



BRAIN TUMOR DETECTION AND CLASSIFICATION USING DEEP FEATURE FUSION AND STACKING CONCEPTS

Saqlain Raza¹, Nasim Gul², Haider Ali Khattak³, Arisha Rehan⁴, Muhammad Imran Farid⁵, Anum Kamal⁶, Dr Jai Singh Rajput⁷, Sajid Mukhtiar^{8*}, Aziz Ullah⁹

^{1,8*}Department of Control Science and Engineering, Xi'an University of Architecture and Technology, Xi'an, Shaanxi 710055, China

²Specialist Neurosurgery Zayed Military Hospital - Abu Dhabi

³Consultant Neurosurgeon, Neurosurgery Department, Ayub Medical Complex, Abbottabad - Pakistan

^{4,6}MBBS Jinnah Sindh Medical University, Karachi - Pakistan

⁵Department of Electrical and Computer Engineering, Air University, Islamabad - Pakistan

⁷Independent Researcher Family Medicine, Montgomery - Alabama

⁹IMS University of Science and Technology, Bannu KPK - Pakistan

***Corresponding author:** Sajid Mukhtiar

*Department of Control Science and Engineering, Xi'an University of Architecture and Technology, Xi'an, Shaanxi 710055, China, E-mail: sajidmukhtiar5@gmail.com

Abstract:

The Classification of brain tumors plays an important role in determining the treatment plan, course of therapy and survival rate. A new technique is proposed in this work for classification of brain tumors based on pre-trained neural networks and a stacking algorithm. Then our method begins with drawing multiple pre-train CNNs on T1 weighted images of MR brain scans where it extracts features from these. Afterwards, an ensemble of these features are used as input to a single layer stacking algorithm, which stacks together the predictions of several base classifiers to arrive at the final prediction. We evaluate our method on two publicly available datasets of brain MRI scans and show it can detect lesions with superior accuracy compared to other methods. In our approach using a pre-trained CNN allows us to leverage the transfer learning concept because the CNN had been trained in advance on a huge image database and extracted relevant features for the task of classifying brain tumors. An enhanced accuracy is achieved through a combination of various base classifiers with a stacking algorithm. The results of our study demonstrate that we have a promising method of categorizing brain tumors and improving healthcare provision.

Keywords: transfer learning; deep learning; ensemble learning; brain tumor classification; PCA; machine learning

1. Introduction

- 1) Making generalization, reviewing of preview research
- 2) Indicate a Gap, field knowledge
- 3) Outlining purposes, announcing present findings, indicate structure.

The brain is the most intricate organ in the human body. It comprises over 100 billion nerve cells and regulates the whole neurological system[1]. This crucial organ is originated in the cerebral cortex of the brain. Thus, any brain anomaly might be dangerous for human health; the deadliest of these

conditions is a brain tumor. A brain tumor can be either primary or secondary. A brain tumor that occurs in the brain is known as primary tumor while a secondary brain tumor develops when malignant cells travel through the bloodstream to the brain and settle there[2]. Brain tumors can be treated in a variety of methods, depending on the size, location and kind of the tumor. Currently, surgery is the most popular method of treating brain tumors since it has no detrimental effects on the brain[7]. It has been effectively adapted to diagnose, analyze, monitor and treat disorders in the human body using medical imaging technologies such as X-rays, ultrasound imaging (UI), computed tomography (CT) and magnetic resonance imaging (MRI).

The radiologists prefer MRI over all other available technologies because, it offers extensive microscopic chemical and physical information of the human body at the molecular level. Due to its great resolution, MRI is more effective than other technologies in diagnosing and classifying diseases. Consequently, MRI scans are frequently employed in medical applications to classify brain tumors.

Recently, early tumor detection using brain MRI imaging has gained considerable importance and is regarded as a lifeline for patients with brain tumors. However, it is vital to classify brain tumors; knowing the kind of tumors is equally necessary in order to enhance patient survival and recommend appropriate therapy. The brain MR images can be categorized as normal and abnormal; a variety of techniques have been reported for effectively classifying brain tumors utilizing incredibly high-resolution brain MRI images with appropriate contrast. The current investigations provide their insights into how brain MR imaging is perceived. The robustness of conventional machine learning (ML) techniques is constrained by their reliance on manually produced features. However, because deep learning-based algorithms automatically identify relevant traits, their performance is far higher. Though, it is difficult to acquire the huge amount of data required for deep learning-based algorithms for training. In this study, we proposed an innovative approach for extracting robust and discrete deep features from brain MR images using a variety of deep convolutional neural networks (CNNs) that have already been trained on ImageNet dataset. The extracted deep features are then subjected to principal component analysis (PCA) to reduce their dimensionality, and before different machine learning (ML) classifiers are used to distinguish between normal and abnormal brain MR images, their hyper-parameters are tuned using the GridSearchCV algorithm to boost their efficiency even more. According to the proposed criteria, the performance of 9 pre-trained CNN models is examined along with 10 distinct Machine learning classifiers. In this study, we fused the top-3 deep features to examine the robustness of combining deep features from 3 distinct pretrained deep models. The features fusion process assists to gather more diverse and robust features than using a single CNN model. Several ML classifiers are then fed the combined top-3 features to predict each output separately, and then the top 5 classifiers are selected to perform the single layer stacking algorithm. Finally, concatenate features obtained through deep learning are inputted into the single-layer stacked classifiers to predict the final output. We used two distinct brain tumor datasets and provided a detailed evaluation of 9 Pre-trained CNN models and 10 distinct Machine Learning classifiers. We named the small dataset as BTS and large dataset as BTL. Both datasets contain two classes (tumor/normal). The outcomes of our experiment show that a combination of deep features and a single layer stacking algorithm can significantly boost model performance. The summary of our main contributions is summed up as follow:

- The overall classification accuracy is enhanced by introducing a new automated approach that can replace traditional brain tumor classification.
- The quality of brain MR images is enhanced by employing an efficient strategy.
- To boost the classification accuracy on a small dataset, a data augmentation strategy is applied, and the influence of over-fitting on classification performance is investigated.
- Pre-trained CNN models are employed for robust and discriminative features extraction.
- PCA algorithm is applied on the acquired deep features from pre-trained CNN models to perform dimensionality reduction.
- Single layer stacking technique is employed on the selective ML classifiers.

- Top-3 deep features are concatenated and subjected to the single layer stacking algorithm to attain the highest level of performance.
- Brain tumor classification using the proposed method is compared to cutting-edge methodologies. The rest of paper is organized as follows: Section (2) elaborates the relevant research work, Section (3) briefly explains the methodology of our proposed framework, Section (4) contains the experimental results, Section (5) includes a brief discussion, and at last, Section (6) offers the conclusion, limitation as well as future work.

2. Related Work

Researchers have proposed numerous traditional and automated techniques for addressing the brain tumor classification problem. This section covers some of the most recent findings.

In traditional machine learning, data preprocessing, feature extraction, feature selection, model training, and model testing are common steps. However, the accuracy of classification primarily depends on the features extracted from input images, making feature extraction the most critical step. Low-level features, including statistical measures, texture features, and intensity, as well as high-level features such as SIFT, SURF, BoW, and HoG, are the two types of feature sets employed in traditional ML. The choice of features is determined by the problem being addressed and the available data. The effectiveness of traditional ML approaches is highly reliant on the chosen features and their capacity to capture the essential information in the images. For instance, Ahmed et al. [8] utilized a noise-reduction approach to extract essential image features by utilizing gray-level co-occurrence matrix (GLCM) characteristics. Brain tumor segmentation was conducted through Discrete Wavelet Transform (DWT) to improve the efficiency of the process by reducing complexity. The DWT was applied to decompose the original image into different sub-bands, which were then filtered by thresholding. Finally, the segmented image was obtained by applying an inverse DWT. The study reported that the proposed method improved the accuracy of brain tumor segmentation by up to 95%. Cheng et al.[9]proposed a brain tumor classification method that involved the use of multi-scale MRI images of the brain. They utilized image dilation techniques to identify regions of interest (ROI) in the MR images, and augmented the tumor area with fine-ring-shaped features to improve the classification accuracy. The team then extracted features from the images using three different feature extraction techniques, namely intensity histogram, GLCM, and BoW models, and evaluated their effectiveness. They evaluated their model using a T1-weighted CE-MRI dataset obtained from General and Nanfang Hospitals, China, and achieved accuracies of 87.54%, 89.72%, and 91.28% for the three different feature extraction methods, respectively.

Researchers have developed several methods to eliminate undesired regions of the brain in medical imaging. Kumar et al. [10] presented a hybrid approach to classify brain tumor magnetic resonance images (MRIs). Their approach involves extracting a feature matrix using the Discrete Wavelet Transformation (DWT), followed by reducing the number of features with Principal Component Analysis (PCA). Finally, the normal and abnormal brain MR images are classified using a Support Vector Machine (SVM) classifier. They claimed 90.9 % accuracy on the SICAS Medical Image Repository brain MR Images dataset. Bahadure and NileshBhaskarrao Ray et al. [11] Firstly, applied Wiener Filter on brain MR images to remove unwanted noise. Secondly, the feature matrix was extracted by Histogram and Co-occurrence Matrix then they used Principal Component Analysis (PCA) for dimensionality reduction. Finally, authors employed a kernel-based Support Vector Machine (SVM) to classify normal and pathological brain MRI images. The study was conducted using a T1-weighted brain MRI dataset, obtained from the Harvard Medical School Website, and reported an accuracy of 94%, with a fraction of 95%, a similarity index of 96.20%, and a total error of 7.5%. An additional fraction of 0.025% was also reported by the authors.

Conversely, machine learning techniques such as deep learning utilize neural networks to extract features directly from the input data, thereby eliminating the need for a separate feature extraction step. During the training process, the network learns to identify the most relevant features for the given classification task. This is especially advantageous when dealing with intricate and multidimensional data, such as brain MRI scans, where selecting significant features can be a difficult

task for human experts. For instance, Jiang et al. [12] proposed a brain tumor classification approach using T1-weighted MRI scans. The authors utilized a region-based active contour model to extract regions of interest (ROIs) in the brain MRI images. After that, a set of intensity and texture features were extracted from the segmented ROIs. The feature space was reduced through Principal Component Analysis (PCA) to address the curse of dimensionality. Finally, the authors employed the Random Forest (RF) algorithm for brain tumor classification. The dataset used in this study was obtained from the Cancer Imaging Archive (TCIA). The proposed method achieved an accuracy of 93.5% in the classification of brain tumors. Yao et al. [13] have utilized fuzzy cognitive maps (FCMs) to develop a model for the classification of low-grade and high-grade gliomas. The study involved the use of magnetic resonance imaging (MRI) data from the Cancer Imaging Archive (TCIA). The proposed model employed FCMs to represent the complex relationships between various imaging features and the tumor grade. The study reported an accuracy of 87.5% in the classification of low-grade and high-grade gliomas using the proposed FCM-based approach. The brain MRI images were segmented and classified using backpropagation neural network (BPN) and radial basis function neural network (RBFN), as proposed by Deep and Devi et al.[14]. First, optimal textural features were extracted by using statistical approach then both (BPN) and (RBF) classifiers were applied to examine their efficiencies. The (RBFN) algorithm performed better than (BPN) with 85.71% accuracy. The statistical approach to extract textural features of MRI brain images, and then BPN and RBF classifiers are used for segmentation and classification of the infected region of the brain. Authors claimed 85.71% accuracy, over all.

Recent advances in machine learning and computer vision have led to the development of various techniques for medical image analysis, including brain tumor segmentation. In a study conducted by Khened et al. [15], they proposed a deep learning-based approach for brain tumor segmentation in magnetic resonance images (MRI). The proposed approach used a convolutional neural network (CNN) architecture called U-Net, which was specifically designed for biomedical image segmentation. The U-Net architecture consisted of an encoder and a decoder, which helped in extracting relevant features and reconstructing the segmentation map, respectively. The study reported promising results, achieving an average dice score of 0.87 on the BraTS 2017 dataset, which is a benchmark dataset for brain tumor segmentation. Soltaninejad M et al. [16] devised an automated approach for detecting and segmenting aberrant tissues associated with brain tumors (edema and tumor core). The superpixel technique and superpixel classification methodology has implemented in the proposed framework. Numerous image features including Gabor textons, intensity, curvatures, and fractal analysis are calculated to ensure the robust classification. The classification task is performed using two different types of classifiers, namely Support Vector Machine (SVM) and Extremely Randomized Trees (ERT), and the outcomes of these classifiers are compared to determine their effectiveness. The proposed method had trained and tested on the BRATS (2012) dataset, the corresponding evaluation results are 88.09% and 0.88% respectively.

Recently, different researchers have combined traditional and automated learning methods to classify brain MRI images. Integrating traditional and automated learning techniques in classifying brain MRI images can improve model performance in several ways. Traditional techniques can help identify key features in the images that are crucial in classification. Automated learning techniques, on the other hand, can help to optimize the classification process, making it more efficient and accurate. For instance, Li et al. [17] proposed a hybrid approach that combined traditional and automated learning methods for the classification of brain MRI images. The proposed method employed a combination of wavelet-based texture analysis, gray-level co-occurrence matrix (GLCM) features, and deep learning-based features extracted using a convolutional neural network (CNN). A support vector machine (SVM) classifier was then used to classify the brain MRI images. The proposed method was tested on a publicly available dataset and achieved an overall classification accuracy of 93.5%, indicating the effectiveness of the hybrid approach in the classification of brain MRI images.

Based on a comprehensive review of the literature, it is evident that there has been a significant amount of research conducted on the segmentation, feature extraction, and classification of brain images obtained from both MRI and CT scans. Researchers have proposed various models and

techniques for detecting, segmenting, and classifying brain tumors. Some of these models have been proposed with the aim of achieving greater accuracy and efficiency in the diagnosis and treatment of brain tumors. Certain researchers have proposed models that focus solely on the segmentation of the tumor region in the brain, while others have developed models that can handle both segmentation and classification tasks simultaneously. Additionally, most of the researchers performed above mentioned tasks by traditional way of machine learning. They manually handled all the recommended steps and then obtained their results. Some of them, fused different traditional models together and improved their overall accuracy rate. Numerous researchers worked the same on deep learning side as well. They performed the same task either by using single model or ensemble different deep learning models.

3. Materials and Methods

This research focuses on a new knowledge-based system for the diagnosis of brain tumors using two different datasets via image-preprocessing, features extraction, dimensionality reduction, features fusion, ML classifiers ensemble, prediction and classification techniques. This section has been divided into certain sub-sections: We first preprocessed the MR Images of both training and validation datasets. In Section 3.1, the MR images undergo preprocessing by cropping, resizing, and normalization of their values. The preprocessed images then undergo augmentation in Section 3.2 before being fed into pre-trained CNNs to extract features through deep models in Section 3.3. To reduce the high dimensionality of the correlated deep features space, Principal Component Analysis (PCA) is utilized in Section 3.4 to convert it into a low dimensional uncorrelated feature space. The low-dimensional features are then evaluated using multiple ML classifiers in Section 3.5, and their hyperparameters are fine-tuned using GridSearchCV technique in Section 3.6. Finally, the performance of the proposed method is analyzed in Section 3.7. Figure 1 depicts the primary schematic of our proposed farmwork.

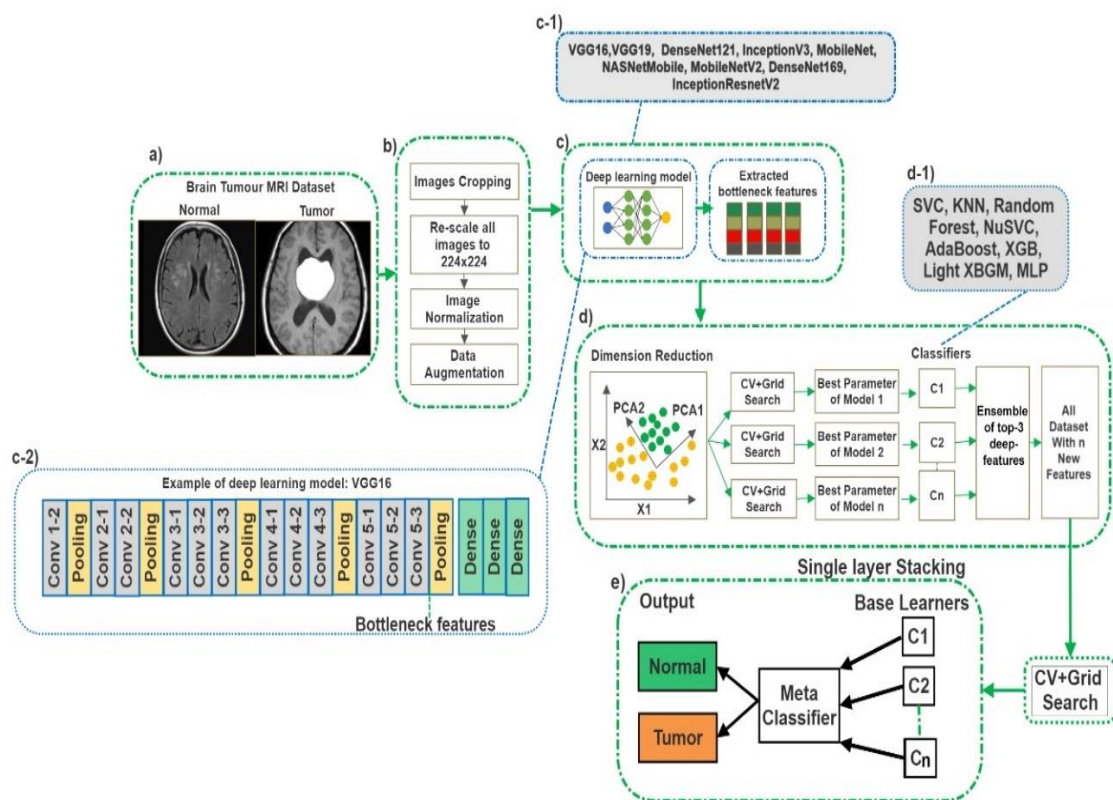


Figure 1 The main diagram of our proposed model

3.1. Pre-Processing

The brain MRI datasets being considered contain unwanted areas that can impact the model's performance. To address this, a cropping method is utilized. The method involves identifying the extreme points along the brain's contour in the image [18]. The process is shown in Figure 2. First, the original MRI dataset is loaded as input. Next, a threshold is applied to separate the brain region from the rest of the image, converting it to a binary image. Then, erosion and dilation operations are used to remove noise and the image is slightly blurred. The extreme points are determined by finding the smallest and largest x-coordinates in the brain's contour. The same process is repeated for the y-coordinate, giving the north and south coordinates. These extreme points are used to encircle the most prominent contour of the threshold image and the image is cropped.

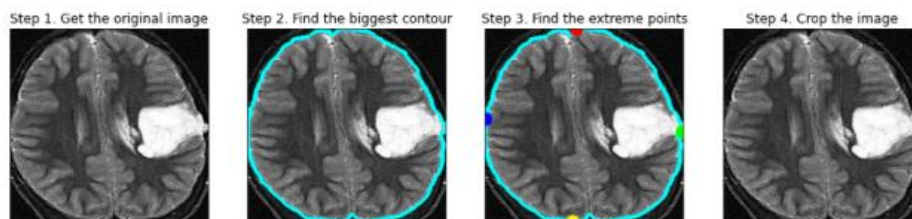


Figure 2. Procedure to crop unwanted voids and regions in MR image.

In this work, pre-trained CCN models are used to extract the feature space. The pre-trained models accept input images with 224 x 224 pixels in size except for inception-V3 which needs a 299 x 299 pixelated input image. Hence, the cropped images have been resized by either 224 x 224 or 299 x 299 pixels.

3.2. Image Augmentation

The concept of data augmentation comes when we deal with the small dataset. Optimizing the data can be done through the inclusion of slight modifications of existing data or by generating new synthetic data utilizing the existing knowledge of the dataset [19]. To prevent overfitting the machine learning model during training, a variety of data augmentation strategies are applied. Also, it enhances the model's overall performance. By enhancing the data, we can configure and fine-tune several settings. In this work, we augmented the BTS MRI dataset by applying left/right mirroring, flipping the image around both the x-axis and y-axis, adding some noise and applying 15-degree rotation. We increased the BTS images from 253 to 977 images.

3.3. Deep Features Extraction

3.3.1. Convolutional Neural Networks

The major applications of a convolutional neural network (CNN), which comprises one or more convolutional layers, are image processing, segmentation, classification, and other auto correlated data. The convolutional layers are essentially a filter that slide all over the input and fetch useful information [20,21]. To understand this concept much better is looking at smaller areas of the image might be more productive than scanning the entire image for specific traits. In order to lower the overall number of parameters, CNN's convolutional layers employ a weight-sharing technique [22,23]. Typically, CNN is made up of three components: First, a convolutional layer that is used to fetch temporal and spatial characteristics. Second, a max-pulling layer that is used for down-sampling the dimensionality of extracted features. Last, a fully connected layer that is used to categorize the data. The architecture of convolutional neural network is represented in Figure 3.

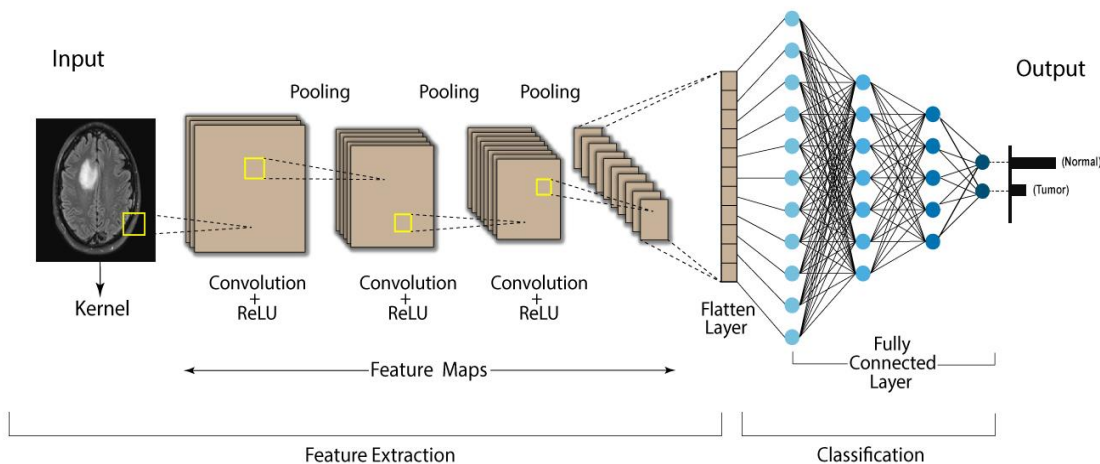


Figure 3.Depicts the design of Convolutional Neural Networks.

3.3.2. Transfer Learning

Training a neural network on a big dataset might take weeks. Fortunately, this time may be reduced by using model weights from previously trained models. Transfer learning is the process of applying a previously trained model to a fresh problem[24]. It is presently very well-liked in deep learning due to its capacity to train deep neural networks with only a modest amount of data. In general, CNN performs better on bigger datasets than on smaller ones. In scenarios where obtaining a substantial training dataset is not feasible, utilizing transfer learning can be an alternative. Transfer learning decreases the lengthy training period typically needed to construct deep learning models from the ground up and eliminates the need for a sizable training dataset[25,26]. The conceptual representation of transfer learning is illustrated in Figure 4.

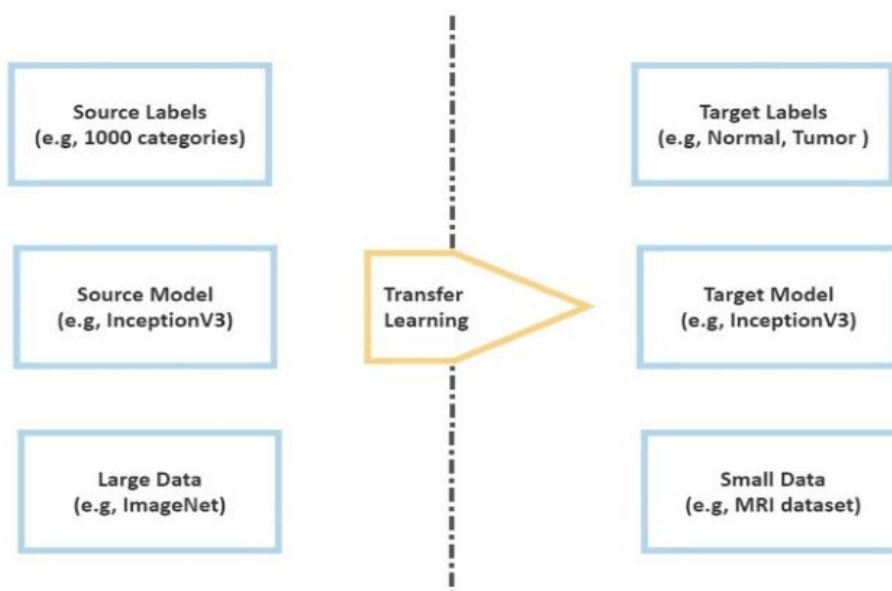


Figure 4 Transfer learning approach.

3.3.3. Deep Feature Extraction

The technique of turning raw data into numerical features that can be handled while keeping the information in the original data set is known as feature extraction[27]. The deep neural networks' multiple convolutional layers enable us to extract from the input image both low-level and high-level characteristics. The MRI datasets that are used in this work are not particularly large and since it is sometimes impractical to train and optimize deep CNNs like InceptionResNetV2 from scratch, we

use a transfer learning approach to utilize pre-trained deep models to extract features from MRI scans. These CNNs are first trained on the ImageNet dataset to acquire a set of fixed weights for each model. The pretrained deep models that are used in this study include InceptionResnetV2[28], DenseNet[29], VGG[30], Inception V3[31], NASNetMobile[32], MobileNetV2[33], and MobileNet[34].

3.4. Dimensionality Reduction

Large data sets are becoming more common, and they are frequently difficult to understand. PCA is a method for reducing the dimensionality of such datasets while preserving information. The PCA approach primarily transforms the feature space through the connection between attributes and maps the original feature space to the low-dimensional feature space [35,36]. PCA employs an unsupervised learning dimensionality reduction approach to correlate multidimensional data groups in order to reduce the data dimensions. It eliminates all parameter restrictions, simplifies the data structure, makes the data set easier to use, and lowers algorithm computation costs by decreasing information loss[37,38]. Eigenvectors with low eigenvalues are eliminated from the transformation matrix for data reduction. As a result, PCA shrinks the size of the eigenvector space to which data is projected. In the current work, various deep feature extractors are used that is why every feature extractor produces different range of features but the number of manually selected PC is set 1250 for BTL and 800 for BTS.

3.5. Machine Learning Classifiers

Several ML classifiers, including K-Nearest Neighbors (k-NN), NuSVC, AdaBoost, SVM with four discrete kernels, Random Forest (RF), XGBoost, Light GBM, and Multilayer Perceptrons (MLPs), are used to examine the deep features acquired by pre-trained models. All of above are integrated with the help of ML library [39]. The subsequent paragraphs elaborate the ML classifiers and their hyper-parameters more in detail.

3.5.1. Adaboost

AdaBoost contains the set of weak learners that are trained iteratively, with the weights of misclassified examples being adjusted at each step to increase the focus on difficult example. This process continues until a satisfactory level of accuracy is achieved, at which point the final model is a weighed combination of all individual weak learners. categories that are initially trained to identify misclassified data then their weights are boosted iteratively[40]. By increasing the number of estimators to 200, we were able to possibly achieve higher precision than other classifiers[41].

3.5.2. K-Nearest Neighbors

K-Nearest Neighbors algorithm is a versatile, non-parametric, instance-based method that performs well in classification and regression tasks, especially when the data is high-dimensional. However, it's computational complexity, sensitivity to noisy features and the importance of finding the optimal parameter, K, make it less desirable for datasets with large number of instances[42]. K-NN uses Euclidean distance and Manhattan distance to determine K objects which are most similar to a fresh sampling of data, and then decides using the majority decision of its k neighbors. The majority vote of its k neighbors determines the outcome. [43].

3.5.3. Random Forest

A random forest classifier, also known as an ensemble of decision trees, is a machine learning algorithm that uses multiple decision trees to make predictions [44]. The classifier builds multiple decision trees using a random subset of the data, and each tree makes its own prediction. The final prediction is made by taking the majority vote of all the trees. This approach helps to reduce overfitting, as well as increase the accuracy of the model. The random forest classifier can be used for both classification and regression problems and is considered one of the most robust and accurate machine learning algorithms. Additionally, it can handle high-dimensional data and missing

values[45]. It is widely used in various fields such as bioinformatics, image processing, and natural language processing[46].

3.5.4. Support Vector Machine (SVM)

It is a supervised learning technique that uses to classify or cluster the higher dimensional data. It allows using of different kernel functions according to the nature of the data. There are four kinds of SVC kernel options available like Linear (which is the linear regression technique), Poly (for nonlinear or poly regression), Sigmoid and RBF. In this work, we used all four kernel functions of SVM algorithm[47]. Their mathematical representation is shown in Table.1.

Table 1.SVM kernel types, mathematical representations and the prerequisites for each.

Kernel	Mathematical Representation	Hyper-Parameters
Linear	$K(x_n, x_i) = (x_n, x_i)$	
Poly	$K(x, y) = (x^T y + C)^d$	C
Sigmoid	$K(x_n, x_i) = (x_n, x_i) + C$	γ, C
RBF	$K(x_n, x_i) = \exp(-\gamma x_n - x_i ^2 + C)$	γ, C

3.5.5. NuSVC

Nu Support Vector Classification, or NUSVC, is a variant of the Support Vector Classification (SVC) algorithm that is used for classification tasks. NuSVC allows to a user to specify the number of support vectors through the use of a parameter called “nu,” which can be useful in situation where the data contains a high level of noise or outliers. NuSVC can be represented as the optimization problem of finding the hyperplane in a high-dimensional space that maximally separates different classes of data point, while minimizing the complexity of the model. This is achieved through the minimization of the sum of squared errors between the predicted and actual values, subject to a set of constraints that ensure that the points are correctly classified. Overall, NuSVC is a powerful and widely used algorithm for classification tasks that allows for the flexibility to control the number of support vectors through the use of the “nu” parameter[48]

3.5.6. XGBoost

The decision-tree-based ensemble method which is also known as XGBoost, in which it follows an iterative procedure to create a strong classifier based on weak learners. It Keeps correcting the errors of previous predictors until the training data is correctly predicted by the final best model. It provides parallel tree boosting and effectively addresses a variety of data science issues.

XGBoost is an efficient and effective algorithm for supervised learning problems, specifically for classification and regression. Its ability to handle large datasets, high-dimensional spaces and the built-in regularization helps prevent overfitting, but its complexity and the importance of parameter tuning makes it a challenging algorithm to use[49].

3.5.7. ML Classifiers Ensemble

We combined the top five-base learners at a base-level and then provided their outputs to a meta classifier as an input. The final class was obtained by applying the hard-voting technique. The algorithm representation is shown in algorithm 1. It was noticed that the accuracy was equal to or greater than the base learners' greatest accuracy[50].

<p>1:Input: training data $D = \{x_i, y_i\}^{m i=1}$ ($x_i \in R^n, y_i \in Y$) 2:Output: ensemble classifier H 3:<i>Step 1: learn a base-level classifiers</i> 4:for $t = 1, 2, 3, \dots, T$ do 5: learn a base level classifier h_t based on D 6:end for 7:<i>Step 2: construct new dataset of predictions</i></p>

```

8:for  $i = 1, 2, 3, \dots, mdo$ 
9: $D_h = \{x'_i, y_i\}$ , where  $x'_i = \{h_1(x_i), \dots, h_T(x_i)\}$ 
10:end for
11:Step 3: learn a meta-classifier
12:learn  $H$  based on  $D_h$ 
13:return  $H$ 

```

Algorithm 1: Stacked Ensemble

3.6. GridSearchCV

In this study, GridSearchCV technique is utilized to optimize the hyperparameters of several machine learning classifiers. By systematically searching through different combinations of hyperparameters, the best set of hyperparameters was found for each model and dataset. The effectiveness of machine learning models was improved because to this method. By determining the ideal hyperparameters for a specific task, the models were able to achieve higher accuracy and better generalization on new data. When using GridSearchCV, there are two critical fundamental terms to understand: the estimator and parameter grid. The estimator is the classifier that will be trained and is passed as an argument to GridSearchCV. The parameter grid is a Python dictionary that specifies the hyperparameters to be optimized for the estimator. It uses parameter names as keys and lists of parameter settings as values.

GridSearchCV then systematically searches through all possible combinations of the hyperparameters specified in the parameter grid to achieve the highest accuracy for the estimator. This process involves training and evaluating the estimator using cross-validation on different subsets of the data. By comparing the performance of the estimator across all hyperparameter combinations, GridSearchCV can identify the optimal set of hyperparameters that result in the best performance for the given task. This technique is an essential tool for machine learning practitioners and researchers looking to optimize their models for specific tasks. All potential combinations of these elements are looked at to achieve the utmost accuracy. [51].The procedure of Grid Search CV is represented in Figure 5.

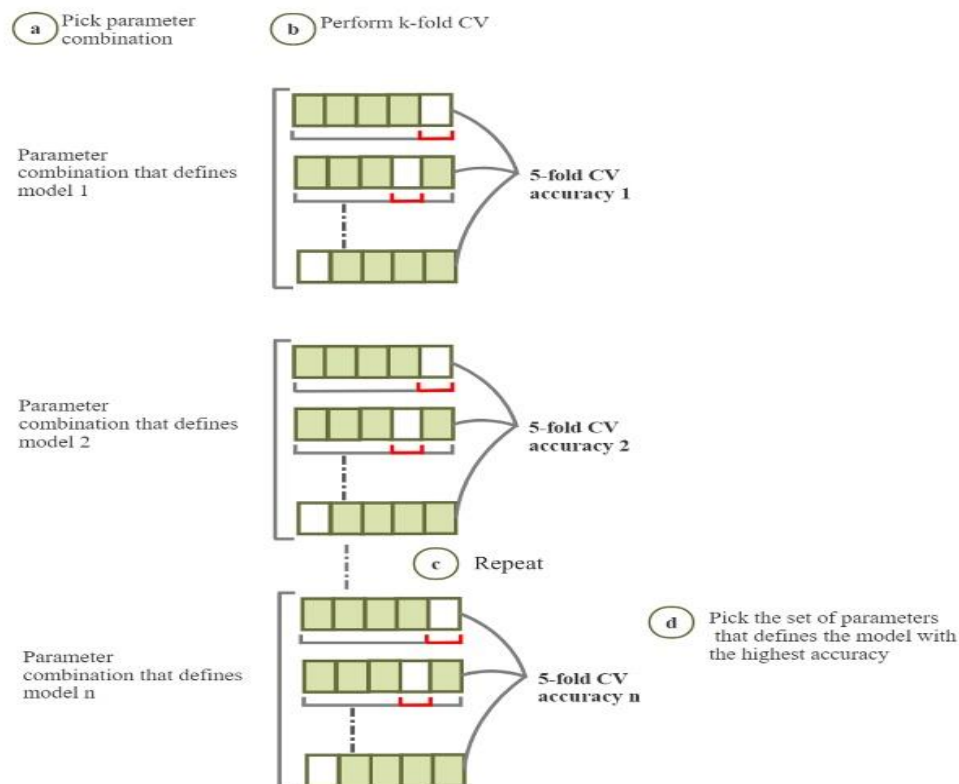


Figure 5. Representation of Grid Search CV process

3.7. Performance Analysis of the Proposed Method

Sensitivity measures the proportion of correctly identified positive events, including genuinely positive cases[52], while specificity measures the percentage of accurately identified negative cases. Accuracy, on the other hand, represents the percentage of correctly identified true positive and true negative cases of brain tumors. Sensitivity is a metric that exclusively focuses on positive cases. It indicates the proportion of correctly classified positive cases compared to the actual number of positive cases. A higher sensitivity value suggests a lower false negative rate.

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (1)$$

Specificity is a metric that solely deals with negative cases. It reflects the proportion of accurately classified negative cases in relation to the actual number of negative cases. A higher specificity value indicates a lower false positive rate.

$$\text{specificity} = \frac{TN}{TN + FP} \quad (2)$$

Positive predictive value (PPV) is a metric that is concerned solely with positive predictive cases. It indicates the proportion of correctly classified positive predictive cases compared to the total number of positive predictive cases.

$$\text{positive predictive value(PPV)} = \frac{TP}{TP + FP} \quad (3)$$

Negative predictive value (NPV) is a metric that exclusively focuses on negative predictive cases. It reflects the proportion of accurately classified negative predictive cases in relation to the total number of negative predictive cases. A higher NPV value indicates a lower false positive predictive rate.

$$\text{negative predictive value(NPV)} = \frac{TN}{TN + FN} \quad (4)$$

Accuracy is a metric that assesses the correctness of classification results. The system's efficiency is deemed high when accuracy value is greater.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

The F-Score is a performance metric that considers both Precision and Sensitivity equally. Between Precision and Sensitivity, it is determined as the harmonic mean.

4. Investigations and Findings

4.1. Dataset

Two publicly accessible brain MRI datasets that we used in a series of experiments were both acquired from the well-known website Kaggle. The first dataset consists of 253 images in total, out of which 98 images are without tumors and the rest of the 155 images contain tumors[53]. The second dataset contains three-thousands photos, of which fifteen- hundreds are tumor-free and remaining fifteen-hundreds are images containing tumors. [54]. We named these datasets BTS and BTL, respectively. At a ratio of 80% to 20%, we divided each dataset into a training set and a testing set. Figure 6 and Table 2 show the visual representation and details of both datasets.respectively.

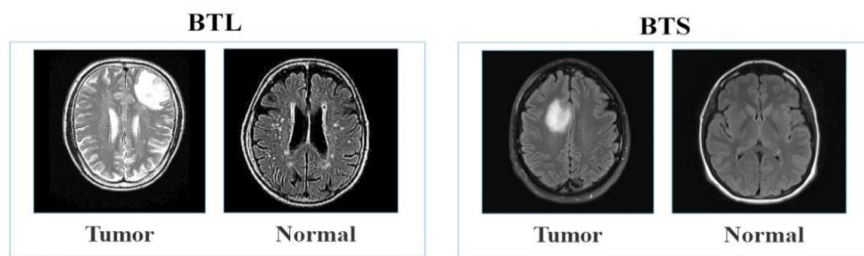


Figure 6. Brain MR images in datasets BTL and BTS.

Table 2.Details of brain MR datasets

Title	Training Set	Test Set	No. Classes
BTS	202	51	2
BTL	2400	600	2

4.2. Experimental Setup and Results

In this section, we analyzed the performance of our approach on two distinct datasets, BTS and BTL. The section is divided into several parts to provide a comprehensive analysis of our approach. Firstly, we compared the performance of nine pre-trained CNN networks on ten different machine learning classifiers. The results of this comparison are presented in Tables 3 and 4, respectively. In the next part of the section, we analyzed the effectiveness of features fusion from two pre-trained models. The experimental results of this analysis were presented in Tables 5 and 6. Finally, we examined the compatibility of our proposed model with the top three fused features, and the outcome of this analysis is presented in Tables 7 and 8. Overall, our analysis demonstrated the effectiveness of our approach and its ability to achieve high accuracy in classifying brain tumors using MRI scans.

Table 3.Accuracies of all participated deep features with ML classifiers on BTL dataset (* sign represents the superior accuracies)

PCA-Transformed Deep Features	Machine Learning Classifiers Accuracy											
	K-NN	SVM (Linear)	SVM (Poly)	SVM (RBF)	SVM (Sigmoid)	NuSVC	RF	AdaBoost	XGBoost	Light GBM	MLP	Average
VGG16 features	0.7272	0.8961	0.8571	0.8831	0.8961	0.8831	0.7922	0.7142	0.7792	0.7792	0.9090	0.8288
VGG19 features	0.8051	0.8701	0.8701	0.8441	0.8571	0.8701	0.8571	0.7402	0.7662	0.8051	0.8831	0.8335
InceptionV3 features	0.8311	0.7792	0.8571	0.8701	0.7922	0.8701	0.7532	0.8441	0.8311	0.8051	0.8831	0.8288
MobileNet features*	0.7792	0.9220	0.9350	0.8571	0.7532	0.8831	0.8960	0.8571	0.8961	0.9220	0.8571	0.8689
MobileNetV2 features*	0.7792	0.9090	0.9090	0.8871	0.8311	0.8961	0.8441	0.8311	0.8441	0.8571	0.8441	0.8575
NASNetMobile features	0.7792	0.8831	0.8831	0.8871	0.7272	0.8831	0.8051	0.8571	0.8181	0.8181	0.8701	0.8374
DenseNet121 features*	0.7792	0.9220	0.8961	0.8871	0.7922	0.8831	0.8181	0.8441	0.8961	0.8831	0.8571	0.8551
InceptionResNetV2 features*	0.6753	0.9220	0.8311	0.8871	0.8181	0.8831	0.8311	0.8311	0.8701	0.8571	0.8961	0.8457
DenseNet169 features*	0.8571	0.8571	0.8961	0.8871	0.8441	0.8831	0.8051	0.8441	0.8571	0.8441	0.8440	0.8566
Average	0.7734	0.8845	0.8861	0.8767	0.8124	0.8817	0.8051	0.8181	0.8398	0.8412	0.8715	

Note. The highest average accuracy is denoted by the bold text.

As shown in Table 3, K-NN performs the least well on the BTS dataset compared to other ML classifiers. This is because K-NN is very dependent on having a dense data. PCA is a linear dimensionality reduction method that is optimized for linearly separable data with correlated features, but KNN is a non-parametric method that doesn't make any assumptions on the underlying data distribution and it's sensitive to the scale of the features. This means that data that is not linearly

separable and has uncorrelated features may not perform well with PCA-transformed data. Another reason is that KNN classifiers rely on the distance between points in feature space to determine the class of a new point, so if the points are far apart in the original feature space but close together in the reduced feature space obtained from PCA, KNN may not be able to accurately classify them. Additionally, when PCA is applied to deep features, it can lead to a loss of information, which can negatively impact the performance of KNN. While PCA can be beneficial for some classification algorithms, it may not be the most effective choice for KNN when working with deep features. It is also noted in Table 3, the deep features acquired by pre-trained MobileNet and DenseNet models cause to produce exceptional results and maintain their superiority on the BTS dataset. This is due to the fact that the deep features gathered from MobileNet and DenseNet cover a broad spectrum of complexity. Consequently, it often provides more smooth decision limits, which can accurately forecast in the absence of sufficient training data. In contrast, the deep features acquired by VGG models yield inferior results for both MRI datasets. This is due to the absence of residual blocks in its architecture.

Table 4. Accuracies of all participated deep features with ML classifiers on BTL dataset (* sign represents the superior accuracies)

PCA-Transformed Deep Features	Machine Learning Classifiers Accuracy											
	K-NN	SVM (Linear)	SVM (Poly)	SVM (RBF)	SVM (Sigmoid)	NuSVC	RF	Ada Boost	XGBoost	Light GBM	MLP	Average
VGG16 features	0.92							0.861	0.92			
VGG19 features	83	0.9800	0.9716	0.9833	0.9533	0.9850	0.9116	6	16	0.9300	0.9833	0.9467
InceptionV3 features	0.89							0.888	0.91			
MobileNet features*	66	0.9800	0.9733	0.9800	0.9433	0.9700	0.9316	3	16	0.9233	0.9833	0.9437
MobileNetV2 features*	0.92							0.921	0.92			
DenseNet121 features*	16	0.9866	0.9850	0.9816	0.9666	0.9816	0.9066	6	83	0.9450	0.9783	0.9548
MobileNet features*	0.92							0.931	0.95			
MobileNetV2 features*	50	0.9916	0.9833	0.9116	0.9866	0.9850	0.9316	6	66	0.9683	0.9800	0.9592
DenseNet169 features*	0.93							0.935	0.93			
NASNetMobile features	00	0.9866	0.9783	0.9500	0.9683	0.9766	0.9116	0	83	0.9550	0.9750	0.9549
DenseNet121 features*	0.95							0.876	0.90			
InceptionResNetV2 features*	16	0.9733	0.9650	0.9883	0.9250	0.9766	0.9366	6	00	0.9450	0.9733	0.9455
DenseNet121 features*	0.95							0.921	0.93			
InceptionResNetV2 features*	66	0.9783	0.9883	0.9800	0.9683	0.9850	0.9466	6	00	0.9450	0.9933	0.9630
DenseNet169 features*	0.93							0.923	0.94			
DenseNet169 features*	66	0.9883	0.9800	0.9816	0.9466	0.9816	0.9183	3	50	0.9483	0.9783	0.9570
InceptionResNetV2 features*	0.97							0.923	0.95			
DenseNet169 features*	16	0.9866	0.9883	0.9733	0.9716	0.9833	0.9566	3	83	0.9550	0.9883	0.9669
Average	0.93	0.9834	0.9792	0.9694	0.9588	0.9805	0.9279	0.909	2	0.9299	0.9416	0.9814

Note. The highest average accuracy is denoted by the bold text.

The similar trends were observed in Table 4. Specifically, the pre-trained MobileNet, DenseNet models and SVM with linear kernel produced exceptional results and maintained their superiority on both datasets BTS and BTL.

Table 5. Accuracies of concatenated deep features with ML classifiers on BTS datasets

PCA-Transformed Deep Features	Machine Learning Classifiers Accuracy										
	K-NN	SVM (Linear)	SVM (Poly)	SVM (RBF)	SVM (Sigmoid)	NuSVC	RF	AdaBoost	XGBoost	Light GBM	MLP
MobileNet features*	0.7792	0.9220	0.9350	0.8571	0.7532	0.8831	0.8960	0.8571	0.8961	0.9220	0.8571
InceptionResNetV2 features*	0.6753	0.9220	0.8311	0.8871	0.8181	0.8831	0.8311	0.8311	0.8701	0.8571	0.8961
DenseNet169 features*	0.8571	0.8571	0.8961	0.8871	0.8441	0.8831	0.8051	0.8441	0.8571	0.8441	0.8440
MobileNet+InceptionResNetV2	0.7792	0.9220	0.9090	0.8961	0.7532	0.8961	0.8571	0.9090	0.8961	0.9220	0.8961

MobileNet+DenseNet169	0.7922	0.9220	0.9090	0.8961	0.7792	0.8961	0.8701	0.8961	0.8961	0.8961	0.9220
InceptionResNetV2+DenseNet169	0.8701	0.9220	0.9090	0.9090	0.8180	0.9090	0.8311	0.8571	0.8831	0.8701	0.9090
MobileNet+DenseNet169+InceptionResNetV2	0.7922	0.9220	0.9220	0.9090	0.8051	0.9090	0.8311	0.9220	0.9090	0.8961	0.9090

Note. The highest average accuracy is denoted by the bold text.

It can be seen in Table 5; an ensemble model fails to yield superior results for most of the ML classifiers on BTS dataset. This is due to the BTS dataset that generates insufficient training samples, which prevents an ensemble model from learning complex deep features.

Table 6. Accuracies of concatenated deep features with ML classifiers on BTL dataset.

PCA-Transformed Deep Features	Machine Learning Classifiers Accuracy										
	K-NN	SVM (Linear)	SVM (Poly)	SVM (RBF)	SVM (Sigmoid)	NuSVC	RF	AdaBoost	XGBoost	Light GBM	MLP
MobileNet features*	0.9250	0.9916	0.9833	0.9116	0.9866	0.9850	0.9316	0.9316	0.9566	0.9683	0.9800
InceptionResNetV2 features*	0.9366	0.9883	0.9800	0.9816	0.9466	0.9816	0.9183	0.9223	0.9450	0.9483	0.9783
DenseNet169 features*	0.9716	0.9866	0.9883	0.9733	0.9716	0.9833	0.9566	0.9233	0.9383	0.9550	0.9883
MobileNet+InceptionResNetV2	0.9416	0.9933	0.9900	0.8400	0.9866	0.9866	0.9366	0.9516	0.9516	0.9666	0.9816
MobileNet+DenseNet169	0.9533	0.9950	0.9883	0.8200	0.9883	0.9850	0.9500	0.9533	0.9583	0.9683	0.9866
InceptionResNetV2+DenseNet169	0.9966	0.9916	0.9850	0.9366	0.9816	0.9883	0.9483	0.9483	0.9466	0.9600	0.9816
MobileNet+DenseNet169+InceptionResNetV2	0.9550	0.9950	0.9866	0.8016	0.9866	0.9866	0.9550	0.9650	0.9583	0.9633	0.9916

Note. The highest average accuracy is denoted by the bold text.

According to Table 6, a concatenated model of deep features from 2-3 pre-trained models performs better than each individual pre-trained model. This is because each pretrained model comes with unique deep features. Therefore, combining the deep features of 2-3 distinct pre-trained models actually maximizes the chances of obtaining a wide variety of deep features that helps to yield better outcomes.

Table 7. Accuracies of best 3 concatenated deep features with best 5 ML Classifiers, and our proposed model on BTS dataset.

Metrics	SVC(Linear)	SVC(Poly)	NuSVC	AdaBoost	MLP	Proposed Model
Accuracy	0.9220	0.9220	0.9090	0.92220	0.9016	0.9434
Precision	0.92	0.92	0.9	0.92	0.9	0.94
Recall	0.92	0.92	0.905	0.92	0.905	0.94
F1 Score	0.92	0.92	0.905	0.92	0.905	0.94

Note. The highest average accuracy is denoted by the bold text.

Table 8. Accuracies of top 3 concatenated deep features with best 5 ML Classifiers, and our proposed model on BTL dataset.

Metrics	SVC (Linear)	SVC (Poly)	SVC(Sigmoid)	NuSVC	MLP	Proposed Model
Accuracy	0.995	0.99	0.99	0.99	0.99	0.9989
Precision	0.995	0.99	0.985	0.99	0.99	0.998
Recall	0.995	0.99	0.985	0.99	0.99	0.998
F1 Score	0.995	0.99	0.99	0.99	0.99	0.998

Note. The highest average accuracy is denoted by the bold text.

Tables 7 and 8 demonstrate that our proposed model is an effective method that achieves higher accuracies on both datasets. In fact, the accuracy achieved by our proposed approach is equal to or surpasses the highest accuracy achieved by the base learners alone. This algorithm is able to effectively combine the predictions of sub-models to produce an overall prediction that is superior to that of each individual sub-model.

5. Conclusions

In order to devise our proposed method for classifying brain tumors that combines the powerful features of various CNNs, we evaluated the performance with various machine learning classifiers. The extracted deep features are fed into PCA to reduce their dimensionality before being assessed by several ML classifiers. A single vector is created by concatenating the best three deep features, which performed well on various machine learning classifiers. Similarly, based on our selection criteria, we selected the top five ML classifiers to apply a single-layer stacking algorithm on fused deep features to predict the final output. In this study, we evaluated nine distinct pre-trained deep convolutional networks and ten various machine learning classifiers on two separate brain tumor datasets (BTS and BTL). Our experiment findings show that (1) DenseNet-169 standalone performs better than alternative machine learning classifiers when the data is comparatively small in size. (2) Regarding the issue of classifying brain tumors utilizing MRI scans, SVM with a linear kernel outperforms among rest of the Machine learning classifiers. (3) The limitations of a single CNN model are overcome by our suggested novel feature ensemble strategy, which yields superior and reliable performance, for both (BTS and BTL) datasets. Our proposed model has achieved optimal accuracy of 99.89, 94.34%, specificity(precision) of 99.89, 94%, sensitivity (recall) of 99.89, 94%, and F1-score 99.89, 94% on dataset BTL and BTS, respectively. These experimental findings demonstrate that our proposed strategy, which employs an ensemble of deep features and a single-layer stacking algorithm, is suitable to classify the brain tumor and aids the model to become more generic in nature. However, our proposed strategy yields satisfactory classification results on both datasets (BTL and BTS) but is computationally costly. The authors anticipated that the computational cost of the proposed model might be reduced by reducing the size of the model and generalized to a variety of clinical applications including liver lesion classification and breast tumor classification using various medical image modalities including X-rays, CT-Scan, and PET.

References

1. Verkhatsky, A.; Nedergaard, M. The homeostatic astroglia emerges from evolutionary specialization of neural cells. *Philosophical Transactions of the Royal Society B: Biological Sciences* **2016**, *371*, 20150428.
2. Muhammad, K.; Khan, S.; Del Ser, J.; De Albuquerque, V.H.C. Deep learning for multigrade brain tumor classification in smart healthcare systems: A prospective survey. *IEEE Transactions on Neural Networks and Learning Systems* **2020**, *32*, 507-522.
3. Muzammal, M.; Talat, R.; Sodhro, A.H.; Pirbhulal, S. A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks. *Information Fusion* **2020**, *53*, 155-164.
4. Pirbhulal, S.; Wu, W.; Mukhopadhyay, S.C.; Li, G. Adaptive energy optimization algorithm for internet of medical things. In Proceedings of the 2018 12th International Conference on Sensing Technology (ICST), 2018; pp. 269-272.
5. Zhang, H.; Zhang, H.; Pirbhulal, S.; Wu, W.; Albuquerque, V.H.C.D. Active balancing mechanism for imbalanced medical data in deep learning-based classification models. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **2020**, *16*, 1-15.
6. Zhang, T.; Sodhro, A.H.; Luo, Z.; Zahid, N.; Nawaz, M.W.; Pirbhulal, S.; Muzammal, M. A joint deep learning and internet of medical things driven framework for elderly patients. *IEEE Access* **2020**, *8*, 75822-75832.

7. Rehman, A.; Naz, S.; Razzak, M.I.; Akram, F.; Imran, M. A deep learning-based framework for automatic brain tumors classification using transfer learning. *Circuits, Systems, and Signal Processing* **2020**, *39*, 757-775.
8. Ahmed, N., Hussain, M., & Malik, A. . Brain tumor segmentation using a hybrid approach of wavelet transform and gray-level co-occurrence matrix. . *Journal of Medical Imaging and Health Informatics*, **2015**, *5(6)*, 1246-1253.
9. Cheng, J.; Huang, W.; Cao, S.; Yang, R.; Yang, W.; Yun, Z.; Wang, Z.; Feng, Q. Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PloS one* **2015**, *10*, e0140381.
10. Kumar, S.; Dabas, C.; Godara, S. Classification of brain MRI tumor images: a hybrid approach. *Procedia computer science* **2017**, *122*, 510-517.
11. Bahadure, N.B.; Ray, A.K.; Thethi, H.P. Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM. *International Journal of Biomedical Imaging* **2017**, *2017*, 9749108, doi:10.1155/2017/9749108.
12. Jiang, Y., Chen, C., & Zhou, Y. . Brain tumor classification using T1-weighted MRI scans based on region-based active contour model and random forest algorithm. . *Journal of Medical Imaging and Health Informatics* **2021**, *11(4)*, 797-804.
13. Yao, J., Xie, B., Ma, H., Jin, M., & Yang, X. . Brain Tumor Classification Using Fuzzy Cognitive Maps Based on MRI Data. *Journal of Healthcare Engineering* **2020**.
14. Deepa, S.; Devi, B.A. Artificial neural networks design for classification of brain tumour. In Proceedings of the 2012 International Conference on Computer Communication and Informatics, 2012; pp. 1-6.
15. Khened, M., Kori, A. V., & Sonawane, R. S. Brain tumor segmentation using deep learning. *International Journal of Advanced Research in Computer Science* **2019**, *10(5)*, 86-90.
16. Soltaninejad, M.; Yang, G.; Lambrou, T.; Allinson, N.; Jones, T.L.; Barrick, T.R.; Howe, F.A.; Ye, X. Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in FLAIR MRI. *International journal of computer assisted radiology and surgery* **2017**, *12*, 183-203.
17. Li, Y., Jiang, B., Ma, Z., & Yang, X. A hybrid approach of traditional and automated learning methods for brain MRI images classification. *Journal of medical systems* **2021**, *45(7)*, 73.
18. Rosebrock, A. Finding Extreme Points in Contours with OpenCV. Available online: <https://pyimagesearch.com/2016/04/11/finding-extreme-points-in-contours-with-opencv/> (accessed on
19. Yang, S.; Xiao, W.; Zhang, M.; Guo, S.; Zhao, J.; Shen, F. Image Data Augmentation for Deep Learning: A Survey. *arXiv preprint arXiv:2204.08610* **2022**.
20. Sundararajan, S.K.; Sankaragomathi, B.; Priya, D.S. Deep Belief CNN Feature Representation Based Content Based Image Retrieval for Medical Images. *Journal of Medical Systems* **2019**, *43*, 174, doi:10.1007/s10916-019-1305-6.
21. Thomas, C. An introduction to Convolutional Neural Networks. **2019**, 2022.
22. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 international conference on engineering and technology (ICET), 2017; pp. 1-6.
23. Indolia, S.; Goswami, A.K.; Mishra, S.P.; Asopa, P. Conceptual understanding of convolutional neural network-a deep learning approach. *Procedia computer science* **2018**, *132*, 679-688.
24. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE* **2020**, *109*, 43-76.
25. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *Journal of Big Data* **2016**, *3*, 9, doi:10.1186/s40537-016-0043-6.
26. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In Proceedings of the Artificial Neural Networks and Machine Learning – ICANN 2018, Cham, 2018//, 2018; pp. 270-279.

27. Salahat, E.; Qasaimeh, M. Recent advances in features extraction and description algorithms: A comprehensive survey. In Proceedings of the 2017 IEEE International Conference on Industrial Technology (ICIT), 22-25 March 2017, 2017; pp. 1059-1063.
28. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-first AAAI conference on artificial intelligence, 2017.
29. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp. 4700-4708.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
31. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; pp. 2818-2826.
32. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018; pp. 8697-8710.
33. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018; pp. 4510-4520.
34. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* **2017**.
35. Jolliffe Ian T. and Cadima Jorge. Principal component analysis: a review and recent developments. *The Royal Society* **13 April 2016**, Volume 374, doi:<https://doi.org/10.1098/rsta.2015.0202>.
36. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, 70, 489-501, doi:<https://doi.org/10.1016/j.neucom.2005.12.126>.
37. Zhang, J.; Ding, W. Prediction of air pollutants concentration based on an extreme learning machine: the case of Hong Kong. *International journal of environmental research and public health* **2017**, 14, 114.
38. Parvathy Jyothi , R.A.S. Classification of Tumors from Brain MR Slices using PCA for Discriminative Models. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY* **14 December 2021**, 10, doi:10.17577/IJERTV10IS120053.
39. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* **2011**, 12, 2825-2830.
40. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* **1997**, 55, 119-139.
41. Hastie, T.; Rosset, S.; Zhu, J.; Zou, H. Multi-class adaboost. *Statistics and its Interface* **2009**, 2, 349-360.
42. Alpaydin, E. Introduction to machine learning 2nd ed. **2010**.
43. Zhang, M.-L.; Zhou, Z.-H. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition* **2007**, 40, 2038-2048.
44. Breiman, L. Random forests. *Machine learning* **2001**, 45, 5-32.
45. Schintler, L.A.; McNeely, C.L. *Encyclopedia of big data*; Springer: 2019.
46. Tang, B.; Pan, Z.; Yin, K.; Khateeb, A. Recent advances of deep learning in bioinformatics and computational biology. *Frontiers in genetics* **2019**, 10, 214.
47. Cortes, C.; Vapnik, V. Support-vector networks. *Machine learning* **1995**, 20, 273-297.
48. Abdar, M.; Acharya, U.R.; Sarrafzadegan, N.; Makarenkov, V. NE-nu-SVC: a new nested ensemble clinical decision support system for effective diagnosis of coronary artery disease. *IEEE Access* **2019**, 7, 167605-167620.

49. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016; pp. 785-794.
50. Kang, S.; Cho, S.; Kang, P. Multi-class classification via heterogeneous ensemble of one-class classifiers. *Engineering Applications of Artificial Intelligence* **2015**, *43*, 35-43, doi:<https://doi.org/10.1016/j.engappai.2015.04.003>.
51. Ranjan, G.; Verma, A.K.; Radhika, S. K-nearest neighbors and grid search cv based real time fault monitoring system for industries. In Proceedings of the 2019 IEEE 5th international conference for convergence in technology (I2CT), 2019; pp. 1-5.
52. Panca, V.; Rustam, Z. Application of machine learning on brain cancer multiclass classification. In Proceedings of the AIP Conference Proceedings, 2017; p. 030133.
53. CHAKRABARTY, N. Brain MRI Images for Brain Tumor Detection. Available online: <https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection> (accessed on
54. HAMADA, A. Br35H:: Brain Tumor Detection 2020. Available online: <https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection> (accessed on