Project Report
Brain Tumor

Sani Abdullahi Sani

27 May 2024

1 Introduction

Brain tumors are collections or masses of abnormal cells in the brain. Due to the rigidity of the skull, any growth within this restricted space can lead to increased intracranial pressure, causing brain damage and potentially becoming life-threatening. Brain tumors can be either cancerous (malignant) or noncancerous (benign). Early detection and classification of brain tumors are critical for selecting the most appropriate treatment methods, which can significantly improve patient outcomes and survival rates.

In this project, we apply deep learning techniques to the detection and classification of brain tumors using Magnetic Resonance Imaging (MRI) scans. According to the World Health Organization (WHO), proper diagnosis involves detecting the presence of a tumor, identifying its location, and classifying it based on malignancy, grade, and type. Our objective is to develop multiple Convolutional Neural Networks (CNNs) to perform multi-task classification for detecting and classifying brain tumors in MRI images, with the goal of identifying the most effective model. This approach aims to streamline the diagnostic process, providing an automated tool that can assist radiologists and medical professionals.

2 Dataset

We sourced our dataset from Kaggle, a well-known platform for data science competitions and datasets. The dataset comprises T1-weighted contrast-enhanced MRI images, available in three different views: axial, coronal, and sagittal. The images are classified into four categories:

- Glioma
- Meningioma
- No Tumor
- Pituitary

The dataset includes:

- Training examples: 2296 images
- Validation examples: 574 images
- Testing examples: 394 images
- Total: 3264 images

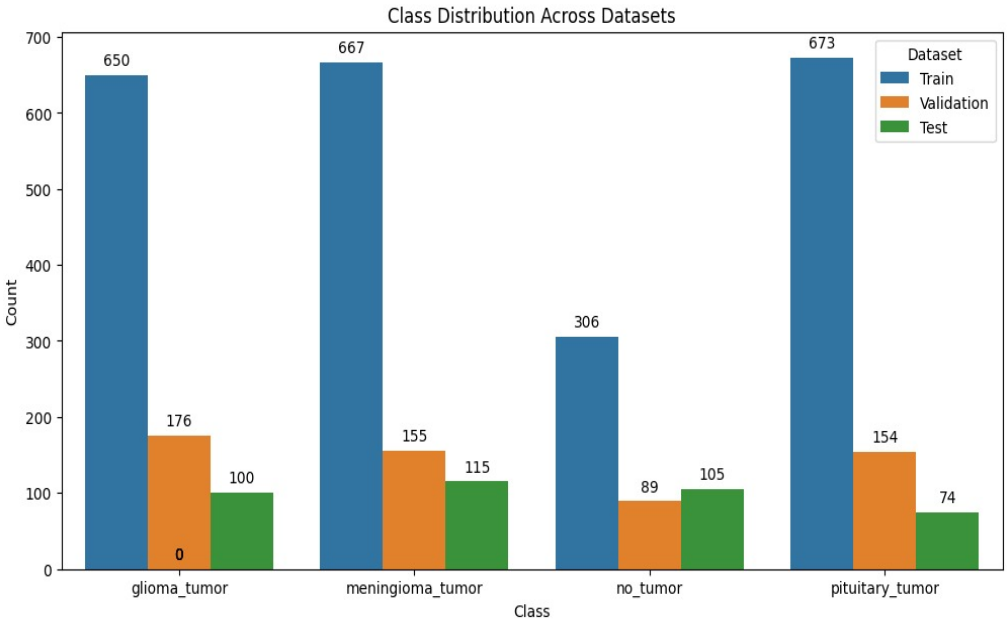


Figure 1: Class Distribution across Dataset

3 Preprocessing

In the pre-processing stage, we enhanced the images' uniformity and quality before feeding them into the neural networks. First, the original color images were converted to grayscale and then blurred using a Gaussian Blur to reduce noise. Next, a binary threshold was applied to the grayscale images to distinguish the brain region from the background. To clean the binary image, a series of erosions and dilations were performed.

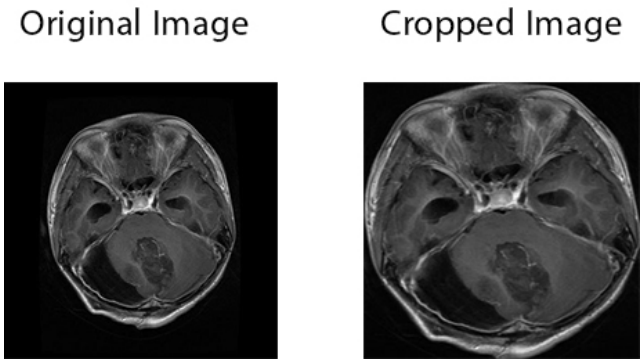


Figure 2: Image Cropping

Contours in the thresholded image were detected, and the largest contour, assumed to be the brain region, was selected. The extreme points of this contour (leftmost, rightmost, topmost, and bottommost) were identified to determine the bounding box for cropping. Finally, the original image was cropped using these extreme points to include only the brain region, effectively removing the background and retaining essential features for the neural networks. This pre-processing ensured high-quality, focused images for improved neural network performance.

4 Model Selection

For our project, we opted for a Convolutional Neural Network (CNN) as our model of choice. CNNs have demonstrated remarkable success in image classification tasks, making them particularly suitable for our dataset of MRI images. Unlike traditional neural networks, CNNs are designed to effectively capture spatial features from images through convolutional layers, enabling them to learn hierarchical representations directly from the pixel values. This characteristic makes CNNs well-suited for tasks involving image analysis and classification.

4.1 Model Architecture

Our CNN architecture is designed to progressively extract and refine features from input images. It consists of five convolutional layers followed by fully connected layers for classification. Each convolutional layer is augmented with batch normalization, ReLU activation, and max-pooling, facilitating effective feature extraction while reducing spatial dimensions. Furthermore, dropout layers are strategically placed to prevent overfitting and improve model generalization.

Activation functions, specifically ReLU, introduce non-linearity to capture complex relationships between features. Max-pooling layers downsample feature maps, reducing computational complexity while retaining essential information. These architectural elements work synergistically to enable the model to learn discriminative features from MRI images and classify brain tumors accurately.

5 Model Training

The training process generally involved optimizing the model parameters to minimize the Cross Entropy Loss while maximizing classification accuracy. To achieve that, We employed Stochastic Gradient Descent (SGD) with momentum as the optimizer, setting a learning rate of 0.001 and incorporating weight decay regularization (weight_decay=0.000001) to prevent excessive weight growth.

We conducted a training for 100 epochs. Within each epoch, the model iterated through the training dataset in mini-batches, updating weights based on computed gradients to converge towards an optimal solution. During each epoch, we computed both training loss and accuracy. The training loss was accumulated over mini-batches and normalized by the total number of training samples. Simultaneously, training accuracy was calculated by comparing model predictions with ground truth labels.

Following training, the model underwent validation on a separate dataset to assess the generalization capabilities. Validation loss and accuracy were computed similarly to training metrics. Throughout training, we monitored the evolution of training and validation losses, as well as training and validation accuracies, to evaluate the model's learning dynamics and identify overfitting or underfitting issues.

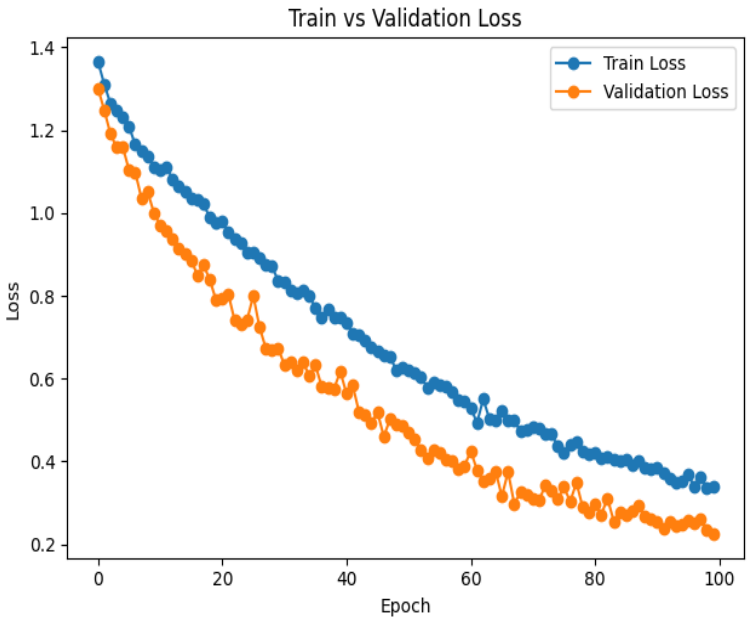


Figure 3: CNN 5 Layer Training and Validation Loss

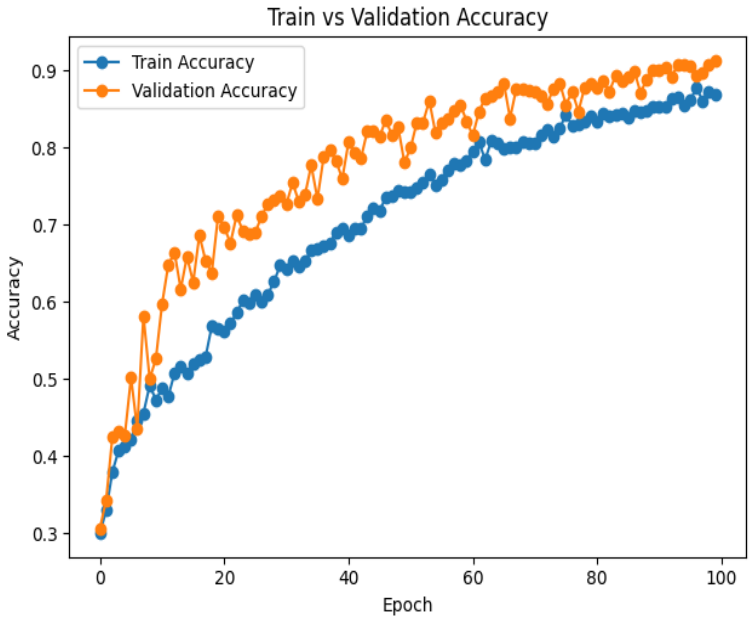


Figure 4: CNN 5 Layer Training and Validation Accuracy

6 Results and Analysis

After Training our model, The result shows the precision, recall, F1-score, and support for each class, as well as overall accuracy, macro average, and weighted average metrics below:

Table 1: Classification Report on Test Data - CNN 5 Layer

Class	**Precision**	**Recall**	**F1-Score**	**Support**	**Specificity**
Glioma Tumor	0.77	0.36	0.49	100	0.96
Meningioma Tumor	0.66	0.80	0.72	115	0.83
No Tumor	0.64	0.93	0.76	105	0.81
Pituitary Tumor	0.87	0.64	0.73	74	0.98
Accuracy			0.69	394	
Macro Avg	0.73	0.68	0.68	394	
Weighted Avg	0.72	0.69	0.68	394	

From the above classification report, we can see a varying performance across the brain tumor classes. our model apparently excels at detecting pituitary tumors, with a high precision of 0.87 and recall of 0.64, resulting in a decent F1-score of 0.73. Additionally, it performs well in identifying cases with no tumor, with a precision of 0.64 and recall of 0.93, yielding an F1-score of 0.76. However, the model struggles with detecting glioma tumors, with a precision of 0.77 but a low recall of 0.36, resulting in a poor F1-score of 0.49. Similarly, it faces challenges in detecting meningioma tumors, with a precision of 0.66 and recall of 0.80, yielding an F1-score of 0.72.

Furthermore, The overall performance of the model is moderate, with an accuracy of 0.69. The macro average and weighted average metrics indicate a similar performance across all classes, with a slight improvement on classes with more support (instances). However, the model's specificity is high for each class, indicating that it is good at correctly identifying cases without tumors. This is especially crucial in medical diagnosis, as false positives can lead to unnecessary treatment or further testing.

From this analysis, we see the need for improvement of our model. So we try to experiment with different architectures and hyperparameters, hoping to enhance the model's performance.

7 Experiments and Hyperparameter Tuning

7.1 Experiment 1

The first experiment we tried is using a 2-layer CNN model, during training, we achieved a low train loss of 0.0597 with a high train accuracy of 98.04%. This indicates that the model performed exceptionally well on the training data, demonstrating a strong ability to learn and generalize patterns from the training set. However, during validation, the model showed a slightly higher validation loss of 0.4957, accompanied by a validation accuracy of 89.90%. While the validation accuracy remains relatively high, the increase in validation loss compared to the train loss suggests a potential for overfitting or suboptimal generalization to unseen data.

Upon testing the model on the unseen test dataset, we observed a higher test loss of 3.1915, indicating that the model's performance decreased when applied to new, unseen data. Despite the increase in test loss, the test accuracy remained relatively high at 72.59%, suggesting that the model still performed reasonably well on the test dataset. However, the significant gap between the train and test losses could indicate a degree of overfitting, where the model may have memorized the training data rather than learned meaningful patterns that generalize well to unseen data. This highlights the importance of further optimizing the model architecture and training process to improve its generalization performance.

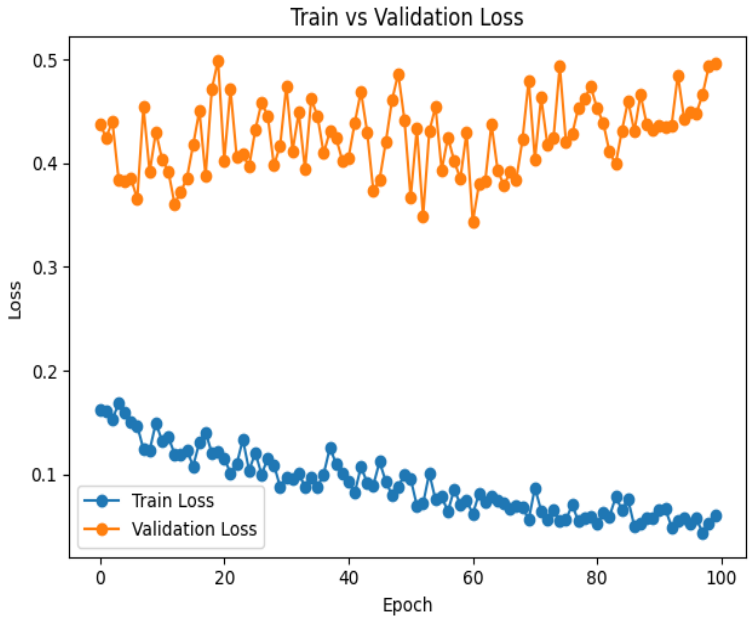


Figure 5: 2 Layer CNN Loss

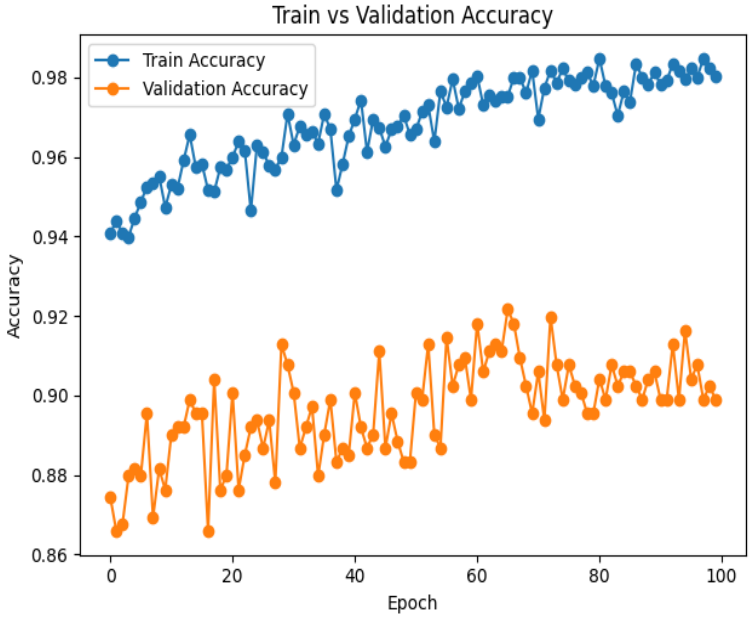


Figure 6: 2 Layer CNN Accuracy

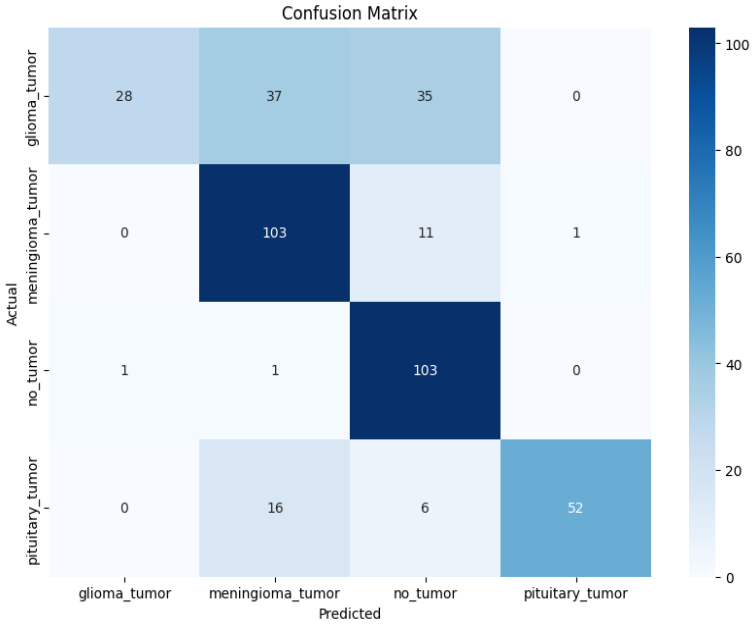


Figure 7: Confusion Matrix

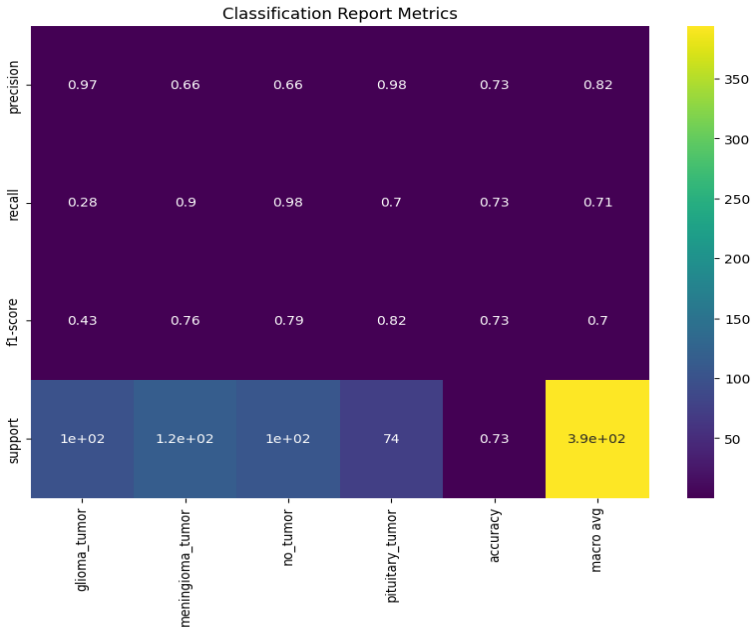


Figure 8: Classification Report Metrics

For that reason, we experimented with adjusting hyperparameters, notably increasing the number of layers to 10. For the 10-layer CNN model, the train loss is higher at 0.2236, indicating that the model had more difficulty fitting the training

data compared to the 2-layer CNN. Similarly, the train accuracy is lower at 91.90%, suggesting that the model's ability to correctly classify examples in the training set was not as strong as the 2-layer CNN.

During validation, the 10-layer CNN model achieved a slightly lower validation loss of 0.3859 compared to the 2-layer CNN, indicating better generalization performance. However, the validation accuracy remained similar at 90.07

When evaluated on the test dataset, the 10-layer CNN model obtained a test loss of 1.2143 and a test accuracy of 75.13%. Although the test loss is lower than that of the 2-layer CNN, the test accuracy is slightly higher, suggesting that the 10-layer CNN model achieved better performance on unseen data.

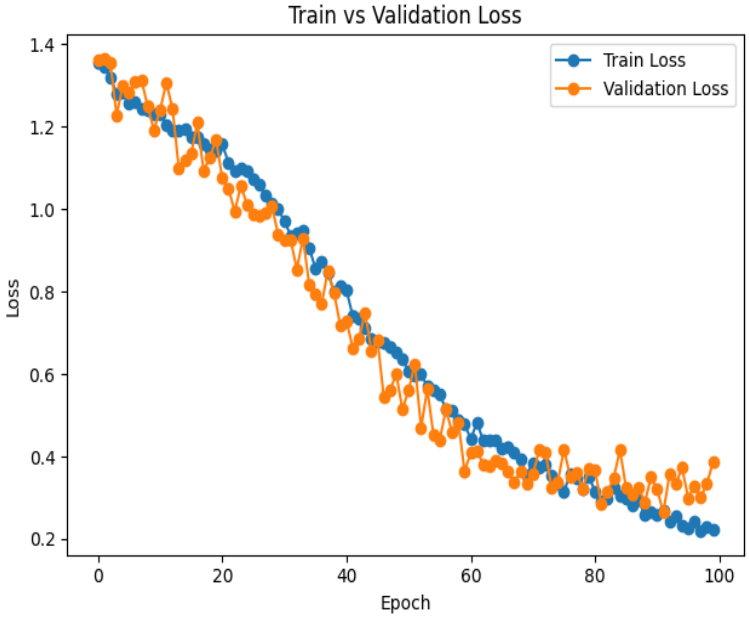


Figure 9: 10 Layer CNN Loss

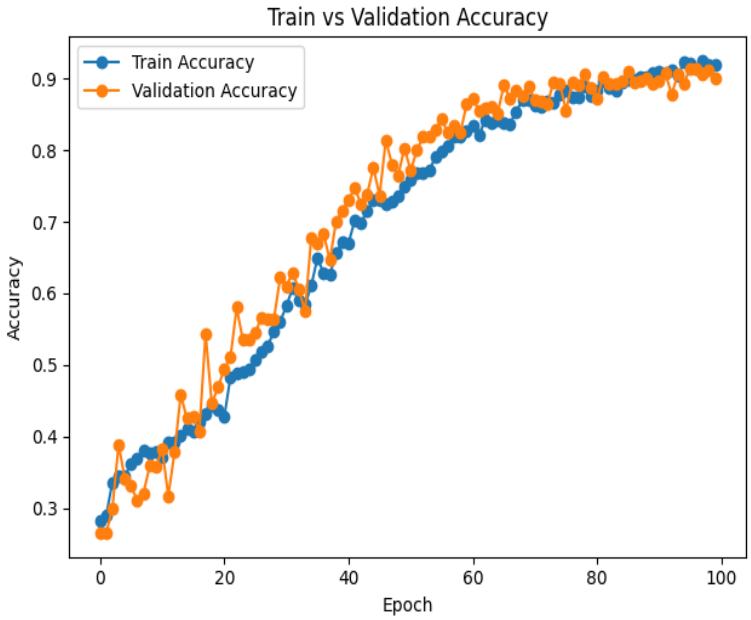


Figure 10: 10 Layer CNN Accuracy

7.2 Experiment 2 - Dilated CNN

The second experiment involved implementing a dilated CNN for comparison with the standard CNN. Dilated convolutions help capture more contextual information by expanding the receptive field without increasing the number of parameters. This can be particularly beneficial for medical imaging tasks, where capturing global context is crucial for accurate diagnosis.

For the experiment with the dilated CNN model, we trained a two-layer dilated convolutional neural network (CNN) to compare its performance with a standard CNN. The dilated CNN model included convolutional layers, each followed by batch normalization and ReLU activation to stabilize training and introduce non-linearity. Dropout layers were incorporated to prevent overfitting, with dropout rates set to 0.3 and 0.5 after the first and second convolutional layers, respectively. MaxPooling layers were used to reduce the spatial dimensions of the feature maps and decrease the computational load. Finally, fully connected layers with ReLU activations facilitated the final classification and learning of intricate patterns. The training was conducted over 100 epochs using the SGD optimizer with a learning rate of 0.01 and momentum of 0.9.

7.2.1 Dilated CNN - 2 Layers

For the 2-layer dilation model, we obtained a training loss of 0.193, with a training accuracy of 92.42%, and a validation loss of 0.340, with a validation accuracy of 87.63%. The test loss was 1.647, with a test accuracy of 75.89%.

Table 2: Classification Report for Test Data - 2D Dilated

Class	Precision	Recall	F1-Score	Support	Specificity
Glioma Tumor	0.86	0.36	0.51	100	0.98
Meningioma Tumor	0.69	0.92	0.79	115	0.83
No Tumor	0.73	1.00	0.84	105	0.87
Pituitary Tumor	0.96	0.70	0.81	74	0.99
Accuracy			0.76	394	
Macro Avg	0.81	0.75	0.74	394	
Weighted Avg	0.79	0.76	0.74	394	

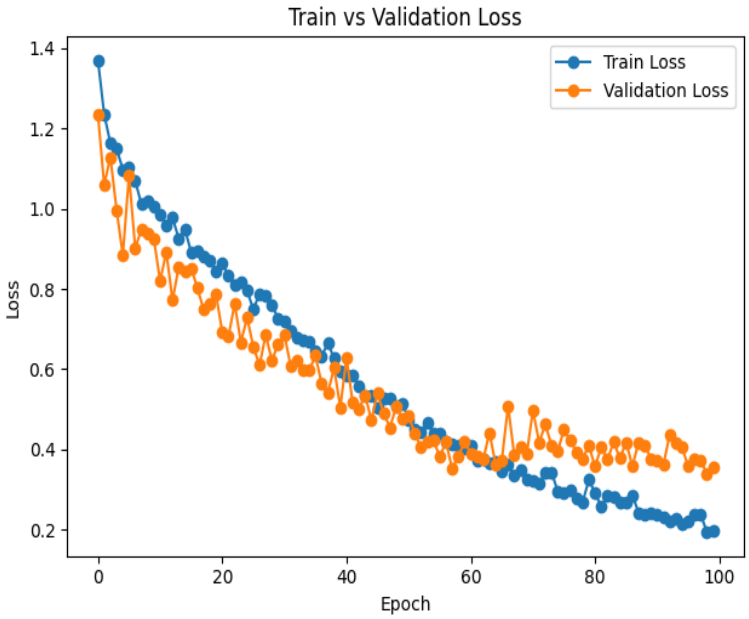


Figure 11: Train and Validation Loss - 2D Dilated

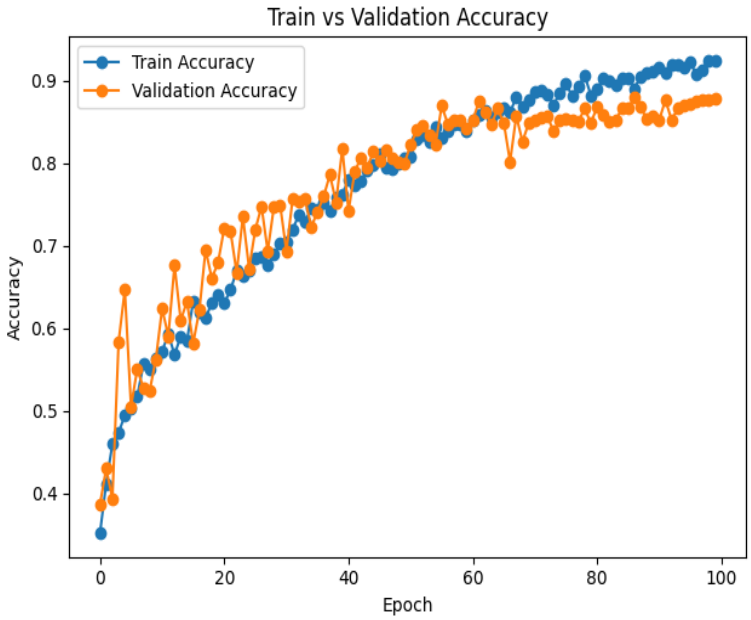


Figure 12: Train and Validation Accuracy - 2D Dilated

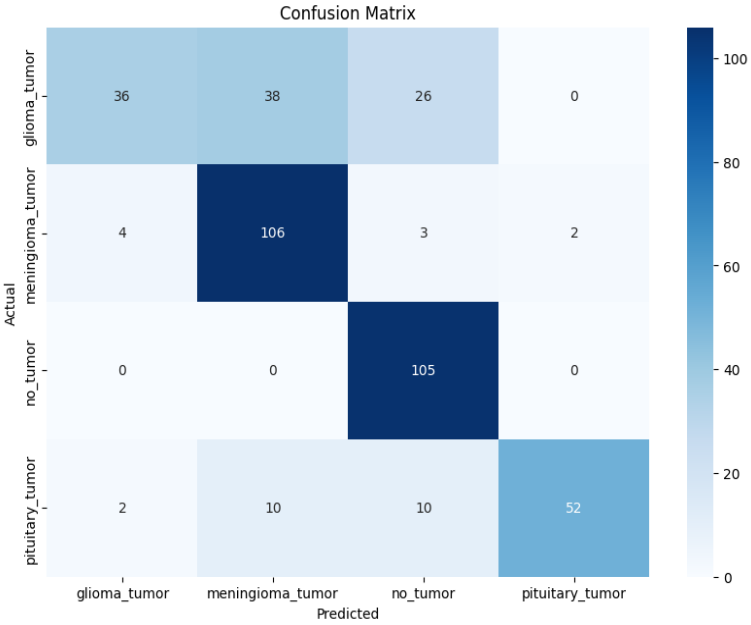


Figure 13: Confusion Matrix - 2D Dilated

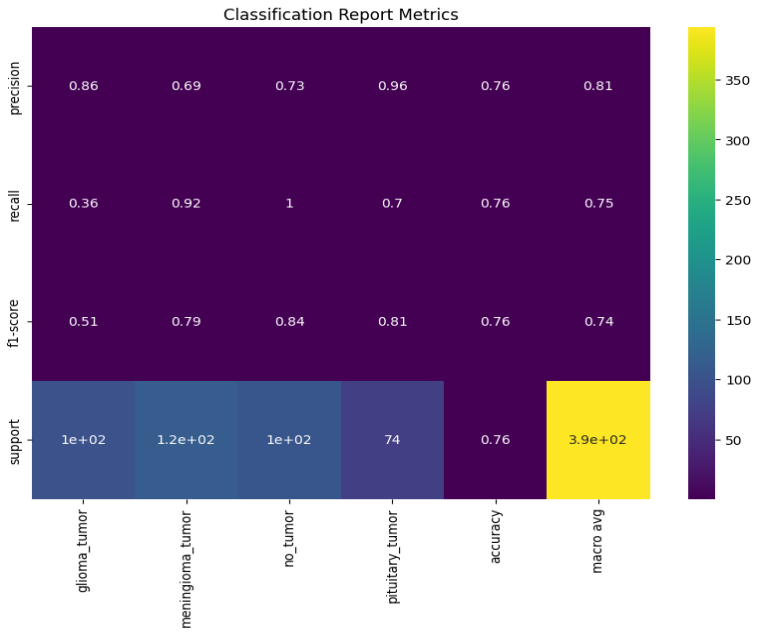


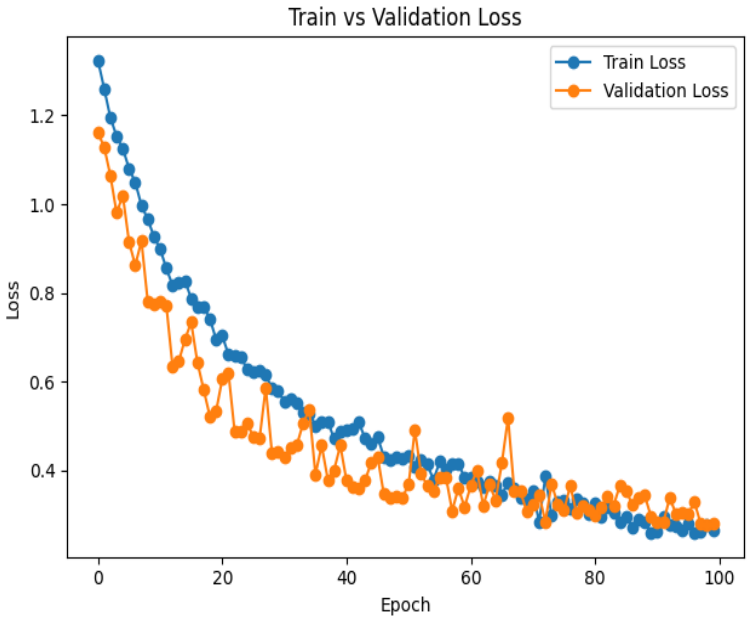
Figure 14: Classification Report Metrics - 2D Dilated

7.2.2 Dilated CNN - 5 Layers

For the 5-layer dilation model, the training loss increased slightly to 0.264, with a training accuracy of 90.42%. However, the validation loss decreased to 0.282, with a validation accuracy of 91.81%. The test loss was 1.189, with a test accuracy of 74.62%.

Table 3: Classification Report for Test Data - 5 Layers Dilated

Class	Precision	Recall	F1-Score	Support	Specificity
Glioma Tumor	0.79	0.38	0.51	100	0.97
Meningioma Tumor	0.65	0.88	0.75	115	0.81
No Tumor	0.74	0.96	0.84	105	0.88
Pituitary Tumor	0.98	0.73	0.84	74	1.00
Accuracy			0.75	394	
Macro Avg	0.79	0.74	0.73	394	
Weighted Avg	0.77	0.75	0.73	394	



H]

Figure 15: Train and Validation Loss - 5 Layers Dilated

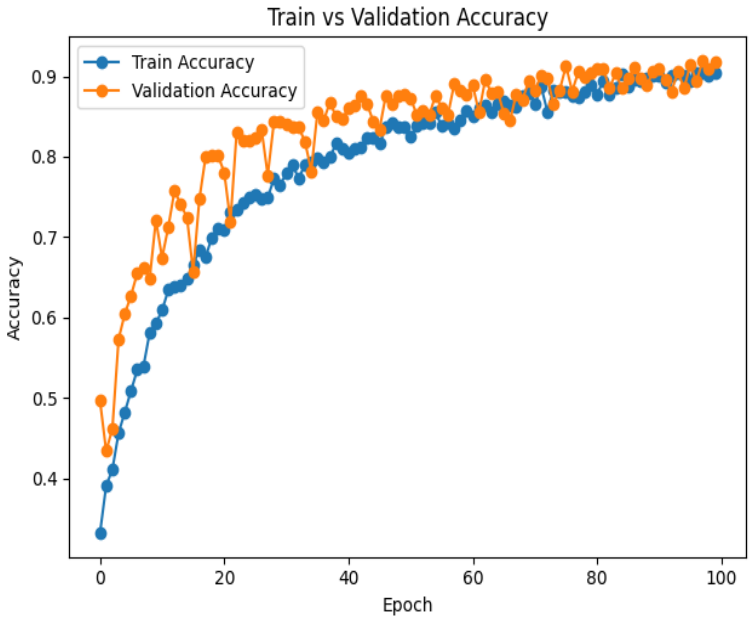


Figure 16: Train and Validation Accuracy - 5 Layers Dilated

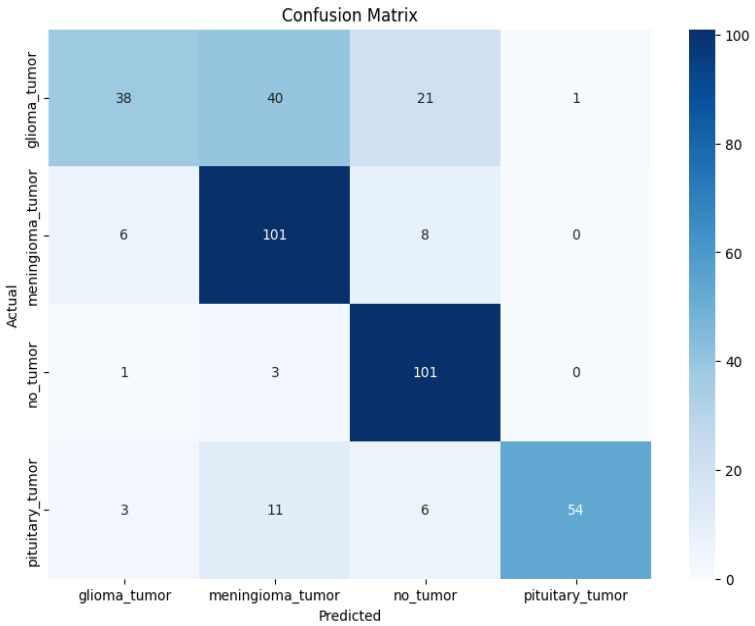


Figure 17: Confusion Matrix - 5 Layers Dilated

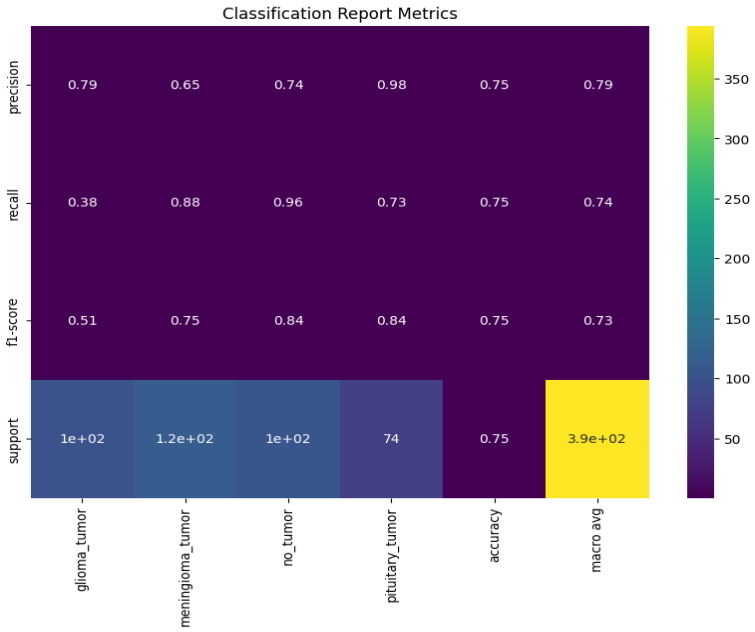


Figure 18: Classification Report Metrics - 5 Layers Dilated

7.2.3 Dilated CNN - 10 Layers

Finally, for the 10-layer dilation model, the training loss was 0.279, with a training accuracy of 90.03%. The validation loss further decreased to 0.272, with a validation accuracy of 91.29%. The test loss was 0.944, with a test accuracy of 72.34%.

Table 4: Classification Report for Test Data - 10 Layers Dilated

Class	Precision	Recall	F1-Score	Support	Specificity
Glioma Tumor	0.73	0.41	0.53	100	0.95
Meningioma Tumor	0.61	0.91	0.73	115	0.76
No Tumor	0.79	0.93	0.86	105	0.91
Pituitary Tumor	0.95	0.55	0.70	74	0.99
Accuracy			0.72	394	
Macro Avg	0.77	0.70	0.70	394	
Weighted Avg	0.75	0.72	0.71	394	

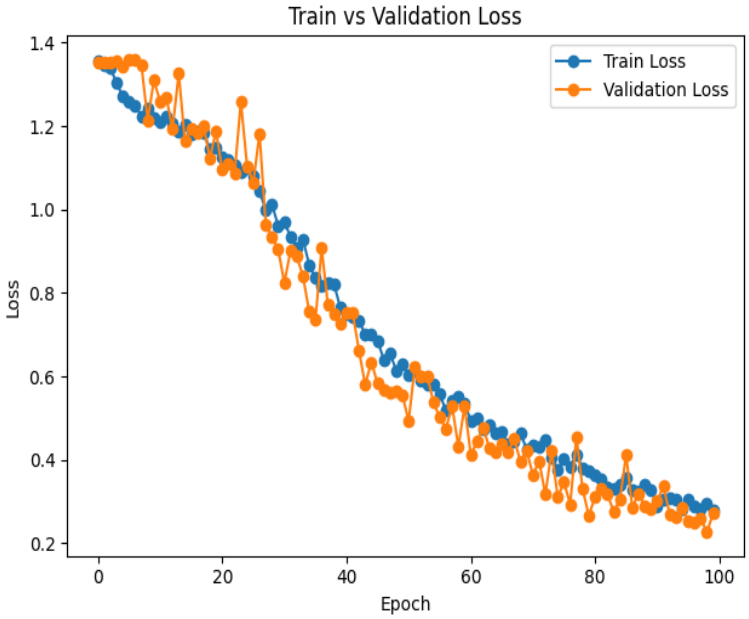


Figure 19: Train and Validation Loss - 10 Layers Dilated

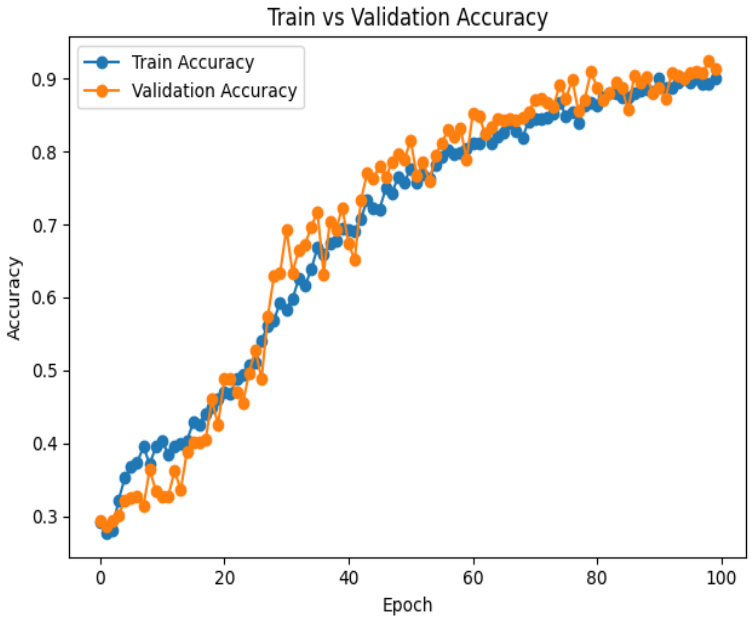


Figure 20: Train and Validation Accuracy - 10 Layers Dilated

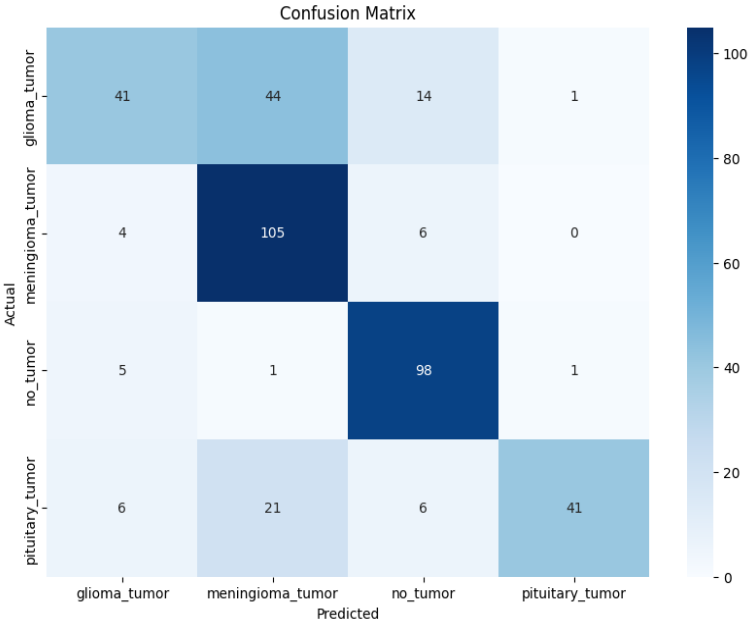


Figure 21: Confusion Matrix

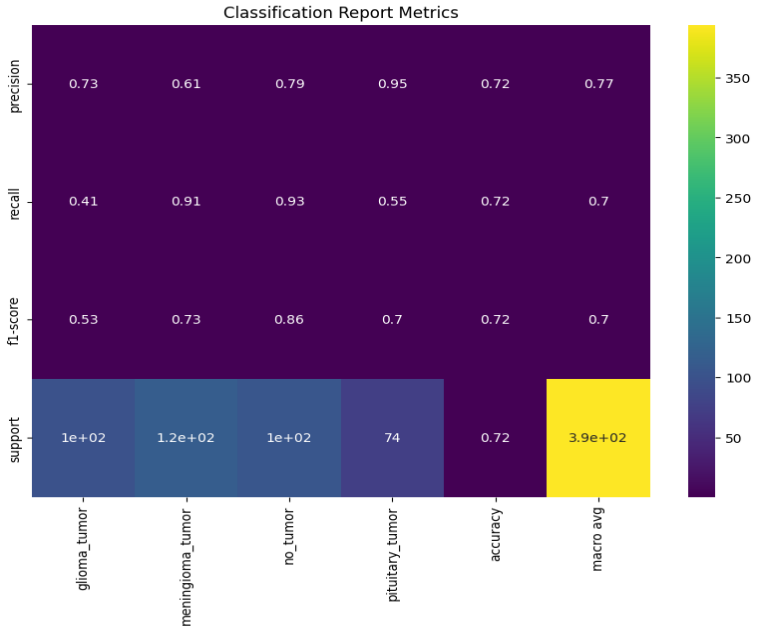


Figure 22: Classification Report Metrics - 10 Layers Dilated

Comparing these results to the standard CNN, we note that the dilation models generally exhibit comparable or slightly improved performance, particularly in terms of validation accuracy. However, the test accuracy varies marginally across the different dilation models, suggesting that increasing the number of layers in the dilation models may not significantly enhance performance beyond a certain point.

7.3 Experiment 3 - Data Redistribution

In another experimental phase, we explored the impact of data preprocessing by combining the dataset, splitting it again, and training without preprocessing. The combined dataset was partitioned into training, validation, and test sets with a ratio of 64:16:20, as depicted in Figure 23. We proceeded to train CNN models with 2, 5, and 10 layers to assess the influence of preprocessing on model performance.

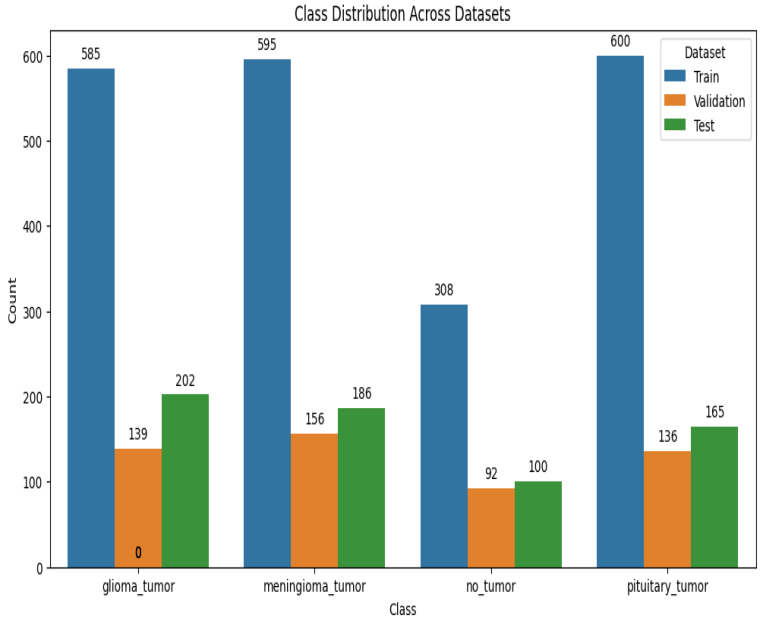


Figure 23: Data Redistribution

Observations:

- For the 2-layer CNN model, we observed a training loss of 0.0021 and a training accuracy of 99.90%. The validation loss was 0.595, with a validation accuracy of 89.10%. The test loss decreased to 0.468, with a test accuracy of 91.58%.
- For the 5-layer CNN model, we obtained a training loss of 0.0755 and a training accuracy of 97.56%. The validation loss decreased to 0.431, with a validation accuracy of 90.06%. Notably, the test loss reduced significantly to 0.389, accompanied by a test accuracy of 90.51%.
- Moving to the 10-layer CNN model, the training loss increased to 0.291, with a training accuracy of 88.03%. The validation loss rose to 1.219, with a validation accuracy of 69.02%. However, the test loss decreased slightly to 0.943, with a test accuracy of 73.05%.

The observation that unprocessed data and also fewer layers somehow leads to better and faster results using the two layer is somehow intriguing, and perhaps suggests that the preprocessing steps might be counterproductive in this scenario. This can happen due to a variety of reasons, including overfitting, or loss of important features.

Below are tables with results:

Table 5: Classification Report for Test Data - 2 Layers Unprocessed

Class	Precision	Recall	F1-Score	Support	Specificity
Glioma Tumor	0.92	0.90	0.91	183	0.97
Meningioma Tumor	0.90	0.91	0.90	191	0.96
No Tumor	0.89	0.83	0.86	104	0.98
Pituitary Tumor	0.95	0.99	0.97	175	0.98
Accuracy			0.92	653	
Macro Avg	0.91	0.91	0.91	653	
Weighted Avg	0.92	0.92	0.92	653	

Table 6: Classification Report for Test Data - 5 Layers Unprocessed

Class	Precision	Recall	F1-Score	Support	Specificity
Glioma Tumor	0.90	0.89	0.90	202	0.96
Meningioma Tumor	0.86	0.87	0.86	186	0.94
No Tumor	0.92	0.92	0.92	100	0.99
Pituitary Tumor	0.95	0.96	0.95	165	0.98
Accuracy			0.91	653	
Macro Avg	0.91	0.91	0.91	653	
Weighted Avg	0.90	0.91	0.90	653	

Table 7: Classification Report for Test Data - 10 Layers Unprocessed

Class	Precision	Recall	F1-Score	Support	Specificity
Glioma Tumor	0.62	0.88	0.72	187	0.78
Meningioma Tumor	0.86	0.36	0.51	194	0.98
No Tumor	0.82	0.75	0.78	97	0.97
Pituitary Tumor	0.78	0.97	0.87	175	0.90
Accuracy			0.73	653	
Macro Avg	0.77	0.74	0.72	653	
Weighted Avg	0.77	0.73	0.71	653	

8 Conclusion

In this project, we developed a Convolutional Neural Network (CNN) model to classify brain MRI images into four categories: glioma tumor, meningioma tumor, no tumor, and pituitary tumor, utilizing T1-weighted contrast-enhanced MRI images from Kaggle. Our exploration encompassed variations in model architecture, dataset preprocessing, and hyperparameter tuning.

Initially, we noted that the two-layer CNN, without preprocessing, achieved commendable accuracy, particularly in distinguishing between glioma and meningioma tumors. This highlights the potential efficacy of simple architectures

in extracting relevant features from raw image data. However, compared to models trained on preprocessed data or with deeper architectures, its accuracy fell short, underscoring the importance of preprocessing techniques in enhancing model performance. Techniques such as normalization and augmentation help mitigate noise and variability, leading to more robust models.

Exploring deeper architectures, like five and ten-layer CNNs, revealed varying performance levels. While deeper networks showed accuracy improvements over the two-layer CNN, the ten-layer CNN did not exhibit significant improvement compared to the five-layer CNN. This suggests that increasing model complexity beyond a certain point may not necessarily yield proportional gains in performance, emphasizing the need to balance complexity with computational efficiency and interpretability.

Furthermore, our investigation into dilated CNNs demonstrated their effectiveness in capturing contextual information, particularly in medical imaging tasks requiring global context for accurate diagnosis. The dilated CNN model showed competitive performance, achieving high accuracy and specificity across different tumor classes.

Finally, our findings underscore the significance of considering various factors, such as dataset characteristics, model architecture, and preprocessing techniques, in developing effective CNN models for medical image classification. By carefully optimizing these factors, robust models capable of accurately classifying medical images can be built, thereby facilitating more accurate diagnoses and treatment decisions in clinical settings.