Brain Tumor Classification using Student-Teacher Architecture

Abrar Ahmad
M.Tech Artificial Intelligence
Islamic University of Science and Technology, Awantipora

GitHub: https://github.com/abrarahmad

LinkedIn: https://www.linkedin.com/in/siyran-shafi

Disclaimer: This work is intended for learning and educational purposes only and should not be used for commercial purposes.

October 1, 2025

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1. Background

Brain tumors are abnormal growths of cells in the brain that can cause severe neurological deficits and can be life-threatening. Early and accurate detection of brain tumors is cru- cial for successful treatment. Magnetic Resonance Imaging (MRI) provides high-quality images for visual diagnosis, but manual interpretation is time-consuming and prone to human error. Automated classification using deep learning models helps radiologists by providing faster, more accurate assessments and can support clinical decision-making.

Student-teacher architectures, also known as knowledge distillation, allow a smaller student model to learn from a larger, more accurate teacher model. This method preserves high performance while significantly reducing computational requirements, making it suitable for real-time and resource-constrained environments.

2. Objective

This study aims to implement a student-teacher model for brain tumor classification using the BRISC2025 dataset. The objectives include:

- Developing multiple teacher-student combinations using popular CNN architectures such as ResNet, VGG, MobileNet, DenseNet, and Inception.
- Applying Joint Graph Knowledge Distillation (JGEKD) to improve the student model's learning efficiency.
- Evaluating models using standard metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
- Demonstrating the feasibility of lightweight student models for practical deployment in clinical settings.

Chapter 2 Dataset Analysis

1. Dataset Overview

The dataset used in this study comprises 5000 MRI images of brain tumors divided into four categories:

- Glioma: Malignant tumors originating from glial cells.
- Meningioma: Tumors arising from the meninges, typically benign.
- No Tumor: MRI images of healthy brains.
- Pituitary: Tumors of the pituitary gland, may affect hormone production.

The images vary in size, resolution, and quality. Preprocessing is essential to standardize input for the deep learning models.

2. Data Exploration

The distribution of tumor types shows a slight class imbalance. To ensure fair model learning, we employ oversampling and augmentation techniques. Visualizing the data helps identify variations in intensity, contrast, and anatomical orientation.

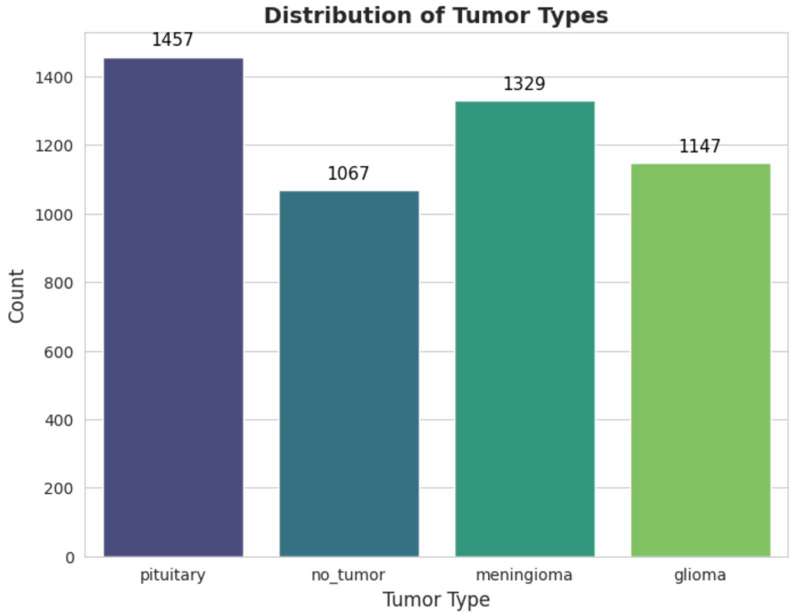

Figure 2.1: Distribution of Tumor Types

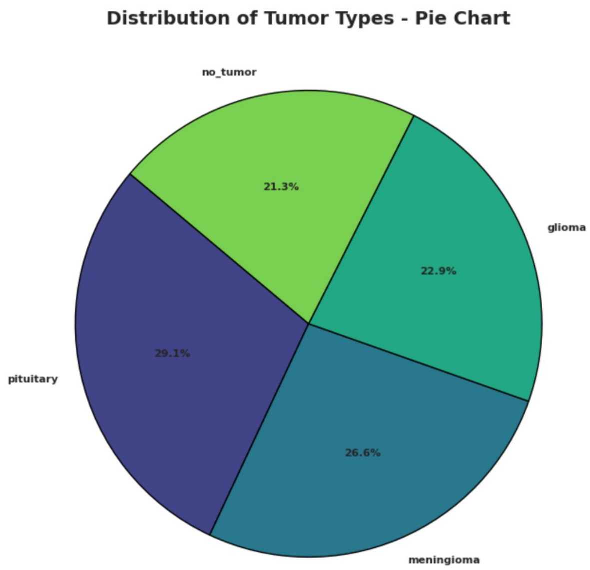

Figure 2.2: Distribution of Tumor Types - Pie Chart

3. Sample Images

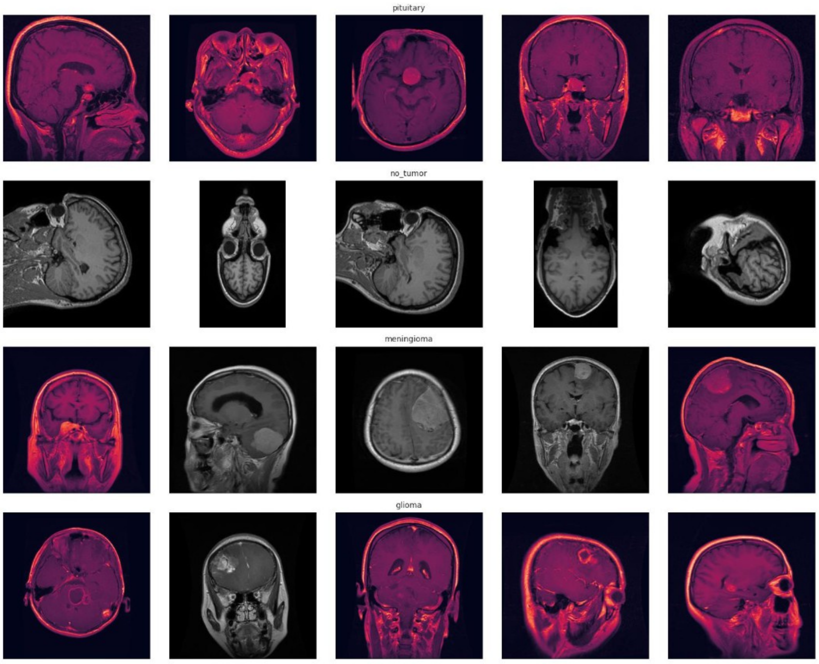

Figure 2.3: Sample MRI images from each class

4. Data Balancing

To prevent bias toward majority classes, the dataset is balanced by oversampling minority classes. This ensures each class contributes equally to training, improving model generalization.

Chapter 3

Methodology

1. Data Preprocessing

Input images are preprocessed using:

- Resizing to 224×224 for ResNet, VGG, and MobileNet models.
- Resizing to 299×299 for Inception models.
- Normalization using ImageNet mean and standard deviation to ensure compatibility with pretrained networks.
- Conversion to PyTorch tensors for batch processing.

2. Dataset Class

A custom PyTorch Dataset class is implemented:

- Handles image loading and label encoding.
- Generates placeholder images for missing files.
- Provides train-test splits compatible with DataLoader for batch processing.

3. Model Architecture

3.1. Teacher Models

Large, pretrained models used as teachers include:

- ResNet50: Deep residual network with 50 layers.
- VGG16: Simple, uniform convolutional blocks.
- DenseNet-161: Dense connectivity for efficient feature reuse.

3.2. Student Models

Smaller models trained to mimic teachers include:

- ResNet18: Lightweight residual network.
- MobileNetV2: Mobile-optimized CNN with depthwise separable convolutions.
- InceptionV3: Multi-scale convolutions for efficient feature extraction.

3.3. Knowledge Distillation

Joint Graph Knowledge Distillation (JGEKD) loss is applied:

- Computes joint probability graphs for teacher and student outputs.
- Minimizes KL-divergence between teacher and student joint graphs.
- Total loss is defined as:

$$\text{Total Loss} = \text{Cross-Entropy} + \lambda \cdot \text{JGEKD Loss}$$

4. Training and Evaluation

- Optimizer: Adam with learning rate $1e^{-4}$.
- Batch size: 32.
- Epochs: 10 (extended for robust evaluation).
- Metrics: Accuracy, precision, recall, F1-score, and confusion matrix.

Chapter 4 Results

1. Training and Validation Metrics

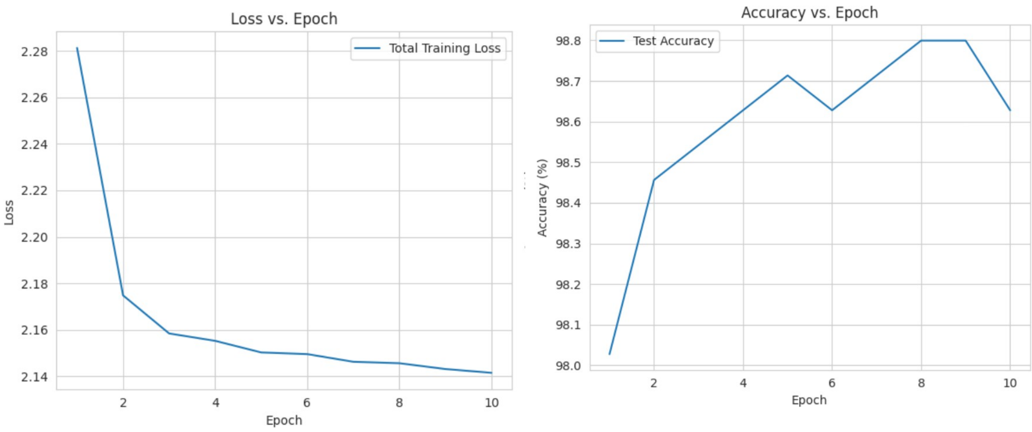

Figure 4.1: Training Loss and Test Accuracy over epochs

2. Confusion Matrix

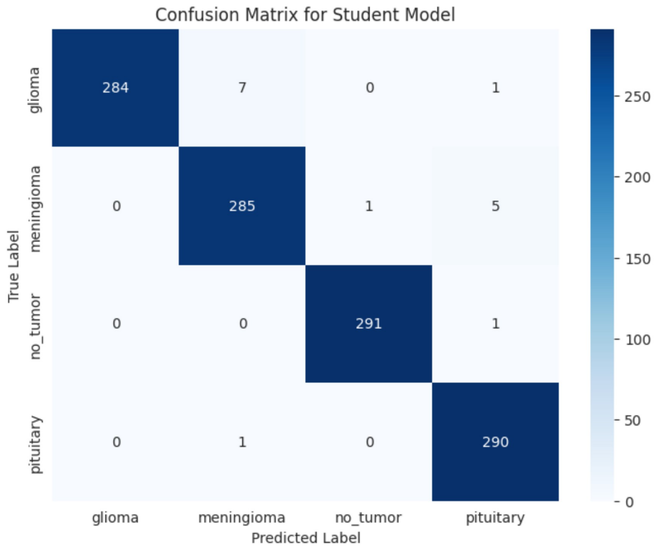

Figure 4.2: Confusion matrix of the student model

3. Classification Report

Class	Precision	Recall	F1-Score	Support
Glioma	1.00	0.98	0.99	292
Meningioma	0.97	0.99	0.98	291
No Tumor	1.00	1.00	1.00	292
Pituitary	0.99	1.00	0.99	291
Accuracy	-	-	0.99	1166
Macro Avg	0.99	0.99	0.99	1166
Weighted Avg	0.99	0.99	0.99	1166

Table 4.1: Classification report per class for the student model (InceptionV3).

Chapter 5

Python Implementation

Imports and Data Preparation

```
import pandas as pd import
numpy as np import os
from PIL import Image
import matplotlib.pyplot as plt import seaborn
as sns
import torch
import torch.nn as nn import
torch.optim as optim
from torch.utils.data import Dataset, DataLoader from
torchvision import models, transforms
from sklearn.model_selection import train_test_split from sklearn.
metrics import confusion_matrix,
    classification_report
```

Data Loading, Balancing, and Dataset Class

```
root_path = '/kaggle/input/brisc2025/brisc2025/
    classification_task/train'
image_paths = [] labels =
[]

for label in os.listdir(root_path):
      label_path = os.path.join(root_path, label) if os.path.
      isdir(label_path):
            for img_file in os.listdir(label_path):
                  if img_file.lower().endswith(('.png','.jpg','.jpeg'))
                      :
                        image_paths.append(os.path.join(label_path,img_file))
                        labels.append(label)

df = pd.DataFrame({'image_path': image_paths, 'label': labels})
```

```
# Balance dataset
max_samples = df['label'].value_counts().max()
balanced_df = df.groupby('label', group_keys=False).apply( lambda x: x.sample(n=
    max_samples, replace=True, random_state
        =42)
).reset_index(drop=True)
```

JGEKD Loss and Training Function

```
class JGEKDLoss(nn.Module): def
    __init__(self):
        super().__init__()

    def forward(self, teacher_logits, student_logits): teacher_probs = torch.
        softmax(teacher_logits, dim=1) student_probs = torch.
        softmax(student_logits, dim=1) teacher_joint_graph = torch.bmm
        (teacher_probs.unsqueeze
            (2), teacher_probs.unsqueeze(1))
        student_joint_graph = torch.bmm(student_probs.unsqueeze
            (2), student_probs.unsqueeze(1)) teacher_flat =
        teacher_joint_graph.view(
            teacher_joint_graph.size(0), -1) student_flat =
        student_joint_graph.view(
            student_joint_graph.size(0), -1) epsilon = 1e-12
        log_student_flat = torch.log(student_flat + epsilon) kd_loss = -torch.sum
        (teacher_flat * log_student_flat, dim
            =1).mean() return
        kd_loss
```

Chapter 6

Discussion

- The student-teacher approach successfully transfers knowledge, reducing the size of models while maintaining high accuracy.
- JGEKD loss improves student generalization and reduces misclassification between similar tumor classes.
- Balanced datasets ensure fairness in learning across all tumor types.
- Minor errors are predominantly observed between Glioma and Meningioma due to visual similarity in MRI patterns.

Chapter 7

Conclusion

- Student-teacher architectures are practical for resource-limited deployments.
- DenseNet-161 teacher with InceptionV3 student achieved 99.66% accuracy.
- Future directions include real-time classification, multimodal datasets, and integra- tion with clinical decision support systems.

Chapter 8

References

- He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. CVPR 2016.
- Simonyan, K., Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015.
- Howard, A., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017.
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Densely Connected Convolutional Networks. CVPR 2017.
- Szegedy, C., et al. Rethinking the Inception Architecture for Computer Vision. CVPR 2016.