Neural Computing and Applications (2021) 33:9877–9889
https://doi.org/10.1007/s00521-021-05758-5

ORIGINAL ARTICLE

Enhancing of dataset using DeepDream, fuzzy color image enhancement and hypercolumn techniques to detection of the Alzheimer's disease stages by deep learning model

Mesut Toğaçar[1] · Zafer Cömert[2] · Burhan Ergen[3]

Received: 14 May 2020 / Accepted: 20 January 2021 / Published online: 17 February 2021

Abstract
Alzheimer's disease (AD), which occurs as a result of the loss of cognitive functions in the brain, causes near-forgetfulness in the case and dementia in subsequent processes. Dataset consists of MR images containing four phases of AD. The dataset was re-enhanced separately with DeepDream, fuzzy color image enhancement, hypercolumn techniques. Visual Geometry Group-16 (VGG-16) deep learning model is used in the enhancing process and deep features are combined. Linear Regression is used for the selection of efficient features. The Support Vector Machine is preferred as a classifier. With the proposed approach, the classification achievement was obtained as 100% in Mild Dementia, 99.94% in Moderate Dementia, 100% in non-Dementia, 99.94% in Very Mild Dementia. The overall accuracy was 99.94%. The proposed approach increased the prediction success in detecting Alzheimer's stages by re-enhancing MR images. Thus, an efficient early diagnosis model was realized at an affordable cost for individuals likely to progress with dementia.

Keywords Decision-making · Alzheimer's disease · DeepDream learning and hypercolumn technique · Fuzzy color image enhancement

1 Introduction

Dementia is a disease that causes loss of talent during the action of activities carried out in daily life and causes the cognitive functions of the brain to gradually disappear. The most common type of dementia in the world is Alzheimer's disease (AD) with a rate of about 80%. It is estimated that approximately 135 million AD cases will occur in the world by 2050 [1]. AD occurs as a result of damage or destruction of neurons that provides cognitive functions within the brain. This situation causes physical changes in the brain over time. As a result, the physical actions of the person decrease over time, and bed care is carried out in the last phase. Ultimately, AD is a deadly disease [2].

Symptoms of AD are the decrease and disappearance of thinking, memory, and behavioral responses seen in the person over the years. Pharmacological and non-pharmacological methods are used in the treatment process of AD cases. However, not all of these methods completely cure AD cases. Only a certain time delays the progression of the disease [2, 3]. As a result, the early diagnosis of AD is important. As a requirement of this condition, it is important to detect the disease stage of AD cases using brain images. Many studies have been conducted recently using deep learning models and methods related to AD. Basaia et al. [4] used healthy and AD brain MR data. They used the transfer learning approach to training the convolutional neural network (CNN). They performed a preprocess step by using the data augmentation techniques to ensure enough data for the proposed CNN model. The binary

✉ Mesut Toğaçar
mtogacar@firat.edu.tr

Zafer Cömert
zcomert@samsun.edu.tr

Burhan Ergen
bergen@firat.edu.tr

[1] Department of Computer Technology, Technical Sciences Vocational School, Fırat University, Elazig, Turkey

[2] Department of Software Engineering, Faculty of Engineering, Samsun University, Samsun, Turkey

[3] Department of Computer Engineering, Faculty of Engineering, Fırat University, Elazig, Turkey

classification success was 99.2%. M. Liu et al. [5] performed segmentation and classification processes together using AD data in their study. The dataset was organized with three classes that were mild cognitive impairment (MCI), AD, and normal contrast (NC). They used the DenseNet deep learning model in the training of the model and achieved 88.9% success by using the Softmax method in the layer where the classification process of the model took place. Lu et al. [6] performed a binary classification using AD and healthy brain MR images in their study. In this study, researchers used Visual Geometry Group-16 (VGG-16) and MobileNet deep learning models. The classification success with the VGG-16 model was 93%, whereas the classification success with the MobileNet model was 98%. T. Shen et al. [7] performed a classification task using stable MCI and progressive MCI data from AD cases. The dataset includes PET images. They used the Region-of-Interest (ROI) method for each image in the dataset. They used Deep Belief Networks (DBN). The classification success with the SVM method was reported as 86.6%. Z. Xiao et al. [8] performed the classification of AD using 2D and 3D brain images. In their three-class dataset, they extracted features using the Gray Level Co-occurrence Matrix (GLCM) and Gabor filter techniques. Besides, features were selected with the Voxel-based Morphometry (VBM) method. They achieved 92.86% classification success using the extracted features with the SVM classifier.

As inferred from related literature, deep learning models that analyze computer images have been developed in this area, recently. Since deep learning models perform the learning process automatically instead of using handcrafted feature engines, they are more efficient compared to traditional machine learning models [9, 10]. This situation explains why a deep learning model has been preferred in this study. The primary goal of this study is to enhance the Magnetic Resonance (MR) images using DeepDream, Fuzzy Color Image Enhancement (FCIE), and Hypercolumn techniques. Then, we validate whether the re-enhancing MR images contribute to the classification achievement by selecting the most efficient features from these three image sets. In this study, deep features, which are extracted from the pretrained VGG-16 model fed with re-enhancing MR image sets, are obtained and these features are concatenated. At the last step, the most efficient 1000 features are selected with the Linear Regression (LR), and then the classification task is carried out using the SVM classifier. This approach shows us that the proposed model has produced satisfactory results in detecting AD stages. In the proposed approach, the aim is to reveal the effects of the mentioned enhancing techniques by comparing the classification achievements of the proposed model considering the original and enhanced datasets, separately. VGG-16 model is preferred in this study since it is used in the Hypercolumn technique and the number of computational parameters of the model is not too large. Also, our hardware resources are sufficient to train and test the VGG-16 at a reasonable time [11, 12]. In other words, the computational environment we have could not be functional in using a more complex deep learning model that requires a large number of computational parameters. The proposed approach has gained a different perspective compared to the studies reviewed in the literature. The contributions of this study to the relevant field can be summarized as enabling the data enhancing of brain MR images with the mentioned techniques and selecting efficient features with the proposed approach.

The rest of this study is organized as follows: the dataset, deep learning models, methods, and techniques are given in Sect. 2. Also, the proposed approach in this study is described in Sect. 2. The results are reported in Sect. 3. Sections 4 consists of discussion and conclusion remarks.

2 Materials and methods

2.1 Dataset

The AD dataset was created by researcher Sarvesh Dubey [13]. The MR images with varying resolutions in the dataset were collected from different websites. The hybrid dataset used in this study was shared on the web. The dataset consists of a total of 6400 images in the JPG file format. The dataset is divided into two parts in its original form. The training data consist of 5121 images, whereas the test data consist of 1279 images. In this case, the dataset was separated into two parts as 75% of the train set and 25% of the test set. The dataset is divided into classes containing four phases of AD. These phases are Mild Demented (MID), Moderate Demented (MOD), Non-Demented (NOD), and Very Mild Demented (VMD) [13]. In addition, no detailed information was given about the patients from whom the dataset was obtained. For this reason, it is not known that the images in the dataset are homogeneous or heterogeneous. Sample images representing the classes of the dataset are shown in Fig. 1.

2.2 Deep learning model: VGG-16

VGG-16 architecture is a convolutional model that made its name known in the ImageNet competition in 2014. This model includes convolutional layers, pooling layers, and fully connected layers in its structure. It moves the activation features by circulating 3×3, 5×5 filters on the convolutional layer input data [14]. The pooling layer is used to reduce the size of the input features, thus

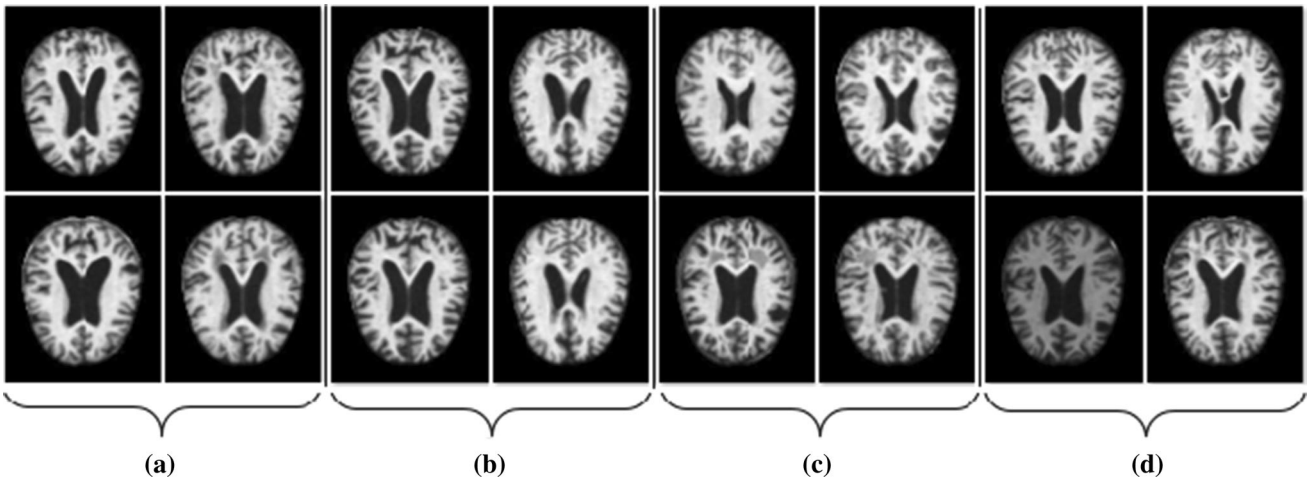

Fig. 1 Example sub-dataset for four classes of AD images; **a** MID, **b** MOD, **c** NOD, and **d** VMD

contributing to the performance of the model. This layer also avoids overfitting. Fully connected layers provide the preparation of probable values of the features before the classification process [15–17]. In fully connected layers, the FC-6 layer and FC-7 layer contain 4096 features. The FC-8 layer contains 1000 features. The input size of the VGG-16 model is 224 × 224 pixels [16]. The general structure of the VGG-16 model is shown in Fig. 2.

The pretrained VGG-16 model was used in the experimental steps of this study. After the FC layers of the VGG-16 model, the SVM method was used as a classifier. VGG-16 was used with its default parameters. Also, important parameters and preferred values of the VGG-16 model [18] are given in Table 1.

To ensure a reasonable training time for the CNN model as well as achieving high generalization results, we preferred the VGG-16 model [19]. As the number of compute nodes increases, the costs are also increased in terms of hardware resources as well as time for the model training [20]. The related papers have already taken attention to these issues [21–25]. So, in this context, we chose the VGG-16 model by taking into account the related literature and the hardware resources that we have.

2.3 Machine learning: SVM method

SVM is a machine learning method that is commonly used in regression and classification processes and finds a decision boundary between the features to be classified. This method creates a decision boundary by using features placed in the hyper-plane. The decision boundary divides the features into two parts. Then, SVM calculates the probabilities of features based on this limit. The traded feature is assigned to that class; whichever probability value is higher [26]. The operation of the SVM method is shown in Fig. 3, and the mathematical formulas used in the SVM method are specified between Eqs. (1) and (3). Here, X and Y represent the coordinate points of the features in the hyper plane. W parameter represents margin width and b parameter represents bias value. The multiple classification process of the SVM method is shown in Fig. 3b. Here, the basic logic solves a probability class k by using the formula $k(k-1)/2$. For example, let's assume that k is 3. Therefore, according to the equation, probability condition $3(3-1)/2 = 3$ is obtained. If we define these possibilities according to Fig. 3b respectively; red-blue, blue-yellow, red-yellow. Then, three probability values are subtracted for each feature. Then, each feature is transferred to the class with the highest probability value [27].

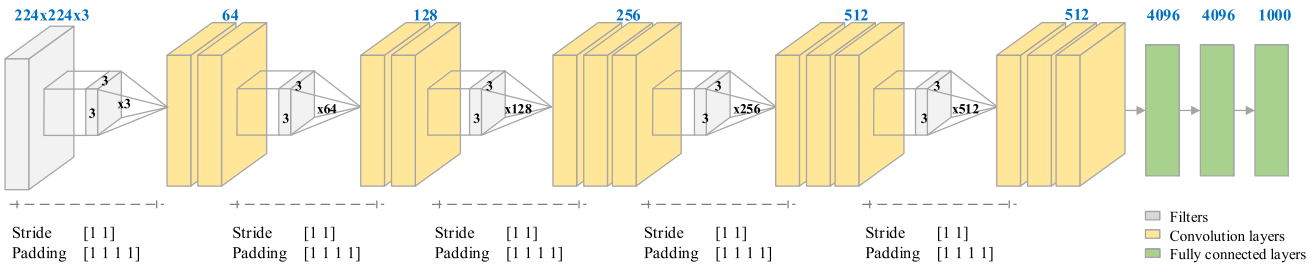

Fig. 2 VGG-16 architecture and general structure

Table 1 Preferred important parameter values of the VGG-16 model

Software used	model	Image size	Optimization	Momentum	Decay	Beta	Mini batch	Learning rate
MATLAB	AlexNet	224 × 224	Stochastic Gradient Descent (SGD)	0.9	1e − 6	–	64	10^{-4}

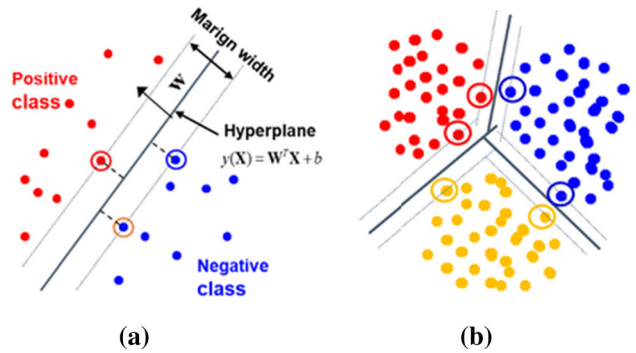

Fig. 3 Classification representation of the SVM method. **a** Binary classification, **b** multiple classification [28]

$$u = \vec{w} \cdot \vec{x} - b \tag{1}$$

$$\frac{1}{2}\|\vec{w}\|^2 \tag{2}$$

$$y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1, \forall i \tag{3}$$

One of the reasons why the SVM method was preferred for this study is its strong potential to provide solutions to data analysis problems encountered in daily life. The other reason is that it can efficiently execute multiple classification processes in pattern recognition and classification [29, 30]. Also, the performances of other machine learning methods (discriminant analysis, nearest neighbor, random forest, etc.) were analyzed. However, SVM was chosen as the classification method in this study because it gives us the best performance. Other parameter values preferred in the SVM method are as follows: it was the kernel scale from which the parameter was automatically selected. The box constraint level parameter value was selected and a one-to-one multi-class method parameter was selected.

2.4 DeepDream

The DeepDream is a simulation technique that is based on the imaginary dimensions of the human brain, using the features of the input images. The DeepDream is developed by Google, and the open-source code of the DeepDream is supported using the Tensorflow library. The DeepDream aims to find patterns in images through algorithmic pareidolia and to develop the patterns found using a CNN. That is, it makes the patterns (feature vector carrying the label of an object) seen in a particular image appear on the data by processing them with pretrained data. For example, after a set of images with cats is trained by the network, cat faces will become a DeepDream-defined pattern. When the network is fed with a sky image, the DeepDream algorithm creates images of cats at objects in the sky (cloud, etc.). The overall process of the DeepDream neural network model is illustrated in Fig. 4. As a result, the DeepDream algorithm is useful for rendering visual content in surreal and abstract styles in over-processed images [31, 32].

The working logic of the DeepDream algorithm is as follows; when an image is an input to a trained neural network model, neurons fire and activations occur. The DeepDream algorithm selects some of these neurons, allowing them to fire more than others. So, it increases their activation. The activation boost is accomplished by gradient ascent. This process is repeated until it contains all the features that the particular layer was originally looking for [31].

In this study, AD data were produced using the DeepDream technique. Data enhancing examples are shown in Fig. 5.

2.5 Fuzzy color image enhancement (FCIE) algorithm

FCIE algorithm plays an important role in image analysis. This algorithm consists of three stages function. These stages are as follows; image coding, fuzzy enhancement, and image decoding as shown in Fig. 6.

Thanks to these functions, the gray level intensities in each image are matched to a fuzzy plane. In the first stage, each image in the dataset is converted from gray level area

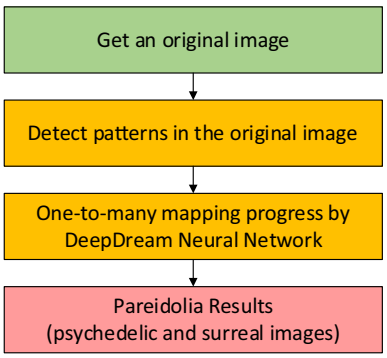

Fig. 4 The overall process of the DeepDream neural network model

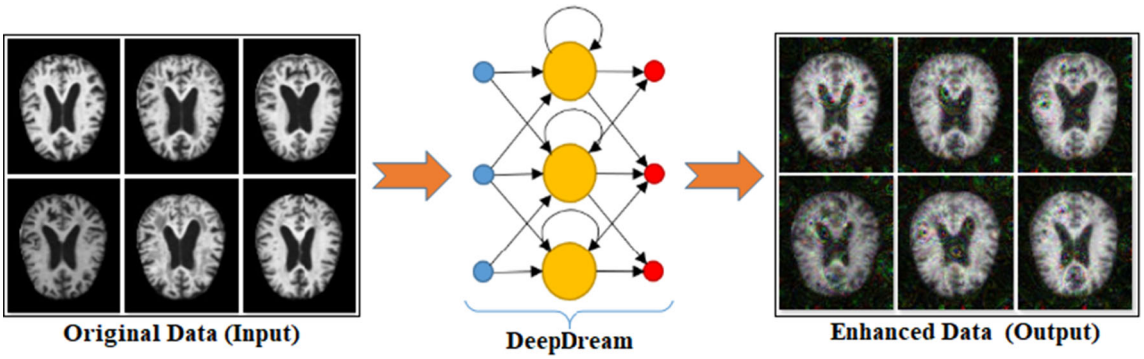

Fig. 5 Sub-data samples of the original dataset obtained by the DeepDream technique

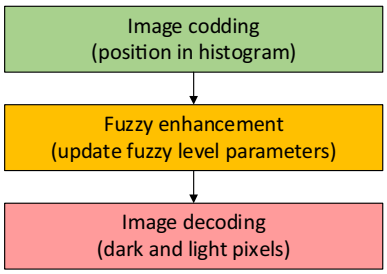

Fig. 6 The stages of FCIE algorithm

to fuzzy level area. At this stage, membership value is given for each pixel. That is, each gray level pixel is assigned a membership degree depending on its position in the histogram. In the second stage, the fuzzy level parameters used to change the image are updated and the aim is to enhance the fuzzy in the image. In the last stage, fuzzy enhanced images are decoded to re-convert them into gray-level images. Each pixel in the fuzzy level is converted into gray-level pixels according to the membership degree. Generally, dark pixels have a low membership degree; light pixels have a higher membership degree [33–35]. Calculation of the membership value the formulas between Eqs. (4) and (6) are used. In these equations; F_d and F_e are conversion coefficients, $f_{\max}$ denotes the maximal gray value, f_{ij} denotes the gray level of the (i,j) th pixel, $T^{(r)}$ is defined as successive applications of T [35].

$$\mu_{ij} = \left[1 + \frac{f_{\max} - f_{ij}}{F_d}\right]^{-F_e} \tag{4}$$

$$\mu'_{ij} = T^{(r)}(\mu_{ij}); r = 1, 2, 3\ldots, \tag{5}$$

$$T(\mu_{ij}) = \begin{cases} 2(\mu_J)^2; & 0 \le \mu_J \le 0.5 \\ 1 - 2(1-\mu_J)^2; & 0.5 \le \mu_J \le 1 \end{cases} \tag{6}$$

Thanks to the FCIE algorithm assigning membership degree based on the histogram distribution of the images, the contrast enhancement speed and quality can be enhanced in each image data [35]. In this study, we enhanced the original dataset using the Python codes with the FCIE algorithm [36]. Data enhancing image examples are shown in Fig. 7.

2.6 Hypercolumn technique

Hypercolumn is a technique that performs the classification of pixels using hypercolumn. That is, each image given as an input to the model has a hypercolumn vector. These hyper vectors hold all the activation features of that pixel in the convolutional model. Thus, instead of deciding according to the pixel value in the final layer of the convolutional model in the classification process, it chooses the most efficient one by examining all the features in the hypercolumn vector. Thus, with this technique, the spatial location information of the most efficient feature is brought from the previous layers and contributing to the classification process [37, 38].

The essence of the Hypercolumn technique is based on heat maps. After the convolution layers of the model, this technique uses bilinear interpolation and creates a transition feature value using two feature values with Bilinear interpolation. In other words, bilinear interpolation creates a smooth transition value between two feature values. In this way, feature maps extracted from other layers of the model are added, and it is processed with the sigmoid function. Heat maps extracted from the model are then combined to produce possible output values. This joining is done by the "Concatenate" function in the Hypercolumn technique. The purpose of our use of the Hypercolumn technique is to obtain a new dataset by enhancing the original dataset. Hypercolumn is a technique that functions with a deep learning model. This technique was compiled in the Python language along with the VGG-16 model [39]. A new set of images was created by selecting efficient features obtained from heat maps combined in the last layer of the VGG-16 model [38, 40]. Some images obtained from the data enhancing with the Hypercolumn technique are shown in Fig. 8.

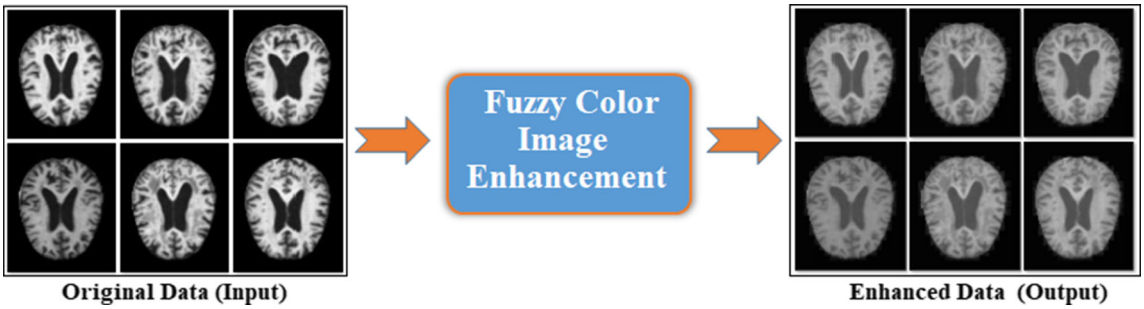

Fig. 7 Sub-data samples of the original dataset obtained by the FCIE algorithm

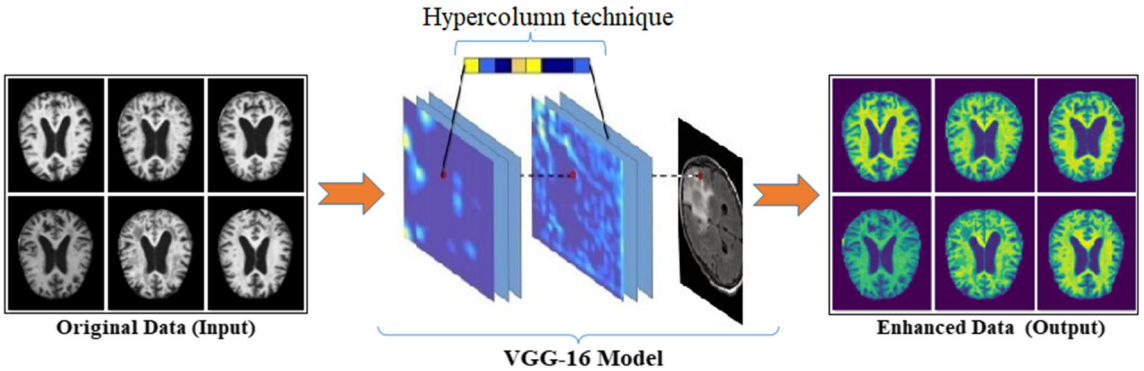

Fig. 8 Sub-image samples obtained by the work of the Hypercolumn technique in the VGG-16 model

2.7 Network training and feature selection methods

The Stochastic Gradient Descent (SGD) optimization is a method that facilitates network training by convolutional networks, contributes to the success of the model, and reduces the error value of the parameters used in the model to a minimum level. This optimization method is generally used by deep learning models. The SGD method updates the weight parameters of the network during training by selecting some data randomly, instead of using all of the input data of the convolutional networks [41, 42]. Also, instead of calculating the total cost of the model, it calculates the cost at the end of each iteration. Thus, it enables evolutionary networks to be trained more easily. The formula that explains this situation is given in Eq. (7). In this equation; Θ represents the weight parameter, t is the time state, α is the learning ratio. Parameters x and y show the coordinate values of the feature extracted from the input data [43].

$$\Theta_t = \Theta_{t-1}\alpha\nabla_\Theta J\left(\Theta; x^i, y^i\right) \tag{7}$$

Feature selection is a method that prevents inefficient features from being processed by the model in deep learning models and contributes to the success of the model. LR is an important statistical method for analyzing medical data. This method allows the relationships between multiple factors to be defined and characterized [44]. A confidence interval (p) is created for the standard deviation error that may occur in the LR feature selection method. It then tests whether the estimated coefficients for the LR method differ significantly from zero. If this value is above the specified confidence interval, it is considered as the real value by the LR. Parameter (σ) given in Eq. (8) estimates the standard deviation value in the LR method. If the standard deviation is small compared to the coefficient, the p interval narrows and the probability of detecting the actual value increases [45].

$$\widehat{\beta} \sim N(\beta, \sigma^2) \tag{8}$$

In this study, the LR method was used for feature selection and it was compiled in Python [46]. The deep feature set extracted from the fully connected layer (FC7—4096 features) of the VGG-16 was used. This feature set contains feature columns that represent each entry. The columns containing the best 1000 features were selected with the LR method and classified using the SVM method.

2.8 Proposed approach

The proposed approach is designed for use in detecting all four phases of Alzheimer's disease. The main idea of the proposed approach is based on the data enhancing by processing the original dataset with different methods and

techniques and its processing in deep networks. In this study, DeepDream, FCIE, and Hypercolumn techniques were used for data enhancing. Here, methods and techniques were used on the convolution model to measure success.

The proposed approach uses the three new datasets from the data enhancing phase in the VGG-16 model. The results (3 x 4096 features) extracted from the FC-7 layer of the VGG-16 model are combined forming a dataset with 12,288 features. This dataset is processed with the LR feature selection method to select the top 1000 features. In the last step of the proposed approach, 1000 features are classified with the SVM method as shown in Fig. 9.

3 Results

The necessary coding and analysis were performed using Python 3.6 with the Jupyter Notebook. Training of the VGG-16 model was carried out using MATLAB (2019b). Due to the inadequacy of the hardware unit, Google Colab was used in the compilation of Python codes. The hardware features offered by Google Colab were the Tesla K80 GPU graphics card, 12 GB DDR5 memory, and 2.30 GHz @ Intel© i5 Xeon (R) processor. In the compilation of MATLAB, the hardware with the following features was used, Windows 10 64-bit OS, 1 GB graphics card, 4 GB memory, and Intel© i5—Core 2.5 GHz processor.

The confusion matrix parameters were used to measure the analyzes performed in the experiment. Here, the formulas between Eqs. (9) and (13) were used to calculate the values of metrics. The metrics used in the experiment are Sensitivity (Se), Specificity (Sp), F-score (F-scr), Precision (Pre), and Accuracy (Acc). Besides, the parameters used in the confusion matrix are True-Positive (TP), True-Negative (TN), False-Positive (FP), and False-Negative (FN) [47, 48].

$$\mathrm{Se} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{9}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{10}$$

$$\mathrm{Pre} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{11}$$

$$\mathrm{F-scr} = \frac{2x\mathrm{TP}}{2x\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{12}$$

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{13}$$

In the first four steps of this study, 25% of the dataset was used as test data and 75% as training data. In the last step, the cross-validation method was used for the dataset and a cross-validation coefficient ($k = 10$) of ten was chosen. The epoch value was 75 for training at each step and the total number of iterations is 5625. In the first step, the original dataset was trained with the VGG-16 model and 4096 features of this model extracted from the FC-7 layer were classified by the SVM method. As a result of the classification, 99.10% accuracy in MID, 99.87% in MOD, 96.98% in NOD, and 96.61% in VMD were obtained. The overall accuracy was 96.31%. The training and validation accuracy and loss graphs of the first step are shown in Fig. 10, and the confusion matrix is shown in Fig. 11.

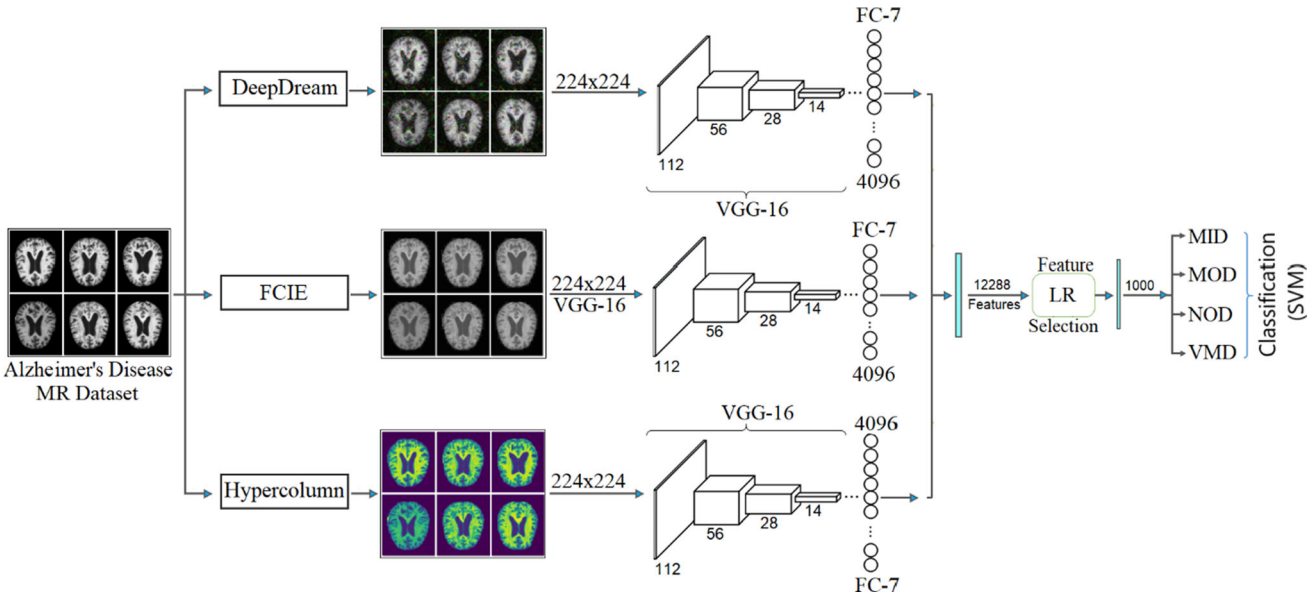

Fig. 9 The general design of the proposed approach

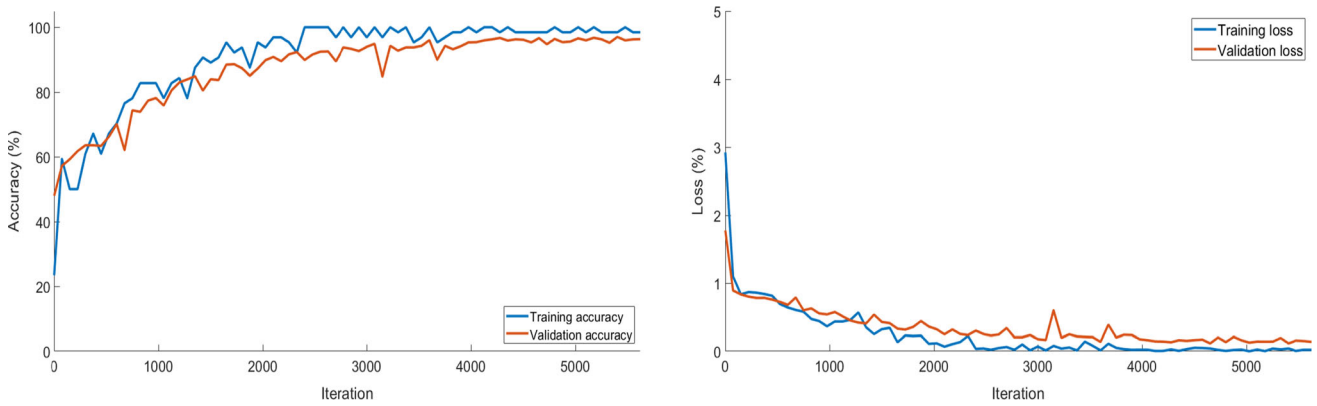

Fig. 10 Accuracy and loss graphs of the VGG-16 model using the original dataset

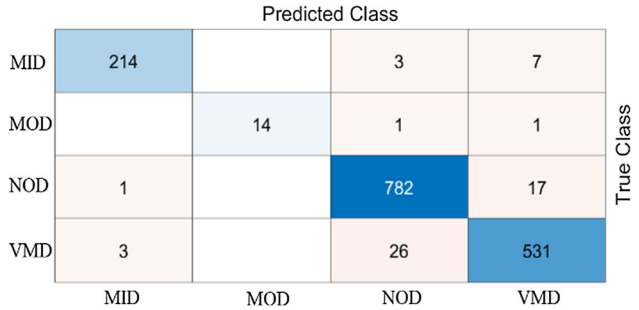

Fig. 11 Confusion matrix of the VGG-16 model using the original dataset

Besides, the analysis results of the first step are given in Table 2.

In the second step of the experiment, the original dataset was recreated with DeepDream, FCIE, and Hypercolumn methods. Three datasets were individually trained with the VGG-16 model. The deep features provided by the VGG-16 model were classified using the SVM method. The overall accuracy success in the dataset created with DeepDream was 82.0%. Overall accuracy success from data enhancing with the FCIE technique was 98.94%. And the accuracy success rate of the data enhancing with the Hypercolumn technique was 95.38%. In this step, it was observed that the success obtained from datasets processed with FCIE and Hypercolumn technique achieved a result close to the success achieved in the original dataset. Training accuracy graphs of data enhancing are shown in Fig. 12 and confusion matrices are shown in Fig. 13. The confusion metric values for this step are given in Table 3.

In the third step of the experiment, datasets were trained with the VGG-16 model as in the second step of the experiment. 4096 deep features extracted from the FC-7 layer of the VGG-16 was considered. Three feature sets obtained using DeepDream, FCIE, and Hypercolumn techniques were combined to create a new feature set with a total of 12,288 features. The combined feature set was then classified using the SVM. Here, as in the previous steps, the test data was set at 25%. The overall accuracy rate obtained as a result of the classification was 99.69%. The success achieved in the third step gave a more successful result than the first two steps. However, since the feature number of the dataset (12,288) contains too many features, it was necessary to re-classify by selecting efficient features.

In the fourth step of the experiment, the LR method was used as a feature selection method. Here, the feature dataset file was compiled with LR codes in Python and 12,288 features were sorted according to the LR method. The first 1000 features were selected from the ranked list and reclassified using the SVM method. As a result of the classification, an overall accuracy of 99.94% was achieved. The success achieved in the fourth step was better than the success achieved in the first three steps. Also, efficient features were selected in the fourth step, contributing to the proposed approach. With the proposed approach, a 100%

Table 2 Confusion matrix metric values of the VGG-16 model using the original dataset

Model and dataset type	AD phase	Se. (%)	Sp. (%)	Pre. (%)	F-Scr (%)	Acc. (%)	Overall Acc. (%)
VGG-16 and original data	Mild demented	95.54	99.70	98.17	96.83	99.10	96.31
	Moderate demented	87.50	100	100	93.33	99.87	
	Non-demented	97.75	96.20	96.31	97.02	96.98	
	Very mild demented	94.82	97.58	95.50	95.16	96.61	

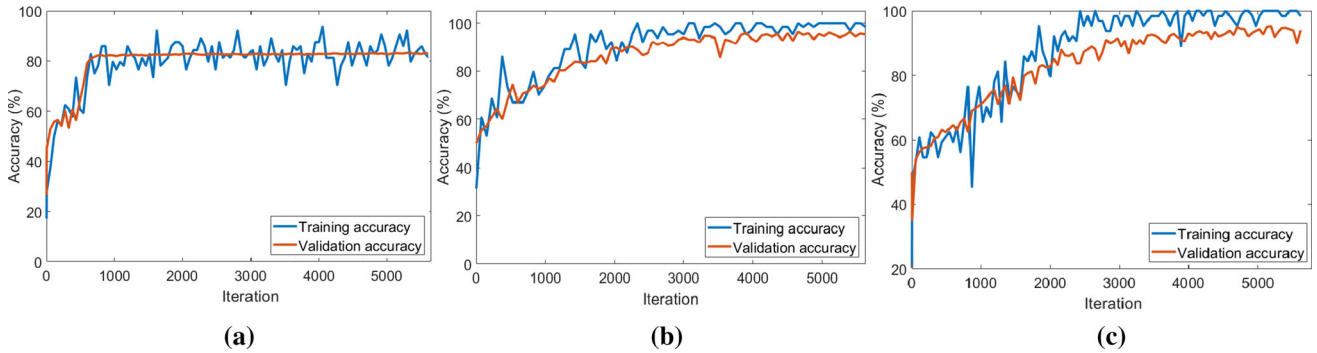

Fig. 12 Training accuracy graphs obtained by using data enhancing techniques and approach with the VGG-16 model; **a** DeepDream, **b** FCIE, **c** Hypercolumn

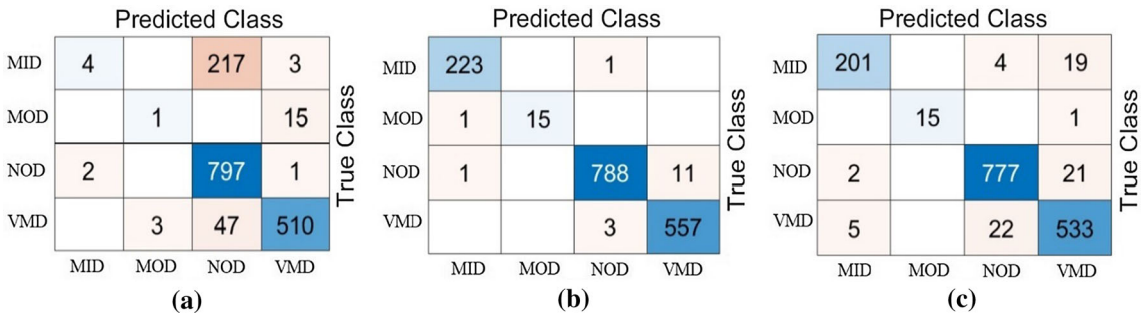

Fig. 13 Confusion matrices obtained by using data enhancing techniques with the VGG-16 model; **a** DeepDream, **b** FCIE, **c** Hypercolumn

Table 3 Confusion matrix metric values of the VGG-16 model using the dataset enhancing

Model and dataset type	Number of features	AD Phase	Se. (%)	Sp. (%)	Pre. (%)	F-Scr (%)	Acc. (%)	Overall Acc. (%)
VGG-16 and Data enhancing with DeepDream	4096	MID	1.78	99.85	66.67	3.47	85.53	82.00
		MOD	6.25	99.77	25.0	10.0	98.65	
		NOD	99.63	66.11	75.12	85.65	83.09	
		VMD	91.07	97.69	96.41	93.66	95.01	
VGG-16 and Data enhancing with FCIE	4096	MID	99.55	99.85	99.11	99.33	99.81	98.94
		MOD	93.75	100	100	96.77	99.93	
		NOD	98.50	99.49	99.49	98.99	98.99	
		VMD	98.46	98.93	98.06	98.75	99.12	
VGG-16 and Data enhancing with Hypercolumn	4096	MID	89.73	99.47	96.63	93.06	98.07	95.38
		MOD	93.75	100	100	96.77	99.93	
		NOD	97.13	96.64	96.76	96.94	96.89	
		VMD	95.18	96.03	92.86	94.01	95.73	

success was achieved from the classification of Mild Demented class; 99.94% success was achieved from the classification of Moderate Demented class; The accuracy rate in the classification of Non-Demented class was 100%, and the accuracy rate in the classification of Very Mild Demented class was 99.94%. The confusion matrices of the third and fourth steps are shown in Fig. 14. Furthermore, the Receiver Operating Characteristic (ROC) curve of the fourth step is shown in Fig. 14c. The metric results of the third and fourth steps in the confusion matrix are given in Table 4.

In the last step of this study, cross-validation method was used to check the validity and reliability of the proposed approach on the dataset, and to avoid problems such as overfitting or underfitting [49, 50]. The cross-validation coefficient value ten was chosen (k = 10). In the fifth step,

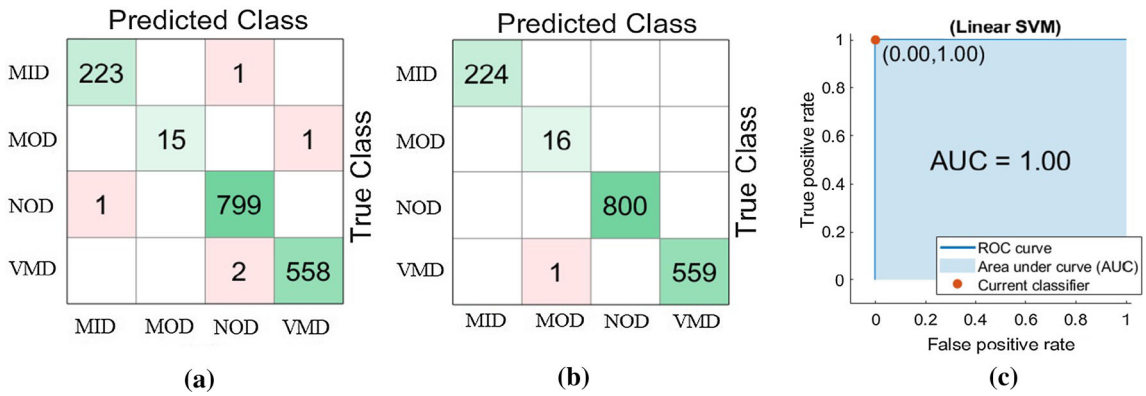

Fig. 14 Classification by SVM method; **a** confusion matrix of features obtained by combining datasets, **b** confusion matrix of combining datasets, after LR feature selection, and **c** ROC curve

Table 4 Analysis results of the proposed approach

Model and dataset type	Total features	Feature selection	AD phase	Se. (%)	Sp. (%)	Pre. (%)	F-Scr (%)	Acc. (%)	Overall Acc. (%)
VGG-16 and Combining of Datasets	12,288	No	MID	99.55	99.92	99.55	99.55	99.87	99.69
			MOD	93.75	100	100	96.77	99.93	
			NOD	99.87	99.62	99.62	99.75	99.74	
			VMD	99.64	99.90	99.82	99.73	99.81	
VGG-16 and Combining of Datasets & Feature Selection (Proposed Approach)	1000	Yes/LR	MID	100	100	100	100	100	99.94
			MOD	100	99.94	94.12	96.97	99.94	
			NOD	100	100	100	100	100	
			VMD	99.82	100	100	99.91	99.94	

the combined feature set (12,288 features) was processed with the cross-validation method and classified using the SVM method. The overall accuracy success achieved in the classification process was 99.66%. The time elapsed in the classification process was 2292 s. Then, the 1000-feature set selected according to the LR feature selection method was processed with the cross-validation method and classified with the SVM method. The overall accuracy success achieved in the classification process was 99.69%. The time elapsed in the classification process was 39 s. The confusion matrices of the fifth step is shown in Fig. 15. Furthermore, the ROC curve of the fifth step is shown in Fig. 15c. The analysis results obtained by the cross-validation method are given in Table 5. Accuracy success and time savings obtained in the fifth step showed the success of the proposed approach. Stable analysis of the proposed approach has been confirmed by the cross-validation method.

4 Discussion and conclusion

The symptoms of AD worsen over time, but the progression of the disease changes. On average, a person with Alzheimer's lives four to eight years after diagnosis [51]. In this study, dementia phases of AD patients were investigated and here, brain MR images of AD are used. Determining the phase of a person with Alzheimer's is a difficult process for experts. Here, we suggested using the deep learning model and image processing techniques together to improve the decision-making process of experts and to facilitate this difficult process. We aimed to increase the performance of the VGG-16 deep learning model we selected in the study. Here, instead of the VGG-16 model, it could be chosen in different convolutional models. The success rate of the VGG-16 model in its original dataset was 96.31%. We increased this success to 98.94% in the datasets enhancing with the FCIE technique. And the FCIE technique has been shown to contribute to improving the prediction success of the proposed approach. Here, we observed that the DeepDream method did not meet the expectations. However, we still included it in the processes of the experiment. We can explain this situation as follows;

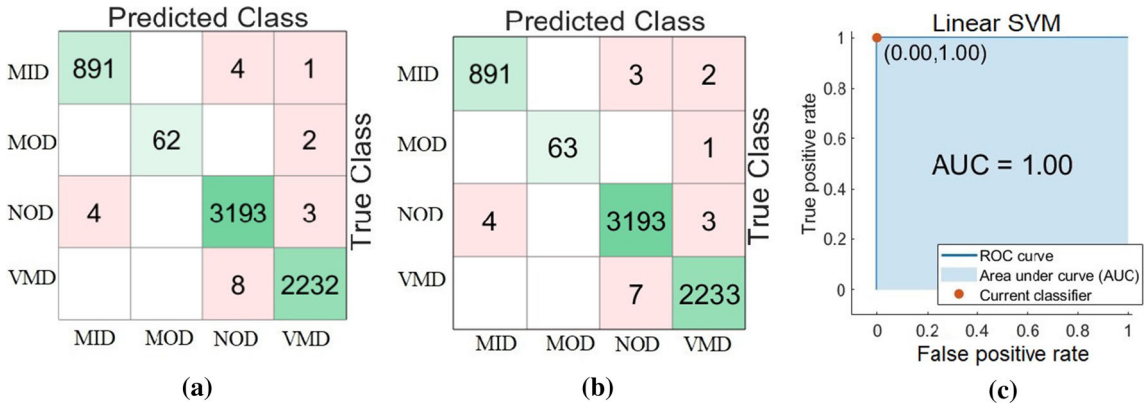

Fig. 15 Classification by SVM method using cross-validation method; **a** confusion matrix of features obtained by combining datasets, **b** confusion matrix of combining datasets, after LR feature selection, and **c** ROC curve

Table 5 Analysis results obtained using the cross-validation method of the proposed approach (k = 10)

Model and dataset type	Total features	Feature selection	AD phase	Se. (%)	Sp. (%)	Pre. (%)	F-Scr (%)	Acc. (%)	Overall Acc. (%)
VGG-16 and Combining of Datasets	12,288	No	MID	99.44	99.92	99.55	99.49	99.86	99.66
			MOD	96.87	100	100	98.41	99.97	
			NOD	99.78	99.62	99.62	99.70	99.70	
			VMD	99.64	99.73	99.73	99.68	99.78	
VGG-16 and Combining of Datasets and Feature Selection (Proposed Approach)	1000	Yes / LR	MID	99.44	99.93	99.55	99.50	99.86	99.69
			MOD	98.44	100	100	99.21	99.98	
			NOD	99.78	99.69	99.69	99.73	99.73	
			VMD	99.69	99.86	99.73	99.71	99.80	

we applied in the last step of the experiment, perhaps considering the possibility of giving efficient features. As a result, FCIE and Hypercolumn techniques significantly contributed to the experimental process. Besides, we did not use the features obtained from the original image data in the combined feature sets. The aim here was to successfully classify the stages of AD with obtained deep features. It has been confirmed in the experimental analysis of this study that MR images yielded successful results in the detection of AD stages with data enhancing techniques and approaches.

AD occurs as a result of the loss of cognitive functions in the brain, causes near-forgetfulness in the case and dementia in subsequent processes. The prevalence rate of AD is increasing the day by day. With the proposed approach, the AD stages were determined using the data obtained from AD patients. With this approach, the data enhancing of MR images, which are seen as preprocess steps, was realized. Here, enhancing MR images using the Hypercolumn and FCIE techniques were observed as useful. One of the basic aims was to understand whether the dataset enhancing contributes to model performance. This subject was validated considering the results. Also, another significant aim of this study is to select more efficient features by combining feature sets. To this aim, the LR feature selection method was employed in the fourth step of the experiment. This process resulted in an overall accuracy of 99.94%. The proposed approach relied upon the data enhancing MR images in this study produced promising results in determining AD stages. However, not providing any statistical info about the patient information of the images in the dataset raises doubts about the accuracy of the approach we recommend. No statement about this situation was made by the researcher who provided the data. Therefore the thought we hesitated, there may be more than one image obtained from the same Alzheimer patient. This situation can increase the training and testing success of the proposed approach and, overfitting learning may have occurred. Although we have used the cross-validation method we performed at the last step of the experiment to minimize overfitting learning, this situation cannot completely eliminate the hesitations in the dataset. In addition, the images in the dataset are not of high

quality, which may negatively affect the success of the proposed approach.

In the future study, we hope to improve the performance of the proposed approach on different datasets by using attention modules (regional focus) on images.

Funding There is no funding source for this article.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest related to this paper.

Ethical approval This article does not contain any data, or other information from studies or experimentation, with the involvement of human or animal subjects.

References

1. Weller J, Budson A (2018) Current understanding of Alzheimer's disease diagnosis and treatment. F1000Research 7:F1000 Faculty Rev-1161. https://doi.org/https://doi.org/10.12688/f1000research.14506.1
2. Weller J, Budson A (2019) 2019 Alzheimer's disease facts and figures. Alzheimer's Dement 15:321–387. https://doi.org/10.1016/j.jalz.2019.01.010
3. Qiu S, Heydari MS, Miller MI et al (2019) P1–119: enhancing deep learning model performance for AD diagnosis using ROI-based selection. Alzheimer's Dement 15:P280–P281. https://doi.org/10.1016/j.jalz.2019.06.674
4. Basaia S, Agosta F, Wagner L et al (2019) Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks. NeuroImage Clin 21:101645. https://doi.org/10.1016/j.nicl.2018.101645
5. Liu M, Li F, Yan H et al (2020) A multi-model deep convolutional neural network for automatic hippocampus segmentation and classification in Alzheimer's disease. Neuroimage 208:116459. https://doi.org/10.1016/j.neuroimage.2019.116459
6. Lu X, Wu H, Zeng Y (2019) Classification of Alzheimer's disease in MobileNet. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/1345/4/042012
7. Shen T, Jiang J, Lu J et al (2019) Predicting alzheimer disease from mild cognitive impairment with a deep belief network based on 18F-FDG-PET images. Mol Imag 18:1–9. https://doi.org/10.1177/1536012119877285
8. Xiao Z, Ding Y, Lan T et al (2017) Brain MR image classification for alzheimer's disease diagnosis based on multifeature fusion. Comput Math Methods Med 2017:1952373. https://doi.org/10.1155/2017/1952373
9. Jo T, Nho K, Saykin AJ (2019) Deep learning in alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data. Front Aging Neurosci 11:220. https://doi.org/10.3389/fnagi.2019.00220
10. Menger V, Scheepers F, Spruit M (2018) Comparing deep learning and classical machine learning approaches for predicting inpatient violence incidents from clinical text. Appl Sci 8:981. https://doi.org/10.3390/app8060981
11. Nguyen G, Dlugolinsky S, Bobák M et al (2019) Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artif Intell Rev 52:77–124. https://doi.org/10.1007/s10462-018-09679-z
12. Sze V, Chen YH, Yang TJ, Emer JS (2017) Efficient processing of deep neural networks: a tutorial and survey. Proc IEEE 105:2295–2329. https://doi.org/10.1109/JPROC.2017.2761740
13. Dubey S (2020) Alzheimer's Dataset four class of Images. In: Kaggle. https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images/data. Accessed 1 Mar 2020
14. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations
15. Toğaçar M, Ergen B (2018) Deep learning approach for classification of breast cancer. In: 2018 International conference on artificial ıntelligence and data processing (IDAP). pp 1–5
16. Guan Q, Wang Y, Ping B et al (2019) Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study. J Cancer 10:4876–4882. https://doi.org/10.7150/jca.28769
17. Cengil E, Çınar A (2016) A new approach for image classification: convolutional neural network. Eur J Tech EJT 6:96–103
18. Demir F, Şengür A, Bajaj V, Polat K (2019) Towards the classification of heart sounds based on convolutional deep neural network. Heal Inf Sci Syst 7:16. https://doi.org/10.1007/s13755-019-0078-0
19. Litjens G, Kooi T, Bejnordi BE et al (2017) A survey on deep learning in medical image analysis. Med Image Anal 42:60–88. https://doi.org/10.1016/j.media.2017.07.005
20. Başaran E, Cömert Z, Çelik Y (2020) Convolutional neural network approach for automatic tympanic membrane detection and classification. Biomed Signal Process Control 56:101734. https://doi.org/10.1016/j.bspc.2019.101734
21. Deniz E, Sengür A, Kadiroglu Z et al (2018) Transfer learning based histopathologic image classification for breast cancer detection. Heal Inf Sci Syst 6:18. https://doi.org/10.1007/s13755-018-0057-x
22. Lu S, Lu Z, Zhang Y-D (2018) Pathological brain detection based on AlexNet and transfer learning. J Comput Sci. https://doi.org/10.1016/j.jocs.2018.11.008
23. Gu J, Wang Z, Kuen J et al (2018) Recent advances in convolutional neural networks. Pattern Recognit 77:354–377. https://doi.org/10.1016/j.patcog.2017.10.013
24. Sertkaya ME, Ergen B, Togacar M (2019) Diagnosis of eye retinal diseases based on convolutional neural networks using optical coherence ımages. In: 2019 23rd ınternational conference electronics, pp 1–5
25. Celik Y, Talo M, Yildirim O et al (2020) Automated invasive ductal carcinoma detection based using deep transfer learning with whole-slide images. Pattern Recognit Lett 133:232–239. https://doi.org/10.1016/j.patrec.2020.03.011
26. Ma Y, Zhang Q, Li D, Tian Y (2019) Linex support vector machine for large-scale classification. IEEE Access 7:70319–70331. https://doi.org/10.1109/access.2019.2919185
27. Bisgin H, Bera T, Ding H et al (2018) Comparing SVM and ANN based machine learning methods for species identification of food contaminating beetles. Sci Rep 8:1–12. https://doi.org/10.1038/s41598-018-24926-7
28. Toğaçar M, Ergen B, Cömert Z (2020) Waste classification using AutoEncoder network with integrated feature selection method in convolutional neural network models. Measurement 153:107459. https://doi.org/10.1016/j.measurement.2019.107459
29. Awad M, Khanna R (2015) Support vector machines for classification BT—efficient learning machines: theories, concepts, and applications for engineers and system designers. In: Awad M, Khanna R (eds). Apress, Berkeley, CA, pp 39–66
30. Doğan Ü, Glasmachers T, Igel C (2016) A unified view on multi-class support vector classification. J Mach Learn Res 17:1–32

31. (2020) DeepDream | TensorFlow Core. In: TensofFlow. https://www.tensorflow.org/tutorials/generative/deepdream. Accessed 30 Aug 2020
32. Suzuki K, Roseboom W, Schwartzman DJ, Seth AK (2017) A deep-dream virtual reality platform for studying altered perceptual phenomenology. Sci Rep 7:1–11. https://doi.org/10.1038/s41598-017-16316-2
33. Bardak T, Bardak S (2017) Prediction of wood density by using red-green-blue (RGB) color and fuzzy logic techniques. J Polytech 20:979–984. https://doi.org/10.2339/politeknik.369132
34. Arnal J, Súcar L (2020) Hybrid filter based on fuzzy techniques for mixed noise reduction in color images. Appl Sci. https://doi.org/10.3390/app10010243
35. Yun HJ, Wu ZY, Wang GJ et al (2016) A novel enhancement algorithm combined with improved fuzzy set theory for low illumination images. Math Probl Eng. https://doi.org/10.1155/2016/8598917
36. Patrascu V (2019) Fuzzy color ımage enhancement algorithm. In: Github. https://github.com/WaseemKn/FuzzyColorImageEnhancement-FuzzyLogicCourse-ITE5thYear. Accessed 1 Mar 2020
37. Pilly PK, Stepp ND, Liapis Y et al (2019) Hypercolumn sparsification for low-power convolutional neural networks. J Emerg Technol Comput Syst 15:20. https://doi.org/10.1145/3304104
38. Hariharan B, Arbeláez P, Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. https://doi.org/10.1109/cvpr.2015.7298642
39. Sreeraman R (2018) Hypercolumn test VGG16. In: Kaggle. https://www.kaggle.com/rupeshs/hypercolumn-test-vgg16. Accessed 1 Sep 2020
40. Hariharan B, Arbeláez P, Girshick R, Malik J (2017) Object instance segmentation and fine-grained localization using hypercolumns. IEEE Trans Pattern Anal Mach Intell 39:627–639. https://doi.org/10.1109/tpami.2016.2578328
41. Gueorguieva N, Valova I, Klusek D (2020) Solving large scale classification problems with stochastic based optimization. Procedia Comput Sci 168:26–33. https://doi.org/10.1016/j.procs.2020.02.247
42. Rzecki K, Sośnicki T, Baran M, Niedźwiecki M, Król M, Łojewski T, Acharya UR, Yildirim Ö, Pławiak P (2018) Application of computational intelligence methods for the automated identification of paper-ink samples based on LIBS. Sensors 18:3670. https://doi.org/10.3390/s18113670
43. Zhuo L, Zhang B, Chen C et al (2019) Calibrated stochastic gradient descent for convolutional neural networks. Proc AAAI Conf Artif Intell 33:9348–9355. https://doi.org/10.1609/aaai.v33i01.33019348
44. Schneider A, Hommel G, Blettner M (2010) Linear regression analysis: part 14 of a series on evaluation of scientific publications. Dtsch Arztebl Int 107:776–782. https://doi.org/10.3238/arztebl.2010.0776
45. Hoffman JIE (2019) Chapter 27—linear regression. In: Hoffman JIE (ed) Basic biostatistics for medical and biomedical practitioners, 2nd edn. Academic Press, Cambridge, pp 445–489
46. (2020) Feature selection: linear regression. In: Scikit - Learn. https://scikit-learn.org/stable/modules/feature_selection.html. Accessed 3 Mar 2020
47. Abdullah AO, Ali MA, Karabatak M, Sengur A (2018) A comparative analysis of common YouTube comment spam filtering techniques. In: 2018 6th ınternational symposium on digital forensic and security (ISDFS), pp 1–5
48. Günay M, Köseoğlu M, Yıldırım Ö (2020) Classification of hand-drawn basic circuit components using convolutional neural networks. In: 2020 ınternational congress on human-computer ınteraction, optimization and robotic applications (HORA), pp 1–5
49. Parvandeh S, Yeh H-W, Paulus MP, McKinney BA (2020) Consensus features nested cross-validation. bioRxiv 2019.12.31.891895. https://doi.org/https://doi.org/10.1101/2019.12.31.891895
50. Yadav S, Shukla S (2016) Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In: 2016 IEEE 6th ınternational conference on advanced computing (IACC), pp 78–83
51. (2020) Alzheimer's Stages - Early, Middle, Late Dementia Symptoms | alz.org. In: Alzheimer's Assoc. https://www.alz.org/alzheimers-dementia/stages. Accessed 4 Mar 2020