ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Optimized MRI Image Enhancement and Brain Tumor Classification Using Xception

Gaddamedi Vamshi Krishna
Master's Student, Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Hyderabad University College of Engineering Jagtial, Telangana - 505501, India
vamshikrishna.gaddamidi@gmail.com

Saddala Praveen Kumar
Associate Professor, Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Hyderabad University College of Engineering Jagtial, Telangana - 505501, India
praveenkumar_00019@yahoo.com

Abstract*—Early and accurate analysis of brain tumors from magnetic resonance imaging requires both spatial localization and reliable category prediction. This work presents an integrated dual-module deep learning framework for MRI-based tumor segmentation, four-class classification, and visual explainability. The segmentation path compares G-Net Light, U-Net, and Residual U-Net, whereas the classification path evaluates MobileNetV2, ResNet50, ResNet18, DenseNet121, VGG19, ResNet152, EfficientNet, Xception, and ConvNeXt-Tiny. Classification training uses 299 × 299 images with horizontal flipping, 20-degree rotation, color jitter, and a WeightedRandomSampler; segmentation deployment uses 256 × 256 input tensors. The recorded results identify RU-Net as the strongest segmentation model with Dice ≈0.69 and IoU ≈0.60, while Xception provides the strongest classification performance with accuracy, precision, recall, and weighted F1-score of approximately 0.982. The selected models are deployed through a Flask application with SQLite authentication. For each valid MRI, the application returns the predicted class, confidence, segmentation mask, red overlay, Grad-CAM heatmap, and LIME explanation.

Keywords*—Brain tumor MRI, RU-Net, Xception, segmentation, classification, Grad-CAM, LIME, Flask, explainable AI.

I. INTRODUCTION

Brain tumors are abnormal cellular growths within or around the brain and may present substantial variation in location, morphology, internal texture, and boundary contrast. Magnetic Resonance Imaging provides strong soft-tissue contrast and is widely used for non-invasive visualization of brain structures, yet consistent interpretation can remain difficult when lesion boundaries are irregular or low contrast. Automated MRI analysis can support this process by combining tumor localization with diagnostic-category prediction.

The developed system treats segmentation and classification as complementary but distinct learning tasks. The segmentation branch predicts a binary tumor mask, while the classification branch predicts one of four MRI categories: glioma, meningioma, pituitary tumor, or no tumor. Three segmentation architectures—G-Net Light, U-Net, and RU-Net—are evaluated using Dice coefficient, Intersection over Union (IoU), and pixel accuracy. Nine transfer-learning classifiers are evaluated using accuracy, precision, recall, and weighted F1-score.

A further requirement is interpretability. A class label and confidence score alone do not show which image regions influenced the classifier. The framework therefore combines Xception prediction with Grad-CAM and LIME. Grad-CAM produces a class-relevant activation heatmap, whereas LIME identifies influential image superpixels through local perturbation. The final application integrates these explanations with the RU-Net mask and red tumor overlay inside an authenticated Flask interface.

The technical problem addressed by the framework is therefore not limited to classification accuracy. A classification-only system can identify a diagnostic category without indicating where the abnormal region is located, whereas a segmentation-only system can delineate a suspicious region without providing the corresponding MRI category. The developed workflow coordinates localization, four-class prediction, confidence reporting, and explainability so that these outputs are available for the same uploaded image.

The project scope is restricted to the implemented two-dimensional MRI workflow. Separate public datasets are used because the segmentation task requires image-mask pairs while the classification task requires class-organized images. The system is designed as a research and decision-support application rather than an independent clinical diagnostic system. This distinction is maintained throughout the model evaluation, software design, and interpretation of the final browser outputs.

II. RELATED WORK

Recent brain-tumor research increasingly combines segmentation, classification, and explainable AI rather than evaluating these tasks in isolation. Ullah et al. [1] presented a multimodal framework integrating segmentation, classification, and explanation. Mithun et al. [2] coordinated denoising, encoder-decoder segmentation, and deep classification, while Kharaji et al. [3] examined advanced nnU-Net segmentation. These studies reinforce the value of structured preprocessing and spatially aware segmentation for MRI analysis.

Classification research has evaluated a broad range of convolutional backbones. Srinivasan et al. [4] studied hybrid deep CNN multi-classification, Abou Ali et al. [5] analyzed augmentation and scenario-oriented MRI classification, and Ahsan et al. [6] linked deep-learning detection with segmentation. Ali et al. [7] compared multi-class segmentation and classification methods, while Chel et al. [8] emphasized LIME and Grad-CAM for interpretable MRI classification. Recent reviews and benchmarking studies [9], [10], [17] further motivate comparative evaluation rather than reliance on a single architecture.

The evaluated models are grounded in established architectural principles. U-Net [21] provides encoder-decoder segmentation with skip connections, residual learning [22]

supports deeper feature extraction, DenseNet [23] emphasizes feature reuse, Xception [24] and MobileNetV2 [25] employ efficient depthwise operations, EfficientNet [26] uses compound scaling, and ConvNeXt [27] modernizes convolutional design. Grad-CAM [28] provides gradient-based visual localization for classification decisions. The present framework uses these components only within the specific model comparison and deployment workflow implemented for the project.

The reviewed literature also clarifies the research gap addressed by the implemented system. Many studies focus on only one task or on a narrow model family. In contrast, the current workflow compares three segmentation models and nine classification backbones under common task-specific metrics before fixed models are selected for deployment. The objective is not to assume that a deeper or newer architecture is automatically superior, but to retain the architecture that records the stronger result under the implemented data and preprocessing conditions.

Explainability is treated as a separate interpretive layer rather than a substitute for segmentation. Grad-CAM and LIME explain the selected Xception decision, whereas RU-Net is evaluated against binary reference masks. Their outputs may visually overlap, but they represent different technical targets: segmentation predicts the tumor region, Grad-CAM visualizes class-relevant activations, and LIME identifies locally influential superpixels. This distinction is preserved in the result interface.

III. MATERIALS AND METHODS

A) Datasets and Preprocessing

Two task-specific MRI datasets are used because segmentation and classification require different targets. The segmentation dataset contains paired MRI images and binary ground-truth masks. The classification dataset is organized into glioma, meningioma, pituitary, and no-tumor folders. Image-mask correspondence is verified before segmentation evaluation, and class mapping is kept consistent across training, validation, and deployment.

Classification training resizes images to 299 × 299 pixels and applies random horizontal flipping, random rotation of 20 degrees, color jitter, and tensor conversion. Validation and frontend classification use deterministic resizing and tensor conversion. A WeightedRandomSampler is used during classification training to balance class exposure. Segmentation inference resizes MRI images to 256 × 256 pixels and preserves binary-mask integrity.

Random augmentation is applied only during classification training. Validation and deployed inference deliberately remove random transforms so that repeated analysis of the same MRI is not altered by unnecessary stochastic preprocessing. The segmentation and classification branches retain different input sizes because the selected model paths were developed for different targets. Maintaining these model-specific transformations during deployment preserves consistency between experimental evaluation and browser inference.

The dataset organization also supports traceability. Segmentation image-mask pairs are checked for correct correspondence before use, while the four classification folders preserve the class mapping required by Xception. A representative classification-dataset preview is shown in Fig. 1, illustrating the MRI image organization used by the implemented workflow.

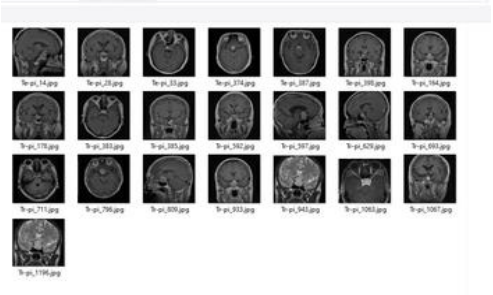


Fig. 1. Representative MRI dataset preview showing class-organized brain images used by the classification workflow.

Aspect	Segmentation Pipeline	Classification Pipeline
Dataset organization	Paired MRI images and binary masks	Class folders: glioma, meningioma, pituitary, no tumor
Primary target	Pixel-level tumor mask	Four-class MRI category
Input size	256 × 256 for deployed inference	299 × 299 for training, validation and inference
Training preparation	Image-mask verification and spatial resizing	Horizontal flip, rotation (20°), color jitter and tensor conversion
Evaluation metrics	Dice coefficient, IoU and pixel accuracy	Accuracy, precision, recall and weighted F1-score
Selected deployment model	RU-Net	Xception

Table I. Dataset, preprocessing, evaluation, and selected-model configuration.

B) Segmentation and Classification Models

G-Net Light is used as a compact segmentation baseline, U-Net provides an encoder-decoder structure with skip connections, and RU-Net combines the U-shaped architecture with residual learning. The classification comparison includes MobileNetV2, ResNet18, ResNet50, ResNet152, DenseNet121, VGG19, EfficientNet, Xception, and ConvNeXt-Tiny. Each classifier is adapted to the four MRI classes and evaluated under the same aggregate metric definitions.

The experimental comparison identifies RU-Net as the strongest segmentation model and Xception as the strongest classifier. Their saved weights are used during deployment. This model-selection process preserves task separation: RU-Net is responsible for mask and overlay generation, whereas Xception is responsible for the tumor category and confidence score.

G-Net Light provides the compact encoder-decoder baseline and is evaluated for efficient mask generation. U-Net contributes the established contracting and expanding paths with skip connections that preserve spatial localization. RU-Net incorporates residual learning into the U-shaped structure to improve gradient flow while retaining encoder-decoder skip connections. The recorded overlap results justify using RU-Net for the deployed segmentation path.

The classification backbones represent several convolutional design strategies. MobileNetV2 uses inverted residual blocks and depthwise separable convolution for efficient inference. ResNet18, ResNet50, and ResNet152 compare different residual depths. DenseNet121 emphasizes feature reuse through dense connectivity, VGG19 provides a conventional hierarchical convolutional baseline, EfficientNet applies compound scaling, Xception uses depthwise separable

convolution, and ConvNeXt-Tiny provides a modernized convolutional design. The final classifier layer of each network is adapted to the four MRI categories.

C) Integrated System Operation

The application workflow begins with signup or signin through SQLite-backed authentication. An authenticated user selects a valid MRI image and previews it before analysis. The backend validates the file and creates separate model-specific input tensors. RU-Net predicts a probability map that is thresholded to obtain a binary mask; the mask is resized to the original image dimensions and converted into a red overlay. Xception produces four-class logits, and softmax probabilities determine the reported class and confidence.

Explainability is generated after classification. Grad-CAM uses class-specific gradients from a convolutional feature layer to form a heatmap, and LIME perturbs image superpixels to identify locally influential regions. The Flask result template receives the original image, segmentation mask, overlay, predicted class, confidence, Grad-CAM heatmap, and LIME explanation and presents these artifacts within one analysis view.

The web application separates persistent user-account information from transient prediction artifacts. SQLite stores signup and signin data, while uploaded MRI files and generated masks, overlays, Grad-CAM images, and LIME explanations are handled by the current analysis workflow. The database is not used as a prediction-history repository in the implemented system. This separation keeps authentication independent from segmentation, classification, and visualization logic.

The software stack is organized around Python, PyTorch, Torchvision/timm, Flask, SQLite3, OpenCV, PIL, NumPy, Pandas, Scikit-learn, and Matplotlib. PyTorch executes the trained models, OpenCV and PIL handle image decoding, resizing, mask processing, overlays, and serialization, and Scikit-learn provides classification metrics. The modular organization permits the segmentation, classification, explainability, and interface components to be evaluated or updated independently while preserving their input and output responsibilities.

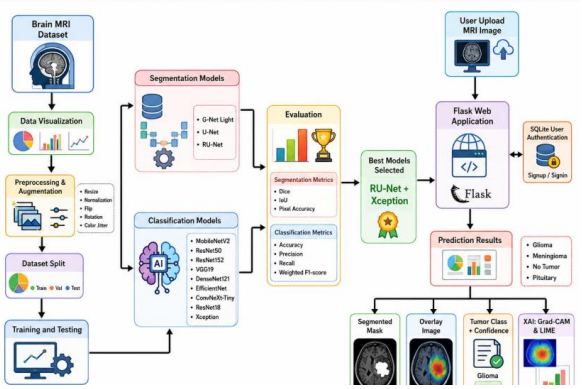


Fig. 2. Block diagram of the integrated brain tumor MRI analysis system.

D) Deployment Configuration and Explainability

The deployed workflow preserves separate processing branches for segmentation and classification. The Flask backend validates the uploaded MRI and creates a 256 × 256 tensor for RU-Net and a 299 × 299 tensor for Xception. RU-Net produces a probability map that is thresholded at 0.5 to form a binary tumor mask, which is resized to the original MRI dimensions before the red overlay is generated. Xception converts its output logits to four-class probabilities through softmax and returns the highest-probability category with its confidence score.

The application uses SQLite only for signup and signin authentication, while MRI images and generated result artifacts remain part of the transient inference workflow. Grad-CAM is produced from class-relevant gradients in the selected Xception model, and LIME perturbs image superpixels to identify locally influential regions. The final result page combines class prediction, confidence, RU-Net mask, tumor overlay, Grad-CAM heatmap, and LIME explanation while preserving the distinct role of each output.

E) Model Selection and Inference Consistency

Model selection is based on the comparative experimental records rather than architectural complexity alone. The classification study compares nine transfer-learning backbones under the same four-class task, while the segmentation study compares G-Net Light, U-Net, and RU-Net under the same overlap criteria. The saved RU-Net and Xception weights are then loaded during deployment so that the browser output is produced by the same selected model states identified during evaluation.

Consistency between validation and deployment is maintained by restricting random augmentation to classification training. Validation and browser inference use deterministic resize and tensor-conversion operations, and the segmentation and classification branches retain their model-specific input resolutions. This preserves the conditions under which the selected weights are evaluated and keeps localization, classification, and explanation as traceable components of the integrated workflow.

IV. EXPERIMENTAL RESULTS

The result analysis uses task-specific metrics because segmentation and classification answer different questions. Accuracy, precision, recall, and weighted F1-score summarize four-class classification behavior. Dice coefficient, IoU, and pixel accuracy evaluate the spatial agreement between predicted tumor masks and reference masks. The graphical comparisons and application screens provide an additional view of model ranking and deployed behavior.

A) Classification Performance

The nine classifiers produce closely grouped performance values, but Xception records the strongest result. Table II reproduces the recorded accuracy, precision, recall, and weighted F1-score values for all nine classification models and supports selection of Xception for deployment.

Model	Acc.	Prec.	Rec.	F1
MobileNetV2	0.967	0.968	0.967	0.967
ResNet50	0.970	0.970	0.970	0.970
ResNet152	0.975	0.975	0.975	0.975
VGG19	0.973	0.973	0.973	0.973
DenseNet121	0.978	0.978	0.978	0.978
EfficientNet	0.973	0.973	0.973	0.973
ConvNeXt-Tiny	0.967	0.968	0.967	0.967
ResNet18	0.974	0.974	0.974	0.974
Xception	0.982	0.982	0.982	0.982

Table II. Comparative performance of the nine classification models.

Accuracy reports the proportion of correctly classified MRI images. Precision measures the reliability of class assignments, recall measures the recovery of relevant class instances, and weighted F1-score combines precision and recall while accounting for class support. Segmentation is evaluated separately using Dice coefficient, IoU, and pixel accuracy.

Performance is evaluated with the following editable equations. TP, TN, FP and FN denote the corresponding classification or pixel-level segmentation counts.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \times 100$$

$$Pixel\ Accuracy = TP + TNTP + TN + FP + FN$$

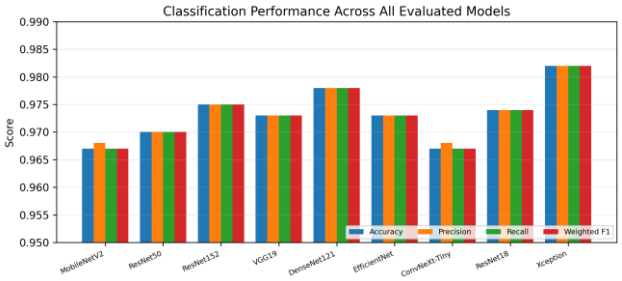


Fig. 3. Combined accuracy, precision, recall, and weighted F1-score comparison for all nine classifiers.

The numerical separation among the classifiers is small, but the ranking is consistent enough to support model selection. Relative to DenseNet121, Xception improves the common metric value by 0.004, and relative to the 0.967 accuracy reported for MobileNetV2 and ConvNeXt-Tiny the improvement is 0.015. The close agreement among accuracy, precision, recall, and weighted F1-score also indicates that the aggregate comparison is not dominated by one metric alone.

MobileNetV2 records 0.967 accuracy, 0.968 precision, 0.967 recall, and 0.967 weighted F1-score. ConvNeXt-Tiny records the same accuracy, recall, and F1-score with precision of 0.968. ResNet50 reaches 0.970 across the four metrics. VGG19 and EfficientNet each record 0.973, ResNet18 reaches 0.974, and ResNet152 reaches 0.975. DenseNet121 records 0.978 and provides the second-highest common performance before Xception at 0.982.

These values show that architectural complexity alone does not determine the deployment model. ResNet152 is deeper than ResNet18 and ResNet50, but Xception still records the strongest common classification metrics. DenseNet121 is also competitive despite using a different connectivity strategy. The resulting ranking therefore supports empirical selection from the implemented comparison rather than selection based only on network depth or architectural popularity.

B) Segmentation Performance

Segmentation produces a clearer separation among the evaluated architectures. RU-Net records a Dice coefficient of approximately 0.69 and an IoU of approximately 0.60. G-Net Light records approximately 0.65 Dice and 0.54 IoU, while the implemented U-Net result is lower in the recorded comparison. RU-Net therefore ranks first on both principal overlap measures and is selected for the deployed mask and overlay path.

Dice coefficient measures the shared region relative to the combined predicted and reference mask sizes, whereas IoU normalizes the shared region by the complete union. Pixel accuracy provides an additional image-wide correctness measure. Because RU-Net leads both Dice and IoU in the recorded comparison, the selected mask generator is supported by two related but distinct overlap criteria.



Fig. 4. Segmentation performance comparison using Dice coefficient and IoU.

Dice coefficient and IoU evaluate related aspects of spatial agreement between the predicted mask and the reference mask. RU-Net is stronger on both measures, which provides direct support for its deployment selection. The segmentation output remains distinct from the Grad-CAM and LIME explanations: the mask is a pixel-level tumor-region prediction, whereas the explanation images visualize evidence associated with the classification decision.

The segmentation comparison exhibits a wider separation than the classification comparison. RU-Net records approximately 0.69 Dice and 0.60 IoU, compared with approximately 0.65 Dice and 0.54 IoU for G-Net Light. The U-Net result is lower than both in the recorded graph. The same ordering is therefore preserved across the two overlap measures, which strengthens the justification for using RU-Net to generate the deployed mask and red overlay.

After segmentation inference, the RU-Net probability output is thresholded at 0.5 to obtain a binary mask. The mask is resized to the dimensions of the uploaded MRI and can be processed with OpenCV to create the visual tumor overlay. This conversion step transforms the model output into a directly reviewable spatial artifact without changing the underlying segmentation decision.

C) Integrated Web-Application Results

The implemented web application connects model evaluation with operational inference. The user uploads an MRI image through the diagnostic workbench, verifies the preview, and submits the image for analysis. The backend applies the fixed RU-Net and Xception preprocessing paths before segmentation, classification, and explainability generation. Figure 5 shows the upload and preview stage, followed by the integrated result views in Figs. 6 and 7.

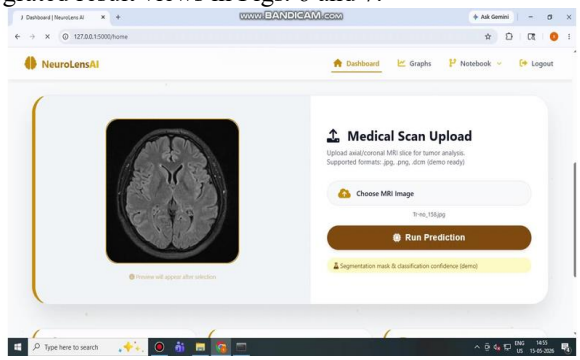


Fig. 5. Diagnostic workbench with MRI upload, preview, and prediction control.

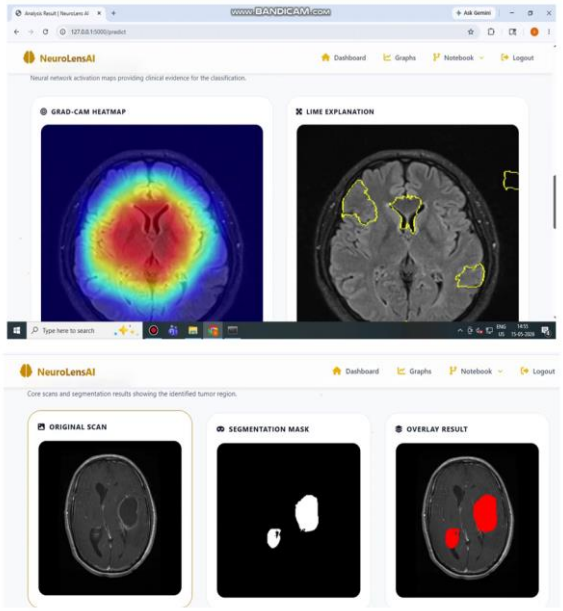


Fig. 6. Integrated application outputs showing Grad-CAM, LIME, segmentation mask, and tumor-overlay visualization.

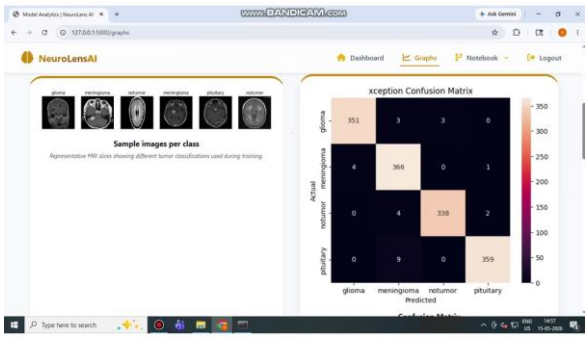


Fig. 7. Model analytics dashboard with sample MRI images and Xception confusion-matrix visualization.

The displayed outputs preserve the responsibility of each processing component. Xception provides the four-class probability distribution and selected class. RU-Net provides the spatial mask and boundary visualization. Grad-CAM and LIME add complementary explanation views without replacing the segmentation target. This separation is important because each output is interpreted according to a different technical objective.

The deployed workflow also preserves fixed model state. The browser request does not retrain RU-Net or Xception; instead, it loads the saved weights associated with the selected experimental models. Classification uses the same deterministic resize and tensor-conversion convention used during validation, and segmentation uses the fixed 256 × 256 transformation expected by RU-Net. This traceability links the reported metrics to the behavior observed in the application.

The frontend supports a complete sequence from account access to result inspection. After authentication, the user selects an MRI image, confirms the preview, submits the image, and receives the integrated result page. The application can display tumor-boundary visualization, classification output, segmentation output, Grad-CAM and LIME explanations, and model-analytics graphs. These views provide a direct connection between the experimental comparison and the deployed inference workflow.

The classification output is generated from Xception logits converted to probabilities with softmax. The highest-probability category is reported as glioma, meningioma, no tumor, or pituitary together with its confidence score. The segmentation probability map is separately thresholded, resized, and overlaid on the original image. By keeping these paths independent until result rendering, the application avoids mixing image-level classification scores with pixel-level mask values.

Generated artifacts are written as separate output images so that the result template can load the segmentation mask, overlay, Grad-CAM heatmap, and LIME explanation independently. This also supports debugging and software maintenance because each visual artifact can be traced to a specific processing function. The model-analytics dashboard connects these per-image outputs with broader classification and segmentation comparisons recorded during experimentation.

D) Experimental-to-Deployment Traceability

The implemented workflow preserves a direct connection between the model-development stage and the browser-based analysis stage. Classification experiments record the common metric table used to compare all nine candidate backbones, while segmentation experiments record the overlap comparison used to select the mask generator. The web application does not initialize or retrain these networks during a request. Instead, it loads the saved RU-Net and Xception weights associated with the selected experimental results, ensuring that the browser output is produced by the same model state retained after evaluation.

Preprocessing consistency provides a second layer of traceability. Classification training uses random augmentation to expose the network to controlled appearance variations, but validation and deployment use deterministic resizing and tensor conversion. Segmentation deployment likewise uses the fixed 256 × 256 input transformation expected by RU-Net. Applying random augmentation during prediction would introduce unnecessary variation into repeated analysis of the same MRI. The deployed preprocessing path therefore remains fixed so that the prediction conditions are aligned with the intended evaluation configuration.

The model comparison further demonstrates why deployment selection is based on measured results rather than architectural complexity. ResNet152 is deeper than the other residual alternatives, yet Xception records the strongest aggregate classification values. DenseNet121 also performs strongly despite a different connectivity strategy. The segmentation comparison reaches the same evidence-based conclusion: RU-Net leads both Dice and IoU, while G-Net Light and U-Net occupy lower positions in the recorded overlap comparison. The selected models are therefore chosen from the common evaluation rather than from architecture depth, model age, or design family alone.

The generated explanation artifacts must also be interpreted according to their technical roles. RU-Net predicts the tumor region and is evaluated with segmentation metrics against reference masks. Grad-CAM and LIME explain the Xception classification result and are not evaluated as tumor masks. Grad-CAM visualizes class-relevant activation, while LIME estimates local superpixel influence. Displaying these outputs together is useful because they offer complementary evidence, but the interface preserves the distinction between spatial segmentation, class prediction, and classification explanation.

System validation completes the traceability chain by confirming that authentication, MRI upload, preprocessing, inference, explanation generation, and result rendering operate as an integrated sequence. The user should not need to execute notebook cells, move intermediate files manually, or invoke separate programs to obtain the combined analysis. The recorded system tests confirm that the saved models, preprocessing functions, and frontend routes work together to

produce the expected result artifacts through the implemented Flask application.

E) Integrated Result Interpretation

The classification results are closely grouped, which makes the common evaluation protocol important for model selection. MobileNetV2 and ConvNeXt-Tiny define the lower end of the recorded aggregate scores at approximately 0.967 accuracy, while ResNet50, VGG19, EfficientNet, ResNet18, and ResNet152 occupy intermediate positions. DenseNet121 records 0.978 across the four reported measures, and Xception reaches approximately 0.982 for accuracy, precision, recall, and weighted F1-score. The narrow numerical range indicates that deployment selection is supported by consistent ranking across several metrics rather than by a large separation in a single score.

The segmentation comparison is more clearly separated. RU-Net records approximately 0.69 Dice and 0.60 IoU, G-Net Light records approximately 0.65 Dice and 0.54 IoU, and the implemented U-Net experiment remains lower in the comparison. Because RU-Net leads on both overlap measures, its selection is supported by two related but distinct criteria. The predicted probability map is thresholded, resized to the original MRI dimensions, and converted into the binary mask and red overlay displayed in the application.

The integrated result page should therefore be interpreted as a combination of task-specific evidence. The Xception label and confidence describe the image-level diagnostic category, whereas the RU-Net mask describes the spatial tumor region. Grad-CAM and LIME are explanation artifacts attached to the classification decision and are not substitutes for the segmentation mask. Presenting these outputs together is useful because they provide complementary evidence, but their technical roles remain distinct throughout evaluation and deployment.

The application also preserves inference reproducibility. Random horizontal flipping, rotation, and color jitter are limited to classification training, while validation and browser inference use fixed resize and tensor conversion. Similarly, the segmentation path retains the 256 × 256 transformation expected by RU-Net and the classification path retains the 299 × 299 transformation expected by Xception. This deterministic deployment sequence ensures that repeated analysis of the same MRI is not changed by unnecessary stochastic preprocessing.

The final browser output provides a direct evidence chain from experiment to result visualization. The comparison tables and graphs identify the selected models, saved weights preserve the evaluated configurations, the Flask backend recreates the required preprocessing, and the result template displays the class, confidence, mask, overlay, Grad-CAM, and LIME artifacts generated for the current MRI. This traceable sequence is central to the technical validity of the implemented workflow because each displayed item can be associated with a defined model and processing stage.

V. CONCLUSION

The developed framework combines MRI tumor segmentation, four-class classification, explainable AI, and authenticated browser deployment in one structured system. Separate task-specific datasets and preprocessing paths are retained because segmentation predicts a spatial mask whereas classification predicts a diagnostic category. Among the evaluated models, RU-Net records the strongest segmentation overlap with Dice ≈0.69 and IoU ≈0.60, and Xception records the strongest classification performance with accuracy, precision, recall, and weighted F1-score of approximately 0.982.

The Flask application uses the selected RU-Net and Xception weights to return the tumor mask, red overlay, predicted class, confidence, Grad-CAM heatmap, and LIME explanation. SQLite supports signup and signin authentication. Future directions identified for the framework include three-dimensional multi-sequence MRI analysis, multi-institutional evaluation, comparison with advanced architectures, confidence calibration and uncertainty estimation, expanded explainability, federated learning, and clinical-system integration after appropriate privacy, regulatory, and clinical evaluation.

The completed evaluation also establishes the boundary between the two task-specific outputs. Segmentation performance is judged by spatial overlap with reference masks, while classification performance is judged by class-level prediction metrics. Their roles are complementary but not interchangeable. The same distinction applies to Grad-CAM and LIME: both explanation methods visualize evidence associated with classification and are not evaluated as segmentation masks.

Future work described for the system includes extending the two-dimensional workflow to three-dimensional multi-sequence MRI, repeating the evaluation on multi-institutional data, comparing advanced architectures under the same datasets and metrics, and adding confidence calibration or uncertainty estimation. Explainability can be expanded with additional attribution approaches, while federated learning may support multi-hospital model development without centralizing sensitive images. DICOM viewing, hospital-information-system connectivity, and secure cloud deployment are further directions after appropriate privacy, clinical, and regulatory evaluation.

External validation is particularly important because the recorded results depend on the segmentation and classification datasets, class mapping, preprocessing settings, and model-selection procedure used in the current experiment. A future study can repeat the same comparison on independently collected MRI data and record the same accuracy, precision, recall, weighted F1-score, Dice, and IoU measures to determine whether the relative model ordering remains stable under different acquisition conditions.

Software reproducibility can also be strengthened by packaging the Python environment, preprocessing settings, saved model weights, and class mappings as a controlled deployment bundle. The current implementation already separates notebook-based model development from Flask inference. Formalizing the required dependencies would make it easier to reproduce the analysis on another supported machine while maintaining the same deterministic input transformations and selected model state.

REFERENCES

[1] M. S. Ullah et al., “Multimodal brain tumor segmentation and classification based on a deep learning and explainable AI framework,” Computers in Biology and Medicine, 2024. https://www.sciencedirect.com/science/article/abs/pii/S00104825 2401268X

[2] M. S. Mithun et al., “Detection and classification on MRI images of brain tumor using deep learning techniques,” Alexandria Engineering Journal, 2024. https://www.sciencedirect.com/science/article/pii/S168785072400 2978

[3] M. Kharaji et al., “Brain tumor segmentation with advanced nnU-Net,” 2024. https://www.sciencedirect.com/science/article/pii/S277252862400 0013

[4] S. Srinivasan et al., "A hybrid deep CNN model for brain tumor image multi-classification," BMC Medical Imaging, 2024. https://link.springer.com/article/10.1186/s12880-024-01195-7
[5] M. Abou Ali et al., "Enhancing MRI brain tumor classification: A comprehensive approach integrating real-life scenario simulation and augmentation techniques," Physica Medica, 2024. https://pubmed.ncbi.nlm.nih.gov/39488993/
[6] R. Ahsan et al., "Brain tumor detection and segmentation using deep learning," 2025. https://pubmed.ncbi.nlm.nih.gov/39231857/
[7] A. Ali et al., "Multi-class brain tumor MRI segmentation and classification using deep learning and machine learning," 2025. https://link.springer.com/article/10.1186/s40644-025-00953-2
[8] Y. H. Chel et al., "Brain Tumor Classification in MRI: Insights From LIME and Grad-CAM Explainable AI Techniques," IEEE Access, 2025. https://ieeexplore.ieee.org/document/11142756/
[9] F. J. Dorfner et al., "A review of deep learning for brain tumor analysis in MRI," 2025. https://pubmed.ncbi.nlm.nih.gov/39753730/
[10] J. Huang et al., "Brain tumor segmentation using deep learning," 2025. https://pubmed.ncbi.nlm.nih.gov/40697313/
[11] L. Yang et al., "Deep learning-based automatic segmentation of meningioma on MRI," 2024. https://pubmed.ncbi.nlm.nih.gov/38443817/
[12] M. Manthe et al., "Federated brain tumor segmentation: An extensive benchmark," Medical Image Analysis, 2024. https://www.sciencedirect.com/science/article/abs/pii/S1361841524001956
[13] B. V. B. et al., "Efficient brain tumor grade classification using ensemble deep learning," BMC Medical Imaging, 2024. https://link.springer.com/article/10.1186/s12880-024-01476-1
[14] K. Kumar et al., "Machine learning for brain tumor classification," 2024. https://link.springer.com/article/10.1007/s44163-024-00214-4
[15] M. V. Narayana et al., "A framework for identification of brain tumors from MR images," 2024. https://link.springer.com/article/10.1007/s12553-024-00844-9
[16] S. K. Mathivanan et al., "A secure hybrid deep learning framework for brain tumor detection," 2025. https://link.springer.com/article/10.1186/s40537-025-01117-6
[17] T. A. Fahim et al., "Brain tumor detection, classification and segmentation by deep learning: A review," 2025. https://www.sciencedirect.com/science/article/pii/S2590005625001985
[18] A. Mehrabi et al., "3D brain tumor segmentation in MRI images using deep learning," 2025. https://www.sciencedirect.com/science/article/pii/S2666521225001073
[19] D. J. Ghadimi et al., "Deep Learning-Based Techniques in Glioma Brain Tumor Segmentation: A Review," 2025. https://pubmed.ncbi.nlm.nih.gov/39074952/
[20] B. Sandhiya et al., "Brain tumour segmentation and classification with intelligent computational methods," 2024. https://www.sciencedirect.com/science/article/abs/pii/S1746809424000636
[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," MICCAI, 2015.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," CVPR, 2016.
[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Weinberger, "Densely Connected Convolutional Networks," CVPR, 2017.
[24] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," CVPR, 2017.
[25] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," CVPR, 2018.
[26] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," ICML, 2019.
[27] Z. Liu et al., "A ConvNet for the 2020s," CVPR, 2022.
[28] R. R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," ICCV, 2017.
[29] N. Tomar, "Brain Tumor Segmentation Dataset," Kaggle. https://www.kaggle.com/datasets/nikhilroxtomar/brain-tumor-segmentation/data
[30] M. Nickparvar, "Brain Tumor MRI Dataset," Kaggle. https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset/data