Computers in Biology and Medicine 146 (2022) 105539

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

# A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models

Nahid Ferdous Aurna [a], Mohammad Abu Yousuf [b], Kazi Abu Taher [a], A.K.M. Azad [c,d], Mohammad Ali Moni [e,*]

[a] Department of Information and Communication Technology, Bangladesh University of Professionals, Bangladesh
[b] Institute of Information Technology, Jahangirnagar University, Bangladesh
[c] Faculty of Science, Engineering & Technology, Swinburne University of Technology Sydney, Australia
[d] ProCan®, Children's Medical Research Institute, Faculty of Medicine and Health, The University of Sydney, Westmead, NSW, Australia
[e] Artificial Intelligence & Digital Health, School of Health and Rehabilitation Sciences, Faculty of Health and Behavioural Sciences, The University of Queensland St Lucia, QLD 4072, Australia

ARTICLE INFO

*Keywords:*
Brain tumor classification
Convolutional neural network
Two stage ensemble
Magnetic resonance imaging
Principal component analysis

ABSTRACT

The brain tumor is one of the deadliest cancerous diseases and its severity has turned it to the leading cause of cancer related mortality. The treatment procedure of the brain tumor depends on the type, location and size of the tumor. Relying solely on human inspection for precise categorization can lead to inevitably dangerous situation. This manual diagnosis process can be improved and accelerated through an automated Computer Aided Diagnosis (CADx) system. In this article, a novel approach using two-stage feature ensemble of deep Convolutional Neural Networks (CNN) is proposed for precise and automatic classification of brain tumors. Three unique Magnetic Resonance Imaging (MRI) datasets and a dataset merging all the unique datasets are considered. The datasets contain three types of brain tumor (meningioma, glioma, pituitary) and normal brain images. From five pre-trained models and a proposed CNN model, the best models are chosen and concatenated in two stages for feature extraction. The best classifier is also chosen among five different classifiers based on accuracy. From the extracted features, most substantial features are selected using Principal Component Analysis (PCA) and fed into the classifier. The robustness of the proposed two stage ensemble model is analyzed using several performance metrics and three different experiments. Through the prominent performance, the proposed model is able to outperform other existing models attaining an average accuracy of 99.13% by optimization of the developed algorithms. Here, the individual accuracy for Dataset 1, Dataset 2, Dataset 3, and Merged Dataset is 99.67%, 98.16%, 99.76%, and 98.96% respectively. Finally a User Interface (UI) is created using the proposed model for real time validation.

## 1. Introduction

Brain is the significant part of our central nervous system that controls all our functionalities through a huge number of connected neurons [1]. Any malfunction or abnormality in the brain cells affects the organs connected to the corresponding part of the brain, consequently damaging the functionalities of that organ. Cancer originating in the brain and other nervous system is considered to be the 10th leading cause of death. The 5-year survival rate of the patients having cancerous brain is only 36% [2]. As brain tumor is caused by the unnatural and uncontrolled growth of brain cells, its severe consequences can be life-threatening. Around 400,000 people are affected by brain tumor and 120,000 people have died in the past years all over the world, as reported by World Health organization (WHO) [3]. Early and proper detection can play an indispensable role in increasing the survival rate by accelerating the treatment process [4]. Manual detection of brain tumors can be tedious, time consuming and erroneous due to the variations on types and sizes. Proper and precise detection needs expertise and it's even harder for complicated cases. Hence, besides human inspection, we can't avoid the necessity of an automated process for precise detection and classification of brain tumors. Deep learning and convolutional neural network can significantly accelerate the whole

* Corresponding author.
*E-mail addresses:* aurna31@gmail.com (N.F. Aurna), yousuf@juniv.edu (M.A. Yousuf), kataher@bup.edu.bd (K.A. Taher), aazad@swin.edu.au (A.K.M. Azad), m.moni@uq.edu.au (M.A. Moni).

https://doi.org/10.1016/j.compbiomed.2022.105539
Received 6 January 2022; Received in revised form 18 March 2022; Accepted 15 April 2022
Available online 22 April 2022

diagnosis process making the classification task automated and conscientious.

The emerging technologies of machine learning and deep learning have profoundly developed different fields of applications [5–7]. Particularly, a huge scope has been created in medical image processing, and numerous researches are ongoing for enhancing this research area. Automating the process of brain tumor segmentation and classification is a significant part of this research field.

Recently, deep learning technology has become a very popular choice in brain tumor classification from brain MRI images. For example, Gumaei et al. [1] introduced a hybrid method for feature extraction called PCA-NGIST. Then, from the extracted features, brain images are classified into three categories (meningioma, glioma, pituitary) using Regularized Extreme Learning Machine classifier and achieved an accuracy of 94.233%.They worked with only one dataset and didn't included any tumor-free or normal brain images. Tandel et al. [8] proposed a deep learning model based on CNN for classification of brain tumor from five individual multi-class datasets. The highest achieved accuracy was 96.65% but they didn't added any analysis on model's robustness.

Sajjad et al. [9] proposed an architecture, where they used Input-CascadeCNN for tumor segmentation and fine-tuned VGG-19 for three types of tumor classification, where the achieved accuracy was 94.58% for brain tumor dataset. Alqudah et al. [10] proposed a deep learning technique for classification of three types of tumor, where the carried out experiments on cropped, uncropped and segmented images. They achieved the average accuracy of 98.93%, 99% and 97.62% for cropped, uncropped and segmented images, respectively. Their dataset doesn't include any normal brain images and more analysis could have been done on the proposed architecture. Deepak et al. [11] used transfer learning approach, where fine-tuned GoogleNet was used for classification of three types of brain tumor. The overall accuracy was 98%. In Refs. [10,11], a particular dataset has been considered using which it is very much possible to achieve a good accuracy by optimizing a model but this certainly does not guarantee the model's robustness for other random data. Moreover, in Ref. [11] there was a considerable misclassification of "meningioma" class and had an overfitting tendency.

Pashaei et al. [12] used combination of CNN and Extreme Learning Machine for classifying tumors into three classes achieving an accuracy of 93.68% but Some other recent approaches attained better accuracy. Irmak et al. [13] proposed three different CNN models for three datasets containing different types of brain tumors. The models individually attained accuracy of 99.33%, 92.66%, and 98.14% for the three datasets, respectively. The accuracy is very high for binary classification but in multi-classification they couldn't achieve that much accuracy.

Balasooriya et al. [14] proposed a less complex CNN model for classifying five types of brain tumors, which resulted in an accuracy of 99.69%. They didn't provide much analysis on the performance of their model and didn't include any comparative study with other models. Das et al. [15] presented a comparatively shallow CNN based model for the classification of three types of brain tumors, which could attain an accuracy of only 94.39%. For increasing generalization capability this could be implemented on other datasets including normal brain images. Afshar et al. [16] proposed Capsule Network architecture for three tumor types classification namely glioma, meningioma and pituitary, where they got an accuracy of 90.89%.Compared to other existing models, their attained accuracy is not much satisfactory.

Hemanth et al. [17] introduced a modified deep CNN architecture to address the computational complexity of deep CNN model, and classified brain tumors into four classes. In this way, they achieved an average accuracy of 96.4%. Any comparative analysis with state-of-the-art models have not been showed in this paper. Badža et al. [18] presented a new CNN architecture for classifying three types of brain tumors. After 10-fold cross validation, they achieved an accuracy of 96.56%. They used only one dataset but generalization capability could have been guaranteed using multiple diverse datasets. Ayadi et al. [19] suggested a new CNN model for multi-classification of brain tumors and attained average accuracy of 94.74%. They did a detail analysis and discussion on their classification model but the attained accuracy was comparatively lower. Sultan et al. [20] proposed a deep CNN based model for classifying tumors into three labels, and also different grades of glioma were differentiated. For two studies, they achieved best accuracy of 96.13% and 98.7%, respectively. Though the model achieved a quite good accuracy but using large scale and diverse dataset, this could have attained more generalization capability. Deepak et al. [21] presented a CNN-SVM based classification model for three types of brain tumors and attained 95.82% accuracy. Better result could have been gained using data augmentation techniques.

Besides multi-classification, deep learning approach is also used for binary classification of tumors. Kumar et al. [22] introduced an optimized deep learning technique for classifying tumor and non-tumor cells and attained accuracy of 95.3% and 96.3%, for two different datasets, respectively. The extra preprocessing, segmentation and feature extraction processed made this model a bit complex. More optimization could be done to avoid these overheads. Toğaçar et al. [23] introduced a new deep learning model called BrainMRNet to differentiate between normal and tumorous brain images, achieving 96.05% accuracy. Their used dataset contain only 253 images which is not enough to train and build a robust model. Hossain et al. [24] proposed a CNN based architecture for classifying tumor and non-tumor images from brain MRI that achieved 97.87% accuracy. They used only 217 images which doesn't ensure the generalization capability of their proposed model. Siar et al. [25] presented a model, where CNN feature extractor and softmax classifier were used for classifying tumorous and normal brain images attaining 98.67% accuracy. They didn't include much analysis and comparison with other methods.

Srinivas et al. [26] proposed a hybrid method based on CNN and K-Nearest Neighbor to classify benign and malignant tumors from MRI of brain, where 96.25% accuracy was achieved. The dataset was very small for a CNN model to train which included only 400 images for training. Besides, no analysis on model's robustness and comparison with other models were shown. Kader et al. [27] introduced differential deep CNN model to classify normal and abnormal (tumorous) images from brain MRI and attained an accuracy of 99.25% but their model was only limited in binary classification.

Along with advanced deep learning techniques, several researches have also been done using different classical machine learning approaches for automatic classification of brain tumors. Rajagopal et al. [28] proposed a method, where the derived features of brain MRI are optimized using the *ant colony optimization* technique, and then classified into Glioma or non-Glioma images using the random forest classifier. This approach achieved 98.01% accuracy. Their work was confined in detecting only glioma and non-glioma tumor detection. Further analysis could be done by detecting other types of tumors and also comparing with other state-of-the-art methods. Arasi et al. [29] proposed a method for classifying benign and malignant tumor from brain MRI, where they could attain 97.69% accuracy. The whole process includes tumor segmentation using fuzzy clustering algorithm, feature extraction using GLCM, and finally classification was conducted using the Boosting Support Vector Machine. They didn't provide any comparison or analysis mentioning the performance of other classifiers on their dataset.

Jayaprada et al. [30] proposed a fast classifier, which is based on hybrid binary Adaboost algorithm to classify normal and tumorous images of brain MRI which resulted in 90.4% accuracy. Using this approach they needed to do a lot of preprocessing work which could be an overhead. Besides, the dataset contained only 253 images which is insufficient to build a robust model. Padlia et al. [31] proposed a fractional Sobel filter and SVM based binary classification of normal and tumorous images and the best accuracy was 99.19%. They focused on the preprocessing part but further analysis could be done in the classification approach using other classifiers.

Feature level ensemble or fusion of different deep learning and machine learning models are also used for increasing model's robustness. Iqbal et al. [32] proposed a model based on the fusion of Long Short Term Memory (LSTM) model and CNN model for segmentation of tumor area which achieved 82.29% accuracy. Further research could be done by classifying those segmented tumors using the same approach. Khan et al. [33] proposed an architecture for multi-modal brain tumor classification, where robust features from VGG-16 and VGG-19 were extracted and fused before they were fed into Extreme Learning machine classifier. The highest accuracy of this approach was achieved as 97.8%. In this work, the researcher only focused on classifying modality of the images but they didn't dealt with tumor type classification. Noreen et al. [34] introduced a concatenation approach using Inception-V3 and DenseNet201. Features from different inception modules and dense blocks are extracted and further concatenated for classification of three types of brain tumor. In this way, DensNet201 performed better by achieving an accuracy of 99.51%. Sachdeva et al. [35] presented a dual level ensemble neural network for multi-classification of brain tumors. This ensemble approach achieved better performance compared to single neural network. Kang et al. [36] proposed an approach of ensemble deep features extracted from different pre-trained models. The best models are chosen based on different classifiers and the extracted features are further fed into the classifiers for final classification. This approach significantly improved the classification performance but no implementation was done to use the model in real-time environment. Amin et al. [37] proposed a score level fusion approach for binary classification of brain tumor. They used AlexNet and GoogleNet for extracting feature vector and the individual classification score are fused before final classification. In this way, they achieved the highest accuracy of 99.44%. This approach can be experimented with different tumor types besides only binary classification. Amin et al. [38] proposed a system, where tumor region is enhanced and segmented, and then the features are extracted through Local Binary Pattern and Gabor Wavelet Transform. These features are further fused for a better classification of tumor and non-tumor images. Based on the fused features, KNN performed better than other classifiers. Shankar K et al. [39] presented a process of binary classification (benign, malignant) of brain tumors, where features are extracted based on Gray Level Co-occurrence Matrix and Maximum Intensity. Those features are fused and classified using Adaptic Neuro Fuzzy Interface System (ANFIS) classifier and the obtained accuracy was 96.23%. This approach can be experimented for multi-classification of tumors. Kaur et al. [40] proposed a voting ensemble technique for detecting benign and malignant brain tumors from brain MRI. They used three classifiers, i.e., Support Vector Machine, K-Nearest Neighbor and Decision Tree for classification, and then the final outcome was calculated using majority voting. Their acquired accuracy was 97.91%. Though they achieved a good accuracy using a very small datasets, further researches could be done for diverse and large dataset.

There are some works based on non-iterative approaches like General Regression Neural Network (GRNN). For example, Izonin et al. [41]. Presented a GRNN based prediction model for small medical dataset. They prepared and applied the input doubling method using small datasets in medical application. Though this has been used for urine analysis, image classification can also be done using this approach. Sinha et al. [42] used a non-iterative convolutional feature learning based approach for brain tumor classification. For faster and accurate classification they used convolutional feature based Euclidean and achieved 97.02% accuracy. However, they have considered a specific dataset and this work should be extended by applying it on more diverse datasets.

Recent important existing works regarding brain tumor detection are summarized in Table 1 considering deep learning, classical ML and ensemble approach. This literature survey suggested that there is a great scope of experimenting with multi stage ensemble technique. One of the common limitations found from the analysis of the existing models is the

**Table 1**
Recent works on brain tumor classification based on deep learning, classical ML and ensemble approach.

| Method used | Ref. | Main contribution | Limitation |
|---|---|---|---|
| Deep learning | Tandel et al. [8] | Proposed a CNN model for classification of brain tumor from five individual multi-class datasets. The highest achieved accuracy was 96.65% | Lack of analysis on model's robustness |
| | Alqudah et al. [10] | Proposed CNN model for classification of three types of tumor and achieved the average accuracy of 98.93%, 99% and 97.62% for cropped, uncropped and segmented images | Dataset didn't include any normal brain images and a particular dataset was considered |
| | Deepak et al. [11] | Applied transfer learning approach, where fine-tuned GoogleNet was used for classification of three types of brain tumor and overall accuracy was 98%. | Considerable misclassification of "meningioma" class and had an overfitting tendency |
| | *Irmak* et al. [13] | Proposed three different CNN models for three datasets and attained accuracy of 99.33%, 92.66%, and 98.14% for the three datasets, respectively | Accuracy was very high for binary classification but in multi-classification it was very low |
| | Toğaçar et al. [23] | Introduced a deep learning model called Brain MRNet to differentiate between normal and tumorous brain images, achieving 96.05% accuracy | Dataset contained only 253 images which is not enough to train and build a robust model |
| Classical ML | Rajagopal et al. [28] | Used ant colony optimization and Random Forest classifier to classify glioma and non-glioma tumors achieving 98.01% accuracy | The study was limited only in glioma tumors, other tumor types would have been considered |
| | Arasi et al. [29] | Classified benign and malignant tumors using fuzzy clustering and boosting Support Vector Machine, attaining an accuracy of 97.69% | Didn't provide any comparison or analysis mentioning the performance of other classifiers on their dataset |
| | Jayaprada et al. [30] | Proposed a classifier based on hybrid binary Adaboost algorithm to classify normal and tumorous images that resulted in 90.4% accuracy | They needed to do a lot of preprocessing work which could be an overhead. Besides, the dataset contained only 253 images |
| Fusion or ensemble approach | Iqbal et al. [32] | Proposed a fusion of Long Short Term Memory (LSTM) model and CNN model for segmentation of tumor area which achieved 82.29% accuracy | The work was limited to tumor segmentation only |
| | Khan et al. [33] | Proposed an architecture for multi-modal brain tumor classification, where features from VGG 16 | Focused on classifying modality of the images but they didn't dealt with tumor type classification. |

(*continued on next page*)

**Table 1** (*continued*)

| Method used | Ref. | Main contribution | Limitation |
|---|---|---|---|
| | | and VGG-19 were fused, attaining the highest accuracy of 97.8% | |
| | Kaur et al. [40] | Proposed a voting ensemble technique for detecting benign and malignant brain tumors from brain MRI using SVM, KNN and Decsion Tree, attaining an accuracy of 97.91% | Used a very small dataset and the study was limited to binary classification |

lack of generalization capability and robustness. To the best of our knowledge, there is lack of studies in conducting a multi or two-stage ensemble approach for such classification of brain tumors from brain MRI, though researchers in Ref. [43] have used this approach for localization of hippocampus. This motivated us to devise a new two-stage ensemble approach of deep CNN models for classification of brain tumors. Besides, a detailed analysis of model selection, building and discussion on model validation is added in our study. The main contributions of the article are outlined below-

- Best CNN feature extractors and classifier are selected through several trials and experiments on three different datasets. Initially total 6 features extractors were considered including 5 pre-trained models and one proposed CNN model.

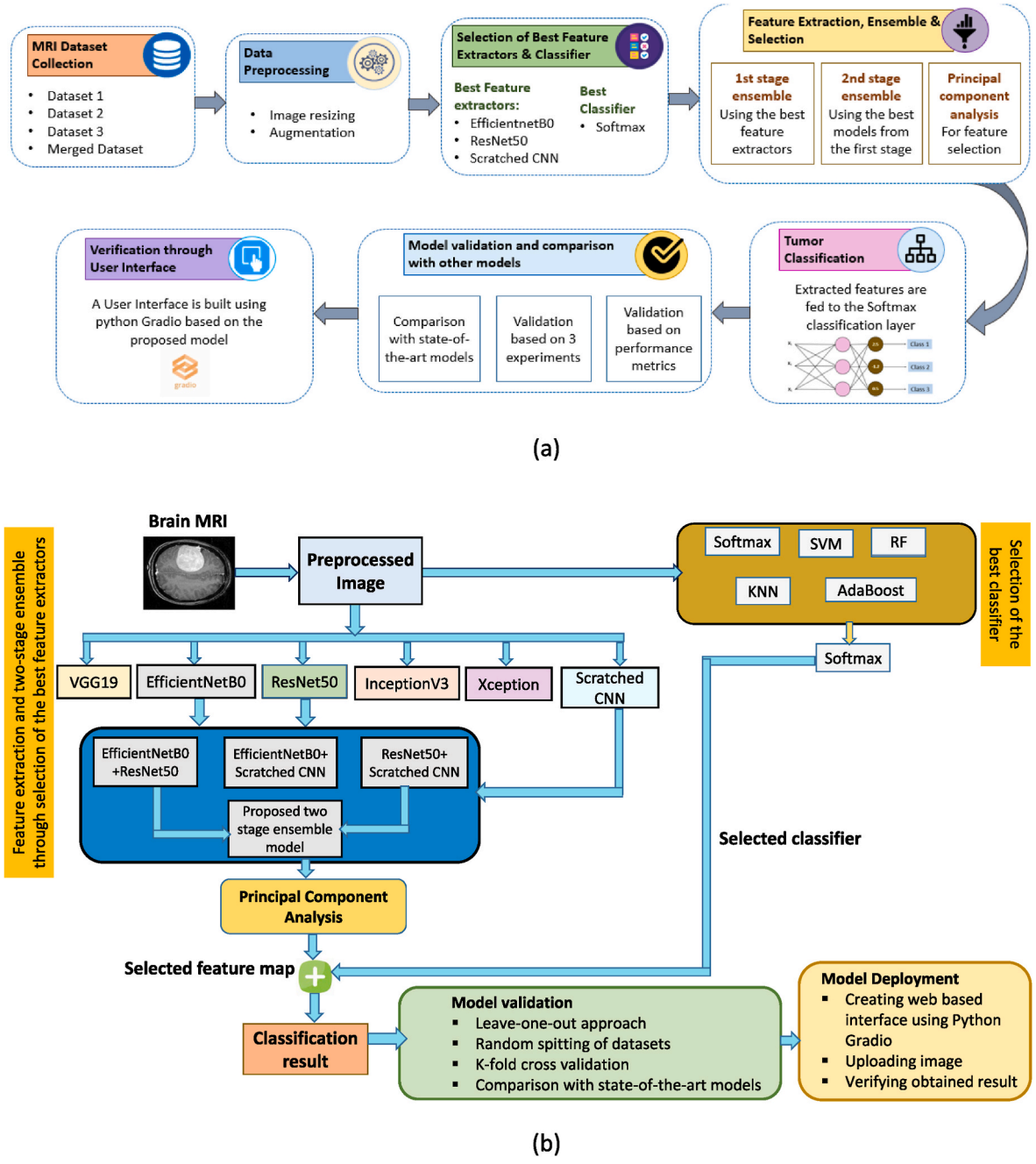


**Fig. 1.** (a) Workflow of the proposed method showing different associative steps including preprocessing, feature extraction, ensemble, classification and model validation. (b) Schematic diagram of the proposed approach.

- A new two stage ensemble model is built by scrutinizing the models twice, i.e., first time when building one-stage ensemble model, second time when building the final two-stage ensemble model.
- A real-time verification is proved through a brand-new User Interface built with the proposed model.

The remaining sections of the paper is structured as follows: A detailed description of all the materials and methodology including feature extraction, selection and classification process is presented in Section 2. The result and discussion including all other validation, verification procedures are discussed in Section 3. Finally, the conclusion is added in Section 4.

## 2. Materials and methodology

Fig. 1(a) and (b) depict the summary of the proposed methodology and the schematic diagram of the proposed approach respectively. The input MRI data are first preprocessed and then fed into the feature extractor. The feature extraction process is done in two stages. For the first stage, the best individual models are selected and several ensemble models are built. In the second stage further ensemble is done with best models found on first ensemble. The final ensemble model is used for feature extraction and from the extracted features most significant ones are selected via principal component analysis. Then the reduced feature map is fed to the classification layer. Finally, the proposed model is validated considering multiple aspects and experiments.

### 2.1. Datasets

In this article, three unique datasets have been used (Brain Tumor Dataset 1, Brain Tumor Dataset 2, Brain Tumor Dataset 3) along with a dataset that is created by merging all of them. These three datasets and the merged dataset will be addressed as Dataset 1, Dataset 2, Dataset 3 and *Merged Dataset*, respectively, throughout this paper. Fig. 2 shows the distributions of all the datasets.

#### 2.1.1. Dataset-1

Dataset 1 contains total 3064 T1-weighted contrast-enhanced images (where fat tissue is highlighted) from 233 patients with three kinds of brain tumor: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices). It includes axial, coronal and sagittal views of all the tumors. This dataset was first used in paper [44] and publicly available in Ref. [45].

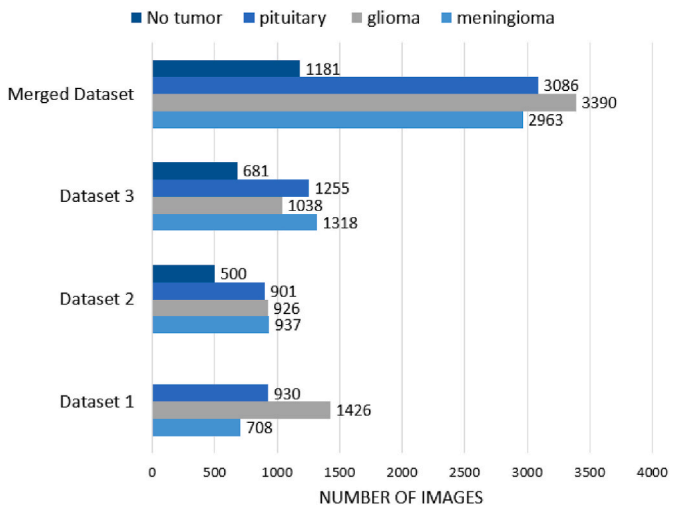


**Fig. 2.** Data distribution of Dataset 1, Dataset 2, Dataset 3 and the Merged Dataset that contains all the data of Dataset 1, 2 and 3.

#### 2.1.2. Dataset-2

This dataset contains total 3264 MRI images of three types of brain tumor: meningioma (937 slices), glioma (926 slices), pituitary (901 slices) and normal brain tissue (500 slices). This images are the combination of T1 (type of MRI where fat tissue is highlighted and seems brighter), T2 (type of MRI where fat tissue and water are highlighted and seem brighter) and Flair types (same as T2 with free flowing water and fat seem dark), which is available in Ref. [46].

#### 2.1.3. Dataset-3

This dataset contains total 4292 MRI images of three types of brain tumor: meningioma (1318 slices), glioma (1038 slices), pituitary (1255 slices) and normal brain tissue (681 slices). This is also publicly available and can be found in Ref. [47].

#### 2.1.4. Merged dataset

This dataset is created by merging all the unique datasets (Dataset 1, Dataset 2, and Dataset-3). So in this dataset, there are total 10,620 brain MRI images which include meningioma (2963 slices), glioma (3390 slices), pituitary (3086 slices) and normal brain tissue (1181 slices).

### 2.2. Data preprocessing

Minimum preprocessing is done on the datasets - image resizing and augmentation. All the images are resized to 256 × 256 pixels. Then six types of augmentation techniques (horizontal flipping, rotation, zoom, height shift, width shift, scaling) are applied on the dataset. Some samples from the datasets containing four types of brain MRI and the image preprocessing techniques are presented in Fig. 3(a) and (b), respectively.

### 2.3. Feature extractors

For feature extraction, primarily five pre-trained models and a CNN model from scratch were considered. All the feature extractors used in this work are depicted in Fig. 4. Transfer learning is a process where a pre-trained model which has been trained on a particular problem is used on a similar other problem. It has the advantage of taking lower training time as it has already been trained with a similar problem. In the case of image classification problem, many CNN models have been recognized through the ImageNet challenge and those pre-trained models are used via transfer learning in different image classification problems. In this work, such 5 pre-trained CNN models have been modified and used for feature extraction - VGG-19 [48], EfficientNet-B0 [49], Inception-V3 [50], ResNet-50 [51] and Xception [52]. We have applied all these pre-trained models on 4 datasets that are mentioned above. Some modifications have been done on the models using random search which reduced overfitting to some extent.

#### 2.3.1. Modified VGG-19

VGG architecture [48] was introduced by Visual Geometry Group at Oxford as the name suggests. It uses deep convolutional layers for the improvement of accuracy. There are several versions of VGG and in our case, VGG-19 architecture has been fine-tuned with some modification. The original VGG-19 architecture consists of 16 convolutional layers and 3 fully connected layers. The modified architecture which has been fine-tuned for feature extraction has been shown in Fig. 4(a). The last three modified layers are added with the VGG-19 base model for feature extraction. Here, instead of using three fully connected layers (according to the original architecture), a global average pooling layer followed by a dropout and a dense layer are used. This modified architecture improved the model's performance by reducing over-fitting. Besides, the input layer is also changed to of size 256 × 256 to make the model compatible with the image size.

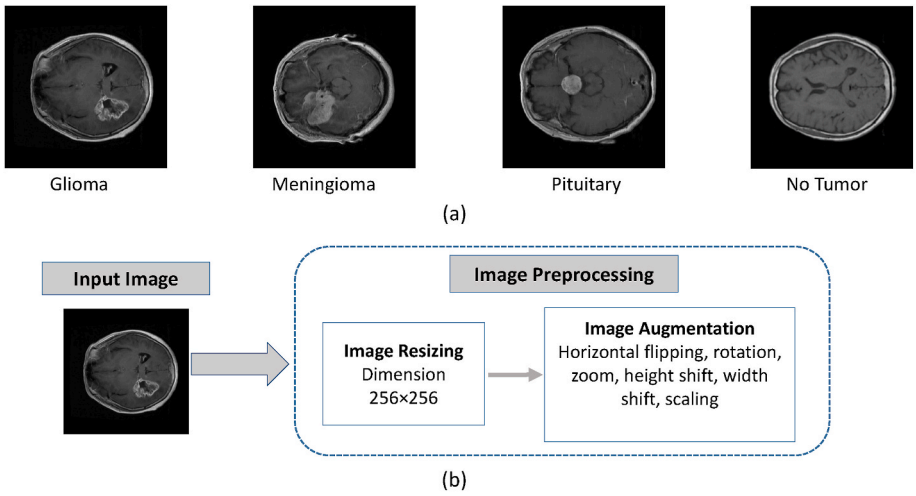


**Fig. 3.** (a) Samples of the MRI datasets including glioma tumor, meningioma tumor, pituitary tumor and normal brain image. (b) Data preprocessing stages where image resizing and augmentation is done.

#### 2.3.2. *Modified EfficientNet-B0*

EfficientNet architecture was introduced by Tan et al. [49] for systematic scaling of a model by balancing the network height, depth and input resolution for a better accuracy. In this work, the base EfficientNet-B0 network is used, which is based on the inverted bottleneck residual blocks of MobileNet-V2 depicted as *MBConv*. The original network uses input size of 224 × 224 and here in the modified network this input size is changed to 256 × 256. Additionally some layers are used on top of the base network, i.e., global average pooling layer, a dropout layer and a dense layer. The original base network with the modified layers on top is shown in Fig. 4(b). With random search and several trial and error this architecture found out to be more convenient.

#### 2.3.3. *Modified Inception-V3*

There are several versions of the Inception network [53] and the Inception-V3 comes after several improvement of the first Inception architecture [50]. This came with the idea of making deeper network by sparsely connected architecture. It's basically built with several inception module as shown in Fig. 4(c). Here, every module takes input from the previous stage. This input passes through multiple convolutional filters and finally the output of these filters are concatenated forming the input for the next stage. The architecture that is used in this paper contains some modified layers at the top of the base model, as shown in Fig. 4(c). Originally this architecture has the input size of 299 × 299 but in this case the input size is modified to 256 × 256. Among The last three layers, average pooling layer is modified to global average pooling layer and fully connected layer is modified to a dense layer of unit 768. This modification has also been done using random search like modified EfficientNet-B0.

#### 2.3.4. *Modified ResNet-50*

The Residual Networks, in short ResNet [51] won the ImageNet challenge in 2015 and is being used in many computer vision-related tasks. The main idea here is to train extremely deep neural network that overcomes the vanishing gradients problem and also reduces the number of parameter to a great extent. It uses skip connections between layers as shown in Fig. 4(d). Original architecture uses the input size of 224 × 224 which was modified to 256 × 256. Besides, at top of the base model, a global average pooling layer, dropout layer and dense layer have been added that improved the model's performance.

#### 2.3.5. *Modified Xception*

Xception network [52] is based on depth-wise separable convolution where point-wise convolution followed by depth-wise convolution. Unlike conventional convolution there is no need of performing convolution across all the channels. Thus it reduces the number of connections and results in decreasing the number of parameters as well. It also uses residual connections to improve accuracy. The model is basically divided in three flows - *entry flow, middle flow* and *exit flow*, as shown in 4 (e). The modified architecture uses input size of 256 × 256 instead of 299 × 299. In this architecture, a global average pooling layer is included by replacing the average pooling layer. Besides, two additional dense layers are used followed by the global average pooling layer.

#### 2.3.6. *Proposed scratched CNN model*

The architecture of the CNN model that is built from scratch is represented in Fig. 4 (f). This architecture is not as deep as the pre-trained models but the performance is at least as good as the state-of-the-art models. This model has been built through several experiments on hyper parameter tuning. The most optimal one is chosen using *keras tuner* and performing random search on different combinations of the model hyper-parameters. The hyper-parameters and their selected values have been shown in Table 2.

It's basically a 16 layer architecture consisting 4 convolution layer, 4 max pooling layer, 4 batch normalization layer, 1 flatten layer, 1 dense layer and a dropout layer. For convenience the whole architecture can be divided into 4 blocks. Each block contains a convolution layer, a batch normalization layer and a max pooling layer. Here the input size is 256 × 256. After passing through the 1st, 2nd, 3rd and 4th block the input size becomes 126 × 126, 62 × 62, 29 × 29 and 12 × 12 respectively. ReLU activation function has been used at each of the convolution layer and for the optimization, the Categorical cross-entropy loss function was used with the *Adam* optimizer.

### 2.4. *Machine learning classifiers*

The extracted features from the CNN models are passed through different classifiers for ultimate classification of brain tumors. We have used softmax classifier as standalone CNN models, besides we have applied four different classifiers- Support Vector Machine (SVM), Random Forest (RF), K Nearest Neigbour (KNN) and AdaBoost. All these classifiers are added with all the CNN models and based on their performance the best one is chosen.

Softmax can be considered as a generalization of logistic regression which is used in multi class classification [54]. This classifier is used at the last dense layer of each feature extractors mentioned earlier. For this, '*softmax*' activation is used in the dense layer with unit 3 or 4

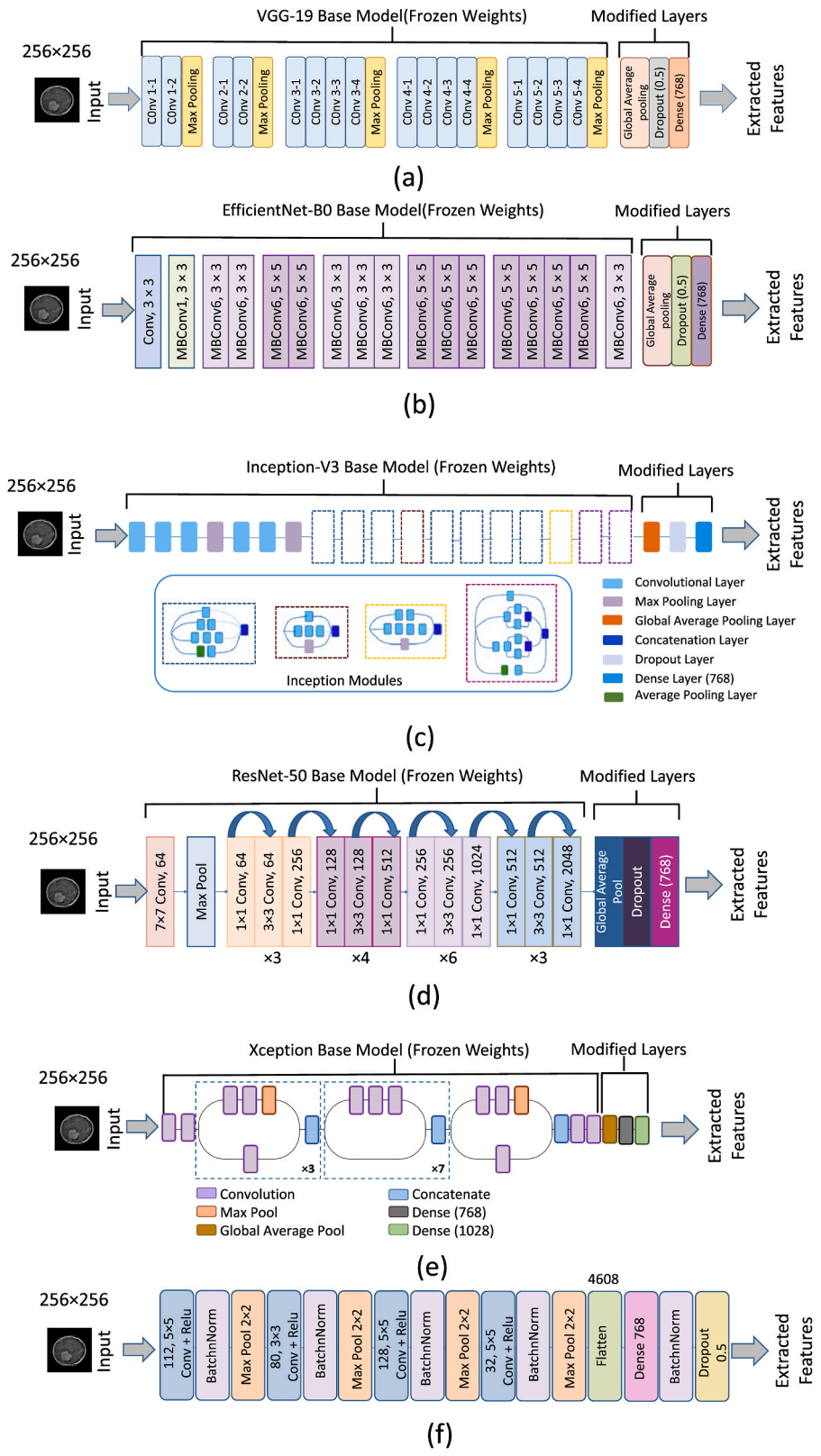


**Fig. 4.** CNN feature extractors used in this paper (a) Modified VGG-19 model. (b) Modified EfficientNet-B0 model. (c) Modified Inception-V3 model. (d) Modified ResNet-50 model. (e) Modified Xception model. (f) Proposed Scratched CNN model consists of 16 layers.

depending on the number of class labels of the datasets. Next, SVM is used as the classifier with each feature extractor. SVM [55] is used for classification tasks where it generates the best hyperplane separating two or more classes. For our problem we focused on three heper-parameters of SVM function, i.e., kernel, regularization *(C)* and *gamma*. The *'Radial Basis Function' ('RBF')* kernel worked best in our case as it is a non-linear problem. Among different values of regularization parameter, *C=0.5* and *gamma=0.01* worked best in our case.

**Table 2**
Search space and the selected values of hyper-parameters for the Proposed CNN model.

| hyper-parameter | Search space (min value, max value) | Selected value (optimum) |
|---|---|---|
| **No of Conv layer** | 2, 5 | 4 |
| **No of filters in Conv 1** | 32, 128 | 112 |
| **No of filters in Conv 2** | 32, 128 | 80 |
| **No of filters in Conv 3** | 32, 128 | 128 |
| **No of filters in Conv 4** | 32, 128 | 32 |
| **Kernel size of Conv 1** | 3, 5 | 5 |
| **Kernel size of Conv 2** | 3, 5 | 3 |
| **Kernel size of Conv 3** | 3, 5 | 5 |
| **Kernel size of Conv 4** | 3, 5 | 5 |
| **Units in Dense layer** | 32, 1028 | 768 |
| **Initial learning rate** | 1e-2, 1e-4 | 1e-2 |

Another classifier that we tried with every feature extractor is Random Forest (RF). It is a powerful supervised machine learning algorithm for performing classification task [56]. There are many parameter values in RF classifiers those need to be tuned. We focused mainly on two parameters, i.e., *n_estimator, random_state*. Using random search, we set the optimal value, *n_estimator* = *60* and *random_state* = *42*. Other parameters are set to default values. Then another classifier that we used is KNN [57] which works based on the similarity of features of the data point. For this classifier, we set the parameter *n_neighbors* = *3*. It denotes the number of neighbors. Other parameter values are set to default. Lastly, we used AdaBoost or *Adapting Boosting* algorithm [58] for classification which can achieve a higher accuracy by combining multiple weak classifiers. The weak individual classifier is called the base estimator. Decision tree classifier has been used as the base estimator in our paper. We set the value of *n_estimator* as 60 which specifies the number of models to be trained in iteration.

### 2.5. Selection of the best classifier and feature extractors

To select the best classifier all classifiers have been tested with the feature extractors for Dataset 1, 2, 3 and the Merged Dataset. The average accuracy of all classifiers for the individual feature extractors have been shown in Fig. 5(a). This depicts the average accuracy considering all four datasets. It is clearly evident that softmax performed

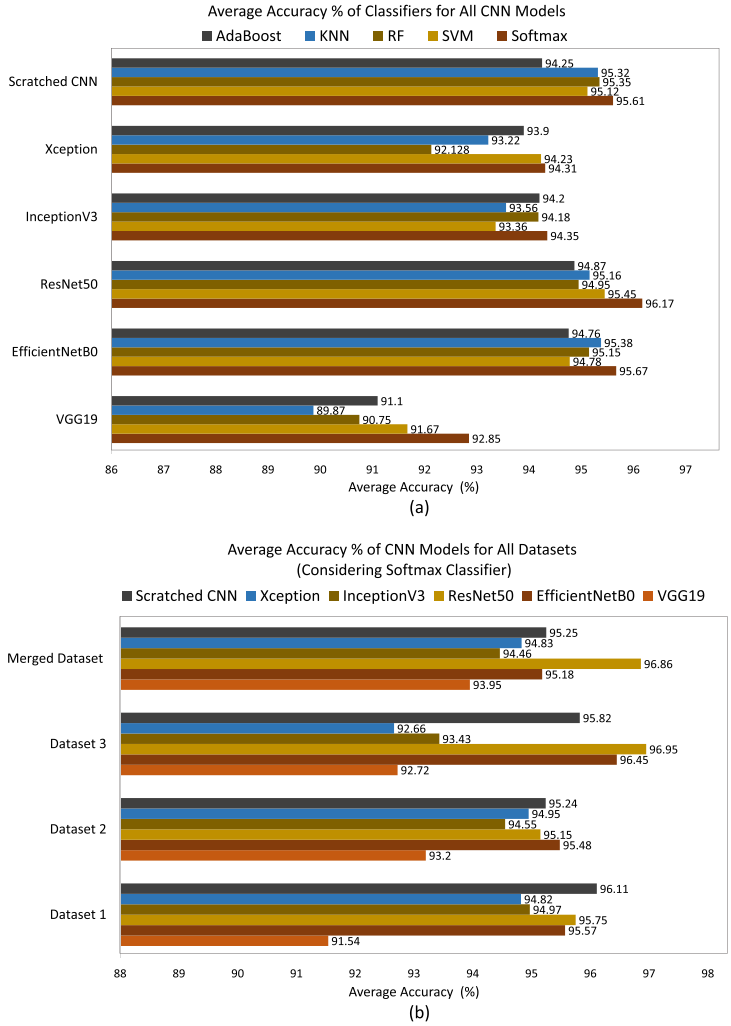


**Fig. 5.** Performance of the classifiers and feature extractors based on accuracy. (a) Among all classifiers softmax obtained the best average accuracy considering all feature extractors. (b) Among all feature extractors the Proposed Scratched CNN, EfficientNet-B0 and ResNet-50 attained the best accuracy scores considering all datasets.

best among all classifiers and so it is chosen for further classification task. Then all the feature extractors (VGG-19, EfficientNet-B0, ResNet-50, Inception-V3, Xception, Proposed CNN) are tested on Dataset 1, 2, 3 and the merged dataset with softmax classifier at the last dense layer. In this way, the best feature extractors are selected based on the performance on individual datasets. According to Fig. 5 (b), the Proposed CNN, EfficientNet-B0 and ResNet-50 are chosen as the best feature extractors for building the ensemble model.

### 2.6. Model hyper-parameters

Finding the optimal values of the hyper-parameters is one of the crucial and significant task for building a robust model. Besides good features extractor and classifiers, the hyper-parameters' values have a great influence in fast convergence of the model. During the training of the proposed model, different values of the hyper-parameters were tried randomly and tested with all our datasets. We focused on activation function, optimizer, learning rate, dropout rate, batch size, number of epochs, train-test splitting ratio etc. All the optimal values that has been chosen and used in the proposed model is shown in Table 3. As input activation function, ReLU [59] is used, which solves the vanishing gradient problem during training. It is defines in equation (1). For any negative input value of $x$, this function returns 0, otherwise it returns the value it receives as input.

$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} = max(0, x) \tag{1}$$

As we are doing multi-classification task, the output activation function is the softmax function that is defined in equation (2). Here, $\vec{x}$ denotes the input vector, $\exp(x_i)$ denotes the exponential of each element of the input vector $\vec{x}$ and $m$ denotes the number of classes. Adam [60] is chosen as the optimizer of the model, which can be considered as the combination of AdaGrad [61] and RMSProp [62]. It is also faster than stochastic gradient descent. The updating rule of Adam optimizer is defined in equation (3). Here, $\Theta$ is the model weight, $\eta$ is the learning rate, $\widehat{m}$ is the first moment vector, $\widehat{v}$ is the second moment vector and $t$ is the timestamp. Categorical cross entropy is used as the loss function and it is denoted in equation (4). Here, $y_i$ denotes the actual class label and $haty_i$ denotes the predicted class label.

$$softmax(\vec{x})_i = \frac{exp(x_i)}{\sum_{j=1}^{m} exp(x_j))} \tag{2}$$

$$\Theta_t = \Theta_{t-1} - \frac{\eta * \widehat{m_t}}{\sqrt{\widehat{v_t}} + \varepsilon} \tag{3}$$

$$Loss = -\sum_{i=1}^{outputsize} y_i * \log \widehat{y_i} \tag{4}$$

### 2.7. Feature extraction and classification using proposed two-stage ensemble approach

In this article, the feature extraction task is conducted in two stages.

**Table 3**
Hyper-parameter values for training the models.

| Hyper-parameter | Value |
|---|---|
| **Input activation function** | ReLU |
| **Output activation function** | Softmax |
| **Optimizer** | Adam |
| **Initial learning rate** | 0.001 |
| **Learning rate decay** | 0.2 |
| **Dropout rate** | 0.5 |
| **Early stopping patience** | 10 |
| **Batch size** | 32 |
| **No of epochs** | 30 |
| **Train-test split** | 90%–10% |

Selection of the models for ensemble is done based on their performance at each stage which is shown in Fig. 6. The whole process of first stage ensemble and second stage ensemble is presented in Fig. 7. The hyper-parameter values for the proposed model is shown in Table 3.

#### 2.7.1. First stage feature level ensemble

In the first stage, the best models are chosen from all six feature extractors those were primarily considered for this work. As shown in Fig. 5(b), the best accuracy was found for the proposed CNN, EfficientNet-B0, ResNet-50, ResNet-50 respectively for Dataset 1, Dataset 2, Dataset 3, and Merged Dataset. So among six models these three models ResNet-50, EfficientNet-B0, and proposed CNN are chosen for first stage ensemble as depicted in Fig. 6.

Prior to the first ensemble stage, a dense layer of unit 128 is added at top of each selected models. After the ensemble of these best three models, we get three combinations of first stage ensemble- (EfficientNet-B0 + ResNet-50), (ResNet-50 + proposed scratched CNN), (EfficientNet-B0 + proposed scratched CNN). These three ensemble models are further applied on all the datasets. The performance of the first stage ensemble models were much better than all the individual models with an average improvement in accuracy of 2.44%.

#### 2.7.2. Second stage feature level ensemble

From the performance of the first stage ensemble models, best models are selected for second stage ensemble. The validation accuracy plot of the first stage ensemble models is depicted in Fig. 8. It shows that for Dataset-1 and Merged Dataset, ensemble of ResNet-50 and proposed scratched CNN model performed the best. For Dataset-2 and Dataset-3, ensemble of EfficientNetB0 and ResNet50 performed the best. According to Table 4, two ensemble models- (ResNet-50 + proposed scratched CNN), (EfficientNet-B0 + ResNet-50) are eligible for second stage ensemble based on the validation accuracy. The best accuracy for all datasets is written in bold.

Finally, the selected two models from first stage ensemble models are further concatenated for building the second stage ensemble model as shown in Fig. 7. Before concatenation, a dense layer of unit 1024 is added on top of each of the ensemble models. This final ensemble model is able to perform even better than all the first stage ensemble models as it can grab more features than the previous ones. Lastly a dropout layer of 0.5 is added to this final two stage ensemble.

#### 2.7.3. Feature selection and tumor classification

After two stage ensemble as depicted in Fig. 7, important features are selected through Principal Component Analysis (PCA). PCA is basically used for reducing the dimension of a large dataset having a huge number of interrelated variables [63]. The performance of PCA in dimensionality reduction is proved to be better than Linear Discriminant Analysis (LDA) [64]. Normally when the dataset is larger, the number of interrelated features or variables also increases. PCA eliminates the correlated variables that have no contribution in decision making. This results in an improvement of the performance of classification model. The overfitting tendency of the model also reduces as a result of feature reduction. Before PCA the number of features were 5.6 million, 6.02 million, 7.9 million, and 19.57 million for dataset 1, dataset 2, dataset 3 and the merged dataset respectively. After PCA, this reduces approximately to a factor of 18.71. As shown in Fig. 9, only with 110 components almost 100% variance can be achieved for all the datasets using PCA. Thus number of features reduces to a great extent. Then this reduced features are passed to the classification layer. The basic formula of PCA is depicted in equation (5). Here, $X$ is the data matrix, $T$ are the scores, $P$ are the loadings and $E$ are the residuals.

$$X = TP^T + E \tag{5}$$

The classification layer is a dense layer with softmax activation function [equation (2)] of unit 3 or 4 depending on the tumor classes.

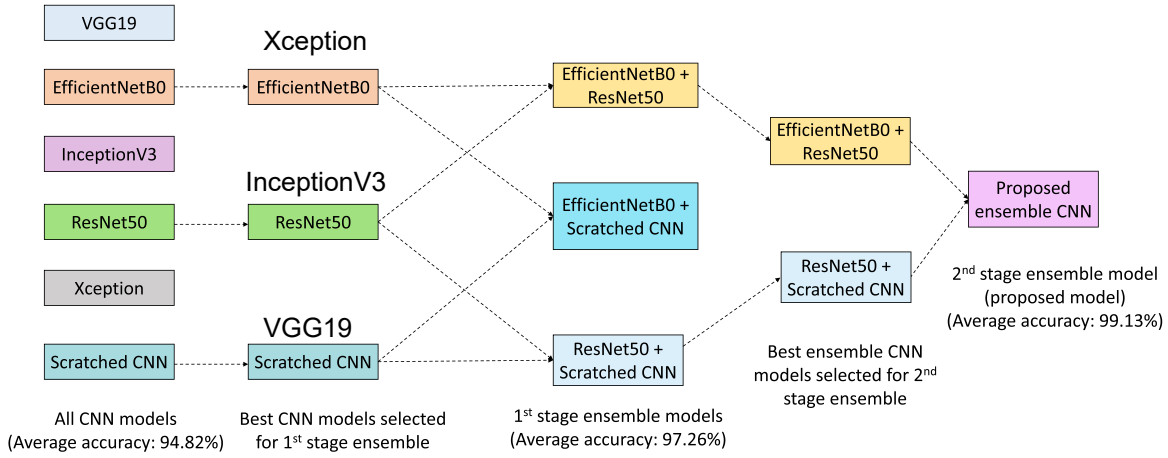


**Fig. 6.** Selection of feature extractors for two stage ensemble approach. Initially EffcientNet-B0, ResNet-50 and the Proposed CNN models are selected for the first stage ensemble. Then from all the combinations of first stage ensemble model, best combinations (EffcientNet-B0+ResNet-50, ResNet-50+Proposed Scratched CNN) are selected for second stage ensemble model.

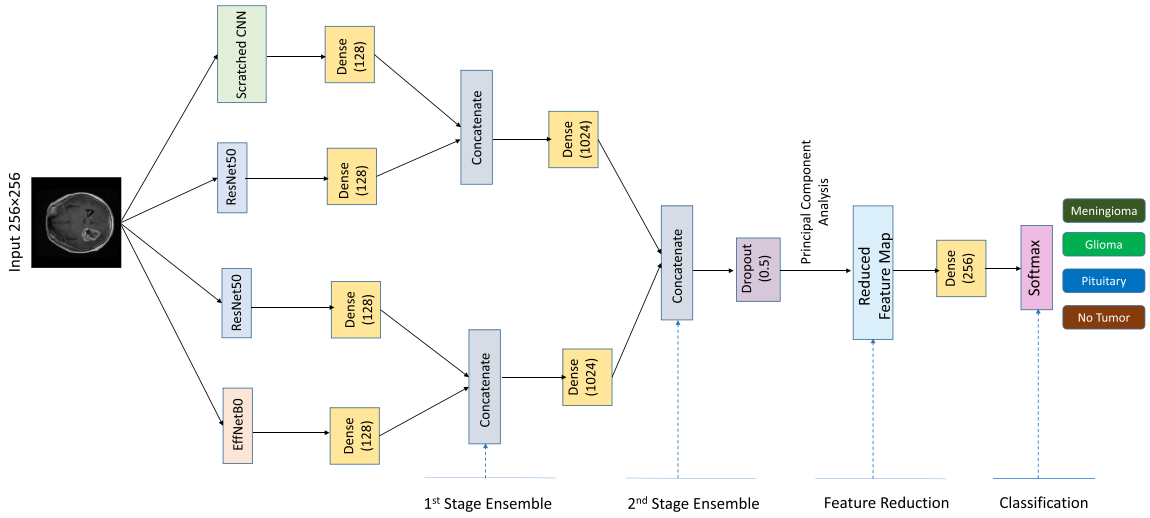


**Fig. 7.** Structure of the proposed two-stage ensemble model. In the first stage ensemble, the features from individual models are concatenated and in the second stage, the features from the ensemble models are again concatenated. Finally, the reduced features after PCA are fed to the classification layer.

## 3. Result & discussion

The proposed second stage ensemble model has been verified in different ways to accurately observe the performance. Firstly, the performance is observed based on different performance metrics. The impact of applying two stage ensemble technique is also discussed. How the performance upgraded after data augmentation and Principal Component Analysis is also analyzed. To check the generalization capability of the model, it is validated through three experiments. The proposed model is also compared with other existing models considering two aspects, i.e., individual datasets for training and validation, separate dataset for training and validation (leave-one-out approach). Finally, for the real-time validation, a user interface is built using python *Gradio* [65].

### 3.1. Performance metrics

For result analysis several performance metrics (precision, recall, f1-score, support, accuracy and ROC AUC score) are evaluated. *Accuracy* is the most common way to determine the performance of a classification model. It determines the ratio of the number of correct prediction to the total number of predictions. Precision is the ratio of true positive predictions and total positive predictions. Recall is the ratio of true positive and sum of true positive and false negative predictions. F1-score is the harmonic mean of precision and recall. ROC AUC score determines the model's capability to distinguish among different classes and higher value refers to a better performance. The formulas of determining precision, recall, f1 score and accuracy are shown in equations (6)–(9) respectively.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

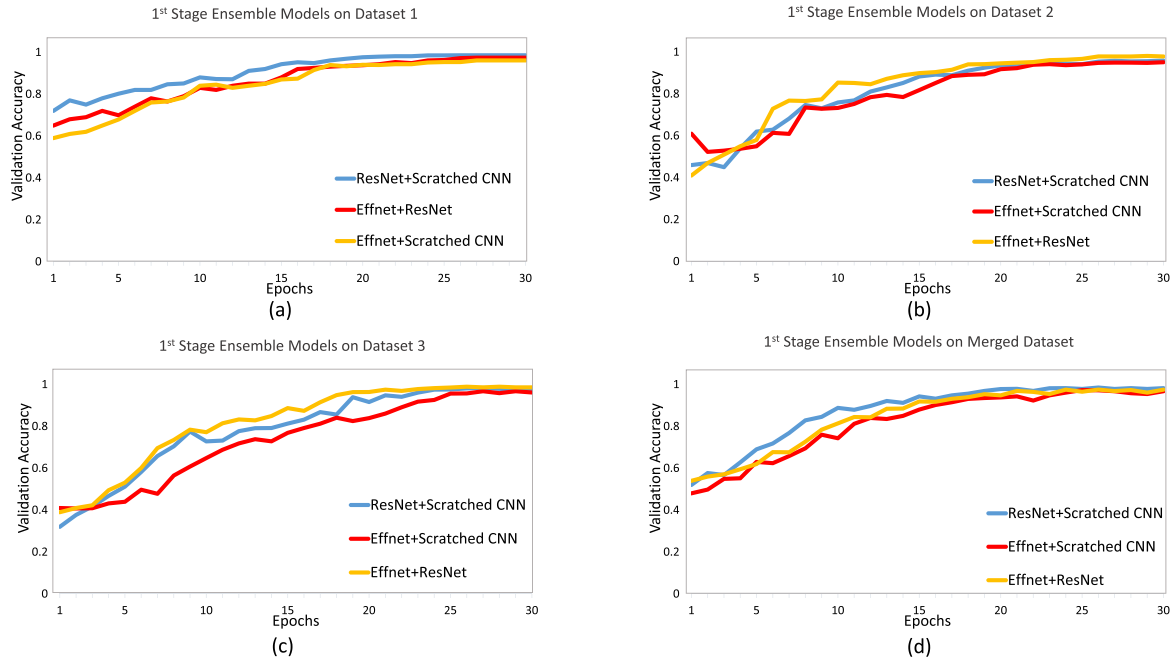


**Fig. 8.** Epochs vs validation accuracy graph of the first stage ensemble models. (a) For dataset 1, ResNet-50+proposed scratched CNN shows the best performance. (b) For datset 2, EfficientNet-B0+ResNet-50 shows the best performance. (c) For dataset 3, EfficientNet-B0+ResNet-50 shows the best performance. (d) For Merged Dataset, ResNet-50+proposed scratched CNN shows the best performance.

**Table 4**
Validation accuracy of the first stage ensemble models.

| Model | Validation Accuracy % | | | |
|---|---|---|---|---|
| | Dataset 1 | Dataset 2 | Dataset 3 | **Merged Dataset** |
| **ResNet50+proposed CNN** | **98.53** | 96.83 | 97.71 | **97.98** |
| **EfficientNetB0+proposed CNN** | 96.15 | 95.72 | 97.41 | 97.23 |
| **EfficientNetB0+ResNet50** | 96.22 | **97.12** | **98.819** | 97.41 |

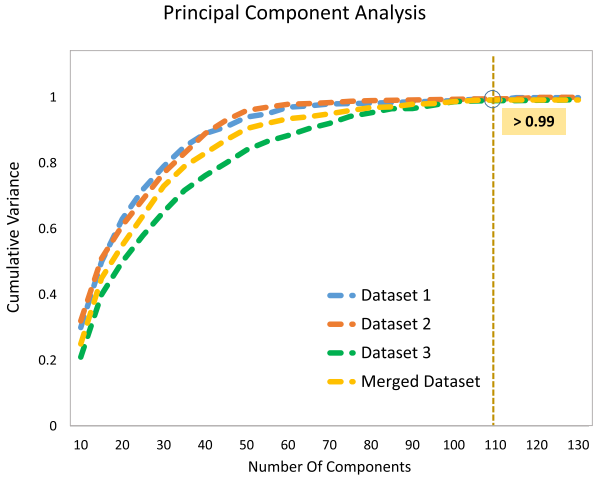


**Fig. 9.** Cumulative variance of the features for four datasets after Principal Component Analysis.

$$F1\ Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} * 100\ \% \tag{9}$$

In equations (6)–(9), TP is the number of true positive predictions, TF is the number of true negative predictions, FP is the number of false positive predictions and FN is the number of false negative predictions. The result of our proposed ensemble CNN model is shown in Table 5 where the precision, recall, f1-score, support and accuracy for all individual datasets have been presented. Confusion matrix of each dataset is shown in Fig. 10. Here, 'M', 'G', 'P' and 'N' refer to Meningioma, Glioma, Pituitary and Normal brain images.

### 3.2. Impact of two stage ensemble

For a single CNN model it is rare to extract all the important features from an image. For training a single model on a particular dataset, it's not possible to achieve a high generalization capability. To address this limitation, two-stage ensemble approach is used in this paper. The performance gain of two-stage ensemble model is clearly visible in Fig. 11(a), Fig. 11(b), (c) and (d), where the epochs-vs-validation accuracy of Daatset 1, Dataset 2, Daaset 3 and Merged Daatset are shown respectively. From the graphs it is evident that the two-stage ensemble model shows better convergence during training. Without ensemble, the average accuracy of all individual models is 94.82%. After the first-stage ensemble the average accuracy of the models improves 2.44% and becomes 97.26%. Finally, after second-stage ensemble the average accuracy of the proposed model becomes 99.13%. Thus the overall accuracy improves 4.31%. Fig. 12 shows the improvement in performance after applying the two-stage ensemble model. Table 6 shows the performance comparison of the models before and after two stage ensemble considering all datasets. Here we can see, for Dataset-1, Dataset-2, Dataset-3, and the merged dataset, the average accuracy improves for two-stage ensemble.

**Table 5**
Performance of the proposed two stage ensemble model considering different performance metrics.

| Dataset | Tumor labels | Precision | Recall | F1 Score | Validation Accuracy % | Validation Loss | ROC AUC Score |
|---|---|---|---|---|---|---|---|
| Dataset 1 | Meningioma | 1.00 | 0.99 | 1.00 | 99.67 | 0.024 | 0.994 |
| | Glioma | 0.99 | 1.00 | 0.99 | | | |
| | Pituitary | 1.00 | 1.00 | 1.00 | | | |
| Dataset 2 | Meningioma | 0.99 | 0.96 | 0.97 | 98.16 | 0.063 | 0.982 |
| | Glioma | 0.97 | 0.99 | 0.98 | | | |
| | Pituitary | 0.99 | 1.00 | 0.99 | | | |
| | No Tumor | 0.98 | 0.98 | 0.98 | | | |
| Dataset 3 | Meningioma | 1.00 | 0.99 | 1.00 | 99.76 | 0.019 | 0.998 |
| | Glioma | 0.99 | 1.00 | 1.00 | | | |
| | Pituitary | 1.00 | 1.00 | 1.00 | | | |
| | No Tumor | 1.00 | 1.00 | 1.00 | | | |
| **Merged Dataset** | Meningioma | 0.98 | 0.99 | 0.99 | 98.96 | 0.059 | 0.989 |
| | Glioma | 0.98 | 0.98 | 0.98 | | | |
| | Pituitary | 1.00 | 0.99 | 1.00 | | | |
| | No Tumor | 1.00 | 1.00 | 1.00 | | | |

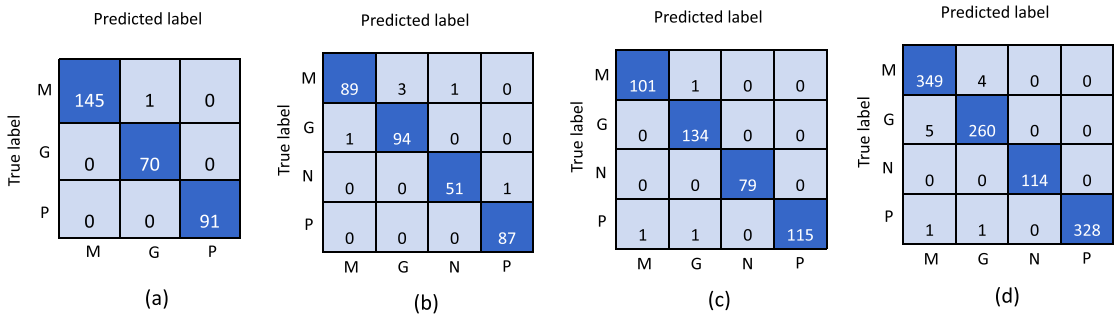


**Fig. 10.** Confusion matrix after applying the proposed two stage ensemble model. Confusion matrices for (a) Dataset 1, (b) Dataset 2, (c) Dataset 3, and (d) Merged Dataset.

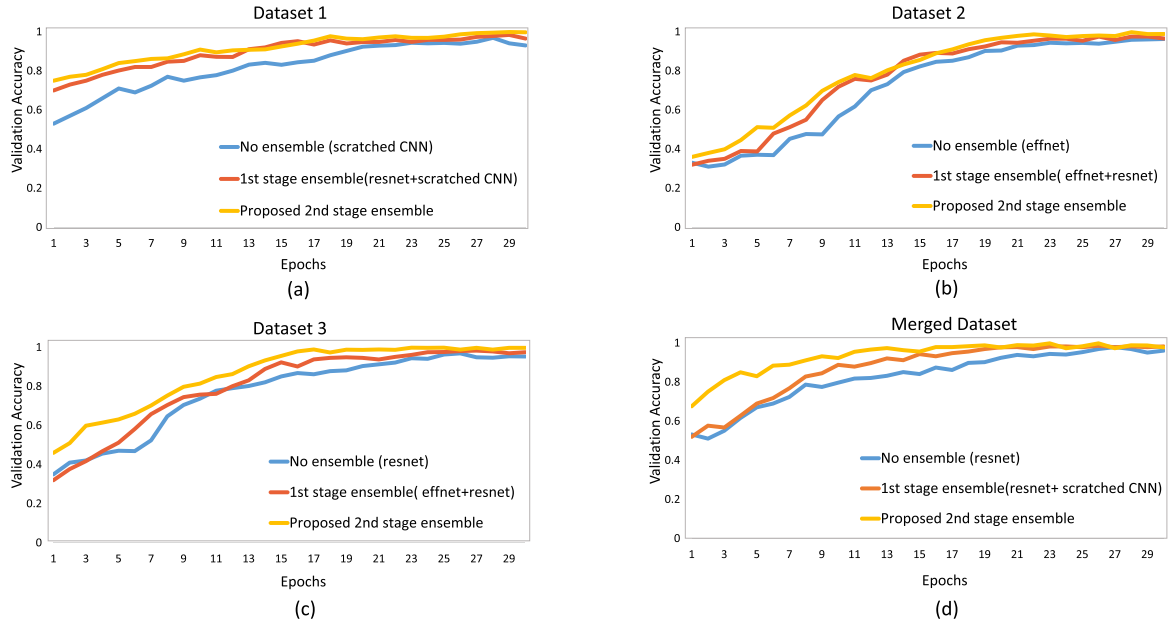


**Fig. 11.** Impact of two stage ensemble on all the datasets where two stage ensemble model outperformed the other models. Epochs vs Validation Accuracy for (a) Datset 1, (b) Dataset 2, (c) Dataset 3, and (d)Merged Dataset.

### *3.3. Impact of data augmentation*

One of the important prerequisites to build a robust model is to use a dataset, which is large enough and diverse in quantity. It's not always possible to collect a rich dataset due to its unavailability. New and different data can be formed by slightly modifying the existing data, which is called data augmentation. In this approach model's robustness and performance can improve and also over-fitting can be avoided [66]. As mentioned earlier, six types of augmentation have been applied on all the datasets to acquire a robust model. We can clearly see from Fig. 13

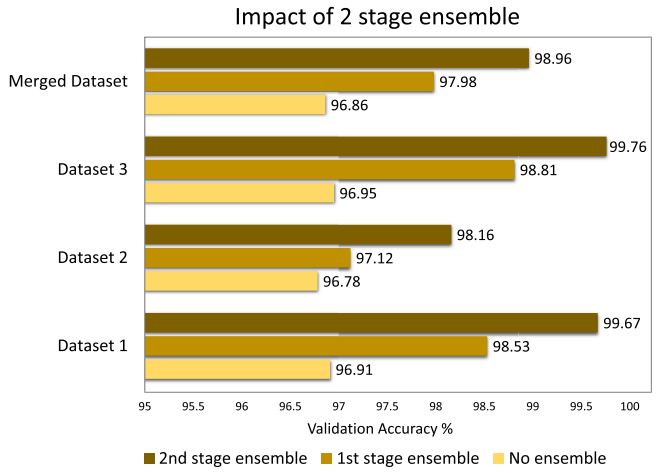


**Fig. 12.** Average improvement in accuracy after applying two stage ensemble model on all datasets.

**Table 6**
Performance comparison of the models before ensemble, after one-stage and two-stage ensembles.

| Validation Accuracy % | | | |
|---|---|---|---|
| Dataset | Without ensemble | After one stage ensemble | After two stage ensemble |
| Dataset 1 | 94.79 | 96.96 | 99.67 |
| Dataset 2 | 94.76 | 96.55 | 98.16 |
| Dataset 3 | 94.67 | 97.97 | 99.76 |
| Merged Dataset | 95.08 | 97.54 | 98.96 |

(a), Fig. 13 (b), Fig. 13 (c), Fig. 13 (d), how data augmentation improves the validation accuracy during training on Dataset-1, Dataset-2, Dataset-3 and the Merged Dataset, respectively. Table 7 shows the validation accuracy of the proposed model before and after performing data augmentation for all the datasets. After augmentation the validation accuracy increases 1.35%, 1.04%, 1.24%, 1.75% for Dataset 1, Dataset 2, Dataset 3 and Merged Dataset, respectively.

### 3.4. Impact of principal component analysis

To improve model's performance, in terms of both execution time and validation accuracy, principal component analysis (PCA) is used. Through PCA, number of parameters is reduced by selecting only the significant features having the maximum explained variance. This approach reduces the number of parameters to a great extent, which results in a huge minimization of model execution time. Moreover, validation accuracy slightly improves because of taking only significant features. Table 8 shows the performance comparison of the proposed model before and after PCA in terms of execution time, validation accuracy and feature maps.

### 3.5. Validating model's robustness and generalization capability

Having a high generalization capability is a prerequisite for a robust classification model. Lack of generalization capability makes a model incompatible for real-world application. To validate the robustness and generalization capability, three experiments have been done, i.e., training and validation on separate datasets, random splitting of dataset and *k*-fold cross validation, as discussed below.

#### 3.5.1. Training and validation on separate datasets

To validate the generalization capability of the model, we adopted the leave-one-out approach, where out of three datasets, i.e., Dataset 1, Dataset 2, and Dataset 3, two of them have been used as the training set and the other as the validation set. In this way, we can find three combinations for validating our proposed ensemble model. The result of applying this approach is shown in Table 9. In Fig. 14 (a), the validation

**Table 7**
Validation accuracy before and after data augmentation.

| | Validation Accuracy % | |
|---|---|---|
| **Dataset** | **Before augmentation** | **After augmentation** |
| Dataset 1 | 98.32 | 99.67 |
| Dataset 2 | 97.92 | 98.16 |
| Dataset 3 | 98.52 | 99.76 |
| Merged Dataset | 97.21 | 98.96 |

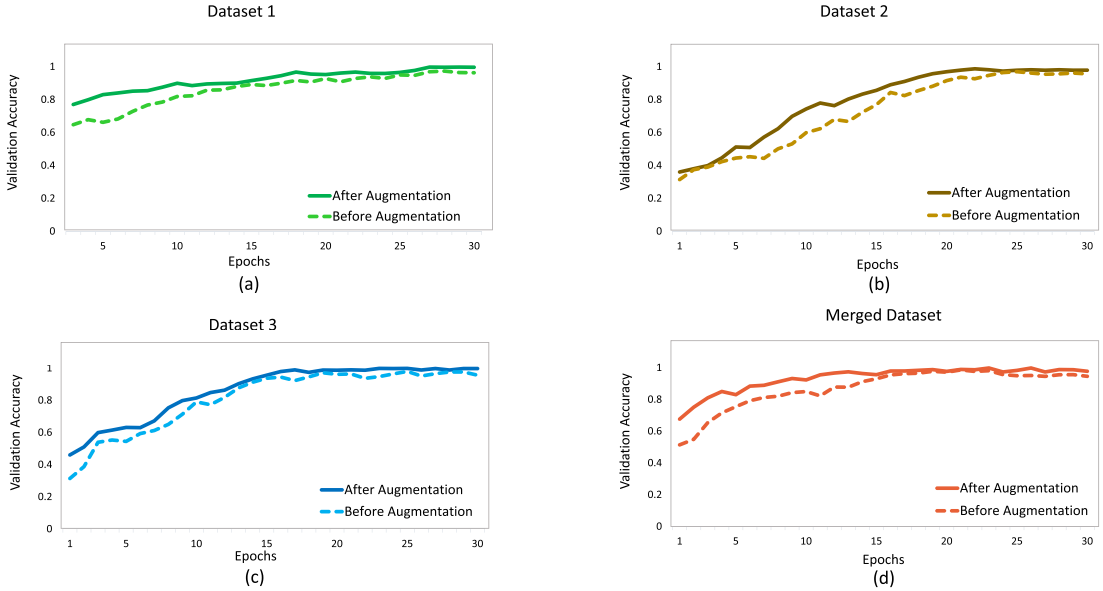


**Fig. 13.** Impact of Data Augmentation. Epochs-vs-Validation accuracy before and after augmentation on (a) Dataset 1, (b) Dataset 2, (c) Dataset 3, and (d) Merged Dataset.

**Table 8**
Impact of Principal Component Analysis based on execution time, validation accuracy and number of feature maps.

| Dataset | Execution time/epoch (second) | | Validation Accuracy % | | Number of feature maps (million) | |
|---|---|---|---|---|---|---|
| | Without PCA | With PCA | Without PCA | With PCA | Without PCA | With PCA |
| Dataset 1 | 42.3 | 6.9 | 99.64 | 99.67 | 5.6 | 0.3 |
| Dataset 2 | 45.4 | 7.3 | 98.11 | 98.16 | 6.02 | 0.32 |
| Dataset 3 | 51.6 | 9.1 | 99.68 | 99.76 | 7.9 | 0.42 |
| **Merged Dataset** | 67.9 | 11.3 | 98.88 | 98.96 | 19.57 | 1.05 |

**Table 9**
Performance of the proposed model considering the leave-one-out approach for training and validation with the Dataset 1, Dataset 2, and Dataset 3.

| Training Set | Validation Set | Validation Accuracy % |
|---|---|---|
| Dataset1+Dataset2 | Dataset3 | 98.96 |
| Dataset2+Dataset3 | Dataset1 | 99.13 |
| Dataset1+Dataset3 | Dataset2 | 98.84 |

accuracy plot is also shown for the three combination of training/validation dataset. It is very clear that our model has shown quite satisfactory performance in each of the cases.

### 3.5.2. Random splitting of datasets

Next, we split our datasets into different ratios of training and test sets as shown in Table 10. This is done to check the performance of our model when the dataset is randomly divided in any ratio. According to the result we get, the performance does not degrade so much due to this random splitting. We have used 3 ratios here. 90% in the training set, 80% in the training set and 70% in the training set. For all cases the accuracy remains quite competitive on all datasets.

### 3.5.3. K-fold cross validation

Lastly, in the third experiment in proving model's robustness, the proposed model is verified by K-fold cross validation. It is a more appropriate approach to validate a model as it uses every observation in test set and training set [67]. In this process the whole dataset is divided into k equal parts where eventually each part is considered as test set and the remaining as the training set. That's how this ensures that no observation is remained without being tested by the model. The value of k depends on the size of the dataset. We have applied 8-fold cross validation for all the datasets and the result of each fold is presented in Table 11. Fig. 14 (b) illustrates the model's performance after applying 8-fold cross validation.

## 3.6. Comparison with other existing models

The proposed model is compared with other existing models in terms of different evaluation metrics. Here, the comparison is done based on three aspects. In the first case, individual dataset is taken under consideration for making comparison. In the second case, comparison is done by training and validating on separate datasets which means considering all the datasets. Lastly, comparison is done considering the models that used feature fusion approach.

### 3.6.1. Comparison on individual dataset

In this case, all four datasets (Datast-1, Dataset-2, and Dataset-3, and Merged Dataset) have been considered individually. Based on these

**Table 10**
Performance of the proposed model considering different ratio of training and validation set.

| Dataset | Training testing data splitting | Accuracy % | Average accuracy % |
|---|---|---|---|
| Dataset 1 | 90%–10% | 99.67 | 99.2 |
| | 80%–20% | 99.28 | |
| | 70%–30% | 98.67 | |
| Dataset 2 | 90%–10% | 98.16 | 98.05 |
| | 80%–20% | 98.11 | |
| | 70%–30% | 97.89 | |
| Dataset 3 | 90%–10% | 99.76 | 98.62 |
| | 80%–20% | 98.67 | |
| | 70%–30% | 97.44 | |
| Merged | 90%–10% | 98.96 | 98.75 |
| | 80%–20% | 98.82 | |
| | 70%–30% | 98.47 | |

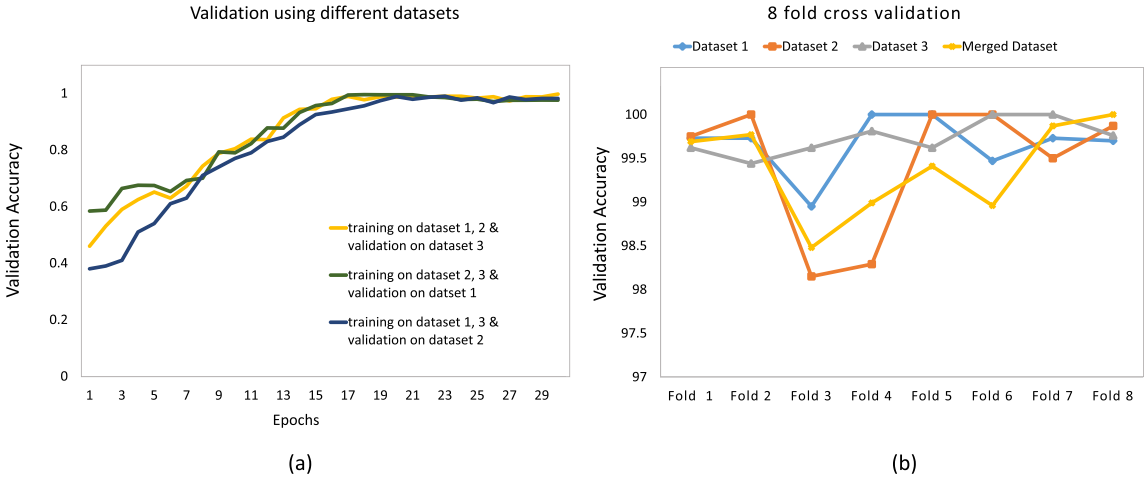


**Fig. 14.** (a) Training and validation on separate datasets. Here, 3 cases are considered i) training with Dataset 1, 2 and validating with Dataset 3, ii) training with Dataset 2, 3, and validating with Dataset 1, and iii) training with Dataset 1, 3, and validating with Dataset 2. (b) Model accuracy after 8-fold cross validation for all datasets.

**Table 11**
Performance of the proposed model after 8 fold cross validation.

| Validation Accuracy % | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Fold 6 | Fold 7 | Fold 8 |
| Dataset 1 | 99.73 | 99.73 | 98.95 | 100 | 100 | 99.47 | 99.73 | 99.7 |
| Dataset 2 | 99.75 | 100 | 98.15 | 98.29 | 100 | 100 | 99.5 | 99.87 |
| Dataset 3 | 99.62 | 99.44 | 99.62 | 99.81 | 99.62 | 100 | 100 | 99.76 |
| Merged Dataset | 99.69 | 99.77 | 98.48 | 98.99 | 99.41 | 98.96 | 99.87 | 100 |

datasets different existing models are compared with the proposed model.

Badža et al. [18] used a CNN architecture of 22 layers including 4 convolutional layers, each followed by a ReLU, dropout and max pooling layer. They used Adam optimizer, mini batch size of 16 and initial learning rate of 0.0004. Ayadi et al. [19] used 10 blocks of convolutional, ReLU and batch normalization layers, each followed by a max pooling layer. They used Adagrad optimizer with a learning rate of 0.003. Here, the batch size was 16 and number of epochs was 20. Sultan et al. [20] used a CNN architecture of total 16 layers each followed by a ReLU and max pooling layer. They used L2 regularization and stochastic gradient descent as optimizer. Anaraki et al. [68] used 6 convolutional layers, max pooling layers and a fully connected layer. ELU has been used as the activation function and Adam as the optimizer with total 100 epochs. Kumar et al. [69] used ResNet-50 model with SDGM optimizer. The number of epochs, batch size and initial learning rate was 10, 2 and 0.001 respectively. Abiwinanda et al. [70] used a CNN architecture of 2 convolutional layer, ReLU and max pooling layer and Adam optimizer. Deepak et al. [21] used a CNN architecture of 5 convolutional layers and 2 fully connected layers. They used batch size of 128, ReLU as activation function, Adam as optimizer, initial learning rate of 0.001 and 20 epochs for training. Swati et al. [71] used fine-tuned VGG-19 with mnibatch size 64 and maximum number of epoch 50. The performance comparison of the proposed model with these existing recent models considering Dataset 1 is shown in Table 12. Here, we can see that the proposed model outperformed all the existing models with the highest precision, recall, f1 score and validation accuracy. For Dataset 2, Dataset 3, and Merged Dataset we rebuilt some existing models and then trained and validated them using Dataset 2, Dataset 3, and Merged Dataset individually. The result after validation of the existing models on Dataset 2, Dataset 3, and Merged Dataset is shown in Table 13, Table 14, and Table 15 respectively. From the performance comparison presented in these tables, it is evident that the proposed model performed quite better than the existing models in terms of all evaluation metrics considering all the datasets. The validation accuracy of all the existing models along with the proposed model is depicted in Fig. 15 using bar graphs.

### 3.6.2. Comparison on separate training and validation data

All the existing models are trained and validated on separate datasets. As, we have three unique datasets, taking one of them as validation set we get three combinations. After training and validating in this way, the validation accuracy is compared in Table 16, where it shows that the proposed model obtained the highest accuracy. Fig. 16 also depicts that the proposed two stage ensemble model outperformed all the existing models. This proves that the proposed model has better generalization capability than the state-of-the-art models.

### 3.6.3. Comparison with other fusion based models

The proposed model is compared with other existing fusion based models. Though the proposed two-stage ensemble approach is not yet available in classifying brain tumors from MRI, several other feature fusion models have been applied by the researchers. For example, Khan et al. [33] used VGG-19 and VGG-16 fusion approach with extreme learning machine classifier. Noreen et al. [34] used Inception-V3 and DenseNet201 module concatenation approach. Amin et al. [37] used score level fusion of AlexNet and GoogleNet. The comparison with these models is presented in Table 17. Our proposed model outperformed these fusion approaches in terms of model accuracy. Moreover, because of applying two stage ensemble approach, it showed more robustness and generalization capability.

## 3.7. Real-time validation through user interface

For real time validation, a User Interface (UI) is created using the proposed two-stage ensemble model. The Interface is created based on Gradio [65] which is a python open source library. We have used *Python 3.7.12* version and *Gradio 2.4.6* version for our work. *Gradio* is used for making quick and customizable web based Graphical User Interface (GUI) where we can deploy Machine Learning models or a specific function. This helps us in building a prototype of the deployed model to test the performance in user environment. Gradio Interface basically takes three arguments-the function where the proposed classification model is specified, the input components and the output components. After executing this, a web page is created containing the User Interface. This URL of the web page is shareable and can be made public. As this is publicly accessible, anyone with their browser can use it as long as the host device is working. Fig. 17 illustrates the UI, where the image is uploaded and submitted for classification. The button 'Interpret' is used for highlighting the area of the image that contributes for classification output. It helps the users to understand easily about the important parts of the images that is responsible for the output. Screenshot of the classification output can be saved clicking the button 'Screenshot'. The 'Flag' button can be used when a user finds any interesting or erroneous output. By clicking this button that particular input will be saved under a folder called flagged and the creator of the interface can be notified. Later, this input can be used to improve the model's performance. This

**Table 12**
Performance comparison of the proposed model with other existing models considering Dataset 1.

| Ref. | Year | Method used | Precision | Recall | F1 score | Validation Accuracy % | Validation Loss |
|---|---|---|---|---|---|---|---|
| **Badža *et al.*** [18] | 2020 | CNN | 0.97 | 0.98 | 0.974 | 97.28 | – |
| **Ayadi *et al.*** [19] | 2021 | CNN | 0.94 | 0.94 | 0.94 | 94.74 | – |
| **Sultan *et al.*** [20] | 2019 | Deep neural network | 0.96 | 0.94 | – | 96.13 | – |
| **Anaraki *et al.*** [68] | 2019 | CNN | 0.943 | 0.942 | – | 94.2 | – |
| **Kumar *et al.*** [69] | 2021 | Residual network and global average pooling | 0.972 | 0.972 | 0.972 | 97.08 | – |
| **Abiwinanda *et al.*** [70] | 2019 | CNN | – | – | – | 84.19 | – |
| **Deepak *et al.*** [21] | 2020 | CNN and SVM | 0.957 | 0.949 | – | 95.82 | – |
| **Swati *et al.*** [71] | 2019 | Fine tuned VGG19 | 0.895 | 0.942 | 0.917 | 94.82 | – |
| **Proposed model** | **–** | **Two stage ensemble CNN model** | **0.996** | **0.996** | **0.996** | **99.67** | **0.024** |

**Table 13**
Performance comparison of the proposed model with other existing models considering Dataset 2.

| Ref. | Year | Method used | Precision | Recall | F1 score | Validation Accuracy % | Validation Loss |
|---|---|---|---|---|---|---|---|
| **Badža *et al.*** [18] | 2020 | CNN | 0.9 | 0.88 | 0.885 | 89.45 | 0.414 |
| **Ayadi *et al.*** [19] | 2021 | CNN | 0.957 | 0.957 | 0.955 | 95.71 | 0.226 |
| **Sultan *et al.*** [20] | 2019 | Deep neural network | 0.9 | 0.91 | 0.905 | 90.21 | 0.614 |
| **Anaraki *et al.*** [68] | 2019 | CNN | 0.912 | 0.897 | 0.905 | 90.51 | 0.785 |
| **Kumar *et al.*** [69] | 2021 | Residual network and global average pooling | 0.952 | 0.952 | 0.955 | 95.1 | 0.176 |
| **Abiwinanda *et al.*** [70] | 2019 | CNN | 0.892 | 0.88 | 0.885 | 88.68 | 0.67 |
| **Deepak *et al.*** [21] | 2020 | CNN and SVM | 0.92 | 0.92 | 0.9 | 90.21 | 0.853 |
| **Proposed model** | – | **Two stage ensemble of CNN models** | **0.98** | **0.98** | **0.98** | **98.16** | **0.063** |

**Table 14**
Performance comparison of the proposed model with other existing models considering Dataset 3.

| Ref. | Year | Method used | Precision | Recall | F1 score | Validation Accuracy % | Validation Loss |
|---|---|---|---|---|---|---|---|
| **Badža *et al.*** [18] | 2020 | CNN | 0.935 | 0.935 | 0.935 | 93.48 | 0.26 |
| **Ayadi *et al.*** [19] | 2021 | CNN | 0.985 | 0.98 | 0.98 | 98.13 | 0.118 |
| **Sultan *et al.*** [20] | 2019 | Deep neural network | 0.972 | 0.96 | 0.965 | 96.64 | 0.137 |
| **Anaraki *et al.*** [68] | 2019 | CNN | 0.9 | 0.892 | 0.897 | 89.76 | 0.323 |
| **Kumar *et al.*** [69] | 2021 | Residual network and global average pooling | 0.992 | 0.992 | 0.992 | 99.3 | 0.031 |
| **Abiwinanda *et al.*** [70] | 2019 | CNN | 0.93 | 0.917 | 0.925 | 92.55 | 0.471 |
| **Deepak *et al.*** [21] | 2020 | CNN and SVM | 0.94 | 0.92 | 0.93 | 93.02 | 0.819 |
| **Proposed model** | – | **Two stage ensemble of CNN models** | **0.997** | **0.997** | **1** | **99.76** | **0.019** |

**Table 15**
Performance comparison of the proposed model with other existing models considering the Merged Dataset.

| Ref. | Year | Method used | Precision | Recall | F1 score | Validation Accuracy % | Validation Loss |
|---|---|---|---|---|---|---|---|
| **Badža *et al.*** [18] | 2020 | CNN | 0.917 | 0.925 | 0.92 | 91.9 | 0.367 |
| **Ayadi *et al.*** [19] | 2021 | CNN | 0.983 | 0.985 | 0.983 | 98.49 | 0.069 |
| **Sultan *et al.*** [20] | 2019 | Deep neural network | 0.97 | 0.965 | 0.967 | 96.61 | 0.166 |
| **Anaraki *et al.*** [68] | 2019 | CNN | 0.912 | 0.915 | 0.912 | 91.05 | 0.303 |
| **Kumar *et al.*** [69] | 2021 | Residual network and global average pooling | 0.982 | 0.98 | 0.982 | 98.02 | 0.061 |
| **Abiwinanda *et al.*** [70] | 2019 | CNN | 0.97 | 0.97 | 0.97 | 97.08 | 0.182 |
| **Deepak *et al.*** [21] | 2020 | CNN and SVM | 0.972 | 0.97 | 0.972 | 97.17 | 0.778 |
| **Proposed model** | – | **Two stage ensemble of CNN models** | **0.99** | **0.99** | **0.99** | **98.96** | **0.059** |

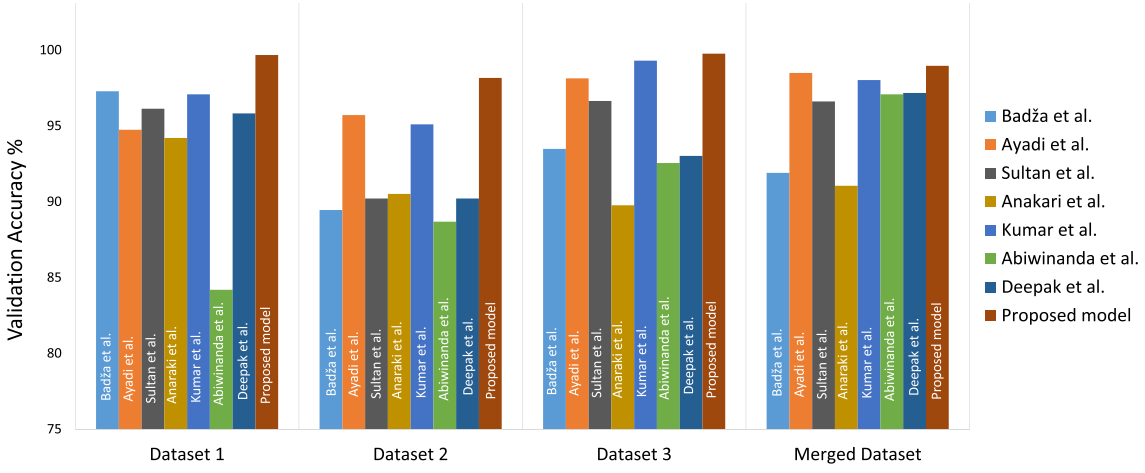


**Fig. 15.** Performance comparison of the proposed two-stage ensemble model with some state-of-the-art models. The proposed model outperformed all the models considering all the datasets.

actually works as a feedback from the users for the advancement of the model. Finally, for clearing the current image 'Clear' button is clicked.

Some random MRI images of different types of tumors are uploaded and tested in real-time that is shown in Fig. 18. Here, for classifying meningioma and glioma tumor, the model shows 98% and 97% confidence respectively. For pituitary and normal brain images it shows 100% confidence. Thus the proposed model performs quite prominently in real-time environment. User can also edit the uploaded image by cropping, rotating, adding noise to see how the output changes.

**Table 16**
Performance comparison of the proposed model with other existing models considering separate training and validation set.

| Validation Accuracy % | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Training set | Validation set | Badža et al. [18] | Ayadi et al. [19] | Sultan et al. [20] | Anaraki et al. [68] | Kumar et al. [69] | Abiwinanda et al. [70] | Deepak et al. [21] | Proposed model |
| Dataset1+Dataset2 | Dataset3 | 91.20 | 95.31 | 95.23 | 88.33 | 96.12 | 90.05 | 92.15 | 98.96 |
| Dataset2+Dataset3 | Dataset1 | 95.89 | 93.51 | 95.29 | 92.56 | 95.40 | 83.21 | 94.57 | 99.13 |
| Dataset1+Dataset3 | Dataset2 | 85.98 | 93.97 | 89.15 | 88.94 | 94.89 | 86.91 | 90.05 | 98.84 |

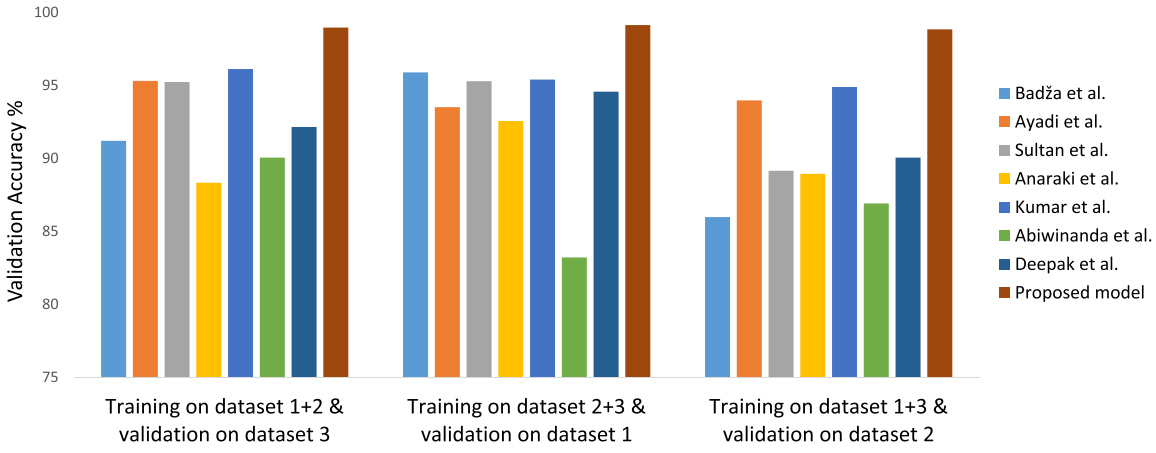


**Fig. 16.** Performance comparison of the proposed model with other existing models considering separate training and validation data. The proposed model showed better performance in all three combinations.

**Table 17**
Performance comparison of the proposed model with other existing models based on feature fusion.

| Ref. | Year | Method used | Classification type | Best Validation Accuracy % |
|---|---|---|---|---|
| Khan et al. [33] | 2020 | VGG-16 and VGG-19 feature fusion | Multi | 97.8 |
| Noreen et al. [34] | 2020 | DenseNet-201 module concatenation | Multi | 99.51 |
| Sachdeva et al. [35] | 2012 | Dual network ensemble using ANN | Multi | 95.85 |
| Amin et al. [37] | 2019 | AlexNet and GoogleNet fusion | Binary | 99.44 |
| Shankar K et al. [39] | 2018 | Optimal Feature Level Fusion (OFLF) | Binary | 96.23 |
| **Proposed model** | **–** | **Two stage ensemble CNN model** | **Multi** | **99.76** |

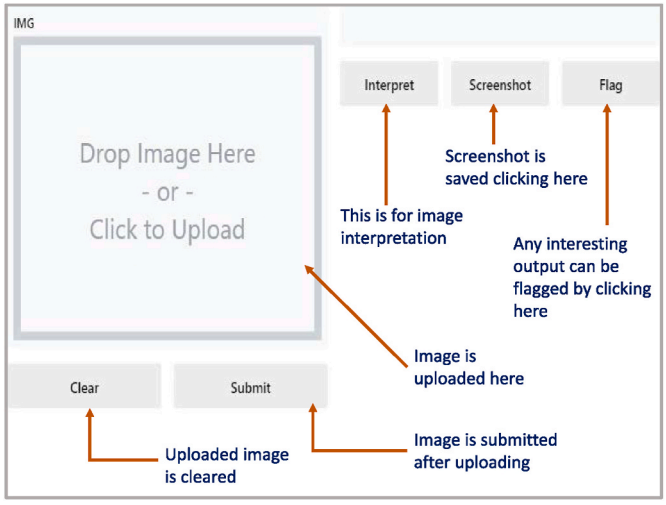


**Fig. 17.** User Interface created using python *Gradio* for evaluating the real time predicting capability of the proposed model.

### 3.8. Novelty and contribution of the study

The main contribution of this work is to utilize the two-stage ensemble approach in classifying brain tumor from brain MRI. In this particular task this approach is new. We have tried to apply this approach in order to acquire an improved performance. In several other studies, researchers have applied the fusion or ensemble techniques for brain tumor classification with different models. However, multi-stage or two-stage ensemble is a new approach in this particular task. Besides this ensemble approach, some other factors have contributed in improving model's performance, i.e., data augmentation and feature selection using PCA. Several experiments have been conducted in order to prove the proposed model's robustness. Firstly, with leave-one-out approach where model is trained with a particular dataset and validated with a different dataset, we have shown that the model's performance didn't degrade. Next with 8 fold cross validation and random splitting of training and validation data, it has been demonstrated that the model's performance remained consistent. Comparison with other state-of-the-art models also have been presented considering two different aspects, i.e., considering a particular dataset and considering leave-one-out approach. This comparative analysis with leave-one-out

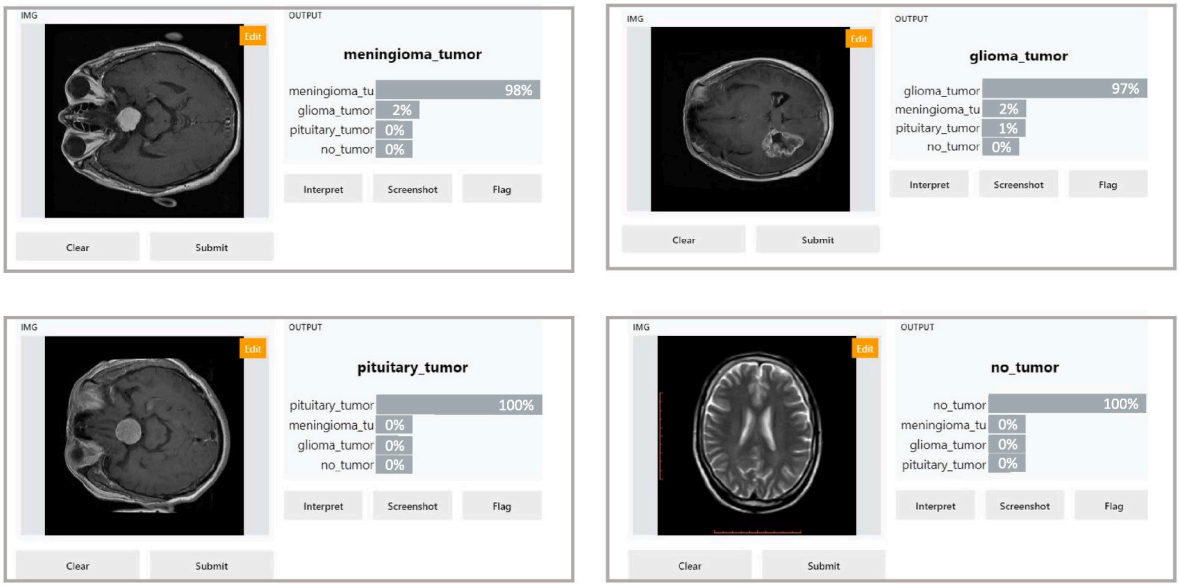


**Fig. 18.** Model validation by uploading some random images of different types of tumors, where it shows the percentage of confidence in predicting the class of a tumor.

approach is also a new addition to this type of study. Lastly, we have tried to validate our model with a new web based User Interface that proves the model's real time prediction capability.

## 4. Conclusion

The consequences of brain tumors can be very dreadful and life-threatening as they can cause cancer in the long term. To avoid the heinous effect, this paper proposes an automated classification system for early and precise diagnosis. To the best of our knowledge, it is the first time a two-stage ensemble of deep CNN models is used for categorizing three different types of tumors and normal brain cells. The whole process of two-stage ensemble and classification has been implemented by analyzing and scrutinizing the best CNN models and classifiers so that the final model becomes robust for precise classification. The impact of applying a two-stage ensemble, PCA and data augmentation shows the significant improvement in the model's performance in different aspects. As a result of applying two-stage ensemble, the overall accuracy increases 4.31%. After applying PCA, the average execution time and the number of features reduce by a factor of 6 and 18.71 respectively. Moreover, three experiments conducted on the datasets considering random splitting, 8-fold cross-validation, and different validation set also prove the robustness and generalization capability of the proposed model. Besides, the comparison with other state-of-the-art models shows the promising performance of the proposed model. Finally, the User Interface built on the proposed model verifies the real-time performance of the model.

Though the proposed model showed a prominent performance in classifying brain tumors from MRI, there were also some challenges in our study. There are around 120 types of brain tumors but because of data inadequacy, our work was limited to three major types of tumor. We only considered MRI because of its harmless nature but other medical imaging techniques, i.e., CT (Computed Tomography) scan, PET (Positron-Emission Tomography) can also be taken under consideration in future work. Moreover, this work can be extended considering patients' history (age, sex, physical condition etc.) along with the image dataset for more precise prediction. Because of having a high generalization capability, this work can be further extended for other classification problems in the medical field and can be considered as a handy tool for classification tasks in other areas as well.

## Declaration of competing interest

We do not have any conflict of interest.

## Acknowledgements

This research was supported by the Information and Communication Technology (ICT) division of the Government of the People's Republic of Bangladesh in 2021–2022.

## References

[1] Abdu Gumaei, Mohammad Mehedi Hassan, Md Rafiul Hassan, Abdulhameed Alelaiwi, Giancarlo Fortino, A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification, IEEE Access 7 (2019) 36266–36273.
[2] CancerNet, Brain tumor: statistics. https://www.cancer.net/cancer-types/brain-tumor/statistics, 2021. (Accessed 8 September 2021).
[3] Kavitha Angamuthu Rajasekaran, Chellamuthu Chinna Gounder, Advanced brain tumour segmentation from mri images, in: Basic Physical Principles and Clinical Applications, High-Resolution Neuroimaging, 2018, pp. 83–108.
[4] Tonmoy Hossain, Fairuz Shadmani Shishir, Mohsena Ashraf, MD Abdullah Al Nasim, Faisal Muhammad Shah, Brain tumor detection using convolutional neural network, in: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), IEEE, 2019, pp. 1–6.
[5] Nahid Ferdous Aurna, Tanjil Mostafa Rubel, Tanveer Ahmed Siddiqui, Tajbia Karim, Sabrina Saika, Md Murshedul Arifeen, Tasmima Noushiba Mahbub, SM Salim Reza, and Habibul Kabir. Time series analysis of electric energy consumption using autoregressive integrated moving average model and holt winters model. TELKOMNIKA (Telecommunication, Computing, Electronics and Control), 19(3), 2021.
[6] Nahid Ferdous Aurna, Faria Shahjahan Anika, Md Tanjil Mostafa Rubel, K Habibul Kabir, M Shamim Kaiser, Predicting periodic energy saving pattern of continuous iot based transmission data using machine learning model, in: 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), IEEE, 2021, pp. 428–433.
[7] Md Badrul Alam Miah, Mohammad Abu Yousuf, Detection of lung cancer from ct image using image processing and neural network, in: 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), ieee, 2015, pp. 1–6.
[8] Gopal S. Tandel, Antonella Balestrieri, Tanay Jujaray, Narender N. Khanna, Luca Saba, S Suri Jasjit, Multiclass magnetic resonance imaging brain tumor classification using artificial intelligence paradigm, Comput. Biol. Med. 122 (2020) 103804.
[9] Muhammad Sajjad, Salman Khan, Muhammad Khan, Wanqing Wu, Ullah Amin, Sung Wook Baik, Multi-grade brain tumor classification using deep cnn with extensive data augmentation, J. Comput. Sci. 30 (2019) 174–182.
[10] Ali Mohammad Alqudah, Hiam Alquraan, Isam Abu Qasmieh, Alqudah Amin, Wafaa Al-Sharu, Brain Tumor Classification Using Deep Learning Technique–A

Comparison between Cropped, Uncropped, and Segmented Lesion Images with Different Sizes, 2020 *arXiv preprint arXiv:2001.08844*.
[11] S. Deepak, P.M. Ameer, Brain tumor classification using deep cnn features via transfer learning, Comput. Biol. Med. 111 (2019) 103345.
[12] Pashaei Ali, Hedieh Sajedi, Niloofar Jazayeri, Brain tumor classification via convolutional neural network and extreme learning machines, in: 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE), IEEE, 2018, pp. 314–319.
[13] Emrah Irmak, Multi-classification of brain tumor mri images using deep convolutional neural network with fully optimized framework, Iran. J. Sci. Technol., Transac. Electr. Eng. (2021) 1–22.
[14] Narmada M Balasooriya, Ruwan D. Nawarathna, A sophisticated convolutional neural network model for brain tumor classification, in: 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), IEEE, 2017, pp. 1–5.
[15] Sunanda Das, OFM Riaz Rahman Aranya, Nishat Nayla Labiba, Brain tumor classification using convolutional neural network, in: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), IEEE, 2019, pp. 1–5.
[16] Parnian Afshar, Konstantinos N. Plataniotis, Arash Mohammadi, Capsule networks for brain tumor classification based on mri images and coarse tumor boundaries, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 1368–1372.
[17] D Jude Hemanth, J. Anitha, Antoanela Naaji, Oana Geman, Daniela Elena Popescu, et al., A modified deep convolutional neural network for abnormal brain image classification, IEEE Access 7 (2018) 4275–4283.
[18] Milica M. Badža, Marko Č. Barjaktarović, Classification of brain tumors from mri images using a convolutional neural network, Appl. Sci. 10 (6) (2020) 1999.
[19] Wadhah Ayadi, Wajdi Elhamzi, Imen Charfi, Mohamed Atri, Deep cnn for brain tumor classification, Neural Process. Lett. 53 (1) (2021) 671–700.
[20] Hossam H. Sultan, Nancy M. Salem, Walid Al-Atabany, Multi-classification of brain tumor images using deep neural network, IEEE Access 7 (2019) 69215–69225.
[21] S. Deepak, P.M. Ameer, Automated categorization of brain tumor from mri using cnn features and svm, J. Ambient Intell. Hum. Comput. 12 (8) (2021) 8357–8369.
[22] Sharan Kumar, Dattatreya P. Mankame, Optimization driven deep convolution neural network for brain tumor classification, Biocybern. Biomed. Eng. 40 (3) (2020) 1190–1204.
[23] Mesut Toğaçar, Burhan Ergen, Zafer Cömert, Brainmrnet: brain tumor detection using magnetic resonance images with a novel convolutional neural network model, Med. Hypotheses 134 (2020) 109531.
[24] Tonmoy Hossain, Fairuz Shadmani Shishir, Mohsena Ashraf, MD Abdullah Al Nasim, Faisal Muhammad Shah, Brain tumor detection using convolutional neural network, in: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), IEEE, 2019, pp. 1–6.
[25] Masoumeh Siar, Mohammad Teshnehlab, Brain tumor detection using deep neural network and machine learning algorithm, in: 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), IEEE, 2019, pp. 363–368.
[26] B. Srinivas, G. Sasibhushana Rao, A hybrid cnn-knn model for mri brain tumor classification, Int. J. Recent Technol. Eng. (IJRTE) 8 (2) (2019) 2277–3878.
[27] Isselmou Abd El Kader, Guizhi Xu, Shuai Zhang, Sani Saminu, Javaid Imran, Salim Ahmad Isah, Differential deep convolutional neural network model for brain tumor classification, Brain Sci. 11 (3) (2021) 352.
[28] R. Rajagopal, Glioma brain tumor detection and segmentation using weighting random forest classifier with optimized ant colony features, Int. J. Imag. Syst. Technol. 29 (3) (2019) 353–359.
[29] P. Rupa Ezhil Arasi, M. Suganthi, A clinical support system for brain tumor classification using soft computing techniques, J. Med. Syst. 43 (5) (2019) 1–11.
[30] S. Jayaprada, G. JayaLakshmi, L. KanyaKumari, Fast hybrid adaboost binary classifier for brain tumor classification, in: IOP Conference Series: Materials Science and Engineering vol. 1074, IOP Publishing, 2021, 012016.
[31] Minal Padlia, Jankiballabh Sharma, Fractional sobel filter based brain tumor detection and segmentation using statistical features and svm, in: Nanoelectronics, Circuits and Communication Systems, Springer, 2019, pp. 161–175.
[32] Sajid Iqbal, Muhammad U Ghani Khan, Tanzila Saba, Zahid Mehmood, Nadeem Javaid, Amjad Rehman, Rashid Abbasi, Deep learning model integrating features and novel classifiers fusion for brain tumor segmentation, Microsc. Res. Tech. 82 (8) (2019) 1302–1315.
[33] Muhammad Attique Khan, Ashraf Imran, Majed Alhaisoni, Robertas Damaševičius, Rafal Scherer, Amjad Rehman, Syed Ahmad Chan Bukhari, Multimodal brain tumor classification using deep learning and robust feature selection: a machine learning application for radiologists, Diagnostics 10 (8) (2020) 565.
[34] Neelum Noreen, Sellappan Palaniappan, Abdul Qayyum, Iftikhar Ahmad, Muhammad Imran, Muhammad Shoaib, A deep learning model based on concatenation approach for the diagnosis of brain tumor, IEEE Access 8 (2020) 55135–55144.
[35] Jainy Sachdeva, Vinod Kumar, Indra Gupta, Niranjan Khandelwal, Chirag Kamal Ahuja, A dual neural network ensemble approach for multiclass brain tumor classification, Int. J. Numer. Methods Biomed. Eng. 28 (11) (2012) 1107–1120.
[36] Jaeyong Kang, Zahid Ullah, Jeonghwan Gwak, Mri-based brain tumor classification using ensemble of deep features and machine learning classifiers, Sensors 21 (6) (2021) 2222.
[37] Javeria Amin, Muhammad Sharif, Mussarat Yasmin, Tanzila Saba, Muhammad Almas Anjum, Steven Lawrence Fernandes, A new approach for brain tumor segmentation and classification based on score level fusion using transfer learning, J. Med. Syst. 43 (11) (2019) 1–16.
[38] Javaria Amin, Muhammad Sharif, Mudassar Raza, Tanzila Saba, Muhammad Almas Anjum, Brain tumor detection using statistical and machine learning method, Comput. Methods Progr. Biomed. 177 (2019) 69–79.
[39] R.M. Vidhyavathi, Mohamed A. Elsoud, Majid Alkhambashi, Optimal Feature Level Fusion Based Anfis Classifier for Brain Mri Image Classification, 2018.
[40] Ranjeet Kaur, Doegar Amit, Gaurav Kumar Upadhyaya, An ensemble learning approach for brain tumor classification using mri, in: Soft Computing: Theories and Applications, Springer, 2022, pp. 645–656.
[41] Ivan Izonin, Roman Tkachenko, Khrystyna Zub, Pavlo Tkachenko, et al., A grnn-based approach towards prediction from small datasets in medical application, Procedia Comput. Sci. 184 (2021) 242–249.
[42] Toshi Sinha, Brijesh Verma, A novel method based on convolutional features with non-iterative learning for brain tumor classification, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, pp. 2799–2805.
[43] Abol Basher, Kyu Yeong Choi, Jang Jae Lee, Bumshik Lee, Byeong C. Kim, Kun Ho Lee, Ho Yub Jung, Hippocampus localization using a two-stage ensemble hough convolutional neural network, IEEE Access 7 (2019) 73436–73447.
[44] Jun Cheng, Wei Huang, Shuangliang Cao, Ru Yang, Wei Yang, Zhaoqiang Yun, Zhijian Wang, Qianjin Feng, Enhanced performance of brain tumor classification via tumor region augmentation and partition, PLoS One 10 (10) (2015), e0140381.
[45] Jun Cheng, Brain tumor dataset. https://figshare.com/articles/dataset/brain_tumor_dataset/1512427, 2017. (Accessed 4 April 2021).
[46] Sartaj Bhuvaji, Brain tumor classification (MRI). https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri, 2020. (Accessed 5 May 2021).
[47] MohamedMetwalySherif, Brain tumor dataset. https://www.kaggle.com/mohamedmetwalysherif/braintumordataset, 2020. (Accessed 25 May 2021).
[48] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2014 *arXiv preprint arXiv:1409.1556*.
[49] Mingxing Tan, Quoc Le, Efficientnet: rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
[50] Christian Szegedy, Vanhoucke Vincent, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[52] François Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.
[53] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Reed Scott, Dragomir Anguelov, Dumitru Erhan, Vanhoucke Vincent, Andrew Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[54] Rui Zeng, Jiasong Wu, Zhuhong Shao, Lotfi Senhadji, Huazhong Shu, Quaternion softmax classifier, Electron. Lett. 50 (25) (2014) 1929–1931.
[55] Mayank Arya Chandra, S.S. Bedi, Survey on svm and their application in image classification, Int. J. Inf. Technol. (2018) 1–11.
[56] Youqiang Zhang, Cao Guo, Xuesong Li, Bisheng Wang, Cascaded random forest for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 11 (4) (2018) 1082–1094.
[57] Yanhui Guo, Siming Han, Ying Li, Cuifen Zhang, Bai Yu, K-nearest neighbor combined with guided filter for hyperspectral image classification, Procedia Comput. Sci. 129 (2018) 159–165.
[58] Arman Zharmagambetov, Magzhan Gabidolla, Miguel A. Carreira-Perpinán, Improved multiclass adaboost for image classification: the role of tree optimization, in: 2021 IEEE International Conference on Image Processing (ICIP), IEEE, 2021, pp. 424–428.
[59] Abien Fred Agarap, Deep Learning Using Rectified Linear Units (Relu), 2018 *arXiv preprint arXiv:1803.08375*.
[60] Diederik P. Kingma, Ba Jimmy, Adam: a method for stochastic optimization, 2014 *arXiv preprint arXiv:1412.6980*.
[61] John Duchi, Elad Hazan, Yoram Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (7) (2011).
[62] Sebastian Ruder, An overview of gradient descent optimization algorithms, 2016 *arXiv preprint arXiv:1609.04747*.
[63] Daniel Fernandez, Carlos Gonzalez, Daniel Mozos, Sebastian Lopez, Fpga implementation of the principal component analysis algorithm for dimensionality reduction of hyperspectral images, J. Real Time Image Proc. 16 (5) (2019) 1395–1406.
[64] V.B. Shereena, M David Julie, Significance of dimensionality reduction in image processing, Signal Image Proc. Int. J. (SIPIJ) 6 (2015).
[65] Abubakar Abid, Abdalla Ali, Abid Ali, Dawood Khan, Abdulrahman Alfozan, James Zou, Gradio: hassle-free sharing and testing of ml models in the wild, 2019 *arXiv preprint arXiv:1906.02569*.
[66] Shorten Connor, Taghi M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48.
[67] Sanjay Yadav, Sanyam Shukla, Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification, in: 2016 IEEE 6th International Conference on Advanced Computing, IACC), 2016, pp. 78–83, https://doi.org/10.1109/IACC.2016.25.
[68] Amin Kabir Anaraki, Moosa Ayati, Foad Kazemi, Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms, Biocybern. Biomed. Eng. 39 (1) (2019) 63–74.

[69] R Lokesh Kumar, Jagadeesh Kakarla, B Venkateswarlu Isunuri, Munesh Singh, Multi-class brain tumor classification using residual network and global average pooling, Multimed. Tool. Appl. 80 (9) (2021) 13429–13438.
[70] Nyoman Abiwinanda, Muhammad Hanif, S Tafwida Hesaputra, Astri Handayani, Tati Rajab Mengko, Brain tumor classification using convolutional neural network, in: World Congress on Medical Physics and Biomedical Engineering 2018, Springer, 2019, pp. 183–189.
[71] Zar Nawab Khan Swati, Qinghua Zhao, Muhammad Kabir, Farman Ali, Zakir Ali, Saeed Ahmed, Jianfeng Lu, Brain tumor classification for mr images using transfer learning and fine-tuning, Comput. Med. Imag. Graph. 75 (2019) 34–46.