



Slice-Group-Level Brain Tumor Classification Using CRNN-GRU

Mohammed Z.M. Almadani¹, Maslita Abd Aziz², T Zalizam T Muda³

¹Quality Assurance & Test Automation Department, Vianeos Dubai, United Arab Emirates

²Faculty of Information & Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

³School of Multimedia Technology and Communication, Universiti Utara Malaysia, Malaysia

Article Info

Article history:

Received Apr 16, 2026

Revised Sep 22, 2026

Accepted Oct 11, 2026

Keywords:

Brain tumor classification;

CRNN-GRU;

group-level evaluation;

inter-slice attention;

T1-contrast-enhanced MRI

ABSTRACT

Interpreting brain MRI for tumor diagnosis is time-consuming and varies between raters, and single-slice decisions can be fragile. This paper compares two classification pipelines on the Brain Tumor Classification (MRI) Dataset from Kaggle (glioma, meningioma, pituitary, no-tumor) of T1-contrast-enhanced images. The first is a per-slice CNN. The second is a per-group CRNN-GRU that reads an ordered sequence of ten slices with a bidirectional GRU and inter-slice attention. Because the dataset provides no patient identifiers, the analysis unit is a surrogate slice group, and the splits are group-disjoint by construction; all models are compared at the group-level. The per-slice CNN reached 80.0% slice-level accuracy. Aggregated to the group-level, the same CNN reached 92.68% by majority voting and 95.12% by mean-probability pooling. The CRNN-GRU reached 90.2% in a single pass, and 87.8% with test-time augmentation. No pairwise difference is statistically significant (exact McNemar, all $p \geq 0.25$). The results do not show that sequential modelling and inter-slice attention improve on simple probability aggregation on this dataset, and the simplest configuration was the most accurate. A content-level audit found a small number of images duplicated across partitions, so all figures should be read as upper bounds. These results describe internal performance on one dataset under group-disjoint evaluation with a content-level leakage audit and do not establish clinical generalization.

This is an open access article under the CC BY-NC license.



Corresponding Author:

Maslita Abd Aziz,

Faculty of Information & Communication Technology,

Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

Jln. Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

Email: maslita@utem.edu.my

1. INTRODUCTION

Brain tumors are serious neurological disorders contributing substantially to global cancer morbidity and mortality. The American Cancer Society estimates approximately 18,330 deaths from brain and central nervous system tumors in the United States in 2025 [1]. Early detection is of paramount importance, insufficient or delayed detection decreases the probability of survival, with glioblastoma patients having a median survival of 12–15 months at advanced stages [2].

Magnetic Resonance Imaging (MRI) is a principal imaging modality in the diagnosis of brain tumors, providing high-resolution images of soft tissue non-invasively. Within a multiparametric MRI protocol, T1-contrast-enhanced (T1-CE) imaging supports the assessment of tumor boundaries and enhancement, while advanced sequences add further tissue characterization [3]. A definitive brain tumor diagnosis however, is not established by any single sequence. It integrates the full MRI protocol with clinical assessment and histopathological confirmation. Automated image classification is

therefore best understood as an assistive computational task rather than a substitute for that pathway. MRI interpretation though clinically valuable is also time consuming [4] and inter-reader variability is a recognized concern, which has motivated structured reporting systems for brain tumor MRI [5]. These challenges have led to the increasing use of deep learning (DL) for medical image interpretation.

Deep learning methods, particularly Convolutional Neural Networks (CNNs), have achieved substantial success in automating medical image diagnosis by extracting rich spatial features from individual 2D image slices [6], [7]. Recent architectures, including EfficientNet and custom multi-block CNNs, report high slice-level classification accuracies exceeding 94% on brain tumor datasets [8], [9]. However, the specific slice-level inference paradigm evaluated in this study exhibits two fundamental limitations; (i) methodologically, it treats every MRI slice as an independent, identically distributed (i.i.d.) observation, discarding the spatial continuity and inter-slice ordering across contiguous slices [10], [11], (ii) clinically, this creates a mismatch with real-world radiological workflows. Radiological interpretation of a brain MRI examination generally considers multiple slices together. It also uses complete sequences not one isolated 2D image. The exact reading workflow varies with the clinical context, the imaging protocol and the diagnostic task [4]. It should be noted that this limitation concerns the inference paradigm, not the 2D convolutional architecture itself. A 2D CNN can also be combined with group-level aggregation. Such an aggregated baseline is evaluated in Section 3.1.2 of this work.

To address this limitation, Recurrent Neural Networks (RNNs), particularly Gated Recurrent Units (GRUs), have been employed for processing ordered sequences in medical imaging [12], [13]. Architecturally, a GRU uses two gates while a LSTM cell uses three. At equal hidden size, a GRU therefore has fewer trainable parameters per unit. Whether this gives faster training or lower memory use in practice can vary. It depends on the implementation, the hidden dimensionality, the sequence length, the hardware and the optimization settings. For this reason, it is not claimed here as an intrinsic property. Bidirectional GRUs have nonetheless been applied successfully to ordered clinical sequence data at scale [13], and to medical image analysis more broadly [14]. In this study the GRU was selected for its architectural compactness on short sequences ($T=10$). An LSTM variant was not trained, so no controlled GRU versus LSTM runtime comparison is reported. Convolutional Recurrent Neural Networks (CRNNs) combine the strengths of both, with CNN layers extracting spatial features from individual slices and RNN layers (LSTM/GRU) capturing sequential dependencies among slices. This hybrid approach allows CRNNs to model anatomical variation of the lesion across spatially ordered slices, improving classification performance [15], [16]. For example, CRNNs have been applied to dynamic functional connectivity networks, demonstrating improved classification of brain diseases by considering inter-slice variation [15]. In multi-slice cardiac MRI, recurrent fully convolutional networks exploit the same inter-slice ordering used here [15]. A CRNN model has likewise shown improved accuracy in brain tumor detection by using both spatial and sequential features in MRI scans [17].

Despite recent progress in combining spatial and sequential features, existing deep learning approaches for brain MRI classification face four notable methodological challenges. First, standard 2D CNNs evaluate contiguous slices independently, which limits their ability to model inter-slice spatial continuity along the axial volume [18]. Second, performing inference on isolated 2D images creates a disparity with clinical workflows, where radiological diagnosis relies on evaluating the patient scan as a complete series [11], [12]. Third, the evaluation of published models may be inclined to group-level leakage whenever multiple slices from the same examination are distributed across partitions [19]. Yagis et al. [20] demonstrated this effect directly for 2D CNNs on brain MRI, reporting substantially inflated accuracy under slice-randomized splitting relative to subject wise splitting. For the specific studies listed in Table 1, however, we do not assert confirmed leakage. One study reports patient-level cross-validation explicitly, while several others do not state the unit at which the split was performed. Therefore, for those studies the correct characterization is an unreported or unverifiable splitting protocol carrying a potential leakage risk, rather than demonstrated patient overlap. Finally, heuristic aggregation methods such as unweighted majority voting over single-slice outputs treat all frames equally, allowing non-diagnostic peripheral slices to influence the final group-level prediction. CNN

encoders, bidirectional GRUs and attention are established components, so their combination is not in itself a methodological contribution. A recurring gap in the multi-slice and CNN-RNN literature is that these models are frequently evaluated at the slice-level, under splitting protocols whose unit of analysis is the image level or is not clearly reported. In such settings, the reported gain of sequence modelling is difficult to separate from any gain that correlated slices may contribute when they are shared across partitions [21]. The contribution of this study is therefore an evaluation protocol, a group-level framework with group-disjoint splits and a content-level leakage audit. This study compares a per-slice CNN and its aggregated variants with the sequential attention model. All models are evaluated at the same clinical decision unit.

Table 1. Related studies on brain tumor MRI classification, showing dataset, unit of analysis, reported split protocol, group separation, architecture and reported accuracy

Study	Reported Dataset	Unit of analysis	Reported Split Protocol	Group/patient separation	Model Architecture	Reported Accuracy
Deepak & Ameer 2019 [18]	Figshare 3,064 slices from 233 patients	Slice	Patient-level five-fold cross validation	Patient-level separation reported	GoogLeNet features + SVM	98.00%
Yagis et al. 2021 [20]	Brain MRI (OASIS, ADNI, PPMI, Versilia); Alzheimer's and Parkinson's classification, not brain tumor	Slice vs subject	Explicit comparison of slice-randomized and subject-wise splits	Subject-wise split enforced in the control condition	2D CNN	Slice-level accuracy inflated by 30 (OASIS), 29 (ADNI), 48 (PPMI) and 55 (Versilia) percentage points under slice-randomized splitting
Swati et al. 2019 [22]	Figshare 3,064 T1-CE slices from 233 patients	Slice	Five-fold cross validation at image level	Not reported at patient level	VGG19 transfer learning with fine tuning	94.82%
Malla et al. 2023 [23]	Figshare 3,064 slices from 233 patients	Slice	Random 70/15/15 image-level partition	Not reported at patient level	VGG16 transfer learning + global average pooling	98.93%
Ahmed et al. 2024 [24]	BrTMHD-2023 primary hospital dataset; Kaggle brain tumor dataset	Slice	Ten-fold cross validation at image level	Not reported at patient level	Hybrid ViT + GRU with explainable AI	98.97% (AdamW, primary dataset)
Priya & Vasudevan	Public brain MRI dataset (four classes)	Slice	Not stated at patient level	Not reported at patient level	Hybrid AlexNet + GRU	97.00%

Study	Reported Dataset	Unit of analysis	Reported Split Protocol	Group/patient separation	Model Architecture	Reported Accuracy
2024 [25]						
This study	Brain Tumor Classification (MRI) Dataset [26] reorganized into 197 slice groups of ten slices	Slice group	Deterministic group-disjoint train/validation/test partition (131/25/41 groups)	Group-disjoint by construction; groups are surrogate not clinically verified	CNN encoder + BiGRU + inter-slice attention	80.0% slice-level (CNN); 90.2% group-level, proposed model (37/41)

Note: These studies use different datasets, units, preprocessing and split protocols, so the accuracies are descriptive and not directly comparable (not a direct benchmark). A slice-level accuracy and a group-level accuracy are not competing scores. Where a study does not state its split unit, we describe a potential, unreported leakage risk, not confirmed leakage.

The core problem is the mismatch between clinical practices where radiologists interpret MRI as coherent sequences of slices and most automated systems that process slices independently. This leads to fragile single-slice decisions and inflated performance metrics due to leakage. This study enforces group-disjoint splits, so no slice group contributes images to more than one partition. It then compares a per-slice CNN baseline with a per-group CRNN-GRU model that uses sequence-level augmentation and inter-slice attention. The splits are group-disjoint by construction, so the evaluation is described as group-disjoint evaluation with a content-level leakage audit rather than as leakage-free. The audit in Section 2 found 14 images duplicated across partitions, and this is disclosed throughout. The sequence dimension, which indexes anatomical position within a single examination, is referred to as inter-slice spatial context.

To position the present study within the literature, Table 1 summarizes a set of related studies on brain tumor MRI classification [18], [20], [22], [23], [24], [25] shown for context rather than as a direct benchmark. A recurrent pattern in work using the Figshare and Nickparvar datasets is the combination of slice-level evaluation with an image-level partition. Yagis et al. [20] showed that the combination of evaluating on single slices while splitting the data by image can make accuracy appear higher than it truly is. This finding motivates the design of this study with three objectives. First, we evaluate a per-slice CNN baseline. Second, we test whether adding a GRU to a convolutional encoder while respecting the anatomical order of slices improves classification. Third, we analyze the performance and computational trade-offs between the approaches at the same group-level decision unit.

The contributions are stated separately as a methodological contribution, an empirical finding and standard good practice; (i) Methodological contribution; a group-level evaluation protocol with group-disjoint splits and a content-level leakage audit for the dataset, in which surrogate slice groups are formed once, held fixed, and kept disjoint across partitions, and several inference and aggregation strategies are compared at the same decision unit. (ii) Empirical finding; under this protocol the proposed sequential model classifies 37 of 41 groups correctly, while the same per-slice CNN, aggregated by majority voting or mean-probability pooling, classifies 38 of 41 and 39 of 41 with no pairwise difference is statistically significant under the exact McNemar test. (iii) The finding is therefore negative; this study does not show that sequential modelling and inter-slice attention outperform simple probability aggregation on this dataset. Group-disjoint splitting, fixed seeds and standard metrics are reproducible practice, and no deployment claim is made.

2. RESEARCH METHOD

This study employs a comparative experimental design with two complementary pipelines to evaluate brain tumor classification from T1-CE MRI; (i) a per-slice CNN baseline that analyzes individual slices

independently as a fast, lightweight reference, and (ii) a per-group CRNN-GRU that processes ordered sequences of slices to capture cross-slice contextual dependencies as in Figure 1. To ensure fair and clinically meaningful comparison, three principles were enforced; (i) no group-level leakage (the same surrogate slice group never appears in two partitions, verified programmatically); (ii) reproducibility through fixed random seeds (42), deterministic cuDNN settings were used where available, and persistence of all artifacts; and (iii) broad metric evaluation including confusion matrices, per-class precision/recall/F1/specificity, ROC-AUC, and PR-AUC.

The software environment is reported for reproducibility. Models and data loading use PyTorch with torchvision, metrics use scikit-learn, and analysis and plotting use NumPy and Matplotlib. The experiments ran on Ubuntu with a single NVIDIA GeForce RTX 3090 GPU (24 GB) and CUDA. The seed was applied to the Python, NumPy and PyTorch generators. Two data-loader workers were used, and evaluation loaders ran without shuffling, so inference order is deterministic.

The experiment used a single random seed and was not repeated across seeds. All point estimates below are therefore reported with 95% confidence intervals from the binomial sampling distribution of the test cohort. The lack of across-seed variance is treated as a limitation (Section 3.4.3).

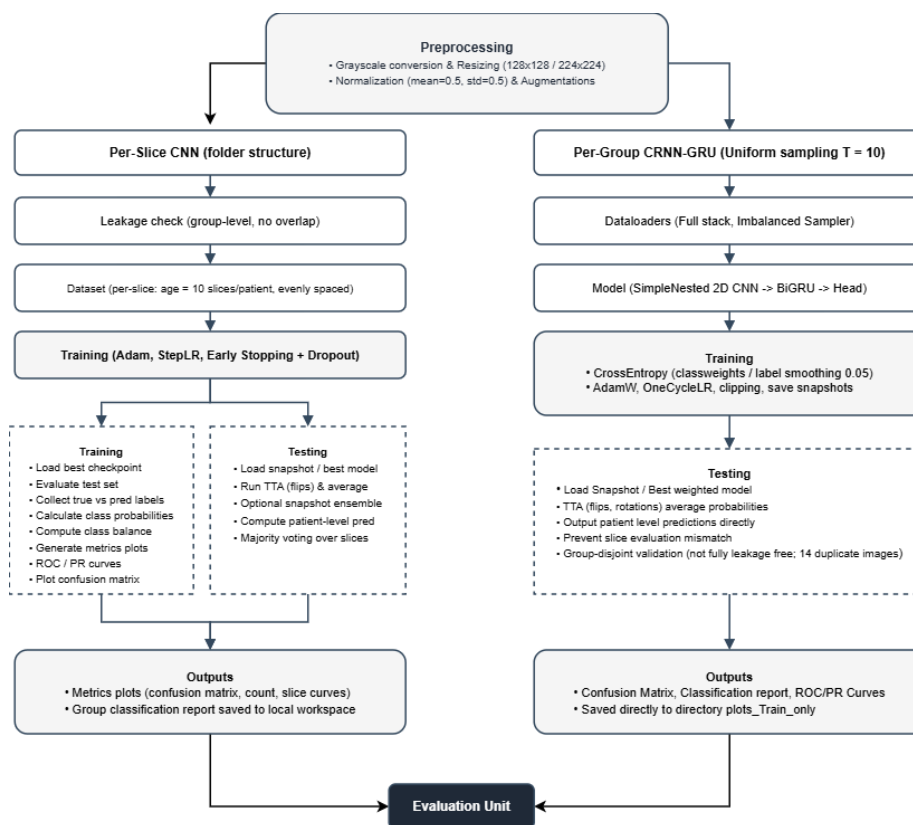


Figure 1. End-to-End Evaluation Pipeline

The study utilized the Brain Tumor Classification (MRI) Dataset of Bhuvaji et al. [26], a publicly available Kaggle collection of 2,870 images consisting of four classes which are glioma (826), meningioma (822), pituitary (827), and no-tumor (395). The images are 2D grayscale T1 contrast enhanced MRI slices distributed as JPEG format. The publisher organizes them into two top-level folders, Training and Testing, each with the four class subfolders. The dataset is redistributed without DICOM headers. Acquisition parameters (field strength, repetition and echo times, slice thickness, in-plane resolution) and patient metadata (age, sex, scanner model) are therefore unavailable, and the number of contributing institutions is not documented. In-plane dimensions vary between files, so

every image is resampled to a fixed size during preprocessing. The class labels follow the publisher; glioma, meningioma and pituitary adenoma are the three tumor categories, and no-tumor denotes a scan with no labelled lesion of these categories. For this study, the local copy was reorganized into a split/class/group/ hierarchy so that all slices of one slice group stay within a single partition. This changes only the folder layout; image content and class labels are unchanged.

The dataset provides no patient identifiers, which affects how the results must be interpreted. The images are stored as a flat set of files per class, with no patient subdirectories. The analysis units were therefore constructed mechanically. Each class file list was sorted and divided into consecutive, non-overlapping blocks of exactly ten slices, and each block was written to a directory named `class_patient_NNN`. All 197 groups contain exactly ten slices, and consecutively numbered files fall on either side of adjacent block boundaries. The directory names thus encode block position, not patient identity. These units are referred to as slice groups, or pseudo-subjects, throughout the manuscript, and not as patients.

No available check can establish patient identity. Anatomical continuity within a folder is consistent with a single examination but does not prove it. Uniform image dimensions and contrast are properties of the acquisition and redistribution pipeline, not identifiers. Inspecting a subset of folders cannot support a conclusion about the remainder. Most importantly, because the publisher supplies no identifier, it cannot be excluded that one individual contributed images to more than one group or class. The protocol was designed so that no slice group appeared in more than one partition, and this was verified exhaustively using the group identifier. However, this verification does not establish that each group corresponds to a single patient. As the content-level audit in the next subsection shows, group-disjoint identifiers do not by themselves prevent leakage of image content, so the reported group-level results should be read as an upper bound. The dataset comprises 197 groups consisting of 1,970 slices: 131 training, 25 validation, and 41 test. The partition was formed once, before modelling, and held fixed for all experiments.

The number of available slices per group varies, so a fixed-length input of $T=10$ slices is obtained by index selection, not by interpolating image content. For a group with n slices, the ten indices are the integer parts of an evenly spaced grid over $[0, n-1]$, so the selected slices span the whole stack rather than ten adjacent, near-duplicate frames. Where $n < 10$, index mirroring reflects the sequence at the stack boundaries until ten indices are obtained; this reuses existing slices without creating new content. In this dataset every group already contains ten slices, so this rule is never triggered. No interpolation is applied along the slice axis; the only interpolation is bilinear resampling within each 2D image to the fixed input size.

Slice order is set by sorting filenames in ascending lexicographic order within each group, which reproduces the publisher's export order. Because the dataset carries no DICOM headers, no acquisition metadata of *InstanceNumber*, *SliceLocation* and *ImagePositionPatient* is available to verify this order against true anatomical position. The BiGRU therefore models dependencies across a consistently ordered sequence whose correspondence to superior-inferior anatomical order is assumed rather than verified. This assumption is listed among the limitations, and the bidirectional formulation reduces, though it does not remove sensitivity to ordering direction. The input tensor has shape $(B, T, C, H, W) = (B, 10, 1, 224, 224)$; a batch of B groups, each a sequence of ten single-channel 224×224 frames. The sequence is processed frame-wise by a shared 2D encoder and then aggregated recurrently; it is not a native 3D volume processed by 3D convolutions.

The partition is deterministic where group identifiers are enumerated in sorted order, shuffled once under seed 42, and allocated class-by-class to the training, validation and test partitions in the proportions 66.5% / 12.7% / 20.8% by group. The split is therefore stratified by class at the group-level and is reproduced exactly on every run. The validation partition is used only by the per-slice CNN, for early stopping and checkpoint selection, and never for gradient updates; the CRNN-GRU does not use it. Disjointness is checked programmatically before training, a routine collects the group identifiers of each partition and raises an exception if any pairwise intersection is non-empty. The group-level class distribution of all three partitions is given in Table 2. Reporting every partition, rather than the test set

alone, makes it possible to check whether any class is concentrated in one partition, which matters here because class imbalance is handled through both a weighted loss and a balanced sampler.

Table 2. Distribution of slice groups by class across the training, validation and test partitions (197 groups in total)

Class	Train	Validation	Test	Total
Glioma	39	7	12	58
Meningioma	36	7	11	54
No-tumor	17	4	6	27
Pituitary	39	7	12	58
Total	131	25	41	197

Note: Counts are slice groups; each group contains exactly ten slices, so slice counts are ten times the values shown.

Prevalence is closely matched across the three partitions, which limits distribution shift, and the ordering of class sizes is preserved. No-tumor is the minority class in every partition: 13.0% of training groups, 16.0% of validation groups and 14.6% of test groups. After each group is capped at ten evenly spaced slices, the per-slice CNN pipeline ingests 1,310 training slices (glioma 390, meningioma 360, no-tumor 170, pituitary 390), 250 validation slices (glioma 70, meningioma 70, no-tumor 40, pituitary 70) and 410 test slices (glioma 120, meningioma 110, no-tumor 60, pituitary 120).

Because group identifiers alone cannot prove that the partitions share no content, a second audit was performed at the level of image bytes. Every image was hashed with MD5, and the hashes were compared across partitions; this check is independent of filenames and folder structure. It found 14 images that are byte-identical to an image in a different partition: 9 between test and training, 3 between training and validation, and 2 between test and validation. By class, the affected images are no-tumor 10, pituitary 3, glioma 1 and meningioma 0. Each pair carries different filenames, which is why the identifier-based check did not detect them. The source collection is known to contain duplicated files, and because the partition was drawn without a de-duplication step, copies of the same image fell on opposite sides of a partition boundary by chance. The consequence is direct: eleven of the 410 test slices are images seen during training or validation, so the test partition is not fully held out and the reported test metrics are optimistically biased by an unknown amount. At most 11 of the 41 test groups contain such an image; the rest are unaffected. Because the contamination concentrates in the no-tumor class, which has the smallest test support (six groups) and for which the proposed model reports perfect recall, the no-tumor result should not be read as evidence of separability.

In the per-slice CNN pipeline, each image is converted to single-channel grayscale, resized to 128×128 by bilinear interpolation, converted to a tensor in $[0, 1]$, and normalized with mean 0.5 and standard deviation 0.5, which maps intensities to approximately $[-1, +1]$. No skull stripping, N4 bias-field correction, z-score standardization or percentile clipping is applied. Each group contributes at most ten evenly spaced slices, selected by the index rule above, to reduce redundancy between correlated adjacent slices. Training-time augmentation consists of a random horizontal flip ($p = 0.5$), a random rotation drawn uniformly from $\pm 15^\circ$, and a random affine transform with translation up to $\pm 5\%$ per axis and isotropic scaling in $[0.95, 1.05]$; no shear and no intensity augmentation are used. Vertical flips are excluded because they would invert the superior-inferior axis. Validation and test data pass through the deterministic chain only. In the CRNN-GRU pipeline, every frame is converted to grayscale, resized to 224×224 by bilinear interpolation, and normalized to $[-1, +1]$ in the same way. Frame-level augmentation consists of a random affine transform (rotation $\pm 10^\circ$, translation up to $\pm 2\%$, isotropic scaling $[0.95, 1.05]$, shear up to 5°) and a random horizontal flip ($p = 0.5$), applied only during training, with narrower ranges than the per-slice pipeline to preserve coherence along the sequence.

Algorithm 1. Per-frame stochastic augmentation of a slice sequence (training only)

Input: slice sequence $X = (x_1, \dots, x_T)$, $T = 10$; each x_t a 224×224 grayscale frame.

Output: augmented sequence $\tilde{X} = (\tilde{x}_1, \dots, \tilde{x}_T)$.

1: for $t = 1$ to T do

2: $\theta_t \leftarrow \text{Uniform}(-10^\circ, +10^\circ)$ // rotation, sampled per frame

```

3:  ( $\Delta x_t, \Delta y_t$ )  $\leftarrow$  Uniform( $-0.02, +0.02$ )  $\times$  224 // translation in pixels
4:   $s_t \leftarrow$  Uniform(0.95, 1.05) // isotropic scale
5:   $\phi_t \leftarrow$  Uniform( $-5^\circ, +5^\circ$ ) // shear
6:   $A_t \leftarrow$  AffineMatrix( $\theta_t, \Delta x_t, \Delta y_t, s_t, \phi_t$ )
7:   $\tilde{x}_t \leftarrow$  BilinearWarp( $x_t, A_t$ ) // interpolation within the 2D frame only
8:  if Bernoulli(0.5) then  $\tilde{x}_t \leftarrow$  HorizontalFlip( $\tilde{x}_t$ )
9:  end for
10: return  $\tilde{X}$ 

```

Transformation parameters are sampled independently for each frame, so inter-slice geometric correspondence is preserved approximately rather than exactly, resulting in small residual geometric jitter that introduces additional variation during training. A synchronized variant that draws one parameter set per sequence is available in the codebase but was not used here. Algorithm 1 specifies the per-frame transformation. Parameters are drawn independently for each frame in the loop over t in Algorithm 1, so the augmentation is not synchronized across the sequence. All interpolation is bilinear and confined to the 2D plane of each frame; no interpolation is applied along the slice axis. Validation and test sequences bypass Algorithm 1 and pass through the deterministic resize and normalize chain only.

The geometric transforms are class preserving for the four categories. Rotation, small translation and mild scaling change the pose of the anatomy without changing the lesion category, and horizontal flipping uses the approximate left to right symmetry of the head without changing the tumor type. The input resolution differs between the two pipelines, at 224×224 for the CRNN-GRU and 128×128 for the per-slice baseline. This reflects their intended roles, where the per-slice model is a fast triage baseline and the sequential model is an accuracy-oriented pipeline. The difference in resolution is a genuine confound and is not resolved here. Higher resolution can improve classification on its own, so no difference between the two models can be attributed to sequence modelling and attention alone. No controlled experiment isolating the effect of attention was performed, because no non-attention sequential model exists in the project. A resolution-matched, protocol-matched and architecture-matched ablation therefore remains open and is listed among the outstanding items. What is reported instead is a comparison of six verified, checkpoint-traceable inference configurations in Table 6, and none of them differs from another at conventional significance.

The two architectures are specified separately in Table 3 and 4, with their layers specified to provide sufficient detail for their reconstruction. In Table 3, weights use the PyTorch defaults, Kaiming uniform for convolutional and linear layers. The layer wise sum matches the checkpoint exactly with 23,520 in the feature blocks and 4,195,588 in the classifier, giving 4,219,108. The distribution is extreme for the three convolutional blocks that perform all feature extraction hold 23,520 parameters (0.6%), while the first fully connected layer alone holds 4,194,560. This baseline is therefore almost entirely a flatten and classify network with a very small feature front end with a point that matters when its accuracy is compared with the sequential model in Section 3.

Table 3. Layer-wise specification of the per-slice CNN baseline, 128×128 input, 4,219,108 trainable parameters

Stage	Layer	Configuration	Output shape	Parameters
Block 1	Conv2d	$1 \rightarrow 16$, kernel 3×3 , stride 1, padding 1	(B, 16, 128, 128)	160
	BatchNorm2d + ReLU	16 channels	(B, 16, 128, 128)	32
	MaxPool2d	kernel 2×2 , stride 2	(B, 16, 64, 64)	0
	Dropout2d	$p = 0.25$	(B, 16, 64, 64)	0
Block 2	Conv2d	$16 \rightarrow 32$, kernel 3×3 , stride 1, padding 1	(B, 32, 64, 64)	4,640
	BatchNorm2d + ReLU	32 channels	(B, 32, 64, 64)	64
	MaxPool2d	kernel 2×2 , stride 2	(B, 32, 32, 32)	0
	Dropout2d	$p = 0.25$	(B, 32, 32, 32)	0
Block 3	Conv2d	$32 \rightarrow 64$, kernel 3×3 , stride 1, padding 1	(B, 64, 32, 32)	18,496
	BatchNorm2d + ReLU	64 channels	(B, 64, 32, 32)	128
	MaxPool2d	kernel 2×2 , stride 2	(B, 64, 16, 16)	0

Stage	Layer	Configuration	Output shape	Parameters
Head	Dropout2d	p = 0.25	(B, 64, 16, 16)	0
	Flatten	64×16×16	(B, 16384)	0
	Linear + ReLU	16384→256	(B, 256)	4,194,560
	Dropout	p = 0.5	(B, 256)	0
	Linear	256→4	(B, 4)	1,028
			Total	4,219,108 (exact)

In Table 4, the encoder is a five block, ten layer VGG style network in which each block applies two bias-free convolutions with batch normalization, with 2×2 max-pooling after the first four blocks and none after the fifth, so the map is held at 14×14 before global average pooling. The layer-wise sum matches the checkpoint's trainable-parameter total of 4,527,012: encoder 2,419,104, bidirectional GRU 1,972,224, attention 132,608 and head 3,076. The two models are close in size, 4,527,012 against 4,219,108 (a ratio of 1.07). This near-equality hides an architectural contrast that matters for interpretation: the baseline spends 99.4% of its parameters on a flattened fully connected classifier and only 23,520 on feature extraction, whereas the sequential model spends 97.0% on the encoder and recurrent module and only 3,076 on its head. The sequential model's feature extractor is thus about 103 times larger, with ten convolutional layers against three. Any accuracy difference between the two therefore cannot be attributed to total parameter count, which is comparable; it must be read against large differences in representational capacity, input resolution and inference protocol.

Table 4. Layer-wise specification of the proposed CRNN-GRU, 224×224 input, comprising the shared encoder, bidirectional GRU, inter-slice attention and classification head, 4,527,012 trainable parameters

Stage	Layer	Configuration	Output shape	Parameters
Encoder block 1	2 × (Conv2d-BN-ReLU)	1→32, 32→32; 3×3, stride 1, padding 1, bias=False	(B-10, 32, 224, 224)	9,632
	MaxPool2d	kernel 2×2, stride 2	(B-10, 32, 112, 112)	0
Encoder block 2	2 × (Conv2d-BN-ReLU)	32→64, 64→64; 3×3, padding 1, bias=False	(B-10, 64, 112, 112)	55,552
	MaxPool2d	kernel 2×2, stride 2	(B-10, 64, 56, 56)	0
Encoder block 3	2 × (Conv2d-BN-ReLU)	64→128, 128→128; 3×3, padding 1, bias=False	(B-10, 128, 56, 56)	221,696
	MaxPool2d	kernel 2×2, stride 2	(B-10, 128, 28, 28)	0
Encoder block 4	2 × (Conv2d-BN-ReLU)	128→256, 256→256; 3×3, padding 1, bias=False	(B-10, 256, 28, 28)	885,760
	MaxPool2d	kernel 2×2, stride 2	(B-10, 256, 14, 14)	0
Encoder block 5	2 × (Conv2d-BN-ReLU)	256→256, 256→256; 3×3, padding 1, bias=False; no pooling	(B-10, 256, 14, 14)	1,180,672
	AdaptiveAvgPool2d	output 1×1	(B-10, 256, 1, 1)	0
	Linear	256→256, bias=True	(B, 10, 256)	65,792
	Encoder subtotal	10 convolutional layers	—	2,419,104
Sequence	Bidirectional GRU	input 256, hidden 256, 2 layers	(B, 10, 512)	1,972,224
Attention	LayerNorm(512); Linear 512→256 + Tanh; Linear 256→1 (bias=False); softmax over T	additive attention over the 10 slice positions	(B, 512)	132,608

Stage	Layer	Configuration	Output shape	Parameters
Head	LayerNorm(512); Linear 512→4	—	(B, 4)	3,076
Total				4,527,012

Note: Input (B, 10, 1, 224, 224); the encoder is applied frame-wise with shared weights by reshaping to (B·10, 1, 224, 224). The layer-wise sum reproduces the checkpoint's trainable-parameter total of 4,527,012 exactly. BatchNorm running statistics are non-trainable buffers and are excluded.

Let a group be the ordered frame set $x_1, \dots, x_T$ with $T = 10$. The shared 2D encoder f_θ maps each frame to $z_t = f_\theta(x_t) \in \mathbb{R}^{256}$, giving $Z = z_1, \dots, z_T$. A two layer bidirectional GRU processes Z in both directions along the slice index as defined in Eq. (1):

$$\begin{aligned} \rightarrow h_t &= GRU_{fwd}(z_t, \rightarrow h_{t-1}), \leftarrow h_t = GRU_{bwd}(z_t, \leftarrow h_{t+1}), \rightarrow h_t, \\ &\leftarrow h_t \in \mathbb{R}^{256}, \rightarrow h_0 = \leftarrow h_{T+1} = 0 \end{aligned} \quad (1)$$

Each direction uses the standard GRU update, with reset gate r_t , update gate u_t and candidate $\hat{h}_t$: $r_t = \sigma(W_r z_t + U_r h_{t-1} + b_r)$; $u_t = \sigma(W_u z_t + U_u h_{t-1} + b_u)$; $\hat{h}_t = \tanh(W_h z_t + U_h(r_t \odot h_{t-1}) + b_h)$; $h_t = (1 - u_t) \odot h_{t-1} + u_t \odot \hat{h}_t$, where σ is the logistic sigmoid and $\odot$ the Hadamard product. The two directional states are concatenated, $h_t = [\rightarrow h_t; \leftarrow h_t] \in \mathbb{R}^{512}$, giving an output of shape (B, T, 512).

Inter-slice attention weights in Eq. (2) then assigns one scalar weight per slice. With $\hat{h}_t = \text{LayerNorm}(h_t)$:

$$\begin{aligned} e_t &= w_a^T \tanh(W_h \hat{h}_t + b_h) + b_a; \alpha_t = \exp(e_t) / \sum_j \exp(e_j), \\ \sum_t \alpha_t &= 1; c = \sum_t \alpha_t h_t \in \mathbb{R}^{512} \end{aligned} \quad (2)$$

Here $W_h \in \mathbb{R}^{256 \times 512}$, $b_h \in \mathbb{R}^{256}$, $w_a \in \mathbb{R}^{256}$ and $b_a \in \mathbb{R}$ are trainable, the softmax is taken over the T slice positions (not over channels), and c is the group-level context vector. Classification applies a second LayerNorm and a linear map: $\hat{y} = \text{softmax}(W_y \text{LayerNorm}(c) + b_y) \in \mathbb{R}^4$. The network is trained end to end.

For the per-slice CNN baseline, the objective is cross entropy with inverse frequency class weights. Optimization uses Adam with learning rate 1×10^{-3} , weight decay 1×10^{-4} , $\beta_1 = 0.9$, $\beta_2 = 0.999$ and with a StepLR schedule that multiplies the learning rate by 0.5 every 20 epochs. Batches of 64 slices are drawn with a WeightedRandomSampler with replacement, so minority classes appear in every batch. No gradient clipping and no label smoothing are used. Training runs for at most 80 epochs with early stopping on validation accuracy, minimum improvement 0.002, patience 10; the best-validation checkpoint is reloaded before test evaluation. For the CRNN-GRU, the objective is cross-entropy with normalized class weights and label smoothing 0.05. Optimization uses AdamW with learning rate 3×10^{-4} , weight decay 1×10^{-4} and with a OneCycleLR schedule stepped once per batch and gradient clipping at a maximum global norm of 5.0. The training batch size is 4 sequences and the evaluation batch size is 1. Training runs for a fixed 48 epochs, with snapshots every 12 epochs, without early stopping or validation monitoring; the final snapshot is used for the reported results. The two pipelines therefore differ in optimizer, schedule, batch size and regularization, adapted to their different input granularity. This is a confounding factor when attributing any performance difference to the architecture, alongside the resolution difference noted above.

At inference, the CRNN-GRU averages softmax probabilities over $n = 12$ augmented passes of the whole sequence; the identity pass, a horizontal flip, and small in-plane rotations of $\pm 5^\circ$, $\pm 10^\circ$ and $\pm 15^\circ$, each applied identically to all ten frames of a pass. Each transform is label-preserving; vertical flips are excluded, since they would invert the superior-inferior axis. Test-time augmentation is an inference-time technique, not an architectural property. Applying it to the proposed model while evaluating the baseline without it means part of any difference may be attributable to the inference strategy; an equivalent protocol was not applied to the baseline, and a paired with and without TTA comparison is required before the effect can be decomposed. Section 3.2 therefore reports the proposed model's accuracy for each protocol separately. Measured against the saved checkpoint, this TTA

reduces group-level accuracy from 90.24% (37/41, single pass) to 87.80% (36/41), so it is not beneficial for this model on this test set; the single-pass result is the headline figure unless stated otherwise.

All results are reported on the held-out test partition. The per-slice CNN uses the validation partition for early stopping and checkpoint selection. The CRNN-GRU is trained without a validation loop: its hyperparameters, schedule length and number of TTA passes were fixed a priori, and the final snapshot was used for evaluation. The test partition was evaluated once for each model, and no hyperparameter was changed on the basis of a test result. Metrics are defined one vs rest per class: $precision = TP / (TP + FP)$; $recall(sensitivity) = TP / (TP + FN)$; $specificity = TN / (TN + FP)$, $withTN = N - (rowtotal + columntotal - diagonal)$; $F1 = 2 \cdot precision \cdot recall / (precision + recall)$. Macro averaging (the unweighted mean of the four per-class values) is the primary reporting mode because it is insensitive to prevalence; weighted averaging weights by support; micro averaging pools TP, FP and FN and equals overall accuracy in single-label classification. ROC-AUC is computed one-vs-rest, and PR-AUC is reported as average precision and must be read against prevalence [27]. Because the cohort contains only 41 groups [28], every headline metric is reported with a two-sided 95% confidence interval from Wilson score intervals for proportions. The statistical unit is the group, never the slice: the 410 test slices are not independent, since ten come from each group, so slice-level intervals are descriptive only. All model comparisons are made at the group-level, so both models are judged at the same clinical decision unit.

3. RESULTS AND DISCUSSION

After enforcing group-disjoint splits, the CNN per-slice pipeline utilized 1,310 training slices (glioma 390, meningioma 360, no-tumor 170, pituitary 390) and 410 test slices (glioma 120, meningioma 110, no-tumor 60, pituitary 120). For the CRNN-GRU per-group pipeline, the test set consisted of 41 slice groups (glioma 12, meningioma 11, no-tumor 6, pituitary 12). These distributions reflect the class imbalance inherent in the dataset while maintaining group-level isolation.

3.1 Diagnostic Performance of the Baseline Model

3.1.1 Slice-level CNN Baseline (N=410 Slices)

The CNN baseline converged smoothly. The per-slice CNN was evaluated first, at the slice-level, as a descriptive baseline. On the 410 test slices it reached 80.0% accuracy (328/410; 95% CI 75.9–83.6%), with macro precision 0.788, macro recall 0.812, macro F1 0.797 and macro specificity 0.933. Per-class results are given in Table 5, and the confusion matrix and the one-vs-rest ROC and precision-recall curves in Figures 2 and 3. Training was stable, Figure 4 shows the accuracy and loss curves, with a small training validation gap and a best validation accuracy of 0.860 at epoch 56. This indicates stable optimization, not generalization, which would require an independent test set. The slice-level figures are reported for description only. Because ten slices come from each group, slices are not independent observations, so all model comparisons below are made at the group-level.

Table 5. Per-class slice-level results for the CNN baseline on the 410 test slices

Class	Precision	Recall	Specificity	F1	Support
Glioma	0.830	0.758	0.930	0.792	120
Meningioma	0.670	0.591	0.910	0.628	110
No-tumor	0.757	0.933	0.983	0.836	60
Pituitary	0.897	0.967	0.990	0.931	120

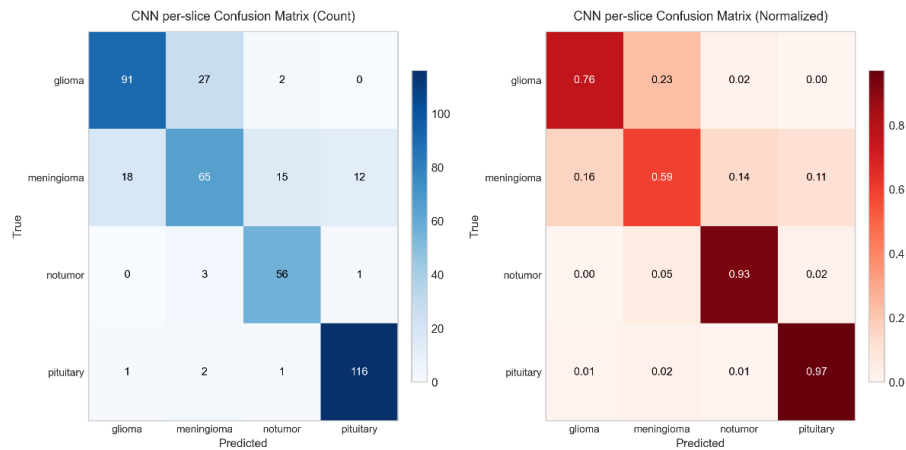


Figure 2. Confusion matrix for CNN per-slice baseline

Meningioma was the weakest class at the slice-level (recall about 0.59), which is consistent with its appearance overlapping other classes and with the label noise known in this dataset. Pituitary and no-tumor were the strongest.

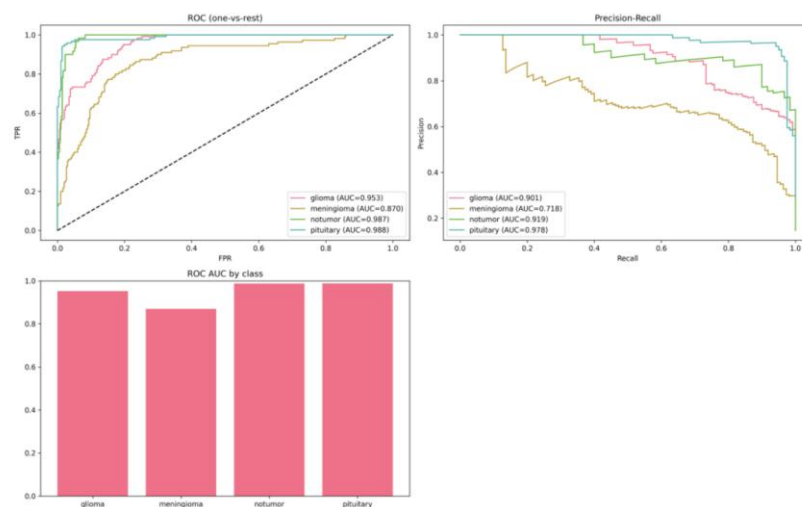


Figure 3. ROC and precision recall curves for CNN per-slice baseline

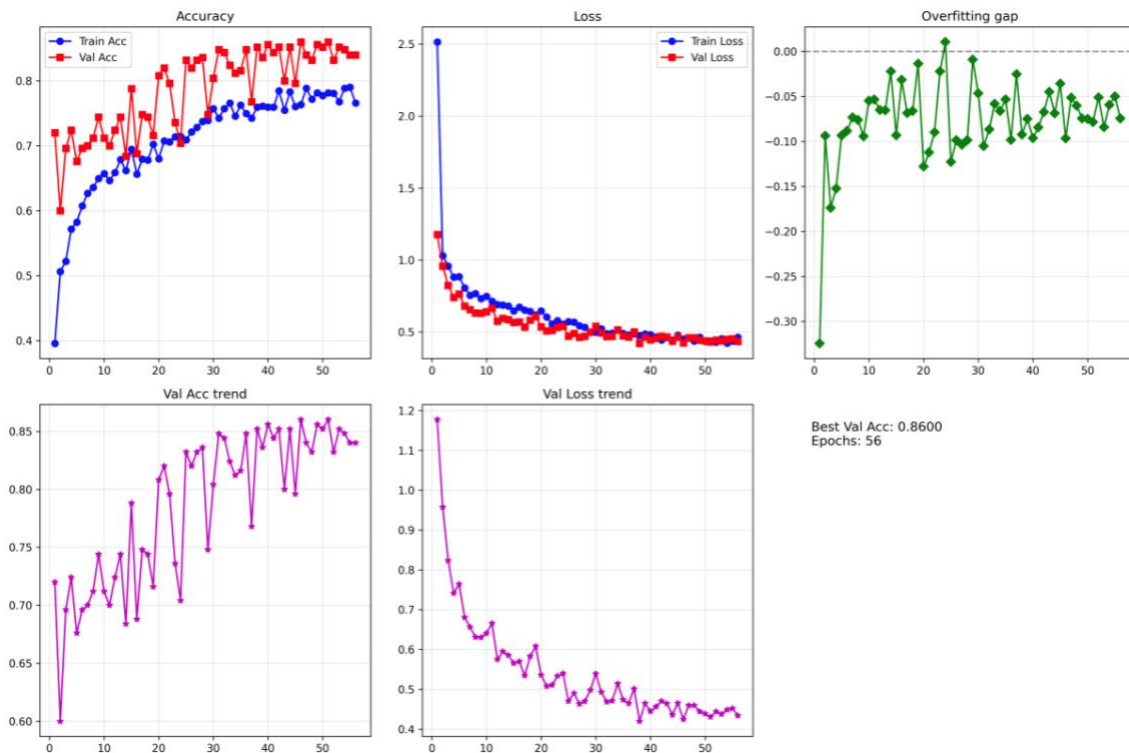


Figure 4. CNN baseline training history. The top row shows training and validation accuracy (left), loss curves (middle), and overfitting gap (right). The bottom row presents detailed validation accuracy and loss trends. The model achieved the best validation accuracy of 0.860 at epoch 56 with smooth convergence and minimal overfitting.

3.1.2 Group-level Baseline: Majority Vote (N=41 slices group)

To evaluate at the clinical decision unit, the CNN's slice predictions were aggregated to the group-level, and the sequential model was evaluated directly at the group-level. Six checkpoint-verified configurations are compared in Table 6, all on the same 41 test groups: the per-slice CNN aggregated by majority voting and by mean-probability pooling, each with and without test-time augmentation, and the proposed CRNN-GRU with and without test-time augmentation.

Table 6. Group-level comparison of the six checkpoint-verified configurations on the 41 test groups, with 95% confidence intervals and macro-averaged metrics

Model	Protocol	Unit	Accuracy (95% CI)	Macro P	Macro R	Macro F1	Macro Spec
CNN baseline	Per-slice (descriptive; different unit)	Slice (410)	80.00% (328/410); CI 75.9-83.6%	0.788	0.812	0.797	0.933
CNN, majority vote	Unweighted vote over 10 slices/group	Group (41)	92.68% (38/41); CI 80.6-97.5%	0.918	0.932	0.915	0.977
CNN, mean probability	Mean of 10 softmax vectors, arg-max	Group (41)	95.12% (39/41); CI 83.9-98.7%	0.938	0.955	0.939	0.986
CNN, mean probability + TTA	Mean prob. + 12-pass TTA on CNN	Group (41)	95.12% (39/41); CI 83.9-98.7%	0.945	0.955	0.946	0.984
Proposed CRNN-GRU	Single inference pass, no TTA	Group (41)	90.24% (37/41); CI 77.5-96.1%	0.889	0.911	0.889	0.970

Model	Protocol	Unit	Accuracy (95% CI)	Macro P	Macro R	Macro F ₁	Macro Spec
Proposed CRNN-GRU + TTA	12-pass test-time augmentation	Group (41)	87.80% (36/41); CI 74.5–94.7%	0.868	0.890	0.872	0.960

Notes: All evaluated on the same 41 test groups unless the evaluation unit is stated otherwise. The series varies the aggregation applied to two trained models; it does not isolate individual architectural components.

The CNN aggregated by majority voting reached 92.68% (38/41; 95% CI 80.6–97.5%), and by mean-probability pooling 95.12% (39/41; 95% CI 83.9–98.7%). The proposed CRNN-GRU reached 90.24% (37/41; 95% CI 77.5–96.1%) in a single pass, and 87.80% (36/41) with twelve-pass test-time augmentation. The numerically strongest configuration was the simplest one, mean-probability pooling of independent per-slice predictions. Figure 5 shows the majority-vote confusion matrix, and Figure 6 compares the mean-probability CNN (a) with the proposed CRNN-GRU (b) at the group-level.

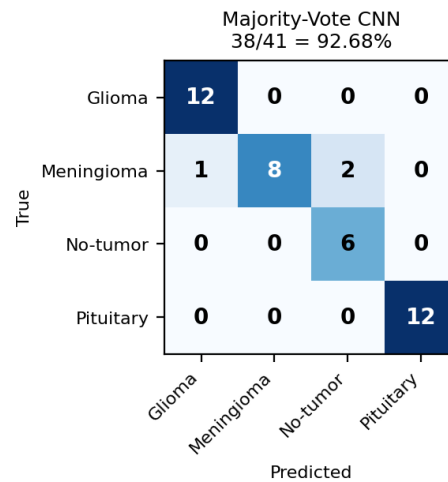


Figure 5. Confusion matrix for the majority-vote CNN at the slice-group-level; 38 of 41 groups correct (92.68%).

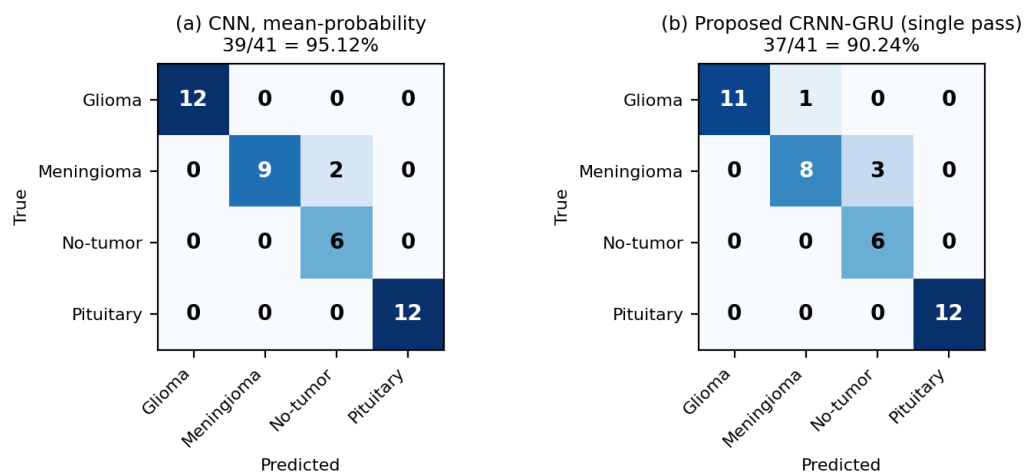


Figure 6. Comparative slice-group-level confusion matrices (a) CNN mean-probability pooling, the strongest configuration (95.12%, 39/41); (b) proposed CRNN-GRU, single inference pass (90.24%, 37/41)

Per-class results for the proposed model (single pass) are given in Table 7, with recall 0.917 (glioma), 0.727 (meningioma), 1.000 (no-tumor) and 1.000 (pituitary). The one-vs-rest ROC and precision-recall curves, with group-level bootstrap confidence intervals ($B = 10,000$), are in Figure 7; macro ROC-AUC

is 0.974 and macro PR-AUC 0.946. These intervals are wide because the cohort is small. With only 6 no-tumor and 11 meningioma test groups, the bootstrap intervals for those classes extend to the ceiling of 1.000, so a point estimate near 1.000 reflects few or no misranked groups in this sample rather than perfect separation in the population.

Table 7. Per-class results, proposed CRNN-GRU

Class	Correct	Total	Recall	Precision	F1
Glioma	11	12	0.917	1.000	0.957
Meningioma	8	11	0.727	0.889	0.800
No-tumor	6	6	1.000	0.667	0.800
Pituitary	12	12	1.000	1.000	1.000

Note: single inference pass (no test-time augmentation), slice-group-level ($N = 41$)

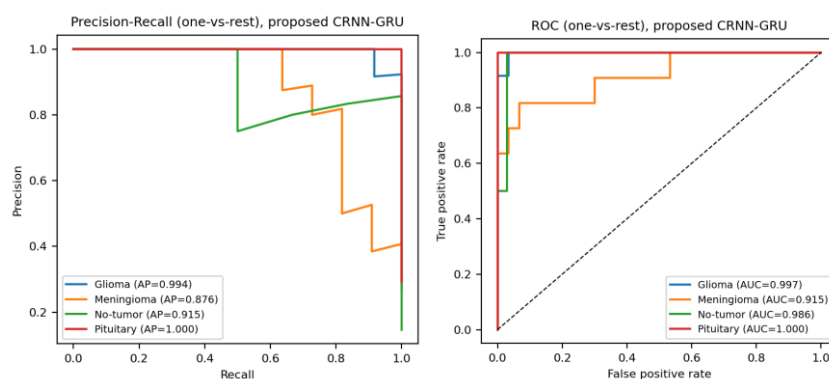


Figure 7. ROC and precision-recall curves, proposed CRNN-GRU, single pass (AUCs glioma 0.997, meningioma 0.915, no-tumor 0.986, pituitary 1.000).

Statistical comparison used the exact two-sided McNemar test on paired per-group outcomes as in Table 8. No pairwise difference among the group-level configurations is significant, majority-vote CNN versus proposed single pass, $p = 1.000$; mean-probability CNN versus proposed, $p = 0.500$; proposed single pass versus proposed with test-time augmentation, $p = 1.000$. The smallest p-value across all pairs is 0.250. A paired group-level bootstrap ($B = 10,000$) gives an accuracy difference of +4.88 percentage points for mean-probability CNN over the proposed model (95% CI [0.00, +12.20]), with the lower bound at zero. The effect size is small, at most a few groups out of 41, and its direction favours the simpler CNN configurations.

Table 8. Paired group-level exact McNemar comparisons among the checkpoint-verified configurations of Table 6 ($N = 41$ groups)

Comparison	Both correct	X only	Y only	Both wrong	McNemar p (exact, two-sided)
CNN majority vote (X, 38/41) vs. proposed CRNN-GRU single pass (Y, 37/41)	36	2	1	2	1.000
CNN mean probability (X, 39/41) vs. proposed CRNN-GRU single pass (Y, 37/41)	37	2	0	2	0.500
CNN majority vote (X, 38/41) vs. proposed CRNN-GRU + TTA (Y, 36/41)	35	3	1	2	0.625
CNN mean probability (X, 39/41) vs. proposed CRNN-GRU + TTA (Y, 36/41)	36	3	0	2	0.250
Proposed CRNN-GRU single pass (X, 37/41) vs. proposed CRNN-GRU + TTA (Y, 36/41)	36	1	0	4	1.000

These results do not show that sequential modelling or inter-slice attention improve group-level classification beyond simple probability aggregation on this dataset. The macro-averaged metrics follow the same ordering as accuracy and do not favour the sequential model. No controlled ablation isolating the recurrent and attention components was performed, because no non-attention sequential model exists in the project. Such an ablation, matched for resolution, protocol and capacity, remains required future work, and none of the numbers in this study should be read as evidence for or against the attention component specifically.

Figure 8 shows the mean inter-slice attention weight at each of the ten positions, averaged over the 41 test groups. The weights stay close to uniform, between 0.06 and 0.14 against the uniform value of 0.10, with a modest maximum at position 3 and no position far from uniform. On average the attention mechanism departs little from equal weighting, which is consistent with its lack of measurable benefit over simple aggregation. Individual groups vary more, but the mean shows no sharp concentration. Attention weights indicate where the network places representational mass; they are not, on their own, an explanation of the decision, and linking them to lesion location would require an interpretability experiment such as occlusion or integrated gradients [29] [30], which was not performed here. Explainable deep-learning frameworks have been proposed for biomedical image analysis [31], which reinforces that attention or saliency maps should be validated rather than read directly as evidence of diagnostic importance.

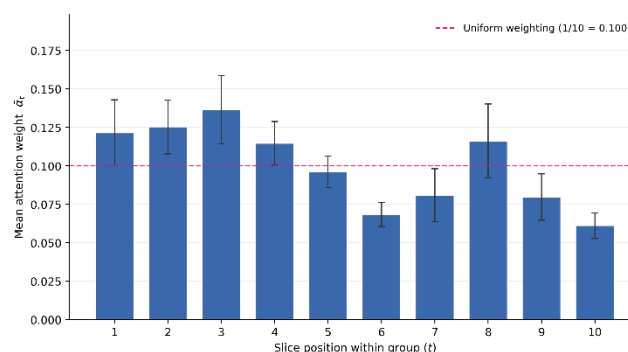


Figure 8. Mean inter-slice attention weights

3.2 Computational Cost

Both models were trained and benchmarked on Ubuntu with a single NVIDIA GeForce RTX 3090 GPU (24 GB) and CUDA using PyTorch. Trainable-parameter counts read from the checkpoints are 4,219,108 for the per-slice baseline and 4,527,012 for the CRNN-GRU, a ratio of 1.07. Training the CRNN-GRU took about 28 s per epoch, or about 23 minutes for the 48-epoch schedule, with a peak GPU memory of about 3.8 GB at a training batch size of four sequences. Computational cost is reported per decision. A single 128x128 slice through the baseline needs about 47.5 million multiply-accumulate operations, and a group-level decision by the CRNN-GRU with twelve test-time augmentation passes needs about 338.6 billion. Latency was measured in evaluation mode, with warm-up passes discarded, and averaged over the test cohort. A single 128x128 slice through the baseline took 1.23 +/- 0.10 ms, and one group-level decision by the CRNN-GRU with twelve passes took 3.59 +/- 0.99 s. No comparison against a real-time clinical threshold is made, since no universal threshold applies across brain-MRI workflows.

3.3 Practical Implications, Limitations and Future Works

The measured profiles do not support a deployment recommendation. On this dataset the per-slice CNN with mean-probability pooling was both cheaper, by roughly two orders of magnitude in operations per decision, and numerically more accurate, so no accuracy versus cost trade-off in favour of the sequential model was observed. A tiered workflow that routes uncertain cases to the sequential model is an appealing idea, but an untested hypothesis for future work rather than a validated pathway.

The proposed model's group-level accuracy of 90.24% is lower than the 95–99% slice-level figures reported by several studies in Table 1. Three factors separate these numbers and none alone explains the gap; the evaluation unit differs (41 groups against thousands of slices), the datasets and class definitions differ, and the splitting protocol differs. Slice-level evaluation can be affected by group-level leakage when slices from one examination fall on both sides of a split, an effect measured by Yagis et al. [20]. Group-disjoint evaluation is therefore more clinically aligned than slice-randomized evaluation, but it does not produce clinically valid metrics or measure real-world generalization, both of which would require external validation, calibration, reader comparison and prospective testing. Even under this protocol, the proposed model does not achieve the highest accuracy, since mean-probability CNN pooling reaches 95.12%.

Several limitations bound what these results support. (i) The analysis units are reconstructed surrogate slice groups, not verified patients. (ii) Slice ordering is assumed, not confirmed from metadata. (iii) The test cohort is small (41 groups; 6 no-tumor), so class-level metrics are uncertain. (iv) A content-level audit found 14 images duplicated across partitions, so the test set is not fully held out and all figures are upper bounds. (v) The Bhuvaji dataset [26] has known labelling noise in the meningioma and no-tumor folders, which is a plausible additional contributor to the meningioma errors and the optimistic bias. (vi) There is no external, multi-site validation. (vii) Only one T1-CE sequence type is used. (viii) The sequence length is fixed at $T = 10$, with no slice-count sensitivity test. (ix) Input resolution and optimizer settings differ between the models and are uncontrolled confounds. (x) A single seed was used, with no across-seed variance. (xi) No controlled ablation isolating the recurrent and attention components was performed. (xii) Test-time augmentation was applied only to the sequential model, and it reduced its accuracy. (xiii) No probability calibration was assessed. (xiv) No interpretability validation, no 3D or transformer baseline, and no prospective or reader study were performed. External validation is required before any generalizability or clinical-reliability claim [31].

4. CONCLUSION

This study evaluated whether reading an ordered sequence of MRI slices improves four-class brain tumor classification, under group-disjoint splits. The CRNN-GRU (convolutional encoder, bidirectional GRU and inter-slice attention, $T = 10$) classified 37 of 41 test groups correctly in a single pass (90.2%). The same per-slice CNN, aggregated by majority voting, reached 38 of 41 (92.7%), and by mean-probability pooling, 39 of 41 (95.1%). No difference among these configurations is statistically significant (exact McNemar, all $p \geq 0.25$), and the simplest configuration was the most accurate. The findings do not indicate that inter-slice context or attention add group-level accuracy beyond simple probability aggregation on this dataset, therefore the central result is negative. Meningioma remained the hardest class for every configuration. The lightweight per-slice CNN is faster and, on this dataset, also more accurate at the group-level, so no accuracy versus cost trade-off in favour of the sequential model was observed. A CRNN-GRU framework occupies a practical middle ground between per-slice 2D CNNs and 3D approaches. This study supports a methodological point, evaluating such models at the group-level under group-disjoint splits with a content-level leakage audit gives a more clinically aligned picture than slice-randomized evaluation. It does not establish clinical reliability. The results come from one retrospective public dataset without external validation, calibration, reader comparison or prospective testing [32], so the framework should be seen as a potential computer-aided analysis tool that needs further validation, not a reliable diagnostic aid. Future work will add multi-sequence fusion, resolution and protocol matched ablation, multiple seeds, calibration and multicentre external validation.

ACKNOWLEDGEMENTS

The author would like to thank Universiti Teknikal Malaysia Melaka, Malaysia and Universiti Utara Malaysia, Malaysia for their support.

DECLARATIONS

AI USAGE STATEMENT

The authors declare that AI-assisted tools, including OpenAI and QuillBot were used to support the drafting and editing of this manuscript, specifically in refining grammar, improving sentence clarity, and checking coherence. No part of the conceptual framework, data interpretation, or conclusions was generated by AI. The authors have thoroughly reviewed and verified all content and accept full responsibility for the work presented.

ETHICAL CONSIDERATIONS

This study was conducted in accordance with ethical standards for secondary research using anonymized human data. The authors did not collect original data, interact with human participants, or contact any clinical institutions. All experiments were performed using a single publicly available secondary dataset, the Brain Tumor Classification (MRI) Dataset published on Kaggle. The dataset consists of 2D grayscale images of JPEG format that lack DICOM headers or personally identifiable metadata. The dataset was accessed in a fully de-identified state, and no attempt was made to identify the patients. Consequently, institutional review board (IRB) review and informed consent were not required for this secondary analysis; responsibility for the ethical approval and consent of original data collection rests with the primary data providers.

AUTHOR CONTRIBUTION

Mohammed Z. M. Almadani carried out the experiments, implemented the models, performed data analysis, and wrote the original draft of the manuscript. Maslita Abd Aziz contributed to the conceptualization and methodology and reviewed and edited the manuscript. T Zalizam T Muda contributed to the validation of results, provided technical feedback, and participated in manuscript review and editing. All authors have read and agreed to the published version of the manuscript.

CONFLICTING INTERESTS

The authors declare no conflicts of interest.

REFERENCES

- [1] R. L. Siegel, T. B. Kratzer, A. N. Giaquinto, H. Sung, and A. Jemal, "Cancer statistics, 2025," *CA Cancer J. Clin.*, vol. 75, no. 1, pp. 10–45, Jan. 2025, <https://doi.org/10.3322/caac.21871>.
- [2] S. Grochans *et al.*, "Epidemiology of Glioblastoma Multiforme–Literature Review," *Cancers (Basel)*, vol. 14, no. 10, p. 2412, 2022, <https://doi.org/10.3390/cancers14102412>.
- [3] L. Hirschler *et al.*, "Advanced MR Techniques for Preoperative Glioma Characterization: Part 1," *Journal of Magnetic Resonance Imaging*, vol. 57, no. 6, pp. 1655–1675, 2023, <https://doi.org/10.1002/jmri.28662>.
- [4] R. Martínez-Del-Río-Ortega, J. Civit-Masot, F. Luna-Perejón, and M. Domínguez-Morales, "Brain Tumor Detection Using Magnetic Resonance Imaging and Convolutional Neural Networks," *Big Data and Cognitive Computing*, vol. 8, no. 9, p. 123, 2024, <https://doi.org/10.3390/bdcc8090123>.
- [5] M. Essien *et al.*, "Interrater Agreement of BT-RADS for Evaluation of Follow-up MRI in Patients with Treated Primary Brain Tumor," *American Journal of Neuroradiology*, vol. 45, no. 9, pp. 1308–1315, 2024, <https://doi.org/10.3174/ajnr.A8322>.
- [6] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, 2021, <https://doi.org/10.1186/s40537-021-00444-8>.
- [7] I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan, and P. Ilono, "Deep Convolutional Neural Networks in Medical Image Analysis: A Review," *Information (Switzerland)*, vol. 16, no. 3, 2025, <https://doi.org/10.3390/info16030195>.
- [8] F. J. Díaz-Pernas, M. Martínez-Zarzuela, M. Antón-Rodríguez, and D. González-Ortega, "A Deep Learning Approach for Brain Tumor Classification and Segmentation Using a Multiscale Convolutional Neural Network," *Healthcare*, vol. 9, no. 2, p. 153, 2021, <https://doi.org/10.3390/healthcare9020153>.
- [9] B. Babu Vimala, S. Srinivasan, S. K. Mathivanan, Mahalakshmi, P. Jayagopal, and G. T. Dalu, "Detection and classification of brain tumor using hybrid deep learning models," *Sci. Rep.*, vol. 13, no. 1, p. 23029, 2023, <https://doi.org/10.1038/s41598-023-50505-6>.

- [10] H. Yoon, D.-Y. Kang, and S. Kim, "Enhancement and evaluation for deep learning-based classification of volumetric neuroimaging with 3D-to-2D knowledge distillation," *Sci. Rep.*, vol. 14, no. 1, p. 29611, 2024, <https://doi.org/10.1038/s41598-024-80938-6>.
- [11] K. Jin, C. Sun, and L. Xia, "Development of a hybrid 2.5D deep learning model for glioma survival prediction using T1-weighted MRI from the CGGA database," *Eur. J. Radiol. Open*, vol. 15, p. 100697, 2025, <https://doi.org/10.1016/j.ejro.2025.100697>.
- [12] A. Saboor *et al.*, "DDFC: deep learning approach for deep feature extraction and classification of brain tumors using magnetic resonance imaging in E-healthcare system," *Sci. Rep.*, vol. 14, no. 1, p. 6425, 2024, <https://doi.org/10.1038/s41598-024-56983-6>.
- [13] J. Lee *et al.*, "Bidirectional gated recurrent unit network model can generate future visual field with variable number of input elements," *PLoS One*, vol. 19, no. 8, p. e0307498, 2024, <https://doi.org/10.1371/journal.pone.0307498>.
- [14] M. Li, Y. Jiang, Y. Zhang, and H. Zhu, "Medical image analysis using deep learning algorithms," *Front. Public Health*, vol. 11, 2023, <https://doi.org/10.3389/fpubh.2023.1273253>.
- [15] K. Lin, B. Jie, P. Dong, X. Ding, W. Bian, and M. Liu, "Convolutional Recurrent Neural Network for Dynamic Functional MRI Analysis and Brain Disease Identification," *Front. Neurosci.*, vol. 16, p. 933660, 2022, <https://doi.org/10.3389/fnins.2022.933660>.
- [16] R. P. K. Poudel, P. Lamata, and G. Montana, "Recurrent Fully Convolutional Neural Networks for Multi-slice MRI Cardiac Segmentation," in *Reconstruction, Segmentation, and Analysis of Medical Images*, vol. 10129, M. A. Zuluaga, K. Bhatia, B. Kainz, M. H. Moghari, and D. F. Pace, Eds., Cham: Springer International Publishing, 2017, pp. 83–94. https://doi.org/10.1007/978-3-319-52280-7_8.
- [17] I. Rethemiotaki, "Brain tumour detection from magnetic resonance imaging using convolutional neural networks," *Współczesna Onkologia*, vol. 27, no. 4, pp. 230–241, 2023, <https://doi.org/10.5114/wo.2023.135320>.
- [18] S. Deepak and P. M. Ameer, "Brain tumor classification using deep CNN features via transfer learning," *Comput. Biol. Med.*, vol. 111, p. 103345, 2019, <https://doi.org/10.1016/j.compbiomed.2019.103345>.
- [19] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, 2023, <https://doi.org/10.1016/j.patter.2023.100804>.
- [20] E. Yagis *et al.*, "Effect of data leakage in brain MRI classification using 2D convolutional neural networks," *Sci. Rep.*, vol. 11, no. 1, p. 22544, 2021, <https://doi.org/10.1038/s41598-021-01681-w>.
- [21] I. E. Tampu, A. Eklund, and N. Haj-Hosseini, "Inflation of test accuracy due to data leakage in deep learning-based classification of OCT images," *Sci. Data*, vol. 9, no. 1, p. 580, 2022, <https://doi.org/10.1038/s41597-022-01618-6>.
- [22] Z. N. K. Swati *et al.*, "Brain tumor classification for MR images using transfer learning and fine-tuning," *Computerized Medical Imaging and Graphics*, vol. 75, pp. 34–46, 2019, <https://doi.org/10.1016/j.compmedimag.2019.05.001>.
- [23] P. P. Malla, S. Sahu, and A. I. Alutaibi, "Classification of Tumor in Brain MR Images Using Deep Convolutional Neural Network and Global Average Pooling," *Processes*, vol. 11, no. 3, p. 679, 2023, <https://doi.org/10.3390/pr11030679>.
- [24] M. M. Ahmed *et al.*, "Brain tumor detection and classification in MRI using hybrid ViT and GRU model with explainable AI in Southern Bangladesh," *Sci. Rep.*, vol. 14, no. 1, p. 22797, 2024, <https://doi.org/10.1038/s41598-024-71893-3>.
- [25] A. Priya and V. Vasudevan, "Brain tumor classification and detection via hybrid alexnet-gru based on deep learning," *Biomed. Signal Process. Control*, vol. 89, p. 105716, 2024, <https://doi.org/10.1016/j.bspc.2023.105716>.
- [26] S. Bhuvaji, A. Kadam, P. Bhumkar, S. Dedge, and S. Kanchan, "Brain Tumor Classification (MRI)," 2020, *Kaggle*. <https://doi.org/10.34740/KAGGLE/DSV/1183165>.
- [27] L. Maier-Hein *et al.*, "Metrics reloaded: recommendations for image analysis validation," *Nat. Methods*, vol. 21, no. 2, pp. 195–212, 2024, <https://doi.org/10.1038/s41592-023-02151-z>.
- [28] G. Varoquaux and V. Cheplygina, "Machine learning for medical imaging: methodological failures and recommendations for the future," *NPJ Digit. Med.*, vol. 5, no. 1, p. 48, 2022, <https://doi.org/10.1038/s41746-022-00592-y>.
- [29] B. H. M. Van Der Velden, H. J. Kuijff, K. G. A. Gilhuijs, and M. A. Viergever, "Explainable artificial intelligence (XAI) in deep learning-based medical image analysis," *Med. Image Anal.*, vol. 79, p. 102470, 2022, <https://doi.org/10.1016/j.media.2022.102470>.

- [30] V. Malik and A. J. Singh, "Explainable Mitochondrial Image Segmentation and Morphological Quantification using Deep Learning Based Framework," *International Journal of Basic and Applied Science*, vol. 14, no. 4, pp. 152–161, 2026, <https://doi.org/10.35335/ijobas.v14i4.845>.
- [31] A. C. Yu, B. Mohajer, and J. Eng, "External Validation of Deep Learning Algorithms for Radiologic Diagnosis: A Systematic Review," *Radiol. Artif. Intell.*, vol. 4, no. 3, p. e210064, May 2022, <https://doi.org/10.1148/ryai.210064>.
- [32] A. S. Tejani *et al.*, "Checklist for Artificial Intelligence in Medical Imaging (CLAIM): 2024 Update," *Radiol. Artif. Intell.*, vol. 6, no. 4, p. e240300, Jul. 2024, <https://doi.org/10.1148/ryai.240300>.