Neural Computing and Applications
https://doi.org/10.1007/s00521-021-05799-w

ORIGINAL ARTICLE

A CNN based framework for classification of Alzheimer's disease

Yousry AbdulAzeem[1] · Waleed M. Bahgat[2,3] · Mahmoud Badawy[3,4]

Received: 11 March 2020 / Accepted: 5 February 2021

Abstract

In the current decade, advances in health care are attracting widespread interest due to their contributions to people longer surviving and fitter lives. Alzheimer's disease (AD) is the commonest neurodegenerative and dementing disease. The monetary value of caring for Alzheimer's disease patients is involved to rise dramatically. The necessity of having a computer-aided system for early and accurate AD classification becomes crucial. Deep-learning algorithms have notable advantages rather than machine learning methods. Many recent research studies that have used brain MRI scans and convolutional neural networks (CNN) achieved promising results for the diagnosis of Alzheimer's disease. Accordingly, this study proposes a CNN based end-to-end framework for AD-classification. The proposed framework achieved 99.6%, 99.8%, and 97.8% classification accuracies on Alzheimer's disease Neuroimaging Initiative (ADNI) dataset for the binary classification of AD and Cognitively Normal (CN). In multi-classification experiments, the proposed framework achieved 97.5% classification accuracy on the ADNI dataset.

Keywords AD-classification · Convolutional neural network (CNN) · Magnetic resonance imaging (MRI) · Adaptive momentum estimation (Adam) · Glorot uniform weight initializer

1 Introduction

Alzheimer's Disease (AD) is a progressive brain disease. It is a neurological disorder in which the death of brain cells causes memory loss and cognitive decline. Also, it is considered the most common type of dementia and has an incredibly negative impact on the individual and social life of humans [31, 33, 42]. According to recent statistics, there are more than 46.8 million people now living with dementia, 44 million diagnosed with Alzheimer's. The number will be increased to 131.5 million in 2050 [31]. Mild Cognitive Impairment (MCI) is a transitional state from Cognitively Normal (CN) to dementia, which has a 10% conversion rate to AD [11].

AD-related neuropathological markers are investigated many years before clinical manifestation of memory symptoms [10, 13, 36], which suggests that AD development could be predicted before clinical onset via in vivo biomarker analysis. PET and MR imaging in addition to blood or cerebrospinal fluid (CSF) are examples for biomarkers [4, 17, 26, 38]. MRI is commonly used in AD diagnosis and classification. MRI measures have many advantages rather than the compared methods. For example, it does not use ionizing radiation, of being noninvasive, less expensive, and more widely spread in most of the medical environments. Besides, MRI markers are capable of gathering multimodal information within the same scanning session.

Previous studies for classification of AD patients have used several machine learning methods applied to structural MRI [32]. Support Vector Machine (SVM) is the most commonly used machine learning method [32]. SVM extracts high-dimensional, informative features from MRI to build the classification models that automate the AD diagnosis. Classification research studies that use machine

✉ Yousry AbdulAzeem
yousry@mans.edu.eg

[1] Computer Engineering Department, Misr Higher Institute for Engineering and Technology, Mansoura, Egypt

[2] Information Technology Department, Faculty of Computer and Information Sciences, Mansoura University, Mansoura, Egypt

[3] Department of Computer Science and Informatics, Taibah University Al Medina Al Munawara, Medina, Saudi Arabia

[4] Computer and System Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt

Published online: 07 March 2021

learning consists of four steps: feature extraction, feature selection, dimensionality reduction, and feature-based classification algorithm selection. This approach has many drawbacks since it needs complex image pre-processing, which consumes much time and demands heavy computations [21]. In addition, the reproducibility of these approaches is also considered as a challenge's issue [34].

Deep-learning algorithms have notable advantages rather than conventional machine learning methods. For example, they do not need image pre-processing and can automatically get an optimal representation of the data from the raw images without requiring prior feature selection. This leads to having less time consuming, more objective, and less bias-prone processes [23, 39]. According to the previous discussion, deep-learning algorithms well suits in dealing with large-scale, high-dimensional medical imaging analysis [30]. The research studies showed that Convolutional Neural Networks (CNN), which is considered a deep-learning approach, outperforms the existing machine learning approaches [23]. A typical CNN consists of three main layers, the Convolutional layer, the Pooling layer, and the Fully connected layer as shown in Fig. 1.

This paper proposes a CNN based end-to-end framework with detailed steps starting from image acquisition landing at AD-classification to classify scanned MRI images to predict whether they have Alzheimer's or not, and to which degree, using a machine learning application with the help of digital image processing. The proposed framework achieves the following:

- Applying adaptive thresholding in the digital image processing stage. Most of the state-of-art techniques used the conventional thresholding operator which uses a global threshold for all pixels. On the other hand, in adaptive thresholding, the threshold is dynamically changed over the image. This can handle the change of lighting conditions in the image.
- Performing data augmentation to expand the size of a training dataset by creating modified versions of images in the dataset. This should improve the trained model if the available samples are relatively small. It should increase the accuracy of the framework and decrease the overfitting.
- Initializing the network weights using Glorot Uniform weight initializer [15]. This enables initializing the weights of the network in such a way the neuron activation functions are not starting in saturated or dead regions. This leads to achieving quicker convergence and higher accuracy.
- Using Adaptive Momentum Estimation (Adam) optimizer in the optimization process. It is appropriate for dealing with neuroimaging data since it perfects match with sparse gradients in noisy environments. Applying Adam optimizer in AD images achieves quicker convergence.

ADNI datasets are used to verify the proposed framework. The simulation results show that the classification accuracy of the proposed framework outperforms the other state-of-the-art approaches.

1.1 Related work

There are two levels of AD-classification; binary-classification in which the technique specifies CN against AD or MCI and multi-classification which classifies the disease in which level (CN, AD, MCI,.. etc). Table 1 shows the most recent techniques in AD-classification, specifying classification technique, class, dataset, and accuracy of detection. Approaches [2, 3, 7, 8, 14, 37, 43] use traditional Computer vision techniques, while approaches [1, 5, 19, 24, 25, 29, 35, 41] use deep CNN for achieving the whole process.

In [43], they proposed a multivariate approach employing wavelet entropy and predator-prey Particle swarm for AD-classification. Single-hidden-layer Neural Network was used as the classifier. They achieved about 92.73% of accuracy for binary classification (AD vs. CN). In [8] images are segmented into Gray Matter (GM), White

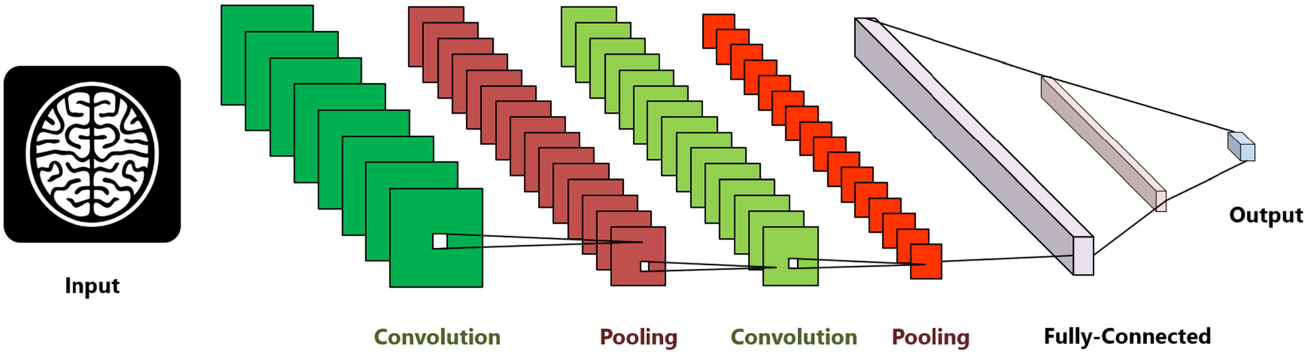


Fig. 1 A typical CNN for image processing

Table 1 Recent AD-classification techniques

Approach	Year	Technique	Dataset	Classification	Accuracy %
Zhang et al. [43]	2017	Multivariate Approach	OASIS	AD vs. CN	92.73
Beheshti et al. [8]	2017	SVM	ADNI	AD vs. CN	84.07
Beheshti et al. [7]	2017	SVM	ADNI	AD vs. CN	93.01
				pMCI vs. sMCI	75
Sorensen et al. [37]	2017	LDA	ADNI, AIBL CADDementia	CN vs. MCI vs. AD	62.7
Asim et al. [3]	2018	SVM	ADNI	AD vs. CN	94
				MCI vs. CN	76.5
				AD vs. MCI	75.5
Altaf et al. [2]	2018	SVM KNN Decision Tree Ensemble	ADNI	AD vs. CN	98.4
				AD vs. MCI	81.2
				MCI vs. CN	86.7
				CN vs. MCI vs. AD	79.8
Duraisamy et al. [14]	2018	based Weighted Probabilistic Neural Network	ADNI	AD vs. CN	98.63
				MCI vs. CN	95
				AD vs. MCI	96.4
Payan et al. [29]	2015	CNN	ADNI	CN vs. MCI vs. AD	85.53
				AD vs. CN	95.39
				AD vs. MCI	82.24
				MCI vs. CN	90.13
Sarraf and Tofighi [35]	2016	CNN	ADNI	AD-CN	98.84
Wang et al. [41]	2018	CNN	OASIS	AD vs. CN	96.43
Liu et al. [25]	2018	CNN	ADNI	AD vs. CN	93.26
				MCI vs. CN	74.34
Aderghal et al. [1]	2018	CNN	ADNI	AD vs. CN	92.5
				AD vs. MCI	85
				MCI vs. CN	80
Jain Rachana et al. [19]	2019	CNN	ADNI	CN vs. MCI vs. AD	95.73
				AD vs. CN	99.14
				AD vs. MCI	99.30
				MCI vs. CN	99.22
Bumshik Lee et al. [24]	2019	CNN	OASIS ADNI	AD vs. CN	95.35
				AD vs. CN	98.74
				CN vs. MCI vs. AD	98.06
Basaia et al. [5]	2019	CNN	ADNI	AD vs. CN	99.2
				c-MCI vs. CN	87.1
				AD vs. c-MCI	75.4

Matter (WM), and (CSF). Similarity matrices are built using the GM-segmented Regions of Interest (ROI). The AD-classification is based on the feature-extraction of the GM similarity matrix. To improve the accuracy of AD-classification, the Functional Activities Questionnaire (FAQ) and SVM are combined. The accuracy achieved is 84.07% for binary classification (AD vs. CN).

Brain dividing based on different atlases and combining features extracted from these anatomical parcellations is considered in [3]. They used baselines images of structured

MRI (sMRI) and F-fluorodeoxyglucose positron emission tomography (FDG-PET) to calculate average GM density and average relative cerebral metabolic rate for glucose Principal Components Analysis (PCA) was used to reduce the dimensionality of the features. SVM is used for binary classification between each couple of statuses with accuracies 94% for AD versus CN , 76.5% for MCI versus CN, and 75.5% for AD versus MCI.

Clinical data alongside with MRI are used to generate a hybrid feature vector [2]. MRIs are segmented into three regions GM, WM, and CSF. Texture features are extracted from both whole and segmented MRIs using different techniques. Clinical features are extracted using techniques including FAQ, Neuropsychiatric inventory and Geriatric depression scale. Both texture and clinical features are taken as input for AD-classification. Classifiers such as SVM, Ensemble, KNN, and Decision Trees are used to carry on binary and multiclass classification. Accuracies achieved are 98.4% for AD versus CN, 81.2% for AD versus MCI, 86.7% for MCI versus CN, and 79.8% for multiclass.

In [14] a combination between Fuzzy C-means and Weighted Probabilistic Neural Network is used for classification. The framework begins with extracting ROIs related to Hippo-Campus and Posterior Cingulate Cortex from brain images. Suspicious samples are removed from the training data to enhance classification performance. The accuracies achieved are 98.63%, 95.4%, and 96.4% for AD versus CN, MCI versus CN, and AD versus MCI respectively.

Convolution Neural Networks (CNN) are used to build a learning algorithm to classify MRI images [29]. 3D Convolutions are applied to the whole MRI image. Classification results obtained using a 3-way classifier (CN vs. AD vs. MCI) with an accuracy of 89.47. Binary classifiers achieved 95.39%, 92.11%, and 86.84% for AD versus CN, MCI versus CN, and AD versus MCI respectively. In [35] deep-learning algorithm is used to classify AD subjects. Scale and sift-invariant low to high-level features are extracted from a massive volume of whole-brain data using CNN architecture. Binary classification (AD vs. CN) achieved 98.4% of accuracy. A deep-learning approach based on CNN is proposed to detect AD subjects [41]. Leaky Rectified Linear unit and max pooling are used in designing the CNN. The approach achieved an accuracy of 97.65% in binary classification (AD vs. CN).

In [25] MRI and PET images are used as input to cascaded CNNs to classify AD, MCI, and CN cases. Multiple 3D-CNNs are constructed to extract features from local brain images. An upper high-level 2D-CNN followed by softmax layer generates multimodal correlation features from extracted features. Binary classification is performed to classify the disease case (AD, MCI) from CN. The accuracy of (AD vs. CN) is 93.26% and for (MCI vs. CN) 74.34%. Cross-modal transfer learning from structural MRI to Diffusion Tensor Imaging modality is used for AD-classification [1]. The combined modalities are taken as input to multi CNNs to perform the classification process. The method achieved accuracies of 92.5%, 85.0% and 80.0% for AD versus CN, AD versus MCI, and MCI versus CN respectively.

Jain et al. [19] proposed a CNN transfer learning architecture for AD diagnosis. They used the pre-trained VGG-16 to classify AD. Their experiment results are based on (ADNI) database. The 3-way classification accuracy of their work is 95.73% for the validation set.

Lee et al. [24] proposed a deep CNN data permutation scheme for classification AD using sMRI. They proposed slice selection to achieve the benefits of AlexNet. Their experimental results showed that their data permutation scheme improved the overall classification accuracies for AD classification. The classification accuracies for both binary and ternary classification on ADNI datasets are 98.74% and 98.06% respectively on the ADNI dataset.

Basaia et al. [5] used 3D sMRI (T1) and built a CNN-based technique to predict MCI that will be converted to AD in a specific period. They showed that their technique is dataset independent. Additionally, it showed their technique is capable of differentiating AD, MCI patients from CN.

The rest of this paper is organized as follows: The proposed framework is described in Sect. 2. Section 3 presents the experimental results. Finally, in Sect. 4, the paper is concluded.

2 Materials and methods

This paper proposes an end-to-end framework for AD-classification based on CNN. The framework consists of five main layers, as shown in Fig. 2, each layer contains its steps and algorithms. The layers of the framework are: (1) Acquisition and Annotation, (2) Preprocessing and Augmentation, (3) Cross-validation, (4) CNN model, and (5) AD-classification.

The main contributions of this framework are:

- Applying adaptive thresholding in the second layer.
- Performing data augmentation in the second layer.
- Initializing the network weights using Glorot Uniform weight initializer in the fifth layer.
- Using Adam optimizer in the optimization process in the fifth layer.

In the first layer, named Acquisition and Annotation, MRI images are acquired in files with an ”*.nii” extension. Each one of the 3D images is converted to a set of ”*.png” files

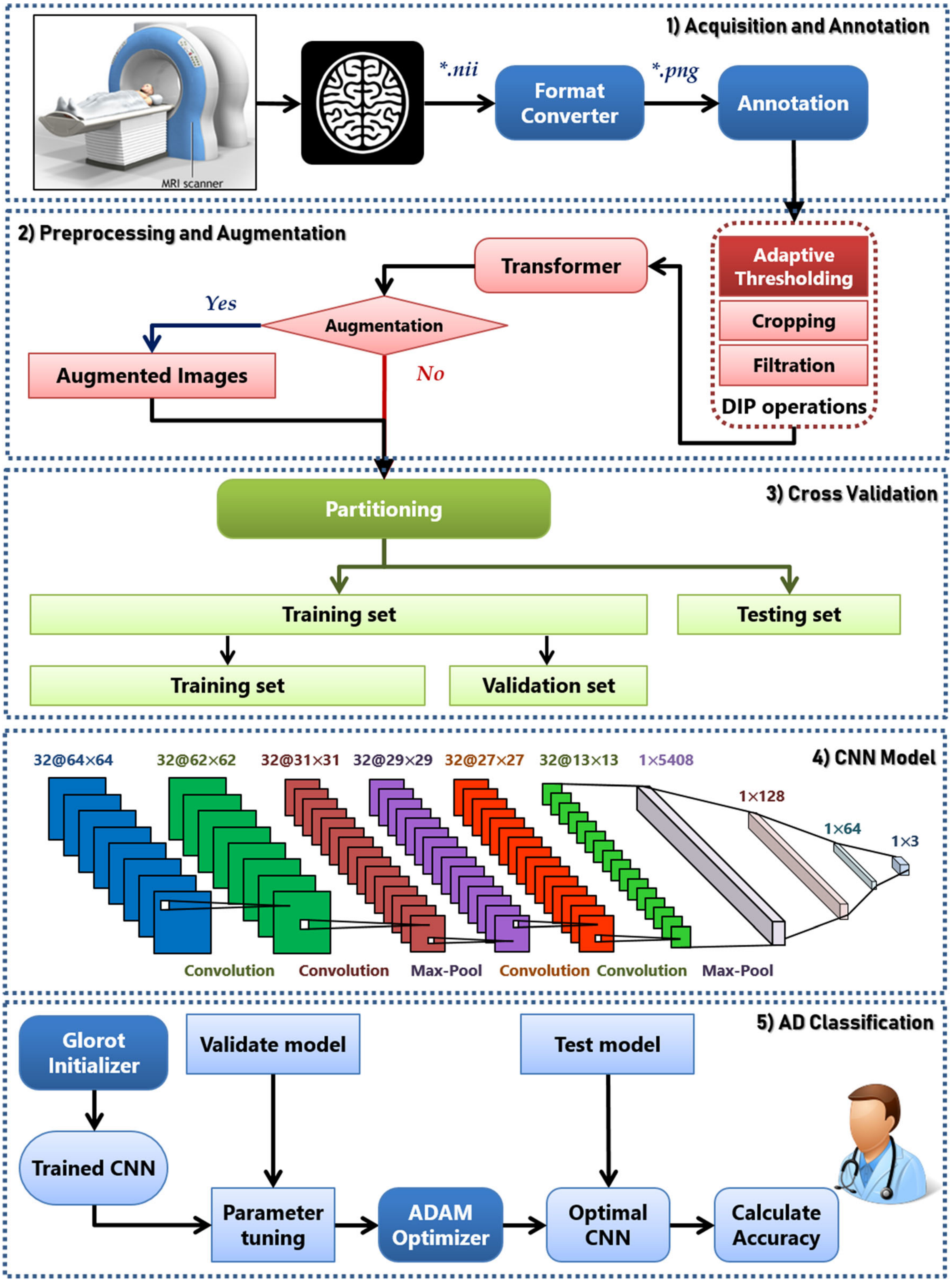


Fig. 2 The proposed framework

through format converter. Only images that contain the full shape of the brain are used and other images are ignored. Images are then annotated with classes to facilitate future processing. A sample of the annotation file (10 records) is shown in Table 2.

The second layer ”Preprocessing and Augmentation” takes the annotated im- ages as input. It starts with gray-

Table 2 A sample of the annotation file

Subject ID	Sex	DX Group	Age	Description	Structure
002_S_0413	F	Normal	76.4	MPR; ; N3; Scaled ← MPRAGE SENS	Brain
002_S_0559	M	Normal	79.5	MPR-R; ; N3 ← MPRAGE SENS	Brain
002_S_0559	M	Normal	79.5	MPR-R; ; N3 ← MPRAGE SENS	Brain
002_S_0559	M	Normal	79.5	MPR; ; N3 ← MPRAGE SENS	Brain
002_S_0559	M	Normal	79.5	MPR; ; N3; Scaled_2 ← MPRAGE SENS	Brain
002_S_0559	M	Normal	79.5	MPR; ; N3; Scaled ← MPRAGE SENS	Brain
002_S_0559	M	Normal	79.5	MPR; ; N3 ← MPRAGE SENS	Brain
002_S_0816	M	AD	71	MPR; ; N3; Scaled ← MPRAGE	Brain
002_S_0816	M	AD	71	MPR; ; N3 ← MPRAGE	Brain
002_S_0816	M	AD	71	MPR; ; N3; Scaled_2 ← MPRAGE	Brain
002_S_0816	M	AD	71	MPR-R; ; N3 ← MPRAGE REPEAT	Brain

scale conversion then adaptive thresholding. Most of the state-of-art techniques use the conventional thresholding operator which uses a global threshold for all pixels. On the other hand, in adaptive thresholding, the threshold is dynamically changed over the image. The threshold value at each pixel location depends on the neighboring pixel intensities. For each pixel, the threshold value is calculated. If the calculated value is below the threshold it is considered as a background value, otherwise, it is considered as foreground value.

The next digital image processing steps are cropping and filtration. After the filtration process, gray-scale images of size (256, 256) are generated. Figure 3 shows a sample image from the generated images. A transformer is used to resize the produced images to (128, 128) and (64, 64).

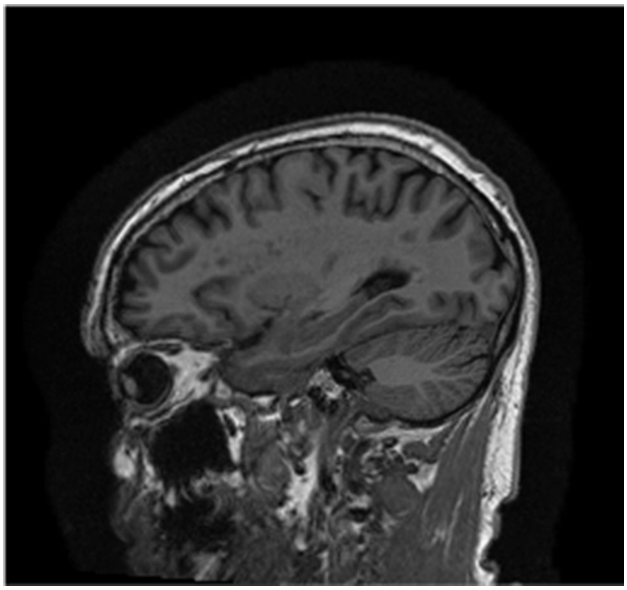


Fig. 3 A sample image from the generated images

If data augmentation is required then the data augmentation generator is initialized. Among several augmentation factors, the framework applies horizontal flipping, shearing, shifting, rotating, and zooming. Data augmentation helps in creating new and different training examples to improve the trained network. Algorithm 1 is the core engine in the second layer.

```
Algorithm 1: Data Compilation and Augmentation Algorithm
Input: $Cfile, (w, h)$ // annotation CSV file, image size
Output: $Pfile$ // Generated Pickle file
Images ← read(Cfile)
foreach $i \in Images$ do
    $c_i \leftarrow annotation(i)$
    $g_i \leftarrow gscale(i)$
    $r_i \leftarrow resize(g_i, w, h)$
    $Images \leftarrow r_i$
    if $augmentation$ then
        $augmentation\_generator()$
        $aImages \leftarrow augment(h_i)$
        $Images \leftarrow aImages$
    end
end
store the processed images into $Pfile$
return $Pfile$
```

The third layer in the framework is the cross-validation strategy used to train the CNN. Data are divided, randomly, into three sets; training set, validation set, and testing set. The whole dataset is divided into (95%) training set and (5%) testing set. The training set is further divided into (90%) training and (10%) validation sets. The main objective of cross-validation is obtaining the best values for training parameters to avoid overfitting.

The fourth layer represents the CNN model used in the study. The CNN architecture consists of three convolutional layers and max-pooling is performed after each convolutional layer. Several parameters have an impact on the model such as activation function, loss function, optimization function, learning rate, and sample size. Different values of these parameters are used in the experiments.

Table 3 Summary of common activation functions

Function	Mathematical	Derivative
Sigmoid	$f(x) = \sigma = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x)) = \frac{e^{-x}}{(e^{-x}+1)^2}$
Tanh	$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$
ELU	$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases}$	$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ f(x) + \alpha & \text{if } x \leq 0 \end{cases}$
ReLU	$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$	$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$
LeakyRelu	$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$	$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ \alpha & \text{if } x \leq 0 \end{cases}$
SoftMax	$f(x) = \sigma = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$	$f'(x) = \sigma(x_j)(1 - \sigma(x_i))$

The activation function helps in solving complex problems via performing non-linear transformation to the input [28]. Table 3 summarizes the most common activation functions with their derivatives.

In the proposed framework, activation functions (sigmoid, tanh, and ReLU) are examined. SoftMax is used for the output layer to enhance the classification process.

Batch normalization is tightly coupled with activation. It normalizes the previous activation to improve the performance and stability of CNN [18].

The fifth layer is the classification layer. Glorot initializer is used to initialize the network weights to achieve quicker convergence and higher accuracy. The training process starts to get the optimal CNN. Optimization function must be applied to minimize the loss (cost) function and obtain a robust model.

Loss function plays a major role in the training process in neural networks. It is used to measure the inconsistency between the predicted value and the actual value [20]. The target is to decrease the value of loss function. There are several loss functions as shown in Table 4. The cross-entropy loss function is used in the proposed framework since it achieves better performance rather than Mean Squad Error (MSE).

The decision boundary in a classification task is large (in comparison with regression). Using Cross-Entropy loss function after the softmax layer speeds the convergence of the neural network due to the gradient vanishing problem. In other words, it is more suitable for classification problems [16].

The optimization function aims at minimizing the loss function. It achieves this goal by changing the parameters (weights) in the model at the training phase. Most of the optimization functions are enhancements of Gradient Descent (GD) [9]. GD is a first-order optimization method. It only takes the first derivatives of the loss function into

Table 4 Summary of loss functions

Function	Formula
Mean square error/quadratic loss	$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$
L2 loss	$\mathcal{L} = \sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$
Mean absolute error	$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left\lvert y^{(i)} - \hat{y}^{(i)}\right\rvert$
L1 loss	$\mathcal{L} = \sum_{i=1}^{n}\left\lvert y^{(i)} - \hat{y}^{(i)}\right\rvert$
Mean bias error	$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})$
Hinge loss/multi class SVM loss	$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\max(0, m - y^{(i)} \cdot \hat{y}^{(i)})$
Squared hinge	$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left(\max(0, 1 - y^{(i)} \cdot \hat{y}^{(i)})\right)^2$
Cross entropy loss	$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log(\hat{y}^{(i)}) + (1 - y^{(i)})\log(1 - \hat{y}^{(i)})\right]$
Negative log likelihood	$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\log(\hat{y}^{(i)})$
Poisson loss function	$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}^{(i)} - y^{(i)} \cdot \log(\hat{y}^{(i)})\right)$
Cosine proximity loss function	$\mathcal{L} = -\frac{\mathbf{y} \cdot \hat{\mathbf{y}}}{\lvert\lvert\mathbf{y}\rvert\rvert_2 \cdot \lvert\lvert\hat{\mathbf{y}}\rvert\rvert_2} = -\frac{\sum_{i=1}^{n} y^{(i)} \cdot \hat{y}^{(i)}}{\sqrt{\sum_{i=1}^{n}\left(y^{(i)}\right)^2} \cdot \sqrt{\sum_{i=1}^{n}\left(\hat{y}^{(i)}\right)^2}}$

account. This causes slow convergence and sticking in the local minimum. GD is given by Eq. 1.

$$\theta = \theta - \eta \nabla J(\theta) \tag{1}$$

where θ is the weights vector, η is the learning rate, ∇ is the gradient, and J is cost function or loss function.

Another very popular technique that is used along with GD is called Momentum [27]. Instead of using only the gradient of the current step to guide the search, it also accumulates the gradient of the past steps to determine the direction to go. Momentum is given by Eq. 2.

$$\theta = \theta - \eta \nabla J(\theta) + \gamma v_t \tag{2}$$

where v_t is the gradient retained from previous iterations and γ is the "Coefficient of Momentum," the percentage of the gradient retained every iteration.

The optimization function used in the proposed framework is Adam. It is an optimization algorithm for stochastic gradient descent for training deep-learning models [22]. Additionally, it combines RMSprop and stochastic Gradient Descent with momentum. Also, it gets the advantage of momentum since it uses the moving average of the gradient instead of the gradient itself. Moreover, it adapts the learning rate for each weight of the neural network by estimating the first and the second moments of the gradient. So, using Adam optimizer fasten the convergence process.

It stores an exponentially decaying average of past squared gradients (v_t) along with an exponentially decaying average of past squared gradients (m_t).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla J(\theta)$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla J(\theta))^2$$

β_1 and β_2 are the exponential decay rates, m_t and v_t are estimates of the first momentum (the mean) and the second momentum (the uncentered variance) of the gradients respectively. The authors of Adam observed that m_t and v_t are biased toward zero. They computed bias-corrected estimates which are used in the Adam update rule Eq. 3:

$$\begin{aligned} \hat{m_t} &= \frac{m_t}{1 - {\beta_1}^t}, \qquad \hat{v_t} = \frac{v_t}{1 - {\beta_2}^t} \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v_t} + \epsilon}} \hat{m_t} \end{aligned} \tag{3}$$

where ϵ is a small term preventing division by zero.

3 Results

The experiments are performed on Google Colab which offers GPU backend, 25.51 GB high-speed RAM and 68.40 GB disk for free. The codes are written in Python programming language and Keras package is used for deep learning.

The kernel sizes are (3,3) and (2,2) for convolutional and max-pooling layers respectively. Dropout is optional and applied after the first and third convolutional layers with a fraction of 0.2 and after the 64-dense layer with a fraction of 0.2. ReLU and SoftMax are used for hidden and output layers respectively.

The classification in the first category of experiments is a binary classification. The effect of dataset size, batch size, and the dropout technique is investigated with different image sizes and data augmentation. The classification in the second category of experiments is multi-classification. The effect of sample size, learning rate and activation function is investigated with different numbers of instances.

3.1 Dataset

Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report[1].

3.2 Experiment category 1

The Experiments in this category measure several performance parameters. These parameters can be interpreted from the confusion matrix. Table 5 shows the confusion matrix for binary classification. Among these parameters, accuracy has the most attention. It is the fraction of predictions the model classified correct as in Eq. 4,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Recall is the fraction of actual positive predictions classified correctly, often referred to as sensitivity or true positive rate as in Eq. 5,

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

Precision is the fraction of positive predictions as in Eq. 6,

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

The receiver operating characteristic (ROC) curve is the curve resulted form plotting True Positive Rate (TPR) versus False Positive Rate (FPR) where,

[1] A complete listing of ADNI investigators can be found at: "http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf."

Table 5 Confusion matrix for binary classification

	Predicted positive (AD)	Predicted positive (NC)
Actual positive (AD)	True positive (TP)	False negative (FN)
Actual positive (NC)	False positive (FP)	True negative (TN)

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}$$

The area under ROC curve (AUC) is a principal parameter in classification performance estimation. Loss is the number indicating how bad the model classification was. The cross-entropy loss is recorded in the results, as stated in the previous section. Dice Similarity Coefficient (DSC) , often referred to as F1-score, combines both the precision and recall into a single parameter as in Eq. 7,

$$DSC = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \qquad (7)$$

Table 6 shows the effect of different dataset sizes and batch sizes with / without applying Dropout. The image size in this experiment is (128, 128) with no data augmentation. Dataset split sizes range from (0.1–0.5). Batch size values are (18, 32, 64, 128, 256). The accuracies, without Dropout, ranges from 99.953–100%. The average accuracy of all results without applying Dropout is 99.981%. The accuracies, with Dropout, have the same range, while the average is 99.987%.

Table 7 shows the effect of different dataset sizes and batch sizes with / without applying Dropout. The image size, in this case, is (64, 64) with no data augmentation. Dataset split sizes range from (0.1 to 0.5).

Batch size values are (18, 32, 64, 128, 256). The accuracies, without Dropout, range from 95.96 to 98.26%. The

Table 6 Performance parameters of the first experiment

Dataset split size	Batch size	With Dropout						Without Dropout					
		Accuracy	Precision	Recall	AUC	Loss	DSC	Accuracy	Precision	Recall	AUC	Loss	DSC
0.1	18	99.95	99.95	99.95	1.000	0.001	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	32	99.95	99.95	99.95	1.000	0.001	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	64	100.00	100.00	100.00	1.000	0.001	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	128	100.00	100.00	100.00	1.000	0.000	1.000	99.95	99.95	99.95	1.000	0.001	1.000
	256	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
0.2	18	100.00	100.00	100.00	1.000	0.000	1.000	100.00	99.99	99.99	1.000	0.000	1.000
	32	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	64	99.95	99.95	99.95	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	128	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	256	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
0.3	18	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	32	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	64	99.95	99.95	99.95	1.000	0.004	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	128	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	256	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
0.4	18	100.00	100.00	100.00	1.000	0.000	1.000	99.95	99.95	99.95	1.000	0.002	1.000
	32	100.00	100.00	100.00	1.000	0.000	1.000	99.95	99.95	99.95	1.000	0.002	1.000
	64	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	128	100.00	100.00	100.00	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	256	99.95	99.95	99.95	1.000	0.000	1.000	100.00	100.00	100.00	1.000	0.000	1.000
0.5	18	99.95	99.95	99.95	1.000	0.002	1.000	99.95	99.95	99.95	1.000	0.001	1.000
	32	99.95	99.95	99.95	1.000	0.003	1.000	99.95	99.95	99.95	1.000	0.002	1.000
	64	99.95	99.95	99.95	1.000	0.001	1.000	100.00	100.00	100.00	1.000	0.000	1.000
	128	99.95	99.95	99.95	1.000	0.001	1.000	99.95	99.95	99.95	1.000	0.001	1.000
	256	99.95	99.95	99.95	1.000	0.001	1.000	100.00	100.00	100.00	1.000	0.001	1.000

Table 7 Performance parameters of the second experiment

Dataset split size	Batch size	With dropout						Without dropout					
		Accuracy	Precision	Recall	AUC	Loss	DSC	Accuracy	Precision	Recall	AUC	Loss	DSC
0.1	18	97.61	97.61	97.61	0.995	0.105	0.976	97.54	97.54	97.54	0.997	0.119	0.975
	32	97.85	97.85	97.85	0.995	0.090	0.978	97.33	97.32	97.32	0.997	0.121	0.973
	64	98.26	98.26	98.26	0.996	0.056	0.983	97.83	97.82	97.82	0.998	0.095	0.978
	128	98.16	98.16	98.16	0.996	0.066	0.982	97.59	97.59	97.59	0.998	0.111	0.976
	256	98.18	98.17	98.17	0.996	0.053	0.982	96.91	96.91	96.91	0.997	0.131	0.969
0.2	18	97.71	97.71	97.71	0.996	0.131	0.977	96.48	96.47	96.47	0.997	0.162	0.965
	32	97.71	97.71	97.71	0.996	0.090	0.977	96.89	96.89	96.89	0.997	0.137	0.969
	64	97.83	97.83	97.83	0.996	0.083	0.978	97.35	97.35	97.35	0.997	0.112	0.974
	128	97.81	97.81	97.81	0.996	0.077	0.978	97.22	97.22	97.22	0.997	0.113	0.972
	256	97.94	97.94	97.94	0.996	0.079	0.979	97.64	97.63	97.63	0.997	0.092	0.976
0.3	18	96.98	96.98	96.98	0.995	0.116	0.970	96.11	96.11	96.11	0.997	0.193	0.961
	32	97.45	97.44	97.44	0.995	0.087	0.974	96.45	96.45	96.45	0.996	0.167	0.964
	64	97.41	97.41	97.41	0.995	0.094	0.974	96.89	96.89	96.89	0.996	0.145	0.969
	128	97.17	97.16	97.16	0.995	0.096	0.972	96.70	96.70	96.70	0.996	0.138	0.967
	256	97.05	97.05	97.05	0.995	0.099	0.970	96.27	96.26	96.26	0.996	0.160	0.963
0.4	18	97.08	97.08	97.08	0.995	0.106	0.971	95.90	95.90	95.90	0.996	0.179	0.959
	32	96.75	96.75	96.75	0.994	0.118	0.967	96.34	96.34	96.34	0.995	0.151	0.963
	64	97.08	97.08	97.08	0.994	0.104	0.971	96.57	96.57	96.57	0.995	0.136	0.966
	128	96.76	96.75	96.75	0.994	0.119	0.968	96.11	96.11	96.11	0.995	0.155	0.961
	256	96.96	96.96	96.96	0.994	0.104	0.970	96.30	96.29	96.29	0.995	0.158	0.963
0.5	18	96.09	96.08	96.08	0.994	0.157	0.961	94.68	94.68	94.68	0.995	0.231	0.947
	32	96.53	96.53	96.53	0.994	0.175	0.965	95.59	95.59	95.59	0.994	0.202	0.956
	64	96.42	96.42	96.42	0.993	0.125	0.964	95.52	95.52	95.52	0.994	0.225	0.955
	128	95.96	95.96	95.96	0.993	0.138	0.960	95.60	95.60	95.60	0.994	0.193	0.956
	256	96.32	96.32	96.32	0.993	0.136	0.963	95.47	95.47	95.47	0.994	0.193	0.955

average accuracy of all results without applying Dropout is 97.18%.

The accuracies, with Dropout, range from 94.68 to 97.83%. The average accuracy, in this case, is 96.42%. The proclaimed results show that images with size (128, 128) have better outcomes in all the experiments.

3.2.1 Data augmentation

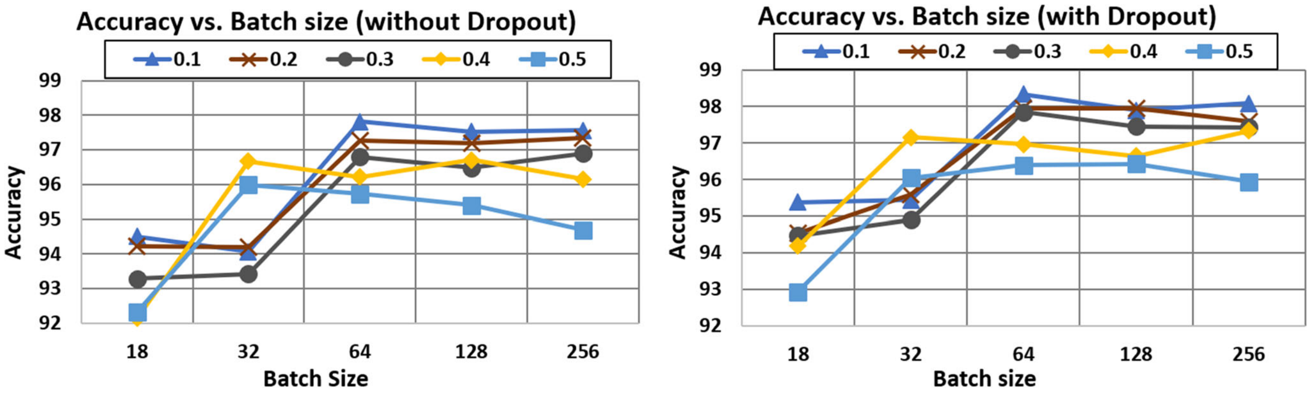
The experiments in this section are carried out to examine the data augmentation effect on the classification process. Four images are generated from a single image. The total number of images in the dataset, after augmentation, reaches 211,655. Data augmentation factors are horizontal flipping, shearing with a range of 0.2, shifting with a range of 0.1, rotating with 15 degrees, and zooming with a range of 0.2. Table 8 shows the effect of different dataset sizes and batch sizes with / without applying Dropout. The image size, in this case, is (64, 64). Dataset split sizes range from (0.1 to 0.5). batch size values are (18, 32, 64, 128, 256).

Figure 4a shows the relation between accuracy and batch size for different dataset split sizes without applying Dropout. The chart shows that the accuracy increases with the increase of batch size until the batch size is 64, then it starts to decrease once again. The greatest accuracy is recorded at batch size 64 and a split size of 0.1. The same observation can be noticed in Fig. 4b. It represents the same relation but with applying Dropout. The accuracy increases with the batch size until 64 then decreases. The accuracy is better in the case of applying Dropout.

Figure 5 shows the relation between accuracy and dataset split size in both cases (with and without Dropout). The figure shows that the accuracy decreases with the increase of Dataset split size. The best accuracy is obtained at split size 0.1. The accuracy achieved in the case of applying Dropout is better than achieved without Dropout.

Table 8 Accuracies of the third experiment

Dataset split size		Batch size									
		18		32		64		128		256	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Without dropout	0.1	98.131	94.501	98.485	94.066	99.658	97.817	99.642	97.524	99.655	97.562
	0.2	98.149	94.217	98.520	94.198	99.728	97.260	99.714	97.194	99.558	97.354
	0.3	98.036	93.291	98.588	93.423	99.675	96.797	99.653	96.475	99.558	96.891
	0.4	98.118	92.138	99.596	96.664	99.614	96.220	99.681	96.702	99.526	96.154
	0.5	98.144	92.318	99.584	95.994	99.549	95.720	99.607	95.398	99.701	94.690
With dropout	0.1	95.522	95.380	96.394	95.436	99.452	98.337	99.475	97.893	99.439	98.091
	0.2	95.316	94.538	96.069	95.578	99.405	97.940	99.494	97.950	99.162	97.581
	0.3	95.320	94.472	96.131	94.907	99.172	97.846	99.346	97.458	99.310	97.430
	0.4	95.451	94.189	99.152	97.156	99.337	96.967	99.458	96.646	99.292	97.335
	0.5	94.419	92.932	98.965	96.050	99.269	96.390	99.216	96.428	99.118	95.937



(a) without Dropout

(b) with Dropout

Fig. 4 The effect of different batch sizes on accuracy

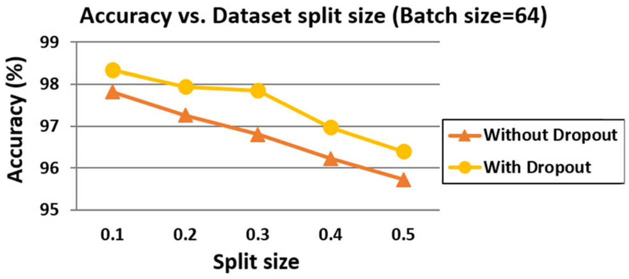


Fig. 5 Effect of different split sizes on accuracy

3.2.2 Ablation study

An ablation study is the scientific examination of a machine learning system by changing its hierarchy. In the proposed framework, the study is carried out by examining two other models. Table 9 shows the different models along with their layer hierarchy, shape, and filters.

The experiments are carried out on (64, 64) images without dropout with a batch size value of 256 and a split value of 0.5. The weight initializer used is the Glorot Uniform algorithm. Figure 6 shows the accuracies of the different models used in the Ablation study. The chart shows that the proposed model surpasses the other models.

3.2.3 The effect of Glorot algorithm

In the proposed framework, the Glorot uniform algorithm is the main weight initializer. Other initialization techniques are examined to encounter the performance of the Glorot algorithm. The techniques examined are zero initialization (Zeros), one initialization (Ones), and Random normal. The experiments are carried out on (64,64) images without dropout with a batch size value of 256 and a split

Table 9 Layer hierarchy of the tested models

Layer	Shape	Filters
Proposed model		
CONV	64 × 64	32 (3 × 3)
CONV	64 × 64	32 (3 × 3)
MAX_POOL		32 (2 × 2)
CONV	64 × 64	32 (3 × 3)
CONV	64 × 64	32 (3 × 3)
MAX_POOL		32 (2 × 2)
FLATTEN		
DENSE	128	
DENSE	64	
DENSE	2	
Testing model 1		
CONV	64 × 64	32 (3 × 3)
MAX_POOL		32 (2 × 2)
CONV	64 × 64	32 (3 × 3)
MAX_POOL		32 (2 × 2)
FLATTEN		
DENSE	128	
DENSE	2	
Testing model 2		
CONV	64 × 64	32 (3 × 3)
CONV	64 × 64	32 (3 × 3)
MAX_POOL		32 (2 × 2)
FLATTEN		
DENSE	128	
DENSE	64	
DENSE	2	

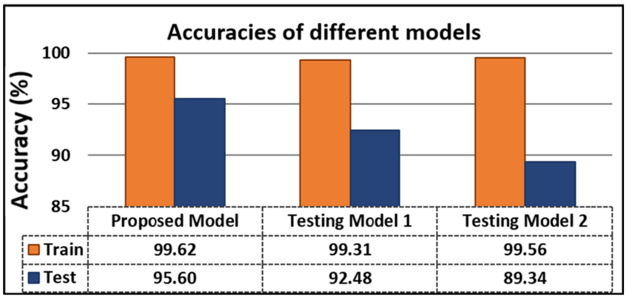


Fig. 6 Accuracies of the models of the Ablation study

value of 0.5. Figure 7 shows that applying Glorot initialization leads to higher accuracy.

3.3 Experiment category 2

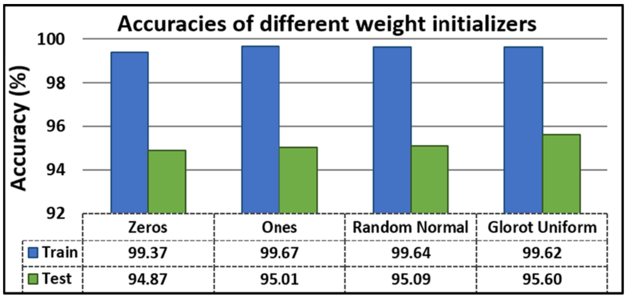


Fig. 7 Accuracies of the models with different weight initializers

Several experiments are performed in this category. These experiments aim to study parameters such as sample size, learning rate, and activation function. The first parameter to study is the sample size and its effect on accuracy. In this experiment, the other parameters are fixed. The activation function used is Sigmoid and the learning rate is fixed at a value of 0.1. Figure 8 shows that accuracy is inversely proportional to the sample size.

The second experiment studies the learning rate and its effect on accuracy. Different learning rates are applied in two different scenarios. The first scenario has a sample size of 200 instances and Sigmoid activation function. The second one has a sample size of 500 instances and Sigmoid activation function. Figure 9 shows that among the learning rate ranged from 0.05 to 0.15, the best accuracy gained at a learning rate of 0.1.

The third parameter to study is the activation function. The experiments are carried out with two different scenarios each with different sample sizes (500 and 1000 instances). The experiments examine three different functions (Sigmoid, Tanh, and ReLU). Figure 10 shows that Tanh function gives higher accuracy than the other in case of a large sample size. In a small sample size, Sigmoid function achieves better performance than the other two functions.

Table 10 shows the comparison between the proposed model and the state-of-the-art models. The proposed model outperforms other state-of-the-art models. The accuracy, precision, and recall enhanced respectively

The accuracy, precision, and recall enhanced respectively by 0.78%, 0.98%, and 0.98% above the highest values listed.

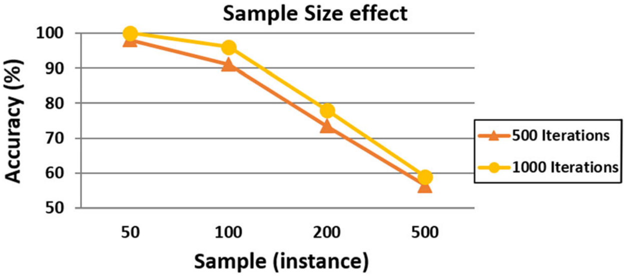


Fig. 8 Sample size effect

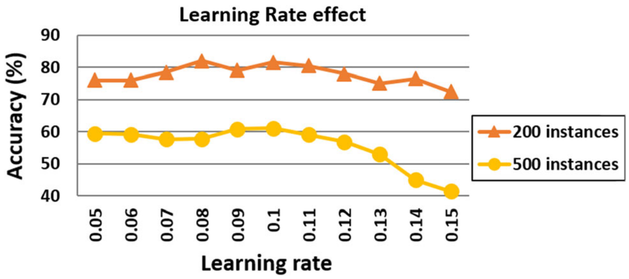


Fig. 9 Learning rate effect

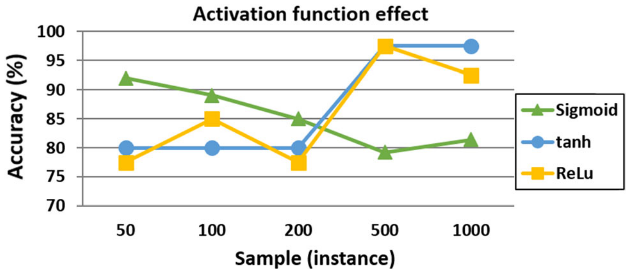


Fig. 10 Activation function effect

Table 10 Results comparison

Method	Accuracy %	Precision %	Recall %
Lee et al. [24]	98.74	96.32	96.32
Jain et al. [19]	99.14	99	99
Basaia et al. [5]	99.2	98.9	98.9
Wang et al. [40]	98.83	98.7	98.7
Choi et al. [12]	93.84	–	–
Basheera et al. [6]	97	97	97
The proposed framework	99.98	99.98	99.98

4 Discussion

This paper proposed an end-to-end framework for AD-classification based on CNN. The framework consists of five layers, the first layer is responsible for the MRI acquisition. In the second layer, the adaptive thresholding and data augmentation are used to enhance the training datasets. In the third layer, the cross-validation strategy is used to train the CNN. The cross-validation obtains the best values for the training parameters to avoid overfitting. In the fourth layer, the CNN model is applied. The CNN architecture consists of three convolutional layers and max-pooling is performed after each convolutional layer. The convolutional layers are followed by two fully connected layers. In the fifth layer, the classification process is done through many different algorithms.

The Glorot Uniform weight initializer is used to prevent neuron activation functions from starting in saturated or dead regions resulting in substantial quicker convergence and higher accuracy. Also, Adam optimizer in the optimization process is used to achieve quicker convergence. The effect of applying different values of sample size, activation function, and the learning rate is addressed though experiments. The experiment results showed that the classification accuracy of the proposed framework outperforms the state-of-art compared techniques for both binary and multi-classification.

As future work, more experiments should be conducted in the multi-classification category. Models of transfer learning should be investigated to improve modeling. Other directions represent a promising future direction, such as prediction of the incidence of AD.

Acknowledgements Data used in the preparation of this article were obtained from the Alzheimers disease Database Initiative (ADNI) database. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

Compliance with ethical standards

Conflicts of interest Yousry AbdulAzeem, Waleed Bahgat, and Mahmoud Badawy declare that they have no conflict of interest.

References

1. Aderghal K, Khvostikov A, Krylov A, Benois-Pineau J, Afdel K, Catheline G (2018) Classification of alzheimer disease on imaging modalities with deep CNNs using cross-modal transfer learning. In: 2018 IEEE 31st international symposium on computer-based medical systems (CBMS). IEEE. https://doi.org/10.1109/cbms.2018.00067
2. Altaf T, Anwar SM, Gul N, Majeed MN, Majid M (2018) Multi-class alzheimer's disease classification using image and clinical features. Biomed Signal Process Control 43:64–74
3. Asim Y, Raza B, Malik AK, Rathore S, Hussain L, Iftikhar MA (2018) A multi-modal, multi-atlas-based approach for alzheimer detection via machine learning. Int J Imaging Syst Technol 28(2):113–123
4. Baldacci F, Lista S, O'Bryant SE, Ceravolo R, Toschi N, Hampel H, Initiative APM et al (2018) Blood-based biomarker screening with agnostic biological definitions for an accurate diagnosis within the dimensional spectrum of neurodegenerative diseases. In: Biomarkers for alzheimer's disease drug development. Springer, pp 139–155
5. Basaia S, Agosta F, Wagner L, Canu E, Magnani G, Santangelo R, Filippi M, Initiative ADN 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
6. Basheera S (2020) A novel CNN based alzheimer's disease classification using hybrid enhanced ICA segmented gray matter of MRI. Comput Med Imaging Graph 18:101713
7. Beheshti I, Demirel H, Matsuda H, Initiative ADN et al (2017) Classification of alzheimer's disease and prediction of mild cognitive impairment-to-alzheimer's conversion from structural magnetic resource imaging using feature ranking and a genetic algorithm. Comput Biol Med 83:109–119

8. Beheshti I, Maikusa N, Matsuda H, Demirel H, Anbarjafari G (2017) Histogram-based feature extraction from individual gray matter similarity-matrix for alzheimer's disease classification. J Alzheimers Dis 55(4):1571–1582
9. Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of (COMPSTAT'2010). Springer, pp 177–186
10. Braak H, Braak E (1995) Staging of alzheimer's disease-related neurofibrillary changes. Neurobiol Aging 16(3):271–278
11. Bruscoli M, Lovestone S (2004) Is MCI really just early dementia? A systematic review of conversion studies. Int Psychogeriatr 16(2):129–140
12. Choi JY, Lee B (2020) Combining of multiple deep networks via ensemble generalization loss, based on MRI images, for alzheimer's disease classification. IEEE Signal Process Lett 27:206–210
13. Delacourte A, David JP, Sergeant N, Buee L, Wattez A, Vermersch P, Ghozali F, Fallet-Bianco C, Pasquier F, Lebert F et al (1999) The biochemical pathway of neurofibrillary degeneration in aging and alzheimer's disease. Neurology 52(6):1158–1158
14. Duraisamy B, Shanmugam JV, Annamalai J (2018) Alzheimer disease detection from structural MR images using FCM based weighted probabilistic neural network. Brain Imaging Behav 13:1–24
15. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256
16. Golik P, Doetsch P, Ney H (2013) Cross-entropy vs. squared error training: a theoretical and experimental comparison. In: Interspeech, vol 13, pp 1756–1760
17. Hampel H, Toschi N, Baldacci F, Zetterberg H, Blennow K, Kilimann I, Teipel SJ, Cavedo E, Melo dos Santos A, Epelbaum S et al (2018) Alzheimer's disease biomarker-guided diagnostic workflow using the added value of six combined cerebrospinal fluid candidates: Aβ1-42, total-tau, phosphorylated-tau, nfl, neurogranin, and ykl-40. Alzheimer's & Dementia 14(4):492–501
18. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167
19. Jain R, Jain N, Aggarwal A, Hemanth DJ (2019) Convolutional neural network based alzheimer's disease classification from magnetic resonance brain images. Cogn Syst Res 57:147–159
20. Janocha K, Czarnecki WM (2017) On loss functions for deep neural networks in classification. arXiv:1702.05659
21. 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
22. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. http://arxiv.org/abs/1412.6980arXiv:1412.6980
23. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
24. Lee B, Ellahi W, Choi JY (2019) Using deep CNN with data permutation scheme for classification of alzheimer's disease in structural magnetic resonance imaging (SMRI). IEICE Trans Inf Syst 102(7):1384–1395
25. Liu M, Cheng D, Wang K, Wang Y (2018) Multi-modality cascaded convolutional neural networks for alzheimer's disease diagnosis. Neuroinformatics 16(3–4):295–308. https://doi.org/10.1007/s12021-018-9370-4
26. Markesbery WR (2010) Neuropathologic alterations in mild cognitive impairment: a review. J Alzheimers Dis 19(1):221–228
27. Nakerst G, Brennan J, Haque M (2020) Gradient descent with momentum—to accelerate or to super-accelerate? arXiv:2001.06472
28. Nwankpa CE, Ijomah W, Gachagan A, Marshall S (2018) Activation functions: comparison of trends in practice and research for deep learning. arXiv:1811.03378
29. Payan A, Montana G (2015) Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. arXiv:1502.02506
30. Plis SM, Hjelm DR, Salakhutdinov R, Allen EA, Bockholt HJ, Long JD, Johnson HJ, Paulsen JS, Turner JA, Calhoun VD (2014) Deep learning for neuroimaging: a validation study. Front Neurosci 8:229
31. Prince M, Wimo A, Guerchet M, Ali G, Wu Y, Prina M et al (2019) Alzheimer's disease international: World alzheimer report 2015: the global impact of dementia: an analysis of prevalence, incidence, cost and trends 2015. Alzheimer's Disease International, London
32. Rathore S, Habes M, Iftikhar MA, Shacklett A, Davatzikos C (2017) A review on neuroimaging-based classification studies and associated feature extraction methods for alzheimer's disease and its prodromal stages. NeuroImage 155:530–548
33. Rizzi L, Rosset I, Roriz-Cruz M (2014) Global epidemiology of dementia: Alzheimer's and vascular types. BioMed research international
34. Samper-Gonzalez J, Burgos N, Bottani S, Fontanella S, Lu P, Marcoux A, Routier A, Guillon J, Bacci M, Wen J et al (2018) Reproducible evaluation of classification methods in alzheimer's disease: framework and application to MRI and pet data. NeuroImage 183:504–521
35. Sarraf S, Tofighi G (2016) Deep learning-based pipeline to recognize alzheimer's disease using FMRI data. In: Future technologies conference (FTC). IEEE, pp 816–820
36. Serrano-Pozo A, Frosch MP, Masliah E, Hyman BT (2011) Neuropathological alterations in alzheimer disease. Cold Spring Harbor Perspect Med 1(1):a006189
37. Sørensen L, Igel C, Pai A, Balas I, Anker C, Lillholm M, Nielsen M, Initiative ADN et al (2017) Differential diagnosis of mild cognitive impairment and alzheimer's disease using structural MRI cortical thickness, hippocampal shape, hippocampal texture, and volumetry. Neuro Image Clin 13:470–482
38. Teipel SJ, Cavedo E, Lista S, Habert MO, Potier MC, Grothe MJ, Epelbaum S, Sambati L, Gagliardi G, Toschi N et al (2018) Effect of alzheimer's disease risk and protective factors on cognitive trajectories in subjective memory complainers: an insight-pread study. Alzheimer's & Dementia 14(9):1126–1136
39. Vieira S, Pinaya WH, Mechelli A (2017) Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications. Neurosci Biobehav Rev 74:58–75
40. Wang S, Wang H, Cheung AC, Shen Y, Gan M (2020) Ensemble of 3d densely connected convolutional network for diagnosis of mild cognitive impairment and alzheimer's disease. In: Deep learning applications. Springer, pp 53–73
41. Wang SH, Phillips P, Sui Y, Liu B, Yang M, Cheng H (2018) Classification of alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling. J Med Syst 42(5):85
42. Wimo A, Guerchet M, Ali GC, Wu YT, Prina AM, Winblad B, Jönsson L, Liu Z, Prince M (2017) The worldwide costs of dementia 2015 and comparisons with 2010. Alzheimer's & Dementia 13(1):1–7
43. Zhang Y, Wang S, Sui Y, Yang M, Liu B, Cheng H, Sun J, Jia W, Phillips P, Gorriz JM (2018) Multivariate approach for alzheimer's disease detection using stationary wavelet entropy and predator-prey particle swarm optimization. J Alzheimers Dis 65(3):855–869