International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.26, July 2025

BrainNet: CNN-Powered Diagnosis to Detect and Classify Brain Tumor from MRI Imaging Technique

Rajdip Ghosh
Department of Computer Science, Sister Nivedita University
DG Block, Action Area I, 1/2, Newtown, West Bengal 700156

Soham Goswami
Department of Computer Science, Sister Nivedita University
DG Block, Action Area I, 1/2, Newtown, West Bengal 700156

Sagnik Bhattacharjee
Department of Computer Science, Sister Nivedita University
DG Block, Action Area I, 1/2, Newtown, West Bengal 700156

Soma Datta
Department of Computer Science, Sister Nivedita University
DG Block, Action Area I, 1/2, Newtown, West Bengal 700156

ABSTRACT

In medical image processing, brain tumor segmentation is a crucial problem. Patients' chances of survival are increased and treatment options are improved when brain tumors are detected early. It is challenging and time-consuming to manually segment brain tumors for cancer diagnosis from the ample number of MRI images produced during clinical routines. Automatic segmentation of brain tumor images is required. The varied image content, crowded objects, occlusion, image noise, non-uniform object texture, and other characteristics make segmentation a difficult challenge even after much research. Although there are numerous algorithms and methods for image segmentation, a quick and effective method for medical image segmentation still has to be developed. MRI brain images were initially subjected to preprocessing and enhancement methods. The damaged brain tumor region was then segmented using a new 2D Convolutional Neural Network (CNN) approach that is created. The proposed method is not only able to segment the affected area but also able to properly classify the type of brain tumor. The proposed technique achieved an overall accuracy of 91.3% and a recall of 88% respectively.

Keywords

MRI Imaging, Brain tumor, Convolution Neural Network, Data augmentation,Computer-assisted diagnosis

1. INTRODUCTION

Brain tumors are a diverse and complex group of cancers that pose major challenges due to their aggressive nature and resistance to treatment. This thesis explores the origin, classification, and treatment of brain tumors, with a focus on primary tumors like glioblastoma—the most lethal type, having a 5-year survival rate below 5% [19, 25]. Despite progress in medical research, the causes and mechanisms behind brain tumors remain poorly understood, though environmental and genetic factors are believed to play a role. The thesis aims to review current knowledge, highlight knowledge gaps, and suggest future research directions to improve diagnosis, therapy, and patient outcomes. Figure 1 shows MRI scan of the frontal view of the brain with tumor.

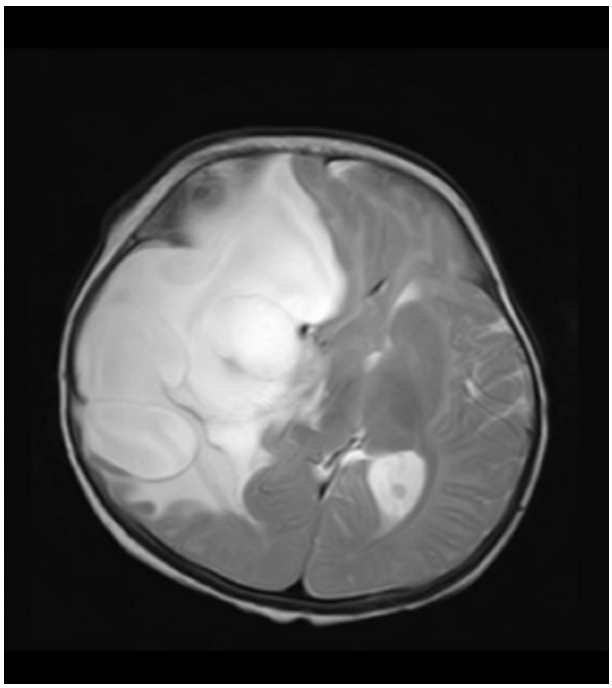

Fig. 1. Sample brain MRI Image

Brain tumors are abnormal cell growths that can be primary (originating in the brain) or secondary/metastatic (spread from other parts of the body)[23]. Primary tumors are further categorized into benign tumors, malignant tumors and secondary brain tumors.

—Benign Tumors – Noncancerous and slow-growing, but may still cause harm due to pressure on nearby brain tissue.

—Malignant Tumors – Cancerous, fast-growing, and capable of invading healthy brain cells, demanding urgent treatment.

—Secondary brain - It tumors arise when cancer from organs like the lungs or breasts spreads to the brain, facilitated by its rich blood supply and protective barriers.

Figure 2 shows different types of brain tumor classifications.

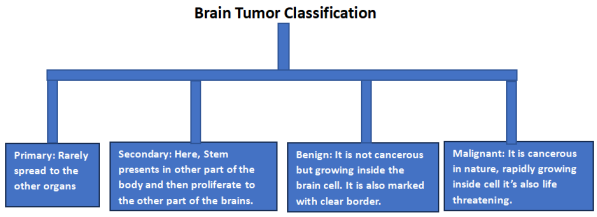

Fig. 2. Brain Tumor Classification

Symptoms range from headaches and seizures to vision problems and cognitive issues, depending on tumor size and location. Diagnosis typically involves MRI/CT imaging, followed by biopsy or surgical confirmation.
Tumor behavior is influenced by genetic mutations, epigenetic changes, and cancer stem cells, which contribute to resistance against therapies. While advances in genomics and bioengineering offer new treatment avenues, challenges like therapeutic resistance, especially in glioblastoma, remain significant.
Some popular treatment approaches are given as follows[21]-

—Surgery – Preferred when tumors are accessible.

—Radiation Therapy – Destroys tumor DNA post-surgery or for inoperable tumors.

—Chemotherapy – Limited by the blood-brain barrier but still used, especially temozolomide for GBM.

—Targeted Therapy – Attacks specific tumor molecules.

—Supportive Care – Manages symptoms and improves quality of life.

In modern medical diagnostics, imaging modalities have gained popularity due to their accuracy and minimal risk to patients. Techniques such as radiography, tomography, echocardiography, and especially Magnetic Resonance Imaging (MRI) are commonly used. MRI stands out for its high-resolution, radiation-free imaging and non-invasive nature, making it ideal for detecting brain abnormalities. Additionally, Computer-Aided Diagnosis (CAD) systems enhance early brain tumor detection by analyzing MRI data and generating diagnostic insights without human input, aiding radiologists in clinical decision-making.
Image segmentation algorithms rely on two core properties of image intensity: discontinuity and similarity. Discontinuity-based methods detect abrupt changes in intensity, such as edges and corners, while similarity-based methods group regions with uniform characteristics. A variety of techniques have evolved, including histogram-based, edge-based, neural network-driven, region-based, physical model-based, and clustering methods like K-means and Fuzzy C-means. Despite these advancements, image segmentation remains a challenging task due to the lack of a universal method suitable for all image types and applications, making the selection of an optimal technique problem-specific and complex [15, 18].
The main contributions of this study are as follows,

—A large dataset of MRI images is used to test our networks, which is ideal for both the training and testing stages.

—The number of layers, the arrangement of the layers adjacent to one another, the kind of parameters and hyperparameters, and their values that can be changed to fine-tune our models to improve accuracy are all changes made to the internal architecture of the modified 2D CNN and convolutional neural network.

—Two-dimensional CNN, preprocessing approaches are used to categorize different types of brain cancers and healthy brains using extracted important information.

In this manuscript, section 2 provides an overview of recent scientific research in medical image segmentation. Section 3 details the methodologies employed in this study. Section 4 presents the experimental results obtained from testing the proposed methods on three different datasets and discusses the key findings. Finally, Section 5 concludes the paper and outlines directions for future research.

2. EXISTING METHODS FOR BRAIN TUMOR SEGMENTATION

Brain tumor segmentation methods are categorized into manual, semi-automatic, and fully automatic approaches, depending on the degree of user interaction involved in it.

2.1 Manual Brain Tumor Segmentation Method

The radiologists have to use the multimodality information provided by the MRI scans in conjunction with their expertise and experience in anatomy and physiology to perform manual segmentation. In order to diagnose the tumor and its locations by hand, the radiologists should go over several image slices one after another. In addition to taking a lot of time, manual segmentation is reliant on radiologists and yields results that vary greatly across and between raters. However, the outcomes of semiautomated and completely automatic techniques are frequently assessed using manual segmentations [11].

2.2 Semi Automated Brain Tumor Segmentation Method

Three primary goals of semi-automatic approaches are important for user interaction: startup, intervention or feedback response, and evaluation. Typically, initialization involves establishing an area of interest (ROI) for the automatic algorithm to handle, which includes the approximate tumor region. It is also possible to modify pre-processing technique parameters to fit the supplied images. During the process, automated algorithms can be guided toward a desired outcome in addition to initialization by getting feedback and making necessary adjustments.
Neither tools nor drugs are injected into the patient's body during the MRI procedure. The entire procedure is extremely safe, and radiation does not harm human health. Moreover, MRI is particularly well-suited for the diagnosis of brain disorders due to its high resolution, precise soft tissue placement, and sensitivity to illness features are reported in [31]. Disease diagnosis also heavily relies on instruments that have gained popularity in the medical industry, in addition to medical imaging techniques. One of the most popular and effective instruments in the medical industry is image processing. The goal of medical image processing is to create electronic systems that address medical diagnosis issues reported in [17].
Hivhare, Sharma, and Singh [28] proposed a fully automated method for brain tumor segmentation utilizing a parameter-free K-

means clustering algorithm combined with mathematical morphological operations such as dilation and hole filling. This approach was applied to the training dataset of BRATS 2015. The segmented tumor regions were compared with the ground truth provided in the dataset, achieving a Dice Similarity Coefficient (DSC) of 75
Filho et al. [24] proposed an adaptive and parameter-free algorithm for medical image segmentation based on the Optimum Path Snakes (OPS) technique. The method begins with pre-processing to extract features such as texture using HU moments, Gray Level Co-occurrence Matrix (GLCM), Human Density Analysis (HDA), and statistical moments. Segmentation is then performed using the OPS algorithm. The approach is evaluated using performance metrics including Hausdorff Distance (HD), Dice Coefficient (DC), and processing time. For lung segmentation, the method is compared against the vector field convolution and gradient vector flow techniques, while for brain segmentation, it is evaluated against the watershed method, region growing, and Level Set algorithm based on the Coherent Propagation Method (LSCPM). Results demonstrate that the proposed technique outperforms existing methods and is generalizable across different medical imaging applications.
Randu et al. [4] proposed a segmentation approach for pancreatic and brain tumors that combines Statistical Region Merging (SRM) with classification using a Back Propagation Neural Network (BPNN). The process begins with image pre-processing using the Decision-Based Couple Window Median Filter (DBCWMF). Segmentation is then performed using the SRM technique. Feature extraction is conducted using Cat Swarm Optimization (CSO) and Scale Invariant Feature Transform (SIFT). Finally, the extracted features are classified using a BPNN. The proposed method demonstrates superior performance, with DBCWMF outperforming traditional median and PGPD filters, and BPNN outperforming both Artificial Neural Network (ANN) and AdaBoost classifiers.

2.3 Fully Automated Brain Tumor Segmentation Methods

There is no need for user intervention in completely automatic brain tumor segmentation techniques. The segmentation problem is primarily resolved by combining artificial intelligence and prior knowledge. A system that uses a convolution neural network (CNN) as a pixel classifier for the segmentation process of some X-ray pictures was proposed by Glavan and Holban [3]. Every pixel in the image is examined by the algorithm, which attempts to categorize them into two groups: bone and non-bone. They made an effort to isolate the region of bone tissue from the remainder of the picture. Compared to other configurations, their CNN produced the best results. They only used the portions of a picture that were of interest in order to guarantee the network's shortest training time. Their approach identified the important bone regions, however issues surfaced when the region displayed abnormalities and required extra training execution time.
The Fuzzy K-C-means approach, developed by Funmilola et al. [1], has more fuzzy C-means characteristics than K-means. The algorithm scans the image, calculates the number of iterations, uses a distance checker to lower the number of iterations, estimates the image's size, concatenates the dimensions, creates huge data items using distance computation, and minimizes repetition once the maximum distance has been reached. The iteration starts by recognizing a substantial component of the data and ends when there is no longer any chance of identification. The drawback is that, with the exception of a few photographs, the results of their suggested approach are comparable to those of the fuzzy C-means algorithm. Fuzzy C-means takes longer than their suggested method by up to two seconds.
A multi-phase method for segmenting the multisequence image of a brain tumor was presented by Lim and Mandava [14]. There are three steps in the suggested method. The first step involves modeling the data using the random wa1ks approach. In the second step, a weighted average technique is employed to fuse the data. Information Theoretic Rough Sets (ITRS) are used in the final step to extract visual objects. The suggested approach is tested using the MICCAI brain tumor dataset. The proposed approach has a correlation with the Principal Component Analysis (PCA) fusion and simple averaging methods. Using the DICE metric, the effectiveness of the suggested approach was assessed, yielding an average DICE accuracy of 0.7 for high-grade tumors and 0.63 for low-grade tumors.
A Multilayer Perceptron (MLP) method for identifying diseased brain was introduced by Zhang et al. [14]. Twelve Fractional Fourier Entropy (FRFE) features are extracted in the first step of feature extraction. The MLP classifier is utilized for classification in the following phase. The pruning procedure is used to obtain the ideal hidden neuron number. Dynamic Pruning (DP), Bayesian Detection Boundaries (BDB), and Kappa Coefficient (KC) are the three pruning methods that are compared. Adaptive Real-Coded Biogeography Based Optimization (ARCBBO) is used to train weights and biases. The results showed that a greater average accuracy of 99.53% was achieved by combining FRFE, KC, MLP, and ARCBBO. When compared to SVM and native Bayesian classifiers, the suggested method performs better than the other two.
The performance of the four most well-known clustering methods—K-means, Fuzzy C-means, evaluated in this work. The authors compared these algorithms and our suggested method in terms of accuracy and processing time. Three distinct data sets including ample amount of MRI images of the brain with tumor cells were used to evaluate the algorithms. By eliminating user input, saving time, preserving picture information, and removing point inference, our integration naturally avoided over- and under-segmentation while achieving accuracy.

3. DESIGN

In this research word, the authors used Convolutional Neural Networks (CNNs) for image classification due to their effectiveness in recognizing visual patterns. CNNs mimic the human visual system, using convolution operations to detect features like edges and shapes, and pooling to reduce data complexity. Their layered architecture allows the network to learn both simple and complex features hierarchically. Advantages include parameter sharing, which reduces the number of learnable parameters and prevents overfitting, and translation invariance, enabling the model to recognize objects regardless of their position in the image. These strengths make CNNs ideal for tasks like facial recognition, medical imaging, and autonomous driving. Figure 3 shows the functional diagram of the proposed method. A Convolutional Neural Network (CNN) is a deep learning algorithm specifically designed for image recognition and processing tasks. Its architecture, inspired by the human visual system, consists of multiple layers—including convolutional, pooling, and fully connected layers. CNNs are particularly effective at capturing hierarchical patterns and spatial relationships within images, making them highly suitable for visual data analysis.

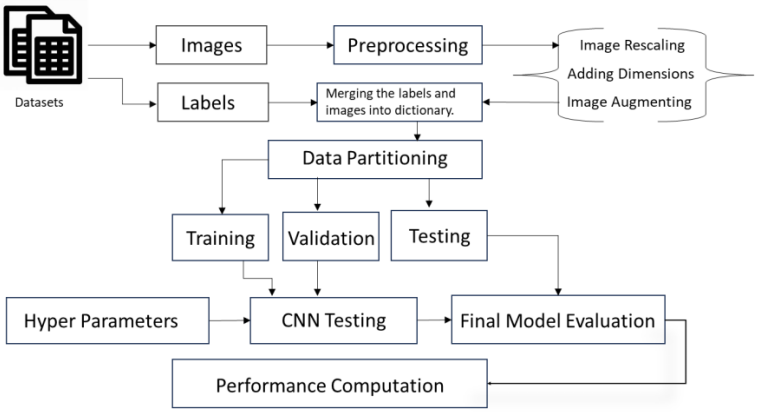

Fig. 3. Functional Diagram of the Proposed Method

3.1 Key components of a Convolutional Neural Network

Convolutional Layers: These layers perform convolution operations on input images using filters (kernels) to identify key features such as edges, textures, and complex patterns. This process preserves the spatial relationships between pixels.
Pooling Layers: Pooling layers reduce the spatial dimensions of the feature maps, lowering computational load and minimizing the number of parameters. A common technique is max pooling, which selects the highest value from a group of neighboring pixels.
Activation Functions: Non-linear activation functions like the Rectified Linear Unit (ReLU) introduce non-linearity into the network, enabling it to learn and model complex patterns within the data.
Fully Connected Layers: Positioned at the end of the network, these layers interpret the high-level features extracted by previous layers and generate the final predictions. Each neuron is connected to all neurons in the preceding and succeeding layers.
In order to train CNNs to identify patterns and features linked to certain objects or classes, a sizable collection of labeled images is used. demonstrated exceptional efficacy in image-related tasks, attaining cutting-edge results across a range of computer vision applications. They are ideal for jobs where the spatial relationships and patterns in the data are essential for precise predictions because of their capacity to automatically learn hierarchical representations of characteristics. CNNs are extensively utilized in fields including medical image analysis, object detection, facial recognition, and image categorization [18, 32, 20, 7]. The main part of a CNN is its convolutional layers, which apply filters to the input image in order to extract characteristics like edges, textures, and forms. In order to down-sample the feature maps and preserve the most crucial information, the output of the convolutional layers is then sent through pooling layers. One or more fully connected layers are then applied to the pooling layers' output in order to classify the image or make a prediction.

3.2 Convolutional Neural Network Design

The construction of a convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order. It is the sequential design that give permission to CNN to learn hierarchical attributes.In CNN, some of them followed by grouping layers and hidden layers are typically convolutional layers followed by activation layers. The pre-processing needed in a ConvNet is kindred to that of the related pattern of neurons in the human brain and was motivated by the organization of the Visual Cortex.

3.3 Convolutional Neural Network Training

CNNs are trained using a supervised learning approach. This means that the CNN is given a set of labelled training images. The CNN then learns to map the input images to their correct labelsc[6, 8]. The training process for a CNN involves the following steps: **Data Preparation:** The training images are pre-processed to ensure that they are all in the same format and size. Also, the resolution of the image is kept higher for the model to detect better. **Loss Function:** A loss function is used to measure how well the CNN is performing on the training data. The loss function is typically calculated by taking the difference between the predicted labels and the actual labels of the training images. **Loss Function Equation:** This loss function is calibrated by Kullback-Leibler Divergence which then, measures difference between 2 distributions based on mutual information. When x and y independent they converge, as x and y are dependent, they diverge.

$$D(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} \, dx \tag{1}$$

Optimizer: An optimizer is used to update the weights of the CNN in order to minimize the loss function.

$$\theta = \theta - \alpha \cdot \nabla J(\theta) \tag{2}$$

Backpropagation: Backpropagation is a technique used to calculate the gradients of the loss function with respect to the weights of the CNN. The gradients are then used to update the weights of the CNN using the optimizer.

3.4 CNN Evaluation

A held-out test set can be used to assess CNN after training. The test set consists of a set of images that the CNN has not seen during training. The CNN's performance on the test set is a reliable indicator of how well it will work with real data. Performance is compared through accuracy, precision and recall as per equation 3,4 and 5.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

In equation 3,4 and 5 TP is true positive, TN is true negative, FN is false negatine and FP is false positive.
The used approach towards the tumor detection from the MRI scan datasets required for classification of the dataset, along with the customization of the image to the desired colour grading required for the model to test and train itself which is grayscale for the model and RGB for the user to understand [10, 5, 26]. Also, the image needs to be resized to the proper dimension for testing and training. A lot of variables have been used for representing different aspects for the training and testing of the model for datasets.

3.5 Importing TensorFlow, Keras and Matplotlib libraries

This code imports the tensorflow and keras Python libraries. The tensorflow library contains tools for building and training machine learning models. The keras library contains tools for building neural networks specifically, which are a type of machine learning model. This will allow us to visualize things like accuracy/loss curves when training models or visualizing the predictions of a model on test data. Having visualization capabilities is important for understanding how machine learning models are behaving. Overall, these three import statements bring in external libraries that will provide the key building blocks needed to build, train, evaluate, and visualize machine learning models in the rest of the code. They don't execute any training or processing themselves, just import the tools needed to do so later.

3.6 Splitting the dataset into the Training set Test set and Validation Set

The $get\_dataset\_partitions\_tf$ function splits a given TensorFlow dataset into training, validation, and test subsets based on specified proportions (defaulting to 80/10/10). It optionally shuffles the dataset using a configurable buffer size. After verifying that the split ratios sum to 1.0, it calculates the sample count for each subset, applies take and skip to extract the corresponding slices, and returns the datasets as $train\_ds$, val_ds, and $test\_ds$. Additionally, the datasets can be optimized for performance using .cache(), .shuffle($buffer\_size$), and .prefetch(). These operations improve efficiency by storing data in memory, randomizing sample order, and overlapping preprocessing with training.

3.7 Preparing the dataset using data augmentation

The $resize\_and\_rescale$ Sequential model standardizes input images by resizing them to a fixed square size ($IMAGE\_SIZE \times IMAGE\_SIZE$) and scaling pixel values to the [0, 1] range using a Rescaling(1./255) layer. This preprocessing ensures uniform input shape and normalized values, improving training consistency for downstream models. The $data\_augmentation$ Sequential model applies random transformations—specifically, horizontal/vertical flipping and small rotations (up to 20%)—to increase the variety and size of the training data, helping reduce overfitting and improve generalization. Each image passes through these layers sequentially. The augmented training data is created by mapping this model to the $train\_ds$ dataset using .map(), which applies the augmentation to each image while keeping the labels unchanged. The .prefetch() method is then used to optimize data loading during training.

3.8 EPOCHS

The EPOCHS variable sets the number of full passes through the training dataset the model will take during training. More epochs allow the model to train for longer and possibly achieve better accuracy but can increase training time.dataset = tf.keras.preprocessing. $image\_dataset\_from\_directory()$ loads images from a directory to create an image dataset.

3.9 Model Initialization

The model that does the main work of tumor detection within the scanned magnetic resonance image of the brain. There is a sequential model in Keras for an image classification task. The variable model.Sequential() constructor creates a Sequential model, which stacks layers linearly. This allows us to define the model layer by layer. The first layer is a $resize\_and\_rescale$ layer which presumably resizes and rescales the input images as preprocessing. The next layers are Convolutional 2D layers (Conv2D) which apply convolutions over the image to extract features. The kernel size of 3x3 is specified to define the size of the convolution window. ReLU activation is used for non-linearity. After each Conv2D layer, a MaxPooling2D layer is added to reduce the spatial dimensions and perform down sampling. The pool size of 2×2 is specified for max pooling. As we stack more Conv2D and MaxPooling2D layers, the model is able to extract higher-level features from the images. After the convolutional base, the Flatten layer flattens the feature maps into a 1D vector. This is fed into a fully-connected Dense layer with ReLU activation to perform classification [13]. Finally, the output layer has 4 nodes with SoftMax activation to output classification probabilities for the 4 classes. In summary, this model takes input images, applies convolutions and pooling to extract features, flattens the features, and applies fully-connected layers to classify the images into 4 classes based on the extracted features. The overall architecture is designed for image feature extraction and classification. ReLU, which stands for Rectified Linear Unit, is an activation function commonly used in neural networks, particularly in deep learning models. It is defined mathematically as equation 6.

$$f(x) = \max(0, x) \tag{6}$$

It means that if the input x is positive. If (x) is negative, the output will be (0). Essentially, ReLU outputs the input directly if it is positive, otherwise, it will output zero. ReLU is favored because it introduces non-linearity into the model without affecting the ability to propagate gradients, which is essential for training deep neural networks. It helps to overcome the vanishing gradient problem, allowing models to learn faster and perform better compared to when sigmoid or hyperbolic tangent activation functions are used. While ReLU is powerful, it has limitations too (e.g., the "dying ReLU" problem for negative inputs). Researchers have proposed variants like Leaky ReLU, Parametric ReLU, and Exponential Linear Units (ELUs) to address these limitations.

4. PERFORMANCE ANALYSIS

This section of the content is configuring and compiling a neural network model in TensorFlow. It starts by calling the compile method on the model variable, which is assumed to be a neural network model that has already been defined. This section also provide the comparative study between existing method with the proposed method.

4.1 TRAINING THE MODEL

Training is the most vital process in this approach as the model needs training again and again it is a lengthy process where a dataset arranged into sets of all the different types of images as in the different type of brain tumor present in the MRI which then is filled into the model for training. The output is a history object containing the training loss, validation loss, and other metrics logged during training. This shows how the model improved over time. By repeatedly looping through the data to minimize the loss, the model learns to make predictions matching the true labels. The validation data helps prevent overfitting by tracking performance on data not used in training.

4.2 TESTING THE MODEL

In order to detect and classify the brain tumor the model needs to be trained firstly and the tested with n number of attempts before the tumor can be detected. Evaluation of the whole training is noted by the model which is then represented by model.evaluate(). It takes two inputs: test_ds. This is a TensorFlow dataset containing the test data that the model performance will be evaluated on. This should be data that the model has not seen during training model. This is the trained machine learning model whose performance is evaluated. This should have already been trained on a separate training dataset. It returns a list of metric values reflecting how well the model performed on the test data. The exact metrics returned depend on the model, but usually include metrics like accuracy, precision, recall, F1 score, etc. Checking performance on test data helps identify overfitting and other issues during training before the model is deployed. The test performance metrics are very important to consider before putting a model into production.

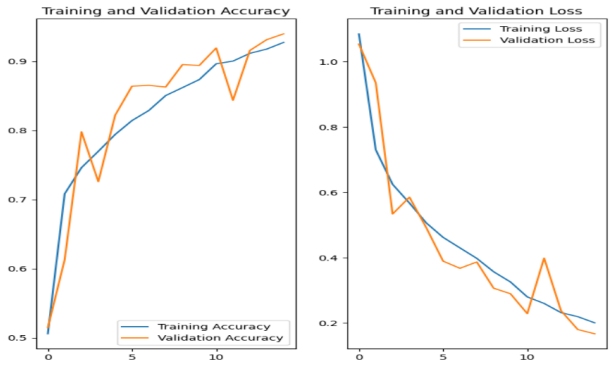

Fig. 4. Pictorial representation of the validation accuracy and loss of the training.

In figure 4, the first subplot on the left plots the training and validation accuracy over epochs and the second subplot on the right plots the training and validation loss over epochs, similar to the accuracy plot. This is a common way to visualize model performance during training. The training and validation curves allow comparing how well the model fits the training data vs generalizes to new data over time.

The predict function is used to make predictions using a trained machine learning model. It takes a model and an image as inputs. 39 The image is first converted into a numpy array using img_to_array. This puts the image data into a format the model can understand. The image array is then reshaped into a single sample with expand_dims. This converts the flat array into a 'batch' of one image. The model makes a prediction on the image by calling model predict. This runs the image through the model and returns the predicted classes and probabilities. The class with the highest probability is considered the model's predicted class. The index of this class is found using np.argmax. The class name itself is looked up from the class_names list using the predicted index. The confidence score is calculated by taking the maximum predicted probability and converting to a percentage. The predict function returns the predicted class name and confidence percentage as outputs. So, in summary, it takes an image, runs it through a trained model, looks up the best matching class, and returns the prediction, converting the outputs into a readable class name and confidence score along the way. The main logic flow is preparing the data, making the prediction, and interpreting the results.

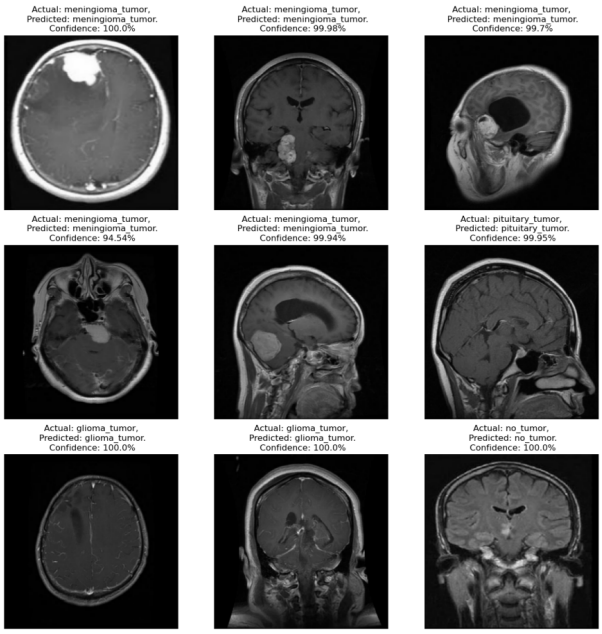

Fig. 5. Shows the final result of the confidence level

It takes a batch of test data, runs the images through the model to make predictions, compares to the true labels, and visualizes the results in a grid to provide an overview of how well the model performs on real examples. The main steps are the prediction logic and mapping between integer encodings and human-readable class names. This provides an intuitive visualization for assessing the model's performance.

Figure 6 shows the confusion matrix based on prediction number 880 and number of wrong prediction is 60. Table 1 shows the comparative study between existing methods and the proposed model. According to table 1, the proposed method provides better result than the existing methods for "BRATS dataset" [2]. Figure 7 and Figure 8 graphically represents whole, core, active tumour and the comparative study Of accuracy, recall against proposed methods and existing methods.

According to the approach, the authors have used results have been released after the model has predicted the type of tumor present inside the MRI picture scan of the brain. These results include the 3 types of tumor that the model was fed upfront for training. Now, as the image fed into the model was just normal JPEG/JPEC/PNG formats, which were then converted into the required grayscale scale which is the output after this the class of the image and the confidence level of the prediction are shown to the user scanning an image.

5. CONCLUSION

In conclusion, the research presented in this dissertation demonstrates the significant potential of Convolutional Neural Networks (CNN) in the classification of brain tumors from MRI scans. The study's findings underscore the accuracy, efficiency, and reliability of CNNs in distinguishing between different types of brain tumors, which is a critical step in the diagnosis and treatment plan-

Table 1. Performance accuracy using different algorithms.

Author	Method	Level of user interaction	Whole Tumor	Core Tumor	Active Tumor	Accuracy in%	Recall
Human Rater[16]	Medical training and experience	Manual	0.88	0.93	0.74	87	.78
Pereira et al.[22]	CNN with small (3x3) filters for deeper architecture	Fully automatic	0.86	0.83	0.76	82	.75
Kwon et al.[12]	Generative model that performs joint segmentation and registration	Semi-automatic	0.82	0.83	0.712	88	.86
Havaei et al.[9]	Cascaded Two-pathway CNNs for simultaneous local and global processing	Fully automatic	0.87	0.79	0.73	86	.80
Tustison et al.[29]	Concatenated RFs, trained using asymmetry and first order statistical features	Fully automatic	0.87	0.78	0.74	90	.89
Urban et al.[30]	3D CNN architecture using 3D convolutional filters	Fully automatic	0.87	0.77	0.73	89	.86
Davy et al.[27]	Two-pathway CNN for simultaneous local and global processing	Fully automatic	0.85	0.74	0.68	85	.80
Zikic et al.[33]	3D input patches are interpreted into 2D input patches to train a CNN	Fully automatic	0.837	0.736	0.69	91	.91
Proposed Method	Four CNNs, one for each modality, with their outputs concatenated as an input into a RF	Fully automatic	.89	.78	.76	91.3	.88

This table summarizes the accuracy and practical implications of different distance measurement algorithms used in geospatial computations.

ning for patients. The implementation of CNN models has shown a remarkable ability to learn complex features from medical imaging data, outperforming traditional machine learning methods. This advancement could revolutionize the field of medical imaging and provide clinicians with powerful tools to enhance patient outcomes. Future work should focus on refining these models, improving their interpretability, and integrating them seamlessly into clinical workflows. The promise of AI in healthcare is immense, and this research is a testament to its transformative potential in oncology diagnostics. In the realm of brain tumor classification, several machine learning techniques can complement Convolutional Neural Networks (CNNs) to enhance performance and accuracy. Ensemble methods, which combine multiple models to make a single prediction, can be particularly effective. Techniques such as Random Forests and Gradient Boosting Machines can provide robustness against overfitting and improve generalization by aggregating the predictions of numerous decision trees. Support Vector Machines (SVMs) are another valuable addition, known for their effectiveness in high-dimensional spaces, which is common in medical imaging data. SVMs can classify data with a clear margin of separation, making them useful for distinguishing between different tumor types. Transfer learning is another approach that can be utilized alongside CNNs. By leveraging pre-trained models on large datasets, transfer learning allows for the application of knowledge gained in one domain to be applied to another, which can be particularly beneficial when dealing with limited medical imaging data. Additionally, feature extraction techniques can be employed to reduce the dimensionality of the data and identify the most relevant features for classification, which can then be fed into machine learning models like k-Nearest Neighbours (k-NN) or Naive Bayes classifiers. Moreover, deep learning techniques such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks can be used to analyse sequential data, which could be useful in tracking tumor development over time. Generative Adversarial Networks (GANs) can also be applied to generate synthetic medical images, which can augment the training data and improve the robustness of CNNs. Furthermore, advanced techniques like Quantum Machine Learning (QML) are emerging, which promise to exploit quantum computing principles to process information in fundamentally new ways, potentially leading to breakthroughs in complex tasks like brain tumor classification. Additionally, the integration of explainable AI (XAI) methods can help in making the decision-making process of CNNs more transparent and interpretable for clinicians. In summary, while CNNs are powerful tools for brain tumor classification, their performance can be significantly enhanced by integrating them with other machine learning techniques. These complementary methods can address the limitations of CNNs and provide a more holistic approach

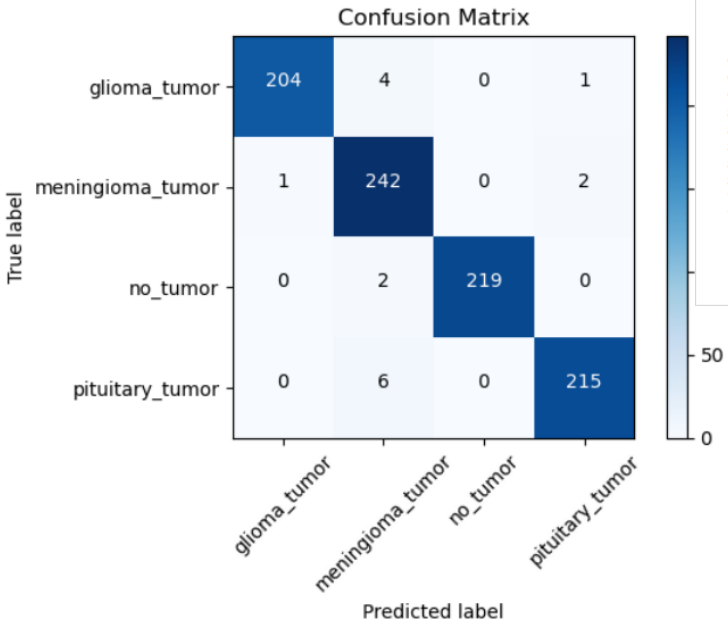

Fig. 6. Shows the confusion matrix of the testing stage

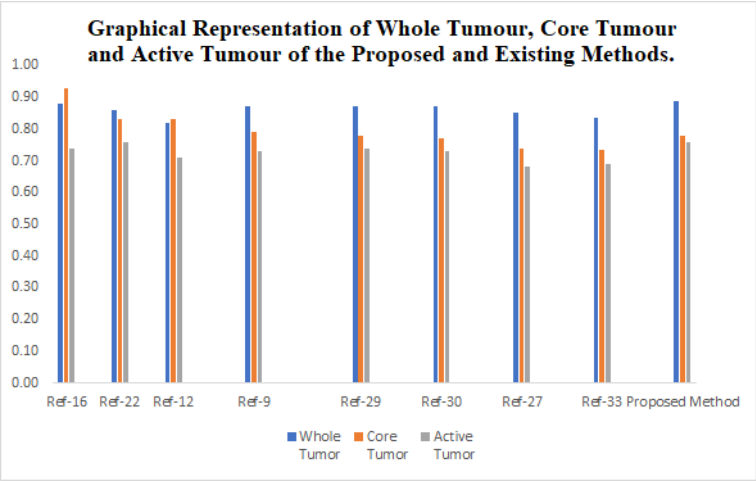

Fig. 7. Shows Whole, Core and Active Tumour of the Proposed and the Existing Methods.

to the classification and analysis of brain tumors, ultimately leading to better diagnostic tools and patient outcomes.

6. REFERENCES

[1] A Ajala Funmilola, OA Oke, TO Adedeji, OM Alade, and EA Adewusi. Fuzzy kc-means clustering algorithm for medical image segmentation. *Journal of information Engineering and Applications, ISSN*, 22245782:2225–0506, 2012.

[2] Stefan Bauer, Roland Wiest, Lutz-P Nolte, and Mauricio Reyes. A survey of mri-based medical image analysis for brain tumor studies. *Physics in Medicine & Biology*, 58(13):R97, 2013.

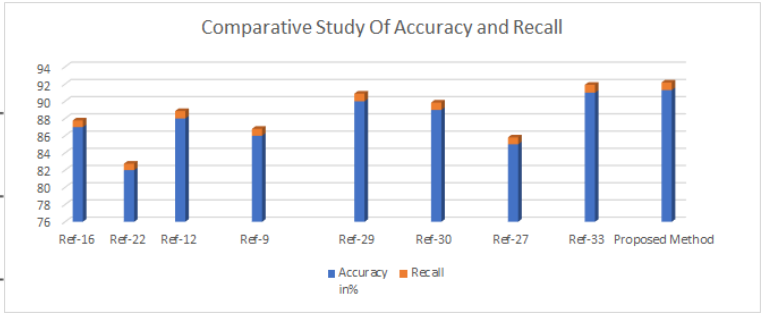

Fig. 8. Shows Comparative Study Of Accuracy and Recall against Proposed Methods and Existing Methods

[3] Cosmin Cernazanu-Glavan, Stefan Holban, et al. Segmentation of bone structure in x-ray images using convolutional neural network. *Adv. Electr. Comput. Eng*, 13(1):87–94, 2013.

[4] Jithendra Reddy Dandu, Arun Prasath Thiyagarajan, Pallikonda Rajasekaran Murugan, and Vishnuvarthanan Govindaraj. Brain and pancreatic tumor segmentation using srm and bpnn classification. *Health and Technology*, 10(1):187–195, 2020.

[5] El-Sayed M El-kenawy, Hattan F Abutarboush, Ali Wagdy Mohamed, and Abdelhameed Ibrahim. Advance artificial intelligence technique for designing double t-shaped monopole antenna. *Computers, Materials & Continua*, 69(3), 2021.

[6] El-Sayed M El-kenawy, Fahad Albalawi, Sayed A Ward, Sherif SM Ghoneim, Marwa M Eid, Abdelaziz A Abdelhamid, Nadjem Bailek, and Abdelhameed Ibrahim. Feature selection and classification of transformer faults based on novel meta-heuristic algorithm. *Mathematics*, 10(17):3144, 2022.

[7] El-Sayed M El-Kenawy, Seyedali Mirjalili, Abdelaziz A Abdelhamid, Abdelhameed Ibrahim, Nima Khodadadi, and Marwa M Eid. Meta-heuristic optimization and keystroke dynamics for authentication of smartphone users. *Mathematics*, 10(16):2912, 2022.

[8] El-Sayed M El-Kenawy, Seyedali Mirjalili, Fawaz Alassery, Yu-Dong Zhang, Marwa Metwally Eid, Shady Y El-Mashad, Bandar Abdullah Aloyaydi, Abdelhameed Ibrahim, and Abdelaziz A Abdelhamid. Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems. *IEEE Access*, 10:40536–40555, 2022.

[9] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

[10] Abdelhameed Ibrahim, Seyedali Mirjalili, Mohammed El-Said, Sherif SM Ghoneim, Mosleh M Al-Harthi, Tarek F Ibrahim, and El-Sayed M El-Kenawy. Wind speed ensemble forecasting based on deep learning using adaptive dynamic optimization algorithm. *IEEE Access*, 9:125787–125804, 2021.

[11] Swati Jayade, DT Ingole, and Manik D Ingole. Review of brain tumor detection concept using mri images. In *2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET)*, pages 206–209. IEEE, 2019.

[12] Dongjin Kwon, Russell T Shinohara, Hamed Akbari, and Christos Davatzikos. Combining generative models for multifocal glioma segmentation and registration. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014: 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part I 17*, pages 763–770. Springer, 2014.

[13] Garam Lee, Kwangsik Nho, Byungkon Kang, Kyung-Ah Sohn, and Dokyoon Kim. Predicting alzheimer's disease progression using multi-modal deep learning approach. *Scientific reports*, 9(1):1952, 2019.

[14] Khai Yin Lim and Rajeswari Mandava. A multi-phase semi-automatic approach for multisequence brain tumor image segmentation. *Expert systems with applications*, 112:288–300, 2018.

[15] David N Louis, Arie Perry, Pieter Wesseling, Daniel J Brat, Ian A Cree, Dominique Figarella-Branger, Cynthia Hawkins, HK Ng, Stefan M Pfister, Guido Reifenberger, et al. The 2021 who classification of tumors of the central nervous system: a summary. *Neuro-oncology*, 23(8):1231–1251, 2021.

[16] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[17] Atul Mishra, Abhishek Rai, and Akhilesh Yadav. Medical image processing: A challenging analysis. *International Journal of Bio-Science and Bio-Technology*, 6(2):187–194, 2014.

[18] Ceena Modarres, Nicolas Astorga, Enrique Lopez Droguett, and Viviana Meruane. Convolutional neural networks for automated damage recognition and damage type identification. *Structural Control and Health Monitoring*, 25(10):e2230, 2018.

[19] Reabal Najjar. Redefining radiology: a review of artificial intelligence integration in medical imaging. *Diagnostics*, 13(17):2760, 2023.

[20] Maria Nazir, Sadia Shakil, and Khurram Khurshid. Role of deep learning in brain tumor detection and classification (2015 to 2020): A review. *Computerized medical imaging and graphics*, 91:101940, 2021.

[21] Ahmad Ozair, Vivek Bhat, Reid S Alisch, Atulya A Khosla, Rupesh R Kotecha, Yazmin Odia, Michael W McDermott, and Manmeet S Ahluwalia. Dna methylation and histone modification in low-grade gliomas: Current understanding and potential clinical targets. *Cancers*, 15(4):1342, 2023.

[22] Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.

[23] Kevin Pierre, Manas Gupta, Abheek Raviprasad, Seyedeh Mehrsa Sadat Razavi, Anjali Patel, Keith Peters, Bruno Hochhegger, Anthony Mancuso, and Reza Forghani. Medical imaging and multimodal artificial intelligence models for streamlining and enhancing cancer care: opportunities and challenges. *Expert Review of Anticancer Therapy*, 23(12):1265–1279, 2023.

[24] Pedro P Rebouças Filho, Antonio C da Silva Barros, Jefferson S Almeida, JPC Rodrigues, and Victor Hugo C de Albuquerque. A new effective and powerful medical image segmentation algorithm based on optimum path snakes. *Applied Soft Computing*, 76:649–670, 2019.

[25] Jeffrey D Rudie, Andreas M Rauschecker, R Nick Bryan, Christos Davatzikos, and Suyash Mohan. Emerging applications of artificial intelligence in neuro-oncology. *Radiology*, 290(3):607–618, 2019.

[26] Nagwan Abdel Samee, El-Sayed M El-Kenawy, Ghada Atteia, Mona M Jamjoom, Abdelhameed Ibrahim, Abdelaziz A Abdelhamid, Noha E El-Attar, Tarek Gaber, Adam Slowik, and Mahmoud Y Shams. Metaheuristic optimization through deep learning classification of covid-19 in chest x-ray images. *Computers, Materials & Continua*, 73(2), 2022.

[27] Liyue Shen and Timothy Anderson. Multimodal brain mri tumor segmentation via convolutional neural networks. *vol*, 18:2014–2015, 2017.

[28] Mohammad Tanveer and Ram Bilas Pachori. *Machine intelligence and signal analysis*, volume 748. Springer, 2019.

[29] Nicholas J Tustison, KL Shrinidhi, Max Wintermark, Christopher R Durst, Benjamin M Kandel, James C Gee, Murray C Grossman, and Brian B Avants. Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with antsr. *Neuroinformatics*, 13:209–225, 2015.

[30] Gregor Urban, M Bendszus, F Hamprecht, J Kleesiek, et al. Multi-modal brain tumor segmentation using deep convolutional neural networks. *MICCAI BraTS (brain tumor segmentation) challenge. Proceedings, winning contribution*, pages 31–35, 2014.

[31] Xiao Xuan and Qingmin Liao. Statistical structure analysis in mri brain tumor segmentation. In *Fourth International Conference on Image and Graphics (ICIG 2007)*, pages 421–426. IEEE, 2007.

[32] Yang Yang, Lin-Feng Yan, Xin Zhang, Yu Han, Hai-Yan Nan, Yu-Chuan Hu, Bo Hu, Song-Lin Yan, Jin Zhang, Dong-Liang Cheng, et al. Glioma grading on conventional mr images: a deep learning study with transfer learning. *Frontiers in neuroscience*, 12:804, 2018.

[33] Darko Zikic, Yani Ioannou, Matthew Brown, and Antonio Criminisi. Segmentation of brain tumor tissues with convolutional neural networks. *Proceedings MICCAI-BRATS*, 36(2014):36–39, 2014.