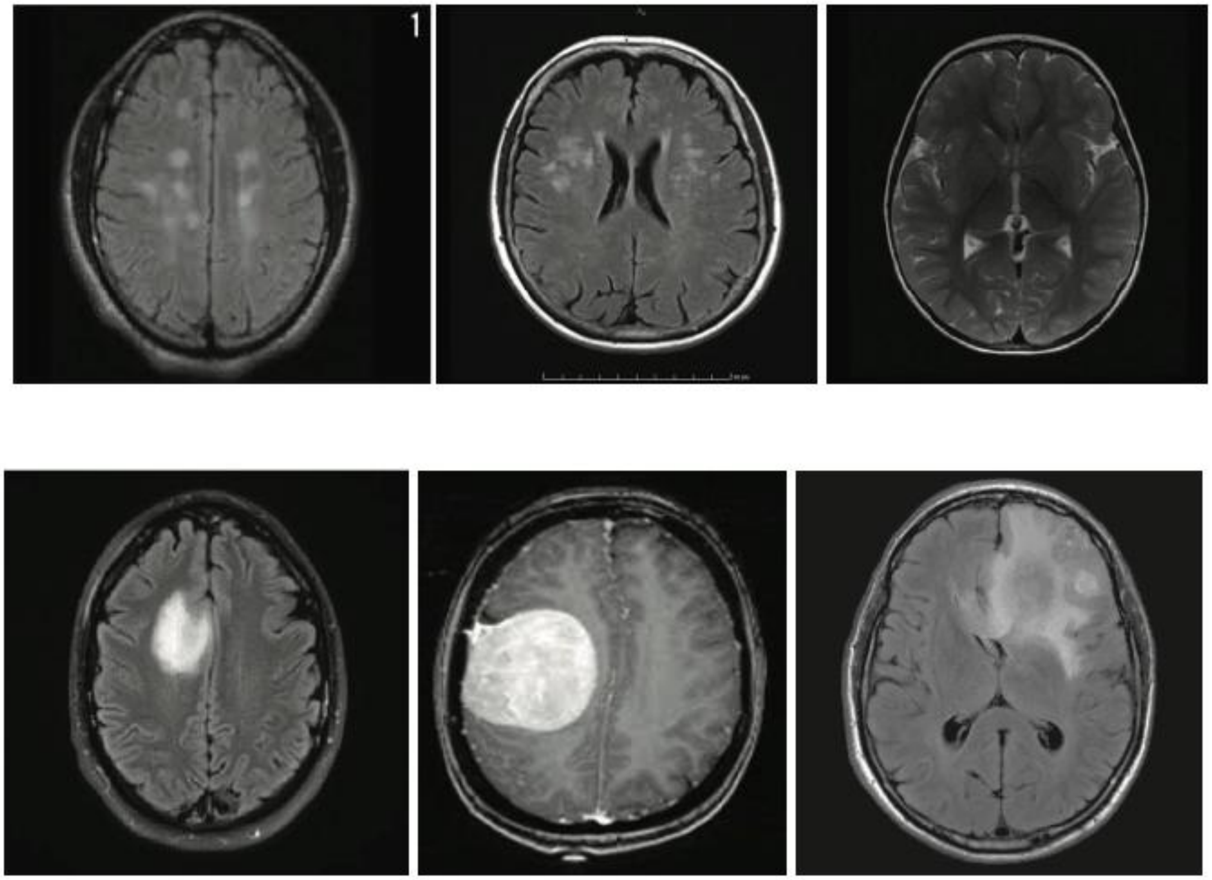

Capstone 3 Project Report

MRI Brain Tumor Classification

Travis Bates

Problem Statement

What opportunities exist for oncologists to automate and decrease tumor diagnosis time by 50% while maintaining an accuracy of 95% or higher with the use of machine learning and MRI images to detect the presence of a tumor?

Background

Brain tumors are among the most serious neurological disorders, often requiring timely and precise diagnosis to improve patient outcomes. MRI imaging is one of the most reliable imaging techniques for diagnosing brain abnormalities, but manual interpretation is time-consuming and subject to human error.

Recent advancements in deep learning, particularly convolutional neural networks (CNNs), have shown great promise in medical imaging tasks. By training a model on labeled MRI scans, we can automate the tumor detection process, reduce diagnostic workload, and potentially enhance diagnostic accuracy

Summary of Findings

The objective of this project was to build a binary image classification model to detect brain tumors from MRI images using deep learning. Early and accurate detection of brain tumors can improve treatment planning and outcomes. This project explores the use of CNNs and transfer learning to automate the detection process.

DATA WRANGLING AND EXPLORATORY DATA ANALYSIS:

The dataset contained 7,023 images of human brain MRI scans which are classified into 4 classes:

- Glioma
- Meningioma
- Pituitary
- No Tumor

My plan was to test 3 different models due to time constraints with this project: Custom Convolutional Neural Network (CNN) model, MobileNetV2 and EfficientNetB0. A custom CNN model allowed me to build and experiment with my own architecture to help me gain a better understanding of how CNN layers (Conv, Pool, Dropout, etc.) work. Transfer learning with both MobileNetV2 and EfficientNetB0 models allowed me to leverage powerful models already trained on large image datasets. This was a good starting point since I am new to image data.

Since I was handling images, I created a data frame with the file path of the MRI image and the label of the image based on the classes above. Those images were then encoded with binary labels to identify if an image contained a tumor or no tumor. I kept the original class labels in the data frame, as my goal is to expand the model to multiclass classification in the future.

The data set contained images of different sizes and modes, I standardized both the image size and image mode of the MRI images. Both the transfer learning models, and my customer CNN required the same image size and mode: (224, 224) for the image size and RGB for image mode.

The final step before building and training my models was exploring the class imbalance, as this could be an issue during training. The data set had a moderate imbalance ratio of 2.5, showing that there were 2.5 times more tumor images than non-tumor images.

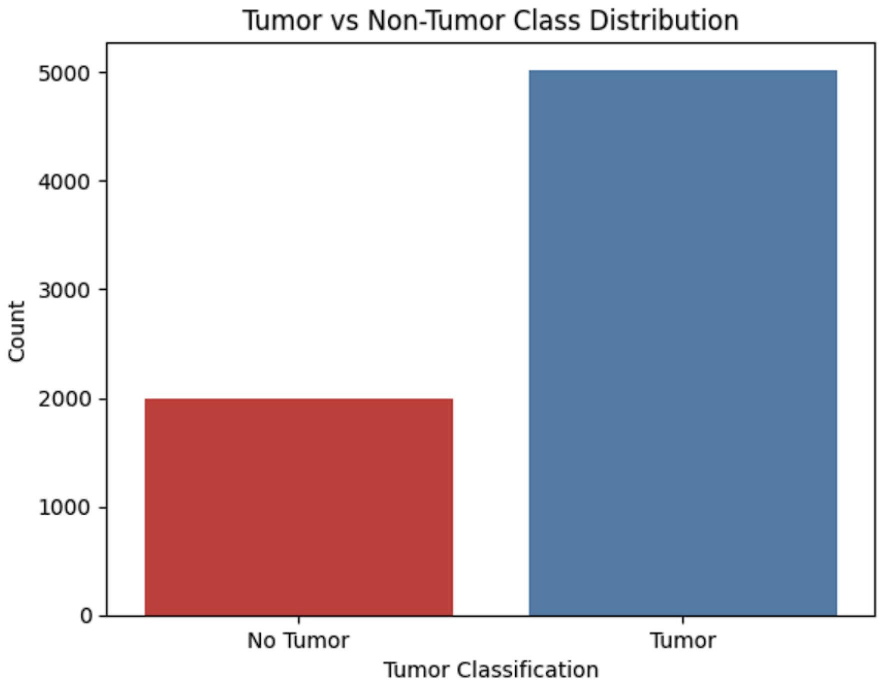

Figure 1: Class distribution between tumor and non-tumor images

PREPROCESSING AND TRAINING:

- Splitting Data

  I had a moderate data size with 7,093 MRI images. To ensure that I had at least 1,000 or more images in my validation and test sets, I used a 70/15/15 split. I stratified my splits to preserve the proportion of classes between my split data.

- Encoding Categorical Features

  This was handled during the data wrangling/EDA stage.

  - One-hot encoding was used for tumor and non-tumor images.

- Standardization

  This was also handled during the data wrangling/EDA stage.

  - Images were resized to (224,224) which is required for all 3 models that I tested.
  - Image modes were converted to RGB which is required for all 3 models that I tested.

Each model required some additional preprocessing that was unique to the model.

- Both EfficientNetB0 and MobileNetV2 have preprocessing functions built into the models that are unique for each model.
- Normalization was done for the images of the custom CNN model.

MODEL SECLECTION:

I evaluated the models using the following metrics:

- **Precision**: High precision ensures fewer healthy patients are incorrectly flagged as having a tumor.
- **Accuracy**: Gives a quick snapshot of performance.
- **Recall**: High recall ensures that as many tumor cases as possible are caught.
- **F1 Score**: It's a better single metric than accuracy in imbalanced medical datasets.
- **Confusion Matrix:** Useful for diagnosing what types of errors the model makes.

Given the project's goal to identify tumors, Recall and F1 score were prioritized. This was important in medical diagnosis as we do not want to miss any actual tumors, and we want to minimize the number of false alarms. False alarms in the case of brain tumors can be costly to the patient finically, physically, and mentally.

I trained and tested the following algorithms:

Model	**Description**
Custom CNN	Simple, interpretable baseline
EfficientNetB0	Higher accuracy, Uses compound scaling (depth + width + resolution) to balance performance
MobileNetV2	Lightweight, great for real-time applications, good baseline model for transfer learning with small datasets

After training and evaluating each model with the original images, there was suspicion of overfitting or data leakage. Therefore, I then introduced augmentation to the training images to help reduce overfitting or data leakage. The following augmentation was performed on the training images:

- Rotation range = 15
- width shift range = 0.05
- height shift range = 0.05
- zoom range = 0.1
- shear range = 0.05
- brightness range = [0.8, 1.2]
- horizontal flip = True
- fill mode = 'nearest'

After augmentation was introduced model performance was affected. There was a drop in performance in both the custom CNN and EfficientNetB0. However, the MobileNetV2 model still performed well.

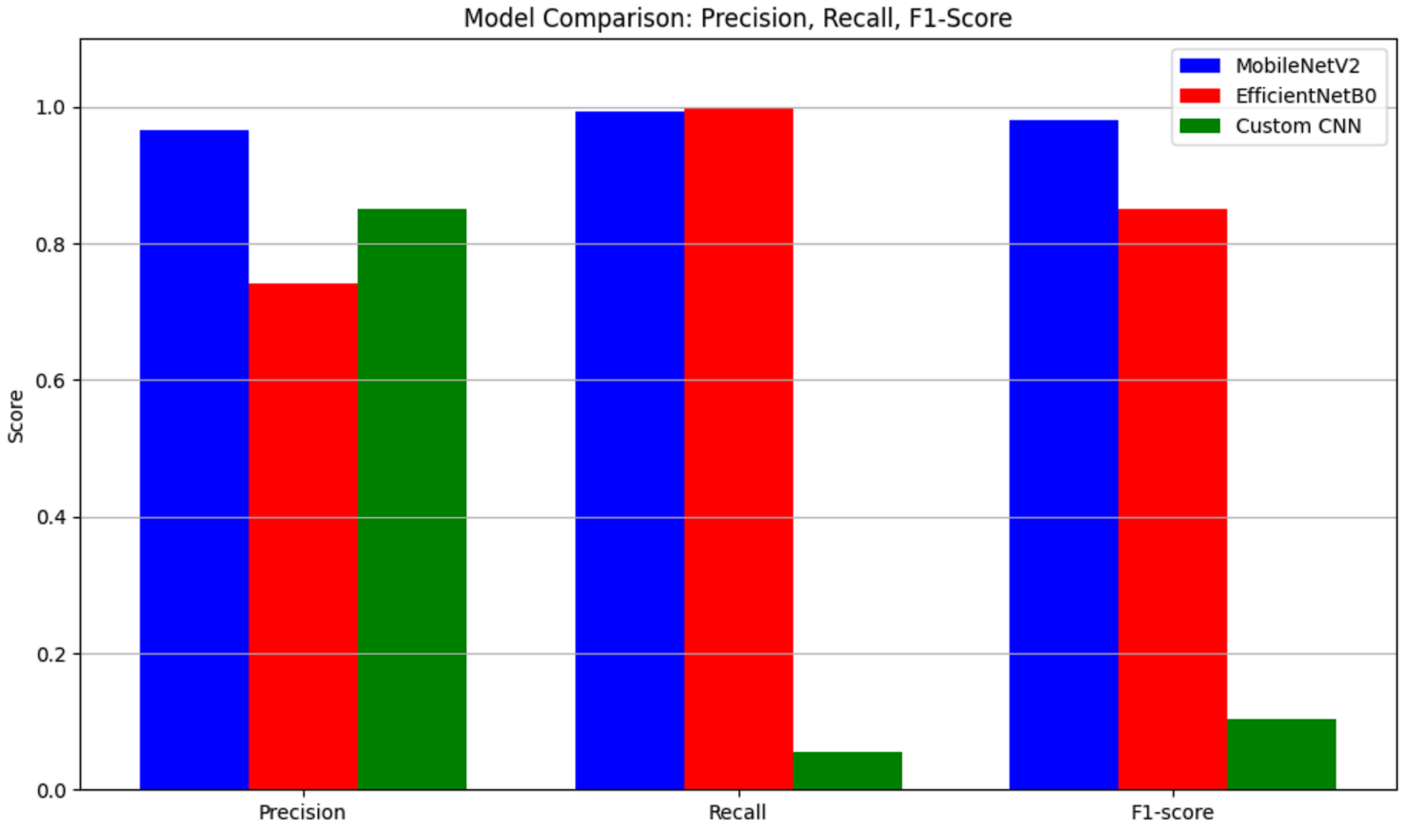

Figure 2: Precision, Recall, and F1 Score comparison between models

One notable observation was with the EfficientNetB0 model, was that the model started overfitting to the tumor class and was only predicting tumors for all images. To help address this issue I attempted the following:

- Freezing and unfreezing layers of the base model.
- Adjusting the number of trainable layers of the base model.
- Modifying learning rates and optimizer settings.
- Balancing classes using class weights.

Despite proper architecture and input preprocessing, EfficientNetB0 consistently overfit or failed to converge meaningfully on this specific dataset.

Due to time constraints, I made the decision to go with MobileNetV2 model. It has lighter architecture, better suited for this binary classification task. Also, after proper data augmentation and fine-tuning, MobileNetV2 produced balanced, high-quality predictions with 95%+ accuracy, strong F1-scores, and stable convergence.

RESULTS AND RECOMMENDATIONS

The final MobileNetV2 had the following performance metrics:

Model	Precision	Recall	F1-score
MobileNetV2	0.97	0.99	0.98

The overall false positives and false negatives were:

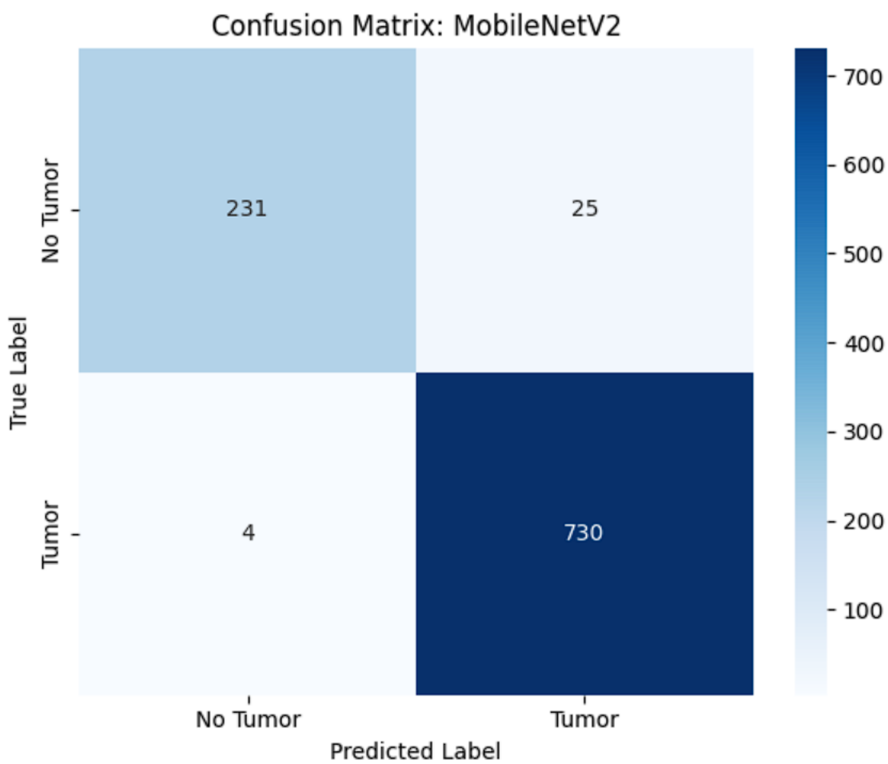

Figure 3: Confusion Matrix for final MobileNetV2 model

Recommendations:

- Continuing with more extensive and comprehensive validation before full deployment:

  While the MobileNetV2 performed well on our test dataset (~95% accuracy and strong precision/recall), real-world deployment in a medical context should include further validation on larger, diverse, and clinically annotated datasets. A human-in-the-loop review system is also recommended.

- Continuing with regular model updates

  As new MRI data becomes available, periodic re-training or fine-tuning should be conducted to maintain model performance and generalization to unseen imaging patterns or scanner types.

- Use model confidence in decision-making

  Model output probabilities can use a more conservative threshold or be used to trigger manual review when confidence is low, reducing risk in clinical decisions.

- Include more metadata

  Adding patient metadata (age, sex, scanner type, etc.) as auxiliary features may improve model performance and interpretability.

FUTURE SCOPE OF WORK

1. Interpretability
   - Implement explainability tools like Grad-CAM to visualize which regions of the MRI image the model uses to make its predictions. This can increase clinician trust and identify potential biases.
2. Multiclass Classification
   - Expand the model from binary classification (tumor vs. no tumor) to classify tumor types (e.g., glioma, meningioma, pituitary tumor) to support more granular diagnostics.
3. Model Ensemble
   - Combine predictions from multiple architectures (e.g., MobileNetV2, EfficientNet, and a custom CNN) through ensembling to potentially boost performance and robustness.
4. Real-Time Web or Mobile App Integration
   - Develop a lightweight interface for doctors to upload MRI images and receive predictions, either via a web app or a mobile device with proper security protocols.
5. Clinical Trial or Validation Study
   - Collaborate with medical institutions to validate the model on real patient data in a clinical trial setting before full deployment that would require regulatory approval.