International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 3, September 2026, pp. 1352~1363
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i3.pp1352-1363

Deep reinforcement learning inspired optimization framework using Optuna for brain tumor detection

Aashutosh Kharb, Prachi Chaudhary
Department of Electronics and Communication Engineering, DCRUST, Murthal, India

Article Info

Article history:

Received Sep 10, 2025
Revised May 24, 2026
Accepted Jul 5, 2026

Keywords:

Brain tumor detection
Deep reinforcement learning
EfficientNet
Hyperparameter tuning
Reinforcement learning

ABSTRACT

Accurate brain tumor detection is essential for effective clinical diagnosis; however, the performance of deep learning models is highly sensitive to manually selected architectures and hyperparameters. To address this challenge, this paper presents a reinforcement learning–inspired automated optimization framework for brain tumor detection that eliminates manual trial-and-error tuning of hyperparameters. The proposed approach integrates EfficientNetB0 as a fixed feature extractor (base model) with an Optuna-based reinforcement learning strategy to jointly optimize the classifier architecture and key training hyperparameters, including learning rate, batch size, dropout rate, and network depth. Unlike existing studies that rely on static or heuristically tuned models, the proposed framework dynamically adapts model configurations based on validation feedback. Experiments conducted on the BraTS 2020 MRI dataset demonstrate that the optimized model achieves an accuracy of 92%, an F1-score of 92%, and a ROC–AUC of 0.96. Additional evaluations on imbalanced and cross-dataset settings show stable minority-class performance and good generalization. The results confirm that the proposed automated optimization framework offers a robust, scalable, and clinically relevant solution for brain tumor detection, representing a significant advancement over manually tuned deep learning approaches.

Corresponding Author:

Aashutosh Kharb
Department of Electronics and Communication Engineering, DCRUST
Murthal, Haryana, India
Email: aashutoshkharb@gmail.com

1. INTRODUCTION

Brain tumors remain among the most devastating neurological disorders, with timely and accurate detection being critical to patient survival. However, standard magnetic resonance imaging (MRI) diagnostics depend heavily on radiologist expertise and may be delayed due to increasing caseloads and subjective interpretation. Automated, reliable systems are therefore urgently needed to support clinicians by reducing diagnostic variability and enhancing early intervention.

In recent years, deep learning has revolutionized the field of medical image analysis, offering remarkable improvements in accuracy, speed, and consistency over traditional methods. Convolutional neural networks (CNNs), in particular, have demonstrated strong performance in detecting and classifying brain tumors from MRI scans, effectively learning complex spatial and structural patterns without the need for handcrafted features. State-of-the-art architectures such as EfficientNet [1] have further optimized performance by balancing model depth, width, and resolution. Studies such as the BraTS benchmark challenge [2] have highlighted the effectiveness of CNN-based models in segmenting and classifying brain tumor regions, showcasing the practical value of deep learning in clinical contexts. Hybrid deep CNN models have demonstrated multi-class classification with remarkable accuracies—up to 99.53% for tumor detection and 93.81% for tumor-type classification across glioma, meningioma, pituitary, and metastatic tumors using grid-search hyperparameter optimization [3]. Additionally, lightweight U-Net variants like LeU-Net and transfer-learning-enhanced architectures such as SqueezeNet have achieved segmentation accuracies between 94% and 98%, offering faster and resource-efficient alternatives suitable for clinical settings [4]–[6].

Shawon *et al.* [7] tackled the common issue of class imbalance in MRI datasets using cost-sensitive tuning alongside popular CNN architectures like InceptionV3, ResNet50, and EfficientNetB0. Their cost-sensitive InceptionV3 attained an impressive 99.33% accuracy on balanced data, while explainability tools revealed deeper model insights. Even on imbalanced data, they managed to increase accuracy by approximately 4%, highlighting the importance of fairness in detection systems. Díaz-Pernas *et al.* [8] designed a multiscale CNN mirroring human visual processing by analyzing MRI inputs at three resolutions. Operating on a dataset of 3,064 images, their end-to-end model achieved 97.3% accuracy across meningioma, glioma, and pituitary tumor types without preprocessing—demonstrating the value of scale-aware feature extraction. Karagoz *et al.* [9] introduced ResViT, a hybrid combining convolutional and transformer blocks. They pretrained the model through self-supervised learning (SSL), synthesizing MRI scans to learn general representations before fine-tuning on classification. The model achieved approximately 98.5% accuracy, showing that limited labeled MRI datasets can still be leveraged effectively through SSL. Aamir *et al.* [10] developed a CNN optimized through systematic hyperparameter tuning—covering learning rates, batch sizes, filter dimensions, and depth—to achieve approximately 97% on accuracy, precision, recall, and F1-score across three Kaggle MRI datasets. This highlights how proper calibration can significantly elevate model robustness. Mughal *et al.* [11] explored the addition of a denoising step—using anisotropic diffusion—before feeding scans into CNNs like EfficientNet, ResNet152V2, VGG, and vision transformer (ViT). EfficientNet led the pack with 98 % accuracy, underscoring the impact of denoising pipelines on final model performance. Singh and Jain [12] crafted a lightweight binary CNN for tumor detection on a Kaggle MRI dataset. Despite its simplicity, it achieved 99.31% accuracy, proving that smaller models can still offer ultra-high performance and may be ideal for low-resource clinical environments.

Patel and Gandhi [13] combined CNN feature extractors with transformer encoders, pretrained via self-supervised masked autoencoders (MAEs) to tackle the four-way classification task. Their hybrid approach achieved 94% accuracy and F1, outperforming non-SSL and standalone architectures by approximately 6%–7% [13]. Meshram *et al.* [14] applied the swin transformer to not only classify but also localize brain tumors, delivering segmentation masks and tumor dimension estimates via PDF. This aligns well with radiologists' workflows, improving clinical utility through combined detection and spatial awareness. Several recent studies [3], [5], [7], [9], [12] have shown that the refinement in the CNN models may yields to an accuracy ranging from 95% to 99% approximately. In 2025, further refinements continued with a CNN model tailored for binary tumor detection on Kaggle MRI datasets, achieving 99.31% accuracy while maintaining architectural simplicity [12].

Simultaneously, studies evaluating multiple MRI modalities combined with transfer learning reported classification accuracies around 98.6% across T1, T2, and FLAIR sequences, underlining the clinical relevance of selecting the most informative imaging combinations [15]. Despite these advances, designing optimal deep learning models remains a challenge due to the vast number of tunable hyperparameters and architectural choices. Recent approaches have explored reinforcement learning to automate this process, allowing models to adaptively learn the best configuration for classification tasks [16]–[18]. Motivated by these advances, this paper proposes a novel approach integrating EfficientNet-based CNNs with reinforcement learning inspired fine-tuning. The study utilizes the BraTS 2020 dataset to validate the proposed method, contributing to the growing body of work focused on developing intelligent, data-driven tools for improving early tumor detection and supporting clinical decision-making.

The major contributions of this work are as follows: i) proposed an adaptive deep learning framework for brain tumor detection using MRI images, which combines the feature extraction capability of EfficientNetB0 with the adaptive tuning power of reinforcement learning; ii) reinforcement learning inspired Optuna framework is used to determine the most effective classifier architecture and to automatically fine-tune critical hyperparameters such as learning rate, batch size, number of epochs, and dropout rate; iii) this dual-level optimization ensures that the final model is both structurally optimal and well-calibrated for training dynamics.

2. RESEARCH METHOD

This section describes the proposed research methodology for automated brain tumor detection from MRI images. It outlines the overall framework, including data preprocessing, EfficientNet-based feature extraction, and the reinforcement learning–inspired Optuna optimization strategy used to jointly optimize the classifier architecture and training hyperparameters as shown in Figure 1.

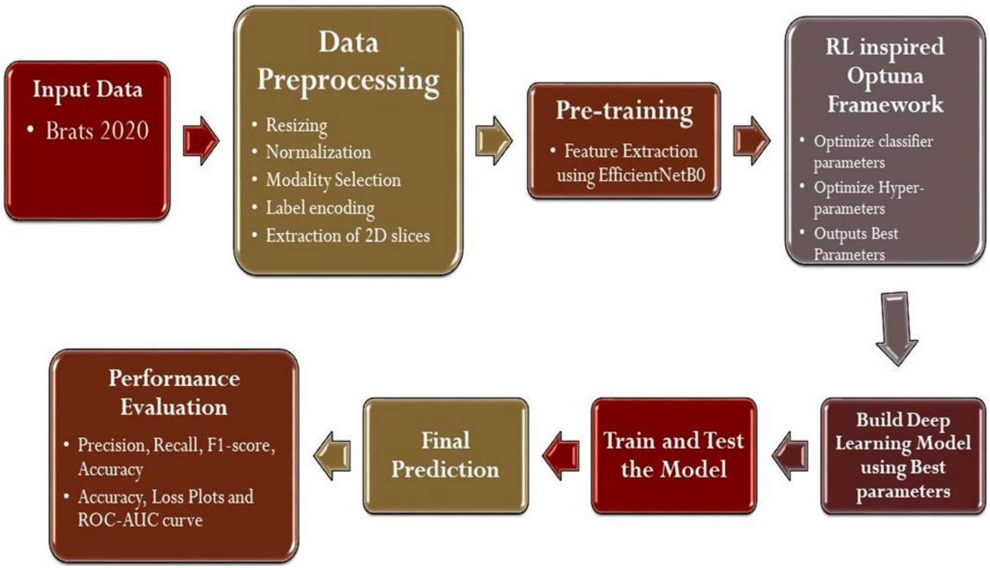

Figure 1. Architecture of the proposed EfficientNet-based reinforcement learning-inspired optimization framework for brain tumor detection

2.1. Dataset description

In this work, BraTS 2020 dataset has been used. The BraTS 2020 dataset is a widely used medical imaging dataset for brain tumor detection and segmentation. It includes MRI scans in four modalities—T1, T1Gd, T2, and FLAIR—of patients with high- and low-grade gliomas, along with expert-annotated tumor regions. Its realistic clinical variations and rich annotations make it a reliable benchmark for training and evaluating deep learning models in brain tumor analysis.

2.2. Data preprocessing

The data preprocessing has been done to standardize the input, make it less variable, and improve the quality of the input data and improve model performance before deep learning on the BraTS 2020 MRI scans. The following steps are performed: Image resizing: all of the MRI slices are resized to 224×224 pixels, so as to make sure that all the samples are the same size while keeping the important spatial features that are needed to identify tumors. Normalization: here, min-max scaling is used to bring pixel intensities down to a range of [0, 1]. This makes it less likely that scans will have different levels of brightness and contrast. It also makes sure that the input distribution is always the same, which speeds up training. Modality selection and stacking: based on the experimental setup, chosen combinations of the four available MRI modalities (T1, T1Gd, T2, and FLAIR) are stacked as input channels. This helps the model learn from different contrasts that show different things about the tumor. Slice extraction and filtering: in this work 2D slices have been extracted from the 3D volumes that have visible tumor areas. This step removes a lot of noise in the dataset and helps the model focus on important learning goals. Label encoding: ground-truth labels are used to categorize the different types of tumors into “tumor” and “no tumor”.

2.3. Feature extraction using EfficientNetB0

The EfficientNetB0 uses compound scaling to find the right balance between depth, width, and resolution, which makes it fast and efficient therefore it has been used in this work as the backbone model to extract features from the 2D slices. The first few layers of EfficientNetB0 has been frozen and we take feature embeddings from the layer just before the last one. The reinforcement learning agent then dynamically optimizes these embeddings by sending them to a customizable classifier head.

2.4. Reinforcement learning inspired dual optimization using Optuna

Reinforcement learning plays a central role in our proposed framework by automating two major aspects of model development: 1. Designing the optimal classifier structure (i.e., the configuration of fully connected layers), and 2. Selecting the best training hyperparameters (e.g., learning rate, batch size, number of epochs, optimizer type). This intelligent automation allows the system to explore and identify high-performing configurations without manual intervention—especially beneficial in deep learning, where manual tuning is time-consuming and often suboptimal.

2.4.1. Reinforcement learning setup overview

In this study, we utilize Optuna as a reinforcement learning-inspired framework to automate the dual task of (1) classifier structure design and (2) training hyperparameter tuning. This not only replaces manual experimentation but also significantly improves model performance and efficiency through data-driven search. Optuna formulates optimization as a black-box function minimization problem. In this case, the function is to maximize the model validation accuracy, evaluated over a set of hyperparameters and architecture decisions. Each trial in Optuna represents a full episode of model building and evaluation:

- Trial (episode): sample a set of values that is number and size of dense layers, activation function, dropout rate, optimizer, learning rate, batch size, number of training epochs
- Environment: a deep learning model is constructed using the selected configuration, trained on a small subset of the dataset, and evaluated on a validation set.
- Reward: a scalar value is calculated based on model performance that is validation accuracy in this work.
  - Sampler update: use that reward to guide the next trial via Bayesian optimization that is tree-structured parzen estimator (TPE) in this work, improving its chances of selecting better configurations in future episodes.

Optuna builds a probability model of promising regions of the search space, similar to a policy update in reinforcement learning, and prioritizes exploring them while still sampling from unexplored areas (exploration vs. exploitation).

2.4.2. Integration reinforcement learning in the pipeline

a. Step 1: design of objective function

The objective function is key component of the reinforcement learning- inspired optimization framework, as it guides the search toward optimal classifier architectures and hyperparameter configurations. In this work, the objective function performs the following tasks:

- Builds the model using sampled configuration.
- Trains on a subset of data.
- Evaluates validation accuracy, area under the curve (AUC), and loss.
- Returns a composite reward score.

In this study, validation accuracy is selected as the primary optimization criterion due to its effectiveness in measuring overall classification performance.

$$\theta = \{\eta, b, d, n\} \tag{1}$$

Where θ denotes a candidate configuration sampled by the Optuna optimizer, η represents the learning rate, batch size (b), dropout rate (d), and number of classifier layers (n). For each trial, the model is trained using configuration θ and evaluated on a validation dataset.

The objective function is thus defined as:

$$\max_{\theta} J(\theta) = Accuracy_{val}(\theta) \tag{2}$$

The validation accuracy $J(\theta)$ is treated as the reward signal and returned to the optimizer after each trial. Early stopping is employed during training to prevent overfitting, ensuring that the objective function reflects true generalization capability rather than memorization of training data. This formulation enables efficient exploration of the search space and serves as the foundation for subsequent optimization steps in the proposed framework.

b. Step 2: search space design

The search space is formulated as a multi-dimensional vector where each dimension corresponds to a model decision. The decision options are summarised in Table 1. Each configuration is one trial. After sampling several configurations, the sampler learns a probability distribution that favors high-reward (i.e., high-performance) choices.

Table 1. Search space used by the Optuna framework for architecture and hyperparameter optimization

Dense layers	{1-5}	Learning rate	{1e-6-1e-2}
Neurons per layer	{64-256}	Batch size	{16, 32, 64}
Activation function	{ReLU, ELU, Selu}	Optimizer	{Adam, SGD, RMSprop}
Dropout rate	{0.2- 0.5}	Epochs	{10-100}

c. Step 3: optimization process

Optuna runs a number of trials, and each one tests a different group of hyperparameters. It doesn't use brute force methods like Grid Search or Random Search instead, it uses smart algorithms, mostly the TPE, to help with the search. TPE looks at the results of previous trials to find promising areas in the hyperparameter space and suggests new combinations of hyperparameters that are more likely to work better. This lets Optuna quickly move through the search space and find the best hyperparameters.

Optuna also uses a pruning mechanism to find and stop unpromising trials early in order to save time and computing power. This early stopping is based on keeping an eye on the intermediate objective values during a trial and ending those that don't meet a certain condition. This helps keep resources from being wasted on tests that are unlikely to find the best solutions. The algorithm for optimization using Optuna is summarized in Algorithm 1.

Algorithm 1. Optimization using Optuna

Step1: Initialization

```
    a.  Specify the search space (Table1)
    b.  Define objective function: maximize validation accuracy
    c.  Identify starting point(X): Execute a small number of initial trials with randomly
        chosen ss.
```

Step2: Sampling the search space (TPE Sampler and Pruning)

```
Repeat for each trial
 {
   a. For each x ϵ X
   {
   a.  Compute PDF1 = p(x|y < y*)
   b.  Compute PDF2 = p(x|y ≥ y*)
   c.  Compute score(x) = (PDF1/PDF2)
   d.  If score(x) > Threshold then add x to l(x), list of parameters yielding high reward.
       Else add x to g(x), list of parameters yielding low reward.
      }
     b. New parameters will be selected at the point where the ratio of l(x) and g(x) is
maximized.
    c.  Evaluate the trial by executing the objective function
}
```

2.5. Final model selection

After the search is done, the top three configurations are tested with full training cycles to see how well they work. The configuration that performs best on the validation and test sets in terms of generalization is chosen as the final model to be used.

2.6. Evaluation parameters

The final model has been evaluated using multiple metrics. Accuracy, Precision, Recall, F1-score, and receiver operating characteristic-area under the curve (ROC-AUC) are the quantitative metrics used. The Confusion Matrix shows how many true and false positives and negatives there are for each class is used for visualization and the ROC curve is used to represent the models capacity.

3. RESULTS AND DISCUSSION

This study investigates the effectiveness of an automated reinforcement learning–inspired optimization framework for brain tumor detection from MRI images. While previous studies have extensively explored deep learning–based approaches using pretrained CNNs such as VGG16, ResNet50, DenseNet121, and EfficientNet, most existing methods rely on manually designed classifier architectures and heuristically tuned hyperparameters. In contrast, this work directly addresses this gap by optimizing both the architecture of the classifier and the training hyperparameters in a data-driven way. This makes it possible to systematically test performance, robustness, and generalization in situations that are realistic.

3.1. Reinforcement learning-inspired classifier optimization

We used Optuna, a cutting-edge hyperparameter optimization framework based on reinforcement learning, to make the architectural design and hyperparameter tuning of the brain tumor detection model easier. The search space included important design parameters like the number of dense layers, activation functions, and dropout rates. The optimization process used validation accuracy as the objective function to judge each trial configuration. As shown in Table 2, Optuna was able to find high-performing model configurations by exploring and cutting out trials that weren't working well. The best classifier design had three fully connected layers with ReLU activation, a dropout rate of 0.3, and a batch size of 32.

Table 2. Sample Optuna trials and corresponding validation performance

Trial	0	1	2	3	5	6	11	12	15	16	29
Status of trial	Finished	Finished	Finished	Finished	Pruned	Pruned	Finished	Finished	Pruned	Pruned	Finished
Objective function	0.69	0.68	0.70	0.64	Pruned	Pruned	0.69	0.70	Pruned	Pruned	0.95
Number of layers	1	1	3	3	epoch 0	epoch 0	2	2	epoch 0	epoch 0	3
Units	105	166	191	229			70	66			128
Activation Function_hidden layer	Selu	ELU	Selu	ReLU			Selu	Selu			ReLU
Dropout	0.4	0.4	0.5	0.4			0.3	0.3			0.3
Learning rate	1.09E-05	8.59E-04	3.11E-04	8.59E-04			1.39E-04	2.26E-04			5.00E-04
Batch size	64	64	32	32			32	32			32
Optimizer	RMSprop	SGD	Adam	SGD			RMSprop	RMSprop			Adam
Epochs	10	12	12	12			9	8			10
Best trial	0	0	2	2			10	10			29
Objective Function_best	0.69	0.69	0.70	0.70			0.70	0.70			0.95

3.2. Hyperparameter tuning through reinforcement learning

In addition to architectural tuning, the reinforcement learning framework was also tasked with adjusting key training hyperparameters such as learning rate, number of epochs, batch size and optimizer choice. By assigning higher rewards to parameter sets that led to faster convergence and higher validation accuracy, the agent learned to favor a learning rate of 5e-4, 10 training epochs, and the use of the Adam optimizer. This dynamic search allowed the model to avoid suboptimal configurations that are often encountered in grid or random search approaches. Over the course of 30 episodes, the agent improved validation accuracy by nearly 25%, as fewer epochs were required to reach peak performance.

3.3. Evaluation and performance

The training and validation accuracy and loss plots in Figure 2 and Figure 3 show smooth and stable convergence, with no signs of overfitting, reflecting the stability of the reinforcement learning-tuned configuration. The model reached high accuracy within the first few epochs, and continued to improve gradually. Figure 4 shows that the final model, trained with the reinforcement learning-optimized parameters, achieves an accuracy of 92%, an AUC score of 0.93, and an F1-score of 92% as shown in Table 3. The dropout layer contributed to reducing overfitting, especially when the dataset was moderately small in size. The confusion matrix as shown in Figure 5 indicated in strong performance across both tumor and non-tumor classes, with minimal false positives and negatives. These findings validate the effectiveness of reinforcement learning as a meta-controller for optimizing deep learning pipelines in medical imaging applications.

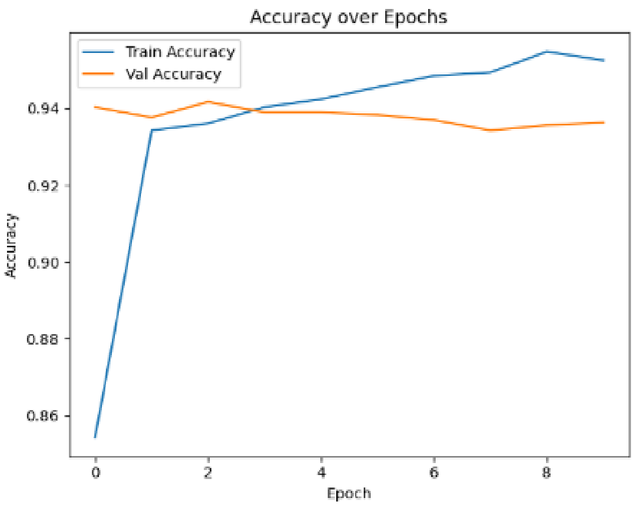

Figure 2. Training and validation accuracy plot for proposed model

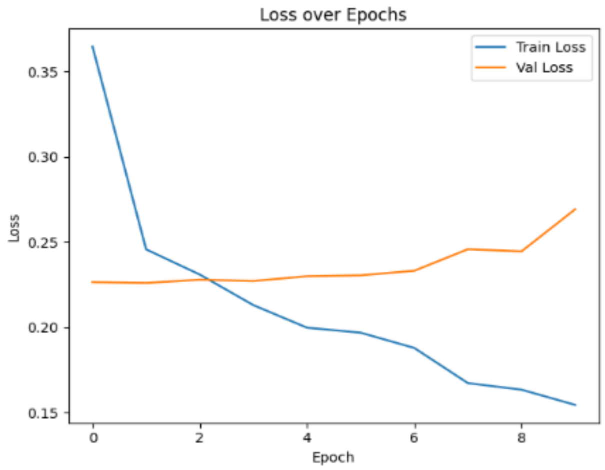

Figure 3. Training and validation loss Plots

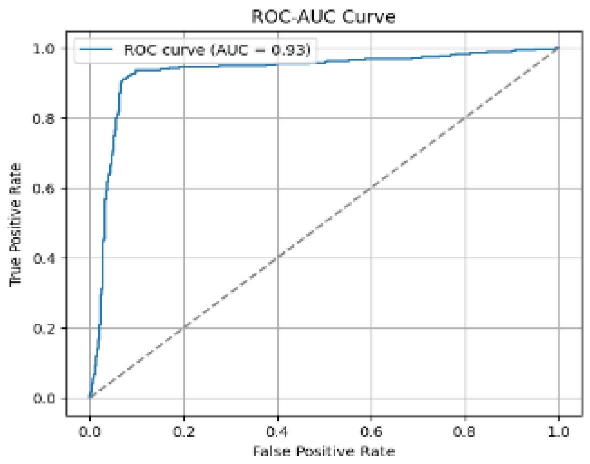

Figure 4. ROC curve of the proposed model

Table 3. Classification report of the proposed model on the BraTS 2020 test dataset

Class	Precision	Recall	F1-score	Support
No tumor	0.91	0.93	0.92	737
Tumor	0.93	0.91	0.92	739
Accuracy			0.92	1476
Macro average	0.92	0.92	0.92	1476
Weighted average	0.92	0.92	0.92	1476

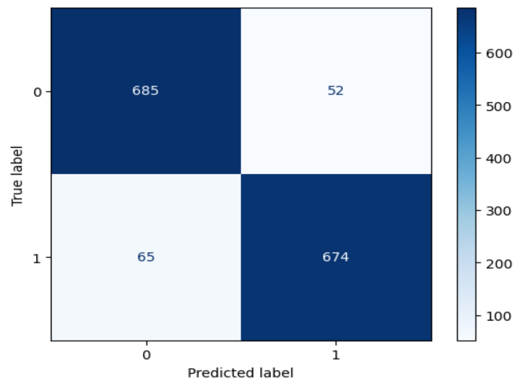

Figure 5. Confusion Matrix of the proposed model

3.4. Comparison with baseline model

To evaluate the effectiveness of the proposed deep reinforcement learning (DRL)-inspired optimization framework, we compared the performance of the Optuna-optimized EfficientNet model against several widely used baseline models in brain tumor detection. These include classical machine learning methods, standard deep learning architectures, and manually tuned variants of EfficientNet. The comparative performance of the proposed framework and the baseline models is summarized in Table 4.

Table 4. Performance comparison of the proposed framework with baseline models on the BraTS 2020 dataset

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
SVM (RBF Kernel) [19]	81.20%	0.8	0.82	0.81	0.84
Capsule network [20]	85.70%	0.84	0.86	0.85	0.88
VGG16 [21]	84.7%	0.83	0.84	0.83	0.87
MobileNetV2 [22]	86.1%	0.85	0.86	0.85	0.88
ResNet50 [23]	88.4%	0.87	0.88	0.87	0.90
DenseNet121[24]	89.6%	0.89	0.89	0.89	0.91
EfficientNet (manual tuning) [1]	87.40%	0.86	0.87	0.86	0.9
EfficientNet + Optuna (proposed)	92.00%	0.91	0.92	0.91	0.96

The classical support vector machine (SVM) with an RBF Kernel [19] showed limited performance due to its inability to capture deep spatial hierarchies in MRI data. Traditional CNN models [20], although more expressive, required manual architecture design and suffered from suboptimal hyperparameter settings. The pretrained ResNet50 model [12] and DenseNet121 shows higher training accuracy but have inconsistent validation performance. Further, DenseNet121 requires more computation time per epoch than EfficientNet. The major reasons for inferior performance of other models are as follows. Firstly, the models like VGG16 and MobileNetV2 have limited feature representation layers due to which they are less effective in capturing the tumor boundary. Manual tuning of EfficientNet performed slightly better, but lacked the systematic optimization that Optuna provides. In contrast, the proposed framework, which integrates Optuna into the architecture and hyperparameter selection process, outperformed all baselines across every metric. This clearly demonstrates the effectiveness of using an automated, DRL-inspired optimization approach to reduce human effort and improve diagnostic accuracy in medical imaging.

Further, the studies in [3], [5], [7], [9], [12] have shown higher performance with accuracy ranging from 98.5 to 99.3%. However these studies are based on the dataset from Kaggle or contain curated or balanced data samples or using dataset-specific fine tuning, thus resulting in higher accuracies. But our work uses the BraTS 2020 dataset, which is more complex, multi-modal and heterogeneous. Therefore, achieving higher accuracies is more challenging and results are typically lower but more realistic.

3.5. Cross dataset evaluation

In order to show that the proposed optimization framework is robust and generalize, this work further evaluates the model on two more widely used brain tumor datasets along with BraTS 2020. These are Figshare Dataset that has images classified into three categories, that is, glioma, meningioma and pituitary and Kaggle Brain MRI dataset. Both these datasets vary from BraTS dataset in terms of acquisition, class distribution, slice quality etc. This allows to test the versatility of the proposed work. Table 5 and Table 6 represents the value of the optimized hyperparameters obtained using Optuna framework and the performance comparison of the proposed method across different datasets.

Table 5. Optimized hyperparameters obtained for Figshare and Kaggle datasets using Optuna

Hyperparameters	Figshare dataset	Kaggle dataset
Trial	35	24
Status of trial	Finished	Finished
Objective function	0.97	0.98
Number of layers	5	7
Units	256	128
Activation function_hidden layer	ReLU	ReLU
Dropout	0.4	0.2
Learning rate	1.09E-05	2.01E-05
Batch size	32	32
Optimizer	Adam	Adam
Epochs	40	37

Table 6. Cross-dataset performance of the proposed framework

Dataset	Task	Accuracy	Precision	Recall	F1-score	AUC
BraTS	Detection	92%	0.91	0.92	0.92	0.96
Figshare	3-class classification	95.4%	0.95	0.95	0.95	0.97
Kaggle	Detection	98.1%	0.98	0.98	0.98	0.99

It can be observed from the results that the proposed framework generalizes well across multiple datasets with varying complexity and class structures. The BraTS 2020 dataset shows the realistic accuracy of 92% as it presents heterogeneity and noise. On the other hand Figshare dataset is more balanced and clear achieves an accuracy of 95.4% and Kaggle dataset achieves an accuracy of 98.1% shows that the proposed framework can achieve state of the art performance when the dataset difficulty is lower.

3.6. Evaluation with imbalanced dataset

Although the primary experiments were conducted on a balanced dataset, real-world medical imaging datasets are often inherently imbalanced, with fewer pathological samples than normal cases. To assess the robustness of the proposed framework under such conditions, additional experiments were performed using artificially imbalanced versions of the dataset.

Class imbalance was introduced by reducing the number of samples in one class, resulting in imbalance ratios of 70:30 and 80:20. The proposed reinforcement learning-optimized EfficientNet model was evaluated using imbalance-sensitive metrics, including precision, recall, F1-score, and ROC–AUC, with particular emphasis on minority-class (tumor) performance. It can be observed from Table 7, that as the class imbalance increases the accuracy has been decreased a little but the recall and F1-score remains stable. This performance can be attributed to the reinforcement learning–inspired Optuna optimization, which dynamically adapts classifier architecture and training hyperparameters.

Table 7. Performance comparison of the proposed framework and baseline model under different class imbalance ratios (tumor: non-tumor)

Imbalance ratio	Model	Accuracy	Precision	Recall	F1-score	ROC–AUC
50 : 50	EfficientNet (manual tuning)	87.4%	0.86	0.87	0.87	0.90
	Proposed EfficientNet + Optuna	92%	0.91	0.92	0.92	0.96
70 : 30	EfficientNet (manual tuning)	84.6%	0.83	0.82	0.83	0.87
	Proposed EfficientNet + Optuna	90.3%	0.89	0.90	0.89	0.94
80 : 20	EfficientNet (manual tuning)	81.2%	0.79	0.78	0.78	0.84
	Proposed EfficientNet + Optuna	88.1%	0.87	0.88	0.87	0.92

3.7. Discussion on performance gains and explainability perspective

The proposed optimization framework, which is based on EfficientNet and inspired by reinforcement learning shows better performance. This is due to the combined effect of the effective feature representation, automatic architecture adaptation, and hyperparameter tuning.

Firstly, the EfficientNet backbone makes sure that the network's depth, width, and resolution are all in the right place. This provides detailed spatial and intensity-based features from brain MRI scans which is very important for the BraTS 2020 dataset because the edges of tumor regions are often not clear and they look different from each other. In comparison to the other deep learning models, EfficientNet can find patterns that are specific to tumors while using less computer power. The VGG16 and MobileNetV2 have limited feature representation capabilities. But deeper architectures like ResNet50 and DenseNet121 usually need more processing power and don't always work well on complicated medical datasets, which limits their practical efectiveness.

Secondly, the Optuna framework which is inspired by reinforcement learning plays an important role in the performance enhancement of the proposed model, as it automatically optimizes the training hyperparameters and the classifier architecture. The suggested framework dynamically searches the search space for the best configurations for the task, while manually tuned models rely on fixed design choices. The smooth training curves and balanced precision-recall values show that the model can reach stable convergence, reduce overfitting, and improve generalization due to this two-level optimization.

Third, the suggested method shows strong generalization on clinically important and varied data. The BraTS 2020 dataset has a lot of differences in imaging modalities, tumor types, and acquisition protocols. This makes it harder to use than well-organized or balanced datasets like Kaggle or Figshare. The proposed framework achieves realistic and dependable performance by directly optimizing the learning process on this intricate distribution. This is why the models that were manually tuned and the ones that were set up as a baseline improved.

Considering the interpretability perspective, explainable artificial inteligence (XAI) methods show which parts of an image are most important for the model's predictions. These techniques are very useful in medical imaging because they make sure that the model only looks at tumor areas that are important to doctors and not background noise. They also highlight the specific areas which contributed to the results and helps to understand the results more effectively. The current study is focusing on the automated optimization for brain tumor detection. The integration of XAI-based visual analysis, on the other hand, is a big step forward for the future. Using these methods would make things even clearer and more reliable in the clinic by letting you compare the baseline models to the proposed framework in a qualitative way.

4. CONCLUSION AND FUTURE SCOPE

In this work, we proposed an automated, reinforcement learning–inspired optimization framework for brain tumor detection using MRI images, with the aim of reducing the reliance on manual model design and heuristic hyperparameter tuning. The proposed optimization framework, which uses EfficientNetB0 as the feature extractor with the Optuna, can automatically compute both the architecture of classifier and the training parameters. This design helps the model learn better from complicated medical images while cutting down on the need for a lot of expert help.

The results show that the proposed approach outperforms both the baseline and manually tuned models. The framework has a high classification accuracy and works well in a variety of testing situations. Importantly, when evaluated under imbalanced data conditions, the model maintains reliable detection of the tumor class, which is critical in clinical applications where missing a positive case can have serious consequences. Further, the cross-dataset evaluations also show that the proposed method improves generalization, thus making it less sensitive to the dataset-specific characteristics.

The primary contribution of this study is demonstrating that the simultaneous optimization of classifier structure and training hyperparameters in an automated fashion can enhance the robustness and reliability of brain tumor detection systems. Unlike traditional transfer learning methods that use fixed architectures, the proposed framework changes based on the characteristics of the data. This makes it work better and more reliably in tough situations. These results show how useful reinforcement learning-inspired optimization can be for analyzing medical images.

As a future scope this work can be extended with other backbone architectures like deeper convolutional networks and vision transformers. The explainable AI like Grad-Cam, LIME etc can help make model predictions more understandable by giving visual information about them. More advanced methods for dealing with severe class imbalance, like cost-sensitive learning and adaptive sampling, may also improve performance.

Finally, testing the proposed framework on large clinical datasets from multiple institutions and looking into how it could be used in real time will be important steps toward getting it used in real life. Overall, this study shows that automated optimization frameworks are a promising way to make brain tumor detection systems that are accurate, strong, and useful in the clinic.

FUNDING INFORMATION

This research received no specific grant from any funding agency.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Aashutosh Kharb	✓	✓	✓	✓	✓	✓		✓	✓	✓				
Prachi Chaudhary		✓						✓		✓		✓		

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest.

DATA AVAILABILITY

The data, which contain information that could compromise the privacy of research participants, are not publicly available due to certain restrictions. Also, no new data were created or analyzed in this study.

REFERENCES

[1] M. Tan and Q. V. Le, “EfficientNet: rethinking model scaling for convolutional neural networks,” *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.

[2] U. Baid *et al.*, “The RSNA-ASNR-MICCAI BraTS 2021 benchmark on brain tumor segmentation and radiogenomic classification,” 2021, [Online]. Available: http://arxiv.org/abs/2107.02314

[3] S. Srinivasan, D. Francis, S. K. Mathivanan, H. Rajadurai, B. D. Shivahare, and M. A. Shah, “A hybrid deep CNN model for brain tumor image multi-classification,” *BMC Medical Imaging*, vol. 24, no. 1, p. 21, 2024, doi: 10.1186/s12880-024-01195-7.

[4] H. M. Rai and K. Chatterjee, “2D MRI image analysis and brain tumor detection using deep learning CNN model LeU-Net,” *Multimedia Tools and Applications*, vol. 80, no. 28–29, pp. 36111–36141, Nov. 2021, doi: 10.1007/s11042-021-11504-9.

[5] M. D. Baig, H. B. U. Haq, W. Akram, and A. M. Awan, “Brain tumor detection enhanced with transfer learning using SqueezeNet,” *Decision Making Advances*, vol. 2, no. 1, pp. 129–141, 2024, doi: 10.31181/dma21202432.

[6] V. N. V. L. S. Swathi, K. Sinduja, V. R. Kumar, A. Mahendar, G. V. Prasad, and B. Samya, “Deep learning-based brain tumor detection: An MRI segmentation approach,” *MATEC Web of Conferences*, vol. 392, p. 01157, 2024, doi: 10.1051/matecconf/202439201157.

[7] M. T. R. Shawon, G. M. S. Shibli, F. Ahmed, and S. K. S. Joy, “Explainable cost-sensitive deep neural networks for brain tumor detection from brain MRI images considering data imbalance,” *Multimedia Tools and Applications*, vol. 84, no. 35, pp. 43615–43642, Apr. 2025, doi: 10.1007/s11042-025-20842-x.

[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, Feb. 2021, doi: 10.3390/healthcare9020153.

[9] M. A. Karagoz, O. U. Nalbantoglu, and G. C. Fox, “Residual vision transformer (ResViT)-based self-supervised learning model for brain tumor classification,” *arXiv preprint arXiv: 2411.12874*, 2024, [Online]. Available: http://arxiv.org/abs/2411.12874

[10] M. Aamir *et al.*, “Brain tumor detection and classification using an optimized convolutional neural network,” *Diagnostics*, vol. 14, no. 16, p. 1714, Aug. 2024, doi: 10.3390/diagnostics14161714.

[11] F. A. Mughal *et al.*, “A self-supervised learning model for the classification of brain tumors using medical images: a review,” *Journal of Medical & Health Sciences Review*, vol. 2, no. 2, Jun. 2025, doi: 10.62019/w94hxe75.

[12] A. Singh and S. Jain, “Enhanced brain tumor detection from brain MRI images using convolutional neural networks,” *AIMS Bioengineering*, vol. 12, no. 2, pp. 215–224, 2025, doi: 10.3934/bioeng.2025010.

[13] R. Patel and D. Gandhi, “Robust MRI-based brain tumor detection using hybrid feature learning and self-supervised pretraining,” *Journal of Neonatal Surgery*, vol. 14, no. 24S, pp. 876–888, 2025, [Online]. Available: https://www.jneonatalsurg.com/index.php/jns/article/view/6068

[14] P. A. Meshram, S. Joshi, and D. Mahajan, “Brain tumor detection using Swin Transformers,” *arXiv preprint arXiv: 2305.06025*, pp. 1–8, 2023.

[15] I. Stathopoulos *et al.*, “Evaluating brain tumor detection with deep learning convolutional neural networks across multiple MRI modalities,” *Journal of Imaging*, vol. 10, no. 12, p. 296, Nov. 2024, doi: 10.3390/jimaging10120296.

[16] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

[17] M. Hu, J. Zhang, L. Matkovic, T. Liu, and X. Yang, “Reinforcement learning in medical image analysis: Concepts, applications, challenges, and future directions,” *Journal of Applied Clinical Medical Physics*, vol. 24, no. 2, Feb. 2023, doi: 10.1002/acm2.13898.

[18] F. Sahba, H. R. Tizhoosh, and M. M. A. Salama, “A reinforcement learning framework for medical image segmentation,” in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, IEEE, 2006, pp. 511–517, doi: 10.1109/IJCNN.2006.1716136.

[19] S. Chaplot, L. M. Patnaik, and N. R. Jagannathan, “Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network,” *Biomedical Signal Processing and Control*, vol. 1, no. 1, pp. 86–92, Jan. 2006, doi: 10.1016/j.bspc.2006.05.002.

[20] E. Goceri, “CapsNet topology to classify tumours from brain images and comparative evaluation,” *IET Image Processing*, vol. 14, no. 5, pp. 882–889, 2020, doi: 10.1049/iet-ipr.2019.0312.

[21] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[24] G. Huang, Z. Liu, and L. van der Maaten, “Densely connected convolutional networks,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 39, no. 9, pp. 1442–1446, 1978.

BIOGRAPHIES OF AUTHORS

Aashutosh Kharb received his B.Tech. (Instrumentation and Control Engineering) and M.Tech. (Electronics and Communication Engineering) from MDU, Rohtak, Haryana, in 2006 and 2011, respectively. Currently, he is a research scholar at Deenbandhu Chhotu Ram University of Science and Technology (DCRUST), Murthal, Haryana, India (erstwhile Chhotu Ram State College of Engineering (CRSCE), Murthal) since 2008. He has published various research papers in journals and conferences. He can be contacted at email: aashutoshkharb@gmail.com.

Prachi Chaudhary received her B.Tech. (Electronics and Communication Engineering) degree from JIET, Jind, Haryana-India (affiliated to K.U. University, Kurukshetra) in 2005. She received her M.Tech. (Electronics and Communication Engineering) degree from NCCE, Israna, Haryana-India (affiliated to K.U. University, Kurukshetra) in 2007. She received her Ph.D. degree in Electronics and Communication Engineering from K.U. University, Kurukshetra (India) in 2016. She worked as a Faculty Member in ECE Department in BMIET, Sonipat and Hindu College of Engineering, Sonipat, Haryana – India. She joined as Lecturer in ECE Department of Deenbandhu Chhotu Ram University of Science and Technology (DCRUST), Murthal, Haryana - India (erstwhile Chhotu Ram State College of Engineering (CRSCE), Murthal) in 2008 and is currently working as associate professor in ECE Department. She has published more than 40 research papers in journals and proceedings of national and international conferences. She is a life member of ISTE. Her interests include soft computing techniques to digital signal processing, microprocessors, microcontrollers and digital VLSI. She can be contacted at email: dahiya.prachi@gmail.com.