Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 14, No. 4, 2026, 501-520.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

DURL-Net: Integrating EfficientNet-B7 U-Net and Deep Q-Network for Brain Tumor Segmentation and Adaptive Morphological Refinement

Omid Khalaf Beigi[1] and Seyed Alireza Bashiri Mosavi[2*]

1. Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran.
2. Department of Electrical and Computer Engineering, Buein Zahra Technical University, Buein Zahra, Qazvin, Iran.

Article Info

Article History:
Received 08 March 2026
Revised 30 June 2026
Accepted 17 July 2026

DOI:10.22044/jadm.2026.17571.2896

Keywords:
MRI, Brain Tumor, Deep Reinforcement Learning, Deep Q-Network, Image Segmentation.

**Corresponding author: abashirimosavi@bzte.ac.ir (S. A. Bashiri Mosavi)*

Abstract

A brain tumor is one of the most serious and life-threatening brain diseases that can profoundly affect an individual's life. Accordingly, the present study addresses the challenge of refining brain tumor segmentation based on Magnetic Resonance Imaging (MRI) data and deep reinforcement learning. Although supervised learning–based approaches have shown satisfactory performance in tumor segmentation and localization, they often suffer from high uncertainty errors along tumor boundaries. In this research, a learning framework combining a supervised model with deep reinforcement learning—referred to as DURL-Net—is proposed for segmentation and refinement purposes. Specifically, the framework first employs a U-Net architecture to generate an initial segmentation mask. This initial output and the corresponding MRI are then partitioned into localized patches, which are sequentially processed by a Deep Q-Network (DQN) agent. The DQN agent interacts with the environment by selecting optimal morphological operations (such as dilation and erosion) to refine tumor boundaries and correct uncertainties patch by patch. The dataset used in this study comprises 3,064 T1-Weighted Contrast-Enhanced MRI images, employed for both segmentation and tumor-type classification tasks. Experimental results demonstrate that DURL-Net achieved a Dice Similarity Coefficient (DSC) of 86.73%, a Jaccard Index (IoU) of 78.68%, a Kappa coefficient (Kap) of 85.21%, a Sensitivity of 87.68%, and a Specificity of 96.06%.

1. Introduction

A brain tumor (BT) refers to an abnormal mass or uncontrolled growth of cells within the brain (intracranially) or its surrounding tissues. These tumors arise from the uncontrolled proliferation of cells and can be either benign (non-cancerous) or malignant (cancerous). Benign brain tumors may grow gradually over time and exert pressure on brain tissue, whereas malignant tumors—commonly referred to as brain cancer—have the ability to invade and destroy brain tissue. The size of brain tumors can vary widely, ranging from very small to very large masses. Some tumors can be detected at early stages due to noticeable symptoms, while others may become considerably large before being diagnosed. Moreover, certain regions of the brain may exhibit signs of tumor-induced damage later than others; therefore, if a tumor develops in a less active area of the brain, it may initially produce no obvious symptoms.

Brain tumors can be classified as either primary, originating within the brain or central nervous system itself, or secondary, which spread to the brain from other parts of the body. To understand the prevalence of brain tumors, it is important to note that between 2017 and 2021, a total of 87,053 deaths were recorded in the United States due to

malignant brain tumors and other central nervous system (CNS) neoplasms, averaging approximately 17,411 deaths per year [1, 2]. Moreover, brain tumors are ranked as the tenth leading cause of cancer-related deaths in the United States [3]. The increase in intracranial pressure, displacement of brain structures, and damage to healthy nerves and brain tissue caused by tumors can disrupt normal brain function and result in severe neurological impairments [4]. Therefore, brain tumors pose a serious health threat, and accurate detection of their location and type is of paramount importance. The identification of brain tumors employs various diagnostic methods. Historically—and even today—this process has typically been performed by skilled physicians and medical specialists. The diagnosis of a brain tumor is a multistage procedure that begins with the assessment of clinical symptoms and is subsequently confirmed through advanced imaging modalities and specialized laboratory tests.

Computed Tomography scanning, commonly referred to as a CT scan, is considered one of the most accessible and fastest medical imaging techniques. This method utilizes X-rays to generate images that can reveal abnormalities and structural issues within or around the brain. It is worth noting that CT scanning is typically used as an initial approach for assessing brain conditions. However, compared to other imaging modalities, the resulting images provide less detailed visualization of soft brain tissues.

In the Magnetic Resonance Imaging (MRI) technique, strong magnetic fields and radio waves are used to produce highly detailed images of the brain's internal structures. It should be noted that, compared to the previous method, MRI is relatively slower; however, it provides images with greater detail and higher resolution of brain structures. For example, in an MRI scan of a healthy brain, the white matter**,** gray matter, and cerebrospinal fluid (CSF) are clearly distinguishable [5].

The Positron Emission Tomography (PET) technique employs more advanced technology compared to the previously mentioned imaging methods. Unlike the latter, which visualize the physical structures of the body, PET focuses on measuring the metabolic activity of tissues. In this process, a small amount of a simple sugar (glucose) labeled with a radioactive tracer is injected into the body. Since cells require glucose for energy, the tracer is absorbed by them. Cells with higher metabolic activity—such as cancerous cells—consume more glucose than normal cells. Consequently, after the tracer is absorbed, gamma rays are emitted from the more active cells. A connected computer then analyzes the location and intensity of these emissions to reconstruct the final image. Examples of the three aforementioned imaging approaches are illustrated in Figure 1.

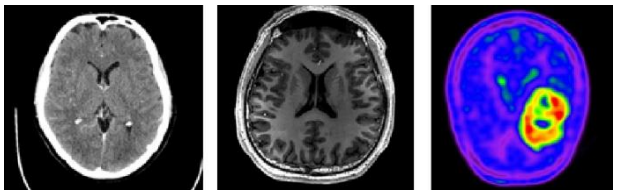

Figure 1. From left to right: CT scan, MRI, and PET

Based on the discussions, it can be concluded that the process of brain tumor segmentation and diagnosis requires substantial time, expertise, and specialized knowledge. With the rapid advancement of computer science and artificial intelligence (AI) in recent years, the detection of various diseases and disorders has undergone significant transformation. Machine learning–based approaches have introduced remarkable speed, reliability, and accuracy in medical detection and diagnostic procedures [6, 7].

In the field of brain tumor analysis, automated segmentation algorithms have been developed using a variety of computational strategies. These methods can generally be categorized into unsupervised, supervised, deep learning, and reinforcement learning–based approaches. Each category offers unique advantages and faces specific challenges in terms of accuracy, robustness, and computational efficiency.

Despite considerable progress, accurately delineating tumor boundaries and refining segmentation outputs remain challenging tasks. Therefore, there is a need for intelligent frameworks that combine powerful feature learning with effective decision-based post-processing. In this work, we propose a hybrid approach that integrates U-Net–based segmentation with a Deep Q-Network (DQN) post-processing strategy to enhance tumor region refinement. A review of related studies is presented in the following section.

2. Related Works

Among unsupervised methods, the study presented in [8] is a notable example. In their proposed approach, to reduce noise and imaging artifacts, the Bayes Shrink algorithm—based on Adaptive Wavelet Thresholding—is first applied to the image. Subsequently, the Patch-Based K-Means algorithm is used to remove the skull from the image. During the skull stripping process, a 3×3 patch is extracted around each pixel, which is then

converted into a patch matrix. The K-Means algorithm with two clusters is applied to this matrix to classify pixels according to their neighborhood features. The goal of skull removal is to isolate the brain tissue for easier and more accurate tumor segmentation. After obtaining the brain tissue, an initial tumor segmentation is performed using a hybrid algorithm called Fuzzy-Possibilistic C-Means (FPCM), consisting of four clusters. The preliminary segmentation (identified as the RTC cluster) contains both brain tumor regions and cerebrospinal fluid (CSF). Therefore, to accurately extract the tumor region**,** connected components are first identified, and holes within each component are filled, resulting in solid, hole-free clusters. Although the proposed framework is relatively simple to implement, it suffers from limited generalizability. This limitation arises because the researchers focused on T2-weighted MRI data (BraTS dataset), where the predefined intensity ranges may not necessarily hold true for other datasets or samples. Furthermore, the proposed method processes data on a slice-by-slice basis, without considering the volumetric relationships between slices.

Another study that introduces a framework based on an unsupervised approach is presented in [9]. Similar to the previous work [8], where a hierarchical sequence of steps was defined for brain tumor segmentation, the study in [9] proposed a five-stage pipeline for brain tumor segmentation using the BraTS 2012 dataset. It is noteworthy that, unlike [8], which utilized only a single MRI modality, this study employed a combination of multiple MRI modalities (T1, T1c, T2, and FLAIR) to achieve more precise tumor segmentation. In the proposed framework, the MRI pixel intensities are first mapped to the range [0, 255] to minimize intensity differences among various MRI scales. Then, the Anisotropic Diffusion filter is applied to remove image noise. After denoising, the pixels are clustered into four primary groups—gray matter, white matter, CSF, and abnormal regions such as tumors—using the K-Means algorithm. The outcome of this stage is the separation of normal brain tissues and CSF from potential tumor regions. Next, the Hierarchical Centroid Shape Descriptor (HCSD) is employed to correct misclassified regions and to eliminate healthy tissues erroneously grouped as abnormal by the K-Means algorithm. The framework proposed in [9] leverages multi-scale information from different MRI modalities for final tumor segmentation, resulting in more accurate delineation. In other words, by integrating spatial and shape information across multiple scales, the final segmentation outcome becomes more reliable. However, for diffuse or irregularly shaped tumors, the framework may not perform optimally. Moreover, due to its multi-stage structure, any misclassification introduced by the K-Means algorithm can propagate through subsequent steps, causing the HCSD correction process to also produce errors.

One of the main limitations of unsupervised clustering algorithms is the need to manually and statically define the value of K. As observed in studies [8] and [9], the number of clusters (K) was predetermined and fixed. Following such an approach reduces the generalizability of the method across different samples and datasets while increasing its sensitivity to outliers. To address this issue, in study [10], researchers first applied a Median Filter to remove image noise. Then, a histogram of pixel intensities was generated using DICOM and BraTS datasets. By identifying the peaks of the gray-level pixels and assigning them as the initial cluster centers, an initial segmentation of the suspected brain tumor regions was performed. Subsequently, the tumor size estimation algorithm was applied to obtain the final tumor segmentation.

Although the K-Means algorithm is known for its simplicity, low computational cost, and high processing speed [11, 12], it assigns each pixel exclusively and definitively to a single cluster. This limitation leads to inaccuracies, particularly along tumor boundaries, where pixels often exhibit intermediate gray-level intensities. To overcome this issue, in study [13], researchers proposed a framework based on the Fuzzy C-Means (FCM) algorithm. Unlike K-Means, FCM employs a soft clustering approach, assigning a membership degree to each pixel for every cluster.

Despite the unsupervised approaches, which perform brain tumor segmentation based on pixel clustering**,** supervised (shallow machine learning) approaches assign a class label to each pixel [14]. In these methods, a feature vector is first extracted from the image, and then tumor segmentation is performed using classification algorithms. In other words, the output of the trained model is a tensor of size H×W×C (corresponding to depth, height, and width of the image), where each element represents a single pixel. Compared to unsupervised approaches, a greater number of studies have applied supervised methods for brain tumor segmentation.

Among the existing studies in this field, the work presented in [15] can be highlighted. In the proposed method, for each pixel of an MRI image (for each of the TD, PD, and FLAIR sequences,

whether inside or outside the tumor), an 11×11 window is considered. From each window, eight features are extracted: Intensity, Mean, Variance, Geometric Mean, Harmonic Mean, Median Absolute Deviation, Skewness, and Kurtosis. After obtaining these features for each window, a feature matrix is generated for each sequence, which is then mapped to a lower-dimensional feature space using Principal Component Analysis (PCA). Features with the highest variance are retained, while the remaining features are discarded. Subsequently, the Multi-Kernel Support Vector Machine (SVM) algorithm is used to classify the pixels of each sequence independently, employing different Gaussian kernels for each sequence. A logical AND operation (fusion phase) is then applied to the results from all three sequences to obtain the initial segmentation. Finally, region growing based on distance from the boundary is applied to generate the final segmentation. It is noteworthy that the proposed method has an adaptive structure, allowing for retraining on subsequent images of the same patient. Compared to unsupervised approaches, this method requires less computation; moreover, the use of a Multi-Kernel SVM and multiple sequences enables better modeling of tumor regions. However, due to the evaluation of the method on a limited dataset and the requirement for alignment among all sequences, its performance may degrade on similar but unseen data.

As observed in the previously discussed method, the extraction of features for classification was performed manually. However, in study [16], the researchers proposed a framework for brain tumor segmentation based on 20 samples from the BraTS 2013 dataset. In the proposed method, the MRI samples are first subjected to bias correction using the N4ITK algorithm to remove the effects of intensity inhomogeneity. Then, Histogram Matching is applied to normalize the gray-level range across the images. After preprocessing, the image is divided into small, homogeneous regions called superpixels using the Simple Linear Iterative Clustering (SLIC) algorithm. Inspired by K-Means, SLIC clusters pixels based on spatial proximity and intensity similarity. From each superpixel, statistical features such as mean, standard deviation, skewness, and kurtosis, along with texture and intensity descriptors, are extracted. These features enable discrimination between healthy tissue and tumor regions. Subsequently, the extracted features are used to train a Support Vector Machine (SVM), and the classification of superpixels produces the final brain tumor segmentation. One of the strengths of this method is the use of superpixels, which contributes to a more stable processing pipeline. However, the proposed approach was trained on only 20 samples, which is a very limited dataset. Moreover, unlike previous methods, it relies solely on the FLAIR sequence for generating the final tumor mask. It is noteworthy that, despite the advantages of SVM in modeling the data, both studies face memory and training time limitations when applied to larger datasets.

In study [17], researchers proposed a hybrid framework for brain tumor segmentation based on 30 samples (20 high-grade and 10 low-grade tumor cases) from the BraTS 2013 dataset, including the T1, T1c, T2, and FLAIR sequences. The framework combines Random Forests with morphological operations. In the proposed method, intensity normalization and bias correction are first applied as preprocessing steps to enhance the performance of the segmentation framework. Then, a feature vector is extracted for each pixel, comprising intensity features, gradient-based features (for boundary detection), and context-aware features (simulating human perception when interacting with tissue regions). For initial segmentation, a Random Forest classifier consisting of 100 decision trees with a maximum depth of 45, trained using the Gini criterion, is employed. After obtaining the initial segmentation, morphological operations are applied to remove small regions caused by misclassification. Although the proposed framework demonstrates satisfactory performance, it is sensitive to class imbalance between high-grade and low-grade tumors. Additionally, similar to previous studies, the number of samples used to evaluate the framework is limited, which may restrict its generalizability.

In study [18], researchers proposed a framework based on the k-Nearest Neighbors (kNN) algorithm to detect the location and type of brain tumors (using a dataset collected by the researchers). In this framework, it is assumed that the brain exhibits approximate natural symmetry. Therefore, the MRI image is divided into left and right hemispheres, and the intensity histograms for each hemisphere are computed. The difference between the two histograms is then calculated, and a significant discrepancy in intensity distribution indicates the presence of an abnormal region (tumor). This process results in the detection of the tumor region. Due to the potential presence of noise and unwanted areas, an Order Statistic Filter (OSF) may be applied after intensity calculation. This filter sorts pixel values, preserves tumor edges, and removes noise. Subsequently, morphological

operations (dilation and erosion) are used to eliminate small regions and remaining noise. Finally, the precise location and shape of the tumor are identified. Once the tumor region is determined, the tumor surface pixels are counted, and their relative percentage with respect to the entire image is calculated. Based on these values, the Euclidean distance between the new sample and the training data is computed to classify the tumor as benign or malignant. The framework proposed in [18] is computationally simple and easy to implement; however, the final tumor type classification heavily depends on the accuracy of preprocessing and tumor extraction. Moreover, the performance of this framework has not been compared with more advanced methods, such as deep learning–based approaches.

In addition to shallow machine learning approaches, a few reinforcement learning–based methods also exist. For instance, in [19], the researchers proposed a framework based on the Q-Learning algorithm for the segmentation of target regions. In the proposed framework, the ultrasound image (dataset containing prostate images) is first divided into several sub-images. Then, using a Q-learning agent, the parameters for local thresholding and structural elements for morphological operations (opening) are determined for each sub-image. The agent is trained through a trial-and-error process, receiving rewards and punishments, which results in the formation of a Q-matrix. Ultimately, the agent can perform the segmentation of the target region. The agent can continue to train online when new images are provided. Moreover, due to the use of reinforcement learning, this approach requires fewer training samples compared to unsupervised or supervised methods. However, because the image is divided into sub-images and computations are performed on each, the computational complexity increases. In addition, the absence of a preprocessing and noise removal step may lead to errors in segmentation performance.

Based on the previously discussed supervised and unsupervised approaches, it can be concluded that unsupervised methods do not require labeled data. In other words, they discover patterns by analyzing the relationships among samples and their values. In the context of brain tumor segmentation, these methods typically rely on simple features, such as pixel intensity and tissue-related details, for modeling. Despite their advantages, unsupervised approaches often perform poorly in defining tumor boundaries and edges, and their generalizability to new data is limited. Shallow machine learning methods, although interpretable, simple, and computationally efficient, require the manual definition of feature vectors. Manual feature engineering can reduce the generalization capability of the model, as the defined features may not hold for new data. Additionally, shallow machine learning approaches struggle to capture complex patterns. To address these limitations, deep learning–based approaches can be effective. In deep learning, features are automatically learned during the training process, and mechanisms such as attention can be employed to improve tumor boundary delineation.

Compared to unsupervised and supervised approaches, a considerably larger number of studies have been conducted using deep learning methods. Among the existing research in this area, study [20] can be highlighted. In this study, the researchers proposed a Convolution Neural Network (CNN)–based framework for brain tumor segmentation using the BraTS 2013 and BraTS 2015 datasets. In the proposed framework, intensity inhomogeneities caused by MRI acquisition are first corrected using the N4ITK method, and then the intensities are normalized to ensure consistency across patients and imaging devices. After preprocessing, image patches are extracted and normalized to have zero mean and unit variance for each sequence (T1, T1c, T2, FLAIR). Following preprocessing, tumor segmentation is performed using a CNN model consisting of small 3×3 kernels (to reduce overfitting and enable learning of complex patterns), max-pooling layers, and fully connected layers. For data augmentation, patches were rotated by 90, 180, and 270 degrees. The cross-entropy loss function was used to classify five categories: healthy tissue, edema, necrosis, non-enhancing tumor, and enhancing tumor. By normalizing the intensities, the framework reduces data heterogeneity. Moreover, the use of data augmentation and 3×3 kernels reduces the likelihood of overfitting compared to similar approaches. However, the patch-based nature of the framework limits the utilization of broader spatial information from the tumor and brain tissue. Additionally, despite data augmentation, class imbalance remains an issue.

One of the challenges in applying deep learning for data modeling is the need to increase the number of layers and model parameters when dealing with more complex data. Increasing the number of layers leads to higher computational time and memory requirements and may reduce the generalizability of the model. To address this issue, in study [21], the researchers proposed a lightweight framework based on the U-Net

architecture. U-Net is an encoder-decoder architecture designed for semantic segmentation. To reduce the number of parameters in the U-Net model, the researchers downscaled the 3D images from the BraTS 2015 dataset from 572×572 to 128×128 and then converted them into 2D slices along three planes. These modifications reduce the computational time and memory requirements. After data preparation, the number of convolution layers in each block was reduced, following an incremental strategy with a step size of 16, i.e., the number of convolution layers starts from 16 and increases to 64 in the middle block. Due to the lightweight nature of the architecture, data augmentation was not required. The model was trained using the Binary Cross-Entropy (BCE) loss function and the Adam optimizer. It should be noted that the proposed architecture was evaluated on a limited dataset of 13 patients using only the T1-weighted modality. Therefore, the performance of the architecture may be limited when applied to larger datasets.

In another study, researchers proposed a U-Net–based model for brain tumor segmentation. In study [22], the goal was to design a model with fewer parameters compared to the standard U-Net. One challenge in the BraTS 2020 dataset is the presence of many MRI slices containing only healthy brain tissue without any tumor. In most slices, no tumor is present, and including these slices in training can introduce noise and complicate the learning process. To address this, the researchers applied morphological operations (opening and closing) and measured the area of regions to remove slices without tumors. They then equalized the number of slices per sample and normalized pixel intensities to the range of 0 to 1. After preprocessing, the proposed model, SEDNet, was trained. SEDNet is based on an encoder-decoder structure but has shallower depth compared to U-Net. In the encoder component, the number of convolution layers starts at 32 and increases to 256 in the deeper layers. In the decoder, the number of channels decreases from 256 back to 32. At the end of the architecture, a final convolution layer is used for tumor segmentation. Unlike U-Net, which includes skip connections from each encoder block to its corresponding decoder layer, SEDNet only establishes connections for the initial layers, referred to as the Selective Skip Path. Despite the architectural improvements over U-Net, SEDNet exhibited suboptimal performance for small tumors. Additionally, the researchers only used the FLAIR sequence for training and did not incorporate other MRI sequences. In the architectures described in the previous sections, it was observed that their operation is based on an encoder-decoder structure and follows a mapping paradigm. Specifically, the encoder extracts high-level features from the images and generates a set of feature representations, while the decoder uses these features to produce the final tumor segmentation. However, this architecture faces limitations in segmenting tumor edges and small tumors. To address these challenges, Generative Adversarial Networks (GANs) have shown promising results. GAN is a deep learning–based framework designed to generate samples that resemble the distribution of real data. It consists of two components: a generator and a discriminator. The generator creates new samples based on a random input vector, while the discriminator evaluates whether a given sample is real or generated. In study [23], the researchers proposed a GAN-based framework for brain tumor segmentation using the BraTS 2021 dataset. The framework is built upon a U-Net structure, consisting of an encoder and decoder that process a preprocessed image (standardized and normalized in size and intensity) to segment the tumor. All MRI modalities (T1, T1c, T2, FLAIR) were utilized in this study. After tumor segmentation, the generator produces the segmentation output, and a 3D fully connected network is used as the discriminator to compute the error between the ground truth and the generated segmentation. Both components are then trained iteratively to improve the segmentation performance.

Based on the reviewed studies, it can be observed that most proposed frameworks and architectures for brain tumor segmentation are based on unsupervised or supervised methods (using labels or masks). However, only a few studies have explored the application of reinforcement learning (RL) or deep reinforcement learning (Deep RL) in brain tumor segmentation. For instance, in study [24], the researchers proposed a framework based on Multi-Agent Reinforcement Learning (MARL) and the Asynchronous Advantage Actor-Critic (A3C) algorithm for brain tumor segmentation using the BraTS 2015, MM-WHS, and NCI-ISBI 2013 datasets. The A3C algorithm is an actor-critic–based method, consisting of a value network and a policy network. The value network is responsible for estimating the value of the current state (i.e., how appropriate the prediction is), while the policy network is responsible for selecting actions.

In the proposed framework, one agent is assigned to each voxel, and each agent adjusts the probability of the corresponding voxel label by a small amount (e.g., 0.1 or 0.2) to achieve the final

segmentation. This process results in gradual and subtle updates to the tumor segmentation at each step. The decision-making agent determines its actions based on the current voxel (time t), the segmentation probability from the previous step (time t-1), the Object Hint Map, and the Background Hint Map. These maps are generated based on clicks provided by an expert user. After applying the action to the current voxel, the error between the current and previous states is computed. If the segmentation improves, the agent receives a reward; otherwise, a penalty is applied. Despite its innovative approach and performance, the proposed framework requires user interaction during training and entails high computational complexity due to the use of a multi-agent reinforcement learning strategy.

In another study [25], the researchers proposed a framework called DGARL, which combines a GAN with Soft Actor-Critic (SAC). In this framework, the SAC component operates to identify the tumor-related region, and once the target region is located, it is passed to the generator network to produce the final segmentation. In the SAC component, both forward exploration and backward exploration strategies are employed. During forward exploration, the agent moves from the center of the image toward the tumor region to gain experience, while in backward exploration, the agent moves from the tumor region back to the center to refine its experiences. Despite the innovative approach and promising performance, the framework is associated with high computational complexity, inefficient training (due to starting a new training session with the generator network), and a potential for segmentation errors in the generator output, since it relies on the agent to first detect the tumor region.

Given the significance and potential risks of tumors for brain function, accurate segmentation and classification of tumor types are of critical importance. Although supervised and unsupervised approaches perform reasonably well in brain tumor segmentation, they often exhibit high uncertainty errors at tumor boundaries. In other words, most of these methods struggle with accurately delineating tumor edges. Such errors may result in healthy brain tissue being misclassified as tumor or portions of the tumor being misidentified as healthy tissue. This issue arises from the variable size of brain tumors, their similarity to healthy brain tissue, and their unpredictable spatial locations.

Based on the above discussion, the present study introduces a framework that combines supervised learning and Deep RL for the segmentation and refinement of brain tumor MRI samples. In this approach, an initial segmentation is first obtained using a U-Net model, which is then refined using a Deep Q-Network (DQN).

The remainder of this paper is organized as follows: In Section 3, Materials and Proposed Framework are presented. Section 4 provides details of the proposed framework. Section 5 describes the evaluation metrics and comparisons, and Section 6 discusses the strengths and weaknesses of the proposed framework as well as directions for future research.

3. Materials and Components

3.1. Dataset

The Figshare Brain Tumor Dataset [26] consists of T1-weighted contrast-enhanced MRI samples. This dataset includes 3,064 samples collected from a total of 233 patients with brain tumors by Jun Cheng and colleagues. The images have dimensions of 512×512 and 256×256, with a slice thickness of 6 mm and an inter-slice gap of 1 mm, and are represented in grayscale. For each sample, a segmentation mask and the tumor type were provided by radiology experts. This dataset has been used in various studies for brain tumor segmentation and classification, including [13, 25, 27, 28]. It comprises 1,426 Glioma samples (89 patients), 708 Meningioma samples (82 patients), and 930 Pituitary tumor samples (62 patients). Glioma refers to a type of tumor that originates from glial cells, which are the supportive cells of neurons in the central nervous system. Meningioma is a type of brain tumor arising from the protective membranes surrounding the brain and spinal cord (meninges) and is mostly benign. Pituitary tumors originate from the pituitary gland, located at the base of the brain. An example of the images from this dataset is shown in Figure 2.

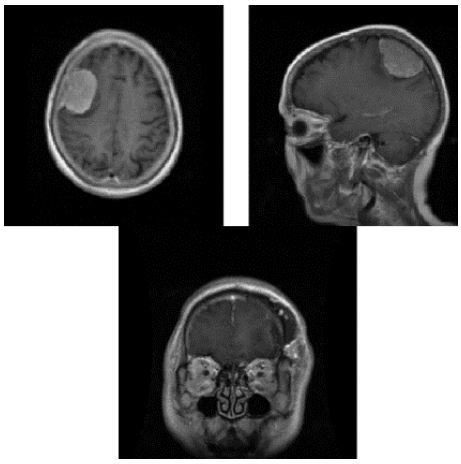

Figure 2. Examples of images from the Figshare Brain Tumor dataset.

The dataset used in the present study, unlike similar datasets such as BraTS, consists of a single channel, which makes its use less challenging. In other words, datasets like BraTS contain a large volume of image dimensions, which increases the complexity of processing and utilization.

3.2. Data Preprocessing and Preparation

In the present study, several sequential steps were performed to preprocess and prepare the dataset. First, the image dimensions were resized to 256×256, and the pixel intensities were scaled to the range [0, 1]. Subsequently, the images were normalized using (1).

$$x\_norm = \frac{x - \mu}{\delta} \quad (1)$$

In (1), μ and δ denote the pixel mean and standard deviation, respectively. During preprocessing, the images were normalized using the mean (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225) vectors. Following this, the dataset of 3,064 images was partitioned into an 80/20 split. Specifically, 80% of the data (2,451 images) was allocated for training and validation, while the remaining 20% (613 images) was strictly reserved as an independent test set for performance evaluation.

3.3. Components of DURL-Net

The proposed framework in the present study consists of two components: a pre-trained U-Net [29] and a Deep Q-Network (DQN) [30]. The U-Net model is an encoder-decoder architecture designed for semantic segmentation of medical (and sometimes non-medical) images. The DQN algorithm, on the other hand, is a reinforcement learning approach that combines deep neural networks with sequential decision-making to solve problems based on the Markov Decision Process (MDP). In the following sections, detailed explanations of each component and their respective functionalities are provided.

3.3.1. EfficientNet-B7 U-Net

The EfficientNet-B7 U-Net architecture (Figure 3) consists of two main components connected by a bottleneck, which serves as the common linking point. It is worth noting that the EfficientNet-B7 U-Net model used in the present study was pre-trained on the ImageNet dataset [31].

The encoder component is responsible for reducing the dimensionality of the input and extracting local image features using convolutional operations. In other words, this component identifies features within the image (such as edges, shapes, and textures) and represents the image as a set of numerical feature maps. In the proposed framework, the encoder module of the U-Net architecture, unlike the traditional version, is based on the EfficientNet-B7 architecture [32]. EfficientNet is a CNN-based architecture introduced in the field of computer vision with the goal of achieving a balance between accuracy, computational complexity, and memory usage. The EfficientNet family consists of eight versions, labeled from B0 to B7. The main differences among these versions lie in the model's depth, number of layers, input image resolution, and the number of parameters.

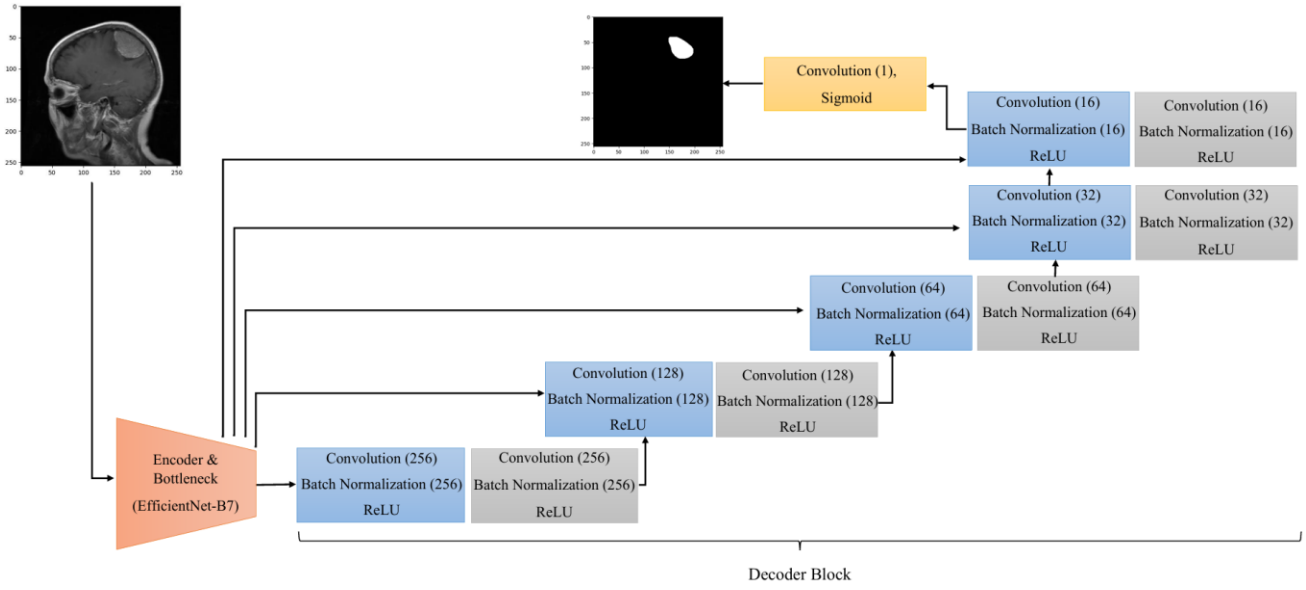

Figure 3. The structure of the EfficientNet-B7 U-Net in the proposed framework.

The next component, which serves as the core section of the U-Net architecture, is known as the bottleneck. In this component, the extracted features from the image reach their highest level of abstraction and become ready for use by the decoder module. Typically, this component consists of two or three convolutional layers accompanied by activation functions, and its output either maintains the same spatial dimensions or undergoes minimal spatial reduction.

The decoder component is responsible for generating the final mask using transposed convolution operations and activation functions. In other words, this component increases the spatial dimensions of the features extracted by the previous modules and produces the final tumor segmentation mask.

As described earlier, the bottleneck component receives the features extracted by the encoder and produces highly abstract representations, while the decoder component utilizes these features to generate the final brain tumor mask. However, due to the considerable number of layers in the U-Net architecture, the decoder may lose some important information—such as tumor edges, tissue structures, and precise spatial details—during the reconstruction process, resulting in a segmentation mask with limited detail. To address this issue, the features extracted at each stage of the encoder are transferred to their corresponding layers in the decoder. This enables the simultaneous use of both low-level and high-level features during the mask generation process. In other words, by combining fine-grained and abstract feature representations, the final segmentation mask is reconstructed with richer and more precise details.

3.3.2. Reinforcement Learning

Reinforcement Learning (Figure 4) is a branch of machine learning based on the interaction between an agent and its environment. The agent selects and performs actions within the current state of the environment, receiving a reward or punishment in response. Its objective is to learn a policy that maximizes the cumulative rewards obtained over time.

Unlike supervised or unsupervised learning, reinforcement learning does not rely on predefined labels or patterns; instead, the agent learns through a process of trial and error [33–36].

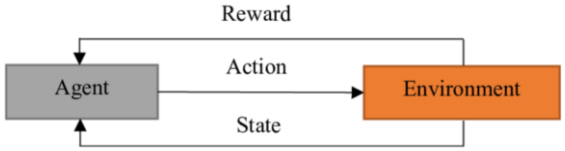

Figure 4. The structure of Reinforcement Learning [37].

The most common framework for reinforcement learning is the Markov Decision Process (MDP), which is defined as shown in (2).

$$MDP = (S, A, P, R, \gamma) \quad (2)$$

In (2), S represents the state space, A denotes the action space, P is the transition probability function from state s to s' given action a (also denoted as $P(s'|s,a)$), R is the reward function for taking action a and transitioning from state s to s', and γ is the discount factor (a value between 0 and 1) used to weigh future rewards relative to immediate rewards, reflecting the agent's consideration of long-term outcomes.

3.3.3. Deep Reinforcement Learning

Deep reinforcement learning (Figure 5) refers to a branch of machine learning that combines automatic feature extraction through deep learning with sequential decision-making based on reinforcement learning and neural networks. In other words, deep reinforcement learning integrates two components: deep learning, for extracting features and modeling complex patterns, and reinforcement learning, for sequential decision-making and interaction with the environment. As explained earlier, reinforcement learning aims to employ an agent to solve sequential decision-making problems. The agent, upon receiving a state, attempts to select the best action (policy) that maximizes the cumulative reward.

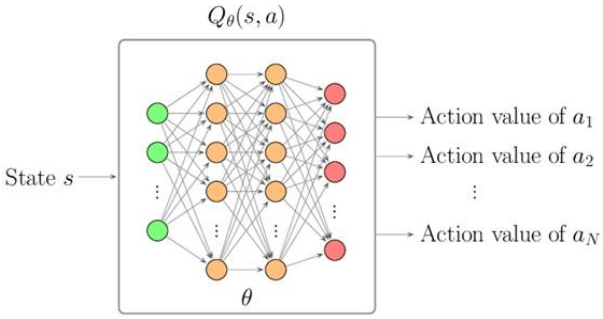

Figure 5. Structure of deep reinforcement learning [41]

Reinforcement learning-based approaches are typically used to solve problems with low complexity and predefined, discrete features. In tasks such as image segmentation, the complexity is high, and defining predetermined features is challenging. Additionally, traditional reinforcement learning methods struggle to handle problems involving continuous states, requiring the transformation of continuous state values into discrete numbers. This approach leads to the loss of many features and properties related to the

problem. Manually defining features reduces the generalization capability of the model. Therefore, by combining reinforcement learning with neural networks, it is possible to create a structure with better generalization ability, capable of processing larger and continuous state spaces, handling complex problems with long-term dependencies, and receiving raw inputs for automatic feature extraction.

3.3.4. Deep Q-Network

The DQN algorithm is a model-free, off-policy deep reinforcement learning method designed to handle continuous state spaces and estimate the value of discrete actions in a given state. The algorithm was first introduced in studies [30, 42] to train models for playing Atari games.

Figure 6 illustrates the training steps of an agent based on the DQN algorithm.

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$
For episode = 1, M do
  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\varphi_1 = \varphi(s_1)$
  For t = 1, T do
    With probability ε select a random action a_t
    otherwise select $a_t = \text{argmax}_a\, Q(\varphi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\varphi_{t+1} = \varphi(s_{t+1})$
    Store transition $(\varphi_t, a_t, r_t, \varphi_{t+1})$ in D
    Sample random minibatch of transitions $(\varphi_j, a_j, r_j, \varphi_{j+1})$ from D
    Set

$$y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \max_{a'} \hat{Q}(\varphi_{j+1}, a'; \theta^-) & \text{otherwise} \end{cases}$$

    Perform a gradient descent step on
      $(y_j - Q(\varphi_j, a_j; \theta))^2$
    with respect to the network parameters θ
    Every C steps reset $\hat{Q} = Q$
  End For
End For

Figure 6. Steps of the DQN algorithm.

The training approach of DQN, like other deep reinforcement learning methods, consists of several components. The first component is the Target Network. In this algorithm, besides the main decision-making network, which estimates the value for each state-action pair, there is an additional neural network used to calculate the target value in the network's update equation. The target network helps stabilize training and prevent instability caused by rapid changes in value estimates. Additionally, after a fixed number of steps, the target network is updated by replacing it with a copy of the current main network, allowing training to continue.

Another component, which is one of the reasons for choosing the current algorithm in this research, is the experience storage module. This component, also known as Experience Replay or Replay Memory, is essentially a queue that stores experiences, including the current state, the action taken, the reward received, and the next state. During training, the agent samples batches of experiences from this memory to update and improve its learning process.

According to the algorithm shown in Figure 6, first, an experience replay buffer named D with size N is defined, and then a deep neural network with a desired structure, named Q, is initialized. It should be noted that at the beginning of the learning process, the target network (denoted as Q') is an identical copy of the main network. After defining these components, the agent interacts with the environment and learns the state-action mapping over M episodes (each episode spans from the start to the end of the interaction with the environment).

At each time step within an episode (until the episode ends), an action is selected randomly with probability ε or the action with the highest estimated value with probability $1-\varepsilon$, and then applied to the current environment. Here, ε is a variable that determines the probability of exploration versus exploitation at each time step, a strategy known as Epsilon-Greedy. After the state changes due to the selected action, a reward and the next state are assigned to the agent, and these experiences are stored in the replay buffer for use in subsequent episodes.

Next, a batch of experiences is randomly sampled from the buffer, and the final output is computed using (3).

$$y_j = r_j + \max_{a'} Q'(s_{j+1}, a_j; \theta^-) \quad (3)$$

To use (3), first the next state (at time step $j+1$) and the reward resulting from the action taken at time j are obtained. Then, using the target network, the maximum estimated value across all possible actions is calculated and combined with the reward. It should be noted that if the environment terminates at time step $j+1$, the output depends solely on the received reward. After calculating the value at step j, the error of the current model's performance is computed using (4).

$$Loss = (y_j - Q(s_j, a_j; \theta))^2 \quad (4)$$

After calculating the error, the parameters of the decision-making network are updated using optimization algorithms such as Gradient Descent. It is important to note that (3) utilizes the target network, while (4) employs the decision network. In other words, the target network serves to evaluate the model's performance over time. Additionally, it should be noted that after every C time steps, the target network is replaced with the decision network.

Considering the steps and structure of the DQN algorithm, it can be observed that implementing and training an agent based on it involves relatively low cost, and due to its well-designed architecture, it benefits from high generalizability.

3.3.5. Morphological Operations

Morphological operations are fundamental tools in image processing that operate based on the shape and structure of image components. These operations are typically applied to binary or grayscale images for enhancement or correction [43]. The key element of these operations is the use of a structuring element (functionally similar to a kernel in convolution operations), which is placed over each position in the image and, based on the neighborhood of pixels, determines whether the central pixel should be retained or removed.

Most morphological operations are based on two fundamental operators called erosion and dilation. The erosion operator (5) reduces the size of objects in an image. When the structuring element is applied to a pixel, if all neighboring pixels covered by the structuring element are white, the central pixel remains white; otherwise, it is set to the background (black). In other words, this operator selects the minimum value within the structuring element. As a result, object boundaries move inward, and small objects are removed. Essentially, erosion shrinks bright regions and enlarges dark regions.

$$(f\ !\ \ b)(x,y) = \min_{(s,t)\in B}[f(x+s,y+t) - b(s,t)] \qquad (5)$$

The dilation operator (6) performs the opposite function of erosion. If any of the neighboring pixels within the structuring element are white, the central pixel is set to white; otherwise, it becomes black. In other words, this operator selects the maximum value within the structuring element. This process enlarges objects, fills gaps, and brings separated components closer together.

$$(f \oplus b)(x,y) = \max_{(s,t)\in B}[f(x-s,y-t) + b(s,t)] \qquad (6)$$

4. DURL-Net

4.1. Framework Overview

The proposed framework in the current study consists of two components for brain tumor segmentation. It is worth noting that this framework is not limited to brain tumor data and can also be applied to other disorders or segmentation tasks in different domains. The main idea behind designing this framework is to correct errors in brain tumor segmentation using deep reinforcement learning. In other words, supervised approaches, while less computationally intensive compared to deep reinforcement learning methods, offer better accuracy and performance than traditional shallow machine learning or unsupervised methods. However, these approaches still suffer from limitations, such as errors along the tumor boundaries. Such errors can have irreparable consequences because tumor type classification relies on the segmented tumor in the MRI sample, and inaccuracies in tumor size, shape, or location may lead to incorrect diagnoses. Therefore, in the current study, an initial brain tumor segmentation is performed using a base U-Net architecture, and final refinement of the tumor is achieved through a DQN-based agent combined with morphological operations.

To provide a clear understanding of the proposed framework and ensure reproducibility, the sequential stages of DURL-Net, along with their explicit inputs and outputs, are detailed below:

Initial Segmentation

- Input: Preprocessed T1-weighted contrast-enhanced MRI image (256×256 pixels).
- Output: Initial binary tumor segmentation mask generated by the U-Net model.
- *Visual Illustration*: The initial MRI and the corresponding U-Net segmentation output are shown in Figure 3 (and Figure 8).

Tumor Region Localization (Cropping)

- Input: The original MRI image and the initial segmentation mask.
- Output: Cropped MRI image and cropped segmentation mask, constrained to the bounding box of the predicted tumor region with an additional 10-pixel margin.
- *Visual Illustration*: This cropped tumor region for the same sample is explicitly shown in Figure 8.

Resizing and Patch Gridding

- Input: Cropped MRI and cropped segmentation mask.
- Output: Resized MRI and mask, divided into a grid layout of 16 non-overlapping 8×8 patches.
- *Visual Illustration*: The gridded version of the same sample MRI and initial segmentation is shown in Figure 9.

State Formulation and Agent Interaction

- Input: A 4-channel state constructed at each time step t, comprising the global resized MRI patch, the global resized mask patch, the local 8×8 MRI sub-patch at step t, and the local 8×8 mask sub-patch at step t. These inputs are directly derived from the outputs shown in Figures 8 and 9.
- Output: The DQN agent outputs one of 11 discrete actions. The selected morphological operation is applied to the local mask patch, yielding the refined local mask.
- *Visual Illustration*: The overall structure of this agent interaction, utilizing the exact same sample inputs from Figures 8 and 9, is illustrated in the workflow diagram in Figure 10.

Final Reconstruction

- Input: The sequence of refined 8×8 mask patches generated over the 16 time steps.
- Output: The final refined binary tumor segmentation mask, assembled from the locally processed patches.

4.2. Initial Segmentation using U-Net

To train the EfficientNet-B7 U-Net component (details in Figure 3), the Figshare Brain Tumor dataset was first preprocessed and prepared. The samples were then randomly split into a training set (80%) and a test set (20%). Although a dataset of 3,064 images is relatively modest for deep learning-based medical image segmentation, the risk of overfitting is significantly mitigated by leveraging an EfficientNet-B7 backbone pre-trained on the ImageNet dataset. This transfer learning approach provides robust, generalized feature extractors, reducing reliance on large task-specific datasets and enhancing the robustness of the initial segmentation. Next, the U-Net model was trained for 30 epochs with a batch size of 4, using $\alpha = 0.7$ in the cost function, (9) as the cost function, and a Sigmoid function as the activation for the output layer (mask generator). The Adam optimizer was employed with a learning rate of 10^{-3}, $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = 10^{-8}$, and a weight decay of zero.

$$L_{Dice} = 1 - \frac{1}{N \cdot C} \sum_{n=1}^{N} \sum_{c=1}^{C} \frac{2 \cdot \sum_{i=1}^{H} \sum_{j=1}^{W} G_{n,c,i,j} \cdot P_{n,c,i,j}}{\sum_{i=1}^{H} \sum_{j=1}^{W} (G_{n,c,i,j} + P_{n,c,i,j}) + 10^{-5}} \tag{7}$$

$$L_{BCE} = -\frac{1}{N \cdot C \cdot H \cdot W} \sum_{n=1}^{N} \sum_{c=1}^{C} \sum_{i=1}^{H} \sum_{j=1}^{W} [G_{n,c,i,j} \cdot \log(p_{n,c,i,j}) + (1 - G_{n,c,i,j}) \cdot \log(1 - p_{n,c,i,j})] \tag{8}$$

$$L_{Final} = \alpha \cdot L_{Dice} + (1 - \alpha) L_{BCE} \tag{9}$$

In (7)–(9), G denotes the *Ground Truth* (the correct mask corresponding to the brain tumor), P represents the output generated by the brain tumor segmentation model, H is the image height, W the image width, N the batch size, and C the number of image channels. Equation (7) calculates the segmentation model's error in terms of the *overlap* between the output and the Ground Truth. In other words, this loss function measures the discrepancy in the intersection and union of the predicted and reference masks. It is worth noting that the inclusion of 10^{-5} in the denominator of (7) serves to prevent division by zero. In addition to (7), which quantifies the segmentation error, (8) BCE is incorporated as part of (9). This equation computes the segmentation loss based on individual pixel values. In other words, (9) evaluates the segmentation error considering both the overlap of the tumor region and the associated pixels. This combined approach, compared to using (7) and (8) independently, leads to faster training convergence and improved model performance. Figure 7 illustrates two MRI samples segmented by the proposed model, in which the corresponding tumor regions have been successfully extracted.

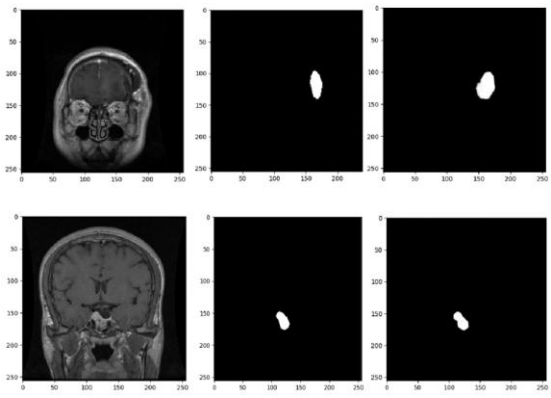

Figure 7. Initial segmentation outputs generated by the U-Net model, highlighting tumor localization and boundary uncertainties.

In Figure 7, the first column (from the left) presents an example MRI scan of a patient, the second column shows the corresponding Ground Truth,

and the third column displays the output produced by the U-Net model. As illustrated, although the model correctly identifies the tumor location, it exhibits inaccuracies in estimating its size and boundaries. After obtaining the initial brain tumor segmentation based on the input MRI and the U-Net model, each pixel value is rounded to the nearest integer. Subsequently, both the initial segmentation map and the input MRI are cropped to the tumor region, extended by a margin of 10 pixels. This modification reduces the computational complexity of the algorithm and facilitates more efficient decision-making by the agent. An example of this process is shown in Figure 8.

4.3. Tumor Region Localization and State Representation

After constraining the MRI region and obtaining the initial segmentation, the resulting image is resized from 256×256 to 32×32. The reason for this resizing is the presence of an experience replay buffer. Following these steps, two of the four channels of the agent's input state are generated. In other words, two of the channels required by the agent correspond to the tumor region in the MRI image and the initial segmentation map. Based on this approach, unlike previous studies, there is no need for a separate mechanism to extract the tumor region.

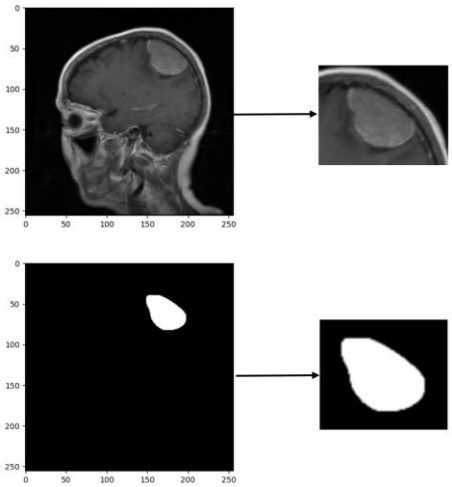

Figure 8. Tumor region localization and cropping process applied to the MRI and initial segmentation mask.

After obtaining the two-channel images representing the agent's state, the constrained MRI and the initial segmentation are divided into patches of size 8×8. In other words, each image is split into 16 smaller patches (corresponding to 16 time steps), and the agent refines and improves the final segmentation by processing each patch at its respective time step. It should be noted that this grid-based division enables the agent to determine how the refinement should be performed within each patch. In other words, by selecting a specific refinement strategy, the agent influences the structure of each patch and, at the end of the episode, outputs the final segmentation mask. As illustrated in Figure 9, the tumor-related region (in both the MRI and the initial segmentation) is transformed into a grid layout.

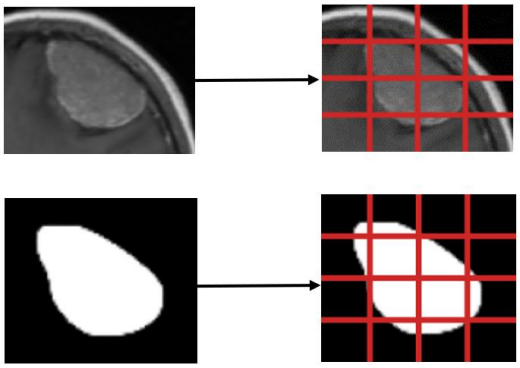

Figure 9. Grid-based partitioning of the localized tumor region into patches for sequential processing by the DQN agent.

After obtaining a grid structure from the aforementioned images, the agent's state at time step t is determined. In summary, the agent's input state can be considered to consist of four components: the MRI data and the initial segmentation (both constrained to the tumor region), and the corresponding patches resulting from the grid division at each time step.

4.4. Action Space

After defining the state structure at each time step, the agent selects an action based on it. In total, there are 11 distinct actions that the agent can choose from at each time step to affect the corresponding patch. Table 1 lists the actions available to the agent.

Table 1. Actions available to the agent for refining each patch

Kernel Size (Structural Element)	Morphological Operators
1	Dilation
1	Erosion
3	Dilation
3	Erosion
5	Dilation
5	Erosion
7	Dilation
7	Erosion
9	Dilation
9	Erosion
NEXT PATCH	

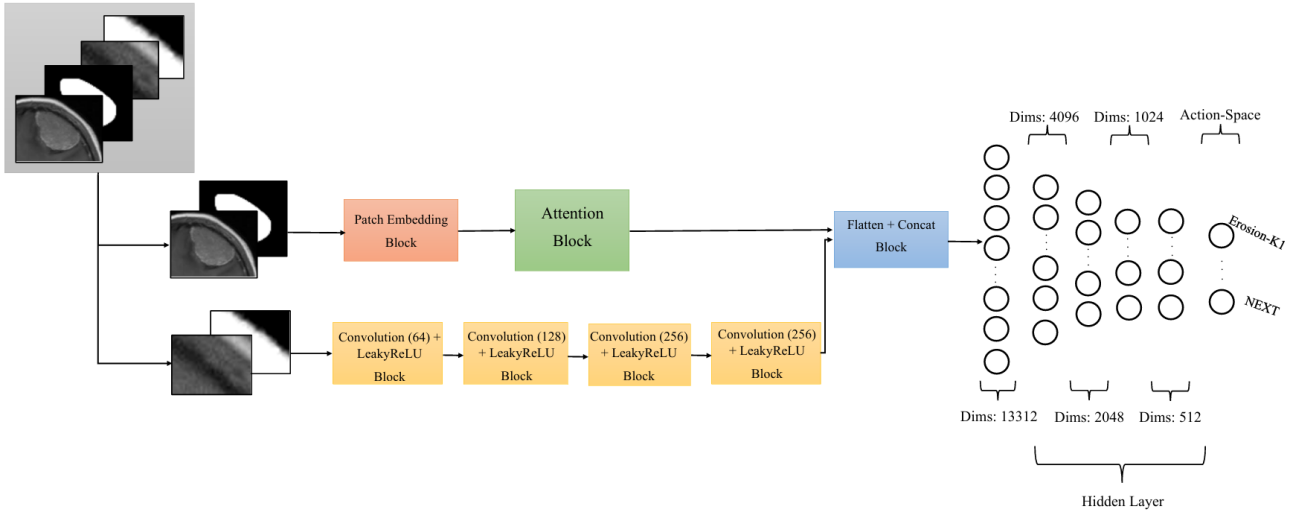

Figure 10. Overall architecture and data flow of the proposed DURL-Net framework.

By selecting any of the actions listed in Table 1, the agent modifies the current patch at a given time step based on the type of morphological operator and the kernel size. In other words, the agent incrementally adjusts a region of the tumor through trial- and- error, receiving rewards or penalties accordingly. It should be noted that one action, termed NEXT PATCH, is also included; when selected, the agent moves to the adjacent patch while leaving the current patch unchanged. The purpose of this action is to allow the agent to skip regions that do not require modification.

4.5. Reward Function

As previously discussed, defining an appropriate reward mechanism is of critical importance. Accordingly, the reward function is computed based on the differences in BCE and DICE between the mask at time step t and that at $t-1$. Using these differences, the agent receives either a reward or a penalty. The structure of the reward function is presented in (10)-(14).

$$BCE_t = -\frac{1}{N}\sum_{i=1}^{N}[G_i \cdot \log(P_{t,i}) + (1-G_i)\cdot \log(1-\log(P_{t,i}))] \quad (10)$$

$$DICE_t = \frac{2\cdot\sum_{i=1}^{N} round(G_i)\cdot round(P_{t,i})}{\sum_{i=1}^{N} round(G_i) + \sum_{i=1}^{N} round(P_{t,i}) + 10^{-5}} \quad (11)$$

$$\Delta BCE = BCE_{t-1} - BCE_t \quad (12)$$

$$\Delta DICE = DICE_t - DICE_{t-1} \quad (13)$$

$$\text{Re}ward_t = (\Delta BCE + \Delta DICE) + (DICE_t - DICE_0) \quad (14)$$

As shown in (10)-(14), the magnitude of the reward or penalty depends on the changes in DICE and BCE. In other words, if the action selected by the agent leads to an improvement in the segmented tumor, the BCE decreases and the DICE increases. In this case, the agent receives a reward; otherwise, it is penalized.

The reason for using BCE and DICE in calculating rewards and penalties is to provide the agent with feedback from the environment for refining tumor segmentation, with the goal of improving both overlap and tumor boundaries. In other words, the agent aims to select the optimal action to enhance the tumor's overlap, edges, and fine details compared to the previous and initial steps. The rationale behind this composite reward design is to provide a dense, informative feedback signal that addresses both pixel-level accuracy (via changes in BCE) and region-level overlap consistency (via changes in the DICE score). By including the difference between the current DICE score and the initial DICE score, the function acts as a reward-shaping mechanism that encourages the agent to maintain or improve upon the initial U-Net segmentation, thereby actively preventing destructive modifications. This step-wise reward formulation mitigates the sparse reward problem commonly encountered in reinforcement learning, which significantly enhances learning stability and promotes steady convergence. Furthermore, this reward structure is highly complementary to the 'NEXT PATCH' action in the action space; by allowing the agent to abstain from modifying already optimal regions, it avoids unnecessary penalties, further stabilizing the training process and preventing policy oscillation. The structure and architecture of the proposed framework are illustrated in Figure 10.

4.6. DQN Agent Architecture

As shown in Figure 10, at each time step, a 4-channel image is provided as the agent's input, comprising the tumor-related region in the MRI,

the initial segmentation (EfficientNet-B7 U-Net output), and the current patch. The deep neural network employed for the agent consists of two feature-extraction pathways with distinct architectures.

Two channels of the input image, representing the MRI patch and the initial segmentation at time step t, are processed through four convolutional layers with LeakyReLU activation functions, enabling feature extraction.

It should be noted that the number of output channels of the layers is 64, 128, 256, and 256, respectively. The use of convolution layers for the two aforementioned channels is due to their lower level of detail compared to the other two channels, making local features sufficient for representation. Additionally, the LeakyReLU activation function is employed to preserve positive values with a slope of 1 and negative values with a slope of 0.01. In other words, the rationale for using LeakyReLU is to prevent vanishing gradient issues (common with functions such as sigmoid) and to avoid setting neuron activations to zero (as can occur with ReLU).

In addition, the two image channels representing the tumor region in the MRI and the initial segmentation are processed using Multi-Head Attention [44] (15)–(18) to extract relevant features. To implement this process, patch embedding is first applied to these two channels, followed by processing with Multi-Head Attention. The purpose of the patch embedding is to apply a convolution layer with 256 channels, enabling the mechanism to process the images effectively. It is worth noting that patch-based feature extraction using an attention mechanism has been previously discussed in [45]. Moreover, the rationale for employing the attention mechanism on these two channels is to allow the model to focus on fine-grained regions within the tumor area, thereby improving the refinement of the initial segmentation.

$$Multihead(Q,K,V) = Concat(head_1, head_2, ..., head_h)W^O \quad (15)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (16)$$

$$Attention(Q,K,V) = Soft\max(\frac{QK^T}{\sqrt{d^k}})V \quad (17)$$

$$Soft\max(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (18)$$

After feature extraction via the Multi-Head Attention and convolution components, the resulting feature matrix is flattened into a vector and passed through a module composed of dense layers and an output layer (19) to determine the action corresponding to the current state. In other words, feature extraction from the vector obtained from these components is further refined by the dense layers, and ultimately, the action appropriate for the input state is selected.

$$y = f(Wx + b) \quad (19)$$

In (19), x denotes the input vector, W the weight matrix, b the bias term, and f the activation function.

4.7. Training Procedure and Hyperparameters

For training the DQN-based agent, the following hyper-parameters were used: 8 attention heads, a discount factor of 1.0, an experience replay buffer of 100,000, weight decay of 10^{-14} (to prevent over-fitting), 3 million training time steps, a batch size of 32, a 3,000-step interval for updating the target network, an exploration fraction of 0.05, and the Adam optimizer with a learning rate of $2 * 10^{-5}$, $\beta 1 = 0.9$, and $\beta 2 = 0.999$.

5. Experimental Results

5.1. Evaluation Metrics

Evaluation metrics are among the most important components in the design and development of a machine learning framework across all domains. By employing appropriate metrics, the performance of models and potential errors can be assessed without causing irreversible consequences, particularly in critical tasks such as brain tumor segmentation and refinement. In this study, the evaluation of DURL-Net was conducted using the following metrics: IoU, DICE, Kappa, Sensitivity, and Specificity.

5.1.1 IoU

One of the fundamental metrics in image segmentation is the Intersection over Union (IoU), also known as the Jaccard Index (20). This metric measures the degree of overlap between two regions. In the context of brain tumor segmentation, these two regions correspond to the ground truth and the predicted segmentation.

$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \quad (20)$$

In (20), TP refers to True Positives, FP to False Positives, and FN to False Negatives.

5.1.2 DICE

Another commonly used metric in segmentation is the DICE Similarity Coefficient (21), or simply DICE. It should be noted that this metric is generally more sensitive than IoU and is therefore

often used as a reference metric in medical image segmentation.

$$DICE(A,B)=\frac{2|A\cap B|}{|A|+|B|}=\frac{2\cdot TP}{2\cdot TP+FP+FN} \quad (21)$$

5.1.3 Kap

Another metric used to evaluate the performance of the segmentation and refinement model is Cohen's Kappa (22)–(24), abbreviated as Kap. Although IoU and DICE have straightforward mathematical formulations and provide meaningful descriptions, they do not adequately account for the background. In other words, in segmentation tasks, most regions in the segmentation mask consist of black pixels, and these metrics tend to focus primarily on pixels with values closer to one. To address this limitation, the Kap metric also considers True Negatives, providing a more balanced evaluation.

$$k=\frac{p_o-p_e}{1-p_e} \quad (22)$$

$$p_o=\frac{TP+TN}{N} \quad (23)$$

$$p_e=\left(\frac{(TP+FN)\cdot(TP+FP)}{N^2}\right)+\left(\frac{(TN+FP)\cdot(TN+FN)}{N^2}\right) \quad (24)$$

In (22)-(24), N denotes the total number of pixels in the image, p_o the observed agreement, and p_e the expected agreement.

5.1.4 Sensitivity

One of the fundamental metrics for evaluating the performance of machine learning models and brain tumor segmentation is Sensitivity (25). This metric, also known as the True Positive Rate or Recall, measures the proportion of positive pixels (pixels belonging to the brain tumor or target object) that are correctly predicted by the segmentation model.

$$Sensitivity=\frac{TP}{TP+FN} \quad (25)$$

5.1.5 Specificity

Similar to Sensitivity, another metric is Specificity, also known as the True Negative Rate. This metric measures the proportion of negative pixels (pixels belonging to the background) that are correctly identified by the model.

$$Specificity=\frac{TN}{TN+FP} \quad (26)$$

5.2. Obtained Results

During the training of U-Net (to obtain the initial segmentation), considering the dataset and relevant details, the plots of the average loss and DICE score are obtained, as shown in Figure 11.

According to the plots in Figure 11, the training steps of the model exhibit significant stability.

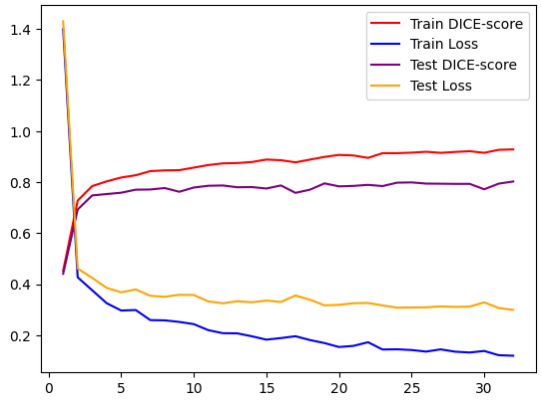

Figure 11. Training progress of the U-Net component, illustrating loss convergence and DICE score improvement over epochs.

In other words, the test data loss, along with the training data loss, decreased steadily over time. This trend is also reflected in the DICE score, which increased consistently.

After training, the model achieved the following performance on the training data: 92.99% DICE, 87.59% IoU, 90.71% Sensitivity, 99.3% Specificity, and 92.88% Kappa. On the test data, the results were 80.37% DICE, 71.70% IoU, 70.49% Sensitivity, 98.87% Specificity, and 80.14% Kappa, indicating the solid performance of the baseline U-Net model on brain tumor MRI samples.

After refining the initial brain tumor segmentation using the DQN-based agent (DURL-Net), the metrics on the test data improved to 86.73% DICE, 78.68% IoU, 87.68% Sensitivity, 96.06% Specificity, and 85.21% Kappa. These results demonstrate that all evaluated metrics improved following tumor refinement, except specificity and sensitivity. Notably, these outcomes, alongside results reported in previous studies, are summarized in Table 2.

Table 2. Comparative performance evaluation of DURL-Net against baseline and state-of-the-art methods.

Method	Metrics				
	Specificity (%)	Sensitivity (%)	Kap (%)	IoU (%)	DICE (%)
DURL-Net	96.06	87.68	85.21	78.68	86.73
U-Net	98.87	70.49	80.14	71.70	80.37
DGARL [25]	99.72	87.72	84.75	77.51	85.02
[13]	-	54.00	-	-	62.00

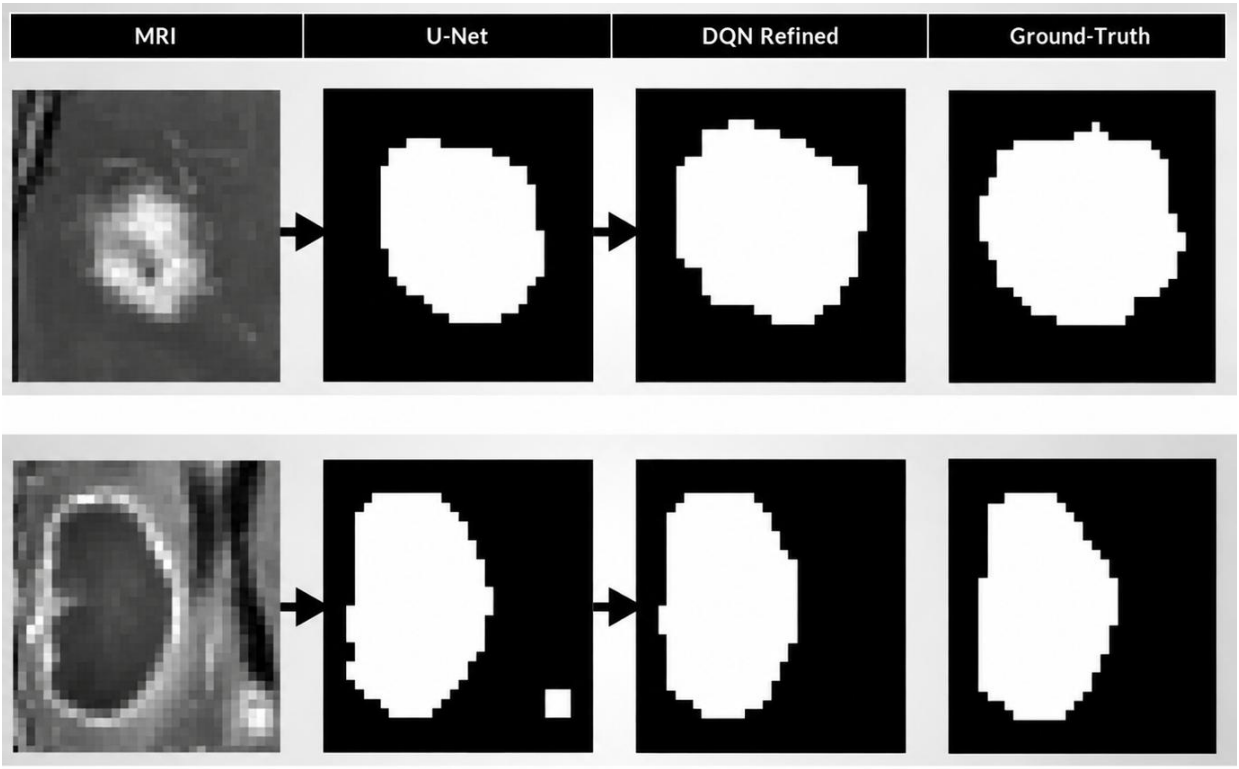

Figure 12. Qualitative comparison of two representative test samples. From left to right: Original MRI, Initial U-Net segmentation, Final DURL-Net refinement, and Ground Truth.

As shown in Table 2, the proposed framework (DURL-Net) demonstrates improvements across four evaluation metrics compared to the baseline segmentation model (U-Net). Furthermore, in comparison with previous models, DURL-Net exhibits higher DICE, IoU, and Kappa scores; however, Sensitivity and Specificity remain higher in some of the other models. This can be attributed to the ability of those models to predict a smaller number of pixels as either background or tumor. Notably, DURL-Net improves Sensitivity relative to U-Net, indicating its accurate performance. In other words, the performance of DURL-Net is dependent on the baseline segmentation model. Therefore, it can be concluded that applying a DQN-based agent for refining prior models, along with potential structural modifications, is expected to further enhance the agent's performance.

5.3. Qualitative Analysis and Visual Comparison

To provide a qualitative assessment of the proposed framework's performance, particularly its ability to refine tumor boundaries, we present visual comparisons of the segmentation results. Although direct visual comparisons with recent state-of-the-art models (such as DGARL [25] or 3D multi-modal architectures) are not feasible due to the unavailability of their source code and pre-trained weights (our quantitative comparisons with them are based strictly on the results reported in their respective published articles), we can clearly demonstrate the refinement capability of DURL-Net by comparing its final outputs with the baseline EfficientNet-B7 U-Net.

Figure 12 illustrates two representative MRI samples from the test set. For each sample (displayed in separate rows), the columns display the original MRI scan, the initial segmentation generated by the EfficientNet-B7 U-Net (before DQN refinement), the final refined segmentation produced by DURL-Net, and the corresponding Ground Truth mask.

As observed in Figure 12, the initial U-Net model, while successfully localizing the tumor, exhibits noticeable inaccuracies along the tumor boundaries, such as irregular edges and slight over- or under-segmentation. However, as shown in Figure 12, the integration of the DQN agent effectively corrects these boundary uncertainties. By adaptively applying morphological operations patch-by-patch, DURL-Net smooths the irregular edges and aligns the predicted mask more closely with the ground truth, thereby validating the visual improvement in segmentation quality achieved by the proposed framework.

5.4. Ablation Study

To comprehensively evaluate the proposed framework and assess the individual contributions

of its core components—namely the EfficientNet-U-Net, the DQN agent, and morphological $B^{\vee}$ operations—we analyze the performance differences between the baseline supervised model and the fully proposed framework.

First, the contribution of the EfficientNet-B7U-Net is isolated by evaluating the model without any reinforcement learning-based refinement. As shown in Table 2 the baseline U-Net achieves a , DICE of 80.37% and an IoU of 71.70%. While it successfully localizes the tumor, it suffers from ,boundary uncertainties and irregular shapes highlighting the limitations of relying solely on a supervised mapping paradigm for precise boundary delineation.

Second, to understand the role of morphological operations and the DQN agent, we compare the baseline U-Net with the full DURL-Net. In traditional image processing pipelines, fixed morphological operations (e.g., standard closing or opening with a static kernel) are often applied as ,post-processing to smooth boundaries. However such static operations lack adaptability; they frequently lead to over-smoothing, loss of fine tumor details, or unintended dilation/erosion of the tumor mass, which can ultimately degrade the overall segmentation metrics. In contrast, DURL-Net integrates a DQN agent that adaptively selects the optimal morphological operation (dilation or erosion) and kernel size (ranging from1 to 9 for (each localized patch, or chooses to skip the patch entirely (via the NEXT PATCH action). This adaptive decision-making prevents the over-correction and detail-loss issues inherent in fixed post-processing.

As demonstrated in Table 2 the integration of the , DQN agent and morphological operations in DURL-Net significantly improves the segmentation quality, achieving a DICE of 86.73% and an IoU of 78.68%. The substantial performance gap between the baseline U-Net and DURL-Net (a 6.36% ,increase in DICE 6.98% in IoU, and a 17.19% (improvement in Sensitivity explicitly validates the critical contribution of the DQN agent. By dynamically refining the tumor boundaries using morphological operations on a patch-by-patch basis, the DQN agent effectively overcomes the high uncertainty errors of the initial supervised segmentation, proving that the adaptive selection of morphological actions is essential for achieving the framework's superior performance.

6. Conclusions

In the current study, the problem formulation, previous research in brain tumor segmentation, prior approaches, the proposed framework, and its results are presented. One of the innovations of the proposed framework is the integration of deep reinforcement learning with U-Net as a supervised approach for brain tumor segmentation. According to recent studies, only a few works have addressed the refinement of brain tumor segmentation. In other words, most prior studies have focused on complete brain tumor segmentation using either supervised methods or deep reinforcement learning approaches; however, these methods—especially those based on deep reinforcement learning—have suffered from instability, errors along tumor edges and fine details, and considerable computational complexity. DURL-Net first performs initial brain tumor segmentation using U-Net and then refines the segmented tumor with a DQN-based agent to improve the results. Moreover, unlike prior approaches, the current framework does not require a separate, independent pipeline to extract the tumor region. Instead, tumor localization is an integrated, sequential step where the initial segmentation itself provides direct access to the relevant region, and subsequent lightweight cropping merely focuses the refinement agent's computational attention. Similar to other approaches in the literature, the proposed framework has both strengths and limitations. Among its limitations are the time required to train the DQN agent and the memory needed to store experiences from previous time steps. Another limitation is the dependency of the refinement model’s performance on the baseline supervised segmentation model. In other words, if the initial segmentation model performs poorly, the refinement agent will also be negatively affected, leading to decreased overall performance. Additionally, while the use of a pre-trained backbone mitigates overfitting, the moderate size of the dataset (3,064 images) remains a constraint. Future work will incorporate k-fold cross-validation and evaluate the framework on larger, multi-institutional datasets to further validate its robustness and generalizability. Deep reinforcement learning is inherently prone to instability. Therefore, it has been employed in this study specifically for refining segmented brain tumors. However, to enhance stability and learning speed in DQN, a Prioritized Experience Replay (PER) buffer can be utilized. Additionally, to evaluate potential improvements or declines in model performance, on-policy deep reinforcement learning algorithms can also be considered.

References

[1] M. Price, C. Ballard, J. Benedetti, C. Neff, G. Cioffi, K. A. Waite, C. Kruchko, J. S. Barnholtz-Sloan, and Q.

T. Ostrom, "CBTRUS statistical report: Primary brain and other central nervous system tumors diagnosed in the United States in 2017–2021," *Neuro-Oncology*, vol. 26, suppl. 6, pp. vi1–vi85, Oct. 2024.

[2] Brain tumor, Mayo Clinic, 2024. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084

[3] Verywell Health, Brain Tumor Facts and Statistics, 2023. [Online]. Available: https://www.verywellhealth.com/brain-tumor-facts-and-statistics-what-you-need-to-know-5524668

[4] E. Moitra, "Brain tumor," in Encyclopedia of Clinical Neuropsychology. New York, NY: *Springer*, 2011, pp. 443–444.

[5] E. S. Biratu, F. Schwenker, Y. M. Ayano, and T. G. Debelee, "A survey of brain tumor segmentation and classification algorithms," *Journal of Imaging*, vol. 7, no. 9, Art. no. 179, Sep. 2021.

[6] Z. Atha and J. Chaki, "SSBTCNet: Semi-supervised brain tumor classification network," *IEEE Access*, vol. 11, pp. 141485–141499, Dec. 2023.

[7] F. Fakouri, M. Nikpour, and A. Soleymani Amiri, "Automatic brain tumor detection in brain MRI images using deep learning methods," *Journal of AI and Data Mining*, vol. 12, no. 1, pp. 27–35, Jan. 2024.

[8] A. Bal, M. Banerjee, P. Sharma, and M. Maitra, "Brain tumor segmentation on MR image using K-Means and fuzzy-possibilistic clustering," in *2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, May 2018, pp. 1–8.

[9] R. Shanker, R. Singh, and M. Bhattacharya, "Segmentation of tumor and edema based on K-mean clustering and hierarchical centroid shape descriptor," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Nov. 2017, pp. 1105–1109.

[10] N. Kaur and M. Sharma, "Brain tumor detection using self-adaptive K-means clustering," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, India, Aug. 2017, pp. 1861–1865.

[11] E. Abdel-Maksoud, M. Elmogy, and R. Al-Awadi, "Brain tumor segmentation based on a hybrid clustering technique," *Egyptian Informatics Journal*, vol. 16, no. 1, pp. 71–81, Mar. 2015.

[12] M. A. Almahfud, R. Setyawan, C. A. Sari, D. R. I. M. Setiadi, and E. H. Rachmawanto, "An effective MRI brain image segmentation using joint clustering (K-Means and Fuzzy C-Means)," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, Indonesia, Nov. 2018, pp. 11–16.

[13] C. J. Sheela and G. Suganthi, "Automatic brain tumor segmentation from MRI using greedy snake model and fuzzy C-means optimization," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 3, pp. 557–566, Mar. 2022.

[14] R. Ayachi and N. Ben Amor, "Brain tumor segmentation using support vector machines," in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 10th European Conference (ECSQARU 2009)*, Verona, Italy, Jul. 2009, pp. 736–747.

[15] N. Zhang, S. Ruan, S. Lebonvallet, Q. Liao, and Y. Zhu, "Multi-kernel SVM based classification for brain tumor segmentation of MRI multi-sequence," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, Nov. 2009, pp. 3373–3376.

[16] W. Chen, X. Qiao, B. Liu, X. Qi, R. Wang, and X. Wang, "Automatic brain tumor segmentation based on features of separated local square," in *2017 Chinese Automation Congress (CAC)*, Jinan, China, Oct. 2017, pp. 6489–6493.

[17] A. Pinto, S. Pereira, H. Dinis, C. A. Silva, and D. M. Rasteiro, "Random decision forests for automatic brain tumor segmentation on multi-modal MRI images," in *2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG)*, Porto, Portugal, Feb. 2015, pp. 1–5.

[18] Q. Nida-Ur-Rehman, I. Ahmed, G. Masood, M. Khan, and A. Adnan, "Segmentation of brain tumor in multimodal MRI using histogram differencing & KNN," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, pp. 72–79, 2017.

[19] F. Sahba, H. R. Tizhoosh, and M. M. A. Salama, "A reinforcement learning framework for medical image segmentation," in *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, Jul. 2006, pp. 511–517.

[20] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1240–1251, May 2016.

[21] J. Walsh, A. Othmani, M. Jain, and S. Dev, "Using U-Net network for efficient brain tumor segmentation in MRI images," *Healthcare Analytics*, vol. 2, Art. no. 100098, Nov. 2022.

[22] C. C. Olisah and S. Van Cauter, "SEDNet: Shallow Encoder Decoder Network for Brain Tumor Segmentation," in *2024 14th International Conference on Pattern Recognition Systems (ICPRS)*, Jul. 2024, pp. 1–8.

[23] B. K. Kalejahi, S. Meshgini, and S. Danishvar, "Segmentation of brain tumor using a 3D generative adversarial network," *Diagnostics*, vol. 13, no. 21, Art. no. 3344, Oct. 2023.

[24] X. Liao, W. Li, Q. Xu, X. Wang, B. Jin, X. Zhang, Y. Wang, and Y. Zhang, "Iteratively-refined interactive 3D medical image segmentation with multi-agent

reinforcement learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9394–9402.

[25] C. Xu, T. Zhang, D. Zhang, D. Zhang, and J. Han, "Deep generative adversarial reinforcement learning for semi-supervised segmentation of low-contrast and small objects in medical images," *IEEE Transactions on Medical Imaging*, vol. 43, no. 9, pp. 3072–3084, Sep. 2024.

[26] J. Cheng, W. Huang, S. Cao, R. Yang, W. Yang, Z. Yun, Z. Wang, and Q. Feng, "Enhanced performance of brain tumor classification via tumor region augmentation and partition," *PLOS ONE*, vol. 10, no. 10, Art. no. e0140381, Oct. 2015.

[27] S. Deepak and P. M. Ameer, "Brain tumor classification using deep CNN features via transfer learning," *Computers in Biology and Medicine*, vol. 111, Art. no. 103345, Aug. 2019.

[28] Z. N. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, S. Ahmed, and J. Lu, "Brain tumor classification for MR images using transfer learning and fine-tuning," *Computerized Medical Imaging and Graphics*, vol. 75, pp. 34–46, Jul. 2019.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Munich, Germany, Oct. 2015, pp. 234–241.

[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, Dec. 2013. [Online]. Available: https://arxiv.org/abs/1312.5602

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 248–255.

[32] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Long Beach, CA, USA, May 2019, pp. 6105–6114.

[33] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[34] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: John Wiley & Sons, 2014.

[35] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 16, no. 1, pp. 1–118, Jan. 2023.

[36] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, Jul. 1966.

[37] *Balancing Excitation and Inhibition of Spike Neuron Using Deep Q Network (DQN)*, ResearchGate, [Online]. Available: https://www.researchgate.net/

[38] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[39] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3–4, pp. 219–354, Dec. 2018.

[40] S. S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: An overview," in *Proceedings of the SAI Intelligent Systems Conference (IntelliSys 2016)*, London, UK, Sep. 2016, pp. 426–440.

[41] *Deep reinforcement learning approach to MIMO precoding problem: Optimality and Robustness*, ResearchGate, [Online]. Available: https://www.researchgate.net/

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[43] D. Chudasama, T. Patel, S. Joshi, and G. I. Prajapati, "Image segmentation using morphological operations," *International Journal of Computer Applications*, vol. 117, no. 18, pp. 16–19, Jan. 2015.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, and J. Uszkoreit, "An image is worth 16×16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, Oct. 2020. [Online]. Available: https://arxiv.org/abs/2010.11929.

DURL-Net: یکپارچه‌سازی U-Net EfficientNet-B7 و شبکه Q عمیق برای بخش‌بندی تومور مغزی و پالایش مورفولوژیکی تطبیقی

امید خلف بیگی[۱] و سید علیرضا بشیری موسوی[۲،*]

[۱] گروه مهندسی برق و کامپیوتر، دانشگاه خوارزمی، تهران، ایران.

[۲] مرکز آموزش عالی فنی و مهندسی بوئین زهرا، گروه مهندسی برق و کامپیوتر، بوئین زهرا، قزوین، ایران.

ارسال ۲۰۲۶/۰۳/۰۸؛ بازنگری ۲۰۲۶/۰۶/۳۰؛ پذیرش ۲۰۲۶/۰۷/۱۷

چکیده:

تومور مغزی یکی از جدی‌ترین و خطرناک‌ترین بیماری‌های مغزی است که حیات فرد را تهدید کرده و می‌تواند تأثیر عمیقی بر زندگی او بگذارد. بر این اساس، مطالعه حاضر به چالش پالایش بخش‌بندی تومور مغزی بر اساس داده‌های تصویربرداری تشدید مغناطیسی (MRI) و یادگیری تقویتی عمیق می‌پردازد. اگرچه رویکردهای مبتنی بر یادگیری نظارت‌شده عملکرد قابل‌قبولی در بخش‌بندی و مکان‌یابی تومور نشان داده‌اند، اما اغلب در امتداد مرزهای تومور با خطاهای بالای عدم قطعیت مواجه هستند. در این پژوهش، یک چارچوب یادگیری که یک مدل نظارت‌شده را با یادگیری تقویتی عمیق ترکیب می‌کند و DURL-Net نامیده می‌شود، به‌منظور بخش‌بندی و پالایش پیشنهاد شده است. به‌طور مشخص، این چارچوب ابتدا از یک معماری U-Net برای تولید یک ماسک بخش‌بندی اولیه استفاده می‌کند. سپس این خروجی اولیه و تصویر MRI متناظر، به تکه‌های محلی تقسیم می‌شوند که به‌صورت متوالی توسط یک عامل مبتنی بر شبکه Q عمیق (DQN) پردازش می‌گردند. عامل DQN با انتخاب عملیات مورفولوژیکی بهینه (مانند اتساع و فرسایش) برای پالایش مرزهای تومور و اصلاح عدم قطعیت‌ها به‌صورت تکه به تکه، با محیط تعامل می‌کند. مجموعه داده مورد استفاده در این مطالعه شامل ۳۰۶۴ تصویر MRI با وزن‌دهی T1 و افزایش کنتراست است که برای هر دو وظیفه بخش‌بندی و طبقه‌بندی نوع تومور به کار رفته‌اند. نتایج آزمایشگاهی نشان می‌دهد که DURL-Net به ضریب تشابه دایس (DSC) برابر با ۸۶.۷۳٪، شاخص جاکارد (IoU) برابر با ۷۸.۶۸٪، ضریب کاپا (Kap) برابر با ۸۵.۲۱٪، حساسیت (Sensitivity) برابر با ۸۷.۶۸٪ و ویژگی (Specificity) برابر با ۹۶.۰۶٪ دست یافته است.

کلمات کلیدی: تصویربرداری تشدید مغناطیسی، تومور مغزی، یادگیری تقویتی عمیق، شبکه Q عمیق، بخش‌بندی تصویر.