Iranian Journal of Science and Technology, Transactions of Electrical Engineering
https://doi.org/10.1007/s40998-024-00726-w

RESEARCH PAPER

A Fine-Tuned EfficientNet B1 Based Deep Transfer Learning Framework for Multiple Types of Brain Disorder Classification

Arpita Ghosh[1] · Badal Soni[1] · Ujwala Baruah[1]

Received: 29 September 2023 / Accepted: 1 April 2024

Abstract

Automated brain disorder classification for convenient treatment is one of the most complicated and widely spread problems. With the help of cutting-edge hardware, deep learning approaches are outperforming conventional brain disorder classification techniques in the medical image field. To solve this problem researchers have developed various transfer learning-based techniques. Pre-trained deep learning architectures are used here for feature extraction. This paper proposes a deep learning framework that includes a pre-trained fine-tuned EfficientNet B1 model to classify three different types of brain disorder and a normal category with 93% of test accuracy. In order to evaluate the proposed framework, the dataset was trained and validated using additional deep learning models Inception V3 and ResNet50 V2 for feature extraction using softmax and support vector machine (SVM) classifiers and employing three primary optimizers: stochastic gradient descent (SGD), root mean squared propagation (RMSProp), and Adam. The EfficientNet B1 with softmax classifier and Adam optimizer outperformed the other two state-of-the-art models and achieved the best results.

Keywords Brain disorder · Transfer learning · Inception V3 · ResNet50 V2 · EfficientNetB1 · Optimizer

1 Introduction

The human brain is a vital organ and is extremely complex in its structure. One of the most important organs for connecting every part of the human anatomy is the brain. Brain disorders are considered to be a serious disease in humans and have the potential to reduce overall brain function (https://www.cancer.net/cancer-types/brain-tumor/statistics). Multiple researches on signal and image analysis methods are being performed in order to learn more about this structure. Magnetic resonance imaging (MRI) is a convenient method in these studies (Deepak and Ameer 2021; Anaraki et al. 2019). MRI is a noninvasive and painless method for generating high-quality 2D and 3D images. Magnetic fields and radio waves are used to create MRI samples rather than radiation. In this regard, it differs from Computed Tomography scans (CT-scan) and X-ray images Haq et al. (2021). Brain MRI has significant symmetry when viewed from coronal and axial perspectives. The asymmetric structure indicated by axial images clearly indicate the existence of an abnormality. To detect abnormalities, a contrast substance or dye can be used Anaraki et al. (2019).

Machine learning and deep learning are two modern techniques that are being employed in a wide range of application areas Deepak and Ameer (2021), Rane et al. (2021). Many research initiatives are currently being carried out to advance the field of medical image processing, which has huge potential. This research area is also concerned with automating brain abnormalities classification. MRI with no radiation produces images with high contrast and resolution. It is therefore the most effective imaging technique for non-invasively diagnosing disease in patients with brain abnormalities Haq et al. (2021). Compared to biopsy specimens, computer-aided screening is quicker, easier, and may provide outcomes that are more reliable

✉ Arpita Ghosh
arpita5102_rs@cse.nits.ac.in

Badal Soni
badal@cse.nits.ac.in

Ujwala Baruah
ujwala@cse.nits.ac.in

[1] Computer Science and Engineering, National Institute of Technology, Silchar, NIT Road, Silchar, Assam 788010, India

Published online: 25 April 2024

and consistent Anaraki et al. (2019). Computer-aided devices and deep learning architectures can expedite the classification of brain abnormalities. The primary goal of these frameworks is to create a structure that will assist physicians Brima et al. (2021). As a result, an automatic and precise diagnosis is possible. Today, MRI-based brain image classification methods have been extensively studied in published research Kibriya et al. (2022). These studies indicate that deep learning architectures can be used to classify brain diseases.

The deep learning model consists of hidden layers, inputs, outputs, loss, activation functions, and so on O'Shea and Nash (2015). Any deep learning model uses an algorithm to generalise the data by making predictions based on previously unexplored data. We need both an optimisation algorithm along with a procedure that maps the instances of the input data to instances of output data Veni and Manjula (2022). An optimisation method specifies weights or factors whose values minimise errors generated by the input-to-output mapping. These optimizers or optimisation approaches have a significant impact on the deep learning model's efficiency (Polat and Güngen 2021; Poyraz et al. 2022). Optimizers can also affect the model's training speed. The weights for each epoch must be modified and the loss function must be decreased while training the deep learning model. An optimizer is a method or algorithm that modifies neural network parameters such as weights and learning rates Mehrotra et al. (2020). As a result, it contributes to lower overall loss and increased accuracy. It is challenging to decide on appropriate weights for deep learning models because they have millions of parameters. It emphasizes the necessity of specifying an optimisation algorithm that is suitable for a particular problem.

This study proposes a classification pipeline of brain MRI data for multiple types of brain disorders such as Atrophy, Ischemia, Normal, and WMI. The MRI dataset used in this work is publicly available in Kaggle TUNCER (2023). The classification was accomplished using fine-tuned deep-learning techniques. Fine-tuned transfer learning Inception V3, ResNet50 V2, and EfficientNet B1 have been used for feature extraction, and as a classifier softmax and SVM have been used, and performance comparison has been carried with three actively used optimizers SGD, RMSProp, and Adam. The primary contributions of this paper are given below:

- To address the classification of multiple types of brain disorders, three transfer learning-based fine-tuned deep feature extraction models, ResNet50 V2, Inception V3, and EfficientNet B1, were used and their performance was compared.
- In the fine-tuned architectures, the feature map produced by the final layer of deep networks has been generalized using a global average pooling layer.
- Comparison of the performance has been explored by two classifiers softmax and SVM.
- Three popularly used optimization techniques Stochastic Gradient Descent (SGD), RMSProp, and Adam with different learning rates (0.1 and 0.001) have been used for the experimental analysis.
- The effectiveness of the proposed pipeline has been compared with previously published brain disease classification state-of-the-art works.

The research article is organised as follows: Sect. 2 highlights some existing related work, Sect. 3 addresses the background study, Sect. 4 describes the proposed methodology, Sect. 5 presents the results, analysis, and discussions, the conclusion and future scope are discussed in Sect. 6.

2 Related Works

This section provides information about existing related works using various deep learning and machine learning techniques. Using a modified AlexNet model, Khawaldeh et al. (2017), addressed the classification problem among healthy, low-grade, and high-grade tumors on the TCIA dataset. Their proposed method achieved 91.16% accuracy. Different types of tumor grade, classification system was developed by Sajjad et al. (2019), using cascaded CNN and VGG 19 architecture on Radiopeadia, brain tumor data set and their proposed approach improved performance on both the data set. The dataset that has 121 numbers of MRI samples with four tumor grading. The dataset has less number of samples to train a heavyweight CNN model VGG-19 which has 144 million parameters for feature extraction that is computationally costly and time-consuming. A fine-tuned GoogleNet-based classification technique has been proposed by Deepak and Ameer (2019), to classify different types of brain tumors using the Figshare dataset and achieved 98% classification accuracy. However, their model suffers from overfitting issues when they use less data for training.

Alyami et al. (2023), proposed a framework for tumor localization and classification that includes AlexNet and VGG19 concatenated features, a slap swarm algorithm for optimal feature selection, and various SVM kernels for classification. Using a cubic SVM kernel, the proposed framework achieved 99.1% test accuracy on a binary brain tumor data set. However, the proposed method has not been validated on multiple types of brain tumors. Deepa et al. (2023), proposed a hybrid optimization algorithm for brain tumor segmentation and classification that incorporates the Jaya algorithm, honey badger algorithm (HBA), and the chronological theory on a deep residual network. The proposed work is only validated for binary classification. Usmani et al. (2023), examined the impact of hyperparameters and their interconnections on residual deep models using a cartesian product matrix-based approach. The effects have been evaluated by three well-known optimizers: Adam, root mean squared propagation (RMSProp), and stochastic gradient descent with momentum (SGDM). Other optimizers can be used to achieve better performance.

To solve the binary brain tumor classification problem Mehrotra et al. (2020), trained AlexNet, ResNet50, ResNet101, GoogleNet, and SqueezeNet on the cancer imaging archive (TCIA) brain tumor MRI data using different optimizers and achieved the highest performance using AlexNet with SGDM. Two convolution neural network (CNN) models were developed by Haq et al. (2021), to classify various tumor grades on the multimodal brain tumor image segmentation benchmark (BraTS) dataset and different brain tumor types on the Figshare dataset and achieved improved results. A hard swish rectified linear unit (ReLU) activation function was incorporated with the CNN model by Alhassan and Zainon (2021), to classify multiple brain tumor types. Thirteen-layer convolution network was proposed by Kibriya et al. (2022), to solve multiple brain tumor classes, and a 97.2% classification accuracy rate was achieved. Using the transfer learning approach Aurna et al. (2022), proposed a two-stage feature ensemble method on three brain tumor data sets along with a combination of three data sets and promising performance attained by their proposed approach. The binary brain tumor classification issue was solved by Alyami et al. (2023), using a framework that includes the slap swarm algorithm feature selection method. The feature was selected from AlexNet and VGG19 concatenated features. Analysis of different SVM kernels has been reported and cubic SVM achieved the highest classification accuracy among them. From the above-studied literature, we can conclude that further research study is required to classify multiple types of brain disorders, and using a fine-tuned transfer learning approach will allow us to investigate this classification problem.

The existing literature emphasizes the critical significance of accurate classification of brain disorders using MRI datasets. Timely and precise diagnoses play a pivotal role in effective treatment planning and patient management. A significant portion of the literature focuses on the emergence and effectiveness of deep learning models like various CNNs. CNN shows remarkable capabilities in automatically learning complex hierarchical features from MRI data. Some studies used transfer learning, utilizing pre-trained models on large-scale datasets to extract features. This approach allows for improved generalization and performance, especially when dealing with limited data. So, there is scope for utilizing the transfer learning approaches and exploring the SVM classification layer combined with the transfer learning architecture. This study explores three extensively popular transfer learning models Inception V3, ResNet50 V2, and EfficientNet50 V2 for feature extraction and softmax, SVM for classification.

3 Background Study

This section provides information about the background analysis of the dataset used in the experiments and different deep learning concepts applied in this work.

3.1 Dataset Description

The brain disorder dataset was obtained from the Department of Radiology of Firat University Hospital Poyraz et al. (2022). This corpus contains 444 transverse T2 weighted magnetic resonance (MR) scans of three different brain disorders and a normal class. Atrophy, ischemia, and white matter intensity (WMI) are the different brain disorder classes. The number of images in each of the categories is given in Table 1. These images are saved as JPG files from the PACS of the Firat University Hospital. The MRI samples of the corpus are also shown in Fig. 1.

3.2 Transfer Learning

In contrast to conventional machine learning, which learns features from scratch, the transfer learning theory includes new learning tasks with earlier extracted features from a large data set Weiss et al. (2016). In this case, feature learning happens much faster compared to traditional machine learning tasks. Transfer learning is an important deep learning technique due to its ability to achieve better results with much less data (Polat and Güngen 2021; Usmani et al. 2023). When there is a scarcity of labeled training data, the traditional machine learning framework struggles to achieve state-of-the-art performance. Transfer learning strategy can manage small training data; the

Table 1 Details of Brain Disorder dataset Poyraz et al. (2022)

Disorder type	No. of samples
Atrophy	100
Ischemia	102
Normal	150
WMI	92

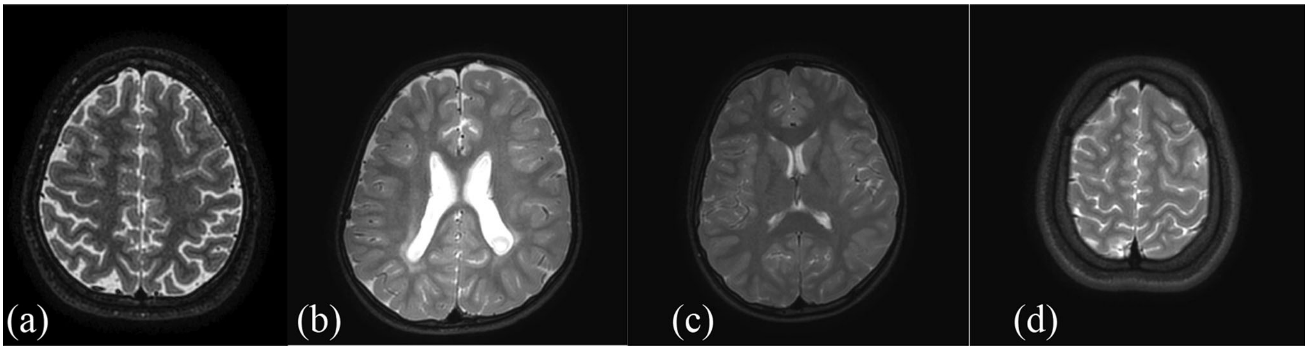


Fig. 1 Sample of collected MRI brain tumor slices. **a** Atrophy; **b** Ischemia; **c** Normal; **d** WMI

benefit is that features from previously labeled information gathered from certain relevant tasks can be used Ghosh et al. (2022). Transfer learning involves leveraging the knowledge gained by a pre-trained model on a large dataset (often from a related task) and fine-tuning it on a smaller dataset for a specific task of interest. The early layers of a pre-trained CNN are excellent at capturing low-level features such as edges, textures, and shapes. These layers are typically frozen during the fine-tuning process, and their outputs are treated as high-level features. In the case of CNNs, this typically involves using the pre-trained layers as feature extractors and adding a few additional layers for task-specific classification. Feature extraction techniques based on transfer learning include,

- Train a classifier, on a pre-trained feature extraction model.
- Fine-tune the pre-trained network while preserving the learned weight values as initial parameters.

The weights generated by training on the ImageNet data set contain features that can aid in identifying shapes, edges, and various other critical elements required for image classification. The ImageNet data was utilised for training Inception V3, ResNet50 V2, and EfficientNet B1 base model on over 14 million images from 1000 various categories Tan and Le (2019). Because our chosen task could not be completed using baseline transfer learning architectures, fine-tuning was required. Figure 2 depicts the transfer learning framework. In this work, three transfer learning model ResNet50 V2, Inception V3, and EfficientNet B1 has been used for brain disorder classification.

3.3 Optimizer

The goal of optimisation techniques in neural networks is to minimize the loss function, defined as the difference between the data that is predicted and the expected output values. In this experiment, three commonly used optimizers SGD, RMSProp, and Adam are discussed.

1. Stochastic gradient descent (SGD) uses a subset of the samples for each iteration, instead of using the entire data set Ruder (2016). In each iteration, the parameters are updated in the direction of the negative gradient of the loss function with respect to the parameters. It uses a fixed learning rate for all parameters. Equation 1 defines the SGD optimizer update rule, where η is the learning rate and θ is the model parameter function.

$$\theta_{i+1} = \theta_i - \eta \frac{\partial \theta}{\partial \theta_i} \tag{1}$$

2. RMSProp is another prevalent gradient-based optimisation algorithm for training neural networks Ruder (2016). RMSProp adapts the learning rate for each parameter by dividing it by the square root of the exponentially weighted moving average of the squared gradients. RMSProp was designed as a stochastic mini-batch learning method. It tries to resolve the problems by normalising the gradient using the moving average of squared gradients. The update rule of RMSProp is mentioned in Eq. 2.

$$\theta_{j+1} = \theta - \eta \frac{\partial \theta}{\partial \sqrt{\upsilon} + \varepsilon} \tag{2}$$

In Eq. 2, υ is the bias-corrected weight parameter, η is the learning rate and ϵ is a small positive constant 10^{-8} used to avoid the error.

3. The working principle of Adam optimizer is a combination of the "Gradient descent with momentum" and "RMSProp" algorithms Ruder (2016). It maintains both the exponentially decaying average of past gradients and their squared values. Equation 3 provides the updated equation for the Adam optimizer.

$$\theta_{i+1} = \theta_i - \frac{\eta}{\sqrt{\widehat{\upsilon}} + \varepsilon} \widehat{m}_t \tag{3}$$

where, θ is the weight vector, $\widehat{m}_t$ and $\widehat{\upsilon}$ are the bias-corrected first and second weight parameters.

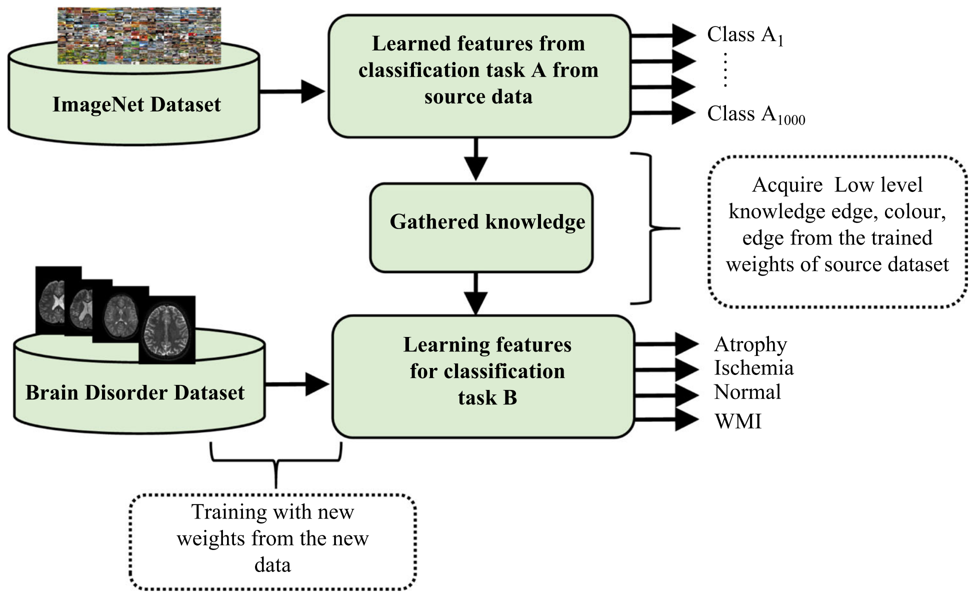


Fig. 2 The transfer learning working principle

4 Proposed Methodology

The proposed method includes several general steps for investigating the correctness of various types of brain disorders. Figure 3 depicts the proposed workflow, which includes data pre-processing, augmentation, feature extraction, and classification.

4.1 Problem Definition

In this study, we can consider a brain disorder dataset $D = \{A_i, b_i\}_{i=1}^{p}$ with p labeled image. The i^{th} sample image of the dataset D can be represented as $A_i \in S^{h \times w \times c}$ where c is considered as number of channels and $h \times w$ is the height and width of MRI image sample. The corresponding classification classes are b_i where 0: Atrophy, 1: Ischemia, 2: Normal and 4: WMI. The aim of the proposed system is to predict b_i for each A_i.

4.2 Data Pre-processing

The primary goal of image pre-processing is to prepare the MRI samples for reading by the model and processing it for better analysis. The MRI samples are re-sized to "224×224" as working with a larger size will require more parameters to handle in order to examine the entire training set in a consistent manner. The neural network can process the inputs with small weight values faster than inputs with large integer values, which can slow down the learning process. As a result, the pixel values in the [0-1]

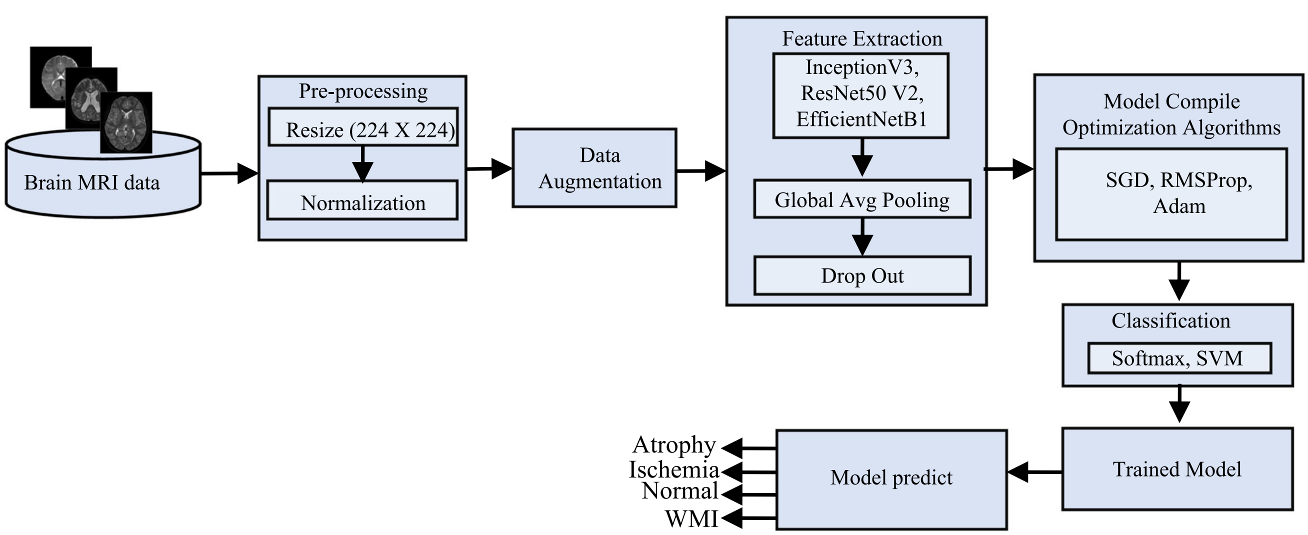


Fig. 3 Proposed workflow of the brain disorder classification framework

range have been standardised using the min-max method. The min-max normalisation formula is given in Eq. 4.

$$\chi_i = \frac{\alpha_i - min(\alpha)}{max(\alpha) - min(\alpha)} \tag{4}$$

The normalized pixel value is χ_i where the maximum and minimum pixel values are defined as $max(\alpha)$ and $min(\alpha)$ respectively.

4.3 Data Augmentation

Image augmentation is the process of applying different transformations to an original image, resulting in multiple replicas of that image. While using simple geometric transformations such as rotation shift various augmented data have been used to change the width/height range, shear and zoom range, horizontal/vertical flip, and brightness change. Keras ImageDataGenerator is used in real-time data augmentation for our work. We have iterated over the data in batches when using the Keras image data generator module. Table 2 summarises the augmentation parameters and Fig. 4 provides the visualization of augmented output.

4.4 Network Training

This section contains details about network training and fine-tuning. Initially from the Keras library ImageNet pre-trained base deep learning models Inception V3, ResNet50 V2, and EfficientNet B1 have been imported. The base model was able to instantly improve its image recognition performance by employing pre-trained weights from ImageNet features. CNN with feed-forward training strategies is used in the pre-trained models. Equation 5 describes the network training process from the input layer to the classification stage. As soon as one pass is finished, error backpropagation starts from the classification phase to the initial input layer. Equation 5 demonstrates the information flow from a neuron in the previous layer to a neuron in the subsequent one, where W stands for the weight of the connections between the two neurons.

Table 2 Data augmentation parameters

Transformations	Values
Rotation range	25
Width shift range	0.2
Height shift range	0.2
Zoom range	0.2
Horizontal flips	True
Vertical flips	True
Brightness range	[0.2, 0.8]

$$Input_p^l = \sum_{q=1}^{n} W_{pq}^l a_q + bias_p \tag{5}$$

Following this phase, fine-tuning was required because the base deep learning structure was unable to accomplish our classification problem. As a result, feature extraction by the final convolution layer generates a feature map that was generalised by using the global average pooling over flattening layers. It averages each generated feature and feeds the resulting vector into the classification layer. One advantage of global average pooling over fully connected layers is that it enforces correlations between feature maps and categories, which makes it more similar to the convolution structure Lin et al. (2013). The over-fitting problem is reduced by adding a drop-out layer subsequent to the global average pooling layer.

For a robust model, choosing the hyperparameter's optimum value is an essential task. The hyperparameter values have a significant impact on the model training process's ability to rapidly converge to a local minima using the most effective transfer learning-based features extraction strategy and classification methods. During the training of the model, a large number of hyper-parameter values were arbitrarily tested with the data set. The train, validation, and test ratios, epochs, drop-out rate, learning rate, loss, batch size, and optimization algorithm have all been taken into consideration. Each time, the dropout rate, the batch size, and the number of epoch parameters are changed, beginning with the lowest dropout rate (0.1), the lowest epochs (10), and the largest batch size (256), till the desired result is obtained. The hyper-parameter values are given in Table 3. For the optimizers the default learning rate has been fixed and given in Table 4.

4.5 Proposed Model

The proposed model includes EfficientNet B1 architecture with a global average pooling layer. The EfficientNet architecture has been developed by Tan and Le (2019) using a compound coefficient to scale the convolution neural network structure consistently in all dimensions of depth, width, and resolution. This convolution neural network network scaling strategy improved the architecture's performance. As a deep learning architecture, the design of the EfficientNet model demonstrates a remarkable balance between model size, computational resources, and accuracy. EfficientNet B1 architecture consists of a baseline network that incorporates inverted bottleneck blocks, similar to MobileNet V2, as well as depth-wise separable convolutions Tan and Le (2019). These components help reduce the number of parameters in the network while retaining its ability to capture complex features. The

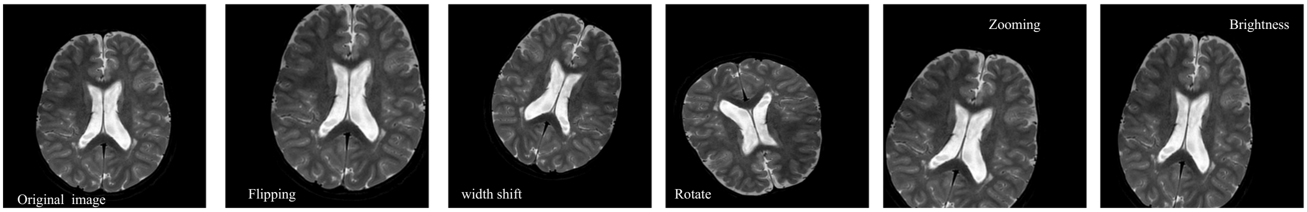


Fig. 4 The sample images after performing data augmentation

modification on EfficientNet B1 is fine-tuning of the network. In the state-of-the-art EfficientNetB1 network, a flatten layer is at the end of the final convolution layer in the feature extraction module and 1000 dense nodes with a softmax classification layer are used. However, in the proposed fine-tuned EfficientNet B1 the global average pooling layer has been included for generalization of the generated feature map after the feature extraction from the final convolution layer of the EfficientNetB1. Additionally, instead of using max pooling or other forms of pooling with fixed-size pooling windows, in the proposed network a global average pooling is used which takes the average of all values in each feature map, resulting in a single value for each channel. In the classification module, 4 dense nodes with softmax and SVM layer have been used and the performance is compared for both classification functions.

The proposed algorithm of the framework of the model training has been stated in Algorithm 1.

4.5.1 Working Principles of Global Average Pooling

In the proposed EfficientNet B1 architecture, the Global average pooling (GAP) layer commonly used in as a method for reducing the spatial dimensions of feature maps while retaining important information. The primary purpose of a global average pooling layer is to condensing the information in the feature maps into a single value per feature map channel. This process helps in reducing the number of parameters in the network and aids in preventing over fitting. By computing the average value of each feature map channel, the spatial dimensions are collapsed into a single value. This reduces the overall number of parameters in the subsequent layers, making the network more computationally efficient. The final feature map generated by GAP layer are less sensitive to the precise location of features within the input data. So, by averaging feature value across the entire feature map, GAP ensures that the presence of important features is captured irrespective of their spatial position.

Algorithm 1 Finetuned EfficientNetB1

```
Input Image
Pre-process the image data
Split the data into train : test : val = 0.70 : 0.10 : 0.20
Define Finetuned EfficientNetB1()
for Each step epoch = 1,2,... n do
    input = Image tensor batch
    model = EfficientNetB1 Feature Map
    for Each layer = 1,2,... l do
        model ← GlobalAveragePooling2D()
        model ← Dropout(rate = 0.3)
        Output ← Dense(4, activation = 'softmax')
    end for
    Train_data //Hyper-parameter initialization
end for
```

4.6 Classification

The brain disorder classification probability will be computed after training networks and feature extraction by the classification layer. Apart from accuracy, the minimization of the loss function is highly beneficial in improving the efficiency of a model used for classification. As a result, the loss during back-propagation should be minimised. The SVM and softmax classification layers have been used to classify four types of brain disorders. The working

Table 3 Hyper-parameter values for model training

Hyper-parameter	Value
Classifier	Softmax, SVM
Loss	Categorical cross-entropy, Squared hinge loss
Dropout	0.3
Batch size	16
No. of epochs	60
Early stopping patience	2
monitor	val_accuracy
Train: Val: Test	70 : 10 : 20

Table 4 Optimization algorithms and corresponding learning rate value for model training

Optimizer	Learning rate
SGD	0.01(default), 0.001
RMSProp	0.001(default), 0.01
Adam	0.001(default), 0.01

principle of both the classification function has been given below:

1. The softmax classifier is a widely used classification function for multi-class classification problems Nwankpa et al. (2018). It is a mathematical function that converts a vector of real numbers into a probability distribution. This distribution assigns probabilities to each class in a way that the sum of probabilities adds up to 1. The loss function used here is the categorical cross-entropy used by the softmax classifier, as stated in Eq. 6. The softmax function value as well as the truth value of the x^{th} point are represented by $f(s_x)$ and t_x, respectively.

$$\begin{aligned} Loss_{CE} &= -\sum_{x}^{c} t_x log(f(s_x)) \\ f(s_x) &= \frac{e^{s_x}}{\sum_{y}^{C} e^{s_y}} \end{aligned} \tag{6}$$

2. SVM was selected for our study as a result of previous work Deepak and Ameer (2021) in which SVM showed effective performance in classifying complex features. The combination of SVM classifiers with transfer learning-based CNN models is a powerful technique for tasks in computer vision. It leverages the strengths of both approaches to achieve high accuracy, particularly when labeled data is limited. SVM needs to identify the best hyperplane for dividing classes in a dataset Cortes and Vapnik (1995). Equation 7 states the Squared Hinge Loss as a loss function. This loss function is used for "maximum margin" classification problems.

$$Loss(p, \hat{p}) = \sum_{i=0}^{N} \left(\max(0, 1 - p_i \cdot \hat{p}_i)^2 \right) \tag{7}$$

The squared hinge loss value will be 0 when the truth and predicted values are equal, $\hat{p} \geqslant 1$ and it will increase in a quadratic way with the error if the truth and predicted values are not equal, $\hat{p} \leqslant 1$.

Table 5 Confusion Matrix Format

	Predicted true	Predicted false
Actual true	True positive(TP)	False positive(FP)
Actual false	False negative(FN)	True negative(TN)

5 Results and Discussion

This section contains in-depth details regarding the experimental results based on optimizers, as well as an analysis of the deep learning architectures using SVM and softmax layers. All the experiments are performed on an Intel(R) Xeon(R) Silver 4108 CPU @ 1.80 processor, 64 GB RAM, and 8 GB GPU.

5.1 Evaluation Measures

The evaluation measures used for the experiments are based on the confusion matrix as shown in Table 5. Table 6 describes the mathematical equation for the evaluation parameters precision, f1-score, recall, and accuracy. In a tabular format, the confusion matrix contains both the correct and incorrect classification.

5.2 Results

In this work, transfer learning-based fine-tuned deep learning models Inception V3, ResNet50 V2, and

EfficientNet B1 are used for feature extraction. The optimization techniques SGD, RMSProp, and Adam are used for training the deep learning models, and the last layer of the proposed model includes softmax and SVM classifiers with four neurons as the data has four types of brain disorder classes. The SVM classifier's kernel regularizer is set to $l2(0.01)$. We used 70% of the total data set for training in this work, with 16 batch sizes, so the model took 20 iterations to complete each epoch. Tables 7, 8 and 9 shows the performance of each optimisation algorithm using softmax classifier, and Tables 10, 11 and 12 shows the performance of each optimisation algorithm using SVM classifier. The confusion matrices of the proposed framework are provided in Figs. 5, 6, and 7. The training-validation accuracy/loss graphs of the EfficientNet B1 model are given in 8, 9 and 10. The average precision, recall, and F1-score of the model is provided in Fig. 13. Figure 11 shows the receiver operating characteristic (ROC) curve of the proposed framework. The comparison of the validation accuracy of EfficientNet B1 with SGD, RMSProp, and Adam optimizer using softmax and SVM is provided in Fig. 14. The visualization of the proposed framework is given in Fig. 15. The optimized EfficientNet B1 in the suggested framework has been trained and tested with varying batch sizes, ranging from the maximum 256 to 16. The best results were seen at batch size 16. The batch size results are displayed in Fig. 11. The larger batches require more memory for both model parameters and intermediate activations during backpropagation. However, smaller batches consume less memory, allowing for training complex models with limited memory resources. Sometimes, larger batches may provide a more accurate estimate of the gradient of the loss function, but they might lead to a more deterministic update, potentially causing the model to converge to a sharp minimum and overfit and smaller batches introduce more stochasticity in the optimization process, which can help the model generalize better and find more robust solutions. We have experimented with different learning rates for SGD (0.001), RMSProp (0.01), and Adam (0.01), in addition to changing the default value of optimizers. The outcomes are given in Fig. 15.

Table 6 Equation of evaluation parameters

Metrics	Equation
Precision	$\frac{(TP)}{(TP+FP)}$
Recall	$\frac{(TP)}{(TP+FN)}$
F1-score	$\frac{2\times(P\times R)}{(P+R)}$
Accuracy	$\frac{(TP+TN)}{(TP+TN+FP+FN)}$

5.3 Ablation Analysis

This section presents the implications based on optimizer behaviors on the test set, as well as its evaluation employing softmax and SVM classifiers.

5.3.1 Effect of Optimizers

SGD, RMSProp, and Adam are popular optimization algorithms used in training neural networks. Each of these optimizers has distinct characteristics that affect the training process and performance of the model. SGD is straightforward and computationally efficient. The path to the minimum can be unpredictable due to its random nature (stochastic), which can lead to more rapid convergence in some cases but also makes it susceptible to getting stuck in local minima. The learning rate can impact the performance of SGD as a very high learning rate may lead to oscillation and a very smaller rate may result in slow convergence. RMSProp adjusts the learning rates individually for each parameter, which can lead to faster convergence. It helps to smooth out the oscillations in the parameter updates, making convergence more stable. Adam adapts the learning rates individually for each parameter and includes a momentum term, making it particularly effective for complex high-dimensional problems.

From the reported results in Tables 7, 8 and 9 we can see that using softmax classifier the performance of each optimizer with each deep learning model achieved more than 80% accuracy. The training and validation accuracy for ResNet50 V2 of each optimizer using softmax increased till 10*th* epoch and then it achieved a stable position. For EfficientNet B1 using softmax function and SGD optimizer the training and validation accuracy increased till 20*th* epoch and then it got reduced after that again it increased from 40*th* epoch and then got a stable position. The accuracy of EfficientNet B1 for Adam and RMSProp optimizer increased till 10*th* epoch and then achieved a stable position. For Inception V3, the accuracy achieved the highest position for each optimizer after 10*th* epoch and after that it got stable. For each deep learning model and Adam optimizer we can see the convergence of both training and validation accuracy was better compared to the other two optimizer.

5.3.2 Effect of Classifier

SVM and softmax are typically equivalent in practice. Researchers will have different opinions about which classification technique performs better, but there is typically not much of a performance difference between the two. The more significant local objectivity of SVM

Table 7 Performance analysis of transfer learning architectures on SGD optimizer using softmax classifier

Architecture	class	Precision	Recall	F1-Score	Accuracy
ResNet 50V2	Atrophy	0.96	0.96	0.96	89%
	Ischemia	0.95	0.86	0.90	
	Normal	0.78	0.93	0.85	
	WMI	0.93	0.76	0.84	
Inception V3	Atrophy	0.84	0.91	0.87	88%
	Ischemia	0.87	0.91	0.89	
	Normal	0.86	0.93	0.89	
	WMI	1.00	0.71	0.83	
EfficientNet B1	Atrophy	0.88	0.96	0.92	83%
	Ischemia	0.89	0.77	0.83	
	Normal	0.77	0.85	0.81	
	WMI	0.80	0.71	0.75	

Table 8 Performance analysis of transfer learning architectures on RMSProp optimizer using softmax classifier

Architecture	Class	Precision	Recall	F1-Score	Accuracy
ResNet 50V2	Atrophy	0.86	0.78	0.82	81%
	Ischemia	0.77	0.77	0.77	
	Normal	0.78	0.93	0.85	
	WMI	0.86	0.71	0.77	
Inception V3	Atrophy	0.88	0.91	0.89	87%
	Ischemia	0.89	0.77	0.83	
	Normal	0.79	1.00	0.89	
	WMI	1.00	0.71	0.83	
EfficientNet B1	Atrophy	0.91	0.91	0.91	85%
	Ischemia	0.86	0.86	0.86	
	Normal	0.77	0.89	0.83	
	WMI	0.92	0.71	0.80	

Table 9 Performance analysis of transfer learning architectures on Adam optimizer using softmax classifier

Architecture	Class	Precision	Recall	F1-Score	Accuracy
ResNet 50V2	Atrophy	0.95	0.78	0.86	87%
	Ischemia	0.85	1.00	0.92	
	Normal	0.81	0.93	0.86	
	WMI	0.92	0.71	0.80	
Inception V3	Atrophy	0.84	0.91	0.87	87%
	Ischemia	0.87	0.91	0.89	
	Normal	0.86	0.89	0.87	
	WMI	0.92	0.71	0.80	
EfficientNet B1	Atrophy	0.90	1.00	0.95	93%
	Ischemia	1.00	0.82	0.90	
	Normal	0.92	0.96	0.94	
	WMI	0.95	0.90	0.93	

compared to the softmax classifier can be considered both an advantage and a drawback. This classifier works by finding the optimal hyperplane that best separates the classes. It aims to maximize the margin between classes while minimizing classification errors. However, the SVM is satisfied once the margins have been met and does not actively handle accurate scores afterward. After the feature extraction from transfer learning models, the parameters of

Table 10 Performance analysis of transfer learning architectures on SGD optimizer using SVM classifier

Architecture	Class	Precision	Recall	F1-Score	Accuracy
ResNet 50V2	Atrophy	0.83	0.87	0.85	82%
	Ischemia	0.94	0.77	0.85	
	Normal	0.70	0.96	0.81	
	WMI	1.00	0.59	0.74	
Inception V3	Atrophy	0.90	0.78	0.84	83%
	Ischemia	0.79	0.86	0.83	
	Normal	0.76	0.93	0.83	
	WMI	1.00	0.71	0.83	
EfficientNet B1	Atrophy	0.88	0.61	0.72	67%
	Ischemia	0.85	0.60	0.63	
	Normal	0.57	0.96	0.71	
	WMI	0.64	0.53	0.58	

Table 11 Performance analysis of transfer learning architectures on RMSProp optimizer using SVM classifier

Architecture	Class	Precision	Recall	F1-Score	Accuracy
ResNet 50V2	Atrophy	0.50	0.26	0.34	52%
	Ischemia	0.67	0.45	0.54	
	Normal	0.47	0.89	0.62	
	WMI	0.55	0.35	0.43	
Inception V3	Atrophy	0.86	0.78	0.82	81%
	Ischemia	0.76	0.86	0.81	
	Normal	0.81	0.93	0.86	
	WMI	0.83	0.59	0.69	
EfficientNet B1	Atrophy	0.81	0.91	0.86	83%
	Ischemia	0.88	0.68	0.77	
	Normal	0.77	1.00	0.87	
	WMI	1.00	0.65	0.79	

Table 12 Performance analysis of transfer learning architectures on Adam optimizer using SVM classifier

Architecture	Class	Precision	Recall	F1-Score	Accuracy
ResNet 50V2	Atrophy	0.95	0.87	0.91	81%
	Ischemia	0.84	0.73	0.78	
	Normal	0.71	0.89	0.79	
	WMI	0.80	0.71	0.75	
Inception V3	Atrophy	0.95	0.78	0.86	76%
	Ischemia	0.62	0.82	0.71	
	Normal	0.79	0.85	0.82	
	WMI	0.75	0.53	0.62	
EfficientNet B1	Atrophy	0.80	0.87	0.83	82%
	Ischemia	0.94	0.68	0.79	
	Normal	0.74	0.96	0.84	
	WMI	0.92	0.71	0.80	

the SVM classifier are fine-tuned using the training data. This step involves adjusting the weights and biases to better fit the specific classification problem. Transfer learning allows the model to leverage knowledge from a broader dataset. The SVM then refines these features for the specific task, which can lead to better generalization. SVMs provide more interpretable results compared to some deep learning models. The decision boundary is determined

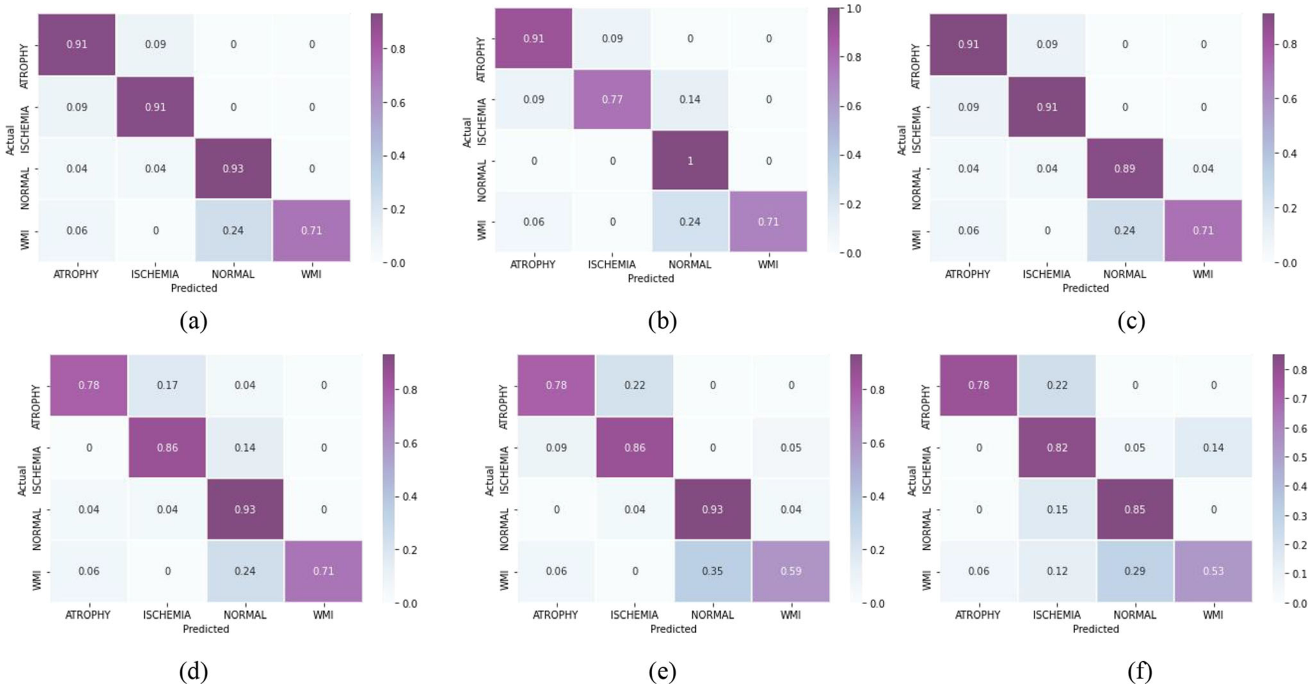


Fig. 5 Confusion matrix of the proposed method using Inception V3 architecture. **a** SGD+softmax, **b** RMSProp+softmax, **c** Adam+softmax, **d** SGD+SVM, **e** RMSProp+SVM, **f** Adam+SVM

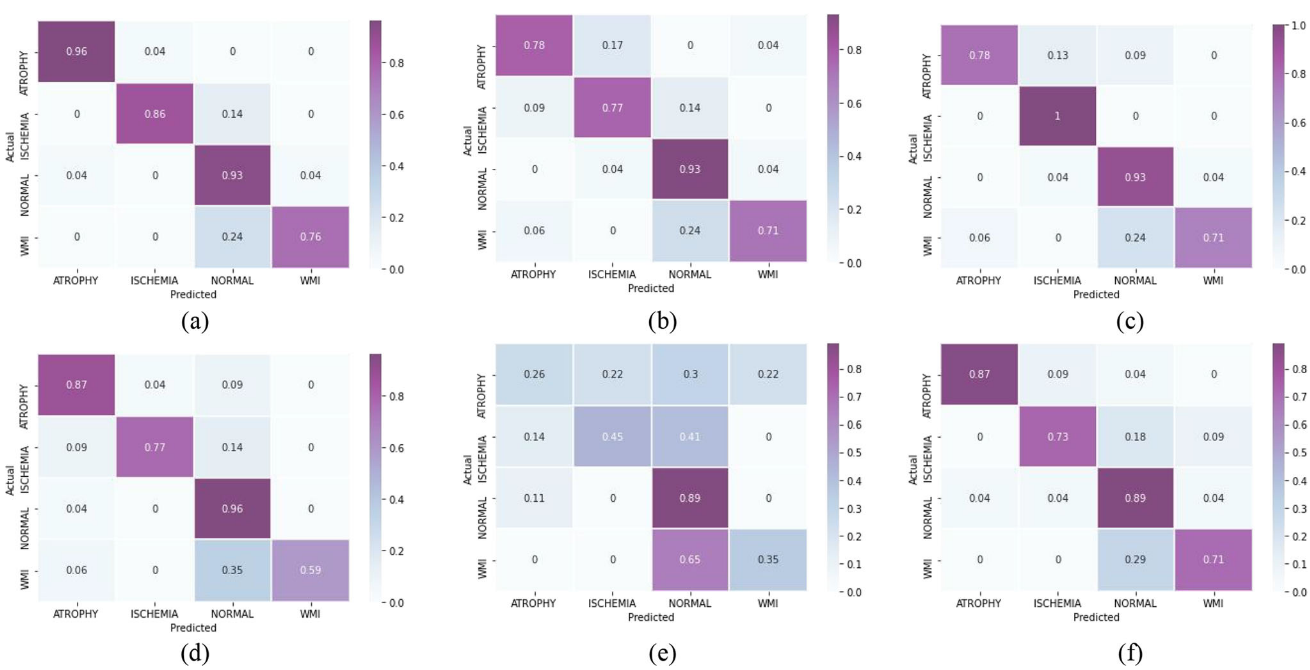


Fig. 6 Confusion matrix of the proposed method using ResNet50 V2 architecture. **a** SGD+softmax, **b** RMSProp+softmax, **c** Adam+softmax, **d** SGD+SVM, **e** RMSProp+SVM, (f) Adam+SVM

by a subset of the training data called support vectors. However, it's worth noting that the effectiveness of this approach can vary depending on the specific task and dataset. The softmax classification layer in transfer learning models has been used to solve a multi class classification problem. The advantage of softmax classification is that it provides a smooth, differentiable transition from assigning high probability to one class to assigning lower

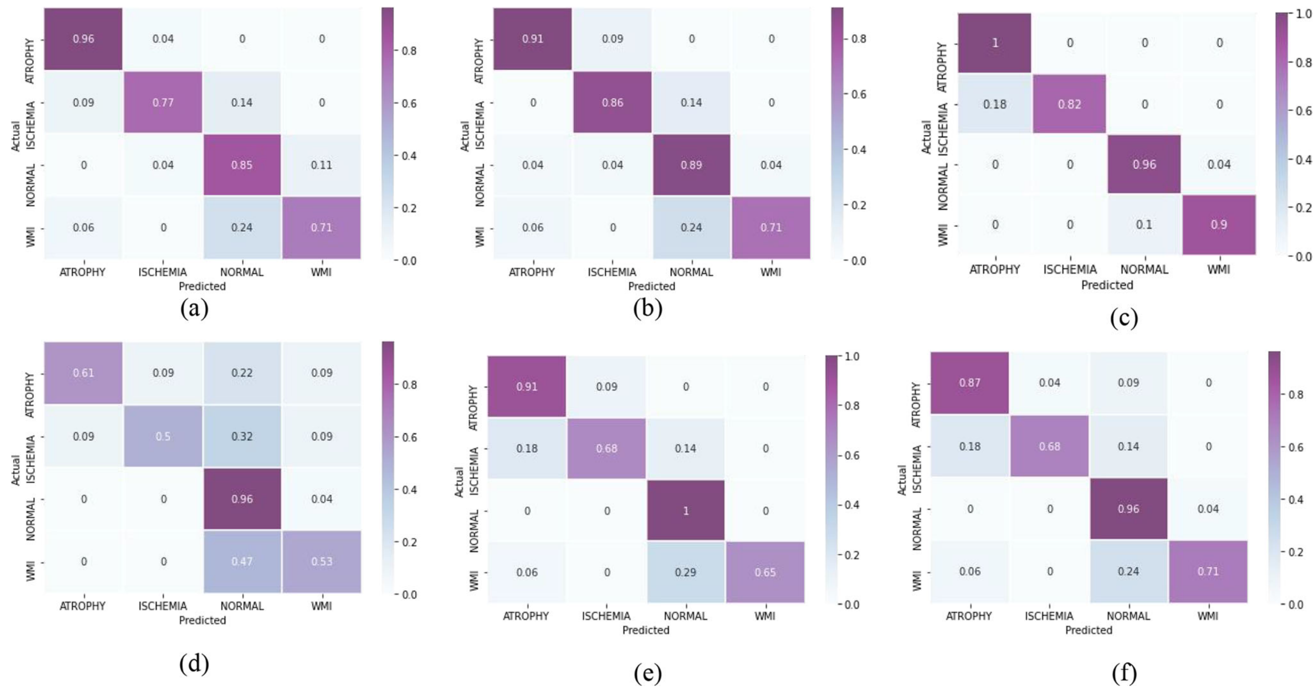


Fig. 7 Confusion matrix of the proposed method using EfficientNetB1 architecture. **a** SGD+softmax, **b** RMSProp+softmax, **c** Adam+softmax, **d** SGD+SVM, **e** RMSProp+SVM, **f** Adam+SVM

probabilities to others. As a result, we can conclude from our experimental results in Tables 7, 8, 9, 10, 11 and 12 that the softmax classification layer performed well. We can see from the confusion matrix of EfficientNet B1 using the SVM classifier and Adam in Fig. 7 that the "Ischcemia" class has 0.68 accuracy whereas using the softmax function it has 0.82 accuracy.

5.4 Time Complexity

The computation of time complexity can be calculated using Big $\mathcal{O}$ notation.

1. Pre-processing: If we consider i number of image samples than time complexity will be Big $\mathcal{O}(i)$.
2. Feature extraction: If we have f number of feature map and p complexity of the processing of EfficientNet B1 based on its parameters than time complexity will be Big $\mathcal{O}(fp)$.
3. Classification: If we consider the complexity of classifier is c than time complexity will be Big $\mathcal{O}(c)$.

So the total time complexity will be

$$T(n) = Big\mathcal{O}(i) + Big\mathcal{O}(fp) + Big\mathcal{O}(c) \quad (8)$$

The Eq. 8, provides a clear understanding of the time complexity associated with each stage of the EfficientNet B1 pipeline and it has a linear time complexity. For our experiments, we noticed that for each iteration it has taken 4 seconds per step.

5.5 Comparative Analysis with Existing Work

This section compares the proposed framework with the state-of-the-art works. Table 13 refers to some similar kind of classification framework. Cheng et al. (2015), used the Figshare dataset for brain tumor classification where the classifier differentiates among three different types of brain tumors glioma, meningioma, pituitary, and using bag of word (BoW) and support vector machine (SVM) classifier. Helwan et al. (2018), used a stacked autoencoder for brain hemorrhage classification using CT scan brain dataset and other auto encoder architecture for comparing their proposed model for binary class classification. Sajjad et al. (2019), used modified VGG on the Radiopedia dataset and achieved 90.6% accuracy. They performed the experiments on other brain tumor datasets and achieved superior performance. In this paper, the proposed approach fine-tuned EfficientNet B1 using Adam optimizer and softmax classifier achieved 93% test accuracy on brain disorder classification problem where it can classify four different types of brain disorder.

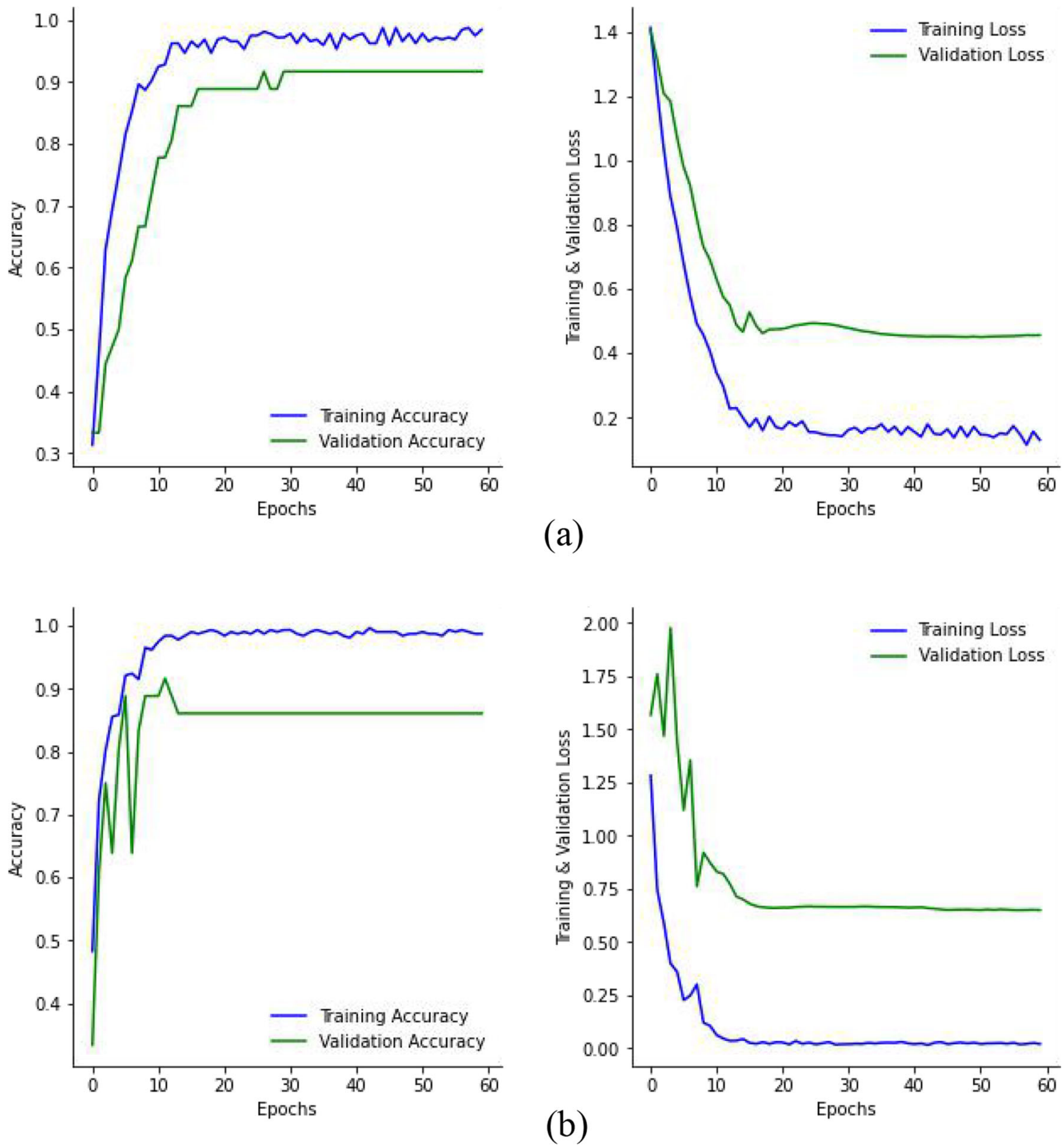


Fig. 8 Training and Validation Accuracy-Loss graph of EfficientNetB1 using softmax. **a** SGD, **b** RMSProp

5.5.1 Analysis of Work on Same Dataset

This subsection provides an analytical review of previously published papers that used the same brain disorder dataset. The paper by Poyraz et al. (2022), proposed a pipeline that includes pre-processing, feature generator, iterative neighborhood component analysis (INCA) feature selection, and classification using pre-trained architectures. MobileNet V2 with the SVM classification layer achieved the highest classification accuracy for brain disorder classification problems. But the INCA feature extraction technique is very complex and the iterative feature selection process selects 877 features from 21000 features and classification has been done using the 877 features. As we can see fewer features are selected by an automatic iterative feature selection method As here we can see less features are selected by an automatic iterative feature selection method so the accuracy of the classification is significantly impacted by the chosen features, resulting in the achievement of improved accuracy. Higher-level feature maps typically contain abstract and complex features that are important for discriminating between different classes. By

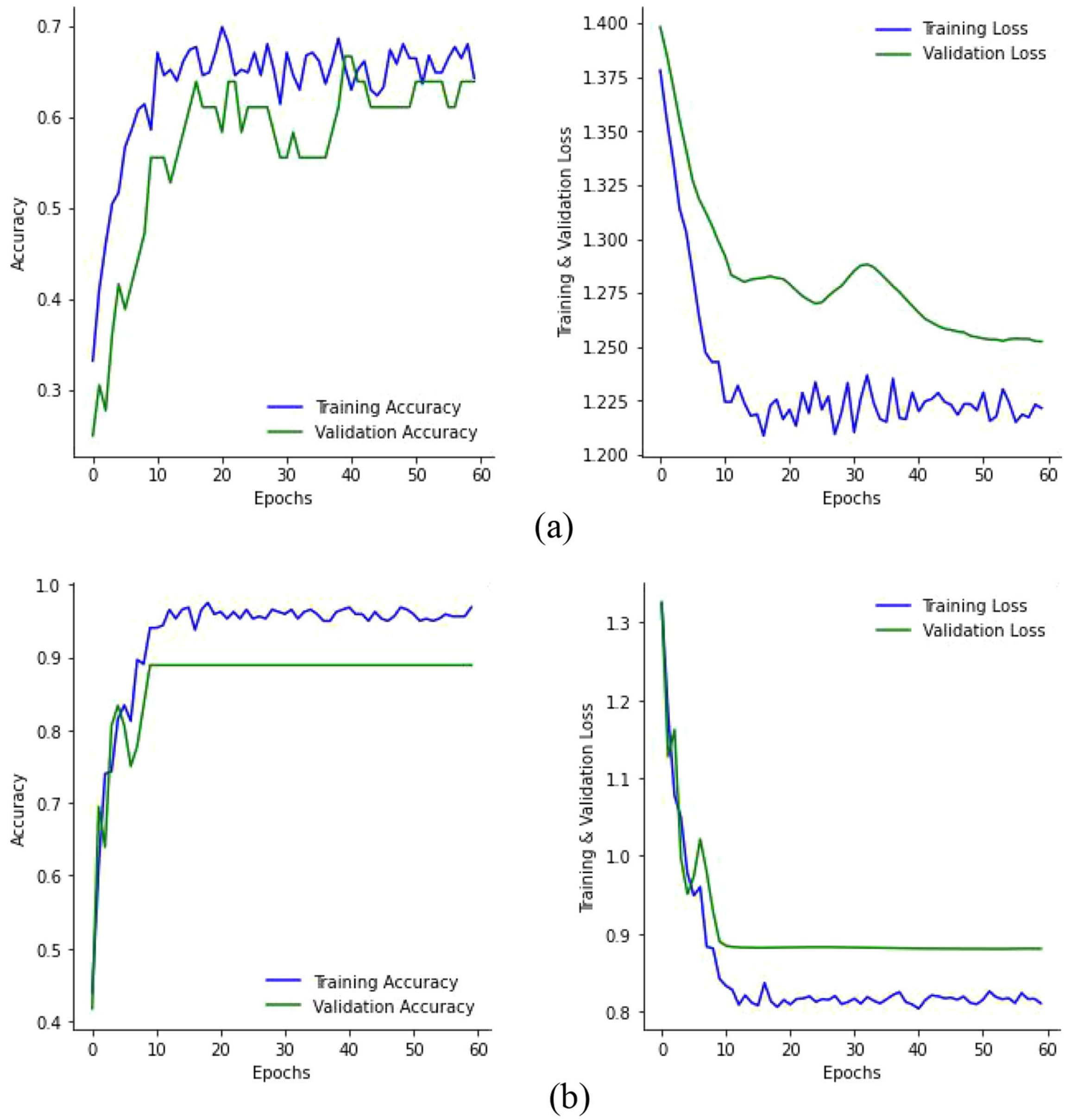


Fig. 9 Training and Validation Accuracy-Loss graph of EfficientNetB1 using SVM. **a** SGD, **b** RMSProp

selecting only a few features, it may lose critical information for the classification. And the framework may become more sensitive to noise or irrelevant patterns in the data. This can lead to overfitting. Moreover, their method focused only to increase the accuracy using Iterative Neighborhood Component Analysis (INCA) dimensionality reduction technique that iteratively refines the embedding of data points by considering their local neighborhoods. INCA's performance heavily relies on the choice of neighborhood size and the method used to define neighborhoods. If the neighborhood selection is sub optimal, it may lead to a poor-quality embedding, resulting in loss of important information or distorted representations. The iterative nature of INCA involves repeatedly updating the embeddings of data points until convergence is achieved. This iterative process can be computationally expensive, especially for large datasets or high-dimensional data, making it less practical for real-time or large-scale applications. But as the size of the dataset was small it worked well for this but for larger dataset it will be computationally expensive. INCA aims to capture local data structures by iteratively adjusting the embedding of each data point. However, this iterative refinement process may lead to over fitting a concern which the paper Poyraz

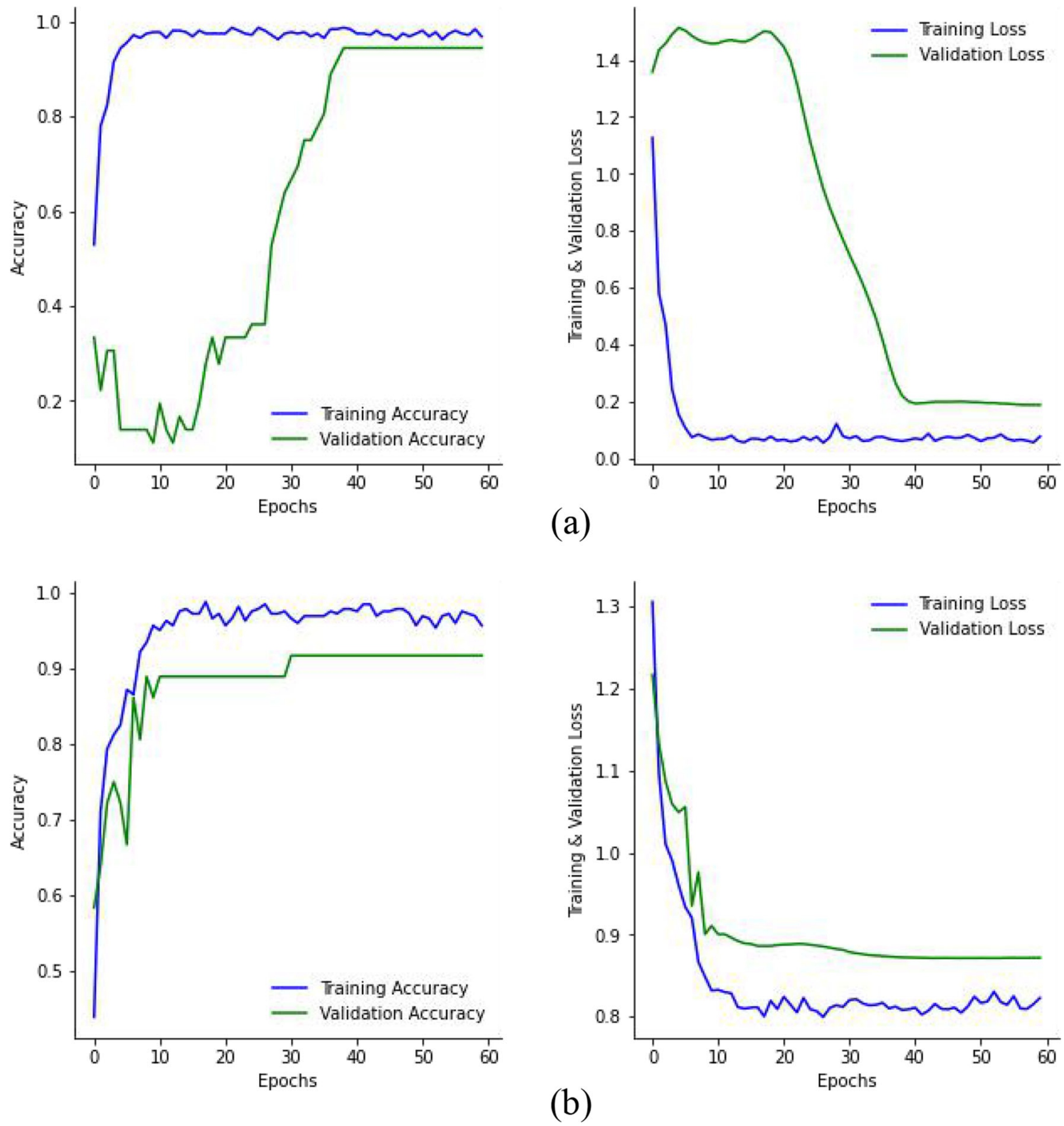
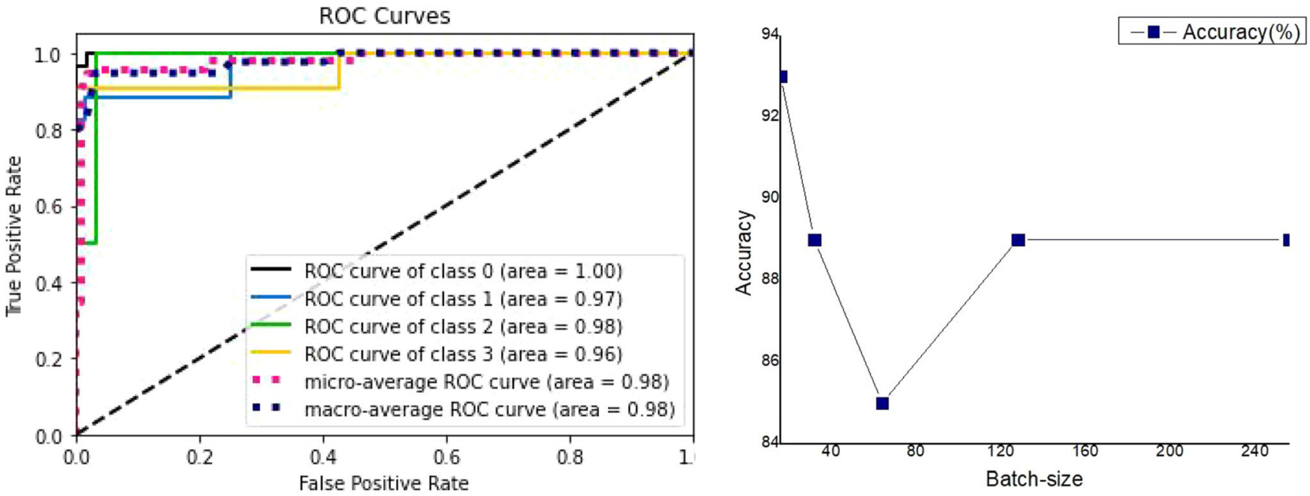


Fig. 10 Training and Validation Accuracy-Loss graph of EfficientNet using Adam optimizer. **a** Softmax, **b** SVM

et al. (2022) notably does not address in terms of how potential over fitting is mitigated. Although INCA provides a potent method for dimensionality reduction by utilizing local neighborhood information, it is crucial to thoroughly assess its limitations and suitability for particular applications.

Tasci (2023) adopted a novel approach utilizing pre-trained CNN architectures for feature extraction and iterative feature selection methods INCA (Iterative Neighborhood Component Analysis) and ImRMR (Iterative Minimum Redundancy Maximum Relevance). Remarkably, their methodology achieved impressive accuracy more than 97% for all the brain disease datasets. Notably, for the brain disorder dataset, a remarkable accuracy of 99.3% was attained using the ImRMR method. However, it's important to note that this high accuracy was achieved with a relatively sparse selection of features. Specifically, out of 32,928 feature vectors, only 302 features were chosen. The result of this could be potentially impacted by the selected features, contributing to the exceptional performance observed. The iterative nature of feature selection methods like ImRMR aims to select features that are highly relevant to the target variable while minimizing redundancy. But, this iterative optimization process can be computationally expensive, particularly when dealing with a large number of features. Moreover, scalability becomes

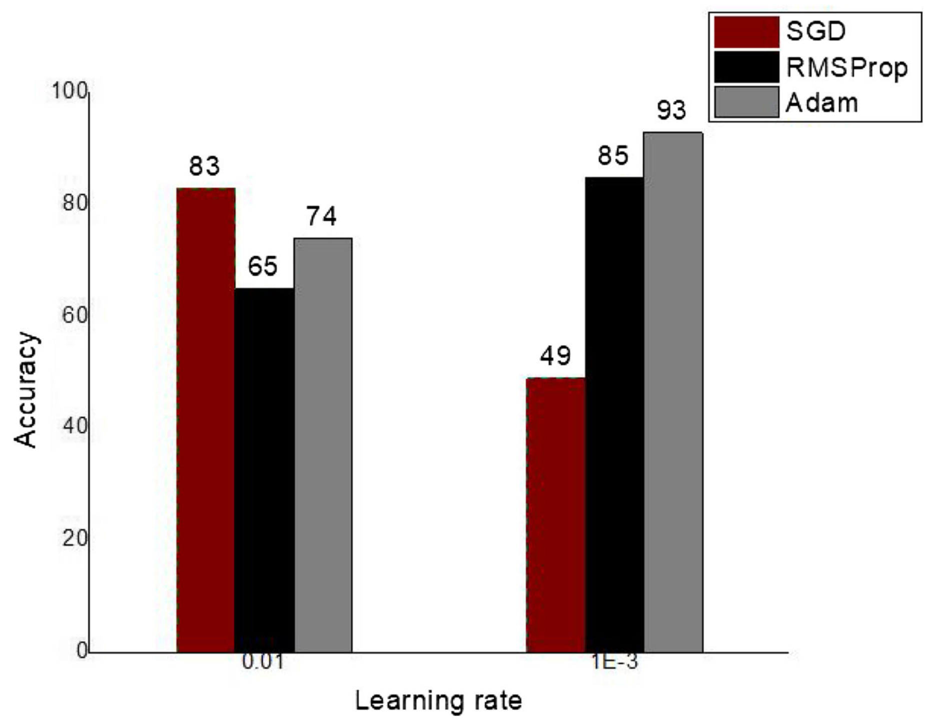


(a) ROC curve

(b) Batchsize-Accuracy

Fig. 11 **a** ROC-AUC curve and **b** batch size-accuracy variation of EfficentNet B1 with Adam optimizer default learning rate (0.001)

Fig. 12 The performance of optimizers SGD, RMSProp, and Adam with learning rate 0.01 and 0.001(1E-3) using finetuned EfficientNet B1 a softmax classifier



a concern, as the algorithm's complexity increases with larger datasets or higher-dimensional feature spaces. These challenges may limit the method's practical applicability in certain computational environments or with particularly large datasets. Despite its principled approach to feature selection, offering a balance between relevance and redundancy, it's essential to consider the computational complexities and scalability limitations associated with ImRMR. Although ImRMR offers a better approach to feature selection by maximizing the relevance and minimizing the redundancy of selected features, it is important to consider its computational complexity and scalability limitations challenges, and potential for over fitting when applying it to larger datasets.

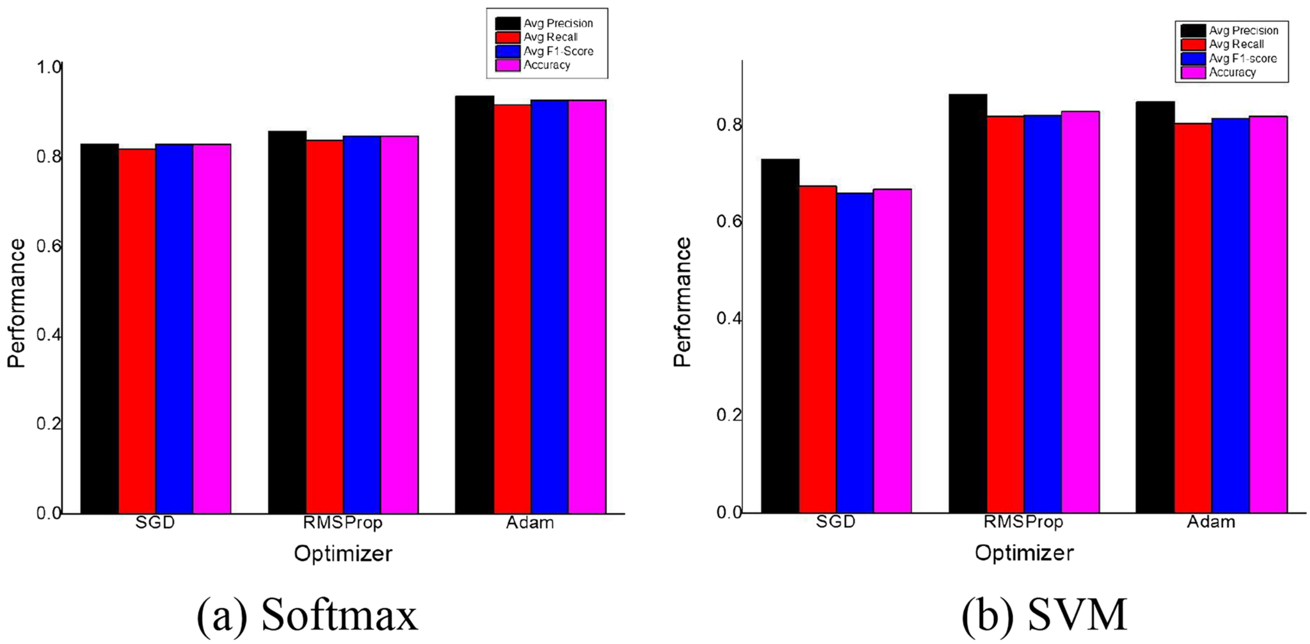


Fig. 13 Average Precision, Recall, F1-score of EfficientNetB1 architecture using SGD, RMSProp and Adam optimizer

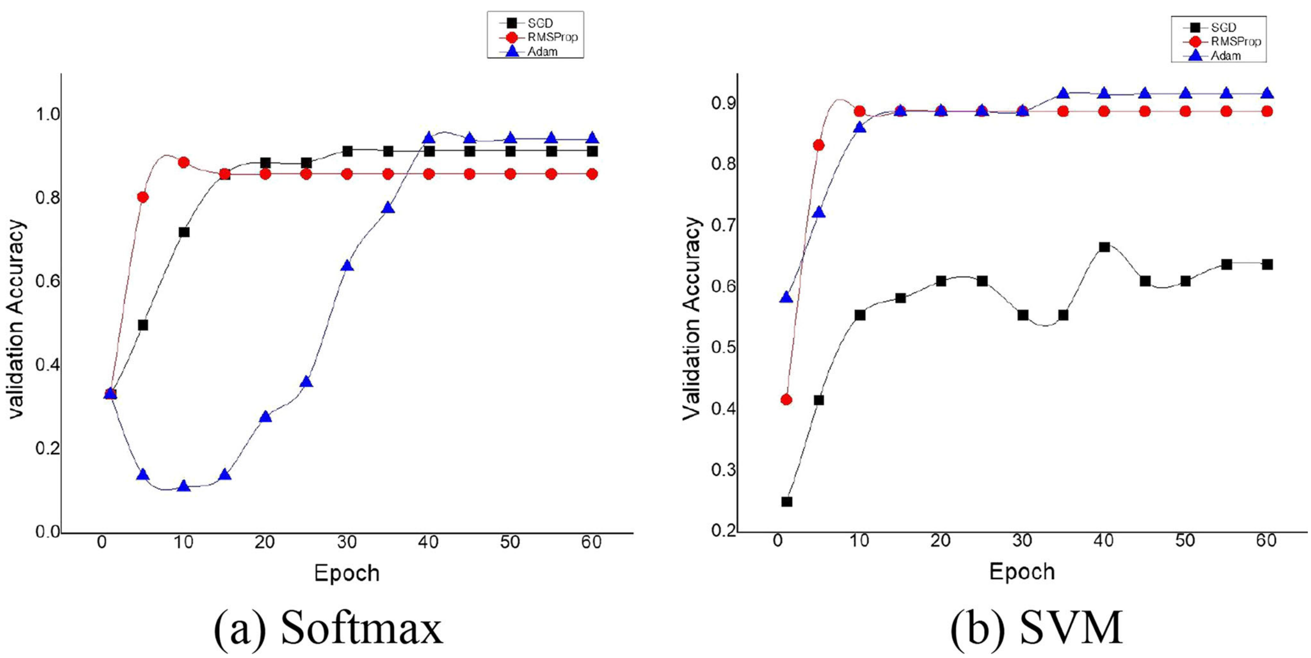


Fig. 14 Comparison of validation accuracy of EfficientNetB1 architecture using SGD, RMSProp and Adam optimizer

While our proposed method may exhibit only marginal accuracy compared to existing approaches, it offers significant advantages in terms of computational efficiency and improved generalization through the use of a global average pooling layer. Additionally, our model shows robustness against over fitting problems. However, we recognize that there is opportunity for improvement, specifically by incorporating larger training samples or employing larger datasets, that could potentially result in higher accuracy. Our future studies will focus on integrating a larger brain disease dataset to improve the performance of our model.

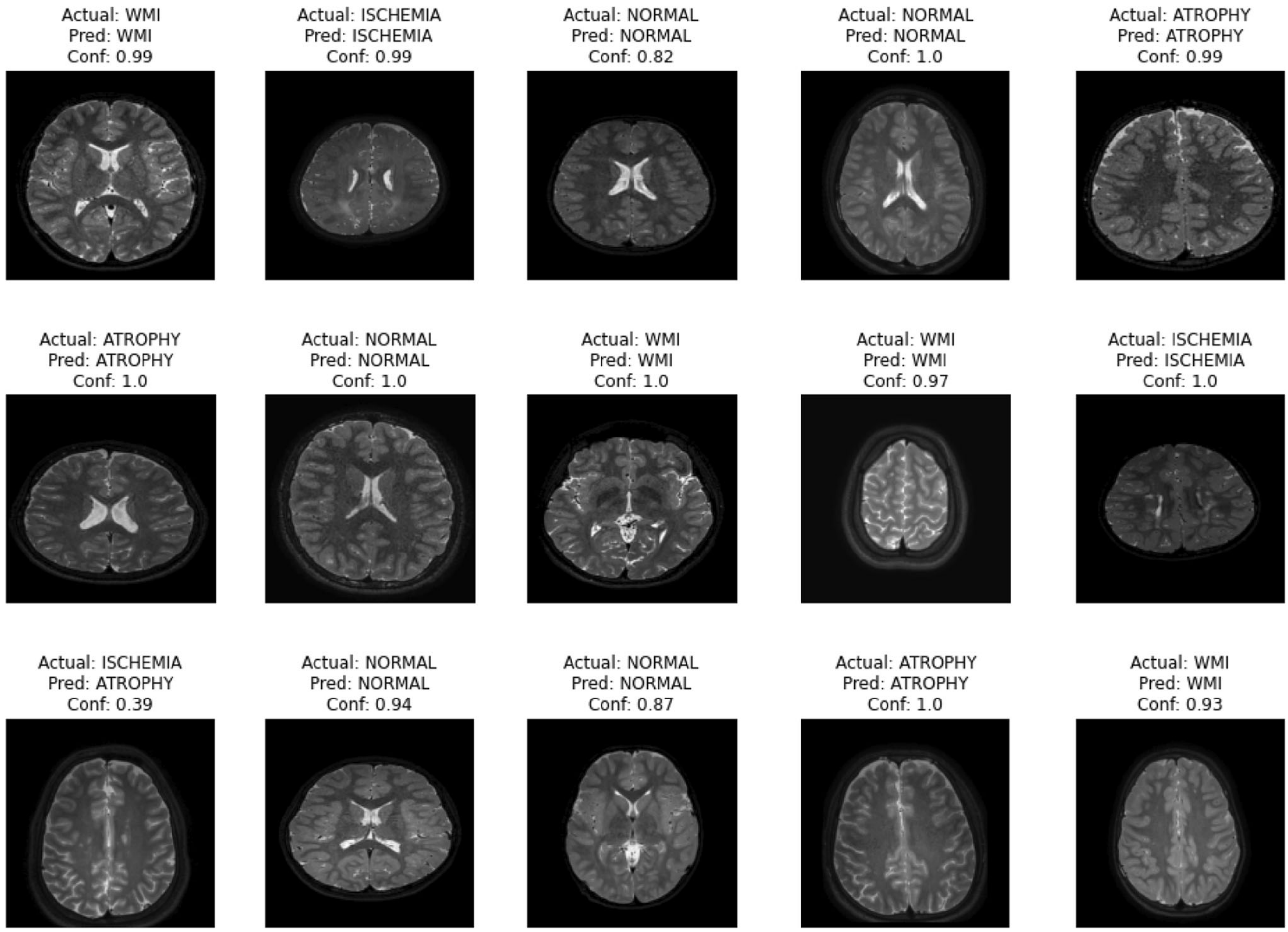


Fig. 15 The predicted visualization of classification result on brain disorder test dataset using proposed architecture

6 Conclusion and Future Work

This article focuses on developing a deep learning framework for various types of brain disorder classification. The proposed framework includes Inception V3, ResNet50 V2, and EfficientNet B1 which has been fine-tuned using transfer learning for feature extraction and classification performance compared using softmax and SVM. Stochastic gradient descent, RMSProp, and Adam are three popular activation functions that are used and compared. Transfer learning can produce optimistic results as large amounts of brain disorder data are difficult to collect. Employing the EfficientNet B1 architecture for brain disorder classification represents a significant advancement in the field of medical image analysis. This powerful deep learning model, characterized by its efficient use of computational resources and impressive performance, has demonstrated its efficacy in accurately classifying multiple types of brain disorders from brain MRI datasets. Furthermore, the efficiency of EfficientNet B1 in terms of computational resources and memory requirements makes it a practical choice for deployment in resource-constrained environments, ensuring that the model can be readily applied in clinical practice without excessive hardware demands.

However, it is important to note that the success of any classification model, including EfficientNet B1, is dependent on the availability of high-quality, well-curated datasets. The quality and diversity of data play a pivotal role in training models that generalize well to real-world clinical scenarios. The performance evaluation of other feature extraction and transfer learning techniques will be part of future research for brain disorder classification by utilizing a larger dataset. Moreover, utilization of transfer learning techniques beyond the pre-trained models is another area of research that can be explored.

Table 13 Comparative results analysis of brain disorder classification system with existing related state-of-the-art systems

Paper	Approach	Dataset	Training Sample	Accuracy	Limitations
Cheng et al. (2015)	BoW + SVM	Figshare	80%	91.2%	Discriminative visual dictionary learning techniques can be used to improvement of BoW's performance.
Helwan et al. (2018)	Stacked Autoencoder	Brain CT Hemorrhage	85%	90.9%	Only two-class classification problem has been solved.
Sajjad et al. (2019)	modified VGG	Radiopedia data	50%	90.6%	Used a heavy-weight CNN architecture to achieve the highest accuracy which is time-consuming.
Khawaldeh et al. (2017)	modified AlexNet	TCIA dataset	69%	91.16%	Only Flair type of MRI samples are used for training purposes.
Poyraz et al. (2022)	MobileNet V2	Brain disorder dataset	10-fold validation	99.1%	The INCA feature selection method is complex to use and fewer features are selected for classification by this method. So the performance is highly biased towards the selected features for classification accuracy.
Tasci (2023)	Pre-trained CNN	4 different brain disease dataset	10-fold validation	more than 97% for each datasets	The ImRMR method included for feature selection achived highest accuracy by its iterative technique but it selects very sparse features that may result bias towards the selected features.
Proposed	Fine-tuned EfficienNetB1 + Adam + Softmax	Brain disorder dataset	70%	93%	This method is lightweight and computationally inexpensive and achieves better generalization on the feature map by global average pooling layer but the dataset has less number of MRI samples and with larger data set the framework can achieve better results.

Data Availability For this study the publicly available Brain disorder dataset Poyraz et al. (2022), TUNCER (2023).

Declarations

Conflict of interest The authors declare that they do not have any Conflict of interest.

References

Alhassan AM, Zainon WMNW (2021) Brain tumor classification in magnetic resonance image using hard swish-based relu activation function-convolutional neural network. Neural Comput Appl 33(15):9075–9087

Alyami J, Rehman A, Almutairi F, Fayyaz AM, Roy S, Saba T, Alkhurim A (2023) Tumor localization and classification from MRI of brain using deep convolution neural network and salp swarm algorithm. Cognit Comput 1–11

Anaraki AK, Ayati M, Kazemi F (2019) Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. Biocybern Biomed Eng 39(1):63–74

Aurna NF, Yousuf MA, Taher KA, Azad A, Moni MA (2022) A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models. Comput Biol Med 146:105539

Brima Y, Tushar MHK, Kabir U, Islam T (2021) Deep transfer learning for brain magnetic resonance image multi-class classification. arXiv preprint arXiv:2106.07333

Cancernet, brain tumor: statistics. https://www.cancer.net/cancer-types/brain-tumor/statistics."

Cheng J, Huang W, Cao S, Yang R, Yang W, Yun Z, Wang Z, Feng Q (2015) Enhanced performance of brain tumor classification via tumor region augmentation and partition. PLoS ONE 10(10):e0140381

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Deepak S, Ameer P (2019) Brain tumor classification using deep CNN features via transfer learning. Comput Biol Med 111:103345

Deepak S, Ameer P (2021) Automated categorization of brain tumor from MRI using CNN features and SVM. J Ambient Intell Humaniz Comput 12(8):8357–8369

Deepa S, Janet J, Sumathi S, Ananth J (2023) Hybrid optimization algorithm enabled deep learning approach brain tumor segmentation and classification using MRI. J Digital Imaging 36:1–22

Ghosh A, Soni B, Baruah U, Murugan R (2022) Classification of brain hemorrhage using fine-tuned transfer learning. Advanced machine intelligence and signal processing. Springer, Berlin, pp 519–533

Haq EU, Jianjun H, Li K, Haq HU, Zhang T (2021) An MRI-based deep learning approach for efficient classification of brain tumors. J Ambient Intell Humaniz Comput 14:1–22

Helwan A, El-Fakhri G, Sasani H, Uzun Ozsahin D (2018) Deep networks in identifying CT brain hemorrhage. J Intell Fuzzy Syst 35(2):2215–2228

Khawaldeh S, Pervaiz U, Rafiq A, Alkhawaldeh RS (2017) Noninvasive grading of glioma tumor using magnetic resonance imaging with convolutional neural networks. Appl Sci 8(1):27

Kibriya H, Masood M, Nawaz M, Nazir T (2022) Multiclass classification of brain tumors using a novel CNN architecture. Multimed Tools Appl 81:1–17

Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint arXiv:1312.4400

Mehrotra R, Ansari M, Agrawal R, Anand R (2020) A transfer learning approach for AI-based classification of brain tumors. Machine Learn Appl 2:100003

Nwankpa C, Ijomah W, Gachagan A, Marshall S (2018) Activation functions: comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378

O'Shea K, Nash R (2015) An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458

Polat Ö, Güngen C (2021) Classification of brain tumors from MR images using deep transfer learning. J Supercomput 77(7):7236–7252

Poyraz AK, Dogan S, Akbal E, Tuncer T (2022) Automated brain disease classification using exemplar deep features. Biomed Signal Process Control 73:103448

Rane C, Mehrotra R, Bhattacharyya S, Sharma M, Bhattacharya M (2021) A novel attention fusion network-based framework to ensemble the predictions of CNNS for lymph node metastasis detection. J Supercomput 77(4):4201–4220

Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747

Sajjad M, Khan S, Muhammad K, Wu W, Ullah A, Baik SW (2019) Multi-grade brain tumor classification using deep CNN with extensive data augmentation. J Comput Sci 30:174–182

Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, PMLR, pp. 6105–6114

Tasci B (2023) Automated ischemic acute infarction detection using pre-trained CNN models' deep features. Biomed Signal Process Control 82:104603

TUNCER T (2023) kaggle datasets download -d turkertuncer/brain-disorders-four-categories. Accessed

Usmani IA, Qadri MT, Zia R, Alrayes FS, Saidani O, Dashtipour K (2023) Interactive effect of learning rate and batch size to implement transfer learning for brain tumor classification. Electronics 12(4):964

Veni N, Manjula J (2022) High-performance visual geometric group deep learning architectures for MRI brain tumor classification. J Supercomput 78:1–12

Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big data 3(1):1–40