Journal of Theoretical and Applied Information Technology
15th February 2026. Vol.104. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AL-BIRUNI EARTH RADIUS SPOTTED HYENA OPTIMIZER BASED TRANSFER LEARNING FOR BRAIN TUMOR CLASSIFICATION USING MRI

SRILAKSHMI ALURI[1]* , S. SAGAR IMAMBI[2]

[1]*Research Scholar, Department of CSE
K L Educational Foundation, Deemed to be a University
Vaddeswaram, Andhra Pradesh, 520 002, India

[2]Professor, Department of CSE
K L Educational Foundation, Deemed to be a University
Vaddeswaram, Andhra Pradesh, 520 002, India
E-mail: alurisrilaxmi@gmail.com

ABSTRACT

Brain tumor classification involves identifying and categorizing various types of tumors from Magnetic Resonance Imaging (MRI), a task critical for effective diagnosis and treatment planning. Existing methods often face challenges such as MRI noise, imprecise tumor localization, limited feature extraction, and high computational requirements, which can reduce classification accuracy and reliability. To address these challenges, an innovative technique named Al-Biruni Earth Radius Spotted Hyena Optimizer based Convolution Neural Network with Transfer Learning (BERSHO_CNN with TL) is proposed for classifying brain tumor. Initially, the pre-processing is carried out on MRI images using Non-Local Means (NLM) filtering. Tumor segmentation is then performed using the Multi-Attention Dense Network (MAD-Net). By utilizing the segmented images, feature extraction is conducted through a combination of 3D-Convolutional Autoencoder+Vision Transformer (3D-CAE+ViT). Finally, the brain tumors are classified using a proposed CNN with TL) model, which is trained using the BERSHO algorithm. BERSHO is developed by combining the Al-Biruni Earth Radius (BER) and the Spotted Hyena Optimizer (SHO). The proposed approach demonstrates enhanced classification accuracy, robust tumor localization, and improved feature representation compared to existing methods. The devised model has achieved a Negative Predictive Value (NPV) of 91.788%, accuracy of 92.847%, True Positive Rate (TPR) of 92.534%, Positive Predictive Value (PPV) of 92.277%, and True Negative Rate (TNR) of 91.747%. These findings indicate that the proposed framework provides a practical and efficient tool for automated brain tumor diagnosis, contributing new knowledge in integrating hybrid optimization with advanced feature extraction for improved MRI-based tumor classification.

Keywords: *Brain tumor classification, Magnetic Resonance Imaging, Al-Biruni Earth Radius, Spotted Hyena Optimizer, GoogLeNet.*

1.INTRODUCTION

Various medical imaging methods have been established to generate diagnostic images for various diseases. The most frequently utilized diagnostic tools include Positron Emission Tomography (PET), Ultrasonic Imaging (UI), X-Rays, Computed Tomography (CT), Single-Photon Emission Computed Tomography (SPECT), Magnetic Resonance Spectroscopy (MRS), and MRI [1]. In MRI, the patient is placed within a large magnet that creates a powerful exterior magnetic field. This field aligns the nuclei of various atoms in the body, together with hydrogen, with its direction. A Radio Frequency (RF) signal is then applied, and the energy released from the body is detected and used by a computer to construct the MR image [2]. Generally, MRI is known for its noninvasive nature that provides high-resolution 3D and 2D images of the human body, including the brain. Its exceptional image quality makes it widely used for cancer classification and identification [3]. Brain abnormalities, commonly termed as tumors in medical terms, are categorized into malignant or benign. Approximately 200 distinct types of brain tumors can be found in various regions of the brain, and these tumors can significantly affect

individuals' lives [4]. Precise recognition and categorization of brain tumors are essential for devising the correct treatment strategy. Brain tumor detection involves identifying the presence of tumors within the brain through a series of diagnostic procedures and imaging techniques. The process often starts with evaluating symptoms, such as persistent headaches, seizures, changes in vision or speech, and cognitive difficulties. Traditional tumor detection methods heavily depend on the visual examination of MRI images, which can be subjective and subject to variability between radiologists.

Brain tumors are one of the deadliest types of cancer, lasting for extended periods and causing considerable psychological distress for those affected. The care required for brain tumour-affected patients is typically more costly than for other cancers [5]. Tumors can differ greatly in shape, size, and type, and they can develop in various locations throughout the brain. They are categorized as either malignant, high-grade or low-grade, cancerous or non-cancerous, or benign [1]. The potential for damage to the human brain varies among different types of brain tumors. The World Health Organization (WHO) classifies them into nine kinds, dependent on the brain region they affect and the origin of the cells. Among these, Gliomas are a major type affecting the Central Nervous System and are further divided into categories, such as Astrocytes, Oligodendrocytes, Ependymal Cells, and Microglia [6]. Accurate brain tumor classification is crucial and can be life-saving for patients. An accurate and prompt diagnosis is crucial for creating an effective treatment plan, but tumor classification continues to be a complex and challenging process. In recent years, significant effort and time have been invested in enhancing tumor classification within MRI brain images. Various methods have been developed to achieve accurate classification outcomes. However, many brain tumor categorization systems are still in the experimental stage. Issues such as limited training data, poor image quality, insufficient image features, and inadequate tumor localization continue to impact the effectiveness of these classification techniques. To tackle these challenges, Deep Learning (DL) techniques are being increasingly utilized for brain tumor classification [7].

Approaches for classifying brain tumors are broadly categorized into Machine Learning (ML) and DL techniques. Based upon classification, ML-based systems typically require lengthy and costly manual feature extraction and segmentation. These processes often necessitate expert intervention to choose suitable algorithms, which can result in inconsistent performance, especially with large datasets. In contrast, DL algorithms, like Convolutional Neural Networks (CNNs) have become prominent for their capability to automate feature extraction and segmentation tasks, offering enhanced reliability and efficiency. Features are autonomously extracted from training data by DL models, enhancing the effectiveness of the classification model [3]. The nodes in CNNs operate like neurons in the brain: they receive input data, process it through the network, and generate output results. Due to their excellent accuracy, CNNs have been applied in numerous studies for detecting tumors in brain MRI images [8]. Despite these benefits, CNNs also pose some significant disadvantages. They demand significant computational resources and large quantities of labelled data for testing and training, which can be both expensive and time-consuming. Moreover, CNNs are susceptible to overfitting, particularly when the training data is limited or does not adequately represent real-world variations. To address these challenges and further enhance classification accuracy, TL is often employed with CNNs [9]. TL leverages pre-trained models to apply knowledge from one task to new related tasks, thereby improving performance and reducing the need for extensive training data. Despite these issues, CNNs remain a powerful tool for detecting tumors in brain MRI images, with their accuracy and adaptability through TL [8].

Problem Statement

Accurate brain tumor classification from MRI is a critical task in clinical diagnosis and treatment planning. However, existing automated classification approaches continue to face significant challenges due to variations in MRI acquisition protocols, the presence of noise and artifacts, and the heterogeneous appearance of tumor tissues. These factors often lead to degraded image quality and ambiguous tumor boundaries, thereby affecting the reliability of classification outcomes. Many conventional ML and DL models rely on either handcrafted features or single deep architectures, which are insufficient to capture the complex spatial, volumetric, and contextual characteristics of brain tumors. Moreover, the absence of precise tumor localization results in the inclusion of irrelevant background information, increasing feature redundancy and reducing classification accuracy. Although TL has shown

promise in medical image analysis, its performance is often constrained by suboptimal parameter tuning and convergence issues when applied to highly complex and imbalanced MRI datasets. Additionally, gradient-based training methods commonly used in deep neural networks may become trapped in local optima, leading to reduced generalization performance. This limitation highlights the necessity for an effective optimization strategy capable of enhancing learning efficiency and classification robustness. Therefore, there exists a clear need for an integrated brain tumor classification framework that effectively suppresses MRI noise while preserving structural details, accurately segments tumor regions to focus on clinically relevant information, extracts discriminative spatial and contextual features, and optimizes the training of deep classification models to achieve reliable and consistent performance.

The core concern of this research is to overcome the limitations of existing automated brain tumor detection and classification methods that struggle with noise sensitivity, imprecise tumor localization, insufficient feature representation, and inefficient training strategies. Brain tumor classification is selected as the focus of this research because it demands a high level of diagnostic accuracy and robustness, making it an ideal and impactful application for advanced deep learning and optimization techniques.

Contribution

The paper presents an innovative approach called BERSHO_CNN with TL for classifying brain tumors. Primarily, the input MRI image is collected, and it is pre-processed utilizing an NLM filter. Subsequently, tumor segmentation is performed by utilizing MAD-Net. Then, the feature extraction is implemented through 3D-CAE+ViT. Finally, the brain tumor is classified into three different forms, like Meningiomas, Pituitary tumors, and Glioma tumors, using a proposed CNN with TL method, which is trained using the BERSHO optimization method. The BERSHO is developed by incorporating the Al-Biruni Earth Radius (BER) and the Spotted Hyena Optimizer (SHO).

Proposed BERSHO_CNN with TL for classifying brain tumor utilizing MRI images: To perform brain tumor classification, the devised BERSHO_CNN with TL method is designed. The CNN with TL is trained utilizing the BERSHO optimization, which is designed by combining the BER and SHO.

The core concern of this research is to achieve accurate and reliable brain tumor classification from MRI images despite challenges such as imaging noise, tumor heterogeneity, complex anatomical structures, and limitations of conventional training strategies. To address these challenges, a set of carefully selected methods is integrated, each contributing distinct advantages to the overall framework. The NLM Filter employed during the preprocessing stage helps in suppressing the noises inherent in MRI images while preserving fine structural details. Unlike conventional local filters, NLM exploits the redundancy of image patterns across non-local regions, making it particularly effective in maintaining tumor boundaries and tissue textures that are critical for accurate segmentation and feature extraction. Also, the MAD-Net selected for tumor segmentation mainly focuses on relevant spatial and channel-wise features simultaneously. The dense connectivity improves feature reuse and gradient flow, while the attention mechanism enhances tumor region localization by emphasizing salient tumor characteristics. This leads to precise tumor segmentation and reduces background interference, directly improving classification reliability. The use of 3D-CAE combined with ViT for feature extraction is effective in capturing the volumetric and spatial features from MRI scans, preserving inter-slice contextual information that is often lost in 2D approaches. The CNN with TL employed for the classification process leverages the knowledge from pre-trained models, thereby improving generalization performance when training data is limited. The BERSHO used for improving the classification performance combines the exploration capability of the BER algorithm with the exploitation strength of the SHO. This hybrid optimization strategy effectively avoids local optima, ensures better parameter tuning, and improves convergence stability compared to traditional gradient-based optimization methods. As a result, the CNN with TL achieves higher accuracy and robustness in brain tumor classification. Thus, the proposed BERSHO_CNN with TL framework can achieve higher accuracy and robust brain tumor classification compared to existing methods by effectively addressing MRI noise, tumor variability, and feature representation limitations.

Structural Organization

The paper organizes its various sections in the mentioned form. Section 2 discusses the challenges encountered by traditional methods and

includes a literature review. Section 3 describes the process involved in the devised method, along with its architecture. Section 4 examines the performance and results attained by the devised method. Section 5 presents the conclusion and the future research direction of the paper.

2. LITERATURE SURVEY

Majib, M.S. *et al.* [10] developed the VGG Stacked Classifier Network (VGG-SCNet) to identify brain tumors (BT) using MRI images. Although this method achieved high precision and recall, it required significant computational time. Khairandish, M.O., *et al.* [11] designed a Hybrid CNN and Support Vector Machine (SVM) approach for tumor detection and classification using MRI scans. The method utilized Maximally Stable Extremal Regions (MSER) to extract more effective features, which improved the classification performance. However, this approach had high execution times and struggled with accurately determining the precise location and size of the tumor. Sharif, M.I., *et al.* [12] introduced the Densenet201 Pre-Trained Deep Learning Model for multimodal brain tumor classification. This approach used a Modified Genetic Algorithm (MGA) technique and an Entropy-Kurtosis-based High Feature Values (EKbHFV) technique for feature selection. While this method achieved high classification accuracy and shorter execution times, it encountered significant computational challenges. Islam, M.M., *et.al* [13] developed an EfficientnetB7 for classifying brain tumor utilizing MRI images. The model incorporated comprehensive preprocessing and augmentation approaches to enhance performance. Additionally, it utilized a novel deep learning approach within the EfficientNet family to enhance brain tumor classification and detection. However, this introduced model struggled with resource-constrained clinical environments due to its computational requirements.

Aamir, M., et.al [1] designed a ResNet50 model to achieve brain tumor categorization. The method included preprocessing steps, such as illumination enhancement and nonlinear stretching, to enhance the visual excellence of MRI images. The technique automated the classification and detection of brain tumors, reducing dependence on radiologists' expertise and minimizing human error. However, it faced challenges with handling large datasets, which could be computationally expensive and may have limited the model's efficiency. Asiri, A.A., et.al [4] developed a Radial Basis Function (RBF) neural network for enhancing the classification of brain tumor. The model was validated on various types of brain tumors, demonstrating significant enhancements in classification and contrast efficiency. Although it employed neural networks, independent component analysis, and adaptive Wiener filtering for enhanced image processing, the model faced challenges due to its computational intensity, requiring substantial processing power. Raza, A., et.al [14] introduced a CNN model for classifying brain tumor. The model utilized a hybrid deep learning approach with a modified GoogLeNet architecture, which enhanced classification and feature extraction. The method was capable of classifying meningioma, glioma, and pituitary tumors. However, the model was highly dependent on the measure and superiority of the testing and training data, which affected its overall performance and generalizability. Almalki, et.al [8] designed a CNN deep feature-trained support vector machine model for brain tumor classification, which exhibited high accuracy and adaptability, making it a promising candidate for real-time clinical applications. However, it required significant training time.

Many existing approaches for brain tumor classification using MRI, such as VGG-SCNet, Hybrid CNN–SVM with MSER, DenseNet201, EfficientNetB7, ResNet50, RBF neural networks, and modified GoogLeNet models shown promising results. However, despite their achievements, these methods face notable challenges. Some models, like VGG-SCNet, DenseNet201, and EfficientNetB7, deliver high accuracy but are computationally intensive, making them difficult to use in real-time clinical settings. Hybrid CNN–SVM and modified GoogLeNet architectures improve feature extraction and classification; however, they often struggle to precisely locate tumor regions and determine tumor size, which can include unnecessary background information and reduce reliability. Pretrained models such as ResNet50 and RBF neural networks reduce reliance on manual intervention but require large datasets and substantial computational resources. Even hybrid CNN–SVM approaches that combine deep features with SVM classifiers need long training times and are sensitive to the quality of the data. Overall, most of these methods depend on conventional training and optimization techniques, which can result in suboptimal learning, lower robustness to MRI noise, and limited generalization. These limitations clearly show the need for a more robust and efficient brain tumor classification framework that can handle noise, accurately segment tumors, extract meaningful features, and

optimize model training for reliable and clinically useful performance.

3. PROPOSED SYSTEM FOR CLASSIFYING THE BRAIN TUMOR USING BERSHO_CNN WITH TL

In medical imaging, achieving timely and precise analysis of brain tumors using MRI continues to be a major challenge. Early detection is crucial as it can greatly reduce the mortality rate associated with brain cancer. MRI is favored for its low ionizing radiation compared to other imaging techniques, but manual inspection is often time-consuming. This research introduces a novel method called BERSHO_CNN with TL for categorizing the brain tumors. The process begins with preprocessing the MRI images through NLM [15] to remove noise. Tumor segmentation is then performed using the MAD-Net [16]. Next, features are extracted using 3D-CAE+ViT, which is the combination of 3-D CAE [17] and ViT [18]. Finally, classification of brain tumors is conducted using a CNN with TL [19], which incorporates hyperparameters from the GoogLeNet [20] model. The CNN with TL is trained using the BERSHO optimization, which is the combination of BER [21] and SHO [22] techniques. The proposed model classifies the detected tumor into meningiomas, pituitary and glioma. Figure 1 shows the block diagram of the devised BERSHO_CNN with TL for categorizing the brain tumor employing MRI images.

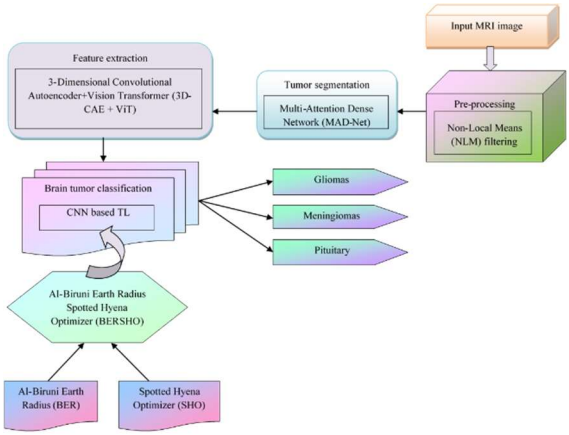

Figure 1: Block diagram of BERSHO_CNN with TL for brain tumor classification

3.1 Image Acquisition

The MRI image utilized for brain tumor classification is acquired from the specified dataset [23], which is mathematically expressed as,

$$S = \{S_1, S_2, \ldots, S_r, \ldots, S_t\} \quad (1)$$

Here, the Figshare database utilized for classifying brain tumor is denoted by S , S_t denotes the total 3064 MRI images from 233 patients and S_r represents the r^{th} MRI image, which is employed as input for subsequent tasks. The specified database comprises three dissimilar categories of brain tumors, like pituitary, meningioma, and glioma tumors.

3.2 Pre-Processing Utilizing NLM Filter

Pre-processing denotes the task of preparing raw MRI data for subsequent analysis by transforming it into a more suitable format for subsequent processing. To perform pre-processing, the NLM filter is utilized as it removes the noise from the MRI image S_r . The NLM filter [15] is a non-linear, edge-preserving method that determines each output pixel by averaging input pixels with different weights. These weights are obtained from a broad area of the input image, known as "non-local." A primary characteristic of the NLM filter is that the weights are calculated by evaluating the similarity within minor image patches. This technique enhances MRI images by reducing noise while preserving important details. It operates on the principle that similar patches within an image should be averaged to minimize noise without blurring the edges. By utilizing this technique, the NLM filter enhances the overall quality of MRI images, resulting in clearer images that are better suited for subsequent tasks like tumor segmentation. The mathematical expression of the NLM filter is given below.

$$A(d) = \frac{1}{Z(d)} \sum_{e \in P(d)} A(d)\xi(d,e) \quad (2)$$

$$L(d,e) = \frac{1}{(2g+1)^2} \sum_{p \in N(0)} \xi(d+p, e+p) \quad (3)$$

Here, $A(d)$ represents the fitter value of pixel d , $Z(d)$ denotes the normalization factor, $P(d)$ specifies the squared neighborhood of pixel d , $\xi(d,e)$ indicates the weight contribution of the pixel d and e , p denotes the offset of the pixel and

g presents the patch value. The output produced by the NLM filter is specified by P_r.

3.3 Tumor Segmentation Utilizing MAD-Net

Tumor segmentation involves recognizing and extracting the tumor region from the input image to support a precise diagnosis and treatment strategy. In this context, MAD-Net is used for tumor segmentation. MAD-Net [16] utilizes an encoder-decoder architecture with skip concatenations. The encoder consists of three block types at each level: the Slide-Window Attention (SWA) component, a convolution layer, and the Dense Convolution (DC) component. Additionally, a Residual-Dual Attention (RDA) module is positioned between the decoder and the encoder. The network leverages skip concatenations between the encoder and decoder to preserve fine details and improve segmentation accuracy. It is designed to enhance the precision of segmentation by capturing and preserving fine-grained details of the tumor's structure.

3.3.1 DC module

The DC module in MAD-Net is designed to enhance gradient flow and enable more effective image representations to be learned by the network. To address computational and memory constraints, additional dense connections are employed instead of concatenating the dense connections. The final result is produced by combining the results from all preceding convolutional layers, with the Rectified Linear Unit (ReLU) and the Batch Normalization (BN) activation function integrated subsequently. Each DC module comprises three convolution layers with a stride of 1×1 and a kernel size of 3×3. The number of input channels is made to equal the number of output channels, with values of 32, 64, 128, and 256 being used across the four distinct encoder scales.

3.3.2 SWA module

To replicate the human visual attention system, channel attention is used to guide the network's focus toward relevant information and filter out unnecessary details. Existing channel attention modules, like Squeeze-and-Excitation (SE) and Convolutional Block Attention Module (CBAM), capture long-range dependencies across feature map positions through average or max-pooling layers. Coordination Attention (CA) improves upon this by encoding long-range dependencies in both vertical and horizontal directions. In contrast, the channel attention module captures four distinct long-range dependencies within each feature map. The input $M_{in} \in \Re^{A\times B\times C}$ processed through the SWA block to produce the output $H_{out} \in \Re^{A\times B\times C}$, which is denoted as.

$$H_s = e_s\left(cat\left[M_{in1}, M_{in2}, M_{in3}, M_{in4}\right]\right) \quad (4)$$

$$H_c = \lambda\left(e_c\left(H_s\right)\right) \quad (5)$$

$$H_f = \delta\left(e_f\left(H_c\right)\right) \quad (6)$$

$$H_{out} = H_f \times M_{in} \quad (7)$$

where, channel number is denoted by A, height and weight are represented using B and C, $M_{in1}, M_{in2}, M_{in3}, M_{in4}$ denotes the feature map in terms of pixels obtained from M_{in} and $cat(.)$ denotes the concatenation function. Here, $e_s(.)$ presents the average pooling layer, $e_c(.)$ specifies the convolution process through a reduction ratio, λ presents the activation function of ReLU and δ specifies the activation function of sigmoid.

3.3.3 RDA block

The Dual Attention (DA) network captures non-local contextual information by employing both spatial and channel attention modules. Each channel in the high-level feature maps from the encoders represents a unique response, making it crucial for the network to identify global information pertinent to the target regions. This is accomplished by generating a channel attention map that highlights the inter-dependencies among channel maps. Beginning with the high-level feature maps $M_{hl} \in \Re^{A\times B\times C}$, the process involves applying a 3 × 3 convolution, followed by normalization and ReLU activation. The output is then reshaped into $M_x \in \Re^{A\times R}$ and transposed to $M_y \in \Re^{R\times A}$, where $R = B\times C$. The attention map $E \in \Re^{A\times A}$ is subsequently computed as follows:

$$E_{y,x} = \frac{\exp\left(M_x \bullet M_y\right)}{\sum_{x=1}^{A}\exp\left(M_x \bullet M_y\right)} \quad (8)$$

The above equation can be divided into two steps. First, the matrix is computed by multiplying $M_x \in \Re^{A\times R}$ with $M_y \in \Re^{R\times A}$. The resulting matrix is then fed into the softmax layer to generate the attention map $E_{y,x} \in \Re^{A\times A}$, which captures the relationships between the x^{th} and y^{th} channels.

Finally, the output $H_{ca} \in \Re^{A \times A}$ is obtained by combining $E_{y,x}$ with M_x. It is denoted by,

$$H_{ca}{}_{y}^{z} = b_z \sum_{x=1}^{A} \left(E_{y,x} M_x \right) + M_y \quad (9)$$

where, H_{ca} represents the final feature obtained from each channel and b_z denotes the learnable weight. After computing CA, the SA map is computed using the average pooling and maximum pooling. The SA map is mathematically denoted using,

$$H_{sa} = \delta\left(cat\left[M_{\max.pool}, M_{avg.pool} \right] \right) \times M_{hl} \quad (10)$$

Here, $M_{\max.pool}$ and $M_{avg.pool}$ represent maximum and average pooling, respectively. The outcome of the RDA block is expressed as,

$$H_{hl} = Conv_{out}\left(H_{ca} + H_{sa} + M_{hl} \right) \quad (11)$$

where, M_{hl} represents the feature map, $Conv_{out}$ denotes the convolution followed by ReLu and BN. The tumor segmentation using MAD-Net utilizes pre-processed input P_r to produce a segmented result T_r. Figure 2 portrays the architecture of MAD-Net.

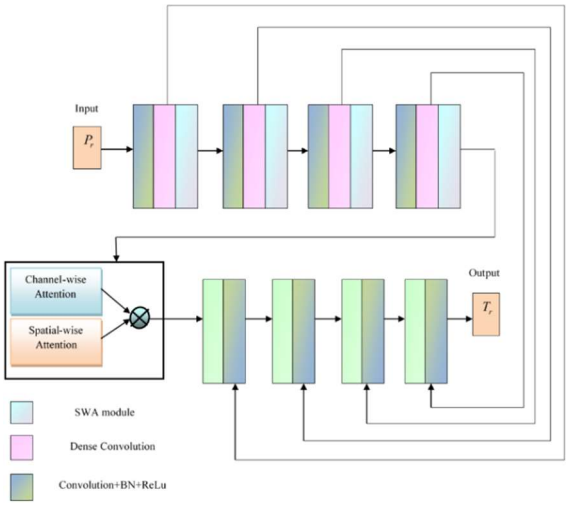

Figure 2: Architecture of MAD-Net

3.4 Feature Extraction Using 3D-CAE+Vit

Feature extraction refers to the method of identifying and obtaining relevant characteristics from raw images. This technique enhances the overall efficiency of the developed model by choosing the most relevant features from the input MRI images. Here, 3D-CAE+ViT is employed to perform feature extraction. The 3D-CAE+ViT helps to enhance feature extraction from 3D or spatiotemporal data by leveraging the strengths of both approaches. 3D-CAEs [17] excel at capturing spatial and temporal patterns by encoding volumetric data or sequences of frames, which helps in understanding detailed structures and relationships within the data. In contrast, ViT [18] use self-attention mechanisms to model global dependencies and complex interactions across different regions or time steps, providing a more integrated and comprehensive understanding. By combining 3D-CAEs with ViT, the model effectively merges detailed spatial-temporal feature extraction with a comprehensive global context. This integration enhances performance in tasks such as video analysis, medical imaging, and other applications involving complex 3D or spatiotemporal data.

3.4.1 3D-CAE

The 3D-CAEs [17] are neural network models designed for dimensionality reduction and representation learning across various tasks. Their innovation in image analysis arises from using convolutional layers to generate abstract representations of the original inputs, effectively reducing noise and eliminating redundant information. During feature extraction, convolutional and pooling layers are utilized by the encoder component of the CAE to progressively reduce the image resolution. Hierarchical features, such as textures, high-level patterns, and edges, are learned from the image by these layers. A compressed representation, which provides a lower-dimensional version of the input image, is produced as a result. The most important and distinctive information is captured while noise and irrelevant details are filtered out through dimensionality reduction. The feature vector size can be adjusted to achieve the anticipated level of compression. Here, 3D-CAE utilizes a segmented image T_r as an input to generate F_r as output.

3.4.2 ViT

The ViT [18] enhances feature extraction by dividing images into fixed-size patches and embedding them into a higher-dimensional space. It employs a Transformer encoder to analyze relationships between image patches and capture global dependencies, thereby learning detailed and hierarchical features from the visual data. The ViT utilizes F_r as an input. The standard Transformer is designed for processing 1D sequences of token embeddings. To adapt the ViT for 2D images, the image $q \in \Im^{T \times U \times G}$ is first divided into a sequence of

flattened 2D patches $q_n \in \Im^{E\times(N^2.G)}$. Here, (T,U) denotes the image resolution, G represents the total amount of channels, (N,N) is the size of each patch image and $E = TU/N^2$ is the total amount of channels, which also determines the length of the input sequence for the Transformer. A fixed-size latent vector is obtained from each flattened patch through a trainable linear layer, resulting in the generation of the patch embeddings.

Similar to BERT's [class] token, a learnable embedding $\left(i_0^0 = q_{class}\right)$ is added to the beginning of the sequence of embedded patches. The state of this embedding at the result of the Transformer encoder, $\left(i_K^0\right)$ represents the image J using Eq. (11). During both fine-tuning and pre-training, a cataloging head is added to i_K^0. This head comprises an MLP with one hidden layer for pre-training and a single linear layer for fine-tuning. Positional information is preserved by combining position embeddings with patch embeddings. This combined sequence is then fed into the encoder. The Transformer encoder is composed of successive layers that is associated between the MLP blocks and multiheaded self-attention. Layer Normalization (LN) is applied before residual connections, and each block is included after each block. The mathematical equations expressed in ViT are mentioned below.

$$i_0 = \left[q_{class}; q_n^1 Q; q_n^2 Q, \ldots; q_n^E Q\right] + Q_{pos}, \quad Q \in \Im^{(N^2.G)\times X}, E_{pos} \in \Im^{(E+1)\times X} \tag{12}$$

$$J = \ln\left(i_K^0\right) \tag{13}$$

Here, J represents the outcome of ViT and $\ln\left(i_K^0\right)$ denotes the LN applied at each block. The output obtained from the 3D-CAV is provided as input for ViT. The feature vector obtained from ViT are expressed as,

$$K_r = \{k_1, k_2, \ldots, k_t\} \tag{14}$$

Here, k_t denotes the total number of features obtained from 3D-CAE+ViT and K_r denotes the feature vector.

3.5 Brain Tumor Classification Using BERSHO_CNN With TL

Brain tumor classification includes the recognition and categorization of tumors using imaging, histological data, and molecular profiles. Brain tumor are categorized by utilizing the introduced BERSHO_CNN with TL. The BERSHO_CNN with TL method is designed to improve both diagnostic accuracy and efficiency for brain tumor classification. The BERSHO optimization algorithm enhances CNN with TL training by precisely optimizing parameters. CNNs are particularly proficient at analyzing medical images due to their capability to automatically extract hierarchical features from image data. This process is refined by TL through the usage of pre-trained models on huge datasets, enabling established knowledge to be applied to the specific process of brain tumor classification. BERSHO's effective global search capabilities boost the CNN's performance by enhancing its ability to accurately distinguish between different tumor types and grades. This integrated approach facilitates timely and reliable brain tumor diagnosis, leading to better patient outcomes and more informed treatment decisions.

3.5.1 TL

TL for brain tumor classification aims to leverage pre-trained models to significantly enhance diagnostic accuracy and efficiency. TL allows the CNN classifier to utilize learned features and patterns from extensive datasets, improving its ability to identify tumors effectively. This approach not only enhances accuracy by enabling the model to better recognize complex features in medical images but also reduces training time and computational resources, as the model benefits from pre-existing knowledge. GoogLeNet is utilized to train the CNN model. It assists as a robust feature extractor for new tasks, like brain tumor classification, by leveraging its pre-trained weights to identify complex features in medical images. This pre-trained knowledge allows for more efficient fine-tuning, where the model's final layers are adjusted to suit specific tasks while retaining valuable learned representations. The feature vector K_r is utilized as input to produce a parameter e. Figure 3 denotes the structure of GoogLeNet.

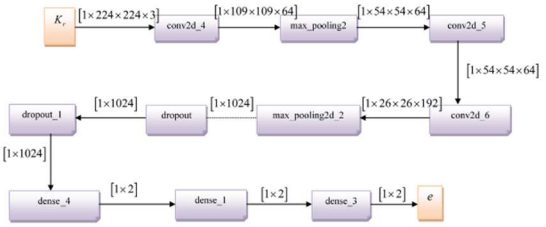

Figure 3: Structure of GoogLeNet

A CNN [19] is designed to map input data into outputs. It usually consists of convolutional layers, pooling layers and the softmax block that

generates a likelihood distribution for classification. In the context of image data, the initial convolutional layer often extracts basic features like edges, while subsequent layers identify more complex, higher-level features. Pooling layers decrease dimensionality by calculating either the maximum or average values within patches of the feature map from preceding layers. Activation layers, like ReLU, address the problem of saturation. Mathematically, a CNN can be described as a function using,

$$D = d_I d_{I-1} d_{I-2} \ldots d_2 d_1 \quad (15)$$

where, D represents the layer that receives the result from the preceding layer. Once trained, the CNN can function as a classifier, especially when employing TL. TL is a method that enhances DL performance by leveraging insights gained from a different but related task. Here, GoogLeNet [20] is used to train a CNN model to produce classified output. The feature vector K_r is employed as input for training the GoogLeNet, which produces a hyperparameter e. The hyperparameter produced by GoogLeNet is used as input for training the CNN model. GoogLeNet is a deep CNN architecture intended to improve accuracy while keeping computational demands constant by expanding both the depth and width of the network. Traditionally, enhancing deep neural network performance involves adding more layers (depth) and units per layer (width), which can significantly raise the computational and parameter resources required. GoogLeNet addresses challenges in deep learning through its inception modules, which utilize a parallel combination of convolutions to handle varying scales of features. These convolutions first reduce dimensionality before applying more computationally intensive operations. The architecture features nine inception modules and achieves a depth of 22 layers when only layers with parameters are considered. It also includes one average pooling layer, positioned before the classifier, five pooling layers, and four max pooling layers. Additionally, GoogLeNet incorporates a dropout block with a 70% dropout rate and employs ReLU activation functions throughout all convolutional layers, including those within the inception modules. Figure 4 displays the training process involved in the CNN with TL.

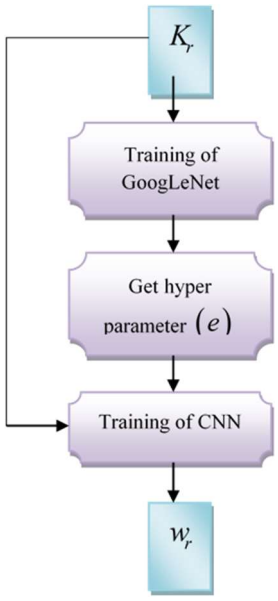

Figure 4: Training process of CNN with TL

Initially, the feature vector K_r obtained from 3D-CAE+ViT is selected as an input for training the Googlenet. After that, hyperparameters e is fetched from Googlenet. The CNN utilizes a feature vector K_r and hyper parameter e for the training process. At last, the CNN produce w_r as classified output. Figure 5 represents the structure of the CNN.

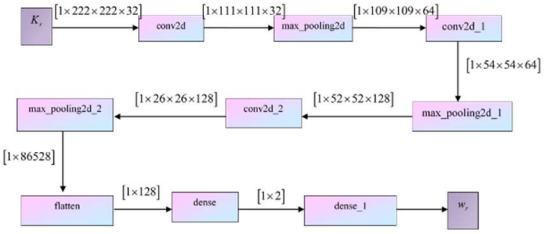

Figure 5: Structure of CNN

3.5.2 Tuning of CNN with TL utilizing hybrid BERSHO

The CNN with TL model utilizes the BERSHO optimization algorithm for training, which significantly enhances the model's capability to adjust its parameters both efficiently and accurately. The BERSHO is developed by integrating BER [21] and SHO [22]. The BER component is inspired by the precise calculations used in determining the Earth's radius, enhancing the optimization of model parameters with high accuracy. Meanwhile, the SHO component draws motivation from the hunting strategies and natural foraging of spotted hyenas, improving the algorithm's global search capabilities and convergence efficiency. This combination aims

to optimize the deep learning model more effectively, leading to improved classification accuracy, faster convergence, and enhanced performance in brain tumor diagnosis. Additionally, the BERSHO algorithm's adaptability to complex problem landscapes and its capability to escape local optima contribute to its effectiveness in refining deep learning models for more reliable and insightful diagnostic results.

a) Solution encoding

The optimal resolution within a specified search area V for an optimization problem is represented by solution encoding. It is mathematically represented by,

$$V = \left[1 \times \kappa\right] \tag{16}$$

Here, the learning parameter of the CNN with TL is represented by κ .

b) Fitness function

The optimal solution within a specified search area in an optimization algorithm is identified using the fitness function. For brain tumor classification, it measures how well the identified tumor regions correspond to the actual tumor locations in medical images, thus helping to select the most accurate and effective detection method. It is expressed as,

$$Y = \frac{1}{t}\sum_{r=1}^{t}\left[Z_r - w_r\right]^2 \tag{17}$$

Here, the result obtained from TL is denoted by w_r , the result of the fitness function is represented by Y and the targeted result is presented using Z_r .

c) Algorithmic steps for BERSHO

The Al-Biruni technique [21] involves measuring the angle of elevation from a known height above sea level, typically using astronomical observations of celestial bodies such as stars or the sun. Al-Biruni calculated the angle between the observer's line of sight to the horizon and the vertical line, and then applied spherical trigonometry to estimate the Earth's curvature. By combining this angle with the measurement of the observer's height, he was able to derive an estimate of the Earth's radius. In BER, individuals are divided into two sub-groups, each assigned to either exploration or exploitation. This division helps ensure a thorough investigation of the search space and prevents stagnation in local optima. Unlike many optimization approaches that require a shift to exploitation by all individuals after a few iterations, which may lead to local optima stagnation, BER maintains a set of search agents dedicated to exploration. If performance is observed to stagnate over three iterations, the number of entities focused on exploration is increased to enhance algorithm effectiveness.

Step 1: Initializing the population

Optimization algorithms aim to find the greatest resolution to a problem while adhering to a set of constraints. In BER, a vector can be used to represent an individual from the population. The vector is described below.

$$\vec{W} = \left\{W_1, W_2, \ldots W_j, \ldots, W_h\right\} \tag{18}$$

where, the parameter used for optimization problem is denoted using W_j and the candidate solution is represented using h .

Step 2: Evaluate fitness function

The fitness function denotes the quality or performance of each candidate solution, guiding the optimization process towards better results. The evaluation of the fitness function for BERSHO is already performed, and it is shown in Eq. (15).

Step 3: Exploration process

In BER, the population is divided into subcategories, with the size of each group dynamically adjusted to balance exploration and exploitation tasks. Initially, two groups are formed: one for exploration and one for exploitation. 70% of the population is assigned to the exploration group, while 30% is assigned to the exploitation group. Over time, the size of the exploration group is gradually reduced from 70% to 30% to improve the global average fitness of individuals. This adjustment helps to achieve significant improvements in fitness values. The role of the exploration group is to identify promising regions in the search space and to prevent stagnation at local optima by moving towards the optimal solution. Individuals in the exploration group focus on advancing toward the best solution by searching for promising areas around their current position in the search space and evaluating surrounding feasible alternatives for better fitness values. It is mathematically represented as follows.

$$l = p\frac{\cos(b)}{1-\cos(b)} \tag{19}$$

$$\vec{O} = \vec{l}_1.\left(\vec{W}(m) - 1\right) \tag{20}$$

$$\vec{W}(m+1) = \vec{W}(m) + \vec{O}.\left(2\vec{l}_2 - 1\right) \tag{21}$$

Here, p denotes the random number selected between $\left[0,2\right]$, $0 < b \le 180$, $\vec{W}(m)$ denotes the solution vector at m^{th} iteration, $\vec{l}_1$ and $\vec{l}_2$ denotes the coefficient vector and $\vec{O}$ specifies the circle

diameter that the search agent will use to identify promising areas.

Step 4: Exploitation process

For the exploitation task, the number of individuals starts at 30% of the population and is progressively increased to 70% over the course of optimization iterations. This dynamic adjustment is used to refine the solutions within the exploitation group. To ensure convergence of the optimization process, an elitism strategy is employed, retaining the best solution found if no superior solution is discovered. In the BER optimization algorithm, if the fitness of a resolution does not show significant improvement over three iterations, it may be considered as a local optimum. In such cases, a new exploratory individual is introduced using a mutation operation. The existing solutions are focused on improving by the exploitation group. During each cycle, the fitness values of all individuals are calculated, and the best one is identified. For exploitation, two methods are employed by BER: Heading in the direction of the best solution and investigating the area around the best solution.

The search agent moves in the direction of the best result, which is mathematically represented as,

$$\vec{W}(m+1) = l^2\left(\vec{W}(m) + \vec{O}\right) \quad (22)$$

$$\vec{O} = \vec{l}_3\left(\vec{F}(m) - \vec{W}(m)\right) \quad (23)$$

$$\vec{W}(m+1) = l^2\vec{W}(m) + l^2\vec{l}_3\vec{F}(m) - l^2\vec{l}_3\vec{W}(m) \quad (24)$$

$$\vec{W}(m+1) = \vec{W}(m)\left(l^2 - \vec{l}_3 l^2\right) + l^2\vec{l}_3\vec{F}(m) \quad (25)$$

where, $\vec{F}$ denotes the optimal solution vector, $\vec{O}$ represents the distance vector and $\vec{l}_3$ specifies the random vector.

From SHO [22], the standard formula of SHO is mentioned below.

$$\vec{g}(u+1) = \vec{g}_l - \vec{P}.\vec{C}_e \quad (26)$$

$$\vec{g}(u+1) = \vec{g}_l - \vec{P}\left|\vec{v}.\vec{g}_l(u) - \vec{g}(u)\right| \quad (27)$$

where, $\vec{g}_l(u)$ denotes the position of the prey, $\vec{C}$ presents the distance between the spotted hyena and the position of the prey, $\vec{g}(u+1)$ denotes the discovered location of the spotted hyena and the vector coefficients are denoted by $\vec{v}$ and $\vec{P}$. Let us consider the following equation,

$$\vec{v}.\vec{g}_l(u) > \vec{g}(u) \quad (28)$$

$$\vec{g}(u+1) = \vec{g}_l(u) - \vec{P}\vec{v}.\vec{g}_l(u) + \vec{P}\vec{g}(u) \quad (29)$$

$$\vec{g}(u+1) = \vec{g}_l(u)\left[1 - \vec{P}.\vec{v}\right] + \vec{P}\vec{g}(u) \quad (30)$$

Let us assume that,

$$\vec{g}(u+1) = \vec{W}(m+1) \quad (31)$$

$$\vec{g}(u) = \vec{W}(m) \quad (32)$$

$$\vec{g}_l(u) = \vec{W}_w(m) \quad (33)$$

The Eq. (30) can be rewritten as,

$$\vec{W}(m+1) = \vec{W}_w(m)\left[1 - \vec{P}.\vec{v}\right] + \vec{P}\vec{W}(m) \quad (34)$$

$$\vec{W}(m) = \frac{\vec{W}(m+1) - \vec{W}_w(m)\left[1 - \vec{P}.\vec{v}\right]}{\vec{P}} \quad (35)$$

By substituting Eq. (35) in Eq.(25), the Eq.(35) can be rewritten as,

$$\vec{W}(m+1) = \frac{\vec{W}(m+1) - \vec{W}_w(m)\left[1 - \vec{P}.\vec{v}\right]}{\vec{P}} \left(l^2 - \vec{l}_3 l^2\right) + l^2\vec{l}_3\vec{F}(m) \quad (36)$$

$$\vec{W}(m+1) - \frac{\vec{W}(m+1)\left(l^2 - \vec{l}_3 l^2\right)}{\vec{P}} = \frac{\vec{W}_w(m)\left[\vec{P}.\vec{v} - 1\right]\left(l^2 - \vec{l}_3 l^2\right) + l^2\vec{l}_3\vec{F}(m)}{\vec{P}} \quad (37)$$

$$\frac{\left(\vec{P} - l^2 + \vec{l}_3 l^2\right)\vec{W}(m+1)}{\vec{P}} = \frac{\vec{W}_w(m)\left[\vec{P}.\vec{v} - 1\right]\left(l^2 - \vec{l}_3 l^2\right) + l^2\vec{l}_3\vec{F}(m)}{\vec{P}} \quad (38)$$

The updated solution of the BER optimization is mentioned below.

$$\vec{W}(m+1) = \frac{\vec{W}_w(m)\left[\vec{P}.\vec{v} - 1\right]\left(l^2 - \vec{l}_3 l^2\right) + l^2\vec{l}_3\vec{F}(m)}{\vec{P}} \quad (39)$$

In the process of investigating the area around the optimal solution, the region surrounding the best solution (leader) is considered the most promising for attaining the optimal solution. Consequently, certain individuals are directed to explore the vicinity of this best solution, to uncover a superior solution. The expression is given below.

$$\vec{W}(m+1) = l\left(\vec{W}*(m) + \vec{f}\right) \quad (40)$$

$$\vec{f} = c + \frac{2\times m^2}{X^2} \quad (41)$$

Here, the best solution is represented by $\vec{W}*$, the random number within $[0,1]$ is represented by c, the iteration count is represented using m, and the total count of iterations is denoted by X.

Step 5: Mutation process

Mutation is an additional technique used by the BER. It acts as a genetic operator to introduce

and sustain diversity in the population. Essentially, mutation involves probabilistically making random alterations to one or more components in individuals. This approach helps to prevent early convergence by avoiding local optima and facilitating exploration of new, promising areas. Mutation is crucial for enhancing the BER's exploration capabilities and driving it towards discovering diverse and valuable solutions. It is mathematically denoted as follows.

$$\vec{W}(m+1) = \vec{f} * c^2 - p\frac{\cos(b)}{1-\cos(b)} \quad (42)$$

Step 6: Re-evaluation step

The BERSHO algorithm includes operations like exploration, exploitation and mutation. The primary iteration ends after updating all the positions of the agent using the three operations. The subsequent iteration completes after repeating the process from Eq. (19) to Eq. (42) for revising all the locations of the agent.

Step 7: Termination process

The iterations involved in the BERSHO algorithm end after updating all the locations of the agent and finding the optimal result for the problem. Algorithm 1 describes the pseudocode of the BERSHO algorithm.

Algorithm 1. Pseudocode for the BERSHO algorithm

SL. No	Pseudocode for BERSHO
1	Initialize the population size using Eq. (18)
2	Initialize the parameters involved in BER
3	Consider $m=1$
4	Evaluate fitness function using Eq. (17)
5	Discover optimal resolution $\vec{W}*$
6	While $m \le \max_{iter}$
7	do
8	For each resolution in the exploration phase
9	do
10	Perform moving in the direction of the best resolution utilizing Eq. (19), Eq. (20) and Eq. (40), subsequently.
11	End for
12	For each resolution in the exploitation phase
13	Exclusive selection of the best result using $\vec{O} = \vec{l}_2\left(\vec{F}(m) - \vec{W}(m)\right)$ and $\vec{W}(m+1) = \frac{\vec{W}_w(m)\left[\vec{P}.\vec{v}-1\right]\left(l^2 - \vec{l}_3 l^2\right) + l^2\vec{l}_3\vec{F}(m)}{\vec{P}}$
14	Detect the area around the best resolution using $\vec{f} = 1 + \frac{2\times m^2}{\max_{iter}^2}$ and $\vec{W}_2(m+1) = l\left(\vec{W}*(m) + \vec{f}\right)$
15	Match $\vec{W}(m+1)$ with $\vec{W}(m+1)$, then select the best result $\vec{W}*$
16	If the optimal fitness has remained unchanged over the last two iterations:
17	Then
18	Perform mutation operation using Eq. (42)
19	End if
20	End for
21	Revise fitness using Eq. (17)
22	End while
23	Provide best resolution

4. RESULTS AND DISCUSSION

The current section discusses the result attained by the BERSHO_CNN with TL model utilizing MRI images.

4.1 Experimental Setup

The BERSHO_CNN with TL model utilizes PYTHON as a tool for classifying brain tumor utilizing MRI input images.

4.2 Dataset Description

This brain tumor dataset [23] includes 3,064 T1-weighted contrast-enhanced images from 233 patients. It features three categories of brain tumors: 930 pituitary tumor images, 708 meningioma tumor images and 1,426 glioma tumor images. In addition, the dataset [24] contains multimodal scans in NIfTI format, including native T1-weighted images, post-contrast T1-weighted images (T1Gd), T2-weighted images, and T2 Fluid Attenuated Inversion Recovery (FLAIR) images. These images were acquired by utilizing different scanners and a range of clinical protocols.

4.3 Evaluation Metrics

The evaluation metrics, like TPR, NPV, PPV, TNR and accuracy are utilized to measure the performance improvement attained by the introduced BERSHO_CNN with TL model.

4.3.1 Accuracy

Accuracy [23] is a metric used to evaluate the performance of image segmentation models. It calculates the ratio of correctly classified pixels in an image to the total number of pixels. It is mathematically denoted by,

$$Accuracy = \frac{\Re_p + \Re_n}{\Re_p + \Re_n + \Im_p + \Im_n} \quad (43)$$

where, $\Re_n$ and $\Re_p$ represents true negative and true positive classes, $\Im_p$ implies the false positive class and $\Im_n$ indicates the false negative class.

4.3.2 NPV

NPV [25] is a statistical metric used to evaluate the performance of a diagnostic test or classification model. It assesses the ratio of true negatives out of all negative predictions made by the model. The NPV is statistically denoted using,

$$NPV = \frac{\Re_n}{\Re_n + \Im_n} \quad (44)$$

4.3.3 PPV

PPV [23], also known as precision, is a statistical metric used to evaluate the performance of a diagnostic test or classification model. It evaluates the ratio of true positives relative to all positive predictions made by the model. The equation of the PPV is mentioned below.

$$PPV = \frac{\Re_p}{\Re_p + \Im_p} \quad (45)$$

4.3.4 TNR

TNR [25], also known as specificity, is a metric used to assess the performance of a diagnostic test or classification model. It assesses the ratio of true negatives correctly identified by the model among all actual negatives. It is mathematically presented below.

$$TNR = \frac{\Re_n}{\Re_n + \Im_p} \quad (46)$$

4.3.5 TPR

TPR [25], also known as sensitivity or recall, is a metric used to evaluate the performance of a diagnostic test or classification model. It assesses the proportion of true positives that the model accurately identifies as positive among all actual positives. The equation of the TPR is mentioned below.

$$TPR = \frac{\Re_p}{\Re_p + \Im_n} \quad (47)$$

4.4 Experimental Results

Figure 6 outlines the result attained by the BERSHO_CNN with TL. Figure 6 a) and Figure 6 b) portray the input MRI image-1 and input MRI image-2. Figure 6 c) and Figure 6 d) show the pre-processed MRI image-1 and pre-processed MRI image-2. Figure 6 e) presents the segmented MRI image-1. Figure 6 f) denotes the segmented MRI image-2.

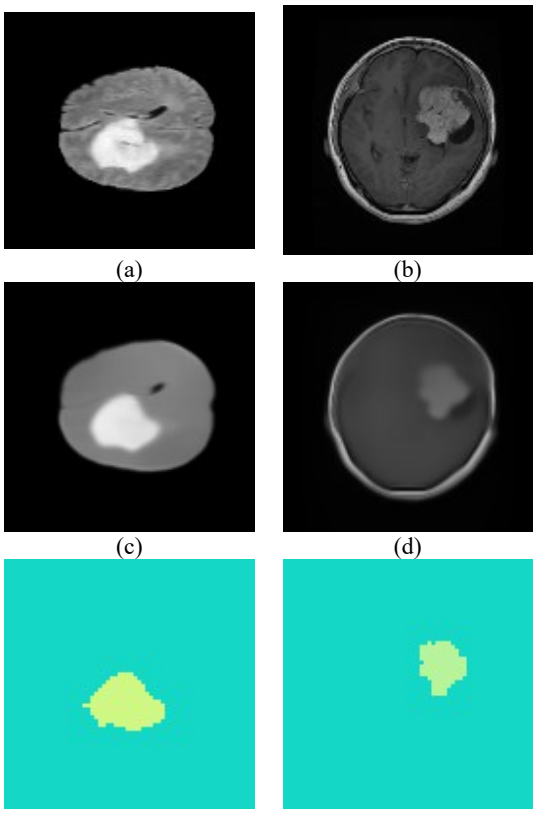

Figure 6: Experimental outcomes, a) Input MRI image-1, b) Input MRI image-2, c) Pre-processed MRI image-1, d) Pre-processed MRI image-2, e) Segmented MRI image-1, f) Segmented MRI image-2.

4.5 Comparative Methods

The conventional techniques, such as VGG stacked classifier [10], CNN+SVM [11], DenseNet 201[12] and EfficientNetB7 [13], are utilized to evaluate the performance of the devised method.

4.6 Comparative Assessment

The developed BERSHO_CNN with TL model is evaluated by changing the k-fold value and epoch. The performance of the developed method is evaluated by utilizing two dissimilar databases.

4.6.1 Analysis of BERSHO_CNN with TL utilizing database-1

The current section examines the performance of BERSHO_CNN with TL utilizing database-1. The assessment includes various evaluation metrics, such as TPR, NPV, TNR, PPV, and accuracy by modifying the k-fold value and epoch.

a) Evaluation of BERSHO_CNN with TL based on epoch

Figure 7 represents the assessment of BERSHO_CNN with TL based on epoch=25 to 100.

Figure 7 a) displays the accuracy attained by the designed BERSHO_CNN with TL model. When epoch is considered as 100, the developed model attained the accuracy of 93.722%, while the traditional schemes, such as VGG stacked classifier, CNN+SVM, DenseNet 201 and EfficientNetB7, GTLO-LeNet, Squeeze-KNet attained the accuracy of 85.111%, 87.226%, 88.328%, 88.823%, 89.799%, 90.982%, respectively. This exhibits that the proposed scheme attained the accuracy enhancement of 9.187%, 6.930%, 5.754%, 5.226%, 4.184% and 2.922% than the existing models. Figure 7 b) presents the NPV achieved by the introduced system. The proposed model achieved an NPV of 91.204%, whereas the traditional methods attained the NPV of 85.718%, 86.029%, 86.872%, 87.949%, 88.954% and 89.132%. This represents performance improvements of 6.014%, 5.673%, 4.749%, 3.568%, 2.466% and 2.271% over these existing methods. Figure 7 c) portrays the evaluation of PPV accomplished by the developed model. The traditional technique and devised technique attained the PPV of 85.864%, 86.671%, 87.253%, 87.510%, 88.577%, 89.708% and 91.720%. This presents the performance gain of 6.384%, 5.503%, 4.869%, 4.590%, 3.426% and 2.192%, when analyzed with the existing schemes. Figure 7 d) shows the performance of the BERSHO_CNN with TL model based on TNR. The devised scheme and existing scheme achieved the TNR of 92.351%, 87.125%, 88.464%, 88.782%, 89.038%, 89.566% and 90.287%. This indicates the performance gain of 5.659%, 4.209%, 3.865%, 3.587%, 3.015% and 2.235%. Figure 7 e) specifies the TPR attained by the introduced scheme. The devised model and conventional models achieved the TPR of 92.867%, 88.045%, 88.623%, 89.334%, 89.781%, 90.244% and 90.461%. This indicates the performance gain of 5.191%, 4.569%, 3.803%, 3.322%, 2.823% and 2.590%.

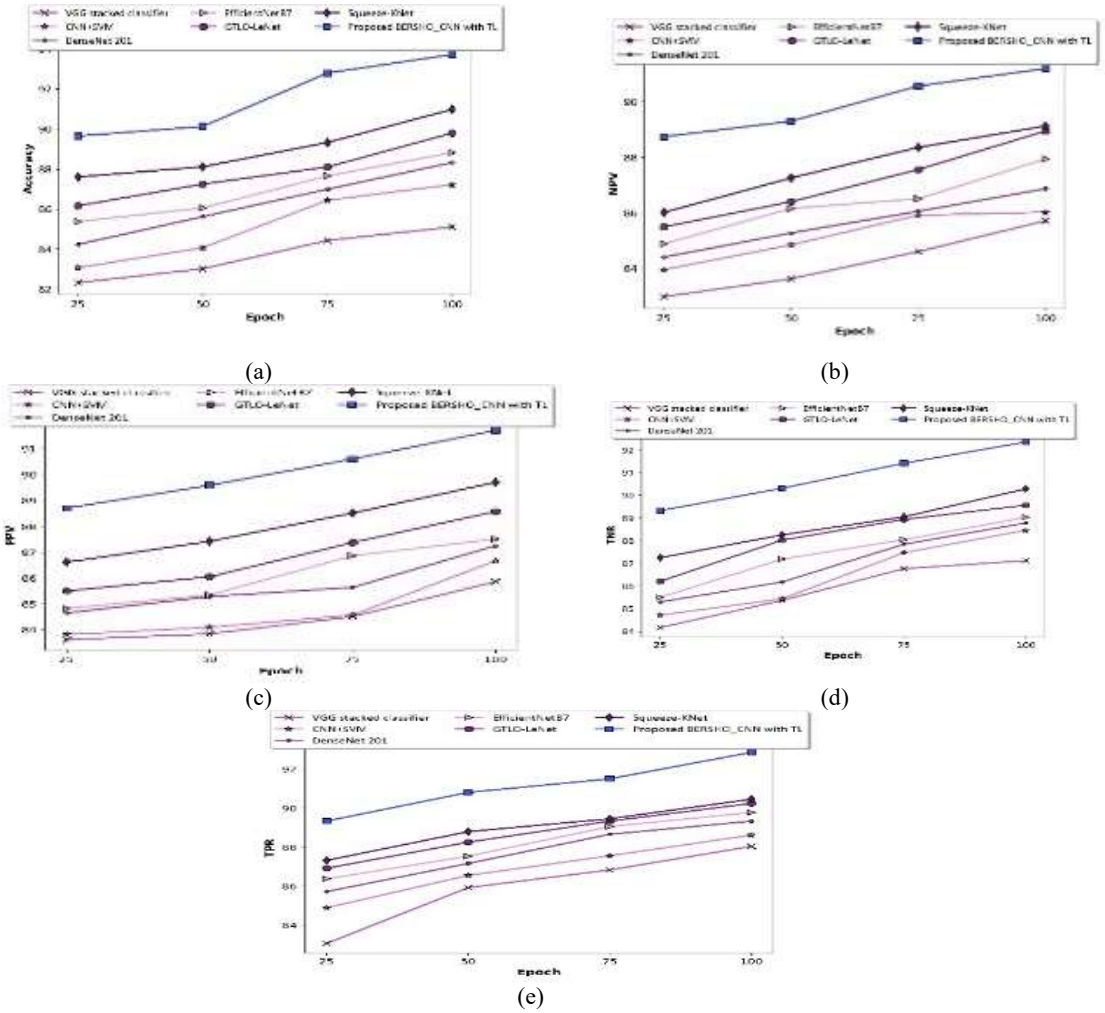

Figure 7: Evaluation of BERSHO_CNN with TL based on epoch, a) Accuracy, b) NPV, c) PPV, d) TNR, e) TPR

b) Evaluation of BERSHO_CNN with TL based on k-fold

Figure 8 provides an assessment of the BERSHO_CNN with TL model by changing the k-fold value from 5 to 9. In Figure 8 a), the accuracy of the BERSHO_CNN with TL is presented. When k-fold is taken as 9, the introduced model yields an accuracy of 93.313%, while traditional schemes such as VGG stacked classifier, CNN+SVM, DenseNet 201, EfficientNetB7, GTLO-LeNet, and Squeeze-KNet attained an accuracy of 86.403%, 87.861%, 88.316%, 89.083%, 90.254% and 91.269%, respectively. This result demonstrates that the proposed BERSHO_CNN with TL method improved performance gain by 7.405%, 5.842%, 5.355%, 4.533%, 3.278%, and 2.190% when compared to the existing schemes. Figure 8 b) shows the assessment of the NPV of the proposed model. The BERSHO_CNN with TL model achieved the NPV of 91.465%. In comparison, the traditional models attained the NPV of 86.318%, 86.640%, 87.117%, 87.372%, 88.414%, and 89.720%, resulting in improvements of 5.627%, 5.275%, 4.753%, 4.474%, 3.336%, and 1.907%, correspondingly. Figure 8 c) depicts the assessment of PPV. The developed model attained the PPV of 91.341%, while the traditional models attained the PPV of 86.986%, 87.133%, 87.987%, 88.516%, 88.632% and 89.287%. This indicates performance gains of 4.767%, 4.606%, 3.671%, 3.092%, 2.965%, and 2.247%. Figure 8 d) presents the TNR achieved by the BERSHO_CNN with TL model. The TNR attained by the proposed model is 92.209%. Traditional models achieved TNR of 87.892%, 88.048%, 88.534%, 88.917%, 89.284% and 90.195%, reflecting performance gains of 4.681%, 4.513%, 3.985%, 3.570%, 3.172%, and 2.184%. Finally, Figure 8 e) displays the TPR achieved by the proposed model. The model achieved the TPR of 92.646%. The traditional systems achieved TPR of 87.076%, 87.286%, 88.388%, 88.709%, 89.365% and 90.203%, indicating performance improvements of 6.011%, 5.785%, 4.595%, 4.249%, 3.541%, and 2.636%.

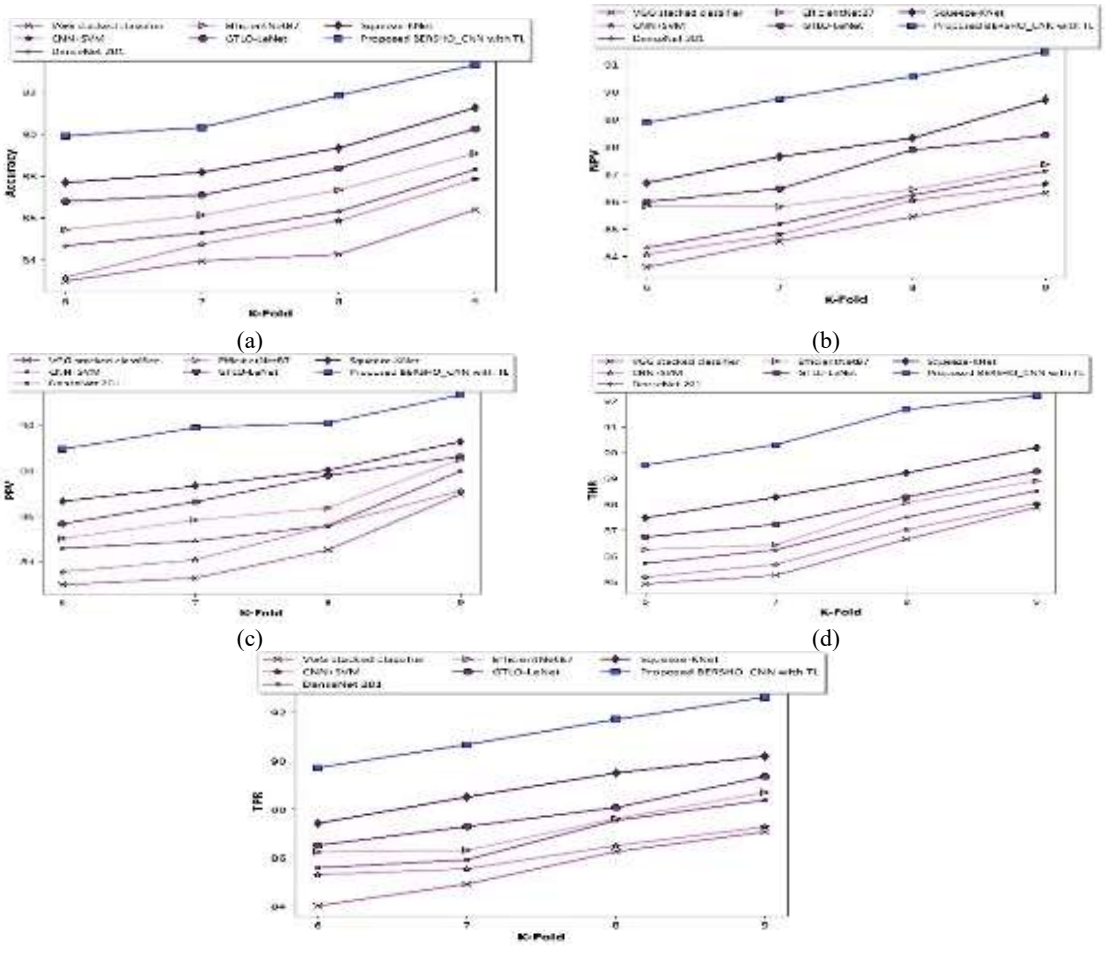

Figure 8: Evaluation of BERSHO_CNN with TL based on k-fold, a) Accuracy, b) NPV, c) PPV, d) TNR, e) TPR

4.6.2 Analysis of BERSHO_CNN with TL utilizing database-2

This section evaluates the performance of BERSHO_CNN with TL using database-2. The assessment involves several metrics, including NPV, TPR, PPV, TNR, and accuracy. The performance analysis is implemented by changing the k-fold value and the number of epochs.

a) Evaluation of BERSHO_CNN with TL based on epoch

Figure 9 presents the evaluation of the BERSHO_CNN with TL model varying the epoch from 25 to 100. In Figure 9 a), the BERSHO_CNN with TL model achieves an accuracy of 92.847% at epoch=100. This outperforms traditional models, such as the VGG stacked classifier, CNN+SVM, DenseNet 201, EfficientNetB7, GTLO-LeNet, and Squeeze-KNet, which have accuracy of 85.664%, 86.076%, 87.226%, 88.665%, 89.607% and 90.599%, respectively. The proposed method shows accuracy improvements of 7.735%, 7.292%, 6.053%, 4.503%, 3.489%, and 2.421% over these existing models. Figure 9 b) denotes that the NPV of the proposed model is 91.788%. In comparison, the NPV of traditional models are 87.151%, 88.201%, 89.437%, 89.803%, 90.337%, and 90.581%, indicating performance gains of 5.051%, 3.907%, 2.561%, 2.162%, 1.580%, and 1.314%, respectively. Figure 9 c) illustrates the PPV graph of the developed model is 92.277%. Traditional models attained PPV of 86.284%, 87.622%, 88.533%, 89.364%, 89.556%, and 90.183%, showing performance improvement of 6.495%, 5.044%, 4.057%, 3.156%, 2.949%, and 2.269%. Figure 9 d) specifies the TNR achieved by the BERSHO_CNN with TL model. The introduced scheme attained the TNR of 91.747%. Traditional models have TNR of 87.596%, 87.830%, 88.116%, 88.521%, 89.285% and 89.701%, reflecting performance gain of 4.524%, 4.269%, 3.958%, 3.515%, 2.683%, and 2.230%. Figure 9 e) displays the TPR of the employed model. The devised scheme achieved the TPR of 92.867%. The TPR of traditional models are 87.141%, 87.332%, 88.043%, 88.719%, 89.427%, and 90.487%, resulting in performance improvements of 5.828%, 5.622%, 4.853%, 4.122%, 3.358%, and 2.212%.

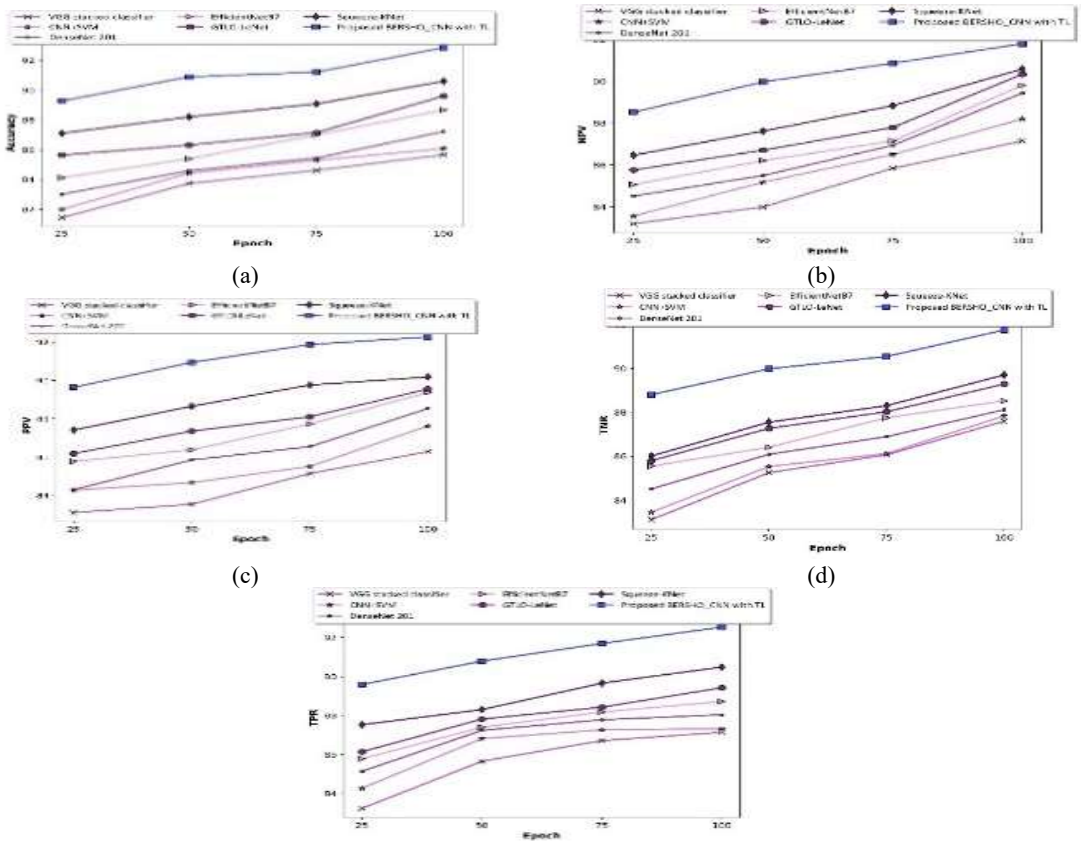

Figure 9: Assessment of BERSHO_CNN with TL based on epoch, a) Accuracy, b) NPV, c) PPV, d) TNR, e) TPR

b) Evaluation of BERSHO_CNN with TL regarding k-fold

Figure 10 portrays the performance evaluation of the BERSHO_CNN with TL model by changing the k-fold from 5 to 9. In Figure 10 a), the BERSHO_CNN with TL model achieves an accuracy of 92.698% during k-fold=9. This surpasses traditional models such as the VGG stacked classifier, CNN+SVM, DenseNet 201, EfficientNetB7, GTLO-LeNet, and Squeeze-KNet, which have an accuracy of 86.786%, 87.352%, 87.522%, 88.330%, 89.517%, and 90.469%, respectively. This indicates accuracy improvements of 6.378%, 5.767%, 5.584%, 4.712%, 3.431%, and 2.405% over these models. Figure 10 b) denotes the evaluation of the NPV of the proposed model. The developed model attained the NPV of 91.937%. In comparison, the NPV of the traditional models are 87.660%, 88.385%, 89.739%, 89.906%, 90.202%, and 90.402%, showing performance gains of 4.651%, 3.863%, 2.390%, 2.209%, 1.887%, and 1.669%, respectively. Figure 10 c) displays the evaluation of PPV for the developed model. When k-fold=9, the BERSHO_CNN with TL yields the PPV of 91.976%. Traditional models achieved PPV of 85.323%, 86.005%, 87.398%, 87.422%, 88.555%, and 89.685%, reflecting performance improvements of 7.233%, 6.491%, 4.977%, 4.951%, 3.718%, and 2.491%. Figure 10 d) illustrates the evaluation of TNR of the BERSHO_CNN with TL model. The developed model attained the TNR of 91.654%. The TNR of traditional models are 86.731%, 87.821%, 88.355%, 88.442%, 88.584%, and 89.593%, indicating performance gains of 5.371%, 4.182%, 3.599%, 3.504%, 3.350%, and 2.248%. Lastly, Figure 10 e) illustrates the TPR attained by the proposed model. The TPR attained by the developed scheme is 92.294%. The TPR for the traditional models are 87.108%, 88.256%, 88.968%, 89.127%, 89.973%, and 90.080%, showing performance improvements of 5.619%, 4.375%, 3.604%, 3.431%, 2.515%, and 2.399%.

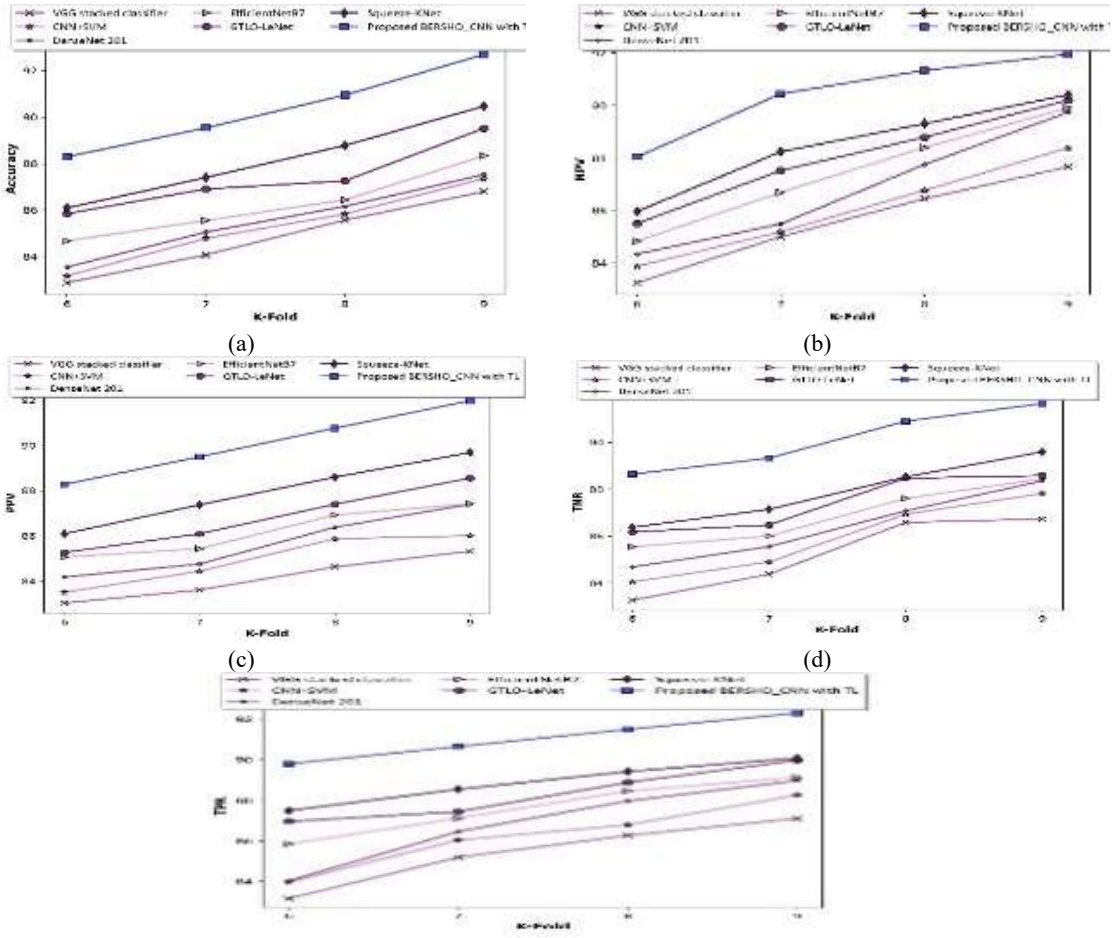

Figure 10: Evaluation of BERSHO_CNN with TL based on k-fold, a) Accuracy, b) NPV, c) PPV, d) TNR, e) TPR

4.7 Comparative Discussion

Table 1 presents the performance of the devised BERSHO_CNN with TL by evaluating it with existing systems, such as VGG stacked classifier, CNN+SVM, DenseNet 201, EfficientNetB7, GTLO-LeNet, and Squeeze-KNet. The proposed model achieved an accuracy of 92.847%, outperforming all existing methods. Similarly, the proposed framework demonstrated superior performance across all other key metrics, including NPV of 91.788%, PPV of 92.277%, TPR of 92.534%, and TNR of 91.747%, indicating its ability to classify brain tumors more reliably and robustly. These improvements are attributed to the combined advantages of the framework's components: NLM filtering reduces MRI noise, enhancing tumor clarity; MAD-Net segmentation allows precise tumor localization, overcoming limitations in boundary detection found in other methods; 3D-Convolutional Autoencoder with Vision Transformer extracts rich volumetric and contextual features, capturing tumor heterogeneity more effectively than traditional CNN or SVM approaches; and the CNN trained with BERSHO efficiently optimizes model parameters, improving classification accuracy and convergence speed. Beyond accuracy, the framework also offers platform independence, and high processing speed, making it suitable for clinical deployment. While the framework demonstrates high performance, it has some limitations. The evaluation is conducted on a limited set of MRI datasets and focuses on specific tumor types, which may affect generalizability. Additionally, the framework requires substantial computational resources, and only single-modality MRI data is used. Overall, this analysis demonstrates that the proposed framework not only outperforms existing methods quantitatively but also achieves its objectives of robust, efficient, and clinically applicable brain tumor classification, highlighting its novelty and contribution to the field.

Table 1. Comparative discussion

Database	data	Methods /Metrics	VGG stacked classifier	CNN+SVM	DenseNet 201	Efficient NetB7	GTLO-LeNet	Squeeze-KNet	Proposed BERSHO_CNN with TL
Database-1	Epoch=100	*Accuracy (%)*	85.111	87.226	88.328	88.823	89.799	90.982	93.722
		NPV (%)	85.718	86.029	86.872	87.949	88.954	89.132	91.204
		PPV (%)	85.864	86.671	87.253	87.510	88.577	89.708	91.720
		TNR (%)	87.125	88.464	88.782	89.038	89.566	90.287	92.351
		TPR (%)	88.045	88.623	89.334	89.781	90.244	90.461	92.867
	K-fold=9	*Accuracy (%)*	86.403	87.861	88.316	89.083	90.254	91.269	93.313
		NPV (%)	86.318	86.640	87.117	87.372	88.414	89.720	91.465
		PPV (%)	86.986	87.133	87.987	88.516	88.632	89.287	91.341
		TNR (%)	87.892	88.048	88.534	88.917	89.284	90.195	92.209
		TPR (%)	87.076	87.286	88.388	88.709	89.365	90.203	92.646
Database-2	Epoch=100	*Accuracy (%)*	85.664	86.076	87.226	88.665	89.607	90.599	**92.847**
		NPV (%)	87.151	88.201	89.437	89.803	90.337	90.581	**91.788**
		PPV (%)	86.284	87.622	88.533	89.364	89.556	90.183	**92.277**
		TNR (%)	87.596	87.830	88.116	88.521	89.285	89.701	**91.747**
		TPR (%)	87.141	87.332	88.043	88.719	89.427	90.487	**92.534**
	K-fold=9	*Accuracy (%)*	86.786	87.352	87.522	88.330	89.517	90.469	92.698
		NPV (%)	87.660	88.385	89.739	89.906	90.202	90.402	91.937
		PPV (%)	85.323	86.005	87.398	87.422	88.555	89.685	91.976
		TNR (%)	86.731	87.821	88.355	88.442	88.584	89.593	91.654
		TPR (%)	87.108	88.256	88.968	89.127	89.973	90.080	92.294

5. CONCLUSION

Brain tumors are a diverse group of neoplastic disorders that can significantly affect neurological function and patient quality of life. Precise classification of these tumors is essential for developing effective treatment strategies. However, variability in tumor types, size, shape, and texture, as well as MRI noise, makes accurate classification challenging. To address this issue, an effective model called BERSHO_CNN with TL is presented for classifying brain tumor utilizing MRI image. The developed method begins with gathering an MRI image and pre-processing it using an NLM filter to reduce MRI noise. Then, tumor segmentation is implemented by employing MAD-Net, which helps in precise localization. By utilizing a segmented image, the feature extraction is accomplished using 3D-CAE+ViT for rich feature extraction from heterogeneous tumor types. Finally, the brain tumor is classified by employing the developed CNN_TL, which is trained using the BERSHO to optimize

classification accuracy and robustness. The BERSHO is the combination of BER and SHO. The devised model achieved 92.847% of accuracy, 91.788% of NPV, 92.277% of PPV, 91.747% of TNR and 92.534% of TPR. The experimental results confirm the hypothesis, showing that the BERSHO_CNN with TL framework provides high classification accuracy, precise tumor localization, and improved feature representation, effectively addressing the challenges identified in brain tumor classification. Beyond its performance metrics, the proposed framework has practical applications in clinical environments. It can serve as a computer-aided diagnostic tool to assist radiologists in faster and more reliable tumor identification, reduce human error, and support treatment planning decisions. Its robustness to MRI noise and variability in tumor types makes it suitable for integration into automated diagnostic pipelines and hospital imaging systems. In the future, the framework will be extended to include additional tumor types, incorporate multi-modal imaging data, and advanced approaches such as ensemble learning and multi-modal data integration can be explored to further improve accuracy and clinical utility.

Declaration Statements:
Funding: This research did not receive any specific funding
Conflict of Interest: The authors declare no conflict of interest
Acknowledgements: I would like to express my very great appreciation to the co-authors of this manuscript for their valuable and constructive suggestions during the planning and development of this research work.
Informed consent: Not Applicable
Ethical approval: Not Applicable
Author Contribution: All authors have made substantial contributions to conception and design, revising the manuscript, and the final approval of the version to be published. Also, all authors agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.
Data Availability Statement:
The data underlying article are available in Figshare dataset, at https://figshare.com/articles/brain_tumor_dataset/1512427.
The data underlying article are available in BRATS dataset, at https://www.med.upenn.edu/sbia/brats2018/data.html.

REFERENCES

[1] Aamir, M., Rahman, Z., Dayo, Z.A., Abro, W.A., Uddin, M.I., Khan, I., Imran, A.S., Ali, Z., Ishfaq, M., Guan, Y. and Hu, Z., "A deep learning approach for brain tumor classification using MRI images", *Computers and Electrical Engineering*, vol.101, pp.108105, 2022.
[2] Katti, G., Ara, S.A. and Shireen, A., "Magnetic resonance imaging (MRI)–A review", *International journal of dental clinics*, vol.3, no.1, pp.65-70, 2011.
[3] Zebari, N.A., Alkurdi, A.A., Marqas, R.B. and Salih, M.S., "Enhancing Brain Tumor Classification with Data Augmentation and DenseNet121", *Academic Journal of Nawroz University*, vol.12, no.4, pp.323-334, 2023.
[4] Asiri, A.A., Soomro, T.A., Shah, A.A., Pogrebna, G., Irfan, M. and Alqahtani, S., "Optimized Brain Tumor Detection: A Dual-Module Approach for MRI Image Enhancement and Tumor Classification", *IEEE Access*, vol.12, pp.42868-42887, 2024.
[5] Anil, A., Raj, A., Sarma, H.A., Chandran, N. and Deepa, R., "Brain Tumor detection from brain MRI using Deep Learning", *International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)*, vol.3, no.2, pp.458-465.2016.
[6] Wahid, F., Fayaz, M. and Shah, A.S., "An evaluation of automated tumor detection techniques of brain magnetic resonance imaging (MRI)", *International Journal of Bio-Science and Bio-Technology*, vol.8, no.2, pp.265-278,
[7] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. and Lew, M.S., "Deep learning for visual understanding: A review", *Neurocomputing*, vol.187, pp.27-48, 2016.
[8] Almalki, Y.E., Ali, M.U., Kallu, K.D., Masud, M., Zafar, A., Alduraibi, S.K., Irfan, M., Basha, M.A.A., Alshamrani, H.A., Alduraibi, A.K. and Aboualkheir, M., "Isolated convolutional-neural-network-based deep-feature extraction for brain tumor classification using shallow classifier", *Diagnostics*, vol.12, no.8, pp.1793, 2022.
[9] Torrey, L. and Shavlik, J., "Transfer learning", In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, *IGI global*, pp. 242-264, 2010.
[10] Majib, M.S., Rahman, M.M., Sazzad, T.S., Khan, N.I. and Dey, S.K., "Vgg-scnet: A vgg net-based deep learning framework for brain

tumor detection on mri images", *IEEE Access*, vol.9, pp.116942-116952, 2021.

[11] Khairandish, M.O., Sharma, M., Jain, V., Chatterjee, J.M. and Jhanjhi, N.Z., "A hybrid CNN-SVM threshold segmentation approach for tumor detection and classification of MRI. brain images", *IRBM*, 2021.

[12] Sharif, M.I., Khan, M.A., Alhussein, M., Aurangzeb, K. and Raza, M., "A decision support system for multimodal brain tumor classification using deep learning", *Complex & Intelligent Systems*, vol.8, no.4, pp.3007-3020,2022.

[13] Islam, M.M., Talukder, M.A., Uddin, M.A., Akhter, A. and Khalid, M., "Brainnet: precision brain tumor classification with optimized efficientnet architecture", *International Journal of Intelligent Systems*, vol.2024, no.1, pp.3583612, 2024.

[14] Raza, A., Ayub, H., Khan, J.A., Ahmad, I., S. Salama, A., Daradkeh, Y.I., Javeed, D., Ur Rehman, A. and Hamam, H., "A hybrid deep learning-based approach for brain tumor classification", *Electronics*, vol.11, no.7, pp.1146, 2022.

[15] Rousselle, F., Knaus, C. and Zwicker, M., "Adaptive rendering with non-local means filtering", *ACM Transactions on Graphics (TOG),* vol.31, no.6, pp.1-11, 2012.

[16] Qin, C., Zheng, B., Li, W., Chen, H., Zeng, J., Wu, C., Liang, S., Luo, J., Zhou, S. and Xiao, L., "MAD-Net: Multi-attention dense network for functional bone marrow segmentation", *Computers in Biology and Medicine*, vol.154, pp.106428, 2023.

[17] Essadik, I., Nouri, A., Touahni, R. and Autrusseau, F., "Comparing Autoencoder to Geometrical Features for Vascular Bifurcations Identification", In 2023 International Symposium on Image and Signal Processing and Analysis (ISPA), *IEEE*, pp. 1-6, September 2023.

[18] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., "An image is worth 16x16 words: Transformers for image recognition at scale", *arXiv preprint arXiv:2010.11929*, 2020.

[19] Shi, Z., Hao, H., Zhao, M., Feng, Y., He, L., Wang, Y. and Suzuki, K., "A deep CNN based transfer learning method for false positive reduction", *Multimedia Tools and Applications*, vol.78, pp.1017-1033, 2019.

[20] Sustika, R., Subekti, A., Pardede, H.F., Suryawati, E., Mahendra, O. and Yuwana, S., "Evaluation of deep convolutional neural network architectures for strawberry quality inspection", *Int. J. Eng. Technol*, vol.7, no.4, pp.75-80, 2018.

[21] El-Kenawy, E.S.M., Abdelhamid, A.A., Ibrahim, A., Mirjalili, S., Khodadad, N., Alduailij, M.A., Al"n, A.A. and Khafaga, D.S., Al-Biruni Earth Radius (BER) Metaheuristic Search Optimization Algorithm", *Comput. Syst. Sci. Eng*., vol.45, no.2, pp.1917-1934, 2023.

[22] Ghafori, S. and Gharehchopogh, F.S., "Advances in spotted hyena optimizer: a comprehensive survey", *Archives of computational methods in engineering*, vol.29, no.3, pp.1569-1590, 2022.

[23] The Figshare dataset is taken from, "https://figshare.com/articles/brain_tumor_dataset/1512427", accessed on August 2024.

[24] The BRATS 2018 database is taken from, "https://www.med.upenn.edu/sbia/brats2018/data.html", accessed on August 2024.

[25] Natarajan, B., Elakkiya, R., Bhuvaneswari, R., Saleem, K., Chaudhary, D. and Samsudeen, S.H., "Creating alert messages based on wild animal activity detection using hybrid deep neural networks", *IEEE Access*, vol.11, pp.67308-67321, 2023.