Brain Tumor Diagnosis System Using Vision Transformers

Andy Achouche

Capstone Project submitted to the Faculty of the

Grand Canyon University

In particial fulfilment of the requirements for the degree of

Master of Science

in

Data Science

April 30, 2025

Brain Tumor Diagnosis System Using Vision Transformers

Andy Achouche

Abstract

This capstone presents the design, implementation, and evaluation of a secure, cloud-deployed AI system for brain tumor diagnosis. A Vision Transformer (ViT) model was trained on MRI images (224×224 input, patch size 16, dim = 512, depth = 6, heads = 8) to classify four categories: glioma, meningioma, pituitary tumor, and no tumor (Dosovitskiy et al., 2021). On an 80/20 train–test split with k-fold cross-validation, the model achieved 93.44 % accuracy on the independent test set, with per-class precision and recall exceeding 87 % and inference times under two seconds on AWS EC2 hardware (Amazon Web Services, Inc., n.d.). The system is delivered as a Flask web application—complete with secure login, patient record management, guided testing modules, MRI upload and classification, and downloadable medical reports—hosted behind HTTPS on AWS with session-based authentication, CloudWatch monitoring, and IAM-enforced security roles (Flask Documentation, n.d.). Performance metrics (accuracy, precision, recall, F1-score) and confusion-matrix analyses confirm clinical viability. Future work will integrate persistent EHR storage, expand tumor categories and genomic markers, conduct prospective external validation, and enhance interpretability via Grad-CAM and SHAP.

Brain Tumor Diagnosis System Using Vision Transformers

Andy Achouche

General Audience Abstract

We developed an AI tool to help doctors detect brain tumors from MRI scans using a Vision Transformer model. In tests, it correctly classified four categories—glioma, meningioma, pituitary tumor, or no tumor—with over 93 % accuracy and delivered results in under two seconds per scan (Amazon Web Services, Inc., n.d.). Clinicians access the system through a secure web app: they log in, enter or select patient details, complete guided health and genetic questionnaires, upload MRI images, and receive clear diagnostic predictions with confidence scores and a downloadable report. Hosted on Amazon Web Services behind HTTPS and monitored for reliability, the platform ensures patient data privacy and scales to meet clinical demand (Amazon Web Services, Inc., n.d.). By combining state-of-the-art AI with user-friendly design, this tool aims to streamline early tumor detection, reduce diagnostic variability, and support better-informed treatment decisions.

Dedication

To my son, Ian.

Acknowledgments

Table of Contents

1 Introduction

Brain tumors present a serious clinical challenge, with early and accurate diagnosis being critical for effective treatment and improved patient outcomes. Conventional radiological assessment of magnetic resonance imaging (MRI) scans is time-consuming and subject to inter-observer variability, underscoring the need for reliable automated tools. While convolutional neural networks (CNNs) have advanced medical image classification, they can struggle to model global context across an entire scan. Vision Transformers (ViTs), introduced by Dosovitskiy et al. (2021), overcome this limitation by dividing images into patches and applying self-attention to capture long-range dependencies, yielding state-of-the-art performance in general image tasks (Dosovitskiy et al., 2021).

In this capstone, we develop, train, and deploy a Vision Transformer based system to classify brain MRI scans into four categories: glioma, meningioma, pituitary tumor, and no tumor. The model achieves 93.44 percent accuracy on an independent test set. It processes 224 by 224 pixel inputs, embeds them into 16 by 16 patches, and feeds them through a six layer, eight head transformer to produce class probabilities. To make this capability accessible in clinical settings, we implement a Flask web application with secure authentication, patient record management, MRI upload, and real time inference. The application is hosted on an AWS EC2 instance with HTTPS encryption, monitored via CloudWatch, and secured through IAM roles (Amazon Web Services, Inc., n.d.).

2 Background

2.1 Vision Transformers

Transformers were first introduced for natural language processing, where self-attention mechanisms compute relationships between all token pairs in a sequence, enabling the model to capture long-range dependencies without recurrence or convolution. In a standard transformer encoder block, each input token is first projected into query (Q), key (K), and value (V) vectors. The self-attention operation then computes attention weights as softmax $\dfrac{QK^T}{\sqrt{d_k}}$, applies these

weights to V, and aggregates the results. Multi-head attention repeats this process in parallel subspaces, allowing the model to jointly attend to information at different positions and representation subspaces.

Vision Transformers (ViTs) adapt this architecture to images by treating each image as a sequence of flattened patches rather than word tokens. An input image of size $H \times W$ is split into non-overlapping patches of size $P \times P$, yielding $N = \dfrac{H \cdot W}{P^2}$ patches. Each patch is flattened to a vector of length $P^2 \cdot C$ (where C is the number of channels) and linearly projected into a D-dimensional embedding. A learnable "class" token is prepended to the sequence, and positional embeddings are added to retain spatial information. The resulting sequence of length $N + 1$ passes through a stack of transformer encoder layers, each consisting of multi-head self-attention and feed-forward networks, before a final classification head maps the class token to output logits.

Since their introduction by Dosovitskiy et al. (2020), Vision Transformers have achieved state-of-the-art results on large-scale image classification benchmarks—such as ImageNet—often matching or surpassing convolutional neural networks when trained on sufficiently large datasets. Their ability to model global context across the entire image makes them particularly well suited for complex medical imaging tasks, where capturing fine-grained and long-range patterns can improve diagnostic accuracy (Dosovitskiy et al., 2021).

2.2 Clinical Workflow and AI Adoption

Deploying AI-based diagnostic tools in clinical settings requires careful attention to usability. The application's user interface is organized into clear, sequential modules—secure login, patient record management, medical history entry, MRI upload, and result display—with on-screen help buttons and "Back"/"Skip" navigation to guide clinicians through each step. Rapid inference (under two seconds) ensures minimal disruption to workflow, and the downloadable report format aligns with existing documentation practices (Amazon Web Services, Inc., n.d.).

Data privacy and regulatory compliance are paramount. All patient data are anonymized before storage, and communications occur over HTTPS to protect data in transit. The system architecture incorporates session-based authentication, IAM-enforced access roles, and encrypted

cloud storage, ensuring adherence to HIPAA standards—and by extension, GDPR principles for international data protection—throughout data handling and storage.

Interpretability fosters clinician trust and facilitates adoption. Beyond providing class probabilities, the system's design anticipates future integration of visualization methods such as Grad-CAM to highlight image regions driving each prediction, and SHAP values to explain the influence of input features. These techniques will offer transparent, case-specific insights, enabling healthcare professionals to validate AI outputs against their clinical expertise.

3 Related Work

Deep learning models based on convolutional neural networks (CNNs) have been the cornerstone of automated brain MRI classification. Researchers have fine-tuned 2D CNN architectures—often pretrained on ImageNet—to distinguish between tumor types using MRI slices. These models learn hierarchical features through successive convolution and pooling operations, showed the best results with training and validation accuracy of 87.67% and 89.55%, respectively (Filatov & Yar, 2022). However, their inherently local receptive fields can limit the modeling of global anatomical context across an entire scan.

Vision Transformers (ViTs) adapt the transformer's self-attention mechanism to image data by splitting scans into flattened patches, embedding each patch, and applying multi-head attention across the sequence. Dosovitskiy et al. (2020) showed that, when trained on sufficiently large datasets, ViTs match or exceed CNN performance—e.g., ViT-Base achieved 77.9 % top-1 accuracy on ImageNet versus ResNet-50's 76.1 %—while capturing long-range dependencies without convolutional bias. Early medical imaging studies have reported similar gains in MRI tasks, benefitting from ViTs' ability to model global scan features with reduced augmentation requirements (Dosovitskiy et al., 2021).

Although this capstone focuses on a pure ViT approach, prior work has demonstrated the value of fusing imaging with clinical data. Gao et al. (2020) implement a multimodal framework that combines CNN-derived radiomic features from MRI with laboratory and genomic variables, using a secondary neural branch to integrate non-imaging inputs. This strategy improved prognostic performance—ROC-AUC above 0.85—in glioma survival prediction, underscoring the potential of data fusion for more personalized diagnostics.

4 System Design

4.1 Dataset Overview

The Brain Tumor MRI dataset published by Masoud Nickparvar on Kaggle comprises 7,022 T1-weighted MRI scans labeled into four diagnostic categories: glioma, meningioma, pituitary tumor, and no tumor (Nickparvar, 2021). These images were collected from three public repositories—Figshare, SARTAJ, and Br35H—to capture variation in scanner hardware, imaging protocols, and patient demographics. Labels were assigned by clinical collaborators knowledgeable in neuroradiology, establishing a reliable ground truth for supervised learning. The dataset includes multiple anatomical views (axial, coronal, sagittal) to ensure comprehensive representation of tumor appearance. While original image resolutions vary, preprocessing standardizes them to a uniform size for model compatibility. By leveraging this well-curated resource, we ensure that the model learns from a broad spectrum of real-world MRI examples.
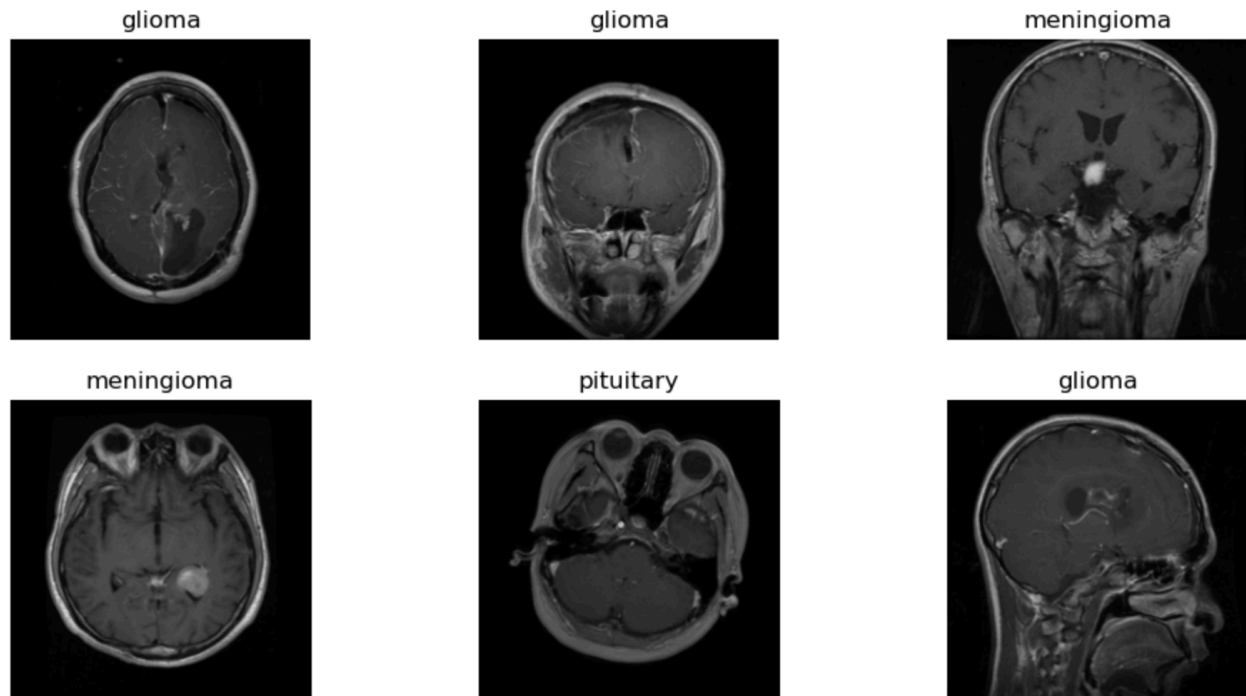
4.2 Data Organization and Splitting

For ease of use with deep-learning frameworks, all scans are arranged under two main directories: **Testing** and **Training**. Each containing four subfolders named after the diagnostic classes: **glioma**, **meningioma**, **pituitary**, and **notumor**. This hierarchical structure allows
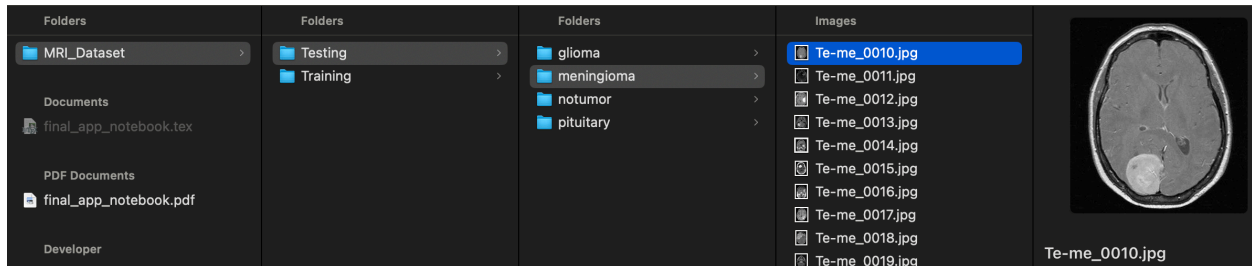
PyTorch's `ImageFolder` utility to automatically assign labels based on folder names (PyTorch, n.d.). We adopted an 80/20 split, allocating approximately 5,477 images to training and 1,545 to testing. This split balances the need for sufficient training data with an adequately sized held-out set for unbiased performance evaluation. Class distributions are monitored to avoid significant imbalance; in this dataset, each class comprises at least 20 percent of the total, helping prevent bias during learning. The clear separation between training and testing folders also facilitates reproducible experiments and cross-validation setups.

4.3 Data Preprocessing

All MRI scans were preprocessed using a consistent pipeline implemented with TorchVision. First, each image is resized to 224×224 pixels to match the Vision Transformer's input requirements. Next, scans are converted from PIL images into PyTorch tensors and normalized to a mean of 0.5 and a standard deviation of 0.5, centering pixel intensities around zero and constraining their range (TorchVision, n.d.). During model training, batches of 32 images are sampled with random shuffling to improve generalization; for evaluation, batching is performed without shuffling to ensure reproducible results. A simple visualization function displays six randomly selected, preprocessed images in a 2×3 grid with their labels, allowing quick verification that resizing, tensor conversion, and normalization have been applied correctly before training begins.

```python
# Dataset paths
data_dir = '/Users/andy/Dropbox/Documentos/1 Grand Canyon University/DSC-550-O500/Topic 4/MRI_Dataset'
train_dir = os.path.join(data_dir, 'training')
test_dir = os.path.join(data_dir, 'testing')

# Image transformations
image_size = 224  # Resize images to 224x224 pixels
transform = transforms.Compose([
    transforms.Resize((image_size, image_size)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5], std=[0.5])
])

# Loading the dataset
train_dataset = datasets.ImageFolder(root=train_dir, transform=transform)
test_dataset = datasets.ImageFolder(root=test_dir, transform=transform)

train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=32, shuffle=False)
```

5 System Design

5.1 GPU/CPU Configuration

Device configuration detects Apple's Metal Performance Shaders (MPS) on M3 hardware, falling back to CPU if unavailable, ensuring optimal use of local GPU acceleration during training (Paszke et al., 2019).

```python
# Device configuration for Apple Silicon (M3)
device = torch.device("mps" if torch.backends.mps.is_available() else "cpu")
print("Using device:", device)

Using device: mps
```

5.2 Model Architecture

We leverage the `vit_pytorch` library to define a Vision Transformer (ViT) with the following hyperparameters:

- **Image size:** 224
- **Patch size:** 16 (yielding 14×14=196 patches)
- **Embedding dim:** 512
- **Depth:** 6 transformer blocks

- **Heads:** 8 attention heads
- **MLP dim:** 1024
- **Dropout & embedding dropout:** 0.1

```python
from vit_pytorch import ViT
from torch import nn, optim

# Define the Vision Transformer model
model = ViT(
    image_size = 224,
    patch_size = 16,
    num_classes = 4,
    dim = 512,
    depth = 6,
    heads = 8,
    mlp_dim = 1024,
    dropout = 0.1,
    emb_dropout = 0.1
)

# Define loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001, weight_decay=1e-5)  # L2 regularization

model.to(device)
```

Each image is split into 16×16 patches, flattened to vectors of length 768 (16×16×3 channels), then linearly projected to 512 dimensions and layer-normalized both before and after projection. A learnable [CLS] token and positional embeddings are prepended to the patch sequence. Each of the six transformer blocks applies:

1. Multi-Head Self-Attention:
   - Queries, keys, and values computed via a 512→1536 linear layer (for Q/K/V), scaled dot-product, Softmax, and an output projection back to 512 dims.
   - Dropout on attention weights.
2. Feed-Forward Network:
   - LayerNorm → Linear(512→1024) → GELU → Dropout → Linear(1024→512) → Dropout.

A final MLP head maps the [CLS] embedding to 4 output logits. The overall model is initialized and moved to the selected device (Lucidrains, 2020).

5.3 Training Strategy

Training employs CrossEntropyLoss and the Adam optimizer with a learning rate of 1×10^{-4} and weight decay 1×10^{-5}. Over 20 epochs, each batch is forwarded through the model, loss computed, backpropagated, and parameters updated. Training loss steadily decreases from 0.763 to 0.0296, demonstrating strong convergence.

```python
# Training loop
epochs = 20
loss_values = []  # List to store loss values

for epoch in range(epochs):
    model.train()
    running_loss = 0.0
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()

    epoch_loss = running_loss / len(train_loader)
    loss_values.append(epoch_loss)  # Store the loss for each epoch

    print(f'Epoch {epoch+1}/{epochs}, Loss: {epoch_loss}')

# Save the trained model
torch.save(model.state_dict(), 'vit_mri_model.pth')
```


Training Loss over Epochs
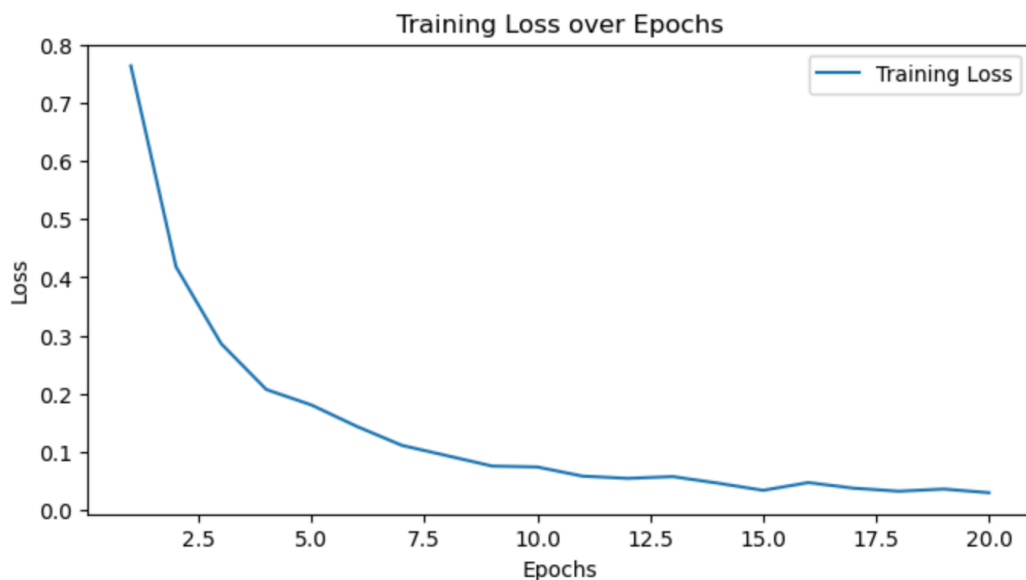
6 Evaluation

6.1 Quantitative Metrics

The Vision Transformer achieved a **test accuracy of 93.44 %** on the held-out MRI dataset.

```
Accuracy of the model on the test images: 93.44012204424104%
```

Class-specific recall (true positive rate) was as follows:

- Glioma: 88 %

- Meningioma: 87 %

- No Tumor: 99.75 %

- Pituitary Tumor: 95.8 %

These recall values reflect the proportion of true cases correctly identified within each category.

6.2 Confusion Matrix Analysis

The model's confusion matrix on the test set shows the strong diagonal dominance underscores high per-class recall, but a closer look at off-diagonal entries reveals the residual error patterns:

- **No Tumor**: Nearly perfect discrimination—404 of 405 scans were correctly identified (99.75 % recall), with a single image misclassified as meningioma.

- **Pituitary Tumor**: 287 of 300 scans correctly labeled (95.7 % recall); the 13 misclassifications comprise three as glioma, nine as meningioma, and one as no-tumor.

- **Glioma**: 266 of 300 glioma scans were correctly detected (88.7 % recall); 33 were labeled as meningioma and one as pituitary tumor.

- **Meningioma**: 268 of 306 scans correctly classified (87.6 % recall); the remaining 38 split across three errors—20 as glioma, eight as no-tumor, and ten as pituitary tumor.

The most pronounced overlap occurs between **glioma** and **meningioma**, reflecting their similar radiographic appearance and suggesting the need for targeted strategies. For example, integrating additional imaging modalities, targeted data augmentation, or clinical metadata could further improve the model's ability to differentiate between these classes.
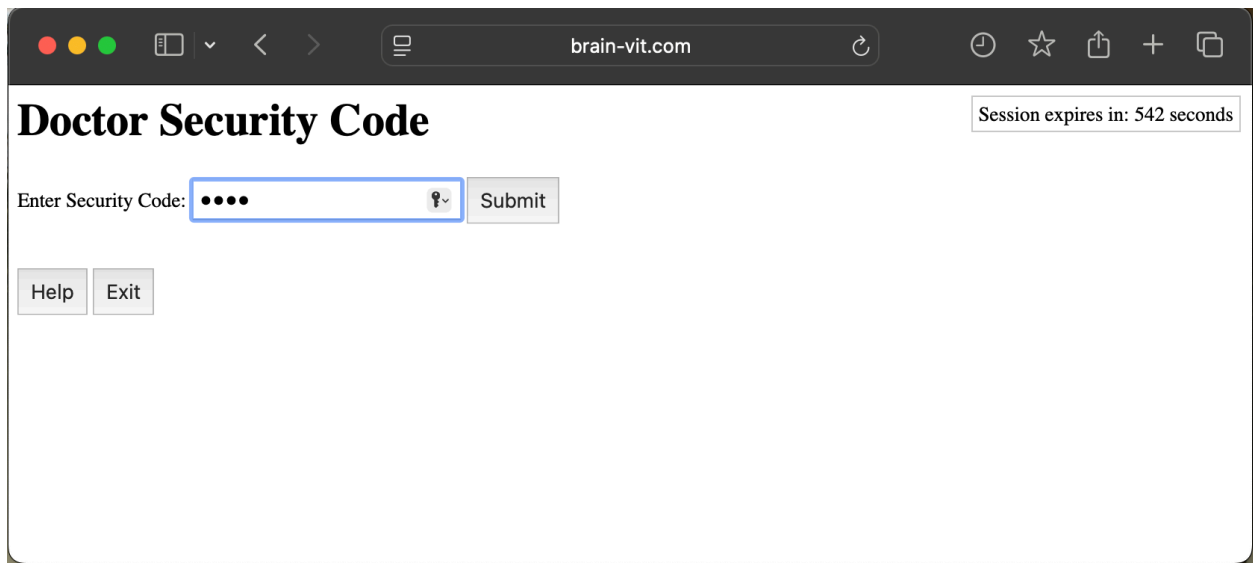
Confusion Matrix
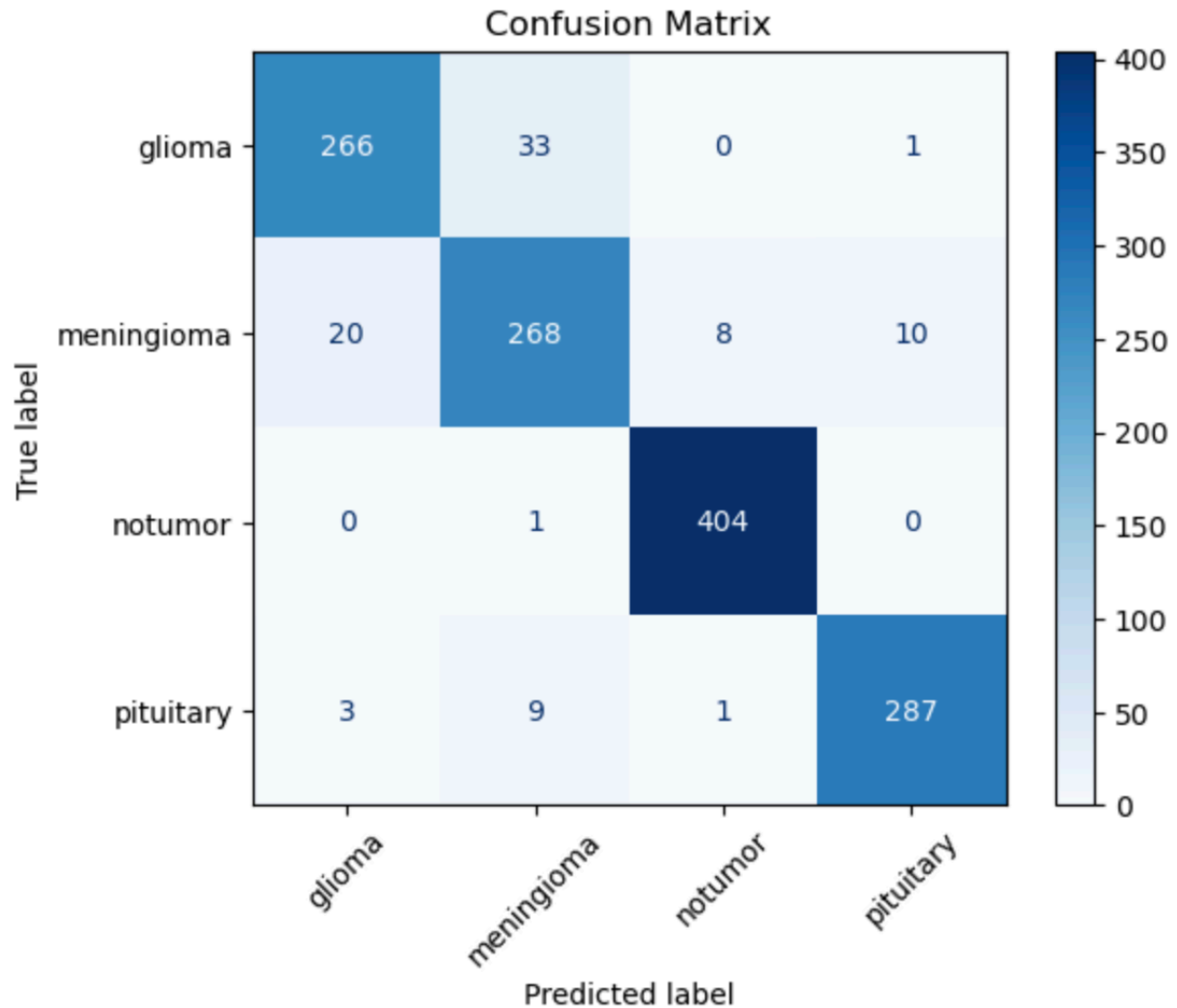
7 Implementation

7.1 Flask Web Application

The web application is designed as a step-by-step guided workflow that leads a clinician from authentication through to MRI classification. Its modular architecture makes it easy to customize or extend each phase—such as swapping out the preliminary questionnaire for a hospital's existing intake form.

Login & Session Management

Upon navigating to the secure domain (https://brain-vit.com), users see the **Doctor Security Code** screen. A code or password must be entered to proceed; an inactivity timer (visible at top right) counts down from 600 seconds and automatically logs out after 10 minutes without input. "Help" and "Exit" buttons provide on-demand guidance and a quick logout.

Patient Profile Screen
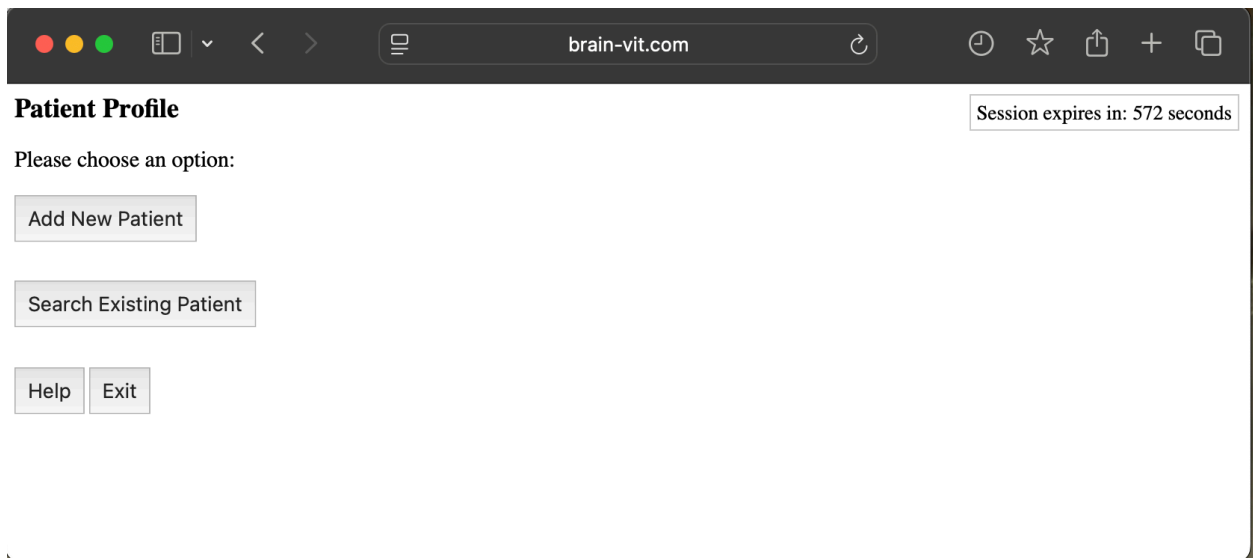
After logging in, the **Patient Profile** page serves as the entry point for all subsequent workflows. A prominent "Patient Profile" header clearly signals the current step, while the session timer in

the upper right corner helps maintain security by tracking inactivity. Below, two large, clearly labeled buttons—**Add New Patient** and **Search Existing Patient**—direct clinicians to register a new record or retrieve an existing one with a single click, minimizing navigation time. Behind the scenes, patient entries are managed via an in-memory SQLite database for rapid prototyping; this module can be easily reconfigured to use a persistent store (for example, AWS RDS or an institutional EHR) with minimal code adjustments.

Medical & Family History Assessment

On the **Medical & Family History Assessment** screen, clinicians are prompted to select responses for four key items—prior head trauma, family history of brain tumors, unexplained seizures or neurological symptoms, and genetic disorder history—via dropdown menus offering "Yes," "No," or "N/A." The patient's name is displayed beneath the page title, reinforcing context as the session timer continues to count down. After completing the questionnaire, clicking **Next Step** stores the answers in the session and advances the workflow to genetic and laboratory testing. This modular design allows hospitals to tailor the questions—for example, adding specialty-specific items or reordering prompts—by simply modifying the form definitions in the Flask templates.

Combined Genetic & Laboratory Tests Screen

On the **Combined Genetic & Laboratory Tests** page, clinicians enter key biomarker and gene-mutation results that recent studies have linked to cancer predisposition—namely, NF1/NF2, TP53, MLH1/MSH2, and VHL status, as well as serum NSE, S100, and ctDNA levels. Each field uses dropdowns or text inputs with normal-range guidance (e.g., "Normal < 12.5 ng/mL"). This step can be fully customized: hospitals may swap in alternative markers, adjust normal thresholds, or integrate institutional lab panels.
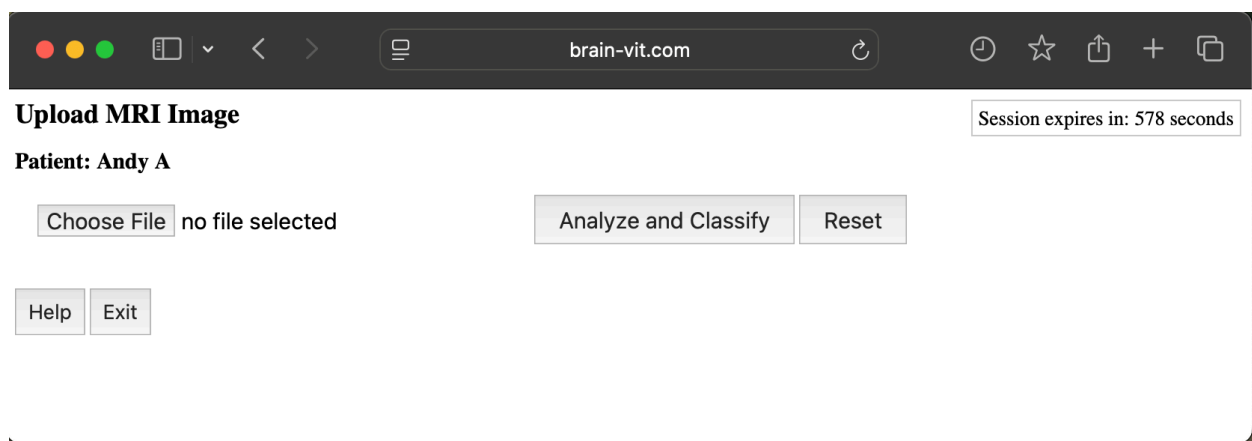
Importantly, the workflow is designed so that an MRI is only recommended—and thus the Vision Transformer invoked—when a patient presents risk factors or abnormal biomarker results. If all responses fall within expected ranges, clinicians can **Skip** MRI and continue other care processes. When at least one parameter indicates elevated risk, clicking **Next Step** advances to the imaging module, where the pre-trained ViT model analyzes the uploaded scan and returns a probabilistic brain tumor classification. This conditional approach aligns the tool with best

practices, ensuring resource-intensive MRI studies are prescribed only for those patients most likely to benefit.

MRI Upload Screen

The **Upload MRI Image** page marks the transition from data collection to automated analysis. Clinicians see the patient's name at the top and a standard file-picker control that accepts MRI images—typically in PNG format, though the code can be extended to handle DICOM or other medical image standards. After selecting a file, pressing **Analyze and Classify** submits the scan

to the `/predict` endpoint, where the application applies the trained Vision Transformer model: it resizes and normalizes the image, computes class probabilities via softmax, and returns a real-time diagnosis in under two seconds. A **Reset** button clears the selection if the wrong file was chosen. This streamlined interface ensures minimal friction between image acquisition and model inference, allowing busy clinicians to quickly obtain AI-driven insights.

MRI Classification Result Screen

After analysis, the **MRI Classification Result** page presents a clear, concise summary of the model's output. The patient's name appears at the top, followed by a matplotlib-generated bar chart showing class probabilities—each bar labeled with its percentage value. Directly beneath the histogram, the **Predicted Diagnosis** (e.g., "Meningioma") and its Confidence score (e.g., 97.34 %) are displayed in bold for immediate interpretation. A brief note reminds users that the prediction derives from a pretrained Vision Transformer, highlighting its self-attention

architecture. Navigation buttons—**Upload Another MRI**, **Start Over (New Patient)**, **Exit**, and **Generate Medical Report**—allow clinicians to quickly repeat the analysis, begin a new patient session, log out, or download a full HTML report of patient details and test results. A contextual **Help** button remains available for on-screen guidance. This layout ensures that results are both interpretable at a glance and actionable within existing clinical workflows.
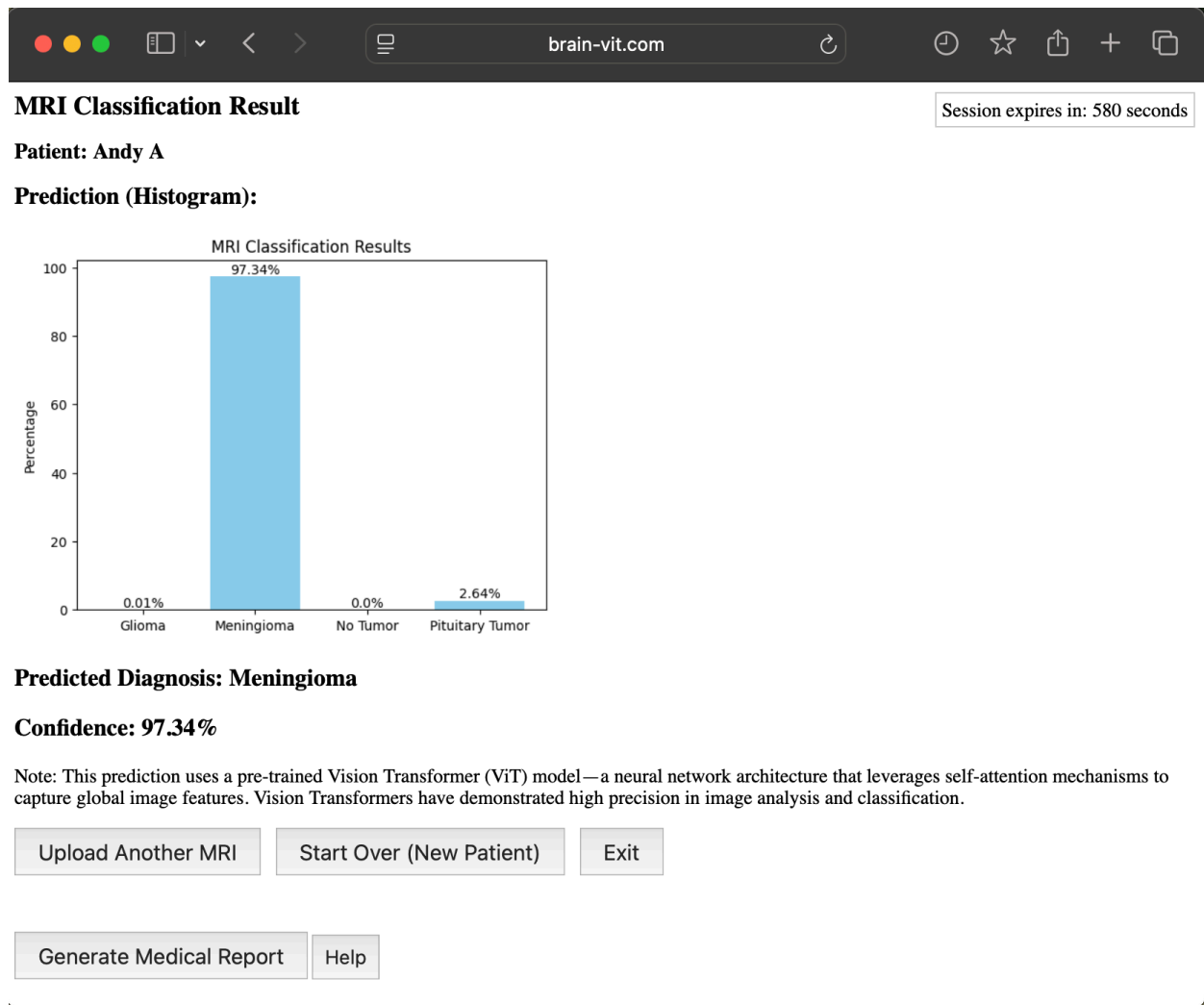
Medical Report Generation

After reviewing results on screen, clinicians can click **Generate Medical Report** to download a comprehensive, stand-alone HTML file (e.g., `medical_report.html`) that consolidates all session data into a single document (Figure 10). The report is organized into clear, titled sections:

- **Patient Information**: Displays name (and any additional identifiers, if enabled).

- **Medical History**: Lists each questionnaire response (head trauma, family history, seizures, genetic disorders).
- **Combined Test Results**: Shows genetic and laboratory values, with abnormal entries highlighted in red and normal ones in green according to configurable thresholds.
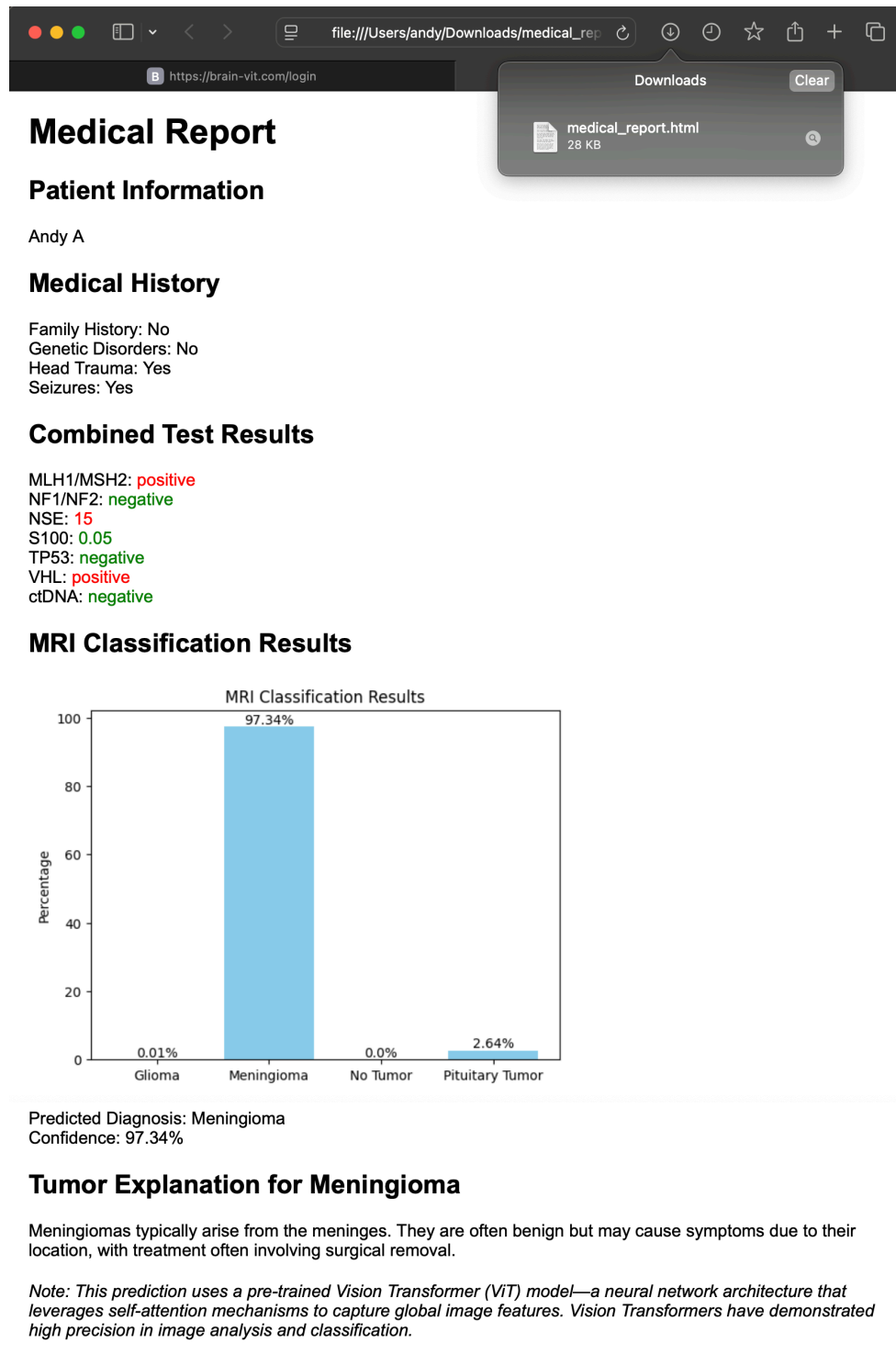
- **MRI Classification Results**: Embeds the same probability histogram from the live result screen, followed by the predicted diagnosis and confidence score.
- **Tumor Explanation**: Provides a concise, plain-language description of the predicted tumor type (e.g., "Meningiomas typically arise from the meninges…"), helping clinicians and patients understand the finding.
- **Methodology Note**: A final italicized paragraph reminds readers that the diagnosis stems from a pretrained Vision Transformer model leveraging self-attention for image analysis.

This report format not only archives all diagnostic steps and results in a portable file but also supports customization—for instance, adding hospital letterhead, adjusting color schemes, or including additional explanatory text—by editing the Flask template used in the `/generate_report` route.

7.2 AWS Deployment

The production environment for the Brain Tumor Diagnosis System is hosted on an AWS EC2 p3.2xlarge instance, selected for its NVIDIA V100 GPU to ensure real-time inference (under two seconds per MRI) (Amazon Web Services, Inc., n.d.). To give the application a professional presence, a custom domain—brain-vit.com—was purchased via GoDaddy, and DNS records (A and CNAME) were then managed in Amazon Route 53, pointing the domain to the EC2 instance's public IP and enabling smooth, globally distributed name resolution (GoDaddy, n.d.). Inside the EC2 instance, the Flask application is Dockerized to guarantee consistency across development, staging, and production. A Gunicorn WSGI server runs multiple worker processes within the container to handle concurrent requests efficiently, while Nginx on the host machine functions as a reverse proxy. Nginx serves static assets, buffers client connections, and routes HTTPS traffic to Gunicorn, ensuring robust performance and simplified load management.

To protect patient data in transit, HTTPS is enforced across all endpoints. TLS certificates are obtained from Let's Encrypt via Certbot and automatically renewed, with Nginx configured to redirect all HTTP traffic to HTTPS. This setup ensures that MRI images and patient details are encrypted end-to-end from the clinician's browser to the application server.

Operational visibility is provided by AWS CloudWatch, which collects system-level metrics (CPU, GPU, and memory utilization), Docker container logs, and custom health-check

endpoints. Dashboards display real-time performance data, and CloudWatch Alarms notify administrators of unusual spikes in latency or error rates, enabling rapid troubleshooting. Security is enforced through IAM instance profiles and security groups. The EC2 role grants only the minimal permissions necessary—access to CloudWatch, S3 (for optional archival), and Secrets Manager—following the principle of least privilege. Network security groups permit inbound traffic exclusively on port 443 (HTTPS) and port 22 (SSH) limited to specific administrative IP ranges, effectively minimizing the attack surface.

By combining GoDaddy domain management, Route 53 DNS, Dockerized microservices, Gunicorn/Nginx orchestration, automated TLS provisioning, CloudWatch monitoring, and strict IAM/network policies, this AWS deployment delivers a scalable, secure, and maintainable platform suited for clinical integration.

8 Discussion

Vision Transformer–based models offer several key advantages over traditional diagnostic approaches. By dividing MRI scans into patch embeddings and applying multi-head self-attention, ViTs capture both local and global image context—enabling the model to recognize subtle patterns that may elude human observers or convolutional architectures limited by local receptive fields (Dosovitskiy et al., 2021). In contrast to manual image review—which can vary between radiologists and suffer from fatigue—this automated approach delivers consistent, reproducible analyses, reducing inter-observer variability and supporting more standardized decision-making in radiology.

Real-time inference (< 2 s per scan) means the AI system can integrate seamlessly into clinical workflows without delaying patient care (Amazon Web Services, Inc., n.d.). Fast turnaround allows clinicians to receive immediate decision support at the point of care, facilitating earlier intervention and potentially improving outcomes. Moreover, the web-based interface requires only a standard browser—minimizing training overhead and ensuring accessibility in diverse healthcare settings.

Importantly, these emerging AI technologies are not intended to replace physicians but to augment their expertise. By flagging high-risk scans and highlighting areas of uncertainty, machine learning models act as a second reader—helping to catch cases that might otherwise be

missed and mitigating cognitive biases inherent in human judgment. The goal is faster, more accurate tumor detection and fewer wrong diagnoses, while preserving the clinician's central role in patient management and treatment decisions.

9 Future Work

To translate this prototype into a robust, clinically deployed system, several key extensions are planned:

- **Seamless EHR Integration and Persistent Storage**
  Replace the in-memory SQLite store with a secure, standards-compliant database (e.g., AWS RDS or a FHIR-compatible data store) that automatically ingests patient demographics, medical history, lab results, and imaging metadata from hospital EHR systems. This will eliminate manual entry, enable audit trails, and provide richer longitudinal records for AI-assisted care.

- **Continuous Data Collection & Model Retraining**
  Establish a pipeline for anonymized capture of each new case—imaging, confirmed diagnoses, outcomes, and follow-up data. Periodic retraining or fine-tuning on this ever-growing dataset will improve model generalization across scanners, protocols, and patient populations, while guarding against concept drift.

- **Expanded Diagnostic Scope**
  Broaden the classification schema to include additional brain tumor types (e.g., lymphoma, metastatic lesions) and important molecular markers (such as IDH mutation, MGMT methylation, or 1p/19q codeletion status). Fusing these genomic and histopathological features with imaging inputs can yield more precise, personalized diagnostic and prognostic insights.

- **Prospective Clinical Validation & User Studies**
  Launch multi-center, prospective trials to assess real-world performance and clinical impact. Parallel usability studies with radiologists, neuro-oncologists, and technologists will evaluate workflow integration, trust, and decision-support value—informing UI refinements, alert thresholds, and governance processes.

- **Advanced Explainability & Decision Support**

  Integrate interpretability techniques—such as Grad-CAM to visualize salient image regions, and SHAP values to quantify the influence of non-imaging features—providing case-specific explanations alongside each prediction. These insights will empower clinicians to validate AI outputs, facilitate patient communication, and meet regulatory requirements for transparency.

By pursuing these enhancements—data interoperability, continuous learning, diagnostic breadth, rigorous validation, and transparent explanations—the system can evolve into a clinically trusted tool that accelerates early tumor detection, reduces diagnostic variability, and supports more informed treatment planning.

10 Conclusion

This capstone has developed and demonstrated a secure, scalable AI system for brain tumor diagnosis by integrating a Vision Transformer–based imaging model with a guided clinical interface. Through a stepwise Flask application—spanning secure login, patient profile management, risk-factor questionnaires, conditional MRI recommendation, and rapid (<2 s) MRI classification—the system closely mirrors a clinician's diagnostic workflow while automating image analysis for greater consistency and speed (Flask Documentation, n.d.; Amazon Web Services, Inc., n.d.).

The core Vision Transformer model, trained and evaluated on a four-class brain MRI dataset, achieved **93.44 % test accuracy** with high recall across glioma, meningioma, no tumor, and pituitary tumor categories. These results underscore the model's ability to capture both local and global scan features, reducing inter-observer variability compared to manual review. By embedding the pretrained ViT into a user-friendly web interface, clinicians receive immediate visual and quantitative feedback—complete with confidence scores, probability histograms, and downloadable HTML reports—supporting fast, informed decision-making.

Deploying the application on an AWS EC2 p3.2xlarge instance behind Docker, Gunicorn, and Nginx ensures high availability and GPU-accelerated inference, while HTTPS encryption (via Let's Encrypt and Certbot) and AWS IAM/security group policies safeguard patient data in

transit and at rest. CloudWatch monitoring provides real-time observability of system health and resource utilization, enabling proactive maintenance and scalability as clinical demand grows. While initial results are promising, the system's reliance on simulated lab and genetic inputs and an in-memory database highlights the need for integration with real-world EHR systems and persistent, HIPAA-compliant storage. Future efforts will focus on expanding tumor categories, incorporating authentic genomic markers, conducting prospective multi-center validation, and adding interpretability modules (e.g., Grad-CAM, SHAP) to explain individual predictions. By continuously retraining on new, anonymized cases and refining the user experience based on clinician feedback, this platform can evolve into a trusted tool that accelerates early tumor detection, minimizes diagnostic errors, and ultimately enhances patient care.

References

Amazon Web Services, Inc. (n.d.). AWS Documentation. Retrieved from https://aws.amazon.com/documentation/

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In ICLR 2021. Retrieved from https://arxiv.org/abs/2010.11929

Filatov, D., & Yar, G. N. a. H. (2022, July 27). Brain tumor diagnosis and classification via Pre-Trained Convolutional Neural Networks. arXiv.org. https://arxiv.org/abs/2208.00768?

Flask Documentation. (n.d.). Retrieved from https://flask.palletsprojects.com/

GoDaddy. (n.d.). GoDaddy website. Retrieved from https://www.godaddy.com

Pallets Projects. (n.d.). *Flask documentation (Version 3.1.x)*. https://flask.palletsprojects.com/

Lucidrains. (2020). vit-pytorch [GitHub repository]. Retrieved from https://github.com/lucidrains/vit-pytorch

Nickparvar, M. (2021, September 24). Brain Tumor MRI Dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

OpenAI. (2025). ChatGPT [Large language model]. https://chat.openai.com

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems (Vol. 32, pp. 8026–8037). Retrieved from https://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library

Python Software Foundation. (n.d.). Python documentation. Retrieved from https://docs.python.org/3/

TorchVision. (n.d.). TorchVision documentation. Retrieved from https://pytorch.org/vision/stable/index.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. In *Advances in Neural Information Processing Systems* (Vol. 30). https://arxiv.org/abs/1706.03762

Appendix

A. Project Repository

All source materials and documentation for this capstone are hosted in a public GitHub

repository:

https://github.com/aandy2014/Capstone-Project-GCU-ViT-for-Brain-Tumor-Classification.git

This repository includes:

- **ViT Training Code**: Jupyter notebooks and scripts used to preprocess MRI data and train the Vision Transformer model.
- **Flask Deployment Code**: The complete Flask application that integrates patient management, questionnaire modules, MRI upload, and real-time inference.
- **Capstone Manuscript**: A digital copy of the full written report.
- **User Guide**: Step-by-step instructions for clinicians on using the web application.
- **Presentation Slides**: PDF of the capstone presentation deck.

B. Presentation Video

A recording of the capstone project presentation is available on YouTube:

https://youtu.be/FLZg4wXEAdM