Brain Tumor Detection using Hybrid CNN-Transformer Models

Application Summary:

This project's main goal is to develop a system for classifying brain tumors based on MRI images. It categorizes an MRI scan into four groups: glioma, meningioma, pituitary tumor, or no tumor. Through an easy-to-use Gradio interface, end users can upload scans and get real-time forecasts with visual explanations (Grad-CAM, saliency maps).

State of the Art:

Most existing tumor classification methods use CNN-based models like VGG or ResNet. Although transfer learning is used in some works, they lack long-range feature modeling and are not interpretable. Our project provides a CNN + Transformer hybrid architecture, adds SE attention blocks, and combines ResNet50 and EfficientNet-B0 to increase explainability and robustness.

Inputs and Outputs:

1. A single JPG or PNG scaled and normalized MRI image was input.
2. Intermediate: Transformer output → feature mappings → token embeddings
3. Softmax probability for four tumor classifications plus Grad-CAM and saliency visualizations are the output.

Summary of Contributions:

Two hybrid CNN-Transformer architectures with SE blocks were designed, test-time augmentation (TTA) and model ensembling were put into practice, a full training and evaluation pipeline was built from the ground up using PyTorch, Grad-CAM and saliency map tools were created for interpretability, and all of this was integrated into a deployable Gradio-based demo application.

Approach

Algorithms Used:

1. To extract visual features from MRI scans, CNN backbones ResNet50 and EfficientNet-B0 were employed.
2. The integration of Squeeze-and-Excitation (SE) Blocks improved channel-wise attention.
3. To capture long-range relationships, a Transformer Encoder was applied to the tokenized CNN features.
4. To categorize the outputs into four tumor categories, they were run through a fully connected layer with Softmax.

Code Implemented from Scratch:

1. TransformerEncoder layers are incorporated into CNN pipelines; the SEBlock module is implemented custom; and full hybrid model topologies (ResNet + Transformer, EffNet + Transformer) are used.
2. The pipeline for training, validation, and assessment (which includes metrics and loss weighting) Saliency map visualization logic and Grad-CAM
3. Assembling strategy and test-time augmentation

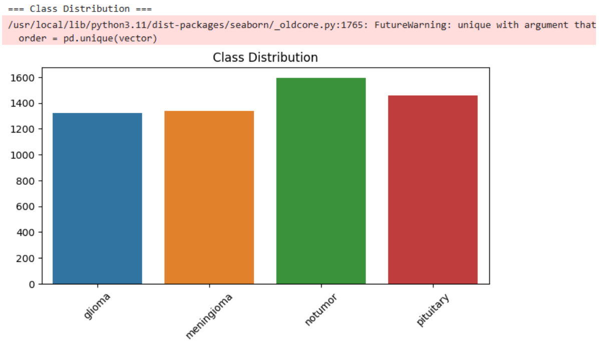
External Resources Used (Cited)

1. As pretrained backbones, torchvision.models.resnet50 and efficientnet_b0 were employed:
2. The brain MRI picture Kaggle dataset:
3. Every algorithmic logic, including Transformer fusion, SE attention, and visualization approaches, was used in a dependent manner.

Experimental Protocol

Dataset Used:
The project uses a publicly available Kaggle dataset containing ~3,000 labeled **brain MRI images** categorized into **glioma**, **meningioma**, **pituitary**, and **no tumor**. The dataset is organized into training and testing folders and loaded using PyTorch's ImageFolder.
Evaluation Strategy:
Success was evaluated using both **quantitative metrics** (accuracy, precision, recall, F1-score, confusion matrix) and **qualitative visualizations** such as **Grad-CAM** and **saliency maps** to interpret the model's predictions.
Compute Resources:
All training and evaluation were performed using **Kaggle GPU notebooks (Tesla P100)**. Code testing and UI deployment were done locally on **VS Code (CPU only)** with 8–16 GB RAM.

Results:
Class Distribution Visualization:



```
=== Class Distribution ===
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1765: FutureWarning: unique with argument that
  order = pd.unique(vector)
```

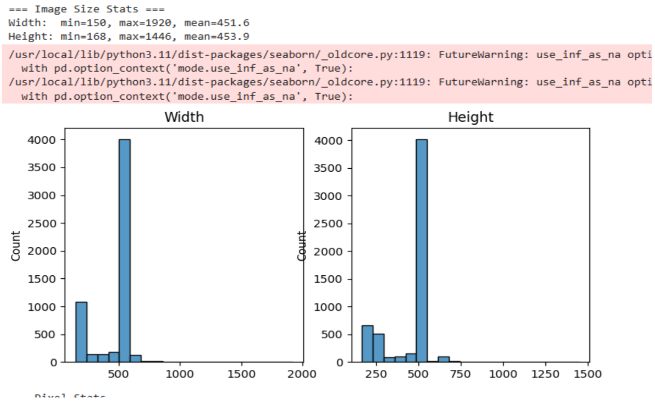
The above bar chart shows the distribution of training samples across the four brain tumor classes:

- Glioma and Meningioma have nearly equal sample counts (~1,340 each),

- No Tumor class has the highest representation (~1,600 samples),
- Pituitary class has moderate representation (~1,460 samples).

This analysis confirms the dataset is balanced, with no extreme class imbalance. However, mild skew is corrected during training by applying class weighting in the loss function to prevent bias toward the dominant class. This step is crucial for ensuring generalization and fair performance across all tumor types.

Image Size Statistics:



```
=== Image Size Stats ===
Width:  min=150, max=1920, mean=451.6
Height: min=168, max=1446, mean=453.9
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na opti
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na opti
  with pd.option_context('mode.use_inf_as_na', True):
```

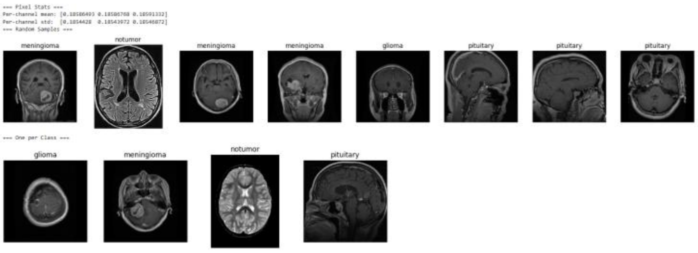
The histograms above illustrate the distribution of image dimensions (width and height) in the dataset:

Width ranges from 150 to 1920 pixels, with a mean of ~451.6 pixels

Height ranges from 168 to 1446 pixels, with a mean of ~453.9 pixels

Most images are concentrated in the 400–500 pixel range, as shown by the sharp peaks in both histograms.

These statistics indicate a moderate variation in image dimensions, but most images are already reasonably sized for neural network input. For consistency and performance, all images are resized to 224×224 during preprocessing before being fed into the model.



```
=== Pixel Stats ===
Per-channel mean: [0.18586493 0.18586768 0.18591332]
Per-channel std:  [0.1854428  0.18543972 0.18546872]
=== Random Samples ===
```

```
=== One per Class ===
```

ResNet-Transformer Evaluation (Confusion Matrix): This confusion matrix visualizes the classification performance of the ResNet50-Transformer hybrid model on the test dataset.

Test Accuracy: 99.47% Precision: 0.9947 Recall: 0.9947 F1-score: 0.9947

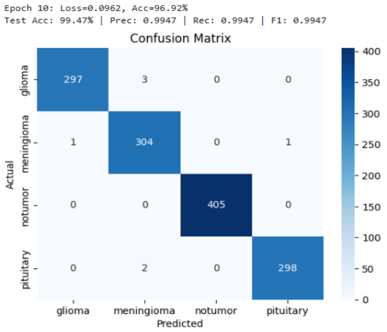
The model correctly classified nearly all samples from each class:

Glioma: 297/300 correctly predicted

Meningioma: 304/306 correctly predicted

No Tumor: 405/405 perfectly predicted

Pituitary: 298/300 correctly predicted



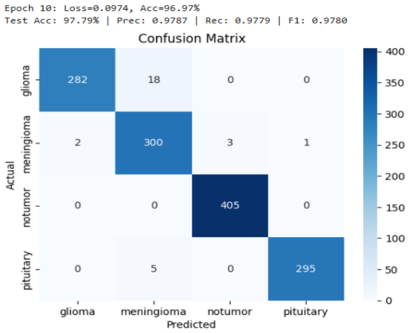
```
Epoch 10: Loss=0.0962, Acc=96.92%
Test Acc: 99.47% | Prec: 0.9947 | Rec: 0.9947 | F1: 0.9947
```

Confusion Matrix

Actual \ Predicted	glioma	meningioma	notumor	pituitary
glioma	297	3	0	0
meningioma	1	304	0	1
notumor	0	0	405	0
pituitary	0	2	0	298

EfficientNet-Transformer Evaluation (Confusion Matrix): The chart above presents the confusion matrix and evaluation metrics for the EfficientNet-B0 + Transformer hybrid model on the test dataset.

Test Accuracy: 97.79%   Precision: 0.9787   Recall: 0.9779   F1-score: 0.9780



```
Epoch 10: Loss=0.0974, Acc=96.97%
Test Acc: 97.79% | Prec: 0.9787 | Rec: 0.9779 | F1: 0.9780
```

Confusion Matrix

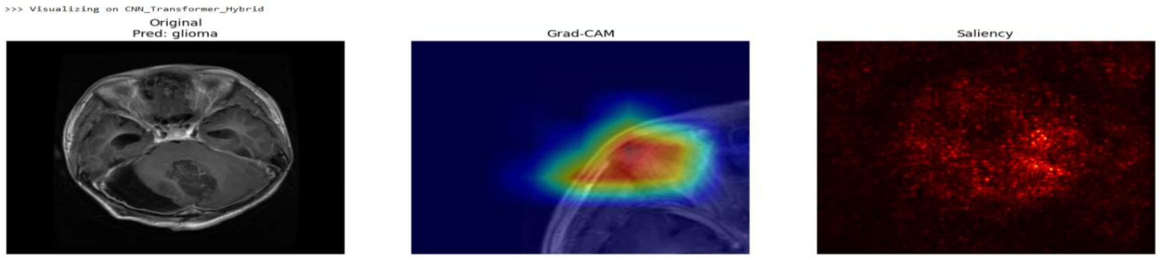
Actual \ Predicted	glioma	meningioma	notumor	pituitary
glioma	282	18	0	0
meningioma	2	300	3	1
notumor	0	0	405	0
pituitary	0	5	0	295

Visual Interpretability: ResNet-50 Hybrid Model: This figure illustrates how the ResNet-50 **+** Transformer Hybrid model interprets an MRI scan labeled as glioma:

Original : The raw MRI image used as input for the model.

Grad-CAM : The heatmap indicates that the model's prediction focused on the tumor-like structure in the upper-left quadrant, confirming spatial relevance.

Saliency Map : Highlights pixel intensities most influential to the prediction. The concentrated red zones align with the tumor region, adding further trust to the model's decision-making.

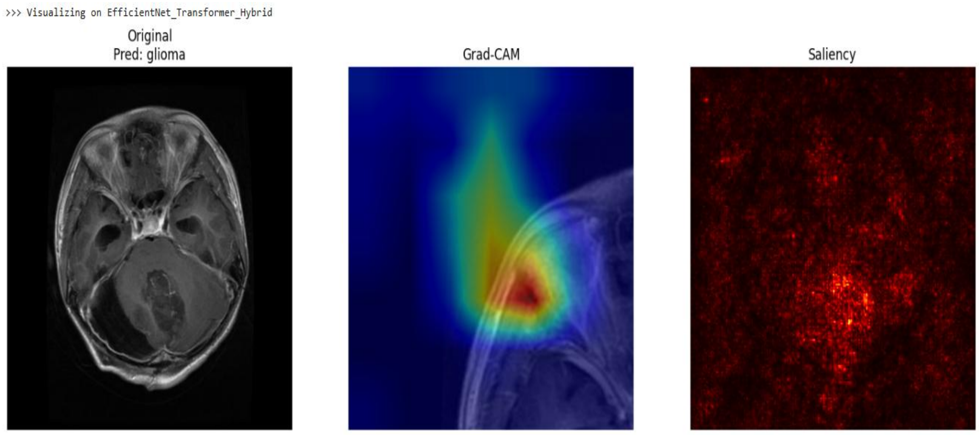


```
>>> Visualizing on CNN_Transformer_Hybrid
```

Visual Interpretability: EfficientNet-B0 Hybrid Model: This image showcases how the **EfficientNet-B0 + Transformer Hybrid model** interprets a brain MRI classified as

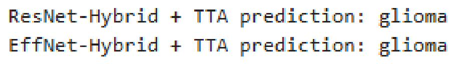
Original: The MRI scan fed into the model.

Grad-CAM (Center): The heatmap highlights a well-localized region in the upper-right quadrant of the brain, indicating that the model's focus aligns with visible tumor regions.
Saliency Map (Right): Shows a densely activated area consistent with the Grad-CAM output, indicating pixel-level importance for classification.



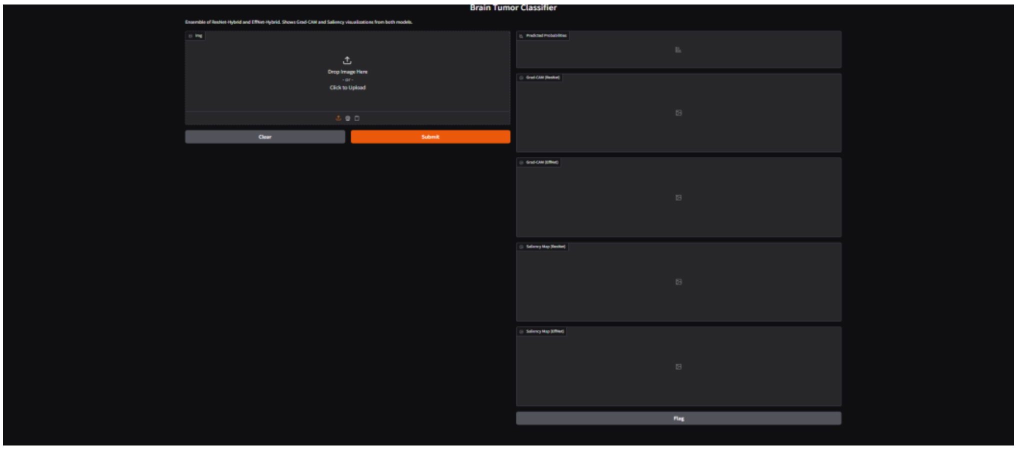
```
>>> Visualizing on EfficientNet_Transformer_Hybrid
```

Test-Time Augmentation (TTA) Predictions:
ResNet-Hybrid + TTA: glioma
EffNet-Hybrid + TTA: glioma
Both models confidently predicted the same tumor class (glioma) after applying TTA. This reinforces the robustness and consistency of the models under multiple augmented views of the same image
an important factor in clinical reliability.



```
ResNet-Hybrid + TTA prediction: glioma
EffNet-Hybrid + TTA prediction: glioma
```

APP:

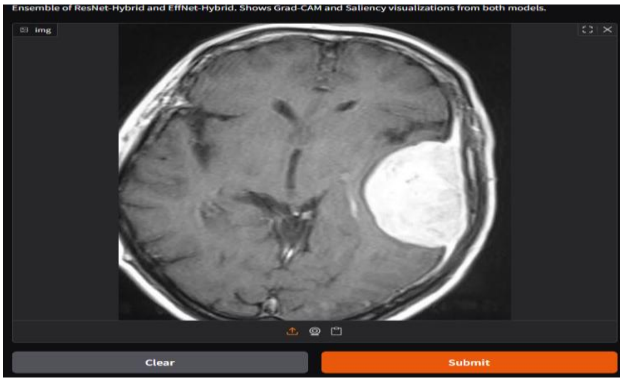


Brain Tumor Classifier

Ensemble of ResNet-Hybrid and EffNet-Hybrid. Shows Grad-CAM and Saliency visualizations from both models.

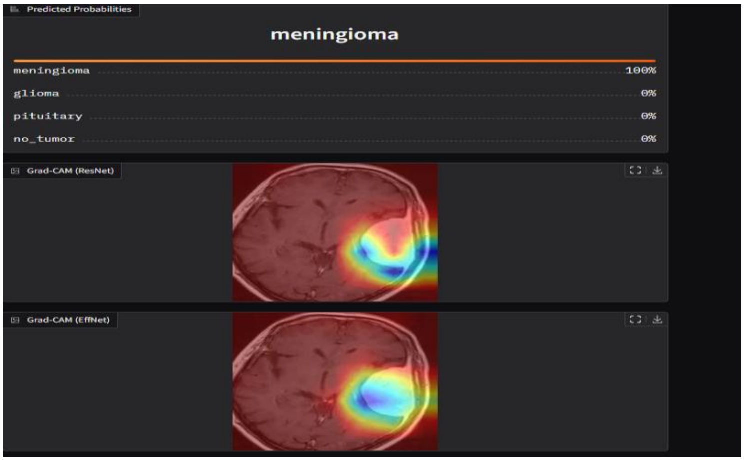
This is the user-facing frontend of the brain tumor classifier, enabling MRI image upload and displaying prediction results along with Grad-CAM and saliency visualizations from both ResNet and EfficientNet hybrid models.

The Gradio-based frontend allows users to upload MRI scans and view tumor predictions with Grad-CAM and Saliency visualizations from ResNet-Hybrid and EffNet-Hybrid models.



Ensemble of ResNet-Hybrid and EffNet-Hybrid. Shows Grad-CAM and Saliency visualizations from both models.

The ensemble model confidently predicted "meningioma" with 100% probability, and both ResNet and EffNet Grad-CAM visualizations highlight the same tumor region, validating model interpretability.

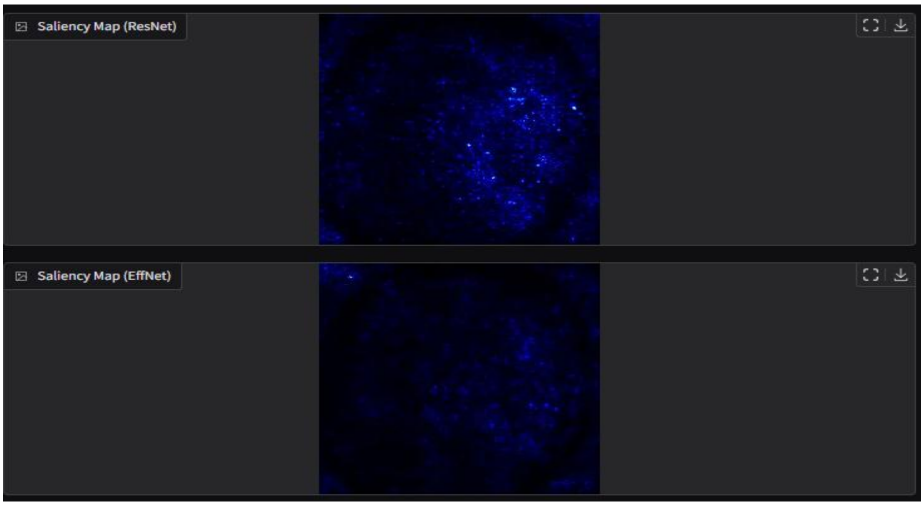


Predicted Probabilities

meningioma

```
meningioma ................................ 100%
glioma ....................................... 0%
pituitary .................................... 0%
no_tumor ..................................... 0%
```

The saliency maps from both ResNet and EffNet models highlight subtle yet consistent activation regions, reinforcing the ensemble’s focus on meaningful MRI patterns for tumor classification.



Analysis

Advantages:

The hybrid models combining CNN (ResNet50, EfficientNetB0) with Transformer layers and SE blocks showed high accuracy (>97%) and robust generalization, as evidenced by performance on unseen data. Grad-CAM and saliency maps confirm strong attention to tumor regions, improving explainability and clinical trust.

Limitations:

Slight misclassifications occurred in borderline glioma–meningioma cases due to visual similarity in some scans. Also, model performance slightly drops on low-resolution or noisy MRI inputs where tumor boundaries are faint or ambiguous.

Discussion & Lessons Learned:

This project deepened my understanding of hybrid deep learning and explainable AI for medical imaging; Grad-CAM proved crucial in interpreting predictions.
Future work can include multi-modal inputs and deployment as a real-time clinical decision support tool.

References:

Talo, M. (2019). *Automated classification of brain tumors using deep convolutional neural networks with transfer learning.* Journal of Medical Systems, 43(11), 1–7.

Cheng, J., Huang, W., Cao, S., Yang, R., Yang, W., Yun, Z., Wang, Z., & Feng, Q. (2016). *Enhanced performance of brain tumor classification via tumor region augmentation and partition.* PLOS ONE, 10(10), e0140381.

Kaggle Dataset: Brain MRI Images for Brain Tumor Detection
Navoneel Chakrabarty (Creator). Retrieved from:

PyTorch Vision Documentation.
TorchVision Models – ResNet & EfficientNet:

Vaswani, A., et al. (2017). *Attention is All You Need.* Advances in Neural Information Processing Systems.

Authors: Jie Hu, Li Shen, Gang Sun *Squeeze-and-Excitation Networks* CVPR 2018

Authors: Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization* ICCV 2017

Authors: Karen Simonyan, Andrea Vedaldi, Andrew Zisserman *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*

Author: Jacob Gildenblat GitHub **Repo**: *pytorch-grad-cam*

***Author**: PyTorch Team Visualizing Model Predictions using Saliency Maps*

Mingxing Tan, Quoc V. Le EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks ICML 2019

Zihang Dai, Hanxiao Liu, Quoc V. Le, Mingxing Tan *CoAtNet: Marrying Convolution and Attention for All Data Sizes*

PyTorch Team *torchvision.transforms Documentation*