

Received 26 January 2025, accepted 17 March 2025, date of publication 24 March 2025, date of current version 3 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3554405



## RESEARCH ARTICLE

# Deep Learning-Based MRI Brain Tumor Segmentation With EfficientNet-Enhanced UNet

PRADEEP KUMAR TIWARY<sup>ID1</sup>, PRASHANT JOHRI<sup>ID1</sup>, ALOK KATIYAR<sup>ID1</sup>, (Member, IEEE), AND MAYUR KUMAR CHHIPA<sup>ID2,3</sup>, (Member, IEEE)

<sup>1</sup>School of Computer Science and Engineering, Galgotias University, Greater Noida 203201, India

<sup>2</sup>Department of Electronics and Communication Engineering (ECE), ISBAT University, Kampala, Uganda

<sup>3</sup>Open Distance and E-Learning (OdeL) Program, ISBAT University, Kampala, Uganda

Corresponding author: Mayur Kumar Chhipa (mayur.fict@isbatuniversity.com)

**ABSTRACT** Medical image segmentation plays a critical role in the field of medical image processing. Precisely delineating brain tumor areas from multimodal MRI scans is crucial for clinical diagnosis and predicting patient outcomes. However, challenges arise from similar intensity patterns, varying tumor shapes, and indistinct boundaries, which complicate brain tumor segmentation. Traditional segmentation networks like UNet face difficulties in capturing comprehensive long-range dependencies within the feature space due to the limitations of CNN receptive fields. This limitation is particularly significant in tasks requiring detailed predictions such as brain tumor segmentation. Inspired by these constraints, this study suggests incorporating EfficientNet as an encoder within UNet, with a thorough reassessment of its fundamental components: the encoder, bottleneck, and skip connections. EfficientNet replaces UNet's encoder, initially frozen to retain learned features from pre-trained weights, adept at extracting detailed features crucial for precise segmentation like brain tumors from MRI scans. Preserving UNet's bottleneck compresses EfficientNet's outputs, while skip connections maintain spatial integrity during decoder upsampling. The decoder reconstructs the original image size by merging encoder-decoder features, refining boundaries with convolutional layers for accurate clinical insights. The study conducted multiclass operations on the Brain-Tumor.npy dataset from Kaggle, consisting of 3064 T1-weighted contrast-enhanced images from 233 patients with meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices). Experimental findings in brain tumor segmentation tasks show that the proposed model achieves performance on par with or better than recent CNN or Transformer models. Specifically, the model achieves an accuracy of 0.9925 and a loss of 0.2991 on the dataset.

**INDEX TERMS** Brain tumor, magnetic resonance imaging, deep learning, EfficientNet, encoder, UNet, segmentation.

## I. INTRODUCTION

This Brain tumors significantly threaten human life and health, with treatment methods and prognoses varying widely among different types. Early diagnosis and treatment are vital for improving survival rates [1]. In medical contexts, magnetic resonance imaging (MRI) is a commonly utilised non-invasive diagnostic technique [2]. It provides a variety of imaging modalities that emphasise various

brain structures and diseases, including T2-weighted (T2w), T1-weighted (T1w), fluid attenuation inversion recovery (FLAIR), and contrast-enhanced T1-weighted imaging. The enhanced tumour region (ET), oedema region (ED), necrotic core region (NCR/NET), and non-enhancing tumour region (NCR) are overlapping areas that show up in brain tumours. Each of these regions has distinct biological properties. The entire tumour (WT), the tumour core (TC), and the extratumor (ET) are the areas that will be the main focus of segmentation. WT refers to the three subregions as a whole, whereas TC also covers NCR/NET and ET. It takes a lot of time, labour, and

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Huang<sup>ID</sup>.

human error to manually divide these areas from MRI data. As such, the precise segmentation of brain tumour areas using multi-modal MRI may be automated, which can significantly improves the speed and accuracy of diagnosis. Still, this is a difficult process. The identification of tiny objects is made more difficult by the considerable differences and poor contrast between modalities caused by different imaging methods. Furthermore, direct matching is difficult because to the great variability in the appearance and location of tumour subregions. Lastly, accurate tumour contour delineation is hindered by the hazy borders between healthy and diseased tissues, which limits the segmentation model's capacity to properly utilise important clinical knowledge.

A number of approaches have been put out to address the issues raised. Conventional strategies for brain tumour segmentation include region-growing algorithms [5], thresholding approaches [4], and machine learning-based techniques [3]. These traditional techniques need manual feature design and rely on intricate feature engineering while providing restricted feature representation. Convolutional Neural Networks (CNNs) have been widely used in automatic segmentation since deep learning has made them extremely successful in a variety of visual tasks. For example, UNet [6] has a symmetric encoder-decoder architecture with skip connections, which makes it possible to make dense predictions at the pixel level and has made it a major player in medical picture segmentation. In order to enhance segmentation performance, a number of UNet variations, including ResUNet, AttnUNet, and UNet++, have been created by combining conventional modules such dense connections, inception, residual connections, and attention techniques.

This study present a notable advancement in this field involves the integration of EfficientNet with the UNet architecture. EfficientNet is a family of convolutional neural networks (CNNs) that balances network depth, width, and resolution, achieving high performance with fewer parameters and computational resources. This efficiency makes it particularly well-suited for enhancing the encoder part of the UNet model. The UNet architecture, widely recognized for its effectiveness in medical image segmentation, consists of a symmetric encoder-decoder structure. The encoder captures features at various levels of abstraction, while the decoder reconstructs the image to achieve pixel-level segmentation. Skip connections between the encoder and decoder layers help preserve spatial information, crucial for precise segmentation. By incorporating EfficientNet into the UNet's encoder, the model benefits from EfficientNet's advanced feature extraction capabilities. EfficientNet's scalable architecture allows the model to capture more detailed and nuanced features from MRI scans, leading to improved segmentation accuracy. This enhanced feature extraction is particularly beneficial for identifying the complex and varied structures of brain tumors. The deep learning framework automates the learning process, eliminating the need for manual feature engineering. The model learns to identify and extract relevant

features from the MRI scans during training, adapting to the specific characteristics of the dataset. This automation not only improves accuracy but also significantly reduces the time and effort required for segmentation.

This study presents three notable contributions to the field of brain tumor segmentation in Magnetic Resonance Imaging (MRI):

- 1) Integration of EfficientNet enhances UNet's encoder, improving feature extraction for more detailed brain tumor segmentation in MRI.
- 2) This research contributes by leveraging deep learning to automate the feature extraction process, thereby reducing manual effort and enhancing segmentation accuracy. The proposed method adapts effectively to the unique characteristics of the dataset, improving overall performance in tumor segmentation.
- 3) Optimized UNet architecture, augmented with EfficientNet, achieves superior pixel-level segmentation performance, preserving spatial information crucial for precise delineation of brain tumor boundaries.

The remainder of the paper is organized as follows: Section II reviews previous research and methodologies related to brain tumor classification and segmentation to set the context and highlight existing research gaps. Section III details the proposed methodology, specifically how EfficientNet is integrated with UNet's encoder to enhance feature extraction and overall performance. Section IV presents the experimental setup and results, including dataset specifics and performance metrics. Section V provides a discussion on the significance and implications of the findings, while Section VI concludes the paper and outlines potential future research directions.

## II. RELATED WORK

In order to differentiate between benign and malignant tumours, the study by [7] employs Convolutional Neural Networks (CNN) with their sophisticated capabilities to categorise brain MRI scans on a public dataset. With no need for manually created models, CNNs are very good at extracting features, which leads to great accuracy in classification. The suggested hybrid model uses threshold-based segmentation for recognition and integrates CNN with a Support Vector Machine (SVM) for categorisation. In this study, the hybrid CNN-SVM model achieves 98.4959% total accuracy.

The author of [8] used the EfficientNetB0 framework and explainable AI approaches to improve interpretability and accuracy. Grad-CAM visualization was employed to identify significant areas in MRI images that influence categorization decisions. In terms of precision and recall, our model outperformed 97% for all four categories of brain tumours (glioma, meningioma, no tumour, and pituitary), with a classification accuracy of 98.72%.

Using T1, T2, and FLAIR images, the methodology proposed in [9] performs 3D segmentation of heterogeneous brain tumours, including gliomas. Using 3D multi-scale

convolutional layers for improved feature extraction, the unique “MS-SegNet” architecture drastically reduces variables to 10 million when compared to the current models. The BraTS 2020 and 2021 datasets resulted in Dice Coefficients ranging from 81% to 91% for the model, while the contemporary dataset from PGIMER, Chandigarh, produced Dice Coefficients between 68% and 79%. To address class imbalance, the model incorporates a customized loss function.

Federated learning falls into three categories: global, personalised, and hybrid. It is well-liked in medical picture analysis due to its privacy-preserving data aggregation. Its applicability to the Federated Brain Tumour Segmentation 2022 dataset has not yet been investigated, nevertheless. This study by [10] compares federated learning algorithms across all categories, demonstrating that although FedAvg performs admirably, certain techniques offer marginally better performance and less bias. It also investigates the behaviour of federated learning under various data distributions, such as restricted data settings and Independent and Identical Distributed (IID). By combining bilinear interpolation upsampling, a convolutional block attention module, and a residual grouped convolution module, [11] improves on the traditional U-net. With an improved Dice score of 97.581%, this enhanced model outperformed the conventional U-net by 12.438% and showed superior brain tumour segmentation in MRI scans.

The research presented in [12] introduces a diagnostic method that integrates deep learning and a meta-heuristic algorithm. This method includes utilising AlexNet to extract features from MRI scans, simplifying AlexNet using an Extreme Learning Machine (ELM) for classification, and using the Amended Grasshopper Optimisation Algorithm (AGOA) to optimise the parameters of the ELM. This approach demonstrated exceptional accuracy, precision, and other metrics when tested on a dataset of patients with glioblastoma. It also demonstrated resilience to noise and different resolutions, indicating its potential for use in more general medical imaging applications.

A separate study by [13] presents a multi-modality picture feature fusion and deep residual learning approach to brain tumour segmentation. Through feature extraction and combination from several modalities, this method improves segmentation efficiency when using a deep convolutional neural network. It received Dice scores of 83.3%, 89.07%, and 91.44% for increased tumour, tumour core, and total tumour, correspondingly, when tested on the BraTS2021 dataset.

The method proposed in [14] utilizes an information learning mechanism to ensure that complete tumor information is preserved, even when other modalities are unavailable, by transferring data from the available modalities to a single modality. To improve shared feature representation, it has a feature augmentation technique and a reconstruction module that use available data to build missing modality features.

Upon assessment using BraTS datasets, our approach outperformed state-of-the-art techniques in situations with missing modalities, obtaining Dice scores of 86.28%, 77.02%, and 59.64% for entire tumours, tumour cores, and enhanced tumours on BraTS2018. Using deep learning architectures, Inception V3 and DenseNet201, [15] presents a sophisticated classification method that extracts important characteristics from MRI pictures. Accurate categorisation is improved by combining these criteria with radiomic parameters. The Particle Swarm Optimized Kernel Extreme Learning Machine (PSO-KELM) is used to classify tumors into No Tumor, Gliomas, Meningiomas, and Pituitary Tumors. Evaluated on two benchmark datasets, the model achieved accuracies of 96.17% and 97.92%, respectively, and testing accuracies of 97.97% and 98.21%.

Author's of [16] employs a transfer learning approach with EfficientNet models to classify brain tumors into glioma, meningioma, and pituitary tumor categories. Five EfficientNet variants (B0-B4) were fine-tuned using the CE-MRI “Figshare - Brain Tumor Dataset.” The fine-tuned EfficientNetB2 model, incorporating ImageNet weights and additional layers, achieved remarkable results: 98.86% accuracy, 98.65% precision, 98.77% recall, and 98.71% F1-score. This model is both computationally efficient and effective in generalizing tumor classification. The Study by [17] presents a method that combines EfficientNet with multi-path convolution and a multi-head attention network for grade classification. This approach uses the pre-trained EfficientNetB4 for feature extraction, followed by a multi-path convolution and multi-head attention network to enhance features. These are then classified using a fully connected double dense network. Tested on TCIA repository datasets for normal, low-grade, and high-grade classifications, the model achieved an accuracy of 98.35% and a Jaccard coefficient of 97.32%.

The study in [18] introduces an advanced framework for brain tumor segmentation and classification using deep learning methods. The approach uses the 3D-UNet model for volumetric segmentation of MRI images, followed by tumor classification through Convolutional Neural Networks (CNNs). Loss and precision metrics are provided to validate the models, with performance evaluated and compared to existing techniques.

The researchers in [19] propose an automated MRI-based tumor categorization method. In preprocessing, min-max normalization is applied to standardize intensity. The Sailfish Political Optimizer (SPO), a novel combination of the Sailfish Optimization Algorithm (SOA) and Political Optimizer (PO), is used to train the Optimal DeepMRSeg technique for tumor segmentation. Convolutional Neural Networks (CNNs) are employed for feature extraction, with data augmentation techniques such as flipping and random translation. For classification, a Generative Adversarial Network (GAN) trained using a Conditional Autoregressive Value at Risk-based Sailfish Political Optimizer (CAViaR-SPO) is

utilized. The system demonstrates impressive performance with an accuracy of 91.7%, segmentation accuracy of 90%, sensitivity of 92.8%, and specificity of 92.5%.

The proposed research by [20] introduces a Salp Water Optimization-based Deep Belief Network (SWO-based DBN) for the automatic categorization of tumors into benign, malignant, core, and edematous categories. Artifact removal is performed during preprocessing. Tumor segmentation is achieved using SegNet, trained with SWO, while CNNs are used for feature extraction. Based on these features, the SWO-based DBN effectively classifies tumors. This method achieved remarkable results on the BRATS 2018 dataset with an accuracy of 0.933, specificity of 0.880, and sensitivity of 0.938. Similarly, on the BRATS 2020 dataset, it produced outstanding results with an accuracy of 0.921, specificity of 0.853, and sensitivity of 0.928.

Another study in [21] presents a binary classification technique utilizing MRS data and deep learning to differentiate between brain tumors, normal brain tissue, and pseudo-brain tumors. Data augmentation was employed to enhance the dataset, and a stacked Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) model was implemented. The model achieved high accuracy in identifying pseudo-brain tumors, with accuracies of 93.44% for glioblastoma, 85.56% for diffuse astrocytoma, 88.33% for metastatic tumors, and 99.23% for normal tissue.

Based on the literature review, several gaps were identified:

- Research frequently relies on single-modality data or fails to fully integrate multiple imaging modalities, which limits potential improvements in classification accuracy and robustness.
- Despite advancements in model performance, there is often insufficient focus on training optimization and generalization across diverse datasets, which can affect model effectiveness.

Addressing these gaps, the presented research offers improved accuracy and a more robust model.

### III. METHODOLOGY - EFFICIENTNET ENHANCES UNET'S ENCODER

#### A. THE UNET ARCHITECTURE

Convolutional neural networks like U-Net are used in image segmentation applications. The two primary components of its symmetrical, which U-shaped design are the decoder (expanding path) and the encoder (contracting path).

##### 1) ENCODER (CONTRACTING PATH)

Context and feature extraction from the incoming picture are the encoder's responsibilities. This is made up of many convolutional blocks, with two  $3 \times 3$  convolutions in each, a rectified linear unit (ReLU), and a  $2 \times 2$  max pooling operation with stride 2 for downsampling coming after. More complicated characteristics may be captured at lower spatial resolution since each block doubles the number of feature

channels. Mathematically, each layer can be represented as:

$$\text{Conv}(x) = \text{ReLU}(\text{Conv2D}(x)) \quad (1)$$

$$\text{Downsample}(x) = \text{MaxPool2D}(x) \quad (2)$$

Here,  $x$  typically refers to the input tensor or feature map.

##### 2) BOTTLENECK

At the bottom of the U, there is a bottleneck layer that consists of two  $3 \times 3$  convolutions followed by ReLU activations, without downsampling. This layer captures the deepest features of the image at the lowest resolution.

##### 3) DECODER (EXPANDING PATH)

The decoder is responsible for upsampling and reconstructing the spatial dimensions of the feature maps while retaining the high-level features captured by the encoder. Each block in the decoder consists of an upsampling operation, typically implemented as a transposed convolution (also known as a deconvolution), followed by two  $3 \times 3$  convolutions with ReLU activations. Each upsampling step halves the number of feature channels.

$$\text{Upsample}(x) = \text{Conv2DTranspose}(x) \quad (3)$$

$$\text{Conv}(x) = \text{ReLU}(\text{Conv2D}(x)) \quad (4)$$

Here,  $x$  typically refers to the input tensor or feature map.

##### 4) SKIP CONNECTIONS

The skip connections, which link the relevant levels of the encoder and decoder directly, are an essential component of the U-Net design. By concatenating feature maps from the encoder to the decoder, these links aid in the preservation of spatial information that is lost during downsampling.

$$\text{Concat}(x, y) = [x; y] \quad (5)$$

Here,  $x$  and  $y$  represent two tensors or feature maps that are being concatenated along a specified axis.

##### 5) FINAL LAYER

The final layer is a  $1 \times 1$  convolution that reduces the number of feature channels to the number of desired output classes, followed by a softmax or sigmoid activation function, depending on the segmentation task.

$$\text{Output} = \text{Conv2D}_{1 \times 1}(\text{features}) \quad (6)$$

Here, features refers to the input tensor or feature map that is passed through a  $1 \times 1$  Convolutional Layer.

#### B. EFFICIENTNET-B6

EfficientNet-B6 is part of the EfficientNet series, introduced by Google Research, aimed at improving the efficiency of CNNs through a balanced scaling approach. Traditional scaling methods typically increase one dimension of the network (depth, width, or resolution) independently, often leading to suboptimal performance and increased computational cost. EfficientNet overcomes this limitation with compound

scaling, which scales all three dimensions uniformly based on a compound coefficient  $\phi$ .

### 1) COMPOUND SCALING

The core innovation in EfficientNet-B6 is the compound scaling method, which uses a set of fixed scaling coefficients  $\alpha$ ,  $\beta$ , and  $\gamma$  to scale the depth  $d$ , width  $w$ , and resolution  $r$  of the network, respectively:

$$d = \alpha^\phi \quad (7)$$

$$w = \beta^\phi \quad (8)$$

$$r = \gamma^\phi \quad (9)$$

For EfficientNet-B6, the compound coefficient  $\phi$  is set to 6, which determines the extent to which each dimension is scaled. The specific values of  $\alpha$ ,  $\beta$ , and  $\gamma$  are derived from a grid search aimed at optimizing the trade-off between accuracy and computational efficiency.

### 2) ARCHITECTURAL BUILDING BLOCKS

EfficientNet-B6 is built using Mobile Inverted Bottleneck Convolution (MBConv) blocks with Squeeze-and-Excitation (SE) optimization. Each MBConv block comprises:

- **Expansion Phase:** Expands the input channels by a factor, enabling richer feature extraction:

$$\text{Conv}_{\text{expand}}(x) = \text{ReLU6}(\text{Conv2D}_{1 \times 1}(x)) \quad (10)$$

- **Depthwise Convolution Phase:** Applies depthwise separable convolutions, which perform spatial convolutions independently across each channel, reducing computational cost:

$$\text{Conv}_{\text{dw}}(x) = \text{ReLU6}(\text{DepthwiseConv2D}(x)) \quad (11)$$

- **Squeeze-and-Excitation (SE) Phase:** Recalibrates channel-wise feature responses by modeling interdependencies between channels:

$$\text{SE}(x) = \sigma(W_2 \text{ReLU}(W_1 \text{GlobalAvgPool}(x))) \quad (12)$$

- **Projection Phase:** Reduces the number of channels back to the desired output size, facilitating efficient information flow through the network:

$$\text{Conv}_{\text{project}}(x) = \text{Conv2D}_{1 \times 1}(x) \quad (13)$$

### 3) EFFICIENTNET-B6 ARCHITECTURE

EfficientNet-B6 is structured with repeated MBConv blocks, where each block is configured according to the compound scaling strategy. The network begins with a standard convolutional layer, followed by a series of MBConv blocks organized into stages. Each stage operates at a different resolution and channel depth, allowing the network to capture hierarchical features.

The architecture of EfficientNet-B6 [Figure 1] can be summarized as follows:

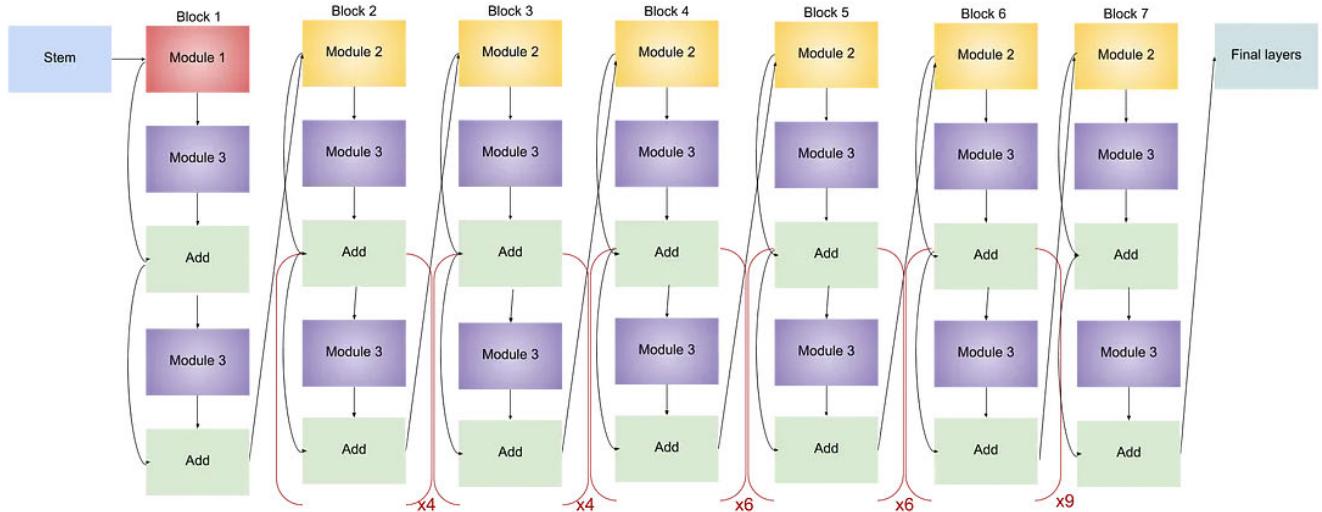
- **Stem:** Initial  $3 \times 3$  convolution with 32 filters.

- **Stage:** Repeated MBConv blocks with varying expansion factors, kernel sizes, and output channels.
- **Top:** Final  $1 \times 1$  convolution to produce the desired number of output classes, followed by a global average pooling layer and a fully connected layer for classification.

EfficientNet-B6 is trained using standard techniques, including data augmentation, dropout, and batch normalization. The Adam optimizer is commonly used to update the model parameters, with a learning rate schedule to facilitate convergence.

EfficientNet-B6 leverages its balanced compound scaling to provide a highly effective and computationally efficient model. The use of MBConv blocks with SE modules enhances its feature extraction capabilities, making it a powerful tool for image classification and other computer vision tasks. By increasing input resolution and scaling depth and width proportionally, EfficientNet-B6 achieves a good trade-off between accuracy and efficiency, making it suitable for high-resolution, complex image analysis tasks.

The MBConv block and squeeze-and-excitation (SE) module are crucial components in the architecture of EfficientNet-B6, contributing to its efficiency and high performance. The MBConv block is an advanced building block that combines depthwise separable convolutions and an inverted residual structure. Depthwise convolution reduces the computational cost by applying separate filters to each input channel rather than using standard convolutions, which significantly lowers the number of parameters and computations. The inverted residual structure further enhances efficiency by first expanding the input features with a lightweight  $1 \times 1$  convolution, then applying a depthwise convolution, and finally reducing the dimensionality with another  $1 \times 1$  convolution. This method enables the network to focus on learning rich feature representations while minimizing the computational overhead. Additionally, the linear bottleneck at the end of the block helps to reduce the number of channels, making it particularly useful for resource-constrained environments. On the other hand, the squeeze-and-excitation (SE) module enhances the representational capacity of the network by explicitly modeling channel-wise dependencies. The squeeze operation uses global average pooling to aggregate spatial information across each channel, essentially summarizing the spatial features into a single value per channel. The excitation operation follows, using fully connected layers and a sigmoid activation to learn the importance of each channel. These learned weights are then used to recalibrate the feature map, scaling the output of each channel based on its importance. This channel-wise recalibration enables the network to focus on more relevant features, enhancing performance, particularly for tasks such as image classification and segmentation. EfficientNet-B6, despite its impressive performance, comes with several limitations that must be considered. One significant drawback is its increased computational overhead, requiring substantial GPU memory and processing power,



**FIGURE 1.** Architecture of EfficientNet-B6.

which can restrict its use in hardware-constrained environments. Additionally, EfficientNet-B6 achieves optimal results with higher-resolution input images, leading to increased preprocessing complexity, storage demands, and extended training times, especially when working with large-scale datasets. This reliance on high-resolution inputs also raises scalability challenges, making it less suitable for real-time applications or deployment on resource-limited devices like mobile or embedded systems. Furthermore, the model's performance depends heavily on high-quality and diverse datasets, and insufficient data augmentation can lead to reduced generalization capabilities. Lastly, the larger architecture of EfficientNet-B6 results in a considerable number of model parameters, complicating its deployment in scenarios where storage efficiency and runtime speed are critical. These limitations underscore the trade-offs involved in leveraging EfficientNet-B6 for high-performance tasks while balancing practical constraints.

### C. EFFICIENTNET-B6 AS AN ENCODER IN U-NET

An EfficientNet-enhanced U-Net [Figure 2] integrates the EfficientNet architecture as the encoder within the U-Net framework to leverage its efficient and powerful feature extraction capabilities for deep learning tasks. EfficientNet, known for its balanced scaling of depth, width, and resolution through a compound scaling method, significantly improves the representation of complex features in input images. By embedding EfficientNet within U-Net, the encoder benefits from EfficientNet's depthwise separable convolutions and squeeze-and-excitation blocks, which enhance the network's ability to capture detailed and hierarchical features while maintaining computational efficiency. The decoder part of the U-Net, responsible for upsampling and reconstruction, remains effective in restoring the spatial

dimensions and enabling precise segmentation through skip connections that concatenate encoder features directly with the corresponding decoder layers. This combination results in a robust architecture that excels in image segmentation tasks, providing high accuracy and detailed segmentations with reduced computational overhead, making it suitable for applications such as medical imaging and other fields requiring precise image analysis.

Here's the step-by-step approach for integrating EfficientNet-B6 as an encoder within a U-Net architecture:

#### Input:

- **I:** Input image of dimensions  $H \times W \times C$ .
- **num\_classes:** Number of output classes for segmentation

#### Output:

- **y\_output:** Segmentation map of dimensions  $H \times W \times \text{num\_classes}$

#### 1) Initialization

- Load the EfficientNet-B6 model pre-trained on ImageNet without the top classification layer.

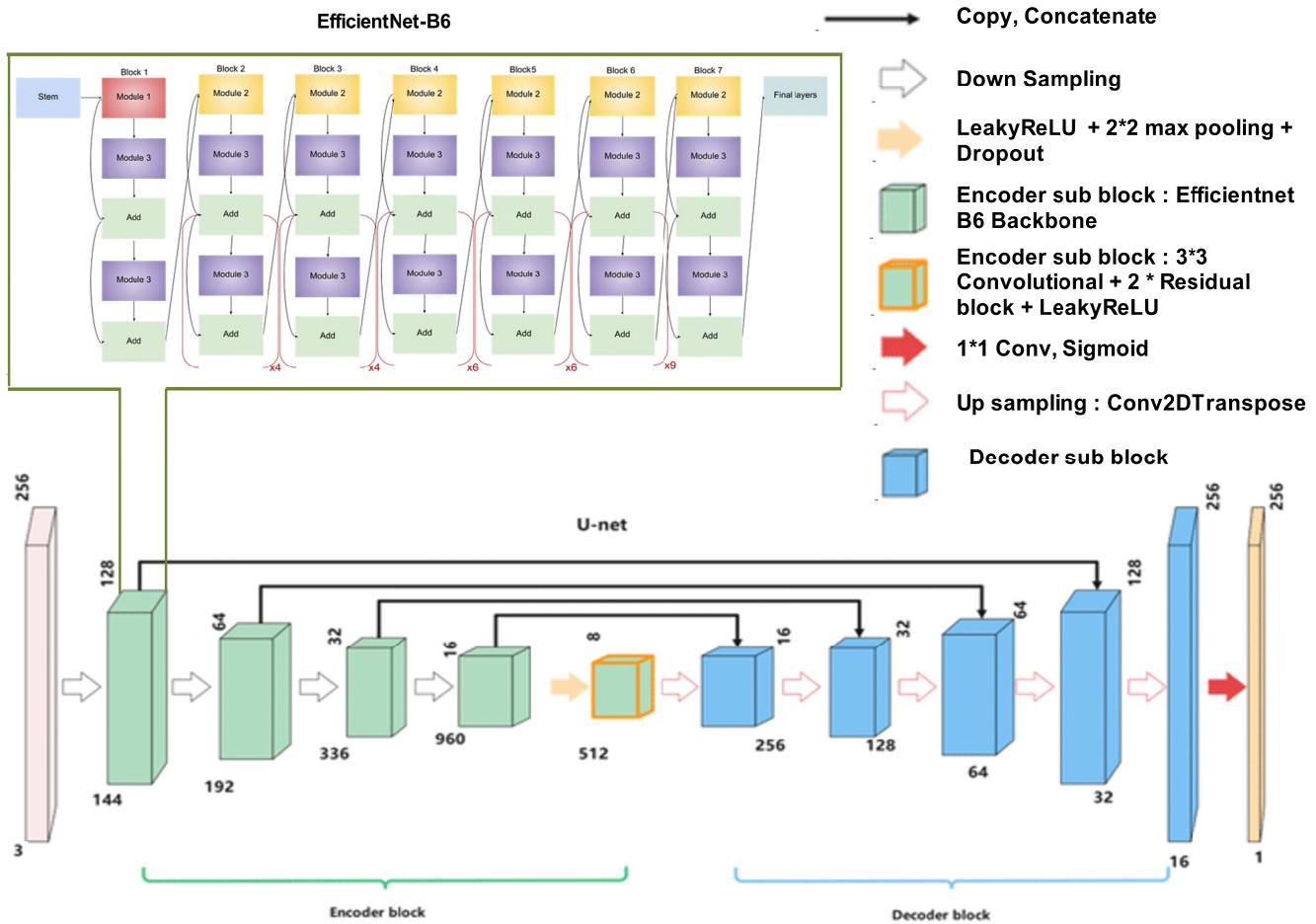
#### 2) Encoder (EfficientNet-B6):

- Pass the input image I through the initial convolution and pooling layers.
- For each MBConv block in EfficientNet-B6:
  - Perform expansion phase:

$$x_{\text{expanded}} = \text{ReLU6}(\text{Conv2D}_{1 \times 1}(x))$$

- Perform depthwise convolution:

$$x_{dw} = \text{ReLU6}(\text{DepthwiseConv2D}(x_{\text{expanded}}))$$



**FIGURE 2.** Architecture of EfficientNet-B6 as an encoder of UNet.

- Apply squeeze-and-excitation (SE) module:

$$s = \sigma(W_2 \text{ReLU}(W_1 \text{GlobalAvgPool}(x_{dw})))$$

$$x_{se} = s \cdot x_{dw}$$

- Perform projection phase

$$x_{projected}(x) = \text{Conv2D}_{1 \times 1}(x_{se})$$

- Save feature maps  $F_i$  from selected layers for skip connections.

- Store the final encoder output as the bottleneck feature map.

### 3) Bottleneck:

- Pass the deepest encoder feature map through an MBConv block to produce the bottleneck feature map B.

### 4) Decoder:

- Initialize x with the bottleneck feature map B.
- For each decoding layer i (from the deepest to the shallowest):

- Perform upsampling

$$U_i = \text{Conv2DTranspose}(D_{i+1}, \text{filters} = f_i, \text{kernal\_size} = 2, \text{stride} = 2, \text{padding} = \text{'same'})$$

- Concatenate the upsampled feature map with the corresponding encoder feature map:

$$C_i = \text{Concatenate}([U_i, F_{n-i}])$$

- Apply convolutional layers:

$$D_i = \text{ReLU}(\text{BatchNorm}(\text{Conv2D}(C_i, \text{filters} = f_i, \text{kernal\_size} = 3, \text{padding} = \text{'same'})))$$

$$D_i = \text{ReLU}(\text{BatchNorm}(\text{Conv2D}(D_i, \text{filters} = f_i, \text{kernal\_size} = 3, \text{padding} = \text{'same'})))$$

## 5) Output Layer:

- Apply a  $1 \times 1$  convolution to produce the final output segmentation map.

$$y = \text{Conv2D}(D_0, \text{filters} = \text{num\_classes}, \\ \text{kernel\_size} = 1, \text{activation} = \text{'softmax'} \\ \text{ifmulti-classelse'sigmoid'})$$

Note:

- ReLU6: This is a variant of the standard ReLU (Rectified Linear Unit) activation function. Unlike ReLU, which can output any positive value, ReLU6 caps the output at 6, ensuring that the activation is within the range of [0, 6]. This can help in certain neural network architectures, particularly in mobile networks, where controlling the range of activations can improve performance and prevent overflow.
- $W_2$ : This generally refers to a weight matrix in a neural network layer, specifically the weights associated with the second layer or operation. The subscript “2” would typically indicate its position in the network.
- $F_i$ : This typically refers to the feature map or the output of the  $i$ -th layer in a neural network. In deep learning,  $F_i$  can represent the activations from the  $i$ -th layer after applying a certain operation (such as a convolution).
- filters =  $f_i$ : In the context of a convolutional layer, filters (often denoted as  $f_i$ ) refer to the set of weights or kernels used to perform the convolution operation. The filters are learned during training and are responsible for extracting features from the input data.
- kernel\_size: This refers to the dimensions of the convolutional filter (or kernel), typically represented as (height, width). It defines the size of the window that moves over the input data during the convolution operation to extract features.
- Deepest Feature: Refers to the high-level abstract representations of the image that are captured by the later layers of the network, typically at a lower spatial resolution.
- Deepest Encoder Feature Map: This term refers to the output of the final layer of the encoder in an encoder-decoder architecture, where the encoder progressively extracts and condenses information from the input. The deepest encoder feature map contains the most abstracted representations of the input image, but with reduced spatial dimensions. These feature maps are used by the decoder to reconstruct or further process the image.

## IV. EXPERIMENT AND ANALYSIS

### A. DATASETS

3064 T1-weighted contrast-enhanced images are included in this collection, including 233 patients with meningioma (708 slices), glioma (1426 slices), and pituitary tumour (930 slices). Figure 3 presents the visualisation. To comply with repository file size restrictions, the dataset was divided

into four portions, each of which was supplied in a.zip file containing 766 slices.

This dataset is available in MATLAB data format (.mat files). For every image, a struct containing the following fields is included in every file:

- **cjdata.label**: 1 for meningioma, 2 for glioma, and 3 for pituitary tumor
- **cjdata.PID**: the patient ID
- **cjdata.image**: the image data
- **cjdata.tumorBorder**: a vector containing coordinates of discrete points along the tumor border, such as [x1, y1, x2, y2, ...], where x1 and y1 represent planar coordinates on the tumor border. This was manually outlined to facilitate the creation of a binary tumor mask image.
- **cjdata.tumorMask**: a binary image where 1s denote the tumor region

### B. VISUALIZATION AND AUGMENTATION

The dataset which include 3064 images for three type of brain tumor. split a dataset into training, validation, and test sets using train\_test\_split from scikit-learn. Initially, the dataset data is divided into a training set and a validation set, with the validation set comprising 8% of the data. This split is achieved by specifying test\_size=0.08 and setting random\_state=42 to ensure reproducibility. Next, the remaining 92% of the data, which forms the training set, is further split into a final training set and a test set. Here, 12% of this training data is allocated to the test set, again using random\_state=42 for consistency. Consequently, the final splits are approximately 80.96% for the training set, 8% for the validation set, and 11.04% for the test set. This approach ensures distinct and appropriately sized subsets for training the model, tuning hyperparameters, and evaluating performance on unseen data, thus promoting robust model development and validation.

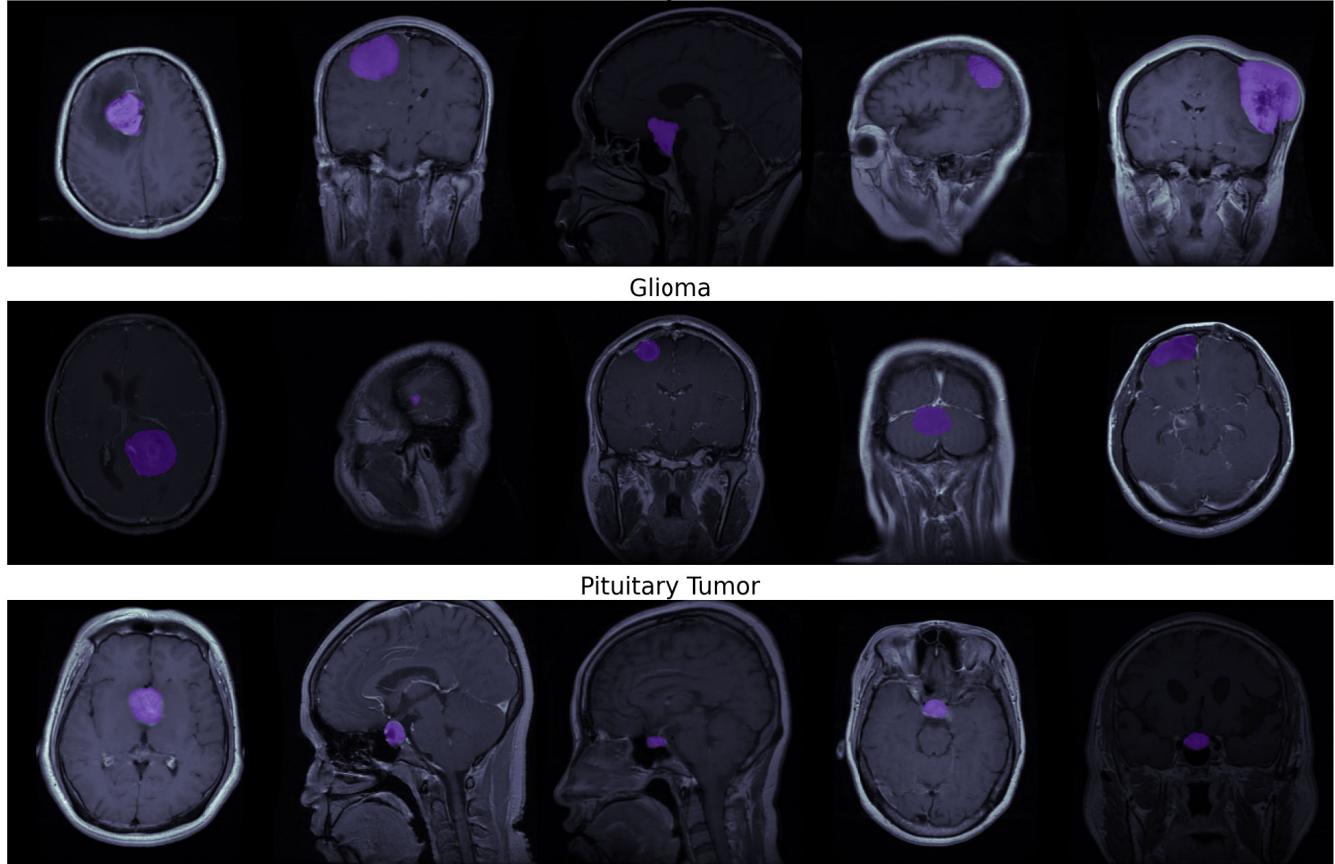
After the distribution retrieves the first element of the first sample in train\_data, which is likely a 2D grayscale image. It then creates a 3D array by stacking three copies of this image along a new dimension, resulting in a 3-channel image where each channel has identical values (similar to an RGB image with the same values in all channels). Next, it transposes the dimensions of this stacked image to reorder the axes. If the original image shape was (H, W), the new shape after stacking and transposing will be (H, W, 3). Finally, it converts the image data type to 8-bit unsigned integers, a common format for image data.

In the next step creates a 3D array of zeros with the shape (512, 512, 3). This array will serve as a mask with the same spatial dimensions as the image but with three channels.

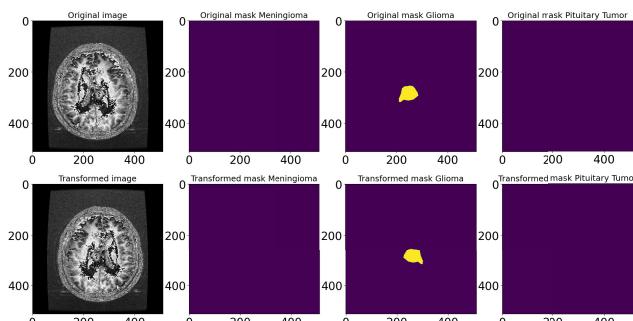
In the context of semantic segmentation, class distribution refers to the frequency or proportion of different classes (or labels) present in the dataset. In this study, the mask array is used to store the labels for different regions of the image. Understanding the class distribution can provide insights into

## Brain MRI Images for Brain Tumor Detection Brain Tumor Retrieval Dataset

Meningiomas

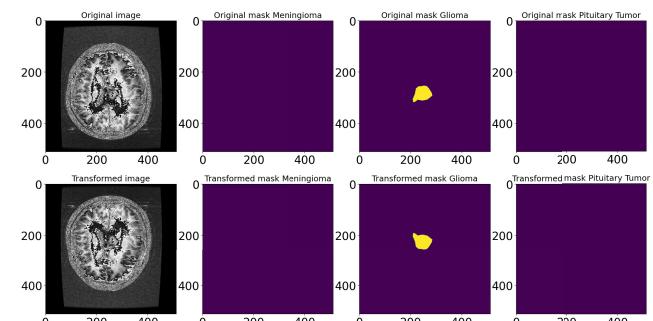


**FIGURE 3.** Brain MRI images for brain tumor detection.



**FIGURE 4.** Data augmentation technique - HorizontalFlip.

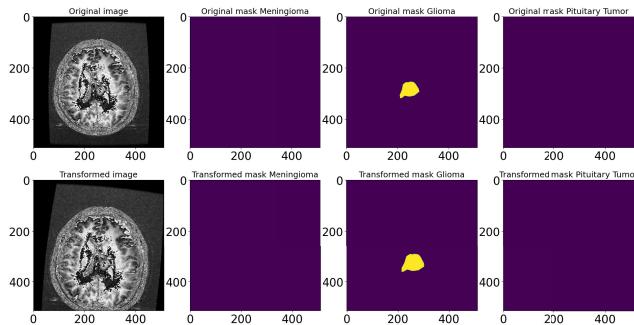
the dataset's balance, which is crucial for training models effectively. The process retrieves the third element of the first sample in `train_data`, likely an integer specifying which channel of the mask should be modified, and the second element of the first sample in `train_data`, presumably a 2D array representing some label or mask data. It converts the label or mask data to 32-bit floating-point numbers and assigns the converted data to the specified channel of the mask array.



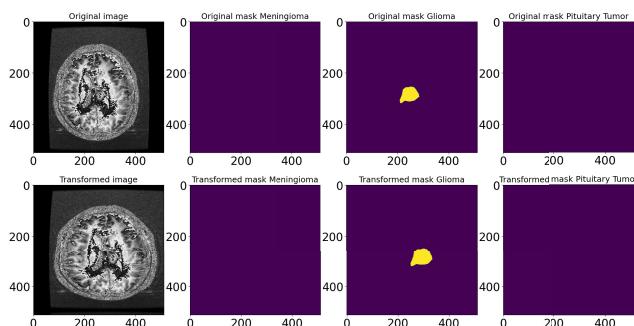
**FIGURE 5.** Data augmentation technique - VerticalFlip.

By creating altered versions of already-existing data, data augmentation creates an artificial expansion of the training dataset. This method involves creating new data points by applying deep learning techniques or by making minor changes to the dataset. The following data augmentation methods are being used in this study:

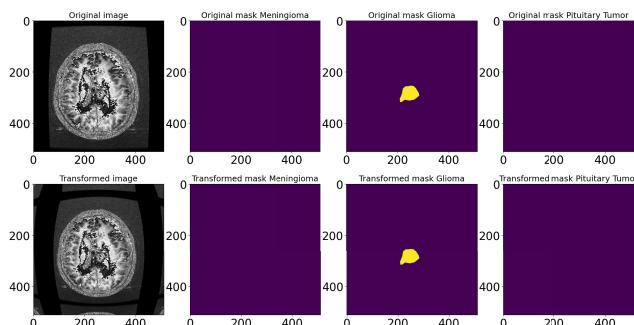
- **HorizontalFlip:** This technique flips the image horizontally, creating a mirror image [Figure 4]. It's particularly



**FIGURE 6.** Data Augmentation Technique - ShiftScaleRotate.



**FIGURE 7.** Data augmentation technique - GridDistortion.



**FIGURE 8.** Data augmentation technique - OpticalDistortion.

useful in scenarios where the orientation of the image doesn't affect the class label, such as in object detection or segmentation tasks involving symmetrical objects.

- **VerticalFlip:** This technique flips the image vertically [Figure 5]. Similar to horizontal flipping, this method is useful when the vertical orientation of the image doesn't impact the class label, providing more variability in the training dataset.
- **ShiftScaleRotate:** This method involves shifting, scaling, and rotating the image [Figure 6]. Shifting moves the image along the X or Y axis, scaling changes the size of the image, and rotating spins the image around its center. This technique helps the model become invariant to position, size, and orientation, enhancing its robustness. The image size is scaled randomly within the range of  $[1 - 0.3, 1 + 0.3]$  (i.e., 70% to 130%

of its original size). The image is rotated randomly between  $[-10^\circ, +10^\circ]$ . The shift is a fraction of the image dimensions, so a limit of 0.1 means the image can shift up to 10% of its width and height.

- **GridDistortion:** Grid distortion warps the image by distorting a grid laid over it [Figure 7]. This involves moving the grid points in various directions, altering the image's shape and features. It's useful for making the model more resilient to geometric transformations and distortions that can occur in real-world data.

- **OpticalDistortion:** This technique simulates optical lens distortions, such as barrel or pincushion distortions, which can occur due to camera imperfections [Figure 8]. By applying these distortions, the model can learn to handle real-world image imperfections and variations, improving its generalization capabilities.

### C. DATA GENERATOR

A data generator is a tool or function that produces data on-the-fly in batches, rather than loading the entire dataset into memory at once. It is commonly used in machine learning to efficiently manage large datasets by dynamically loading and preprocessing small portions of data during each iteration of the training, validation, or testing process. Data generators handle tasks such as data augmentation, shuffling, and batching, enabling scalable and memory-efficient training of machine learning models.

A data generator [Figure 9] is an essential component in the deep learning workflow, dynamically loading and preprocessing data in batches during the training, validation, and testing phases. For training, the data generator begins by creating a custom dataset using a class such as BrainMriDataset, initialized with the training data and a set of transformations defined by `get_training_augmentation()`. These transformations typically include data augmentation techniques like rotations, flips, and scaling, which help to artificially expand the dataset and improve the model's ability to generalize. The data loader for the training data, created using PyTorch's `DataLoader` class, is configured with a specific batch size, multiple worker processes to speed up data loading, and shuffling enabled. Shuffling ensures that the training data is presented in a different order each epoch, preventing the model from learning the sequence of the data and reducing the risk of overfitting.

For validation, a similar process is followed, but with some key differences. The validation dataset is created with transformations defined by `get_validation_augmentation()`, which typically include only necessary preprocessing steps such as resizing and normalization, without any data augmentation. This is because validation data should reflect real-world scenarios as closely as possible to accurately assess the model's performance. The data loader for validation is also created with a specific batch size and multiple worker processes but does not shuffle the data. Consistent ordering of the validation data ensures that the evaluation is stable and reliable across different epochs.

```

# Data Generators
# train
train_dataset = BrainMRI Dataset(data=train_data, transforms=get_training_augmentation())
train_dataloader = DataLoader(train_dataset, batch_size=BATCH_SIZE,
                             num_workers=2, shuffle=True)

# validation
val_dataset = BrainMRI Dataset(data=val_data, transforms=get_validation_augmentation())
val_dataloader = DataLoader(val_dataset, batch_size=BATCH_SIZE,
                            num_workers=2, shuffle=False)

# test
test_dataset = BrainMRI Dataset(data=test_data, transforms=get_validation_augmentation())
test_dataloader = DataLoader(test_dataset, batch_size=BATCH_SIZE,
                            num_workers=2, shuffle=False)

```

**FIGURE 9.** Data generator for training, testing and validation.

The testing phase follows the same pattern as the validation phase to maintain consistency. The test dataset is created with the same transformations as the validation dataset, ensuring that the preprocessing is identical. The data loader for the test data is also configured with a specific batch size, multiple worker processes, and no shuffling. This setup ensures that the model's final evaluation on the test data is consistent and repeatable, providing a reliable measure of its performance on completely unseen data.

#### D. LOSS FUNCTION AND EVALUATION MATRICS

##### 1) SOFT DICE LOSS

The resemblance between predicted and ground truth segmentations is assessed using a differentiable metric known as Soft Dice Loss, commonly employed in image segmentation tasks. This metric is particularly valuable as it directly enhances the Dice Coefficient, a popular metric for evaluating the quality of segmentation models.

The Dice Coefficient is defined as:

$$\text{Dice Coefficient} = \frac{2|A \cap B|}{|A| + |B|} \quad (14)$$

where:

- $|A \cap B|$ : The number of overlapping pixels between  $A$  (ground truth) and  $B$  (prediction).
- $|A|$ : The total number of pixels in the ground truth segmentation.
- $|B|$ : The total number of pixels in the predicted segmentation.

For predicted segmentation  $P$  and ground truth  $G$ :

$$\text{Dice Coefficient} = \frac{2 \sum_i (P_i G_i)}{\sum_i P_i + \sum_i G_i} \quad (15)$$

Here,  $P_i$  and  $G_i$  represent the predicted and ground truth values for the  $i$ -th pixel, respectively.

To make this differentiable and applicable to neural networks, we use the Soft Dice Loss. The Soft Dice Loss for a single class is given by:

$$\text{SoftDiceLoss} = 1 - \frac{2 \sum_i (P_i G_i)}{\sum_i P_i^2 + \sum_i G_i^2 + \epsilon} \quad (16)$$

##### 2) ACCURACY

The performance of a classification model is commonly evaluated using the accuracy measure. It figures out what proportion of accurately anticipated cases there are overall. The confusion matrix, which comprises True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), can be used to more formally calculate accuracy, especially for binary classification.

$$\text{Accuracy} = \frac{TP + TN}{FP + FN + TP + TN} \quad (17)$$

##### 3) PRECISION

When dealing with unbalanced datasets or high false positive costs, precision is a performance parameter for classification models that is very helpful. It calculates the percentage of all positively anticipated cases that were accurately forecasted as positive.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (18)$$

##### 4) RECALL

Recall, a vital performance metric for classification models, is also known as sensitivity or the true positive rate. It measures the percentage of actual positive cases that the model successfully identifies. Recall is particularly critical in situations where it is essential to detect all positive

instances, even if it leads to an increase in false positives.

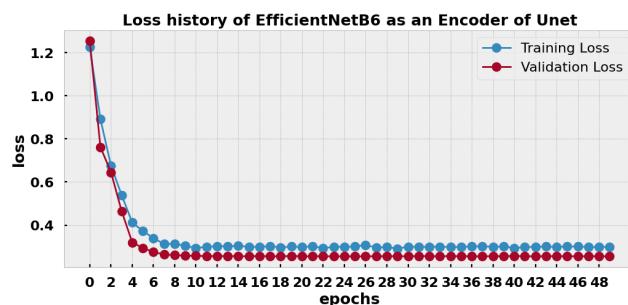
$$\text{Recall} = \frac{TP}{TP + FN} \quad (19)$$

## V. RESULT

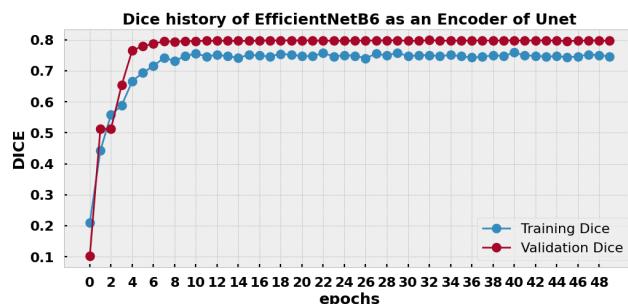
Table 1 and Figure 10, 11, 12 presents the performance of the proposed network on the presented dataset.

**TABLE 1.** Experimental result.

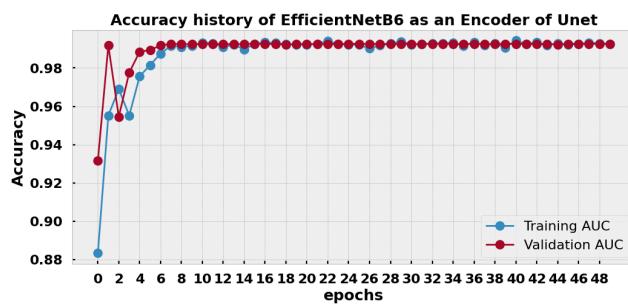
S No	Evaluation Parameter	Training	Validation
1	Loss	0.299173	0.255529
2	Dice Coefficient	0.746353	0.796635
3	Accuracy	0.992527	0.992507



**FIGURE 10.** Loss history of EfficientNetB6 as an encoder of UNet.



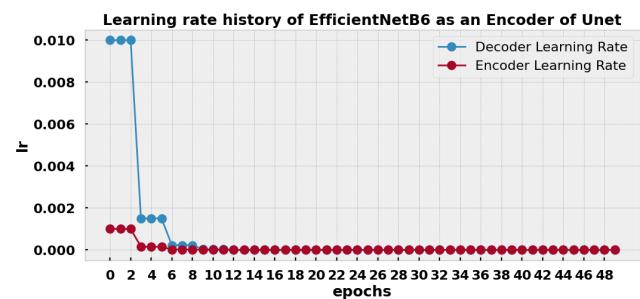
**FIGURE 11.** Dice history of EfficientNetB6 as an encoder of UNet.



**FIGURE 12.** Accuracy history of EfficientNetB6 as an encoder of UNet.

The amount that the model's weights are altered in relation to the loss gradient during training is determined by the learning rate (LR) [Figure 13]. A proper learning rate must be set for the training process to be successful. A learning

rate that is too high might lead to the model diverging or converge too soon to a suboptimal solution, while a learning rate that is too low could cause the training process to become unduly sluggish and even locked in local minima. The learning rate history of EfficientNetB6 as an encoder for a U-Net model is crucial for understanding how the model's training progresses and adjusts to the learning task. EfficientNetB6 is a powerful convolutional neural network known for its efficiency and performance, and using it as an encoder in U-Net combines its feature extraction capabilities with U-Net's strong segmentation abilities. Figure 13 shows the learning rate history for the decoder and encoder process.



**FIGURE 13.** Learning rate history of EfficientNetB6 as an encoder of UNet.

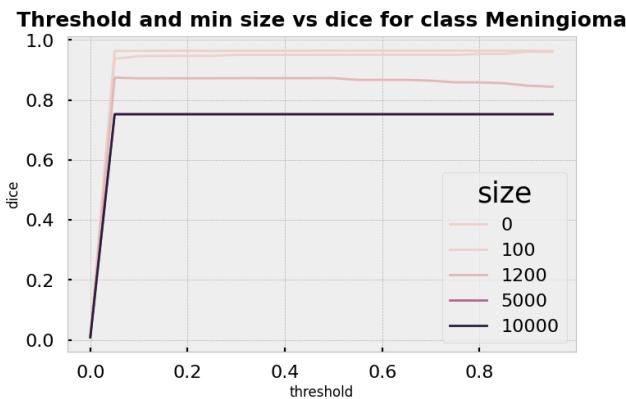
In deep learning, a threshold is a value that sets a boundary for decisions or operations within a neural network. It is often used in activation functions, like the step function, to determine whether a neuron should be activated based on the input value. Setting appropriate thresholds and minimum size parameters is crucial for segmenting meningioma, glioma, and pituitary tumors because these adjustments directly influence the accuracy and clinical relevance of the model's predictions. In medical imaging, precise tumor segmentation is vital for diagnosis, treatment planning, and monitoring disease progression.

**Threshold Importance:** The threshold determines which regions of the image are classified as tumor tissue. A poorly chosen threshold can either miss parts of the tumor (low recall) or include too much non-tumor tissue (low precision), leading to inaccurate diagnoses and suboptimal treatment plans.

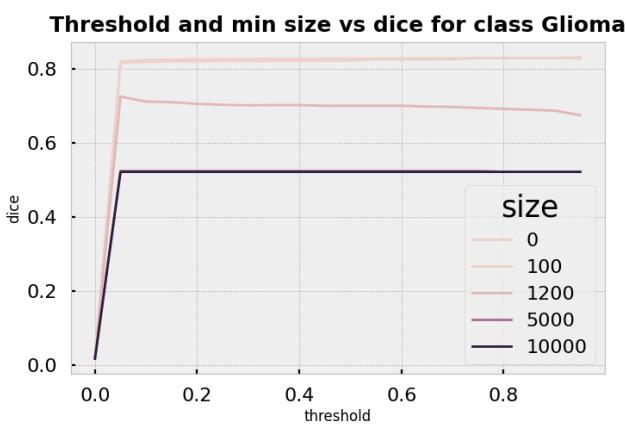
**Minimum Size Importance:** The minimum size parameter filters out small, potentially erroneous segments that are unlikely to be true positives. Tumors typically have a certain minimum size, and segments below this threshold are often noise. By removing these small segments, the model's output becomes cleaner and more clinically useful.

**Dice Coefficient Importance:** The Dice coefficient measures the overlap between the predicted segmentation and the ground truth, reflecting the model's performance. High Dice scores indicate that the model accurately captures the tumor's shape and size, crucial for effective medical interventions.

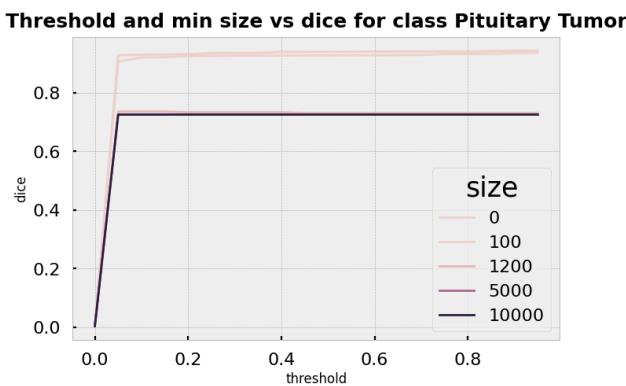
Figure [14, 15, 16] represent the threshold and minimum size vs dice based on the classes meningioma, glioma and pituitary tumor.



**FIGURE 14.** Threshold and min size vs Dice for class meningioma.



**FIGURE 15.** Threshold and min Size vs Dice for class glioma.



**FIGURE 16.** Threshold and min size vs Dice for class pituitary tumor.

The Dice coefficient [2], also known as the Dice similarity index, is a statistical measure used to evaluate the accuracy of image segmentations by comparing the overlap between the predicted segmentation and the ground truth. It ranges from 0 to 100%, with higher values indicating better agreement between the predicted and actual segmentations. The Dice coefficient is particularly important in medical imaging for assessing the quality of tumor segmentation.

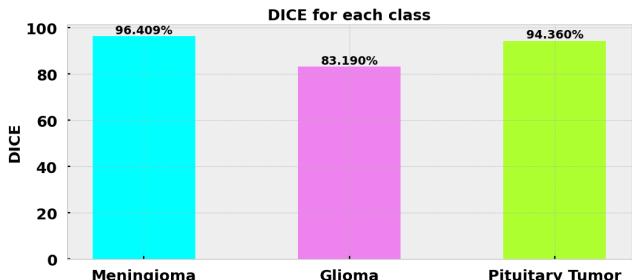
For meningioma, a Dice value of 96.409% indicates an excellent overlap between the predicted tumor region and the actual tumor, signifying that the segmentation model is highly accurate for this type of tumor. This high accuracy is crucial for effective treatment planning and monitoring.

For glioma, a Dice value of 83.190% suggests a moderate level of accuracy. While the model is relatively good at segmenting gliomas, there is still room for improvement. This is important because gliomas are often more heterogeneous and infiltrative, making accurate segmentation more challenging but essential for precise treatment.

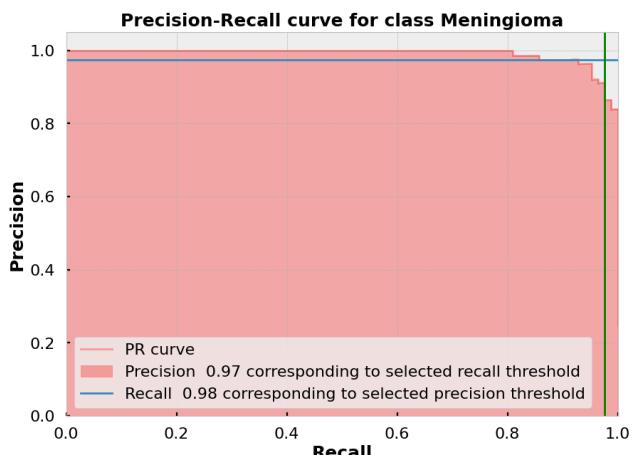
For pituitary tumors, a Dice value of 94.360% indicates a very high level of accuracy, similar to meningiomas. Accurate segmentation of pituitary tumors is vital for planning surgeries and other interventions, as these tumors are located in a sensitive and complex region of the brain. Figure 17 shows the Dice for each class.

**TABLE 2.** Experiment result for Dice, Precision and Recall for each class.

Class	Dice	Precision	Recall
Meningioma	0.96409	0.97	0.98
Glioma	0.83190	1.00	0.99
Pituitary	0.94360	1.00	1.00

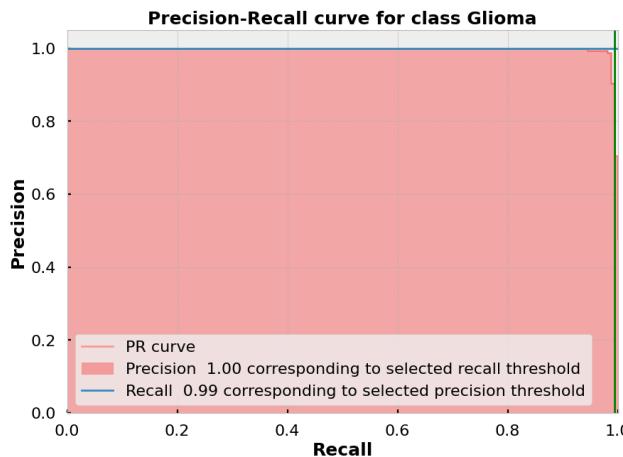


**FIGURE 17.** Dice for each class.

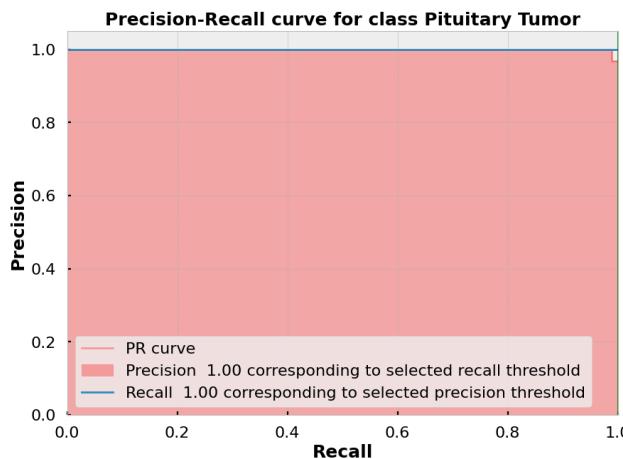


**FIGURE 18.** Precision-recall curve for class meningioma.

The precision-recall curve (PRC) provides a comprehensive view of the trade-offs between precision and recall for different thresholds, which is particularly valuable for evaluating models in medical imaging for tumor segmentation. The



**FIGURE 19.** Precision-recall curve for Class Glioma.



**FIGURE 20.** Precision-recall curve for Class Pituitary Tumor.

curve will be close to the top-right corner, indicating a strong balance between precision and recall. High recall slightly outweighs precision, meaning the model prioritizes capturing most tumor regions with few false positives. Figure [18, 19, 20] shows the trade-offs between precision and recall.

## VI. DISCUSSION

The EfficientNetB6 as an Encoder of UNet model 3 achieves the highest accuracy at 0.9925, utilizing EfficientNetB6 for robust feature extraction within the UNet architecture. Figure 22 is showing the prediction for the proposed model. The comparative analysis highlights the performance of state-of-the-art segmentation methods based on accuracy. The CNN-SVM model achieves an impressive accuracy of 0.9849 by integrating convolutional neural networks for feature extraction and support vector machines for classification, showcasing robust synergy. The ARGA-UNet, incorporating attention mechanisms into the U-Net framework, achieves 0.9758 accuracy, emphasizing its ability to focus on relevant features but slightly trailing behind more advanced methods. The PSO-KELM + Inception

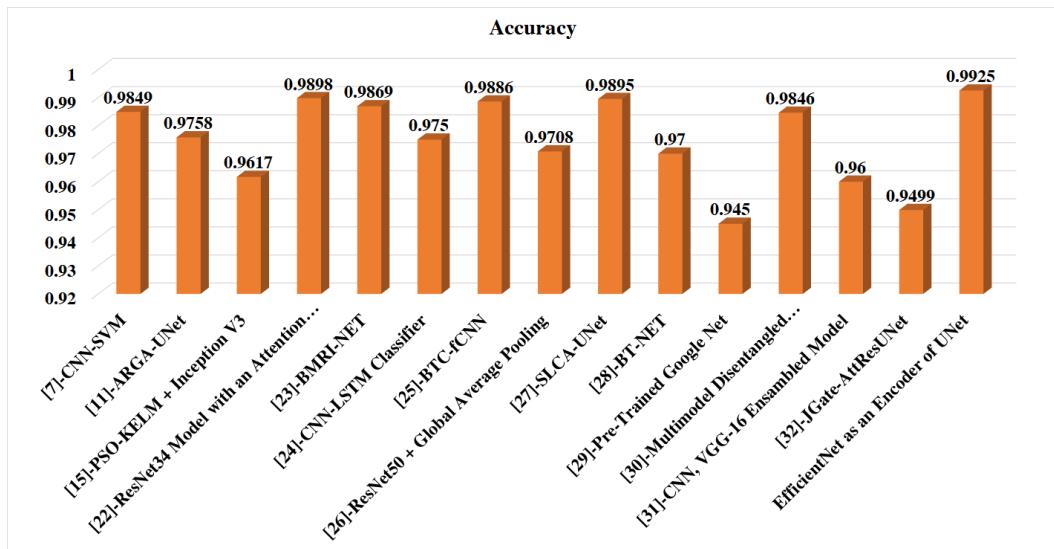
**TABLE 3.** Comparison from previously proposed work.

S No	Reference	Model and Framework	Accuracy
1	[7]	CNN-SVM	0.9849
2	[11]	ARGA-UNet	0.9758
3	[15]	PSO-KELM + Inception V3	0.9617
4	[22]	ResNet34 Model with an Attention Block	0.9898
5	[23]	BMRI-NET	0.9869
6	[24]	CNN-LSTM Classifier	0.9750
7	[25]	BTC-fCNN	0.9886
8	[26]	ResNet50 + Global Average Pooling	0.9708
9	[27]	SLCA-UNet	0.9895
10	[28]	BT-NET	0.9700
11	[29]	Pre-Trained Google Net	0.9450
12	[30]	Multimodel Disentangled Variational Autoencoder	0.9846
13	[31]	CNN, VGG-16 Ensembled Model	0.9600
14	[32]	JGate-AttResUNet	0.9499
15	Proposed	EfficientNet as an Encoder of UNet	0.9925

V3 approach, combining particle swarm optimization and kernel extreme learning machines with Inception V3, reaches 0.9617 accuracy, reflecting solid performance despite being outpaced by deeper architectures. The ResNet34 model with an attention block stands out with 0.9898 accuracy, leveraging attention mechanisms to enhance feature refinement. Similarly, BMRI-NET achieves 0.9869 accuracy by employing tailored strategies for brain tumor segmentation. The CNN-LSTM classifier, blending convolutional layers with long short-term memory (LSTM) networks for temporal data modeling, delivers 0.975 accuracy. The BTC-fCNN reaches 0.9886 accuracy, demonstrating the efficiency of fully convolutional networks for segmentation tasks. Meanwhile, ResNet50 + Global Average Pooling achieves 0.9708 accuracy, benefiting from its deep residual learning capabilities and global pooling for feature aggregation. The SLCA-UNet achieves 0.9895 accuracy, integrating spatial and channel attention modules for enhanced performance. BT-NET, tailored for brain tumor segmentation, delivers 0.97 accuracy, while the pre-trained Google Net lags slightly with 0.945 accuracy. The Multimodal Disentangled Variational Autoencoder reaches 0.9846 accuracy, showcasing its capability to handle diverse data modalities effectively. The CNN and VGG-16 ensembled model achieves 0.96 accuracy, reflecting the benefits of ensemble learning. The JGate-AttResUNet, with its attention-based enhancements, delivers 0.9499 accuracy. Notably, the EfficientNet as an encoder for UNet outperforms all others with an exceptional accuracy of 0.9925, demonstrating the power of combining efficient feature extraction with the UNet architecture.

EfficientNetB6 serves as a potent encode for U-Net models, offering several compelling strengths across various image segmentation tasks.

- Firstly, its advanced architecture facilitates efficient feature extraction, capturing high-quality features crucial for precise segmentation. This capability is complemented by its scalability, allowing it to handle



**FIGURE 21.** Comparative analysis against other state-of-the-art segmentation methods.

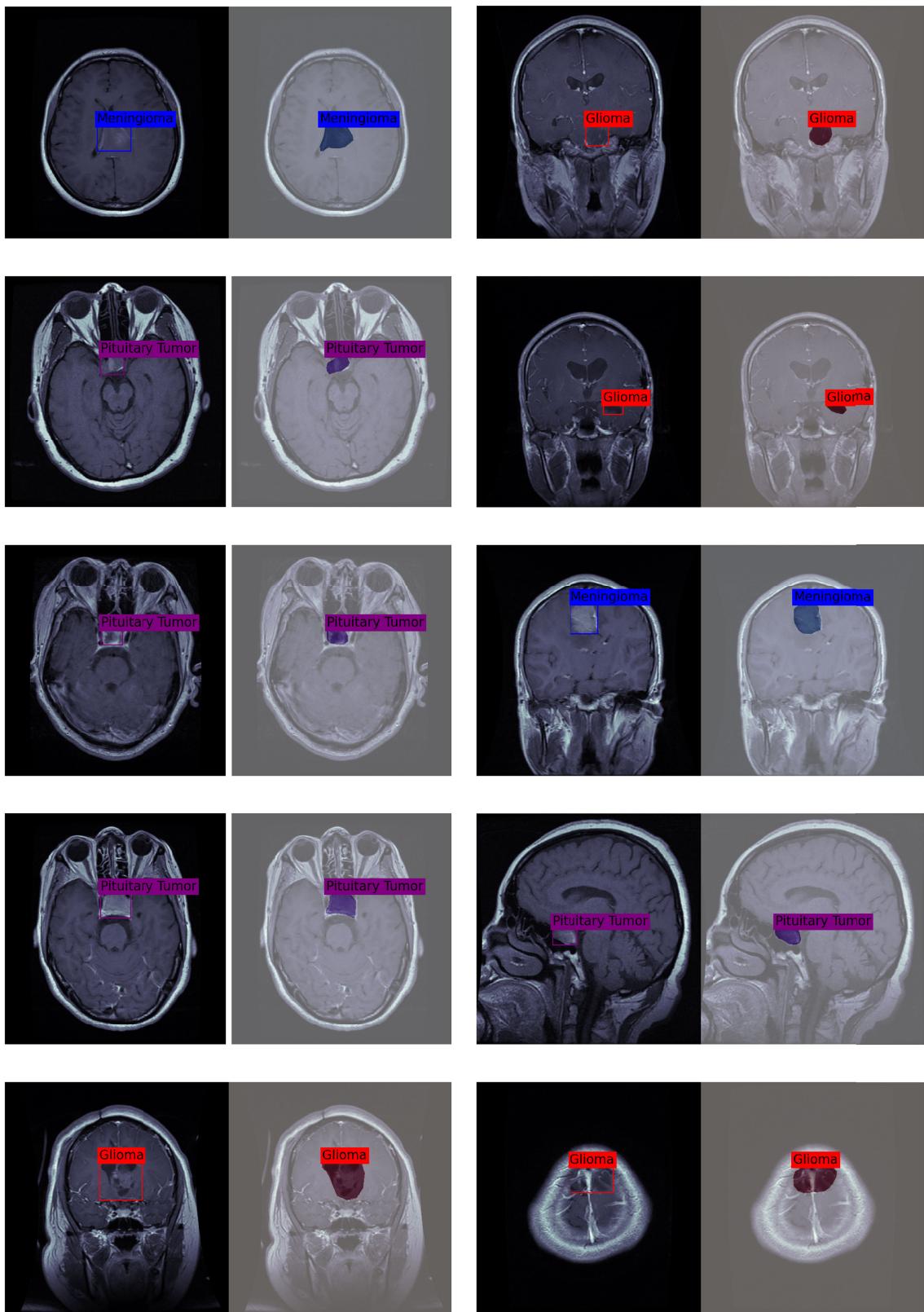
- diverse datasets and image resolutions effectively without a significant increase in computational demand.
- Moreover, leveraging EfficientNetB6's pretrained weights from datasets like ImageNet enhances training efficiency through transfer learning. This approach not only accelerates convergence but also ensures robust feature reuse, particularly beneficial in scenarios with limited annotated data. In practice, these advantages translate into superior performance metrics, often achieving state-of-the-art results in segmentation benchmarks.
  - EfficientNetB6 is also renowned for its efficiency, meticulously balancing model parameters and computational operations. This efficiency not only accelerates training times but also supports faster inference, making it well-suited for applications requiring real-time or resource-efficient segmentation tasks.
  - Adaptability is another notable strength, as EfficientNetB6 can be tailored to diverse segmentation challenges across different domains. Its multi-scale feature extraction capabilities prove invaluable for accurately delineating objects of varying sizes within complex images.
  - Robustness is further ensured through advanced techniques like squeeze-and-excitation, which enhances model generalization, and effective regularization methods such as drop connect and BatchNorm. These attributes collectively fortify EfficientNetB6's performance stability across diverse datasets and operational environments.
  - Integration with U-Net architecture seamlessly combines EfficientNetB6's robust feature extraction capabilities with U-Net's effective reconstruction and segmentation prowess. This integration, bolstered by skip connections that merge low-level and high-level features, optimally enhances segmentation accuracy and performance.

In essence, EfficientNetB6 as an encoder for U-Net models represents a formidable choice, leveraging advanced architectural design, transfer learning benefits, efficiency gains, adaptability across domains, robustness through regularization, and seamless integration with U-Net architecture to deliver state-of-the-art performance in image segmentation tasks.

In the context of brain tumor segmentation, EfficientNet brings distinct advantages due to its ability to efficiently capture both global and local features from MRI images. The architecture's compound scaling approach allows it to achieve high performance without excessively increasing computational requirements, which is crucial when working with large datasets of medical images. By integrating EfficientNet as the encoder in the U-Net framework, we leverage its power to extract more discriminative features while maintaining efficiency in processing. This results in a better representation of the tumor regions, particularly in the case of subtle or irregularly shaped tumors, which can often be challenging for traditional segmentation models.

Despite its strengths, EfficientNetB6 as an encoder in U-Net models does have some limitations that need to be considered in practical applications. One notable limitation is related to its computational intensity, especially when used in scenarios requiring real-time processing or deployment on resource-constrained devices. EfficientNetB6's deep and complex architecture, designed to balance depth, width, and resolution, can lead to higher computational demands during both training and inference phases. This may necessitate optimizations in hardware resources or model architecture adjustments to achieve satisfactory performance without compromising efficiency.

Another limitation lies in its training requirements, particularly when fine-tuning on domain-specific datasets. While transfer learning from pretrained weights generally accelerates convergence and enhances feature extraction



**FIGURE 22.** Prediction for tumor classes.

capabilities, EfficientNetB6's robustness heavily relies on the quality and representativeness of the initial training data.

In cases where domain-specific nuances or data distribution differ significantly from the pretrained dataset (e.g., medical

imaging or specialized satellite imagery), additional data augmentation or transfer learning strategies may be required to ensure optimal performance.

## VII. CONCLUSION

In conclusion, medical image segmentation, particularly in brain tumor analysis using multimodal MRI scans, is pivotal for clinical diagnosis and treatment planning. The challenges posed by similar intensity patterns, diverse tumor shapes, and indistinct boundaries underscore the complexity of this task. Traditional CNN-based segmentation models like UNet struggle with capturing comprehensive long-range dependencies critical for detailed predictions in such scenarios. This study addresses these limitations by integrating EfficientNet as an encoder within the UNet framework, leveraging its efficient feature extraction capabilities. EfficientNet's architecture, designed to balance depth, width, and resolution scaling, enhances the model's ability to extract meaningful features from MRI scans. By initializing the encoder with pretrained weights from large-scale datasets like ImageNet, the model effectively learns rich representations of tumor features, crucial for accurate segmentation. The integration of EfficientNet's features with UNet's bottleneck and skip connections ensures spatial integrity and enhances segmentation accuracy across meningioma, glioma, and pituitary tumor classes. The model architecture allows for effective information flow between encoder and decoder stages, enabling the preservation and refinement of fine-grained details in tumor boundaries. Experimental validation on the Brain-Tumor.npy dataset, comprising 3064 T1-weighted contrast-enhanced MRI images from 233 patients, demonstrates the proposed model's superior performance. It achieves an exceptional accuracy of .9925 and a low loss of 0.2999, surpassing benchmarks set by traditional CNN models. These results underscore the model's potential to advance clinical insights and patient care in medical imaging applications, promising improved diagnostic accuracy and personalized treatment planning for brain tumor patients. Overall, the EfficientNet-enhanced UNet framework represents a robust approach to address the complexities of brain tumor segmentation, offering clinicians more reliable tools for accurate diagnosis and treatment guidance in neuroimaging.

## REFERENCES

- [1] D. N. Louis, A. Perry, G. Reifenberger, A. von Deimling, D. Figarella-Branger, W. K. Cavenee, H. Ohgaki, O. D. Wiestler, P. Kleihues, and D. W. Ellison, “The 2016 world health organization classification of tumors of the central nervous system: A summary,” *Acta Neuropathologica*, vol. 131, no. 6, pp. 803–820, May 2016, doi: [10.1007/s00401-016-1545-1](https://doi.org/10.1007/s00401-016-1545-1).
- [2] S. Bauer, R. Wiest, L.-P. Nolte, and M. Reyes, “A survey of MRI-based medical image analysis for brain tumor studies,” *Phys. Med. Biol.*, vol. 58, no. 13, pp. R97–R129, Jun. 2013, doi: [10.1088/0031-9155/58/13/r97](https://doi.org/10.1088/0031-9155/58/13/r97).
- [3] K. Held, E. R. Kops, B. J. Krause, W. M. Wells, R. Kikinis, and H.-W. Müller-Gartner, “Markov random field segmentation of brain MR images,” *IEEE Trans. Med. Imag.*, vol. 16, no. 6, pp. 878–886, Dec. 1997, doi: [10.1109/42.650883](https://doi.org/10.1109/42.650883).
- [4] S. Taheri, S. H. Ong, and V. F. H. Chong, “Level-set segmentation of brain tumors using a threshold-based speed function,” *Image Vis. Comput.*, vol. 28, no. 1, pp. 26–37, Jan. 2010, doi: [10.1016/j.imavis.2009.04.005](https://doi.org/10.1016/j.imavis.2009.04.005).
- [5] T. Weglinski and A. Fabijanska, “Brain tumor segmentation from MRI data sets using region growing approach,” in *Proc. Perspective Technol. Methods MEMS Design*, May 2011, pp. 185–188. [Online]. Available: <https://ieeexplore.ieee.org/document/5960339/>
- [6] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proc. Medical Image Computing Computer-Assisted Intervention (MICCAI)*, in Lecture notes in computer science, 2015, pp. 234–241, doi: [10.1007/978-3-319-24574-4](https://doi.org/10.1007/978-3-319-24574-4).
- [7] M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, and N. Z. Jhanjhi, “A hybrid CNN-SVM threshold segmentation approach for tumor detection and classification of MRI brain images,” *IRBM*, vol. 43, no. 4, pp. 290–299, Aug. 2022, doi: [10.1016/j.irbm.2021.06.003](https://doi.org/10.1016/j.irbm.2021.06.003).
- [8] T. R. Mahesh, M. Gupta, T. A. Anupama, and O. Geman, “An XAI-enhanced efficientNetB0 framework for precision brain tumor detection in MRI imaging,” *J. Neurosci. Methods*, vol. 410, Oct. 2024, Art. no. 110227.
- [9] J. Sachdeva, D. Sharma, and C. K. Ahuja, “Multiscale segmentation net for segregating heterogeneous brain tumors: Gliomas on multimodal MR images,” *Image Vis. Comput.*, vol. 149, Sep. 2024, Art. no. 105191, doi: [10.1016/j.imavis.2024.105191](https://doi.org/10.1016/j.imavis.2024.105191).
- [10] M. Manthe, S. Duffner, and C. Lartizien, “Federated brain tumor segmentation: An extensive benchmark,” *Med. Image Anal.*, vol. 97, Oct. 2024, Art. no. 103270, doi: [10.1016/j.media.2024.103270](https://doi.org/10.1016/j.media.2024.103270).
- [11] S. Xun, Y. Zhang, S. Duan, M. Wang, J. Chen, T. Tong, Q. Gao, C. Lam, M. Hu, and T. Tan, “ARGA-unet: Advanced U-net segmentation model using residual grouped convolution and attention mechanism for brain tumor MRI image segmentation,” *Virtual Reality Intell. Hardw.*, vol. 6, no. 3, pp. 203–216, Jun. 2024, doi: [10.1016/j.vrih.2023.05.001](https://doi.org/10.1016/j.vrih.2023.05.001).
- [12] J. Zhu, C. Gu, L. Wei, H. Li, R. Jiang, and F. R. Sheykhhahmad, “Brain tumor recognition by an optimized deep network utilizing ammended grasshopper optimization,” *Heliyon*, vol. 10, no. 7, Apr. 2024, Art. no. e28062, doi: [10.1016/j.heliyon.2024.e28062](https://doi.org/10.1016/j.heliyon.2024.e28062).
- [13] G. Zhang, J. Zhou, G. He, and H. Zhu, “Deep fusion of multi-modal features for brain tumor image segmentation,” *Heliyon*, vol. 9, no. 8, Aug. 2023, Art. no. e19266, doi: [10.1016/j.heliyon.2023.e19266](https://doi.org/10.1016/j.heliyon.2023.e19266).
- [14] Y. Diao, F. Li, and Z. Li, “Joint learning-based feature reconstruction and enhanced network for incomplete multi-modal brain tumor segmentation,” *Comput. Biol. Med.*, vol. 163, Sep. 2023, Art. no. 107234, doi: [10.1016/j.combiomed.2023.107234](https://doi.org/10.1016/j.combiomed.2023.107234).
- [15] B. Sandhiya and S. Kanaga Suba Raja, “Deep learning and optimized learning machine for brain tumor classification,” *Biomed. Signal Process. Control*, vol. 89, Mar. 2024, Art. no. 105778, doi: [10.1016/j.bspc.2023.105778](https://doi.org/10.1016/j.bspc.2023.105778).
- [16] F. Zulfiqar, U. Ijaz Bajwa, and Y. Mehmood, “Multi-class classification of brain tumor types from MR images using EfficientNets,” *Biomed. Signal Process. Control*, vol. 84, Jul. 2023, Art. no. 104777, doi: [10.1016/j.bspc.2023.104777](https://doi.org/10.1016/j.bspc.2023.104777).
- [17] B. V. Isunuri and J. Kakarla, “EfficientNet and multi-path convolution with multi-head attention network for brain tumor grade classification,” *Comput. Electr. Eng.*, vol. 108, p. 108700, May 2023, doi: [10.1016/j.compeleceng.2023.108700](https://doi.org/10.1016/j.compeleceng.2023.108700).
- [18] P. Agrawal, N. Katal, and N. Hooda, “Segmentation and classification of brain tumor using 3D-UNet deep neural networks,” *Int. J. Cognit. Comput. Eng.*, vol. 3, pp. 199–210, Jun. 2022, doi: [10.1016/j.ijcce.2022.11.001](https://doi.org/10.1016/j.ijcce.2022.11.001).
- [19] G. Neelima, D. R. Chigurukota, B. Maram, and B. Girirajan, “Optimal DeepMRSeg based tumor segmentation with GAN for brain tumor classification,” *Biomed. Signal Process. Control*, vol. 74, Apr. 2022, Art. no. 103537, doi: [10.1016/j.bspc.2022.103537](https://doi.org/10.1016/j.bspc.2022.103537).
- [20] P. S. Bidkar, R. Kumar, and A. Ghosh, “SegNet and salp water optimization-driven deep belief network for segmentation and classification of brain tumor,” *Gene Expression Patterns*, vol. 45, Sep. 2022, Art. no. 119248, doi: [10.1016/j.gep.2022.119248](https://doi.org/10.1016/j.gep.2022.119248).
- [21] E. Dandil and S. Karaca, “Detection of pseudo brain tumors via stacked LSTM neural networks using MR spectroscopy signals,” *Bio-cybernetics Biomed. Eng.*, vol. 41, no. 1, pp. 173–195, Jan. 2021, doi: [10.1016/j.bbce.2020.12.003](https://doi.org/10.1016/j.bbce.2020.12.003).
- [22] M. Masood, R. Maham, A. Javed, U. Tariq, M. A. Khan, and S. Kadry, “Brain MRI analysis using deep neural network for medical of Internet things applications,” *Comput. Electrical Eng.*, vol. 103, Oct. 2022, Art. no. 108386, doi: [10.1016/j.compeleceng.2022.108386](https://doi.org/10.1016/j.compeleceng.2022.108386).

- [23] S. Asif, M. Zhao, X. Chen, and Y. Zhu, "BMRI-NET: A deep stacked ensemble model for multi-class brain tumor classification from MRI images," *Interdiscipl. Sciences: Comput. Life Sci.*, vol. 15, no. 3, pp. 499–514, May 2023, doi: [10.1007/s12539-023-00571-1](https://doi.org/10.1007/s12539-023-00571-1).
- [24] S. Shanthi, S. Saradha, J. A. Smitha, N. Prasath, and H. Anandakumar, "An efficient automatic brain tumor classification using optimized hybrid deep neural network," *Int. J. Intell. Netw.*, vol. 3, pp. 188–196, Jan. 2022, doi: [10.1016/j.ijin.2022.11.003](https://doi.org/10.1016/j.ijin.2022.11.003).
- [25] B. S. Abd El-Wahab, M. E. Nasr, S. Khamis, and A. S. Ashour, "BTCfCNN: Fast convolution neural network for multi-class brain tumor classification," *Health Inf. Sci. Syst.*, vol. 11, no. 1, pp. 1–44, Jan. 2023, doi: [10.1007/s13755-022-00203-w](https://doi.org/10.1007/s13755-022-00203-w).
- [26] R. L. Kumar, J. Kakarla, B. V. Isunuri, and M. Singh, "Multi-class brain tumor classification using residual network and global average pooling," *Multimedia Tools Appl.*, vol. 80, no. 9, pp. 13429–13438, Jan. 2021, doi: [10.1007/s11042-020-10335-4](https://doi.org/10.1007/s11042-020-10335-4).
- [27] P. S. Tejashwini, J. Thriveni, and K. R. Venugopal, "A novel SLCA-UNet architecture for automatic MRI brain tumor segmentation," *Biomed. Signal Process. Control*, vol. 100, Feb. 2025, Art. no. 107047, doi: [10.1016/j.bspc.2024.107047](https://doi.org/10.1016/j.bspc.2024.107047).
- [28] S. F. Rabby, M. A. Arafat, and T. Hasan, "BT-Net: An end-to-end multi-task architecture for brain tumor classification, segmentation, and localization from MRI images," *Array*, vol. 22, Jul. 2024, Art. no. 100346, doi: [10.1016/j.array.2024.100346](https://doi.org/10.1016/j.array.2024.100346).
- [29] Y. Yang, L.-F. Yan, X. Zhang, Y. Han, H.-Y. Nan, Y.-C. Hu, B. Hu, S.-L. Yan, J. Zhang, D.-L. Cheng, X.-W. Ge, G.-B. Cui, D. Zhao, and W. Wang, "Glioma grading on conventional MR images: A deep learning study with transfer learning," *Frontiers Neurosci.*, vol. 12, pp. 1–10, Nov. 2018, doi: [10.3389/fnins.2018.00804](https://doi.org/10.3389/fnins.2018.00804).
- [30] J. Cheng, M. Gao, J. Liu, H. Yue, H. Kuang, J. Liu, and J. Wang, "Multimodal disentangled variational autoencoder with game theoretic interpretability for glioma grading," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 2, pp. 673–684, Feb. 2022, doi: [10.1109/JBHI.2021.3095476](https://doi.org/10.1109/JBHI.2021.3095476).
- [31] A. Younis, L. Qiang, C. O. Nyatega, M. J. Adamu, and H. B. Kauwua, "Brain tumor analysis using deep learning and VGG-16 ensembling learning approaches," *Appl. Sci.*, vol. 12, no. 14, p. 7282, Jul. 2022, doi: [10.3390/app12147282](https://doi.org/10.3390/app12147282).
- [32] T. Ruba, R. Tamilselvi, and M. Parisa Beham, "Brain tumor segmentation using JGate-AttResUNet—A novel deep learning approach," *Biomed. Signal Process. Control*, vol. 84, Jul. 2023, Art. no. 104926, doi: [10.1016/j.bspc.2023.104926](https://doi.org/10.1016/j.bspc.2023.104926).



**PRADEEP KUMAR TIWARY** received the B.Com. degree (Hons.) from Tilka Manjhi Bhagalpur University, in 1994, and the M.C.A. degree from Maharshi Dayanand University, in 2011. He is currently pursuing the Ph.D. degree in computer science with Galgotias University, Greater Noida, India. He was a Senior Manager of IT and an Assistant Professor with Manav Rachna International University, Faridabad. He is also an Assistant Professor and the General Manager of

IT with the School of Computing Science and Engineering, Galgotias University. He has been involved in IT infrastructure setup, data center smooth running, team management, security systems, computer laboratory management, vendor management, and teaching and training.



**PRASHANT JOHRI** received the B.Sc. (Hons.) and M.C.A. degrees from Aligarh Muslim University, Aligarh, in 1992 and 1995, respectively, and the Ph.D. degree in computer science from Jiwaji University, Gwalior, India, in 2011. He was a Professor and the Director (M.C.A.) of the Galgotias Institute of Management and Technology (GIMT) and Noida Institute of Engineering and Technology (NIET) Greater Noida. He is currently a Professor with the School of Computing Science and Engineering, Galgotias University, Greater Noida, India. He has supervised six Ph.D. students and many M.Tech. students for their thesis. He has published more than 200 scientific articles, including journal articles,

book chapters, and conference papers. He has published edited books in Elsevier and Springer. His research interests include artificial intelligence, machine learning, data science, information security and network security, blockchain, healthcare, agriculture, entrepreneurship, sustainable development, image processing, software reliability, and cloud computing. He is actively publishing in these areas. He has served as the chair for many conferences and affiliated as a member of the program committee in many conferences in India and abroad. Apart from scholarly contributions toward the scientific community, he organized several conferences/workshops/seminars at the national and international levels. He voluntarily served as a reviewer for various international journals and conferences.



**ALOK KATIYAR** (Member, IEEE) received the B.Sc. degree from Kanpur University, the M.C.A. degree from Agra University, in 1998, the M.Tech. degree from IP University, in 2009, and the Ph.D. degree from Sharda University. He is currently a Professor with the School of Computing Science and Engineering, Galgotias University, Greater Noida, India. He has over 26 years of rich experience in academia and research. His expertise spans across various domains, including cyber security, artificial intelligence, machine learning, and genetic algorithms. His extensive experience and contributions to these fields have established him as a leading authority and innovator in technology and research. He served as a coordinator at various national-level and state-level exams of NTU. He has supervised two Ph.D. students and ten M.Tech. students. He has published 38 scientific articles, including journal articles, book chapters, and conference papers. His research interests include AI, machine learning, data science, cybernetics, genetic algorithms, healthcare, agriculture, software reliability, and cloud computing. He has served as the chair for many conferences and is affiliated as a member of the program committee in many conferences wide span of India. He voluntarily served as a reviewer for various international journals and conferences.



**MAYUR KUMAR CHHIPA** (Member, IEEE) received the B.Tech. and M.Tech. degrees from Rajasthan Technical University, in 2011 and 2015, respectively, and the Ph.D. degree from KL University, in June 2023. His Ph.D. thesis was titled "Design and Analysis of 2D Photonic Crystal-Based Passive Devices for Photonic Integrated Circuits." He is currently a Senior Lecturer, the Head of the Program of Electronics and Communication Engineering (ECE), and the Head of the Open Distance and E-Learning (OdeL) Program at ISBAT University, Kampala, Uganda, East Africa. He is also the Founder Counsellor of the IEEE ISBAT University Student Branch, under the IEEE Uganda Section, Region 8. He is having an experience of nine and half years in teaching and research. He is the author or co-author of more than 30 research papers in referred SCI/SCOPUS journals and IEEE/OSA/SPIE/Springer/APC conference proceedings. He has attended about 30 international and national conferences around the world. He has attended around 20 workshops and organized 25 workshops. At ISBAT University teaching students from 24 different nation countries and international students from different cultures and diversity. His current research interests include electronics and communication, AI and ML in photonics, optical communication, photonic crystals, photonic band gap structures, optical devices, optical communication networks, and photonic integrated circuits. He is an active member of the IEEE Photonic Society, IES Society, and CASS Society. He was nominated and awarded for The Photonics 100, a people list by Electro Optics, in 2023.