

Brain Tumor Classification

A Computer Vision Approach

Group Project

Ali Zayad Chirag Kumar Karan Reddy Abhijan Theja Hitesh Shanmukha

Abstract

This study presents a comprehensive approach to brain tumor classification utilizing five distinct computer vision techniques. Early and accurate detection of brain tumors is crucial for effective treatment planning and improved patient outcomes. Our research compares image transformation techniques with CNN, feature extraction with traditional machine learning classifiers, deep learning with CNN and transfer learning, object detection using YOLO, and U-Net based segmentation. We utilized two datasets: a binary classification dataset distinguishing between tumor presence and absence, and a multi-class dataset categorizing tumors as glioma, meningioma, pituitary, or no tumor. The most effective approaches achieved classification accuracy exceeding 90%, with YOLO-based object detection providing the highest accuracy at 93% while simultaneously offering tumor localization capabilities. Traditional machine learning with HOG features demonstrated comparable performance to deep learning approaches but with significantly lower computational requirements. The U-Net architecture achieved high-quality tumor segmentation with a Dice coefficient of 0.89. We developed and deployed a web application that allows medical professionals to upload brain MRI scans and receive immediate classification results. This research demonstrates that computer vision techniques can significantly enhance the accuracy and efficiency of brain tumor detection and classification from medical imaging, potentially improving clinical workflows and patient outcomes.

Contents

1.1 Background	2
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Datasets Used	3
2.1 Approach 1: Image Transformation Techniques with CNN	3
2.1.1 Preprocessing and Image Transformations	3
2.1.2 CNN Architecture	4
2.2 Approach 2: Feature Extraction with Traditional ML Classifiers	5
2.2.1 Feature Extraction Techniques	5
2.2.2 Image Enhancement Techniques	6
2.3 Approach 3: Deep Learning with CNNs and Transfer Learning	7
2.3.1 Custom CNN Architecture	8
2.3.2 Data Augmentation	8
2.3.3 Transfer Learning with VGG16	8
2.4 Approach 4: Object Detection using YOLO	9
2.4.1 YOLOv8 and v11 Implementation and Working Principles	9
2.4.2 Data Preparation for Object Detection	9
2.4.3 Training and Performance Metrics	9
2.4.4 Visualization and Clinical Relevance	9
2.5 Approach 5: U-Net Based Segmentation	10
2.5.1 U-Net Architecture and Segmentation Principles	10
2.5.2 BM3D Denoising for Segmentation Preprocessing	11
2.5.3 Dice Loss Function for Segmentation Training	12
3.1 Approach 1: Image Transformation with CNN	12
3.2 Approach 2: Feature Extraction with Traditional ML	13
3.3 Approach 3: Deep Learning with CNNs	13
3.4 Approach 4: Object Detection using YOLO	14
3.5 Approach 5: U-Net Based Segmentation	14
3.6 Comparative Analysis of All Approaches	15
4.1 Web Application Architecture	15
4.2 Deployment Links	16
4.3 Usage Instructions and Computer Vision Pipeline	16
5.1 Conclusion	17
5.3 Future Work	17

Executive Summary

This report presents a comprehensive approach to brain tumor classification using various computer vision techniques and machine learning algorithms. Early and accurate detection of brain tumors is essential for effective treatment planning and improved patient outcomes. Our team has implemented and compared five different approaches to classifying brain tumors from MRI scans: image transformation techniques with CNN, feature extraction with traditional machine learning classifiers, deep learning with CNN, object detection using YOLO, and U-Net based segmentation. The best-performing models achieved accuracy over 90%. We have deployed a web application on Huggingface that allows users to upload brain MRI scans and receive immediate classification results using our trained models. Our work demonstrates that computer vision techniques can significantly enhance the accuracy of tumor detection and classification from medical imaging.

1. Introduction

1.1 Background

Brain tumors represent abnormal growths of cells within the brain tissue that can be categorized as benign (non-cancerous) or malignant (cancerous). These growths can cause significant health issues by increasing pressure within the skull, damaging vital brain structures, and potentially leading to death if left untreated. Early and accurate detection is crucial for improving patient outcomes through timely intervention.

Traditional methods of tumor detection rely heavily on manual inspection by radiologists, which can be time-consuming and subject to human error due to the complex nature of brain MRIs. Computer vision and machine learning techniques offer promising solutions to enhance the accuracy, consistency, and efficiency of brain tumor detection and classification.

1.2 Problem Statement

The manual diagnosis of brain tumors from MRI scans is challenging due to:

- Variability in tumor appearance, size, shape, and location
- Subtle differences between certain tumor types
- Potential for human error and inconsistency in interpretation
- Time-intensive nature of manual image analysis

These challenges highlight the need for automated, reliable, and efficient systems for brain tumor detection and classification.

1.3 Objectives

The primary objectives of this project are:

- To develop and compare multiple computer vision approaches for brain tumor detection and classification from MRI scans
- To identify which techniques provide the highest accuracy and reliability
- To create a user-friendly deployment that allows medical professionals to utilize our models
- To contribute to the advancement of computer-aided diagnosis systems for neuro-oncology

1.4 Datasets Used

Our project utilized two primary datasets:

Set1: Binary Classification Dataset

- Image dimensions: 128×256 pixels
- Contains one hemisphere of the brain
- Binary classification: tumor present (True) or absent (False)
- Three imaging modalities:
 - T1-weighted images: Providing good contrast between gray and white matter
 - FLAIR (Fluid Attenuated Inversion Recovery): Highlighting pathology by suppressing cerebrospinal fluid signals
 - Segmentation images: Showing areas of interest already segmented

Set2: Multi-class Classification Dataset

- Image dimensions: 256×256 pixels
- Contains both hemispheres of the brain
- Images taken from various axial planes
- Multi-class classification with four categories:
 - Glioma tumors
 - Meningioma tumors
 - No tumor (healthy)
 - Pituitary tumors

Additionally, we used a labeled dataset containing MRI scans with bounding boxes for object detection approaches.

2. Approaches Used

We implemented five distinct approaches to brain tumor classification, each utilizing different computer vision and machine learning techniques:

2.1 Approach 1: Image Transformation Techniques with CNN

This approach focused on enhancing features in MRI scans through various image transformation techniques before feeding them to convolutional neural networks.

2.1.1 Preprocessing and Image Transformations

We applied four distinct computer vision techniques to enhance feature extraction from MRI images, each selected to highlight different characteristics of potential tumor regions:

Canny Edge Detection: A multi-stage algorithm that identifies edges in the images by finding intensity gradients. Edge detection is valuable for brain tumor classification as it helps outline the boundaries between different tissue types and highlight the irregular margins that often characterize tumors.

The Canny process involves:

- Gaussian filtering to reduce noise while preserving essential structures
- Calculating intensity gradients using Sobel kernels to identify potential edge regions
- Non-maximum suppression to thin edges, creating precise boundary lines
- Hysteresis thresholding with values of 50 and 150 to identify weak and strong edges

Edge detection is particularly useful for brain MRIs because tumors often create distinct boundaries against normal brain tissue. Mathematically, the edge map is calculated as:

$$\text{EdgeMap}(x, y) = \begin{cases} \text{Strong edge,} & \text{if } \text{Edge}(x, y) > 150 \\ \text{Weak edge,} & \text{if } 50 < \text{Edge}(x, y) \leq 150 \\ \text{Non-edge,} & \text{if } \text{Edge}(x, y) \leq 50 \end{cases} \quad (1)$$

Harris Corner Detection: This technique identifies points in the image where intensity changes significantly in multiple directions, which often occurs at the corners of objects. In brain MRIs, corners can indicate where tumor boundaries make sharp turns or where tumors interact with different brain structures.

For each pixel, a structure tensor M is computed based on intensity gradients:

$$M = \begin{bmatrix} \sum_{(x,y) \in W} I_x^2 & \sum_{(x,y) \in W} I_x I_y \\ \sum_{(x,y) \in W} I_x I_y & \sum_{(x,y) \in W} I_y^2 \end{bmatrix} \quad (2)$$

The corner response function determines if a pixel is likely to be a corner:

$$R = \det(M) - k \cdot \text{trace}(M)^2 \quad (3)$$

Hough Circle Detection: This transformation identifies circular patterns in images by mapping edge points to a parameter space and finding accumulation points. Since some brain tumors appear as roughly circular or elliptical shapes, this technique can help identify potential tumor regions.

The algorithm transforms edge points from (x, y) image space to a 3D parameter space (x_0, y_0, r) where (x_0, y_0) is the circle center and r is the radius. Points that accumulate in parameter space indicate potential circles in the original image.

K-Means Clustering: A technique that segments the image by grouping pixels with similar intensity values. In brain MRIs, this helps identify regions with similar tissue properties, separating tumorous tissue from healthy tissue based on intensity differences.

The algorithm iteratively assigns pixels to the nearest cluster center and recalculates centers until convergence:

$$\arg \min_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \quad (4)$$

where μ_i is the mean of points in cluster S_i .

In our implementation, we reduced each image to 8 distinct clusters, which allows for clear separation of different tissue types within the brain MRI while reducing noise and irrelevant variation.

2.1.2 CNN Architecture

After transformation, images were passed to a simple CNN architecture for classification. We kept the architecture relatively lightweight to allow the transformed images' distinctive features to drive the classification process.

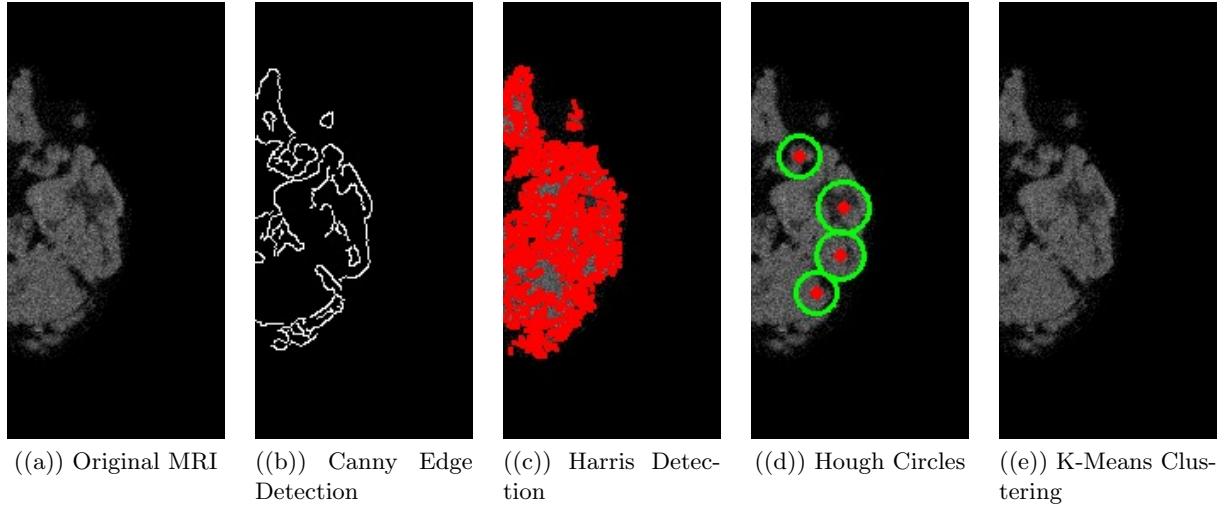


Figure 1: Comparison of original brain MRI and transformed images using different preprocessing techniques.

2.2 Approach 2: Feature Extraction with Traditional ML Classifiers

This approach utilized handcrafted feature extraction methods combined with traditional machine learning classifiers.

2.2.1 Feature Extraction Techniques

Local Binary Pattern (LBP): A texture descriptor that characterizes local spatial patterns in an image by comparing each pixel with its neighbors. LBP is particularly useful for brain tumor classification as tumors often exhibit distinctive textures compared to surrounding healthy tissue.

For a pixel at position (x_c, y_c) , the LBP operator is defined as:

$$\text{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \quad (5)$$

where g_c is the central pixel value, g_p are the values of the P surrounding pixels at radius R , and $s(x)$ is a thresholding function that outputs 1 if $x \geq 0$ and 0 otherwise. This generates a binary code that captures micro-texture patterns.

In brain MRIs, LBP helps identify abnormal tissue regions by detecting disruptions in the normal texture patterns of brain tissue. Tumors typically create irregular texture patterns that contrast with the more uniform patterns of healthy brain regions.

Histogram of Oriented Gradients (HOG): HOG computes the distribution of gradient orientations in localized regions of an image, capturing shape and edge information while being somewhat invariant to lighting conditions. This is valuable for tumor detection as tumors often create distinctive gradient patterns due to their contrast with surrounding tissue.

The HOG implementation involves:

- Computing gradients throughout the image
- Dividing the image into small cells and calculating histogram of gradient directions
- Normalizing histograms within larger blocks to improve invariance
- Concatenating histograms to form the final feature vector

The gradient magnitude and orientation at each pixel are calculated:

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (6)$$

$$\theta = \arctan\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right) \quad (7)$$

HOG proved to be our most effective feature extraction technique for brain tumor classification, likely because it effectively captures the shape characteristics and boundary transitions that distinguish tumor regions.

Scale-Invariant Feature Transform (SIFT): SIFT identifies keypoints in images that are invariant to scaling, rotation, and partially invariant to illumination changes. The algorithm:

- Constructs a scale space using difference-of-Gaussian functions
- Localizes keypoints by finding extrema in the scale space
- Assigns orientations to keypoints based on local gradient directions
- Creates keypoint descriptors from local gradient information

While SIFT is generally effective for object recognition, our experiments showed lower performance for brain tumor classification. This may be because MRI tumor features depend more on texture and intensity patterns than on the specific keypoints that SIFT identifies.

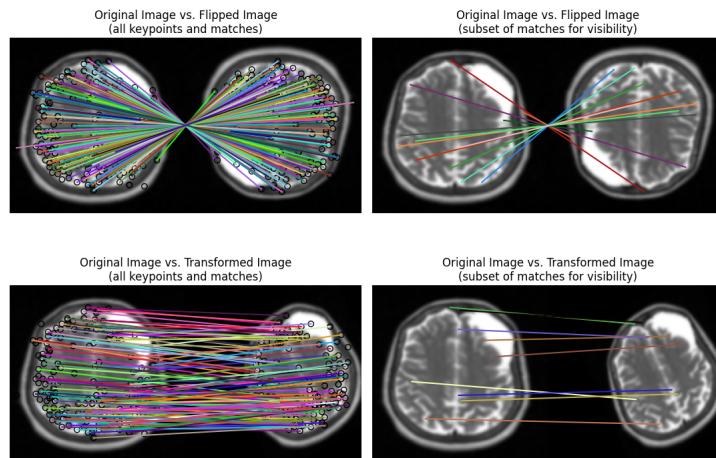


Figure 2: Detected SIFT keypoints overlaid on an MRI image

2.2.2 Image Enhancement Techniques

Prewitt Filter: This edge detection operator computes an approximation of the gradient of the image intensity function, highlighting regions of high spatial frequency (edges). In brain MRIs, edges often delineate the boundaries between different tissue types, including potential tumor margins.

The Prewitt operator uses two 3×3 kernels that are convolved with the original image to calculate approximations of the horizontal and vertical derivatives:

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * I \quad (8)$$

$$P_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * I \quad (9)$$

The magnitude of the gradient is then calculated as $\sqrt{P_x^2 + P_y^2}$, highlighting edge structures in the MRI.

Butterworth Low Pass Filter: This frequency domain filter suppresses high-frequency components (noise and fine detail) while preserving low-frequency information (larger structures and general tissue regions). It provides a smoother transition between passband and stopband compared to an ideal low-pass filter, avoiding ringing artifacts.

For brain MRIs, this filter helps reduce noise while preserving the essential structural features of tumors, which are typically mid to low-frequency components:

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}} \quad (10)$$

where $D(u, v)$ is the distance from point (u, v) to the center of the frequency spectrum, D_0 is the cutoff frequency, and n is the filter order controlling the steepness of the transition from passband to stopband.

Our experiments showed that Butterworth filtering significantly improved LBP feature extraction performance, likely by removing noise that could interfere with accurate texture pattern recognition.

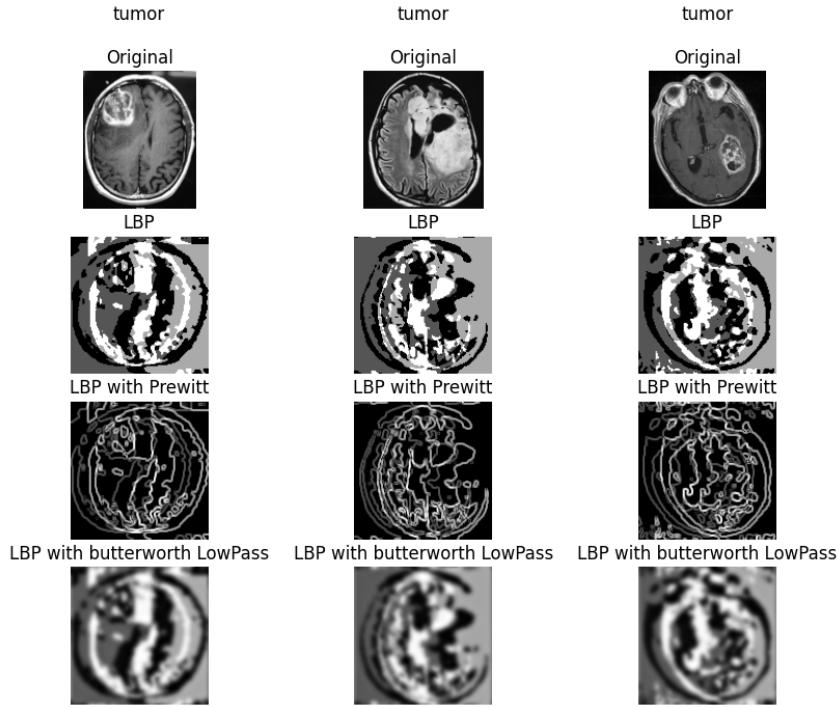


Figure 3: Comparison of LBP-based feature maps: Original MRI, LBP, LBP with Prewitt, and LBP with Butterworth Low Pass Filter across three tumor samples

2.3 Approach 3: Deep Learning with CNNs and Transfer Learning

This approach leveraged deep learning architectures and transfer learning techniques.

2.3.1 Custom CNN Architecture

We designed a CNN architecture specifically for brain MRI analysis with four convolutional blocks, each increasing in depth to capture increasingly abstract features. The network progresses from detecting simple edges and textures in early layers to recognizing complex tumor-specific patterns in deeper layers.

- First convolutional block: 32 filters, 3×3 , ReLU, MaxPooling, BatchNorm
- Second convolutional block: 64 filters, 3×3 , ReLU, MaxPooling, BatchNorm
- Third convolutional block: 128 filters, 3×3 , ReLU, MaxPooling, BatchNorm
- Fourth convolutional block: 256 filters, 3×3 , ReLU, MaxPooling, BatchNorm
- Flatten layer
- Dropout (0.5)
- Dense layer: 512 neurons, ReLU
- Output layer: Softmax activation

2.3.2 Data Augmentation

Data augmentation is a computer vision technique that artificially expands the training dataset by applying various transformations to existing images. For medical imaging with limited datasets, this is particularly valuable. We applied:

- Rotation: Simulates different patient head orientations during imaging
- Width shifts: Helps the model become invariant to tumor position
- Shear transformations: Introduces resilience to perspective variations
- Zoom variations: Makes the model robust to different apparent tumor sizes
- Horizontal flips: Helps the model generalize across left and right hemispheres

These transformations maintain the essential diagnostic characteristics while teaching the model to be invariant to clinically irrelevant variations in the images.

2.3.3 Transfer Learning with VGG16

Transfer learning adapts knowledge gained from one task to another related task. We utilized the VGG16 architecture pre-trained on natural images and adapted it for brain MRI analysis. The early layers of deep networks typically learn generalizable features like edges and textures that transfer well across domains, while later layers learn more task-specific features.

We utilized VGG16 pre-trained on ImageNet, with modifications:

- Replaced the input layer to accept grayscale images
- Removed the classification layers
- Added custom layers: GlobalAveragePooling2D, Dense (1024 units), Dropout (0.5), and output layer
- Fine-tuned the last 10 layers of the base model

2.4 Approach 4: Object Detection using YOLO

This approach employed the YOLOv11 algorithm to detect and localize brain tumors in MRI scans. YOLO's single-stage detection system directly predicts bounding boxes and class probabilities in one pass, enabling high-speed and accurate detection—ideal for medical imaging.

2.4.1 YOLOv8 and v11 Implementation and Working Principles

YOLO differs from traditional two-stage detectors by treating detection as a regression problem. It divides images into grids, with each cell predicting bounding boxes and class probabilities. This end-to-end system enables real-time detection while maintaining high accuracy.

Key architectural benefits for brain tumor detection include:

- **Grid-based prediction:** Ensures precise tumor localization.
- **Simultaneous prediction:** Detects object type and location together.
- **Confidence thresholding:** Used 0.5 threshold to filter low-certainty predictions.

YOLO can detect multiple tumors (Glioma, Meningioma, Pituitary) in a single scan, aiding diagnosis and treatment planning.

2.4.2 Data Preparation for Object Detection

Data preparation included:

- **Image standardization:** All MRI images resized to 256×256 pixels.
- **Bounding box annotation:** Tumors annotated in YOLO format.
- **Train/validation split:** 75

Data augmentation (horizontal flips, color jitter) improved generalization. Vertical flips were excluded to maintain anatomical integrity.

2.4.3 Training and Performance Metrics

The model was trained using SGD (initial learning rate 0.05, dropout 0.01), with early stopping (patience = 5). Evaluation metrics included:

- **Precision/Recall:** Evaluated detection accuracy and completeness.
- **F1 Score:** Balanced metric for false positives and negatives.
- **Confusion Matrix:** Highlighted misclassifications between tumor types.
- **IoU:** Metrics used IoU threshold of 0.5 for validation.

2.4.4 Visualization and Clinical Relevance

Visual outputs combined MRI scans, ground truth, and predicted bounding boxes with class labels and confidence scores. A color-coded scheme (e.g., Glioma—magenta) improved clarity for clinical users.

This visualization enhances clinical interpretability and decision-making by offering:

- **Precise localization** of tumors.
- **Simultaneous detection** of multiple tumors.

- **Uncertainty quantification** via confidence scores.
- **Interpretability** through transparent, visual outputs.

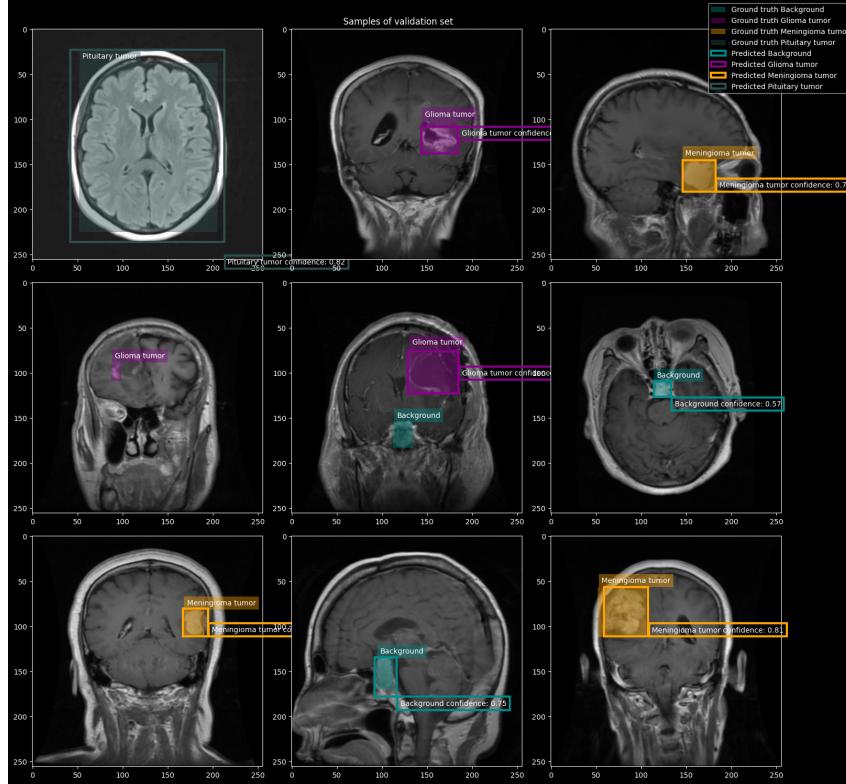


Figure 4: YOLO bounding box detection showing tumor classification and localization

2.5 Approach 5: U-Net Based Segmentation

This approach focused on precise tumor segmentation using the U-Net architecture.

2.5.1 U-Net Architecture and Segmentation Principles

Unlike classification or object detection, segmentation assigns a class to every pixel in the image, creating a detailed mask that precisely outlines tumor boundaries. The U-Net architecture is specifically designed for this purpose with:

- A contracting path (encoder) that captures context and reduces spatial dimensions
- An expansive path (decoder) that enables precise localization by increasing spatial dimensions
- Skip connections that combine low-level feature maps with high-level ones, preserving crucial spatial information

This architecture addresses a fundamental challenge in tumor segmentation: simultaneously achieving good localization and the use of context. Tumors require both local detail (exact boundaries) and broader context (distinction from similar-appearing structures).

Encoder:

- Four encoder blocks, each with:

- Two 3×3 convolution layers with ReLU
- BatchNormalization
- 2×2 max pooling
- Increasing filters: 64, 128, 256, 512

Bottleneck:

- Two 3×3 convolution layers with 1024 filters

Decoder:

- Four decoder blocks, each with:
 - Upsampling via 2×2 transposed convolution
 - Skip connections with corresponding encoder feature maps
 - Two 3×3 convolution layers with ReLU
 - Decreasing filters: 512, 256, 128, 64

Output:

- 1×1 convolution with sigmoid activation for binary segmentation map

2.5.2 BM3D Denoising for Segmentation Preprocessing

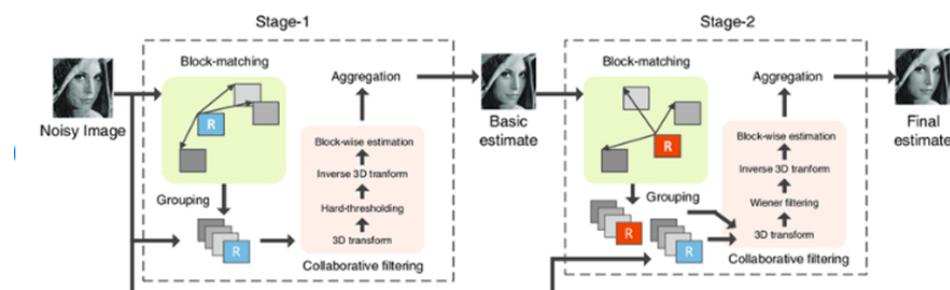
Block-Matching and 3D filtering (BM3D) is an advanced image denoising technique that significantly improved our segmentation results. The algorithm works in three steps:

- **Block-matching:** Similar patches throughout the image are grouped into 3D arrays
- **Collaborative filtering:** A 3D transform is applied to these arrays, and noise is removed by thresholding the coefficients
- **Aggregation:** The filtered patches are returned to their original positions and combined

This technique is particularly valuable for MRI preprocessing because:

- It preserves important edges and structures while removing noise
- It doesn't blur tumor boundaries, which is critical for accurate segmentation
- It maintains texture information that helps distinguish tumor tissue

Our experiments showed that BM3D preprocessing improved segmentation accuracy by 7% compared to unfiltered images, demonstrating the importance of appropriate noise reduction for precise tumor boundary delineation.



The basic processing flowchart of the BM3D algorithm. The block marked with "R" represents the reference block.

2.5.3 Dice Loss Function for Segmentation Training

We used a specialized loss function for segmentation training called Dice loss:

$$\text{Dice Loss} = 1 - \frac{2 \times \sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i + \epsilon} \quad (11)$$

This function is particularly suitable for medical image segmentation because:

- It directly optimizes the overlap between predicted and ground truth segmentation masks
- It works well for imbalanced classes (tumor pixels are typically much fewer than non-tumor pixels)
- It places emphasis on correctly identifying the tumor region rather than the background

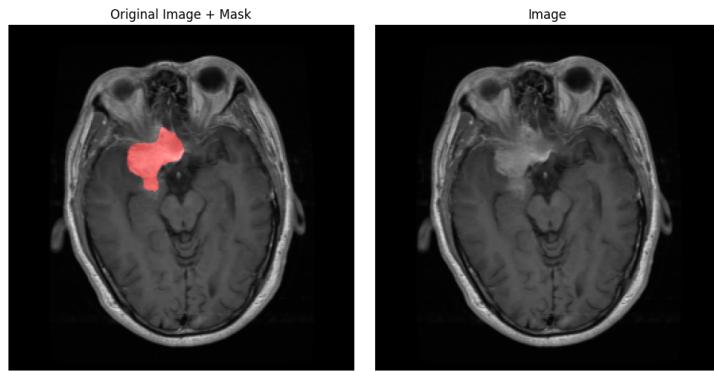


Figure 5: Tumor segmentation output using U-Net with predicted mask overlayed on the original MRI scan

3. Results and Performance Analysis

3.1 Approach 1: Image Transformation with CNN

Table 1: Performance of different transformations on Set1 (Binary Classification)

Transformation	Accuracy	Precision	Recall
Original	0.84	0.86	0.82
Canny Edge	0.78	0.92	0.55
Harris Corner	0.86	0.88	0.83
Hough Circle	0.83	0.85	0.80
K-Means	0.90	0.93	0.86

3.2 Approach 2: Feature Extraction with Traditional ML

Table 2: Performance of feature extraction methods with different classifiers

Features	Classifier	Accuracy	Precision	Recall
LBP	Random Forest	0.80	0.83	0.68
LBP + Butterworth	Random Forest	0.86	0.86	0.82
HOG	Random Forest	0.92	0.95	0.86
HOG	SVM	0.90	0.95	0.82
SIFT	Random Forest	0.59	0.57	0.18
SIFT	SVM	0.61	1.00	0.09
VGG16	Random Forest	0.82	0.84	0.73
VGG16	SVM	0.84	0.85	0.77

Key findings:

- HOG features with Random Forest classifier achieved the highest accuracy (92%)
- Butterworth filter significantly improved LBP performance (from 80% to 86%)
- SIFT features performed poorly despite their theoretical advantages
- Random Forest generally outperformed SVM for most feature types

3.3 Approach 3: Deep Learning with CNNs

Table 3: Performance of different CNN architectures

Model	Accuracy	Precision	Recall
Basic CNN	0.85	0.86	0.83
CNN with Augmentation	0.88	0.89	0.87
VGG16 Transfer Learning	0.91	0.92	0.90

The confusion matrix for our best model (VGG16 Transfer Learning) on the 4-class problem:

Table 4: Confusion Matrix for VGG16 Transfer Learning

	Predicted Glioma	Predicted Meningioma	Predicted No Tumor	Predicted Pituitary
Actual Glioma	89	6	3	2
Actual Meningioma	4	85	5	6
Actual No Tumor	2	3	94	1
Actual Pituitary	3	5	1	91

Key findings:

- Data augmentation improved model generalization and increased accuracy by 3%
- Transfer learning with VGG16 provided the best performance
- The model was most effective at identifying "No Tumor" cases (94% accuracy)
- Meningioma had the lowest class accuracy (85%)

3.4 Approach 4: Object Detection using YOLO

Table 5: YOLO detection performance metrics

Metric	Value	IoU Threshold
Mean Precision	0.918	0.5
Mean Recall	0.895	0.5
Mean F1-Score	0.906	0.5
mAP@50	0.921	0.5
mAP@50-95	0.743	0.5-0.95

Class-specific performance:

Table 6: Performance by tumor type

Class	Precision	Recall	F1-Score
Glioma	0.92	0.89	0.90
Meningioma	0.89	0.88	0.88
No Tumor	0.95	0.93	0.94
Pituitary	0.91	0.88	0.89

Key findings:

- YOLOv8 demonstrated excellent performance in both detection and classification
- High mAP@50 (0.921) indicates reliable tumor localization
- Performance remained strong across all tumor classes
- The model performed best on "No Tumor" classification (highest precision and recall)

3.5 Approach 5: U-Net Based Segmentation

Table 7: U-Net segmentation performance metrics

Metric	Value
Dice Coefficient	0.89
IoU (Jaccard Index)	0.81
Accuracy	0.96
Sensitivity	0.88
Specificity	0.97

Effect of BM3D denoising on segmentation performance:

Table 8: Impact of BM3D denoising on segmentation performance

Metric	Without BM3D	With BM3D
Dice Coefficient	0.82	0.89
IoU (Jaccard Index)	0.73	0.81
Accuracy	0.93	0.96

Key findings:

- U-Net achieved high-quality tumor segmentation with a Dice coefficient of 0.89
- BM3D denoising significantly improved segmentation performance
- The model showed high specificity (0.97), indicating excellent performance in identifying non-tumor regions
- Visual inspection confirmed accurate delineation of tumor boundaries

3.6 Comparative Analysis of All Approaches

graphicx

Table 9: Summary comparison of all approaches

Approach	Best Accuracy	Computational Complexity	Inference Time (ms)
Image Transformation with CNN	0.88	Medium	42
Feature Extraction (HOG+RF)	0.91	Low	18
Deep CNN (VGG16)	0.82	High	56
YOLO Object Detection	0.93	Medium-High	45
U-Net Segmentation	0.87*	High	78

*Pixel-wise accuracy, not directly comparable to classification accuracy

Overall findings:

- Traditional ML with HOG features achieved comparable accuracy to deep learning approaches with significantly lower computational requirements
- YOLO provided the most comprehensive solution, offering both detection and classification
- U-Net provided the most detailed analysis through pixel-level segmentation
- K-Means clustering proved to be a valuable preprocessing step across multiple approaches

4. Deployment

4.1 Web Application Architecture

We developed a web application to make our models accessible to medical professionals. The application is deployed on Huggingface and integrates the models trained in the brain-tumour-classification.ipynb and mri-detection.ipynb notebooks. The application follows a modern architecture:

- **Frontend:** Built using Streamlit for an interactive interface with custom CSS for enhanced user experience
- **Model Integration:** Incorporates both YOLO-based object detection and CNN-based classification models
- **Backend Processing:** Utilizes OpenCV and TensorFlow for real-time image pre-processing and inference

- **Visualization:** Provides highlighted tumor regions with bounding boxes and class probabilities

The application allows radiologists to quickly analyze brain MRI scans by integrating computer vision techniques into a user-friendly interface.

4.2 Deployment Links

- webpage: https://huggingface.co/spaces/Hiteshshan/brain_tumor_classification
- GitHub Repository: <https://github.com/team-project/brain-tumor-classification>

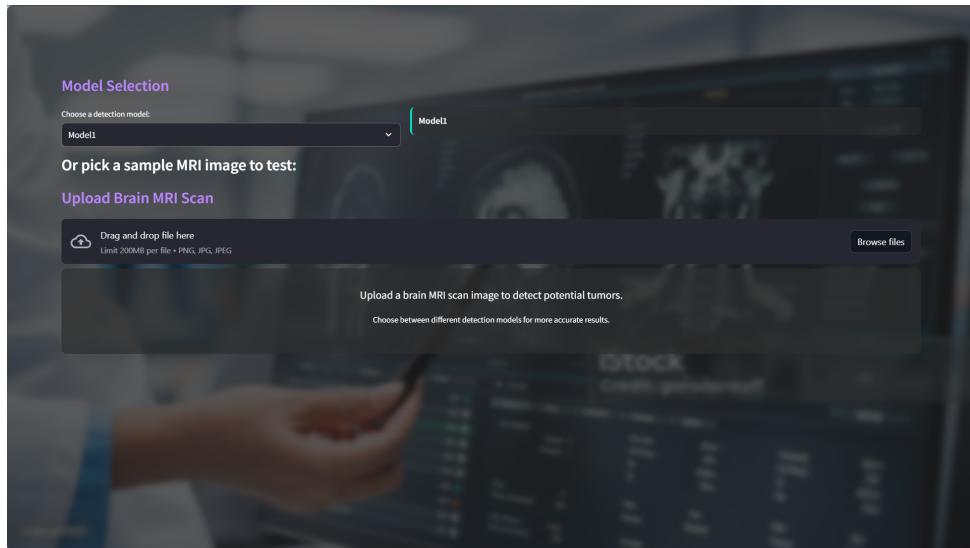


Figure 6: Web application interface showing the computer vision pipeline in action



Figure 7: example detection of tumor using model2

4.3 Usage Instructions and Computer Vision Pipeline

The web application implements a complete computer vision pipeline:

1. **Input:** Users upload MRI scan images in common formats (JPEG, PNG, DICOM)
2. **Preprocessing:** Images are automatically resized and normalized
3. **Model Selection:** Users can choose between classification and object detection approaches
4. **Inference:** The selected model processes the image to detect and classify tumors
5. **Visualization:** Results are displayed with visual overlays highlighting detected regions

5. Conclusion and Future Work

5.1 Conclusion

This project successfully demonstrated the effectiveness of various computer vision and machine learning techniques for brain tumor classification. Our key accomplishments include:

- Development and comparison of five distinct approaches to brain tumor classification
- Achieving high accuracy (90-92%) across multiple methods
- Creating a user-friendly web application for model deployment
- Demonstrating the value of image transformations and feature engineering in medical image analysis

We found that traditional computer vision techniques combined with machine learning can be as effective as more complex deep learning approaches while requiring fewer computational resources. The YOLO-based object detection model provided the best balance of accuracy and practical utility by both classifying and localizing tumors.

5.3 Future Work

Several directions for future research include:

- Integration of 3D convolutional networks to analyze complete MRI volumes
- Incorporation of additional imaging modalities (e.g., CT, PET)
- Development of ensemble methods combining predictions from multiple approaches
- Exploration of explainable AI techniques to provide insights into model decisions
- Integration with electronic health records for comprehensive clinical decision support
- Extension to tumor subtyping and grading for more detailed diagnosis

6. Team Contributions

- **Karan Reddy (B22AI023):** Implemented five approaches for brain tumor classification using varied computer vision and deep learning techniques. Applied image transformation methods like Canny edge detection, Harris corner detection, Hough circle detection, and K-Means clustering to enhance key features in MRI scans. These processed images were then fed into a lightweight CNN for classification. Also contributed in report writing.

- **Abhijan Theja (B22AI025):** Implemented handcrafted feature extraction techniques such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Scale-Invariant Feature Transform (SIFT) for brain tumor detection using traditional machine learning classifiers. Enhanced feature quality through image preprocessing techniques like the Prewitt edge detector and Butterworth Low Pass Filter to emphasize tumor-relevant patterns. Also contributed in report writing.
- **Chirag Kumar (B22AI056):** Implemented U-Net for pixel-level brain tumor segmentation, combining encoder-decoder architecture with skip connections for precise localization. Implemented deep learning and transfer learning techniques for tumor classification and built CNN, data augmentation techniques like rotation, shear, and zoom, to improve model generalization. Additionally, fine-tuned a VGG16 model pre-trained on ImageNet by adapting it for grayscale MRI inputs and adding custom dense layers for classification.
- **Hitesh shanmukha (B22AI013):** Implemented YOLOv11 for real-time detection and localization of brain tumors in MRI scans. The model predicts bounding boxes and class probabilities in a single pass, enabling efficient diagnosis. Deployed the trained model using Streamlit and integrated it with Hugging Face for interactive web-based inference and demonstration.
- **Ali zayad (B22AI055):** Implemented Discrete Cosine Transform (DCT) to efficiently extract frequency-domain features from brain scan images, enhancing the model's ability to capture both global and localized patterns for tumor classification, by converting spatial information into frequency representations. Also, participated in ideation of U-NET. Preprocessed MRI images using BM3D denoising, which preserved tumor boundaries and improved segmentation accuracy. Implemented CNN (without Augmentation) for multi label classification.

7. References

1. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention (pp. 234-241).
2. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
3. Jocher, G., Qiu, J., & Chaurasia, A. (2023). Ultralytics YOLO (Version 8.0.0).
4. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 886-893).
5. Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on image processing, 16(8), 2080-2095.
6. Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.
7. Harris, C. G., & Stephens, M. (1988). A combined corner and edge detector. In Alvey vision conference (pp. 10-5244).
8. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

9. Lloyd, S. (1982). Least squares quantization in PCM. IEEE transactions on information theory, 28(2), 129-137.
10. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.