaluation**:
  - The model achieved perfect precision (1.000) in classifying Glioma images, indicating no false positives in this category.
  - No tumor classification had a high precision of 0.998, showing a low number of false positives.
  - The model performed consistently in distinguishing Meningioma images, with precision, score of 0.987.
  - Pituitary classification achieved high recall (0.997) and a commendable F1-score of 0.993.

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}$$

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}$$

Class	Precision	Recall	F1-Score
Glioma	1.000	0.987	0.993
Meningioma	0.987	0.990	0.989
No Tumor	0.998	1.000	0.999
Pituitary	0.990	0.997	0.993

Figure 12: Classification metrics for model's prediction on test data

Nine Random Samples From Final Model

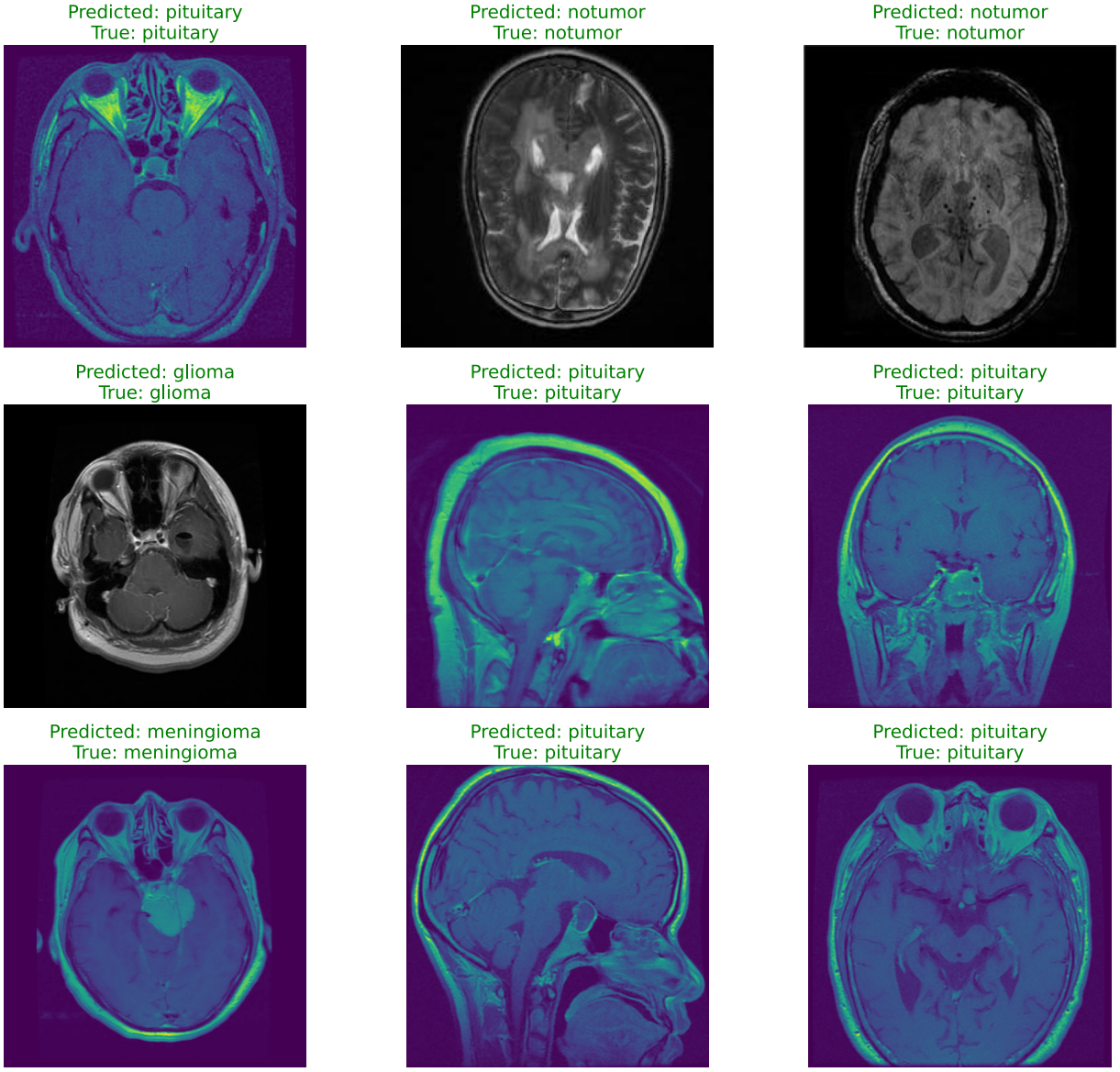


Figure 13: Model's predictions on ten random samples from test data

Miss-Classified MRI Images - Test Data

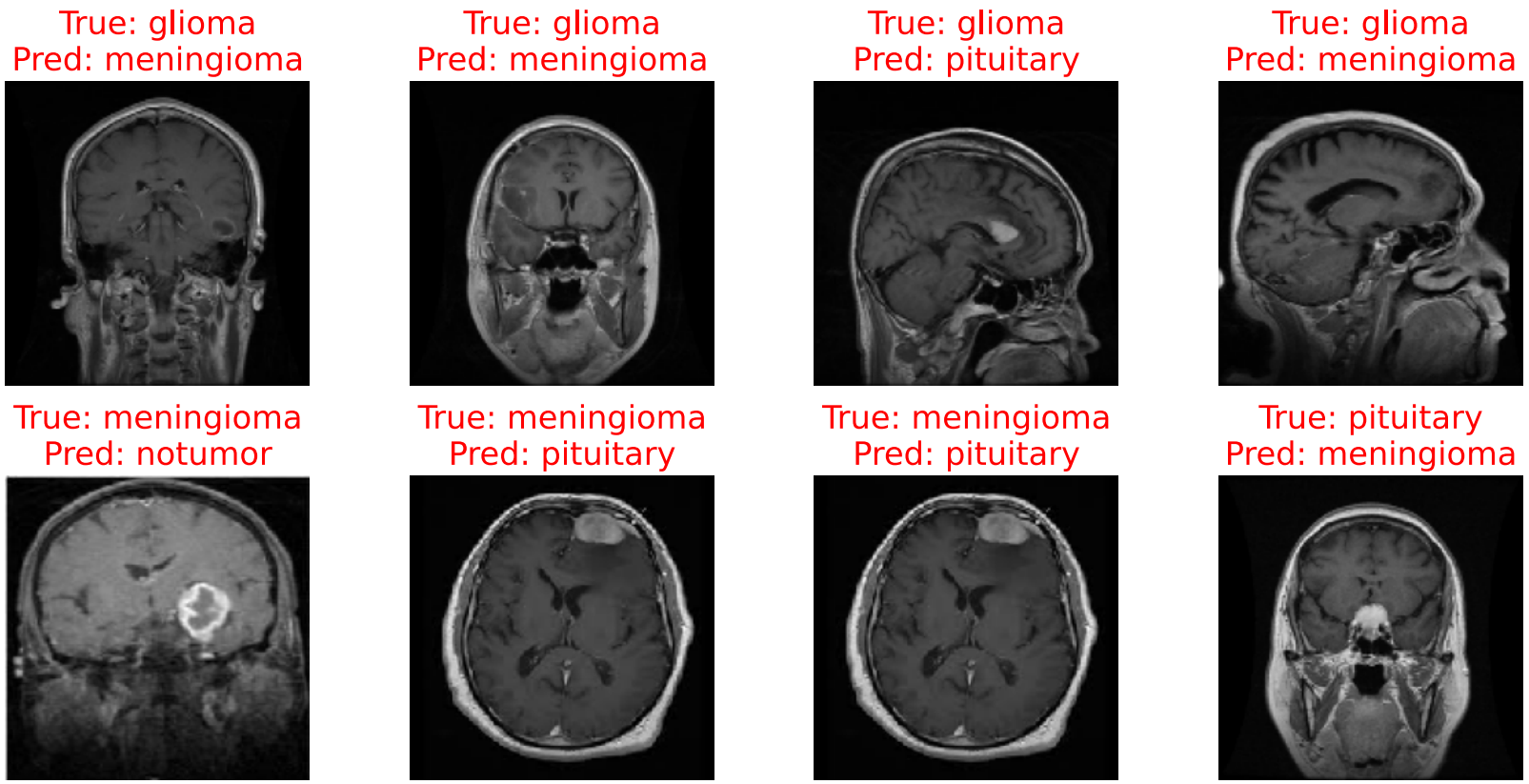


Figure 14: All miss-classified MRI images from predictions on the test data. We see an interesting miss-classification on the bottom left, there is clearly a tumor in the middle right of the image.

Conclusion

During this project we have successfully developed a robust Convolutional Neural Network model capable of accurately classifying MRI Brain Tumor images into four categories: Glioma, Meningioma, No Tumor, and Pituitary

- Highlights:
  - Utilized a multi-layered CNN architecture for feature extraction and classification.
  - Achieved an impressive accuracy rate of 99.4% on the test dataset.
  - Provided valuable insights through classification metrics and confusion matrix analysis.
- Significance:
  - The CNN model offers the potential for earlier and more precise brain tumor detection.
  - Enables informed decision-making and personalized treatment plans for patients at a faster rate.
  - Revolutionizes medical diagnostics to assists healthcare professionals in making accurate diagnoses of ill patients.

References

- [1] Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | upGrad blog. (n.d.). *Basic CNN architecture: Explaining 5 layers of Convolutional Neural Network*. upGrad blog. https://www.upgrad.com/blog/basic-cnn-architecture/