IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 10, 2022, accepted May 25, 2022, date of publication May 30, 2022, date of current version June 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3179376

On the Performance of Deep Transfer Learning Networks for Brain Tumor Detection Using MR Images

SAIF AHMAD AND PALLAB K. CHOUDHURY, (Member, IEEE)
Department of Electronics and Communication Engineering, Khulna University of Engineering & Technology (KUET), Khulna 9203, Bangladesh
Corresponding author: Pallab K. Choudhury (pallab@ece.kuet.ac.bd)

ABSTRACT A brain tumor need to be identified in its early stage, otherwise it may cause severe condition that cannot be cured once it is progressed. A precise diagnosis of brain tumor can play an important role to start the proper treatment, which eventually reduces the survival rate of patient. Recently, deep learning based classification method is popularly used for brain tumor detection from 2D Magnetic Resonance (MR) images. In this article, several transfer learning based deep learning methods are analyzed using number of traditional classifiers to detect the brain tumor. The investigation results are based on a labeled dataset with the images of both normal- and abnormal brain. For transfer learning, seven methods are used such as VGG-16, VGG-19, ResNet50, InceptionResNetV2, InceptionV3, Xception, and DenseNet201. Each of them is followed by five traditional classifiers, which are Support Vector Machine, Random Forest, Decision Tree, AdaBoost, and Gradient Boosting. All the combinations of deep learning based feature extractor and classifier are investigated to evaluate the relevant performance in terms of accuracy, precision, recall, F1-score, Cohen's kappa, AUC, Jaccard, and Specificity. Later on, learning curves for all of the combinations that achieved the highest accuracies were presented. The presented results show that the best model achieved an accuracy of 99.39% with a 10-fold cross validation. The results presented in this article are expected to be useful for the selection of suitable method in deep transfer learning based brain tumor detection.

INDEX TERMS Brain tumor, magnetic resonance imaging (MRI), transfer learning, deep learning, VGG-19, support vector machine (SVM), DCNN.

I. INTRODUCTION

A tumor is caused by an abnormal growth of cells that has no purpose. In the case of benign tumors that do not invade surrounding tissues and thus, they grow in a contained area. However, if such tumors grow near to a vital area, they can still cause troubles. On the other hand, malignant tumors grow and spread in such a way that can cause life-threatening cancerous disease. When the majority of the cells are damaged or old, they are removed or replaced with new cells. It may cause problems if the damaged or old cells are not removed. The development of a mass of tissue, which refers to the growth or tumor, is often the result of the creation of additional cells. Because of the size, shape, position, and form of tumor in the brain, the identification of brain tumor is a challenging task. In particular, early-stage brain tumor diagnosis is quite difficult due to the lack of precise information about tumor's size results from low resolution image of tumor areas. The patients can be treated in good way if the tumor is detected and treated early in the tumor formation process. As a result, tumor treatment is highly dependent on the timely diagnosis of tumor with its proper classification. To diagnose the brain tumors, there are several medical imaging technologies are used, for example, Magnetic Resonance Imaging (MRI), Computerized Tomography (CT) scan, Ultrasound, Simple Photon Emission Computed Tomography (SPECT), Positron Emission Tomography (PET), and X-ray. Among these, MRI is the most commonly used medical imaging technique as it offers better contrast images of brain tumor in compared to other medical imaging techniques. Recently, machine learning (ML) based approaches are gained much popularity to identify the brain tumor from the MR images as it gives quite accurate and precise detection results. Especially, transfer learning technique has demonstrated in several investigations,

The associate editor coordinating the review of this manuscript and approving it for publication was Yu Zhang.

where the knowledge learned from a task can be reused for another similar task to achieve improved performance in classification on target dataset [1], [2]. Conventionally, the amount of computational complexity is quite high to train a deep convolutional neural network (DCNN) model using a massive dataset. Therefore, such learning procedure can be simplified by reusing the model weights from previously trained models. The trained model's layers are then employed in a new model to be trained with new dataset of interest. As a result, the training time and generalization error is significantly reduced. However, a detail study of using different traditional deep transfer learning models followed by well-known classifiers is necessary for the selection of best performing model in target application.

In this article, combination of several transfer learning based deep learning methods with different classifiers are investigated to detect the brain tumor from MR images and finally, compared their relative performance. An effective deep transfer learning system is identified to detect and classify brain tumor with greater accuracy even in the presence of lower dataset. In particular, seven transfer learning methods are used as feature extractors such as VGG-16, VGG-19, ResNet50, InceptionResNetV2, InceptionV3, Xception, and DenseNet201. Moreover, each of the CNN model is further followed by five traditional classifiers namely SVM, Random Forest, Decision Tree, Adaptive Boosting (AdaBoost), Gradient Boosting. Pre-trained deep CNNs are used for MR brain images to extract the necessary features and further categorized using classifiers with 10-fold cross-validation. Finally, a detail comparative results are computed in the presence of different performance matrices.

The rest of the paper is organized as follows. Section II covers the recent works for brain tumor detection from MR images using CNN models. Section III presents the investigation framework including the detail description of brain image dataset and data augmentation, image pre-processing, CNN models and classifiers used for this research. The evaluation matrices and the corresponding results are discussed in section IV. Finally, the conclusion of this work is presented in section V.

II. RELATED WORKS

In [3], a hybrid technique is introduced using wavelet transform, principal component analysis, and supervised learning algorithms, where the detection accuracy of brain tumor is reached to 98.6%. However, the proposed system requires to train in each time if the image database is changed.

Beside this, a novel combination of methods such as discrete wavelet packet transform (DWPT), Shannon entropy (SE), Tsallis entropy (TE) and generalized eigenvalue proximate support vector machine (GEPSVM) are also utilized to classify the brain images [4]. In [5], tenserflow is used to implement a 5-layer convolutional neural network for MRI-based brain tumor detection. However, a limited number of training data are used for machine learning. In [6], spatial gray level dependency (SGLD) matrix is used for MRI images to extract the necessary features of brain tumor and finally applied to an ANN model for classification. The proposed method shows the accuracy 99% and sensitivity 97.9%. However, the system increases the computational complexity due to the long processing time. In [7], three multi-resolution techniques such as wavelet transform, curvelet transform and shearlet transform are used to detect the brain abnormality. By using only fifteen shearlet features, the SVM classifier with the radial basis function (RBF) kernel approach achieved a maximum classification accuracy 97.38%. In [8], a CNN model named as BrainMRNet is proposed using the combination of residual blocks, attention module, and hypercolumn technique followed by dense layer and softmax to detect brain tumor. Here, the proposed study claimed to reach the accuracy of 96.05%. In [9], Support Vector Machine along with a Fully Automatic Heterogeneous Segmentation (FAHS-SVM) process is utilized to locate the tumor areas, where the model accuracy reached to 98.51%. A modified ResNet50 model is also constructed in [10], where the 5 layers are removed from the existing structure and new 10 layers are added at the end. Even though the modified model shows the classification accuracy of 97.01%, the system complexity is increased due to the presence of additional layers. In [11], Le-Net and U-Net models are combined to develop a new model LU-Net that provides less number of layers to reduce the system complexity. A detail comparative analysis is performed by considering the Le-Net, VGG-16 and proposed LU-Net, where the new model achieved the highest accuracy of 98.00% in compare to other models. However, there is an uncertainty of system performance in the presence of large dataset. In [12], authors are used superpixels and Principal Component Analysis (PCA) for the feature extraction, which is again followed by a filter to enhance the images. Moreover, TK-means clustering is added in the model for image segmentation and brain tumor detection. However, the study is carried out using low number of image dataset. By incorporating clinical presentations and traditional MRI analysis, a deep learning based paradigm is proposed in [13]. Here the backward propagation for the gradients is used to increase the depth of the network, which eventually improves the model accuracy. However, the model suffers with long computational time as well as increase of development complexity due to the presence of additional layers. In [14], a demonstration of ensemble features and ensemble classifiers was proposed. The DenseNet-169 model achieved an average accuracy of 92.37% using small dataset, whereas the ResNeXt-101 model achieved an average accuracy of 96.13% using large dataset. However, the model size is insufficient for a real-time medical diagnostic system based on knowledge distillation techniques. Moreover, a single classifier shows better results for some cases compared to ensemble configuration with average results. A brief summary of related works using ML based brain tumor detection are presented in Table 1.

In summary, the aforementioned ML approaches are mostly used the standard CNN models for brain tumor detection. On the other hand, the pre-trained model by transfer

TABLE 1. Brief summary of related works ML based brain tumor detection.

Author	Dataset	Feature Extraction Method	Classification Method	Accuracy
El-Dahshan *et* al. [3]	70 MR images	Hybrid technique of Discrete Wavelet Transformation (DWT) and Principal Component Analysis (PCA)	Feed Forward Back Propagation Artificial Neural Network (FP-ANN)	97%
			K-Nearest Neighbor (K-NN)	98.60%
Zhang *et* al. [4]	T2 weighted 255 MR images	Discrete Wavelet Packet Transform (DWPT), Shannon entropy (SE), and Tsallis entropy (TE)	Generalized Eigenvalue Proximate Support Vector Machine (GEPSVM)	99.61%
Sawant *et* al. [5]	1800 MR images	CNN	CNN	98.60%
Abdalla *et* al. [6]	239 MR images	Spatial Gray Level Dependency (SGLD) Matrix	ANN	99%
Gudigar *et* al. [7]	T2 weighted brain MR images. Dataset: 66, Dataset: 160, and Dataset: 255	Wavelet transform, Curvelet transform. and Shearlet transform	Particle Swarm Optimization, and Support Vector Machine (PSO-SVM)	97.38%
Togacar *et* al. [8]	253 MR images	Hypercolumn, attention module, and residual block	CNN	96.05%
Jia *et* al. [9]	500 MR images	Fully Automatic Heterogeneous Segmentation (FAHS)	SVM	98.51%
Cinar *et* al. [10]	253 MR images	Improved ResNet50	CNN	97.01%
Rai *et* al. [11]	253 MR images	Combination of Le-Net and U-Net known as LU-Net	Fully connected layers and sigmoid activation function	98%
Islam *et* al. [12]	253 MR images	Superpixels and Principal Component Analysis (PCA)	Tk-means clustering	95%
Das *et* al. [13]	253 MR images	Deep-CNN	Deep-CNN	98%
Kang *et* al. [14]	253 MR images	DenseNet-169	Ensemble of multiple classifiers	92.37%
	3000 MR images	ResNeXt-101	Ensemble of multiple classifiers	96.13%
Proposed	**250 MR images**	**VGG-19**	**SVM**	**99.39%**

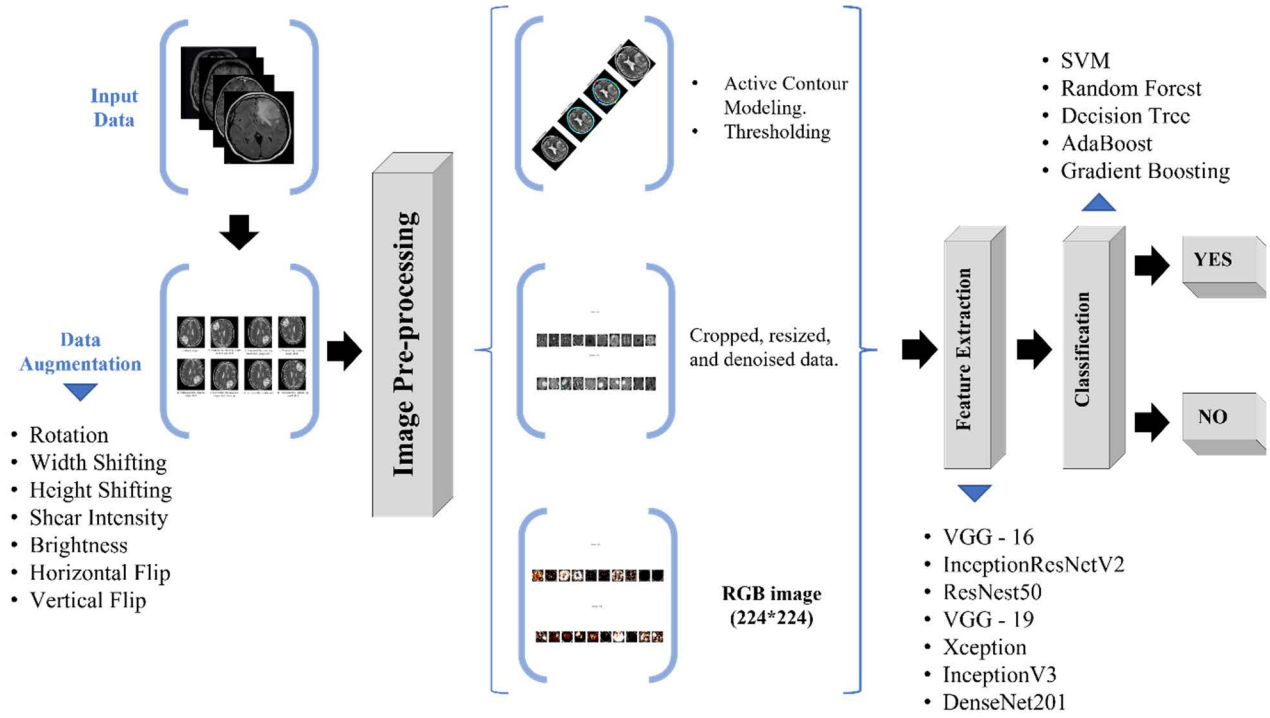


FIGURE 1. Overview of the investigation framework.

learning technique results in less computational time, higher accuracy and removes the constraint of maintaining large dataset for training. Moreover, the classification performance of traditional classifiers is better than softmax or fully connected layers used in previous investigations. Overall, the major contributions of this study can be summarized as follows:

- To provide in-depth analysis of seven pre-trained models such as VGG-16, VGG-19, ResNet50, InceptionResNetV2, InceptionV3, Xception, and DenseNet201. The transfer learning techniques are used to extract deep features from target dataset of MR brain images.
- To provide in-depth analysis of five classifiers such as SVM, Random Forest, Decision Tree, Adaptive Boosting (AdaBoost), Gradient Boosting. Different classifiers are used to classify the brain MR images into benign and malignant.
- To conduct an extensive analysis on seven pre-trained models followed by five classifiers considering all the combinations and finally, compare the effectiveness of all the CNN models and ML classifiers on the target dataset.
- To propose the best-performing model that achieved the highest accuracy and optimal computational time among all the models. Moreover, the corresponding parameter settings are also explored.
- To provide a comparison with the state-of-the-art models that justify the use of best performing model for classifying the brain tumor MR images to achieve the highest accuracy.

III. INVESTIGATION FRAMEWORK

The investigation framework used for this study is presented in Fig. 1. The process is started with MR brain image dataset, which is further used for data augmentation. The dataset splits in three ways namely train set, test set, and validation set. Later on, the MR brain images are further processed to reduce the noise and ready for feature extraction. In feature extraction part, several CNN models are tested such as VGG-16, VGG-19, ResNet50, InceptionResNetV2, InceptionV3, Xception, and DenseNet201. The pre-processed images were fed into the transfer learning models with a batch size of 32. Finally, classification stages are prepared using different classifiers like SVM, Random Forest, Decision Tree, AdaBoost, and Gradient Boosting. Based on these feature extractors and classifiers, the relative performance of detecting the brain tumor is evaluated to select the best performing machine learning model using brain MR images.

A. BRAIN IMAGE DATASET AND DATA AUGMENTATION

In this investigation, a publicly accessible MRI dataset from Kaggle [https://www.kaggle.com/navoneel/brain-mri images-for-brain-tumor-detection] is used to analyze and evaluate the developed framework. The images are in two folders labeled as 'yes' and 'no' corresponding to the abnormal- and normal brain images as shown in Fig.2. Originally, it contains 152 abnormal brain images and 98 normal brain images, thus a total of 250 images of varying dimensions. The images are grayscale in JPG format. Later on, augmentation technique is applied to increase the size of the dataset.

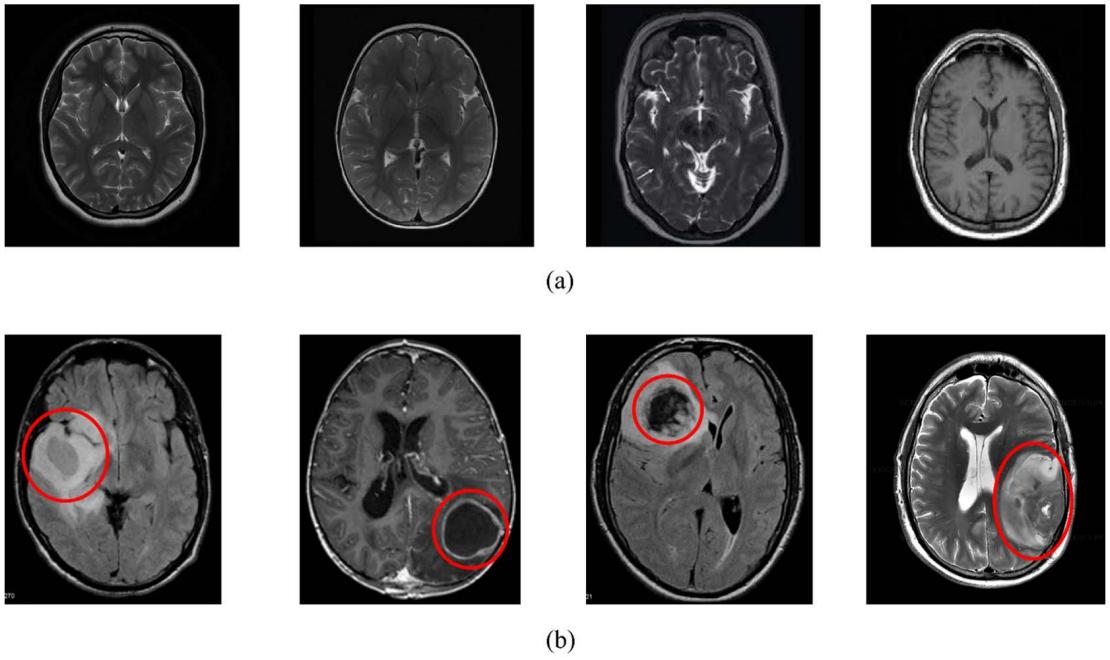


FIGURE 2. Brain MR images a) Without tumor b) With tumor.

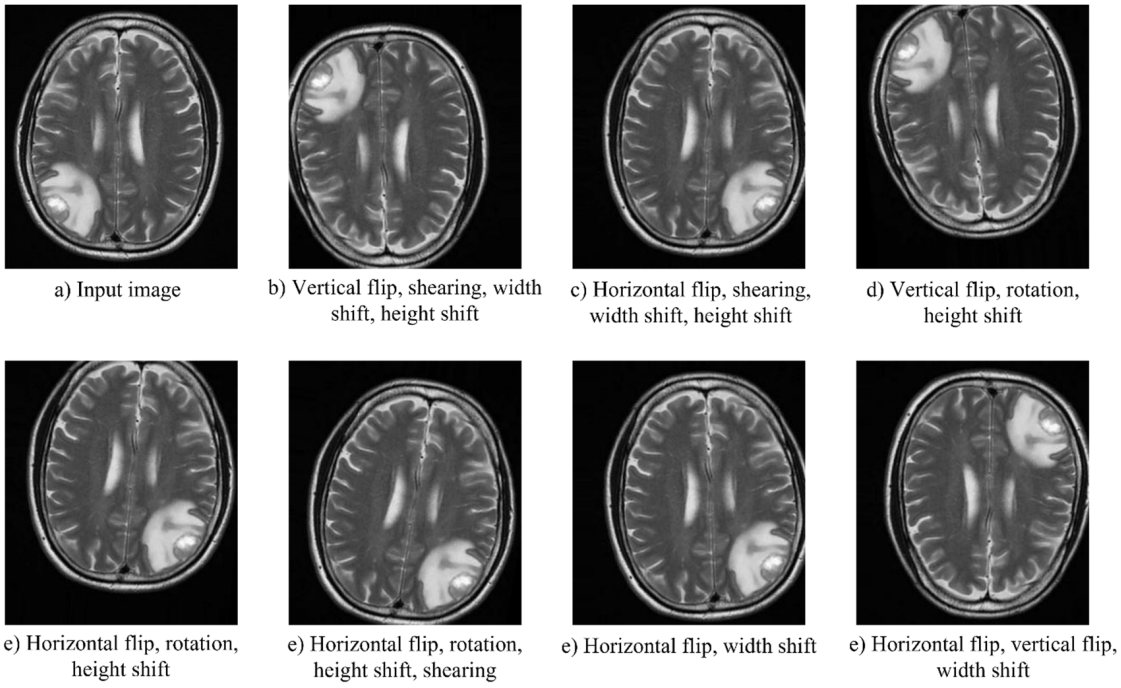


FIGURE 3. Overview of data augmentation process.

Data augmentation is a process of adding slightly changed copies of current data or newly created synthetic data from existing data to expand the size of present dataset. By generating new and varied samples of dataset, data augmentation process can help to improve the performance of machine learning models. When a machine learning model's dataset is large and diverse, the model performs better to get more accurate results. Several methods can be used for augmentation, however, the present article used the process like width shifting, height shifting, shear intensity, brightness, horizontal flip, and vertical flip for dataset size improvement as shown in Fig.3. After applying the augmentation process, the dataset is converted to 1240 abnormal- and 1078 normal- brain images. Using this dataset, the 5 images of each category are used as

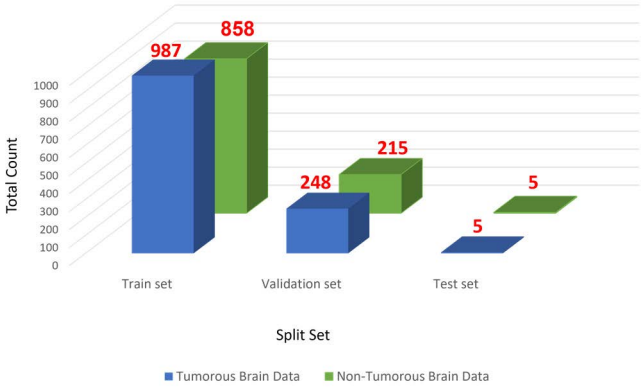


FIGURE 4. Data distribution based on train, validation and test set.

test set and the remaining data is divided into; 80% as train set and 20% as validation set. Based on this distribution, the train set has 987 abnormal- and 858 normal brain images; the test set has 5 abnormal- and 5 normal brain images; the validation set has 248 abnormal- and 215 normal brain images. Fig. 4 shows the dataset distribution using a bar graph.

B. IMAGE PRE-PROCESSING

In machine learning, the used dataset is typically not organized as it comes from different sources. Therefore, the dataset needs to be standardized and processed before being fed to the ML model. Moreover, MR images may contain defects such as inhomogeneity distortions and motion heterogeneity due to the person's body motion during image acquisition or instability of the scanning hardware. These distortions eventually add unwanted intensity rates in the acquired images to develop false positives. Image pre-processing is commonly used to reduce these unwanted noises by collecting the useful information from the images and hence, such process improves the classification performance.

In this research, the image pre-processing stage comprises with number of steps as shown in Fig. 5. Firstly, the original grayscale MR images in varying sizes are loaded for pre-processing. In step 2, the active contour-based segmentation technique in used to select the region of interest area by defining the biggest contour. A contour is a set of points that are interpolated together using different interpolation methods like linear, splines, or polynomial to describe the curve in an image [15]. In step 3, the extreme points are selected by thresholding technique. Thresholding is a basic non-contextual segmentation technique that converts a greyscale or color image into a binary image to create a binary area map with one threshold [16]. The binary map has two potentially disjoint domains, one containing pixels with input data values less than a threshold and the other containing pixels values equal or greater than the threshold. In step 4 and 5, the images are cropped to collect the useful portion and resized 224 $\times$ 224 pixels with RGB format to fit for the input layer dimension of the feature extractors. Moreover, the small patches of the unnecessary noises are also removed by applying the erosion and dilations operations.

C. FEATURE EXTRACTION USING DCNN

Deep learning technique has proved an essential tool in various applications due to its feature learning ability and thus, highlighted its potential in many research articles including a review work published in nature [17]. In particular, convolutional neural network, a popular part of deep learning family, has attracted by many researchers just after the published results at ILSVRC-2012 (ImageNet Large Scale Visual Recognition Challenge) image classification competition using AlexNet model [18]. Even though such deep CNN shows good performance in the presence of large labeled dataset like ImageNet, the model has limited in application for medical imaging like MR images classification due to the availability of small sample size. Especially for small dataset applications, a well investigated and good alternative approach to train the deep CNN using a pre-trained model with transfer learning. The pre-trained models are proven to be easier and faster to build with improved accuracy for the target application [2]. In recent years, various CNN architectures using transfer learning have outperformed classical machine learning models. They have also shown considerable success to improve the image classification performance. In image classification, extracting the key features of the images is an important part of the process and thus, the models are properly trained to distinguish multiple levels of visual representation thanks to the concept of deep learning. Conventionally, there are two ways to use the pre-trained models, firstly, the off-the-shelf pre-trained models are used for image dataset to extract the features and train a separate classifier to classify those features. Secondly, the pre-trained models are fine tuned in selected or all the layers to get the desire results [19]. Here, the first approach is adopted with the combination of number of pre-trained models and traditional classifiers. In this article, seven pre-trained CNN models are utilized for the feature extraction using MR brain image dataset. The pre-trained CNN models are trained on large ImageNet dataset [20]. The pre-trained CNN models used in this study are VGG-16 [21], InceptionResNetV2 [22], ResNet50 [23], VGG-19 [21], Xception [24], InceptionV3 [25], and DenseNet201 [26]. A summary of these models are presented in Table 2 and more details are available in the mentioned references. The performance results of each model are presented in the later section to show the relative efficiency for the detection of brain tumor from MR images.

D. CLASSIFIERS

Classifiers are used to divide a batch of data into categories. It is a method to map the input data in a certain category using an algorithm. In this study, the extracted features from deep CNN models are classified using five classifiers namely Support Vector Machine [27], [28], Random Forest [29], [30], Decision Tree [31], AdaBoost [32], and

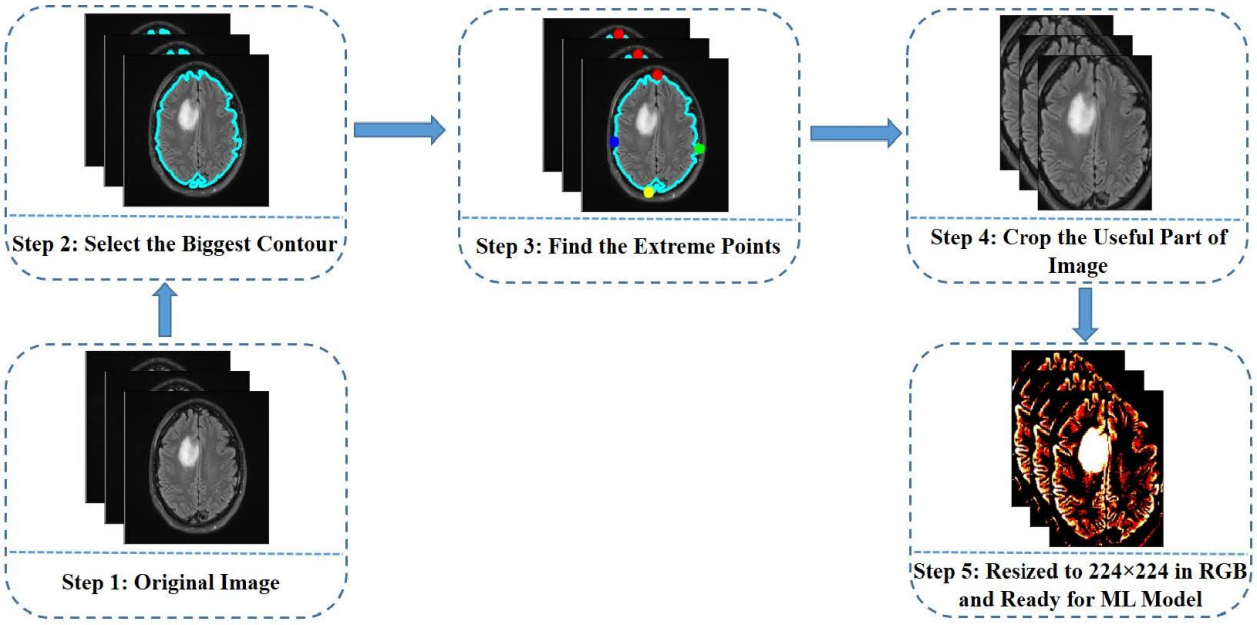


FIGURE 5. Process of image pre-processing.

Gradient Boosting [33]. The brief details of these classifiers are presented in Table 3. The performance results of each CNN model followed by classifiers will be discussed in later section.

IV. RESULTS ANALYSIS AND DISCUSSION

This section mainly highlights the performance analysis of several transfer learning based CNN models for the features extraction from brain MR images. The extracted features are further classified using number of classifiers. All the combination of feature extractors and classifiers are evaluated in terms of computational time and accuracy with 10-fold cross validation as shown in Table 4. Moreover, the presented deep learning frameworks are also tested using different evaluation matrices like accuracy, precision, recall, F1-score, Cohen's kappa, AUC (Area Under ROC (Receiver Operating Characteristic) Curve), Jaccard, and Specificity scores as shown in Table 5. Based on the evaluation results, the best performing model is identified for effective classification of brain tumor into Benign and Malignant using brain MR images. The main parameter settings of best pre-trained model and different classifiers are also highlighted in Table 6 and Table 7 respectively. Moreover, the best performing model is compared with the state-of-the-art methods as shown in Table 8.

A. EVALUATION MATRICES

The efficiency of the proposed deep transfer learning framework is measured using four key outcomes: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The following performance matrices are used to evaluate the proposed ML framework:

The accuracy can be considered as a capacity to successful detection of brain tumor from the target image dataset. The fraction of true positive and true negative in all the cases under investigation are used to estimate the accuracy as follows [34]:

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

Precision is a true positive measure, which is calculated as [34]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall (Sensitivity) is a metric that evaluates the system's capacity to accurate classification of brain tumors, and it is determined by the percent of true positives as [34]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The F1-score takes the harmonic mean of a classifier's precision and recall to create a single statistic. The F1-score is given by [34]:

$$\text{F1-score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{\text{TP}}{\text{TP} + {}^{1}/_{2}(\text{FP} + \text{FN})}$$

Cohen's Kappa is a statistical measure that determines how often two raters agree on the same quantity and is measured as [34]:

$$\text{K} = \frac{p_o - p_e}{1 - p_e}$$

where,

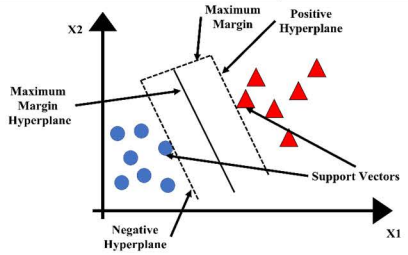
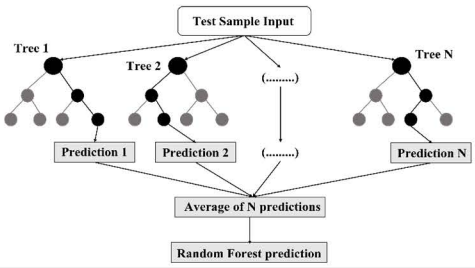
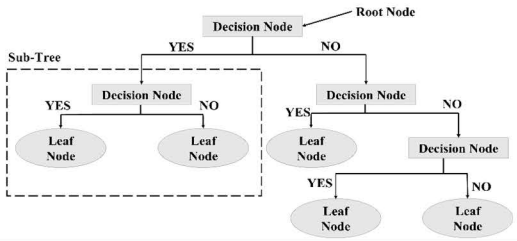
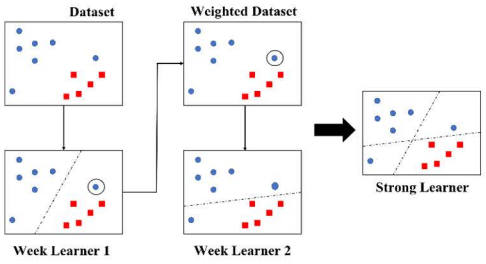
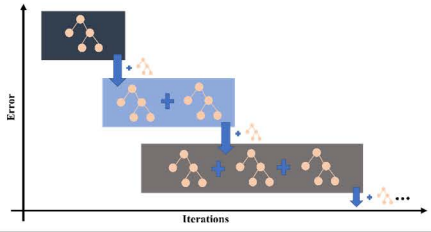
p_o = Overall accuracy of the model

p_e = Metric for the degree of agreement between model predictions and actual class values.

TABLE 2. Brief description of different pre-trained models used in this research.

Pre-trained models	Key Features
VGG16 [21]	• 16 layers. • Input image size is 224×224 in RGB format. • 13 convolution layers with 3×3 filter size. • 5 max-pool layers with 2×2 pool size. • 3 fully connected layers.
Inception-ResNet-V2 [22]	• 164 layers. • 1 stem block, 5 InceptionResNet-A block, 1 Reduction-A block, 10 InceptionResNest-B block, 1 Reduction-B block, 5 InceptionResNet-C block. • Option to incorporate various size of convolutional layers.
ResNet50 [23]	• 50 layers. • 1 ConvB1 block, 3 ConvB2 blocks, 4 ConvB3 blocks, 6 ConvB4 blocks, and 3 ConvB5 blocks, with average-pooling and fully connected softmax layers as output layers. • ConvB1 has one convolution layer and one max-pooling layer, whereas the remaining blocks have three convolution layers with 1×1 kernel, 3×3 kernel, and 1×1 kernel. • The number of filters used in each convolution layer varies.
VGG19 [21]	• 19 layers. • Input image size is 224×224 in RGB format. • 16 convolution layers with 3×3 kernel size with stride 1. • 5 maxpool layers with 2×2 pool size with stride 2. • 3 fully connected layers and softmax function as the final layer.
Xception [24]	• 71 layers. • The network's feature extraction basis is made up of 36 convolutional layers. • Except for the first and last modules, the convolutional layers are divided into 14 modules, all containing linear residual connections around them. • Depth wise separable convolution layer stack with residual connections.
InceptionV3 [25]	• 48 layers. • It's an advance version of InceptionV2 and suggested in the same publication. • RMSProp optimizer, factorized 7x7 convolutions, BatchNorm in the Auxillary classifiers, and label smoothing are all included in InceptionV3.
DenseNet201 [26]	• 201 layers. • DenseNet take advantage of feature reuse. • All layers use the same feature maps. • If there are L layers in a dense block then there will be $\frac{L(L+1)}{2}$ direct connections. • Reducing parameters and feature sharing as well as reducing the vanishing gradient problem, DenseNet201 provides better results than other model.

TABLE 3. Summary of classifiers used in this research.

Classifier	Basic Diagram	Basic Principle
Support Vector Machine (SVM) [27-28]		• A supervised learning method that uses an *N*-dimensional hyperplane. • The hyperplane categorizes the data into two classes in efficient way possible. • Support vectors are considered to construct the hyperplane.
Random Forest [29-30]		• An ensemble learning approach for classification and regression • It works by training a large number of decision trees. • Rather than relying on a single decision tree, it takes the forecasts from each tree and calculates the final performance based on the majority votes of predictions.
Decision Tree [31]		• A tree-format supervised learning approach. • Decision nodes and leaf nodes. • Decision is using simply yes/no inquiry to the decision nodes. • The decisions are made depending on the input data's qualities.
AdaBoost [32]		• An ensemble classifier using supervised learning technique to construct and gather sets of base learners. • The weak classifier is a learner that is boosted to develop a strong classifier. • Except first, each subsequent learner is an improved version of the previous learners. • Error estimated from weighted data points
Gradient Boosting [33]		• Uses the gradients in the loss function to determine the error. • The prediction starts with a simple decision tree and continue to reach the final improved forecast with minimum error.

The ROC curve is a binary classification task evaluation metric. It is a probability curve that compares true positive rate (TPR) to false positive rate (FPR) at various threshold levels, effectively separating the signal from the noise. The AUC is a summary of the ROC curve that measures a classifier's ability to distinguish between classes and is given by [35]:

$$\int_{-\infty}^{\infty} TPR\,(T)\,FPR'\,(T)\,dT$$

Here,

FPR' (T) = First derivative of FPR with respect to T.

T = The sample data

Jaccard similarity coefficient is measured to address the similarities between sample sets. The mathematical formula is [36]:

$$\mathrm{J(A,\,B)} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

The fraction of real negatives that were projected as negatives, also known as true negatives, is defined as specificity. In other

TABLE 4. Ten-fold cross validation results for accuracy and computational time.

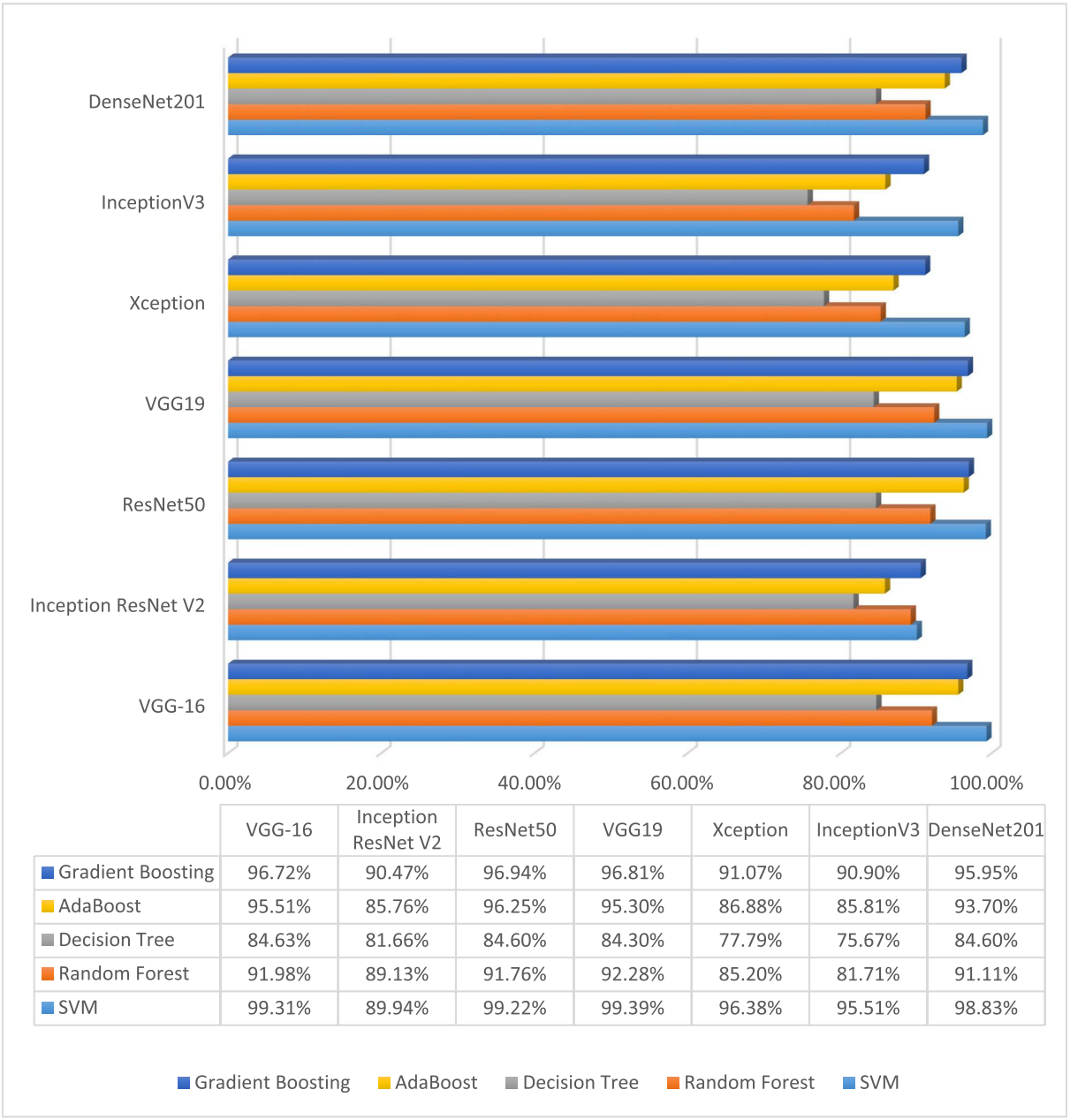
Feature Extractor	No. of Features	Classifiers	Time (Seconds)	Accuracy (10-fold Cross Validation)
VGG-16	25088	SVM	858.315	**0.9931±0.008**
		Random Forest	**7.939**	0.9198±0.032
		Decision Tree	153.730	0.8463±0.054
		AdaBoost	1408.495	0.9551±0.024
		Gradient Boosting	3533.297	0.9672±0.016
Inception-ResNet-V2	38400	SVM	1259.227	0.8994±0.043
		Random Forest	**21.569**	0.8913±0.046
		Decision Tree	731.277	0.8166±0.058
		AdaBoost	6482.768	0.8576±0.043
		Gradient Boosting	18213.972	**0.9047±0.037**
ResNet50	100352	SVM	4006.161	**0.9922±0.012**
		Random Forest	**22.666**	0.9176±0.039
		Decision Tree	843.608	0.8460±0.039
		AdaBoost	7866.894	0.9625±0.013
		Gradient Boosting	19052.306	0.9694±0.029
VGG-19	25088	SVM	814.773	**0.9939±0.006**
		Random Forest	**7.691**	0.9228±0.035
		Decision Tree	163.697	0.8430±0.054
		AdaBoost	1521.014	0.9530±0.031
		Gradient Boosting	4637.814	0.9681±0.022
Xception	100352	SVM	5189.341	**0.9638±0.024**
		Random Forest	**15.810**	0.8520±0.020
		Decision Tree	384.721	0.7779±0.055
		AdaBoost	5167.590	0.8688±0.035
		Gradient Boosting	10159.634	0.9107±0.042
InceptionV3	51200	SVM	2523.732	**0.9551±0.040**
		Random Forest	**19.976**	0.8171±0.049
		Decision Tree	647.890	0.7567±0.074
		AdaBoost	6738.413	0.8581±0.024
		Gradient Boosting	17534.069	0.9090±0.043
DenseNet201	94080	SVM	3763.9662	**0.9883±0.017**
		Random Forest	**27.418**	0.9111±0.045
		Decision Tree	1001.061	0.8460±0.040
		AdaBoost	11817.573	0.9370±0.023
		Gradient Boosting	25049.207	0.9595±0.032

words, specificity is addressed as True Negative Rate (TNR). The mathematical formula is [37]:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

B. COMPARATIVE ANALYSIS

This section highlights the performance of seven pre-trained CNN models i.e., VGG16, InceptionResNetV2, ResNest50, VGG19, Xception, InceptionV3 and DenseNet201 and



	VGG-16	Inception ResNet V2	ResNet50	VGG19	Xception	InceptionV3	DenseNet201
Gradient Boosting	96.72%	90.47%	96.94%	96.81%	91.07%	90.90%	95.95%
AdaBoost	95.51%	85.76%	96.25%	95.30%	86.88%	85.81%	93.70%
Decision Tree	84.63%	81.66%	84.60%	84.30%	77.79%	75.67%	84.60%
Random Forest	91.98%	89.13%	91.76%	92.28%	85.20%	81.71%	91.11%
SVM	99.31%	89.94%	99.22%	99.39%	96.38%	95.51%	98.83%

FIGURE 6. **Graphical representation of accuracy results for different pre-trained models with classifiers.**

further followed by five classifiers such as Support Vector Machine, Random Forest, Decision Tree, AdaBoost, and Gradient Boosting. The relative performance of each feature extractor and classifier pair is tested to identify the best performing model. In this investigation, the transfer learning models are used as standalone feature extractors and later on, the traditional classifiers are used to classify those features to detect the tumor from brain images. As a standalone feature extractor, the pre-trained network is used to process the images, extract features and the fully connected layers (classification layers) are kept inactive.

Conventionally, the CNN networks are used up to the last pooling layer and 'include_top' argument is defined as 'False' to unload the fully connected layers (classification layers). After the last pooling layer, here an additional flatten layer is added and the networks are incorporated with different traditional classifiers. The flatten layer works as a dimensionality reduction function as it reduces the number of parameters. It also converts the feature map that pooled from the last pooling layer to a single dimensional array and forwards the output to the classifiers in the next step. All the pair of extractor-classifier are analyzed using the performance parameters of accuracy and time in 10-fold cross validation. Cross-validation is used to estimate the skill of a model based on unseen data. In 10-folds cross-validation method, the dataset is shuffled randomly and split into 10 groups of

TABLE 5. Ten-folds classification results for precision, recall, F1-score, Cohen's kappa and AUC.

Feature Extractor	Classifiers	Precision	Recall	F1-score	Cohen's Kappa	ROC AUC	Jaccard	Specificity
VGG-16	SVM	**0.9951**	0.9903	0.9927	0.9844	0.9924	0.9832	0.9944
	Random Forest	0.9493	0.9056	0.9270	0.8471	0.9250	0.8689	0.9443
	Decision Tree	0.8678	0.8573	0.8625	0.7063	0.8535	0.7725	0.8497
	AdaBoost	0.9475	0.9613	0.9545	0.9010	0.9500	0.9266	0.9388
	Gradient Boosting	0.9710	0.9710	0.9710	0.9376	0.9688	0.9518	0.9666
Inception-ResNet-V2	SVM	0.9186	0.8919	0.9051	0.7993	0.9005	0.8286	0.9091
	Random Forest	0.9291	0.8661	0.8965	0.7861	0.8950	0.8116	0.9218
	Decision Tree	0.8257	0.8024	0.8139	0.6063	0.8038	0.6728	0.8052
	AdaBoost	0.8784	0.8565	0.8673	0.7187	0.8600	0.7787	0.8655
	Gradient Boosting	0.9267	0.8871	0.9064	0.8038	0.9032	0.8285	0.9193
ResNet50	SVM	**0.9951**	0.9903	0.9927	0.9844	0.9924	0.9815	0.9944
	Random Forest	0.9448	0.9105	0.9273	0.8470	0.9246	0.8479	0.9388
	Decision Tree	0.8581	0.8532	0.8556	0.6906	0.8454	0.7377	0.8377
	AdaBoost	0.9679	0.9476	0.9576	0.9099	0.9557	0.8999	0.9462
	Gradient Boosting	0.9804	0.9685	0.9744	0.9454	0.9731	0.9487	0.9777
VGG-19	SVM	**0.9951**	**0.9927**	**0.9939**	**0.9870**	**0.9936**	**0.9863**	**0.9963**
	Random Forest	0.9381	0.9040	0.9207	0.83311	0.9177	0.8456	0.9314
	Decision Tree	0.8669	0.8565	0.8617	0.7046	0.8526	0.7502	0.8553
	AdaBoost	0.9573	0.9573	0.9573	0.9081	0.9540	0.9162	0.9508
	Gradient Boosting	0.9699	0.9637	0.9668	0.9289	0.9647	0.9434	0.9703
Xception	SVM	0.9724	0.9661	0.9693	0.9341	0.9673	0.9373	0.9685
	Random Forest	0.8757	0.8298	0.8522	0.6917	0.8472	0.7368	0.8646
	Decision Tree	0.7980	0.7871	0.7925	0.5574	0.7790	0.6372	0.7709
	AdaBoost	0.8953	0.8823	0.8887	0.7627	0.8818	0.7939	0.8813
	Gradient Boosting	0.9089	0.9258	0.9173	0.8203	0.9096	0.8509	0.8933
InceptionV3	SVM	0.9567	0.9444	0.9505	0.8943	0.9576	0.9092	0.9508
	Random Forest	0.8687	0.7734	0.8183	0.6337	0.8194	0.6894	0.8655
	Decision Tree	0.7974	0.7839	0.7906	0.5540	0.7774	0.6519	0.7709
	AdaBoost	0.8803	0.8718	0.8760	0.7348	0.8677	0.7867	0.8636
	Gradient Boosting	0.9185	0.9000	0.9092	0.8069	0.9041	0.8321	0.9082
DenseNet201	SVM	0.9888	**0.9927**	0.9907	0.9801	0.9899	0.9785	0.9879
	Random Forest	0.9361	0.8863	0.9105	0.8135	0.9084	0.8342	0.9304
	Decision Tree	0.8786	0.8524	0.8653	0.7154	0.8585	0.7692	0.8720
	AdaBoost	0.9505	0.9452	0.9478	0.8882	0.9443	0.8964	0.9416
	Gradient Boosting	0.9651	0.9581	0.9616	0.9177	0.9591	0.9215	0.9573

equal sizes. At first, it takes data from one group for the validation test and the data from other nine groups use for training. The system evaluates the validation test based on the training set and stores the result. The process continues 10 times (a total of 10 observations) and each time, it takes data from a different group for validation test and the data from other nine groups as training set. The final result is the average of all the 10 processes.

According to the Table 4, the number of features that extracted by VGG16, InceptionResNetV2, ResNest50,

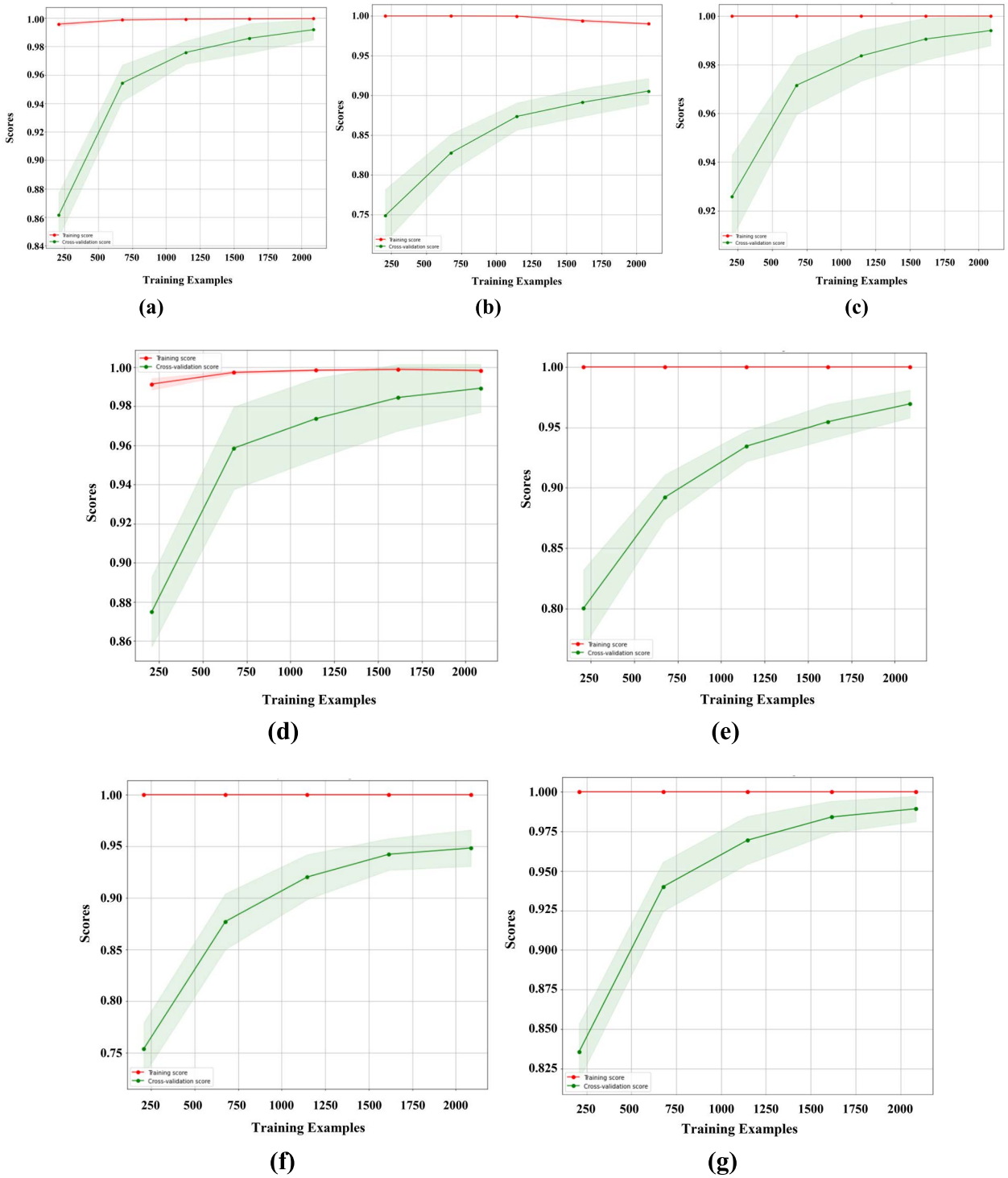


FIGURE 7. Learning curves (a) VGG16 - SVM pair (b) InceptionResNetV2 - Gradient Boosting pair (c) ResNet50 - SVM pair (d) VGG19 - SVM pair (e) Xception - SVM pair (f) InceptionV3 - SVM pair (g) DenseNet201 - SVM pair.

VGG19, Xception, InceptionV3 and DenseNet201 are 25088, 38400, 100352, 25088, 100352, 51200 and 94080 respectively. The features are extracted from the final pooling layer. Moreover, the features are used by five classifiers to classify the MR brain images into benign and malignant. The highest accuracy values are marked in Bold considering all the combinations of pre-trained models and classifiers as shown in Table 4. Table 4 clearly illustrates that SVM classifier shows the best accuracy results compared to other classifiers while working with the ML models of VGG-16, ResNet50, VGG-19, Xception, InceptionV3 and DenseNet201. For Inception-ResNet-V2, Gradient Boosting classifier shows the improved performance in terms of accuracy to classify the MR images. In particular, the accuracy results of

TABLE 6. Hyper-parameter settings of VGG-19.

Stage	Hyper Parameter	Value
Initialization	Bias	Zero
	Weights	Glorot_uniform
Training	Batch size	256
	Momentum	0.9
	Weight decay	5×10^{-4}
	Dropout ratio	0.5
	Learning rate	0.01
	Epochs	74
	Total iterations	370K

SVM classifier are 99.31%, 99.22%, 99.39, 96.38%, 95.51%, and 98.83% using VGG-16, ResNet50, VGG-19, Xception, InceptionV3 and DenseNet201 respectively. On the other hand, Inception-ResNet-V2-Gradient Boosting model shows the accuracy of 90.47%. Based on the accuracy performance as mentioned above, the VGG-19-SVM model achieved the highest accuracy i.e., 99.39% among all the investigated models for this study. There are another two models that demonstrate the accuracy above 99% are VGG-16-SVM and ResNet50-SVM with the value of 99.31% and 99.22% respectively. Beside this, InceptionV3-Decision Tree shows the lowest accuracy score of 75.67 %. Fig. 6 shows the summary of accuracy results in graphically for all the combinations of ML models followed by classifiers.

Beside accuracy, the computational time of each of the feature extractor-classifier pair is also estimated as shown in Table 4, where the lowest values are marked in Bold. The presented results clearly indicate that the Random Forest classifier performs the classification operation faster than the other classifiers maintaining the lowest value of 7.691 seconds while working with the VGG-19 model. Even though the Random Forest classifier performs better in classification time, however, it shows the accuracy of around 90% that indicates the performance degradation to accurately classify the brain MR images into benign and malignant. Overall, the performance of different pair of feature extractors and classifiers shows a tradeoff between computational time and accuracy.

The presented deep learning frameworks are also tested using different evaluation matrices as formulated in section IV-A and the corresponding results are appeared in Table 5. All the combination of deep learning based feature extractors with different classifiers are analyzed using precision, recall, F1-score, Cohen's kappa, AUC, Jaccard and Specificity. The performance matrices with highest values are marked in Bold as appeared in Table 5. For precision, VGG-16-SVM, ResNet50-SVM, and VGG-19-SVM are achieved the highest value of 99.51%; For recall, VGG-19-SVM and DenseNet201-SVM are achieved the best value of 99.27%; For F1-score, Cohen's kappa, AUC, Jaccard, and Specificity the highest values are measured for VGG-19-SVM model as 99.39%, 98.70%, 99.36%, 98.63% and 99.63% respectively. Additionally, the learning curves for the pairs that achieved the highest accuracies compared to the other pairs are provided in Fig. 7. In summary, VGG-19-SVM

TABLE 7. Main parameter set of classifiers.

Classifiers	Parameter	Value
SVM	kernel	linear
	C	1
	class_weight	balanced
Random Forest	n_estimators	10
	max_depth	None
	min_samples_split	2
	random_state	0
	class_weight	balanced
Decision Tree	max_depth	None
	min_samples_split	2
	min_samples_leaf	1
	random_state	None
	class_weight	balanced
AdaBoost	base_estimator	None
	n_estimators	100
	learning_rate	1
	random_state	None
Gradient Boosting	loss	deviance
	learning_rate	0.1
	n_estimators	100
	max_depth	3
	min_samples_split	2
	min_samples_leaf	1
	random_state	None

model is considered to be the best performing deep learning system with respect to all the measured values of performance matrices as mentioned in Table 4 to Table 5. Based on this evaluation, the best performing CNN model is shown in Fig. 8. Moreover, Table 6 shows the hyper-parameter settings of best performing model VGG-19. The main parameter settings of all the classifiers are also shown in Table 7.

Finally, Table 8 shows a comparison of best performing model as presented here with state-of-the-art architectures proposed in [8]–[13]. Toğaçar *et* al. [8] used a combination of residual blocks, attention module, and hypercolumn technique with the claimed accuracy of 96.05%. Jia *et* al. [9] utilized the FAHS-SVM technique, where the mentioned accuracy of 98.51%. Besides this, Çinar *et* al. [10] achieved 97.01% accuracy with the improved ResNet50 model. Moreover, Rai *et* al. [11] combined Le-Net and U-Net to form LU-Net model that achieved an accuracy of 98.00%. Islam *et* al. [12] utilized superpixels and Principal Component Analysis (PCA) followed by TK-means clustering that achieved an accuracy of 95.00%. The study of Deep-CNN by Das *et* al. [13] achieved an accuracy of 98.00%. By comparing with all the aforementioned results, the presented model VGG-19-SVM shows the highest

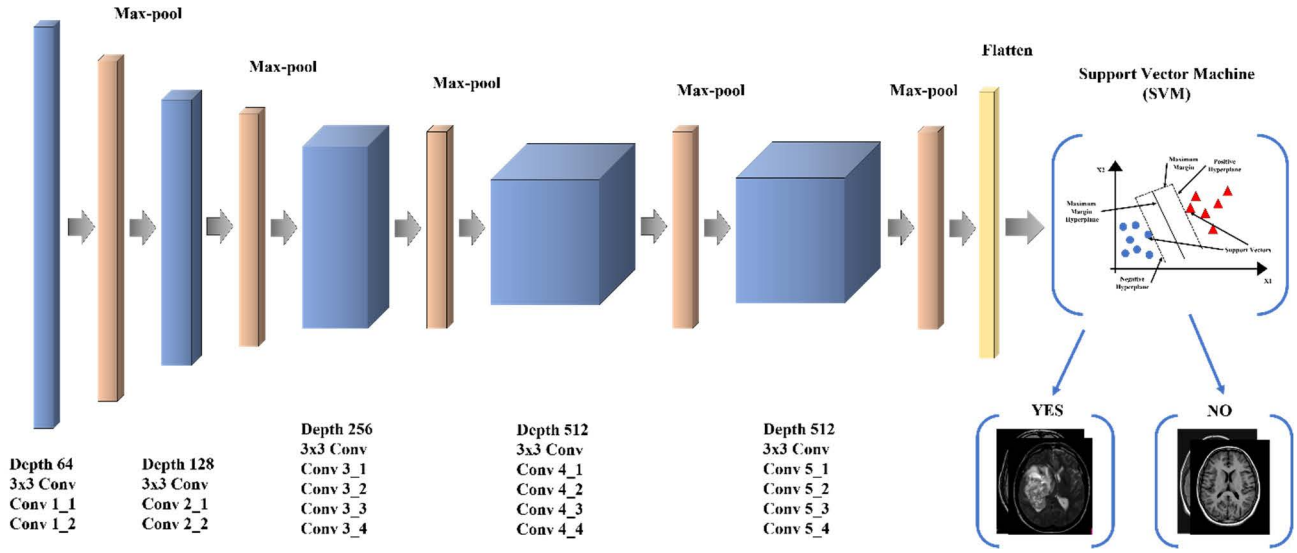


FIGURE 8. Architecture of best performing CNN model using VGG-19-SVM.

TABLE 8. Results comparison with state-of-the-art methods.

Author(s)	Method	Classifier	Accuracy
Toğaçar *et al.* [8]	Hypercolumn & Attention Module & Residual Block	Dense Layer & Softmax	96.05%
Jia et *al.* [9]	Fully Automatic Heterogeneous Segmentation	SVM	98.51%
Çinar *et al.* [10]	Improved ResNet50	Softmax + Claasification layers	97.01%
Rai *et al.* [11]	LU-Net	Fully connected layers + Sigmoid activation function	98.00%
Islam *et al.* [12]	Superpixels + PCA	TK-means clustering	95.00%
Das *et al.* [13]	Deep-CNN	Fully connected layers	98.00%
Proposed	**Deep Transfer Learning with VGG-19**	**SVM**	**99.39%**

classification accuracy of 99.39% and thus, it is expected to show good performance for detecting the brain tumor from MR images.

V. CONCLUSION

In this article, several transfer learning based deep learning methods are analyzed and corresponding results are compared to select a best performing CNN model for the detection of brain tumor from MR images. There are seven classical feature extractors are used to develop the deep learning framework, where the extracted features from each of the pre-trained model are classified using five traditional classifiers. The performance matrices such as Accuracy, computational time, Precision, Recall, F1-score, Cohen's kappa, AUC, Jaccard, and Specificity are computed for all the combination of feature extractors and classifiers with 10-folds cross validation. The best performing model i.e., VGG-19-SVM shows the highest accuracy of 99.39% among all the presented models in this investigation. Moreover, VGG-19-SVM model also performs better in compared with recent works of brain tumor detection using ML model. However, the presented model was not tested for different brain MRI modalities along with other imaging techniques. Also the proposed technique can also be extended for the classification of tumor types like Glioma, Meningioma, Pituitary using the MR image dataset. Above all, the use of larger dataset and better GPU based processing can also improve the accuracy results as well as computational speed of presented models. We aim to highlight those issues as a part of the future works.

REFERENCES

[1] P. K. Chahal, S. Pandey, and S. Goel, "A survey on brain tumor detection techniques for MR images," *Multimedia Tools Appl.*, vol. 79, nos. 29–30, pp. 21771–21814, May 2020.

[2] K. Muhammad, S. Khan, J. D. Ser, and V. H. C. D. Albuquerque, "Deep learning for multigrade brain tumor classification in smart healthcare systems: A prospective survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 507–522, Feb. 2021.

[3] E. S. A. El-Dahshan, T. Hosny, and A. B. M. Salem, "Hybrid intelligent techniques for MRI brain images classification," *Digit. Signal Process.*, vol. 20, pp. 433–441, Mar. 2010.

[4] Z. Dong, G. Ji, J. Yang, Y. Zhang, and S. Wang, "Preclinical diagnosis of magnetic resonance (MR) brain images via discrete wavelet packet transform with Tsallis entropy and generalized eigenvalue proximal support vector machine (GEPSVM)," *Entropy*, vol. 17, no. 4, pp. 1795–1813, Mar. 2015.

[5] A. Sawant, M. Bhandari, R. Yadav, R. Yele, and M. S. Bendale, "Brain cancer detection from MRI: A machine learning approach (tensor flow)," *Int. Res. J. Eng. Technol.*, vol. 5, no. 4, pp. 2089–2094, Apr. 2018.

[6] H. E. M. Abdalla and M. Y. Esmail, "Brain tumor detection by using artificial neural network," in *Proc. Int. Conf. Comput., Control, Electr., Electron. Eng. (ICCCEEE)*, Khartoum, Sudan, Aug. 2018, pp. 1–6.

[7] A. Gudigar, U. Raghavendra, T. R. San, E. J. Ciaccio, and U. R. Acharya, "Application of multiresolution analysis for automated detection of brain abnormality using MR images: A comparative study," *Future Gener. Comput. Syst.*, vol. 90, pp. 359–367, Jan. 2019.

[8] M. Toğaçar, B. Ergen, and Z. Cömert, "BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model," *Med. Hypotheses*, vol. 134, Jan. 2020, Art. no. 109531.

[9] Z. Jia and D. Chen, "Brain tumor identification and classification of MRI images using deep learning techniques," *IEEE Access*, early access, Aug. 13, 2020, doi: 10.1109/ACCESS.2020.3016319.

[10] A. Çinar and M. Yildirim, "Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture," *Med. Hypotheses*, vol. 139, Jun. 2020, Art. no. 109684.
[11] H. M. Rai and K. Chatterjee, "Detection of brain abnormality by a novel Lu-Net deep neural CNN model from MR images," *Mach. Learn. Appl.*, vol. 2, Dec. 2020, Art. no. 100004.
[12] M. K. Islam, M. S. Ali, M. S. Miah, M. M. Rahman, M. S. Alam, and M. A. Hossain, "Brain tumor detection in MR image using superpixels, principal component analysis and template based K-means clustering algorithm," *Mach. Learn. Appl.*, vol. 5, Sep. 2021, Art. no. 100044.
[13] T. K. Das, P. K. Roy, M. Uddin, K. Srinivasan, C.-Y. Chang, and S. Syed-Abdul, "Early tumor diagnosis in brain MR images via deep convolutional neural network model," *Comput., Mater. Continua*, vol. 68, no. 2, pp. 2413–2429, 2021.
[14] K. Jaeyong, U. Zahid, and G. Jeonghwan, "MRI-based brain tumor classification using ensemble of deep features and machine learning classifiers," *Sensors*, vol. 21, no. 6, p. 2222, Mar. 2021.
[15] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.
[16] O. Tarkhaneh and H. Shen, "An adaptive differential evolution algorithm to optimal multi-level thresholding for MRI brain image segmentation," *Expert Syst. Appl.*, vol. 138, Dec. 2019, Art. no. 112820.
[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, Nov. 2015.
[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.* Stateline, NV, USA: Harveys Lake Tahoe, 2012, pp. 1097–1105.
[19] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jul. 2020.
[20] D. Garcia-Gasulla, F. Parés, A. Vilalta, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés, and T. Suzumura, "On the behavior of convolutional nets for feature extraction," *J. Artif. Intell. Res.*, vol. 61, pp. 563–592, Mar. 2018.
[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*.
[22] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 4278–4284.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[24] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1800–1807.
[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.
[26] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.
[27] S. R. Telrandhe, A. Pimpalkar, and A. Kendhe, "Detection of brain tumor from MRI images by using segmentation & SVM," in *Proc. World Conf. Futuristic Trends Res. Innov. Social Welfare (Startup Conclave)*, Coimbatore, India, 2016, pp. 1–6.
[28] S. Abe, *Support Vector Machines for Pattern Classification*. London, U.K.: Springer, 2010, pp. 163–226.
[29] M. Murty and R. Raghava, *Support Vector Machines and Perceptrons: Learning, Optimization, Classification, and Application to Social Networks*, 1st ed. India: Springer, 2016.
[30] C. J. Mantas, J. G. Castellano, S. Moral-García, and J. Abellán, "A comparison of random forest based algorithms: Random credal random forest versus oblique random forest," *Soft Comput.*, vol. 23, no. 21, pp. 10739–10754, Nov. 2019.
[31] A. V. Shichkin, A. G. Buevich, and A. P. Sergeev, "Comparison of artificial neural network, random forest and random perceptron forest for forecasting the spatial impurity distribution," *AIP Conf. Proc.*, vol. 1982, no. 1, 2018, Art. no. 020005.
[32] K. Grąbczewski, *Meta-Learning in Decision Tree Induction*. Cham, Switzerland: Springer, 2014.
[33] R. Sonavane and P. Sonar, "Classification and segmentation of brain tumor using AdaBoost classifier," in *Proc. Int. Conf. Global Trends Signal Process., Inf. Comput. Commun. (ICGTSPICC)*, Dec. 2016, pp. 396–403.
[34] M. A. B. Siddique, S. Sakib, M. M. R. Khan, A. K. Tanzeem, M. Chowdhury, and N. Yasmin, "Deep convolutional neural networks model-based brain tumor detection in brain MRI images," in *Proc. 4th Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, Palladam, India, Oct. 2020, pp. 909–914.
[35] C. Calì and M. Longobardi, "Some mathematical properties of the ROC curve and their applications," *Ricerche di Matematica*, vol. 64, no. 2, pp. 391–402, Oct. 2015.
[36] K. Usman and K. Rajpoot, "Brain tumor classification from multimodality MRI using wavelets and machine learning," *Pattern Anal. Appl.*, vol. 20, no. 3, pp. 871–881, Aug. 2017.
[37] Z. N. K. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, S. Ahmed, and J. Lu, "Brain tumor classification for MR images using transfer learning and fine-tuning," *Comput. Med. Imag. Graph.*, vol. 75, pp. 34–46, Jul. 2019.

SAIF AHMAD received the B.Sc. degree in electronics and communication engineering from the Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh, in 2022. He is currently doing an Internship at the BRAC Learning Division, Learning and Leadership Development Unit. His research interests include medical image analysis, deep learning, machine learning, transfer learning, and pattern recognition.

PALLAB K. CHOUDHURY (Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh, in 2003, the M.S. degree in information and communication technologies from the Asian Institute of Technology, Thailand, in 2007, and the Ph.D. degree in optical communication engineering from the Integrated Research Center for Photonics Networks and Technologies (IRC-PhoNET), Scuola Superiore Sant'anna, Pisa, Italy, in 2012. From 2012 to 2013, he was a Research Assistant with the IRCPhoNET, Research Group of Optical System Design. During 2018–2019, he worked as a Postdoctoral Fellow with the LiDAR and Intelligent Optical Node (LION) Research Laboratory, jointly formed by the Chongqing University of Technology (CQUT), Chongqing, China, and the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. He is currently a Professor with the Department of Electronics and Communication Engineering, KUET. He is the author of more than 50 international journals and conferences. He also holds one U.S. patent. His research interests include the design and development of LiDAR for autonomous vehicle, machine learning, medical image analysis, and optical wireless transmission systems. He is an Associate Editor of IEEE Access.