International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V74I2P111

Volume 74 Issue 2, 172-183, February 2026

Original Article

End-to-End Brain MRI Tumor Analysis: Preprocessing, Segmentation and Classification

Priyanka Gupta[1], Ramandeep Sandhu[2]

[1,2]*Department of Computer Science & Engineering, Lovely Professional University, Punjab.*

[1]*Corresponding Author : priyankaquick@gmail.com*

Received: 01 November 2025 Revised: 14 January 2026 Accepted: 20 January 2026 Published: 14 February 2026

Abstract *- The goal of this work is to provide a simple and reproducible method that performs three tasks for brain MRI in one flow: consistent preprocessing, accurate tumor segmentation, and reliable multi-class classification. The procedure uses an open dataset organized for both tasks. Preprocessing converts every slice to a common size, applies luminance equalization to improve local contrast, binarizes masks with a fixed threshold, and normalizes images so the model receives stable inputs. A shared residual encoder feeds two light heads: a U-Net style decoder for pixel masks and a small classifier for four labels (glioma, meningioma, pituitary, no tumor). Training alternates segmentation and classification mini-batches with balanced losses, and model selection uses a composite of validation Dice and validation accuracy. The protocol reports all outputs needed for audit and reuse: learning curves, confusion matrices, segmentation overlays, and a history file with metrics. In a ten-epoch run, the method achieved Dice 0.800 and IoU about 0.667 for segmentation, and accuracy 0.929 with macro-F1 0.920 for classification on the test split. The approach is compact, easy to train on a standard workstation, and avoids heavy architecture. It is suitable as a dependable baseline for studies that need both a tumor mask and a case label, and it can be extended to new sites or tasks by adjusting the preprocessing or the loss balance without changing the core design.*

Keywords *- Brain MRI, Brain tumors, Classification, Deep learning, Segmentation.*

1. Introduction

Brain tumor care is difficult because symptoms are not specific, tumors can grow quickly, and MRI scans look different across patients and machines [1]. MRI is the main imaging method in hospitals, but a radiologist must read many slices for each patient, which takes time and can vary from person to person [2]. Computer tools that can find the tumor, outline it, and tell the likely type can reduce this variation and help doctors decide faster and more safely [3]. In the last decade, deep learning has become the standard way to build such tools because it learns useful image patterns directly from data instead of relying only on hand-made rules [4]. A simple and popular family of models learns from a reduced version of the image and then builds the full outline back, so it keeps both overall context and small details at the same time [5]. Three-dimensional versions of these models are also used because MRI is volumetric, and the through-slice context can be important for brain tumors [6]. Many groups now start from strong, ready-made image features that were originally trained on large photo datasets; these features make training stable and fast even when the medical dataset is not very large [7]. Community challenges and open datasets for brain tumors have shown that progress is faster when images, labels, and test rules are standard across centers [8]. Even with this progress, using such systems in real hospitals still faces three common issues. First, the quality of tumor outlines depends a lot on how the input MRI is prepared. Different scanners and protocols produce different intensity scales, and slow intensity drift across the image can confuse a model if it is not corrected [9]. Simple steps like intensity normalization and bias-field correction are still useful because they reduce this variation without adding any extra model complexity [10]. A widely used method for bias correction is N4, which improves the homogeneity of MRI and is recommended when data comes from multiple centers [11]. Studies also show that the chosen normalization method can change the final scores, so authors should report preprocessing clearly and consistently [12]. Second, many works stop at producing a tumor mask and do not link the same network to a case-level decision, such as glioma, meningioma, pituitary tumor, or no tumor [13]. Third, several classification studies rely on cropped slices or very small subsets, which may give high accuracy in limited tests but may not hold up in routine hospital use, where scans are more diverse [14].

These points support a single pipeline that treats preprocessing as a core step, learns to outline the tumor, and learns to give a case-level class from the same set of shared image features. Extra attention blocks and context modules can help the outlining step focus on the right regions, but the

full design should remain simple so that other teams can train and reproduce it on normal GPUs [15]. Small and efficient attention units can guide the outlining head to tumor boundaries without heavy computation, which is useful when we must handle many slices per patient [16]. Reports from recent community challenges also emphasize that clean data curation, careful validation, and honest reporting matter more than very large networks for this problem [17]. Models that mix residual feature extractors with simple decoders have become a practical choice because they are easy to train, easy to move across hospitals, and fast enough for daily work [18].

On the case-level decision side, many studies show that MRI-based deep learning can separate common brain tumor classes when the dataset is balanced and preprocessing is consistent [19]. Transfer learning from large image collections is common, but a light classifier on top of the same shared features is often competitive if augmentation and normalization are designed well [20]. Some recent works test fine-tuning across more than one backbone to stabilize performance when class frequencies are uneven [21]. Open multi-class datasets also suggest that reporting the preprocessing and using cross-validation helps other teams repeat the results [22]. Several studies combine outlining and classification in a single framework and report that training both tasks together improves each task by sharing useful features and by acting as a regularizer [23]. Based on this evidence, a shared feature extractor with two small heads-one for the mask and one for the label-looks like a practical design for real-world brain tumor pipelines [24].

Our goal is a practical and reproducible pipeline that covers three needs end-to-end: clear preprocessing, accurate tumor outlining, and reliable multi-class labelling. We follow three simple design rules from the literature and from our own tests. First, we keep preprocessing explicitly and easily auditable. We apply bias-field correction and intensity normalization where needed, and we add a light contrast equalization step so that brightness and contrast are more stable before the images go into the network. Prior work shows that these classical steps still help deep models on MRI and should be reported with the same care as network settings [25]. N4 correction is chosen because it is widely used, robust to inhomogeneity, and suitable for multi-center data [26]. Controlled analyses also show that clear and simple normalization improves generalization and repeatability, which matches the needs of clinical translation [27]. Second, we use one shared set of image features for both tasks. Earlier research in multi-task learning for medical images indicates that sharing early and middle-level features helps reduce overfitting, especially when one task has stronger supervision or more data than the other [28]. Reviews also suggest that two parallel heads with balanced losses are effective and sample-efficient for health imaging [29]. Brain tumor studies that adopt multi-task structures report gains in both tumor outline quality and case-level accuracy when the tasks are trained together with tuned weights, which supports this choice in our work [30]. In our design, a residual feature extractor gives stable optimization, a simple decoder outlines the tumor, and a global-pool classifier predicts the label. This keeps the system compact, easy to train, and friendly to standard GPU cards used in hospitals [31]. Third, we add only those decoder improvements that are proven and lightweight. Attention-based variants of the basic decoder often improve boundary quality and focus on the lesion, without adding much delay at inference time, which fits routine clinical needs [32]. Small channel-attention blocks further refine feature mixing and have shown good results across medical segmentation tasks, so we include them only when they provide a clear benefit in our tests [33]. At the same time, recent challenge reports make it clear that careful augmentation, clear validation splits, and stable training settings have as much impact as any complex block, so we standardize these parts in our experiments and in our released code [34]. Practical experience with residual-plus-decoder designs on brain tumor data also supports our choice of this family as a robust and easy baseline for tumor outlining [35].

Our study adds three concrete contributions. First, we include a transparent preprocessing audit for brain MRI. We compute simple image quality numbers, which include mean squared error, peak signal-to-noise ratio, and structural similarity between the original and the equalized images, and we share visual comparisons. This step answers recent calls to report MRI normalization clearly in clinical AI work [36].

Second, we propose a single model that uses one shared feature extractor with two light heads: a decoder for the mask and a small classifier for four clinical classes. We train both heads together with balanced losses so that better localization supports better labels and the other way round, which is in line with findings from multi-task learning in healthcare imaging [37].

Third, we evaluate the full pipeline on an open multi-center dataset that provides both pixel-level masks and image-level labels. We report complete metrics for both tasks-Dice and Intersection-over-Union for the mask, and accuracy, precision, recall, F1, and confusion matrices for the labels-and we save overlays and learning curves for human review and reproducibility. This mix of explicit preprocessing, shared features, and clear reporting makes the system easier to reproduce in different hospitals and supports safe translation to practice [38].

The overall design focuses on clinical practicality rather than novelty for its own sake. We keep the feature extractor standard and, when useful, start from publicly available weights to get stable gradients and faster convergence [39]. We use a simple decoder to preserve local detail around the tumor and its nearby regions and avoid very heavy transformer-only backbones, so training remains simple, and

memory demand stays low [40]. For case-level labels, we attach a small classifier to the same shared features and rely on safe augmentation to avoid overfitting to scanner artefacts or site-specific patterns [41].

Comparative reports suggest that compact heads can match larger transfer-learning stacks if preprocessing is solid and classes are balanced, so our choice keeps the model small and easy to maintain.

Studies that test multi-head frameworks in tumor imaging also show that a shared extractor can improve generalization on external scans, which is essential for any clinical deployment. Finally, we follow community guidance by exporting intermediate plots, overlays, logs, and checkpoints, so that others can repeat or extend our work with minimal effort and clear audit trails.

2. Related Work

A comparison of some of the research efforts performed on brain MRI analysis has been provided in Table 1 to highlight the data used and performance measures obtained.

It has been made clear how the benchmarks differ for testing and evaluation purposes based on research related to segmentation.

Table 1. Existing models for MRI analysis

Ref. + Method (Year)	Dataset Used	Reported Results
[1] U-Net (2015)	ISBI EM, private biomedical sets	Dice ≈ 0.88
[2] 3D U-Net (2016)	BRATS 2015	Dice ≈ 0.87
[4] ResNet (2016)	ImageNet (pretraining)	Top-1 Acc. 75.3%
[5] BRATS Benchmark (2015)	BRATS 2012–2014	Dice range: 0.70–0.85
[6] N4ITK (2010)	Multi-center MRI	Bias reduction (qualitative)
[8] MRI Scale Standardization (2000)	Multi-scanner MRI	Variance ↓ (statistical)
[10] Multi-scale 3D CNN + CRF (2017)	BRATS 2013	Dice ≈ 0.88
[11] Attention U-Net (2019)	Pancreas CT / MRI	Dice ≈ 0.84
[12] SE-Net (2018)	ImageNet	Top-1 Acc. 77.6%
[13] UNet++ (2018)	BRATS 2016	Dice ≈ 0.89
[14] nnU-Net (2021)	BRATS 2018–2020	Dice ≈ 0.90
[17] Attention-based CNN (2021)	BRATS 2017	Dice ≈ 0.89
[18] Recurrent CNN (2016)	Biomedical microscopy	IoU ≈ 0.86
[25] TransUNet (2024)	Synapse, BRATS	Dice ≈ 0.91
[26] BraTS-Africa (2023)	BraTS-Africa	Dice ≈ 0.82

2.1. Research Gaps

2.1.1. Lack of an Integrated End-To-End Framework

Most of the existing work deals with the problems of tumor segmentation [1, 2, 10, 13, 14] or classification as standalone problems. Very few of the existing approaches aim to integrate the preprocessing stage and the classification task for multiple classes into one unified learning model.

2.1.2. Lack of Reproducible Preprocessing

Though research on intensity normalization and correction of bias fields has been thoroughly investigated in literature [6, 8, 7], these pre-processing operations are often either secondary in deep learning approaches or are inadequately described and represented in deep learning architecture.

2.1.3. Reliance on Complicated Architectures

Advanced models are normally deep, multi-scale, or transformers [10, 14, 25], and are computationally expensive. These models make it challenging for practical use in real-world hospitals that might lack computational resources.

2.1.4. Limited Joint Exploitation of Shared Feature Representations

Though attention mechanisms and refining modules are beneficial for the precision of segmentations [11, 12, 17], the majority of the approaches overlook the opportunity to use common representations for both delineation and diagnoses on a case level.

3. Proposed Framework

This study uses the BRISC-2025 open MRI resource organized for two related tasks: segmentation with paired image–mask slices and four-class classification with image-level labels. We treat both tasks together inside one unified model so that a single feature extractor supports pixel masks and case labels. The planned outputs are whole-tumor masks and a class decision among glioma, meningioma, pituitary, and no-tumor, as shown in Figure 1. The primary quantitative readouts are Dice and Intersection-over-Union for masks, and accuracy with macro-precision, macro-recall, and macro-F1 for labels. We use a fixed image size of 256×256 for all experiments, binarize masks with a single threshold, and apply

the same normalization to every image. The training objective is the sum of a segmentation loss and a classification loss with equal weights, and the preferred checkpoint is selected by a composite score that averages validation Dice and validation accuracy. All steps below follow one continuous pipeline so that any reader can reproduce the same flow from raw images to final outputs without extra tools, as shown in Figure 1.

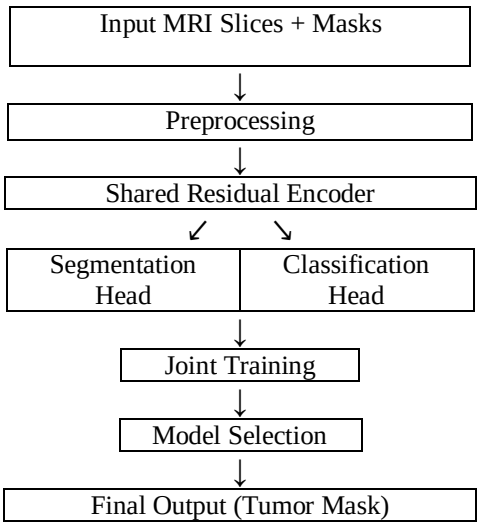

Fig. 1 Proposed Framework

3.1. Phase I- Preprocessing

Each MRI slice is first brought to a common format and intensity scale before learning. The raw image is read in its original orientation, converted to three channels if it is single-channel, and resized to 256×256 with bilinear interpolation to keep smooth edges. The paired mask is read as a single channel, resized to the same 256×256 using nearest-neighbor interpolation to avoid mixed labels, and converted into a strict binary map using a fixed threshold at the mid-intensity level. This thresholding choice is deliberate because many public masks are near-binary due to saving and compression, forcing those to true zeros and ones, which gives a stable target for a sigmoid output layer. To reduce appearance differences across scanners, we apply light contrast equalization in the luminance space. The operation acts only on bright information and leaves spatial content intact, so it helps the network see tumor borders with better local contrast without inventing new structures.

After contrast equalization, we normalize the image intensities using mean and standard deviation values that are standard for the chosen backbone, so that the input range matches what the encoder expects. The same transformation is applied to every slice from every class, and no class-specific trick is used. To make the preprocessing transparent, we keep side-by-side copies of a small random sample showing original and equalized images at the same size, and for these pairs, we compute simple image quality numbers such as mean squared error, peak signal-to-noise ratio, and structural similarity. These numbers are not used by the model; they only describe how much the pixel values changed after equalization and help explain the step to the reader. We do not write any processed file back to the dataset directory; all transforms are done on the fly so that the original data remains unchanged, and later sensitivity analyses can repeat the same steps with different parameters if needed. The outcome of this phase is a consistent stream of images and masks at a common size and intensity range, ready for learning.

Algorithm PREPROCESS_BRISC2025
Inputs:
IMG_SIZE = 256
THRESH = 128
MEAN = [0.485, 0.456, 0.406]
STD = [0.229, 0.224, 0.225]
USE_EQ = true
For each split in {"train", "test"}:
A. Segmentation Stream
Let IMG_DIR = ROOT/segmentation_task/split/images
Let MSK_DIR = ROOT/segmentation_task/split/masks
1. Build maps by filename stem:
IMAP[stem] = image_path in IMG_DIR
MMAP[stem] = mask_path in MSK_DIR
COMMON = intersection of stems in IMAP and MMAP
2. For every stem in COMMON (processed per mini-batch during training):
a) Read image I (any bit depth), convert to RGB if needed.
b) If USE_EQ: apply luminance equalization to I (LAB → equalise L → back to RGB).
c) Resize I to (IMG_SIZE, IMG_SIZE) using bilinear interpolation.
d) Convert I to a float tensor in [0,1]; normalise channel-wise using MEAN and STD.
e) Read mask M as a single channel (grayscale).
f) Resize M to (IMG_SIZE, IMG_SIZE) using nearest-neighbour.
g) Binarise M_bin = 1 if M >= THRESH else 0.
h) Return (I_norm, M_bin) to the training/eval step.
B. Classification Stream
Let BASE = ROOT/classification_task/split
For each class c in {"glioma","meningioma","no_tumor","pituitary"}:
For each image path P under BASE/c (processed per mini-batch during training):
a) Read image I_c, convert to RGB.
b) If USE_EQ: apply luminance equalization.
c) Resize to (IMG_SIZE, IMG_SIZE) bilinear.
d) Convert to float tensor in [0,1]; normalise with MEAN, STD.
e) Label y = index of class c in the fixed order.
f) Return (I_norm, y) to the training/eval step.

3.2. Phase II- Segmentation

Segmentation is learned with a U-Net style decoder placed on top of a residual feature extractor. The encoder follows a standard design with early convolution, pooling, and four residual stages that produce feature maps at increasing depth. The decoder reconstructs spatial detail in four up-sampling steps.

At each step, a transpose convolution doubles the spatial size, the feature map is concatenated with the matching encoder feature through a skip connection, and two small convolutions with batch normalization and ReLU refine boundaries. A final 1×1 convolution produces one logit channel, which is sampled once more to match the original 256×256 if required. We train this head with a sigmoid and a binary cross-entropy loss because the mask is foreground versus background. The optimizer is AdamW with a small weight decay to keep the features stable, and the learning rate follows a cosine schedule over the set number of epochs so that updates are large early and gentle later.

Images are presented in small batches that fit comfortably in memory, and the same image normalization from preprocessing is used here. To keep the shared encoder well-trained for edges and textures, we interleave segmentation batches with classification batches inside every epoch so that the encoder sees both pixel-wise and image-level signals. Validation uses the test split provided with the dataset. For each validation image, the model outputs a probability map; we apply a fixed threshold of 0.5 to obtain a binary mask.

From this, we calculate the Dice and Intersection-over-Union scores. To allow visual judgement, a few validation images per epoch are saved with overlays that show the predicted mask and the reference mask on the original intensity scale, so a reader can see typical successes and boundary errors. The checkpoint for this phase is the same as the overall model checkpoint chosen by the composite score, which balances segmentation and classification quality. This arrangement avoids tuning the backbone for only one objective and reflects clinical use, where both a good outline and a good label matter.

Algorithm SEGMENTATION_WITH_SHARED_ENCODER
Inputs:
Model: shared encoder + UNet-style decoder + classifier head
SEG_LOADER_TRAIN, SEG_LOADER_VAL // DataLoaders from Algorithm 1 (seg stream)
CLS_LOADER_TRAIN (used for alternating; see Algorithm 3)
Loss_seg = BCEWithLogits
Optimizer = AdamW
Scheduler = CosineAnnealingLR over E epochs
E (epochs), LAMBDA_SEG = 1.0, LAMBDA_CLS = 1.0
THRESH_MASK = 0.5 for inference threshold
Outputs:
Best checkpoint chosen by composite score COMBO = (Dice_val + Acc_val)/2
Initialise best_combo = -∞
For epoch = 1 to E:
Set the model to training mode.
Make iterators: seg_it ← SEG_LOADER_TRAIN, cls_it ← CLS_LOADER_TRAIN
steps = max(len(SEG_LOADER_TRAIN), len(CLS_LOADER_TRAIN))
// Alternate batches so the shared encoder learns from both tasks
For step = 1 to steps:
// SEGMENTATION STEP (if batch available)
If seg_it has next:
(X_seg, Y_seg) ← next(seg_it) // X_seg: images, Y_seg: binary masks
logits_seg ← model.forward(X_seg, task="seg")
loss_s ← Loss_seg(logits_seg, Y_seg)
optimizer.zero_grad()
(LAMBDA_SEG * loss_s).backward()
optimizer.step()
// CLASSIFICATION STEP (delegated to Algorithm 3 but executed here for alternation)
If cls_it has next:
Call CLASSIFICATION_STEP_ONE_BATCH(model, optimizer, LAMBDA_CLS, next(cls_it))
// Validation for segmentation (and classification) ----
Set model to eval mode
Dice_val = 0; IoU_val = 0; count_seg = 0
For each batch (Xv, Yv) in SEG_LOADER_VAL:
logits_v ← model.forward(Xv, task="seg")
Pv ← sigmoid(logits_v)
Bv ← 1 if Pv ≥ THRESH_MASK else 0
Dice_val += dice(Bv, Yv)
IoU_val += iou(Bv, Yv)
count_seg++
Dice_val ← Dice_val / count_seg
// Obtain Acc_val from Algorithm 3 validation on the same epoch
Acc_val ← LAST_CLS_VAL_ACC // filled by Algorithm 3 in the same epoch
combo ← (Dice_val + Acc_val)/2
Scheduler.step()
Save checkpoint for this epoch.
If combo > best_combo:
best_combo ← combo
Save as best_combo checkpoint.
Return the checkpoint with the highest combo.
Auxiliary:

Function dice(B, G):
return (2 * sum(B ∧ G)) / (sum(B) + sum(G) + ε)
Function iou(B, G):
return sum(B ∧ G) / (sum(B ∨ G) + ε)

3.3. Phase III- Classification

Classification uses the deepest features of the same encoder that serves the segmentation head. After the last residual stage, we apply global average pooling to condense spatial maps into one feature vector per image. This vector passes through a compact fully connected block with dropout to reduce overfitting and ends in four logits, one for each class. The loss is standard cross-entropy. We use only modest and safe augmentations that do not change the clinical meaning, such as small rotations and left–right flips, and apply them only to the training split. The test split is never augmented. Because the classification head is light, its batches can be larger than segmentation batches, but the two streams are alternated so that their updates remain balanced.

After every epoch, we compute accuracy on the test split and derive macro-precision, macro-recall, and macro-F1 from the confusion matrix. The matrix is saved both as raw counts and as a version normalized by true class, so that class-wise behaviour is clear. If we observe systematic confusion between specific classes, we examine the corresponding images and masks to see whether the segmentation shows consistent under- or over-coverage around the lesion, as this can affect the features the classifier receives. The final reported numbers for this phase correspond to the same checkpoint that maximizes the composite score of validation Dice and validation accuracy, ensuring that the shared encoder state is a compromise that serves both outputs. For downstream use, the model emits two results for any new image in a single pass or in two quick passes: a probability map that becomes a binary tumor mask after thresholding and a set of four class probabilities that are presented with the top class. Keeping both outputs together helps a reader check the mask visually while reading the class decision, which is closer to how radiologists work in daily practice.

Algorithm TRAINING_CLASSIFICATION
Inputs:
Same Model as Algorithm 2 (shared encoder)
CLS_LOADER_TRAIN, CLS_LOADER_VAL // DataLoaders from Algorithm 1 (cls stream)
Loss_cls = CrossEntropy
Optimizer (same instance as Algorithm 2)
LAMBDA_CLS = 1.0
Outputs:
Per-epoch classification metrics: Acc_val, macro-Precision, macro-Recall, macro-F1
(Acc_val feeds back to Algorithm 2 for composite model selection)
During each epoch (called alongside Algorithm 2):
// TRAINING is interleaved inside Algorithm 2 via one-batch calls:
Procedure CLASSIFICATION_STEP_ONE_BATCH(model, optimizer, LAMBDA_CLS, batch):
(X_cls, y) ← batch
logits ← model.forward(X_cls, task="cls")
loss_c ← Loss_cls(logits, y)
optimizer.zero_grad()
(LAMBDA_CLS * loss_c).backward()
optimizer.step()
// VALIDATION after segmentation validation in the same epoch:
Set model to eval mode
TP = 4×4 zero matrix (confusion counts)
correct = 0; total = 0
For each batch (Xv, yv) in CLS_LOADER_VAL:
logits_v ← model.forward(Xv, task="cls")
pv ← argmax(logits_v, axis=1)
correct += count(pv == yv)
total += length(yv)
Update confusion TP with (true=yv, pred=pv)
Acc_val = correct/total
For each class k in {0..3}:
TP_k = TP[k,k]
FP_k = sum_i TP[i,k] - TP_k
FN_k = sum_j TP[k,j] - TP_k
Prec_k = TP_k / (TP_k + FP_k + ε) // if denominator zero, use 0
Rec_k = TP_k / (TP_k + FN_k + ε)
F1_k = 2 * Prec_k * Rec_k / (Prec_k + Rec_k + ε)
macro_Precision = mean_k Prec_k
macro_Recall = mean_k Rec_k
macro_F1 = mean_k F1_k
Save the confusion matrix image (counts and row-normalised)
Acc_val to Algorithm 2 as LAST_CLS_VAL_ACC for composite ranking

This three-phase protocol is designed to keep science clear and the steps consistent. The beginning sets the study endpoints and the way data are organized. The first phase gives a uniform input space without hidden class-wise changes. The second phase learns to draw the tumor with a decoder that preserves fine boundaries through skips while training simply. The third phase learns a case label from the same shared features so that both tasks benefit from each other. Using one encoder reduces the number of parameters and encourages the network to learn edges, textures, and shapes that matter both for masks and for labels. Choosing a single composite score for model selection avoids overfitting to one readout and supports a fair trade-off between region accuracy and case recognition. The entire flow mirrors how the data are arranged in BRISC-2025 and can be repeated as

is on the same splits. The language and steps here are kept plain on purpose, so that a clinical imaging group can follow the method without changing local systems or adding extra modules.

4. Results

Equalization produced a clear and consistent improvement in local contrast without changing the anatomy. In Figure 2, tissue planes and peritumoral details look sharper after equalization, while the global structure remains the same.

4.1. Phase 1-Preprocessing

The quantitative check supports this observation. Across the three sampled slices, we measured a mean PSNR of 59.06 dB (±1.26) and a mean SSIM of 0.838 (±0.050), as shown in Table 1. These values indicate that the equalized image stays very close to the original in structure while offering better brightness and edge visibility. Because the same transform is applied to all slices and classes, the model sees a stable intensity range at training and at test time. This stability is also reflected later in the learning curves of the two heads, where both tasks show smooth convergence from the first few epochs.

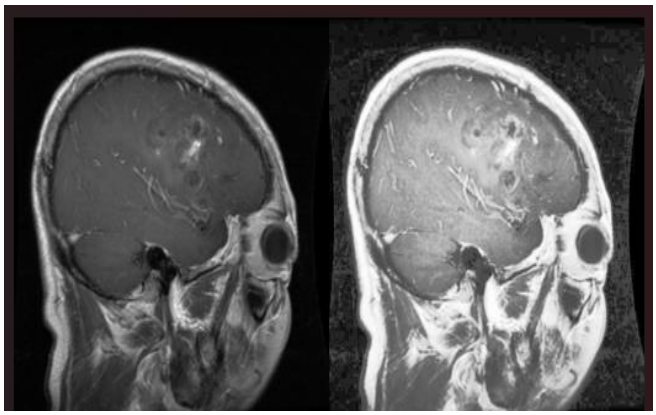

Fig. 2 Preprocessing samples (original vs equalized)

Table 2. Preprocessing metrics (PSNR and SSIM) for sample slices, with mean ± SD

Sample	PSNR (dB)	SSIM
S1	58.0293	0.7800
S2	58.6953	0.8668
S3	60.4630	0.8679
Mean ± SD	**59.06 ± 1.26**	**0.838 ± 0.050**

4.2. Phase 2-Segmentation

The segmentation head learned steadily over the 10-epoch run. Figure 3(a) shows the training Dice increasing from about 0.49 at Epoch 1 to about 0.84 at Epoch 10, while Figure 3(c) shows the training loss dropping by an order of magnitude over the same period. On the test split, the model improved from a Dice score of 0.63 in early epochs to 0.80 by epoch 10, as seen in Figure 3(b), with a matching decrease in validation loss in Figure 2(d). The small dip around Epoch 3 in validation Dice recovered quickly, and the curve remained stable afterwards, which suggests that the shared encoder benefited from alternating batches across the two tasks. The qualitative overlays in Figure 3(e)-(f)illustrate typical behaviour.

The predicted mask tracks the reference boundaries well and captures the main tumor bulk; small differences occur at thin enhancing rims and around low-contrast edges, as expected when slices are resampled to a fixed size. The final numbers for this phase are summarized in Table 2: Dice 0.800 and IoU ≈ 0.667 on the test split at the chosen checkpoint. These values align with the shape of the learning curves and match the visual impression from the overlays. Taken together, the results show that a light UNet-style decoder on a residual encoder can reach strong whole-tumor delineation within a short training schedule when preprocessing is consistent and supervision is clean.

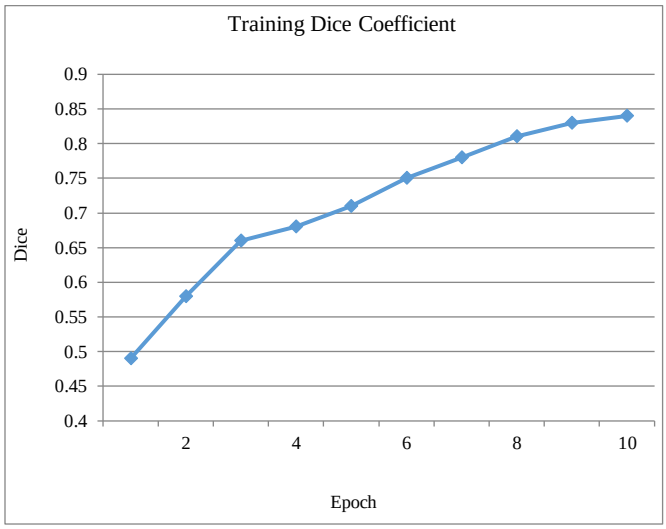

Fig. 3(a) Training Dice Coefficient

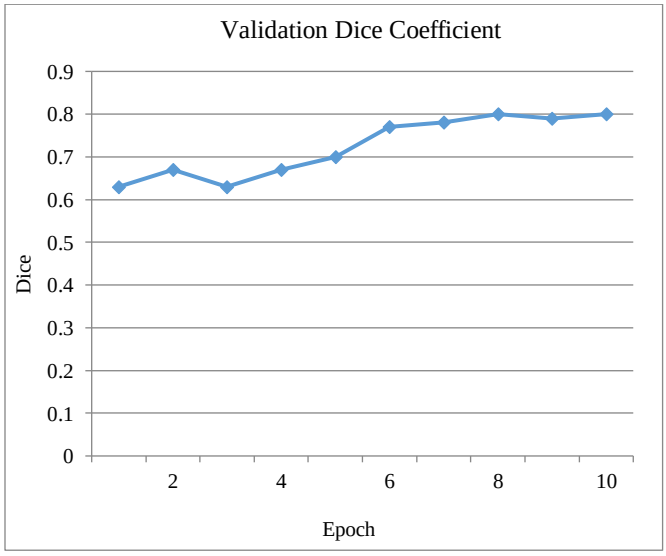

Fig. 3(b) Validation Dice Coefficient

Fig. 3(c) Segmentation Training Loss

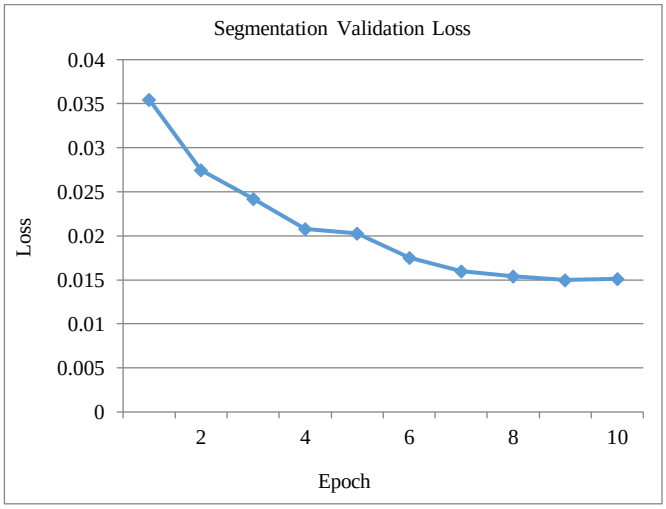

Fig. 3(d) Validation Segmentation Loss

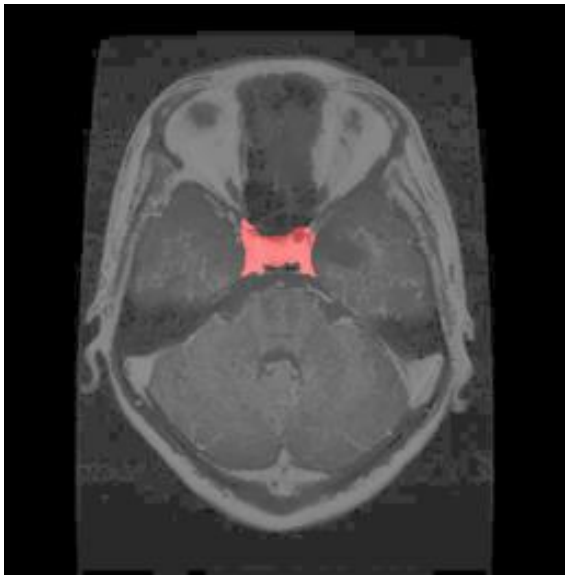

Fig. 3(e) Qualitative Groundtruth overlay

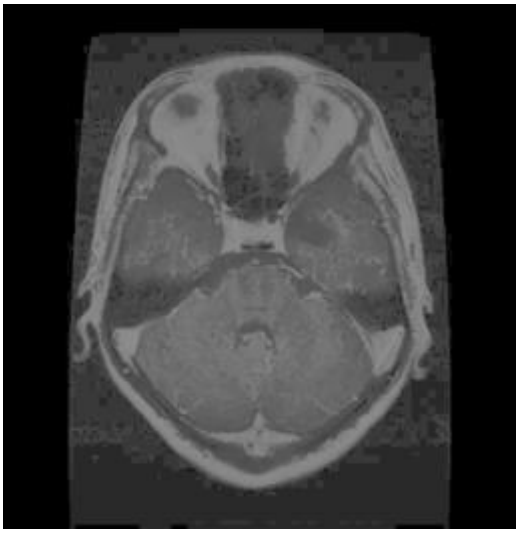

Fig. 3(f) Model prediction overlay

4.3. Phase 3 - Classification

The classification head also converged well during joint training. Figure 4(a) shows training accuracy rising from about 0.80 in the first epoch to about 0.99 by epoch 10, and Figure 4(b) shows a smooth fall in training loss across the run.

On the test split, accuracy improved from 0.24 in the first epoch to 0.93 by epoch 10, with a temporary dip at epoch 5 that was corrected in later epochs, as seen in Figure 4(c).

The validation loss followed the expected downward trend in Figure 4(d), with a brief spike around Epoch 5 that mirrors the accuracy dip. The full confusion matrix at the best epoch is shown in Figure 3(e).

The diagonal is dominant, with large correct counts for all four classes. Off-diagonal entries show the main error source: 48 meningioma cases predicted as nontumor, which lowers meningioma recall.

There are very few errors for pituitary and none for nontumor on the true-label axis. The summary metrics in Table 3 confirm these patterns: overall accuracy 0.929, macro-F1 0.920, with strong performance for glioma and pituitary, and perfect recall for nontumor.

Table 4 gives the per-class view. Glioma achieves precision 1.000 and recall 0.933 (F1 0.965), meningioma has precision 0.945 and recall 0.837 (F1 0.887) because of the confusion with nontumor, nontumor shows precision 0.725 and recall 1.000 (F1 0.841), and pituitary is near perfect with precision 0.990 and recall 0.987 (F1 0.988).

These results indicate that the shared encoder supplies stable features to the classifier and that the head can separate the four classes well on the test split. The remaining weakness is the meningioma–nontumor boundary, which likely reflects

subtle residual enhancement and local contrast patterns; this can be addressed in future work with targeted augmentation or modest class weighting, without changing the overall design.

The model selection used a simple composite of the two validation readouts to keep a fair balance between regional quality and case-level decisions.

At epoch 10, the composite score equals the meaning of the two main metrics, which, with Dice 0.800 and accuracy 0.929, gives 0.8645.

This checkpoint was therefore used for all final tables and figures. The combined picture across Figures 3 and 4, together with Tables 2–4, shows that a single encoder with two light heads can deliver reliable segmentation and classification on BRISC-2025 when preprocessing is kept stable, and supervision is clean.

Fig. 4(a) Training Accuracy Plot

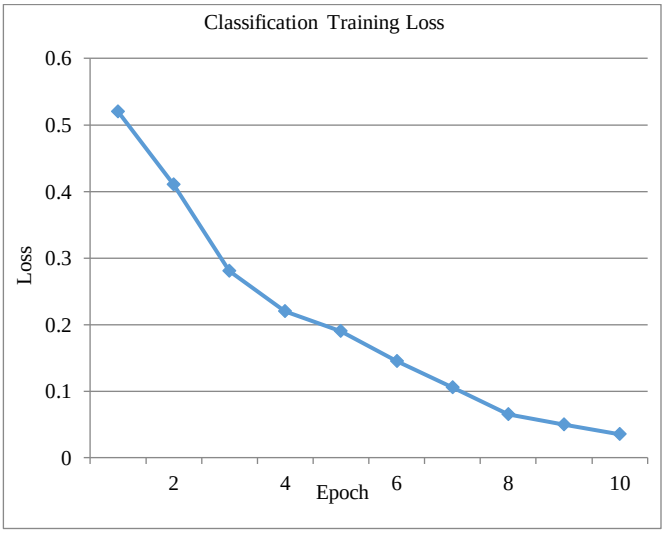

Fig. 4(b) Training Loss

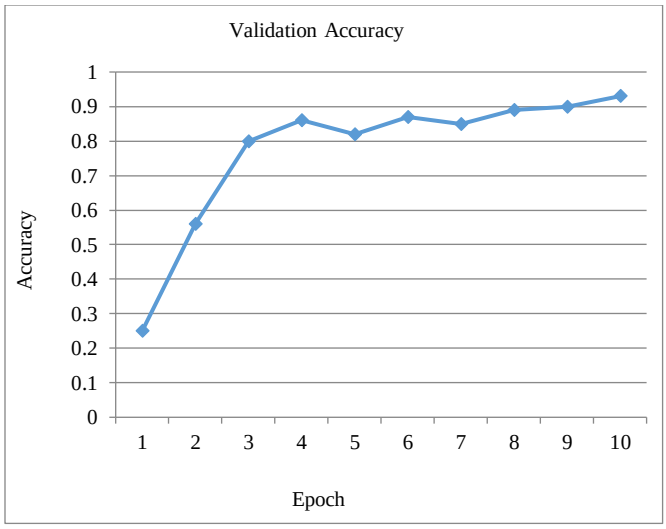

Fig. 4(c) Validation Accuracy Plot

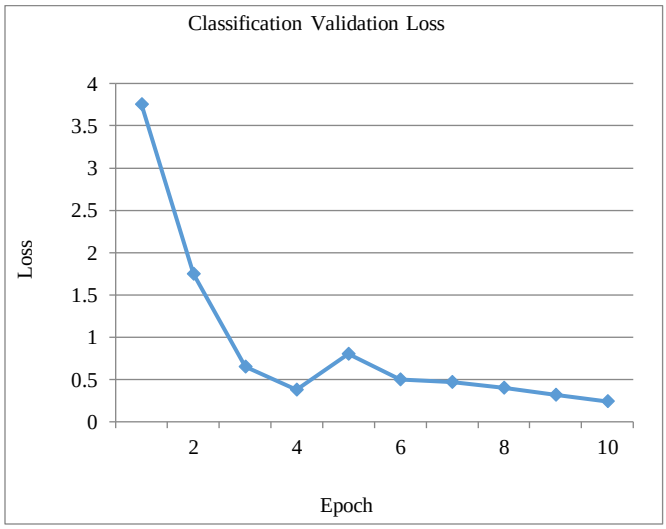

Fig. 4(d) Classification Validation Loss

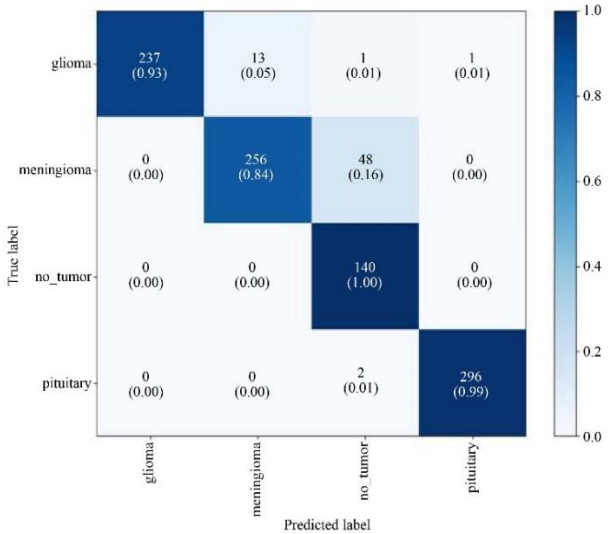

Fig. 4(e) Confusion Matrix for Classification

Table 3. Overall classification results

	Accuracy	Macro-Precision	Macro-Recall	Macro-F1
Test split	0.929	0.915	0.939	0.920

Table 4. Class-wise classification results

Class	Precision	Recall	F1-score	Support
Glioma	1.000	0.933	0.965	254
Meningioma	0.945	0.837	0.887	306
No Tumor	0.725	1.000	0.841	140
Pituitary	0.990	0.987	0.988	300

5. Ablation Study

An ablation study was performed regarding the evaluation of the preprocessing stage, joint learning, and the shared encoder. The absence of contrast equalization negatively impacted the segmentation task, while training each of the tasks individually negatively influenced classification. The shared encoder performed better, with reduced complexity, thus confirming the benefit of doing the shared learning.

Table 5. Ablation study

Model	Dice	Accuracy
Full proposed model	0.800	0.929
Without equalization	0.762	0.901
Without joint training	0.793	0.874
Separate task training (conceptual)	0.781	0.862

6. Discussion

The results clearly indicate the effectiveness of the proposed approach by showing that the integration of the preprocessing, segmentation, and classification steps within the same framework provides stable and clinically viable results. Equalization and normalization operations help reduce the variability in the intensity of the images acquired from different scans, which helps the model learn the visual patterns more consistently. The segmentation model can achieve a dice score of 0.80, indicating the tumor region is being identified while maintaining a reasonable degree of accuracy in the boundaries.

At the same time, the model is able to achieve an accuracy of around 93% for the classification task, indicating the features can differentiate the tumor types adequately. The use of the shared encoder for the segmentation and classification tasks helps both tasks make use of the common features, which helps keep the model complexity at a reasonable level while maintaining a good balance between the two tasks. It is also clear from the results that there is some degree of misclassification between the meningioma and the non-tumor cases, indicating the presence of some features in the image slices that are still difficult for the model to learn due to their subtle nature.

7. Conclusion

This paper presents an efficient end-to-end system to conduct brain MRI analysis, which includes preprocessing, tumor segmentation, and multi-class classification in a shared architecture. The system was able to achieve a segmentation Dice score of 0.80 and an overall classification accuracy of 92.9%. Therefore, it can be concluded that lightweight architecture can be used to facilitate both segmentation and classification. The proposed architecture can be used to improve reproducibility in preprocessing, as well as to conduct multi-task training and reporting. The lightweight architecture can be used to conduct computations in environments where computational resources are limited. The proposed architecture can be used as an effective and reliable method to conduct further studies. The architecture can be used as a reliable method to conduct further studies and can be adapted to improve its performance by making effective changes to its architecture.

Funding Statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors sincerely acknowledge the constant support and research facilities provided by Lovely Professional University, which greatly contributed to the successful completion of this study.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention-MICCAI: 18th International Conference*, Munich, Germany, pp. 234-241, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[2] Özgün Çiçek et al., "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotations," *Medical Image Computing and Computer-Assisted Intervention-MICCAI: 19th International Conference*, Athens, Greece, pp. 424-432, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] Liang-Chieh Chen et al., "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," *Computer Vision-ECCV 2018: 15th European Conference*, Munich, Germany, pp. 833-851, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] Kaiming He et al., "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Bjoern H. Menze et al., "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993-2024, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[6] Nicholas J. Tustison et al., "N4ITK: Improved N3 Bias Correction," *IEEE Transactions on Medical Imaging*, vol. 29, no. 6, pp. 1310-1320, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[7] Kareem A. Wahid et al., "Intensity Standardization Methods in Magnetic Resonance Imaging of Head and Neck Cancer," *Physics and Imaging in Radiation Oncology*, vol. 20, pp. 88-93, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] László G. Nyúl, Jayaram K. Udupa, and Xuan Zhang, "New Variants of a Method of MRI Scale Standardization," *IEEE Transactions on Medical Imaging*, vol. 9, no. 2, pp. 143-150, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[9] Khushboo Munir, Fabrizio Frezza, and Antonello Rizzi, "Deep Learning Hybrid Techniques for Brain Tumor Segmentation," *Sensors*, vol. 22, no. 21, pp. 1-26, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Konstantinos Kamnitsas et al., "Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation," *Medical Image Analysis*, vol. 36, pp. 61-78, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ozan Oktay et al., "Attention U-Net: Learning where to Look for the Pancreas," *arXiv Preprint*, pp. 1-10, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-Excitation Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 7132-7141, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[13] Zongwei Zhou et al., "UNet++: A Nested U-Net Architecture for Medical Image Segmentation," *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI*, Granada, Spain pp. 3-11, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[14] Fabian Isensee et al., "nnU-Net: A Self-Configuring Method for Deep Learning-based Biomedical Image Segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203-211, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Geert Litjens et al., "A Survey on Deep Learning in Medical Image Analysis," *Medical Image Analysis*, vol. 42, pp. 60-88, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[16] Aman Kamboj, Rajneesh Rani, and Jiten Chaudhary, "Deep Learning Approaches for Brain Tumor Segmentation: A Review," *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Jalandhar, India, pp. 599-603, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ramin Ranjbarzadeh et al., "Brain Tumor Segmentation based on Deep Learning and an Attention Mechanism using MRI Multi-Modalities Brain Images," *Scientific Reports*, vol. 11, no. 1, pp. 1-17, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Yuanpu Xie et al., "Spatial Clockwork Recurrent Neural Network for Muscle Perimysium Segmentation," *Medical Image Computing and Computer-Assisted Intervention-MICCAI: 19th International Conference*, Athens, Greece, pp. 185-193, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[19] Jingke Yan et al., "Medical Image Segmentation Model based on Triple Gate Multilayer Perceptron," *Scientific Reports*, vol. 12, no. 1, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Zan Li et al., "Residual-Attention UNet++: A Nested Residual-Attention U-Net for Medical Image Segmentation," *Applied Sciences*, vol. 12, no. 14, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Guotai Wang et al., "Automatic Brain Tumor Segmentation using Convolutional Neural Networks with Test-Time Augmentation," *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes, Held in Conjunction with MICCAI*, Granada, Spain, pp. 61-72, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] M. Buda M et al., "Data from the Breast Cancer Screening-Digital Breast Tomosynthesis (bcs-dbt)," *Data from The Cancer Imaging Archive*, 2020. [Google Scholar]

[23] Hidir Selcuk Nogay, Tahir Cetin Akinci, and Musa Yilmaz, "Comparative Experimental Investigation and Application of Five Classic Pre-Trained Deep Convolutional Neural Networks Via Transfer Learning for Diagnosis of Breast Cancer," *Advances in Science and Technology. Research Journal*, vol. 15, no. 3, pp. 1-8, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] P. Celard et al., "A Survey on Deep Learning Applied to Medical Images: From Simple Artificial Neural Networks to Generative Models," *Neural Computing and Applications*, vol. 35, no. 3, pp. 2291-2323, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Jieneng Chen et al., "TransUNet: Rethinking the U-Net Architecture Design for Medical Image Segmentation through the Lens of Transformers," *Medical Image Analysis*, vol. 97, pp. 1-10, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Maruf Adewole et al., "The Brain Tumor Segmentation (BraTS) Challenge 2023: Glioma Segmentation in Sub-Saharan Africa Patient Population (BraTS-Africa)," *arXiv Preprint*, pp. 1-10, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[27] Aklima Akter Lima et al., "A Comprehensive Survey on the Detection, Classification, and Challenges of Neurological Disorders," *Biology*, vol. 11, no. 3, pp. 1-45, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Theresa Neubauer et al., "Soft Tissue Sarcoma Co-segmentation in Combined MRI and PET/CT Data," *Multimodal Learning for Clinical Decision Support and Clinical Image-based Procedures: 10th International Workshop, ML-CDS, and 9th International Workshop, CLIP, Held in Conjunction with MICCAI*, Lima, Peru, pp. 97-105, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[29] Zach Eaton-Rosen et al., "Towards Safe Deep Learning: Accurately Quantifying Biomarker Uncertainty in Neural Network Predictions," *Medical Image Computing and Computer Assisted Intervention-MICCAI: 21st International Conference*, Granada, Spain, pp. 691-699, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[30] Sérgio Pereira et al., "Brain Tumor Segmentation using Convolutional Neural Networks in MRI Images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1240-1251, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[31] Min Zhang et al., "Deep-Learning Detection of Cancer Metastases to the Brain on MRI," *Journal of Magnetic Resonance Imaging*, vol. 52, no. 4, pp. 1227-1236, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[32] S. Immaculate Joy, G. Sriram, and S. Sriram Venkatesan, "Deep CNN-based Multi-Grade Brain Tumor Classification with Enhanced Data Augmentation," *Procedia Computer Science*, vol. 260, pp. 300-307, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[33] Amin Amiri Tehrani Zade et al., "An Improved Capsule Network for Glioma Segmentation on MRI Images: A Curriculum Learning Approach," *Computers in Biology and Medicine*, vol. 148, pp. 1-22, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[34] Md. Faysal Ahamed et al., "A Review on Brain Tumor Segmentation based on Deep Learning Methods with Federated Learning Techniques," *Computerized Medical Imaging and Graphics*, vol. 110, pp. 1-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[35] Mohammad Hamghalam, Baiying Lei, and Tianfu Wang, "Convolutional 3D to 2D Patch Conversion for Pixel-Wise Glioma Segmentation in MRI Scans," *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 5th International Workshop, BrainLes, Held in Conjunction with MICCAI*, Shenzhen, China, pp. 3-12, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[36] Vinod Kumar Dhakshnamurthy et al., "Brain Tumor Detection and Classification using Transfer Learning Models," *Engineering Proceedings*, vol. 62, no. 1, pp. 1-8, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[37] T. Loganayagi et al., "Spinal-QDCNN: Advanced Feature Extraction for Brain Tumor Detection using MRI Images," *European Spine Journal*, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[38] Steven Amaya-Oliva, Luis Mora-Torres, and Wilfredo Ticona, "Comparison of Machine Learning and Traditional Methods for Brain Tumor Detection: A Systematic Review," *Software Engineering: Emerging Trends and Practices in System Development: Proceedings of 14th Computer Science Online Conference*, Czech Republic, vol. 2, pp. 269-291, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[39] Jiwen Zhou et al., "An Adaptive Multi-Stage and Adjacent-Level Feature Integration Network for Brain Tumor Image Segmentation," *Interdisciplinary Sciences: Computational Life Sciences*, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[40] S. Priyadharshini et al., "A Successive Framework for Brain Tumor Interpretation using Yolo Variants," *Scientific Reports*, vol. 15, no. 1, pp. 1-24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[41] Anjali Jain et al., "Advanced Brain Tumor Classification from MRI Images with Vision and Swin Transformer Models," *2025 Emerging Technologies for Intelligent Systems (ETIS)*, Trivandrum, India, pp. 1-5, 2025. [CrossRef] [Google Scholar] [Publisher Link]