Arabian Journal for Science and Engineering (2025) 50:7781–7800
https://doi.org/10.1007/s13369-024-09954-y

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Early Detection of Alzheimer's Disease from MR Images Using Fine-Tuning Neighborhood Component Analysis and Convolutional Neural Networks

Öznur Özaltın[1]

Received: 25 June 2024 / Accepted: 29 December 2024 / Published online: 19 January 2025

Abstract
This study develops an automatic algorithm for detecting Alzheimer's disease (AD) using magnetic resonance imaging (MRI) through deep learning and feature selection techniques. It utilizes a dataset of 6400 MRI images from Kaggle, categorized into four classes. Initially, the study employs pretrained CNN architectures—DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet—for classification using five fold cross-validation, with DenseNet-201 achieving the highest accuracy of 82.11%. Due to the dataset's size and imbalance, as well as the long training times, the study aims to create a more efficient algorithm. The CNNs are used as deep feature extractors from AD images, and the extracted features are reduced using a new fine-tuning neighborhood component analysis (FTNCA) algorithm, which minimizes loss and determines the optimal tolerance value. The essential features are then classified using various machine learning algorithms, including artificial neural network (ANN), K-nearest neighbor (KNN), Naïve Bayes, and support vector machine (SVM). Experimental results reveal that reducing the feature set from 2048 to 344 allows the ResNet-50-FTNCA-KNN model to achieve 100% accuracy, significantly enhancing AD detection. This approach will aid in the early diagnosis and treatment of AD patients.

Keywords Convolutional neural networks · Classification · Deep feature extraction · Feature selection · Pretrained architectures · Neighborhood component analysis

1 Introduction

Alzheimer's disease (AD) is a neurodegenerative disorder characterized by significant cerebral pathology and neuronal apoptosis, leading to progressive brain atrophy and functional impairment resembling dementia [1–4]. According to the projected figures, it is anticipated that by the year 2050, the prevalence of AD will reach a ratio of 1 in 85 individuals within the population [1]. Alterations in the morphology of neurons serve as the fundamental underpinning for the pathophysiological mechanisms of AD. As the TAU protein, which is essential for preserving the internal integrity of cells, undergoes chemical alteration in patients with AD, it leads to the formation of tangles in neurofibrillary tissue [1]. The cellular units exhibit an initial onset of malfunction, followed by the eventual termination of their vital functions. The initial regions of the brain that might experience effects can be visualized on an MRI image, encompassing the hippocampal lobe and medial temporal lobe. The hippocampus is a neural structure situated in the cerebral cortex, specifically in the medial temporal lobe [1, 5]. It forms intricate connections with various brain regions implicated in cognitive processes such as cognition and executive functions. Moreover, the hippocampus plays a crucial role in the encoding, consolidation, and retrieval of memories. Furthermore, it affects the cerebral cortex, resulting in the depletion of neurons and synapses within the cortical, temporal, and parietal lobes. The diagnosis of AD is predominantly established through the assessment of clinical evidence across different stages. The magnitude of cognitive impairment is classified into preclinical, mild, and dementia-stage classifications. The main symptoms include periodic short-term memory loss and relatively sparse long-term memory.

Imaging procedures are executed with the purpose of excluding alternative etiologies of dementia, including syphilis, vitamin B12 deficiency, and cerebrovascular disease. Moreover, many procedures, such as EEG, blood

✉ Öznur Özaltın
oznurozaltin@hacettepe.edu.tr; doanoznur09@gmail.com

[1] Department of Actuarial Sciences, Faculty of Science, Hacettepe University, Beytepe, Ankara, Turkey

testing, and genotyping, are executed. The enlargement of the third ventricle and the presence of cerebral atrophy can be observed in brain imaging. However, despite its generic nature, it exhibits predictability in relation to AD. Based on volumetric MRI analysis, it has been observed that the medial temporal lobe is reduced in size. The brain regions in the medial temporal and parietal lobes are used to map dysfunction patterns via functional brain imaging techniques such as fMRI, positron emission tomography (PET), and single-photon emission computed tomography (SPECT) [1, 6]. Based on these findings, it is imperative for clinical studies to focus their attention on individuals in the preliminary stages prior to the manifestation of evident brain atrophy. However, detecting this condition in its early stage is challenging. Multiple scientific studies have demonstrated that the process of brain degeneration induced by AD initiates several years prior to the manifestation of any observable clinical symptoms. Implementing a computer-aided methodology is of utmost importance for achieving precise and timely detection of AD, as there is a projected continuous increase in the expenses associated with treating individuals afflicted with AD [1–4, 6].

This study investigates how artificial intelligence (AI)-based biomedical imaging categorization can detect and improve AD diagnosis at an early stage. For this purpose, convolutional neural networks (CNNs), which are frequently employed on MR images from AI-based algorithms, are utilized [7–11]. Figure 1 shows the MR images utilized in the present study, which encompassed various stages of AD and no dementia.

The purpose of a feature extraction strategy is to select the most crucial features or knowledge to perform a classification operation [12–18]. However, performing this process manually is both time-consuming and requires expert knowledge [8]. Hence, CNN algorithms fully automate the process of extracting features and performing classification [19, 20]. AD is the most prevalent etiology of dementia on a global scale, and the timely identification of AD is of utmost importance in optimizing the well-being of individuals affected by this condition [2]. Therefore, the development of an automatic system to detect AD is undertaken by utilizing the inherent capabilities of CNNs to generate deep features from MR images. In this study, pretrained architectures are utilized, because designing a new architecture is quite time-consuming [10]. In this study, pretrained architectures DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet are employed. These architectures are used both as classifiers and feature extractors. In this study, a new feature selection algorithm is proposed that reduces the obtained features with minimum loss when used as a feature extractor. Many classical feature selection methods exist in the literature [21–23], each with specific assumptions and limitations. The proposed algorithm automatically selects important features, minimizing errors and enhancing classification success. A new feature selection algorithm is named fine-tuning NCA (FTNCA). Why it is called fine-tuning can be explained as follows: Normally, when selecting features in the NCA algorithm, they are determined according to the tolerance value (weight threshold) determined by the user [8]. This part is the main contribution of the present study. The algorithm uses machine learning techniques as classifiers, including ANN, KNN, naïve Bayes, and SVM. The optimum tolerance value of the features is determined according to the minimum error of each classifier. As a result, the best tolerance value and number of features are automatically identified for each classifier. In addition, it is presented that this hybrid structure is successful in classifying AD. The experimental findings are measured with accuracy, sensitivity, specificity, precision, F1-score, Matthews correlation coefficient (MCC), and kappa coefficient. The results support this finding by showing superior performance compared to using the pretrained algorithm as a classifier, namely, ResNet-50-FTNCA-KNN, shows that the best algorithm is determined AD from MR images. This innovative approach not only enhances classification accuracy but also streamlines the feature selection process, making it more efficient. Consequently, it paves the way for further research into improving diagnostic tools for AD and similar cognitive disorders. A summary of this study's contributions and novelties is presented in the next subsection. Figure 2 shows a flowchart of this study.

1.1 Contribution and Novelty of the Present Study

The study presented in this paper offers notable contributions and novel findings, as follows:

- Utilizing pretrained CNN architectures such as DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet showcases significant advantages in extracting deep features from MR images without manual intervention.
- The proposed FTNCA algorithm autonomously selects essential features by determining the optimal tolerance value that minimizes loss, enhancing the efficiency of the feature selection process.
- Empirical results indicate that KNN, when combined with FTNCA, outperforms other classifiers, achieving remarkable accuracy rates.
- The study utilizes a substantial and imbalanced dataset from Kaggle, consisting of 896 mild dementia, 64 moderate dementia, 3200 nondementia, and 2240 very mild dementia instances, demonstrating the robustness of the proposed methodology.

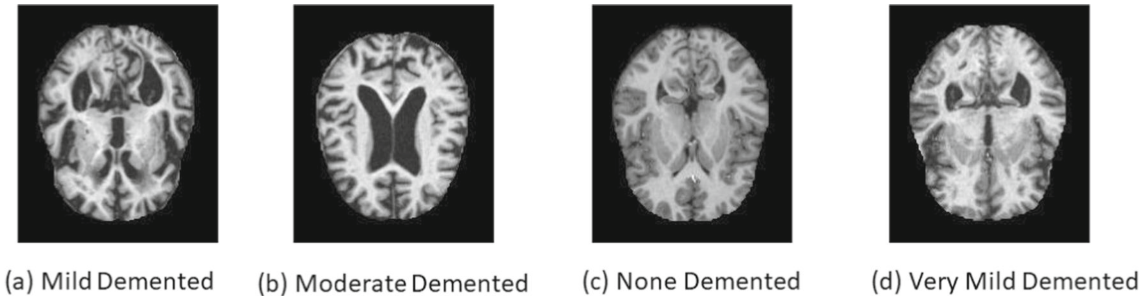


Fig. 1 MRI images of AD stages

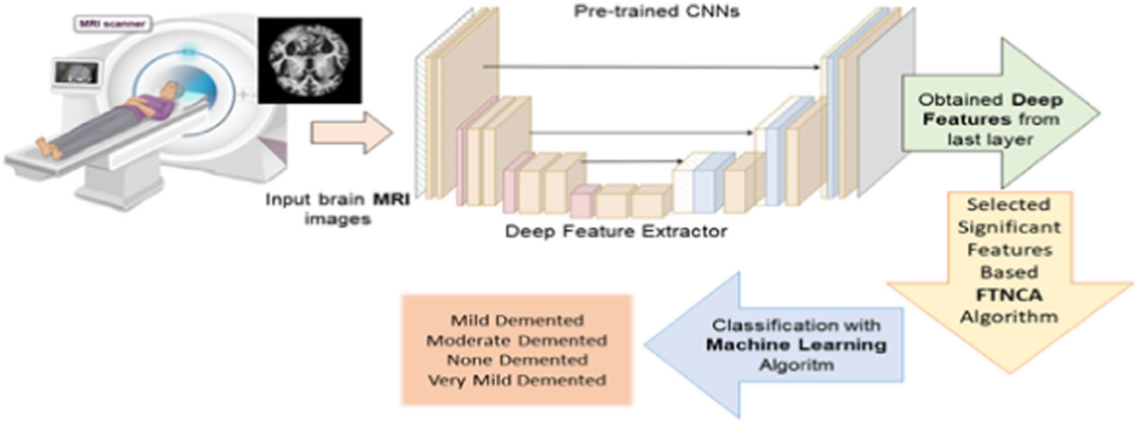

MRI scanner
Input brain MRI images
Pre-trained CNNs
Deep Feature Extractor
Obtained Deep Features from last layer
Selected Significant Features Based FTNCA Algorithm
Classification with Machine Learning Algoritm
Mild Demented
Moderate Demented
None Demented
Very Mild Demented

Fig. 2 Flowchart of the present study

- The classification performances of all architectures are improved using the proposed algorithm, with the ResNet-50-FTNCA-KNN model achieving a perfect accuracy rate of 100%.
- The integration of FTNCA with deep learning models not only enhances classification accuracy but also significantly reduces computational time and resources, making the approach practical for real-world applications. This method proves superior to traditional deep learning classifiers in both accuracy and time efficiency.

This study highlights the significant potential of integrating deep learning with advanced feature selection technique to enhance the accuracy and efficiency of AD detection from MR images. While traditional deep learning classifiers are effective, they often require considerable computational time and resources. The proposed approach, particularly the ResNet-50-FTNCA-KNN model, not only achieves superior accuracy but also significantly reduces computational requirements, providing a practical and efficient solution for early AD detection.

1.2 Literature Review

The study of AD detection through advanced imaging techniques has seen significant progress in recent years. Various deep learning and machine learning approaches have been employed to enhance the accuracy and efficiency of AD diagnosis. However, there are still gaps and limitations in the existing literature that this study aims to address. For instance, Sethi, Rani, Singh, and Mazón [24] aimed to detect AD from MR images using a hybrid deep learning algorithm. First, they employed CNNs utilizing publicly available datasets to classify AD stages. Then, to improve the classification accuracy, CNNs were used as feature extractors and merged with SVM. Finally, the hybrid approach obtained an accuracy of 86.2%. Similarly, Ismail, Alabdullatif, and Bchir [25] suggested an automatic system for identifying AD based on NCA and the KNN algorithm. First, they extracted features from MR images using voxel-based morphometry (VBM) analysis and then used NCA for feature selection. Finally, KNN was used in the classification phase, and an accuracy of 83% was obtained.

In a more recent work, Al Fahoum and Zyout [26] enhanced early detection of schizophrenia through multimodal EEG analysis, demonstrating the importance of combining wavelet transform, reconstructed phase space, and deep learning neural networks for improved diagnostic accuracy [26]. This indicates a growing interest in hybrid methods and feature selection in medical imaging, which this study builds upon by introducing the FTNCA method.

The FTNCA method was chosen over other feature selection methods due to its superior ability to handle high-dimensional data and enhance classification performance. Unlike principal component analysis (PCA), which focuses on variance maximization, or sequential feature selection (SFS), which sequentially selects features, FTNCA transforms the feature space to enhance discriminative power and selects features that maximize the distance between classes.

Previous studies, such as Jin and Deng [27], demonstrated that NCA outperformed PCA and SFS in terms of feature selection and classification accuracy. This motivated the integration of FTNCA in this study to leverage its strengths in handling complex datasets and improving AD detection accuracy.

Existing methods for AD detection from MR images have primarily relied on CNNs, SVM, and hybrid approaches. For instance, Kaplan et al. [2] proposed the ExHiF method combining HOG features, NCA, SVM, and KNN, achieving 100% accuracy on two datasets. While their results are impressive, the study did not explore the potential of more advanced feature selection methods like FTNCA. Similarly, Ozbay and Ozbay [28] applied a hybrid method based on CNN, NCA, and machine learning algorithms to detect AD from MR images, achieving an accuracy of 99.83%. They also used the gradient-weighted class activation mapping (Grad-CAM) method and obtained 3000 features from each image. Their study highlighted the effectiveness of combining deep learning with feature selection, yet the dimensionality reduction from 3000 to 400 features using NCA suggests room for improvement in feature extraction techniques. Kaplan et al. [29] suggested a handcrafted feature extraction method, local phase quantization (LPQ), from MR images to identify AD. They obtained 1536 features via the LPQ method and selected 256 important features using NCA for an image. In the classification phase, they utilized six different machine learning algorithms and tested their proposed method on different AD datasets. Finally, their experimental results showed that the proposed method achieved an accuracy greater than 99.63% in detecting AD without computational complexity. Asgharzadeh-Bonab et al. [30] presented a hybrid approach including preprocessing methods, CNNs for feature extraction, and NCA for feature selection, achieving an accuracy of 90.3%. This reinforces the importance of advanced feature selection in enhancing diagnostic accuracy, which this study aims to further optimize through FTNCA. AlSaeed and Omar [31] recommended ResNet-50 as a deep feature extractor and classifier for identifying AD in MR images. In their study, softmax, SVM, and random forest classifiers were compared in terms of classification performance. Finally, they obtained the best results, with a softmax accuracy of 99%.

To validate the effectiveness of the FTNCA method, a comparative analysis was conducted against existing state-of-the-art methods. Table 1 summarizes the performance of different approaches in AD detection.

This comparative analysis highlights the superior performance of the FTNCA method to existing techniques. The proposed approach demonstrates improved accuracy and efficiency in feature selection and AD detection, further validating its potential as a robust diagnostic tool.

The organization of the present study is as follows. Section 2 includes the AD dataset, information about CNNs, suggested algorithms, classifiers, hyper-parameter selection, and performance metrics. Then, the experimental results are presented in Sect. 3. Findings and evaluations are given in Sects. 4 and 5.

2 Methods

This section details the methodologies employed to conduct the study on AD detection using MR images. The methods involve dataset acquisition, CNN-based feature extraction, the implementation of a novel feature selection approach (FTNCA), and the validation of these methods through rigorous cross-validation techniques. Each subsection outlines specific parameters, algorithms, and steps undertaken to ensure the robustness and accuracy of the proposed approach.

2.1 Dataset

The dataset used in this study, obtained from Kaggle at https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images, is ideal for AD detection research due to its comprehensive coverage of different AD stages [42]. This dataset contains 6400 MR images, categorized into four classes: 896 mild dementia, 64 moderate dementia, 3200 nondementia, and 2240 very mild dementia MR images. The unbalanced categorization reflects real-world scenarios, where the prevalence of these conditions varies. Moreover, the dataset was approved for both labels and classes [33]. The dataset comprises 200 subjects, each associated with 32 horizontal slices of the brain. The training and testing sets were consolidated to prevent information leakage, followed by implementing fivefold cross-validation techniques. The original image dimensions were 176 × 208 pixels; however, all images were resized to 224 × 224 pixels for uniformity [40].

Table 1 Results of state-of-the-art technologies for detecting AD

Study	Dataset	Classes	Method	Result
Sethi et al. [24]	OASIS	AD MCI CN	CNN-SVM	86.2% Accuracy
Hazarika et al. [32]	ADNI	AD MCI CN	DenseNet-121 (improved)	88.17% Accuracy
Khalid et al. [33]	Kaggle	Mild dementia Moderate Nondementia Very mild dementia	DenseNet-121-handcrafted features-PCA-feedforward neural network (FFNN)	99.7% Accuracy
Lanjewar et al. [34]	Kaggle	Mild dementia Moderate Nondementia Very mild dementia	CNN-KNN	99.58% Accuracy
Shojaei et al. [35]	ADNI	AD CN	3D-CNN	96% Accuracy
Mohi ud din dar et al. [36]	ADNI	AD CN MCI EMCI LMCI	MobileNet	96.22% Accuracy
Agarwal et al. [37]	ADNI	AD CN MCI	3D-CNN	85.66% Accuracy
Venkatasubramanian et al. [38]	ADNI	AD MCI NC	CapsNet-DHO	93% Accuracy
Roopa et al. [39]	ADNI	AD CN MCI	Deep CNN	97.03% Accuracy
	Kaggle	Mild dementia Moderate Nondementia Very mild dementia		95.80% Accuracy
Chabib et al. [40]	Kaggle	Mild dementia Moderate Nondementia Very mild dementia	Fast discrete curvelet transform-CNN	98.62% Accuracy
Abbas et al. [41]	ADNI	AD HC	Jacobian map with CNN	96.61% Accuracy

ADNI: Alzheimer’s disease neuroimaging initiative; CapsNet: capsule network; CN: cognitive normal; CT: computed tomography; DHO: deer hunting optimization; EMCI: early mild cognitive impairment; HC: healthy control; LMCI: late mild cognitive impairment; MCI: mild cognitive impairment; NC: normal control

2.2 CNNs: Automatic Feature Extractor

CNN architectures are characterized by their distinctiveness from machine learning algorithms due to their utilization of multiple layers for the purpose of extracting deep features without the need for assistance by hand [11, 33].

Convolutional networks, which are biological in nature, draw inspiration from the neural connections within the visual cortex of felines. It is a notable methodology in the field of deep learning, wherein multiple layers are trained with specific and robust intentions [43]. The efficiency of the system is attributed to its high degree of accuracy and rapid execution. In computer vision, CNNs hold significant prominence as one of the most crucial methodologies [43].

The convolution layer, the pooling layer, and the fully connected layer are the three fundamental components that are shared by all versions of CNNs. The convolutional layers learn the input feature representations. The convolution layer kernels compute the feature maps. Each feature map neuron connects to an area of neighboring neurons in the preceding layer. The receptive field of the neurons in the prior layer is this neighborhood. Convolving the input with a learned kernel and implementing an elementwise nonlinear activation function yields the new feature map. Each feature map uses the kernel from all input spatial locations. Multiple kernels provide comprehensive feature maps. Mathematically, the feature value at point (i, j) in the kth feature map of the lth layer, $z^{l}_{i,j,k}$ is determined as follows: [44]

$$z^{l}_{i,j,k} = \mathbf{w}^{\mathbf{l}T}_{\mathbf{k}} \mathbf{x}^{l}_{i,j} + b^{l}_{k} \tag{1}$$

Here, $\mathbf{w}^{\mathbf{l}}_{\mathbf{k}}$ and b^{l}_{k} represent the weight vector and bias parameter of the kth filter of the lth layer, respectively, and $\mathbf{x}^{l}_{i,j}$ is expressed as the input patch centered at point (i, j) of the lth layer [44]. The kernel $\mathbf{w}^{\mathbf{l}}_{\mathbf{k}}$ that creates the feature map $z^{l}_{:,:,k}$ is shared. Weight sharing can simplify models and make networks easier to train. The CNN activation function adds nonlinearities that allow multilayer networks to identify nonlinear information. The nonlinear activation function is $a(\cdot)$. The activation value $a^{l}_{i,j,k}$ of convolutional feature $z^{l}_{i,j,k}$ is:

$$a^{l}_{i,j,k} = a\left(z^{l}_{i,j,k}\right) \tag{2}$$

The sigmoid, tanh [45], rectified linear unit (ReLU) [46], and leaky ReLU activation functions are typical examples [44]. By decreasing the resolution of the feature maps, the pooling layer works toward the goal of achieving shift invariance. Typically, it is positioned in the middle of two convolutional layers. Each feature map that is part of a pooling layer is connected to the feature map that was a part of the layer that came before it was convolutional. The pooling function is denoted as $pool(\cdot)$ for every feature map $a^{l}_{:,:,k}$ as follows: [44]

$$y^{l}_{i,j,k} = pool\left(a^{l}_{m,n,k}\right), \quad \forall (m, n) \in \Re_{ij} \tag{3}$$

Here, $\Re_{ij}$ is the local neighborhood around point (i, j). In general, the maximum pooling layer, global pooling layer, and average pooling layer are used in CNNs [44].

Each of these layers is responsible for a certain function. Additionally, CNNs offer two different training levels. Initially, the input image was fed into the CNNs by performing straightforward dot multiplication between the input and the neuron parameters. This was then followed by the execution of a convolutional multiplication on each layer of the network. The network error is calculated based on the output from the network training. To accomplish this goal, it compares the output of the network by employing a loss-error function and the proper answer, and it calculates the error rate. Then, the backpropagation phase begins based on the determined error rate, during which the chain rule is used to derive the derivative of each parameter [43]. This study effectively leveraged CNN's powerful automatic feature extraction capabilities.

2.2.1 Pretrained Architectures

Creating a new CNN architecture is not easy. Pretrained architectures provide time savings. Therefore, in this study, pretrained algorithms, such as DenseNet-201 [47], MobileNet-v2 [48, 49], ResNet-18 [50], ResNet-50 [50], ResNet-101 [50], and ShuffleNet [51], were used for the automatic detection of AD.

DenseNet-201, developed by Huang, Liu, Van Der Maaten, and Weinberger [47], consists of 201 deep layers. Every layer is connected to all the preceding layers in DenseNet-201. The feature mappings are merged and passed to the next layer. Classical CNNs consist of L layers and L connections. In the DenseNet-201 architecture with L layers, there are a total of L(L + 1)/2 connections. Therefore, it is a CNN architecture equipped with robust memory capabilities, enabling it to perform computations with exceptional efficiency [19].

MobileNet-v2 [48, 49] filters features with lightweight depth wise convolutions. Two key blocks contain the 32-filter fully convolutional layer. Nineteen residual bottleneck layers follow. Mobile devices are the main target.

ResNets (ResNet-18, ResNet-50, and ResNet-101) [50] include deep layers and were inspired by VGG-19 [52]. This design had several convolutional layers that functioned well. ResNet relies on shortcut jump relations. In this leaping process, the connection pressures the architecture, accelerating network learning. Due to their complicated layered nature, these architectures are expressed as a DAG network [7, 53].

ShuffleNet [51] helps mobile vehicles such as MobileNet-v2 compute with low energy. Pointwise group convolution and channel shuffling, two innovative methods, reduce computing costs while maintaining accuracy in this architecture [7].

This study employed various architectures for automatic feature extraction from AD MR images, with each architecture having a specific number of features in its final layer: DenseNet-201 has 1920, MobileNet-v2 has 1280, ResNet-18 has 512, ResNet-50 has 2048, ResNet-101 has 2048, and ShuffleNet has 544 features. These features undergo automatic reduction through fine-tuning NCA (FTNCA) to enhance AD detection. The architectures functioned as classifiers to assess the proposed method's effectiveness in detecting AD. The study outlines the used hyperparameters [10] as follows in Table 2. Furthermore, Algorithm 1 presents classification tasked of pretrained architectures.

Table 2 CNN training parameters for optimal performance

Parameter	Value/description
Optimization method	Adam optimizer was selected for its efficient computation and low memory requirements
Maximum number of epochs	Training was conducted for a maximum of five epochs to balance training time and performance
Batch size	A minimum batch size of 8 was chosen to ensure efficient training and avoid overfitting
Initial learning rate	The initial learning rate was set to 0.00001 to facilitate gradual convergence
Activation functions	Various activation functions, including ReLU, were employed to introduce nonlinearities necessary for complex pattern recognition

Algorithm 1 *The steps for using pretrained architectures as classification algorithms for AD detection are as follows:*

```
Step 1: Load and preprocess dataset
  1. Load MR images dataset from the specified directory
  2. Split the dataset into training and testing subsets using k-fold cross-validation
Step 2: Apply data augmentation
  1. Define augmentation parameters:
     Random rotation = [− 5 5],
     Random X reflection=1,
     Random X shear =[− 0.05 0.05],
     Random Y shear=[− 0.05 0.05]
  2. Apply augmentation to training images and resize all images to 224 × 224 pixels
Step 3: Define CNN Architectures and Training Parameters
  1. Initialize "ResNet-50" architecture with pretrained weights.
     (The same steps are applied to the DenseNet-201, MobileNet-v2, ResNet-18, ResNet-101, and ShuffleNet architectures.)
  2. Replace the final fully connected and classification layers with new layers for the current problem.
  3. Set training options: optimizer (Adam), epochs (5), mini-batch size (8), initial learning rate (0.00001).
Step 4: Train and Validate the Model.
  1. For each fold in k-fold cross-validation:
     a. Split data into training and testing sets for the current fold
     b. Train the model with the augmented training data
     c. Validate the model with the testing data
     d. Save the trained model and results for the current fold
Step 5: Evaluate model performance
  1. Calculate and display confusion matrix and accuracy for the predictions
  2. Calculate other performance metrics: sensitivity, specificity, precision, F1-score, MCC, and kappa
Step 6: Save results
  1. Save the final trained model for future
```

2.3 Feature Selection with Fine-Tuning Neighborhood Component Analysis (FTNCA)

High-dimensional data require feature selection and dimension reduction and may lose essential properties at that time [8, 27, 54]. Currently, researchers prefer NCA because of its lack of assumptions, lack of loss of essential features, and practical use [7, 8, 55–58].

According to Goldberger, Hinton, Roweis, and Salakhutdinov [59], NCA is a technique that makes use of a Mahalanobis distance metric in the supervised KNN algorithm with leave-one-out (LOO). This approach instantaneously maximizes the stochastic variation in the LOO KNN score on the training set. NCA trains a feature weighting vector by attempting to maximize the projected LOO classification accuracy using a regularization term [54, 59, 60].

The NCA approach is an effective discriminator that may identify the most important characteristics. In the present study, this nonparametric feature selection method was used to improve the classification performance of a specific type of AD stage. However, user support is needed to determine the tolerance value in feature selection [8]. This problem is mentioned in the study of Ozaltin, Coskun, Yeniay, and Subasi [8]. For this reason, the present study improved the FTNCA based on the automatic determination of the best tolerance value and the best number of features. In fact, an automatic feature selection system was developed for NCA (called iterative NCA-INCA) by Tuncer, Dogan, Özyurt, Belhaouari, and Bensmail [61]. However, there are algorithm differences between INCA and FTNCA.

2.3.1 Differences Between FTNCA and Similar Methods

INCA [61] is an automatic feature selection system for NCA. INCA selects one feature vector with minimal loss for a dataset. This approach, while effective, does not account for the variability in optimal feature sets across different architectures and classifiers. In contrast, FTNCA dynamically determines the best tolerance value for each architecture and classifier via an iterative process. This method allows for the selection of different feature sets tailored to the specific needs of the dataset, thereby enhancing classification accuracy.

Methodological Differences:

- Tolerance value adjustment: FTNCA iteratively adjusts the tolerance value and selects features that maximize classification accuracy, whereas INCA selects a single feature vector based on a predefined criterion.
- Architecture and classifier specificity: FTNCA's approach of dynamically adjusting the tolerance value for each architecture and classifier allows for more tailored and accurate feature selection, unlike INCA's more generalized approach.

FTNCA and INCA [61] are two algorithms designed for the same task. However, algorithms have methodological differences. In this suggested algorithm, the same number of features is not selected owing to the determination of the best tolerance value for each architecture and classifier via minimum loss for a dataset. However, in INCA, one feature vector is identified with the minimum loss for a dataset [62–69]. Algorithm 2 presents steps of hybrid algorithm using FTNCA.

Algorithm 2 *The steps for using pretrained architectures for feature extraction, FTNCA feature selection, and machine learning classifiers in AD detection are as follows:*

```
Step 1: Load and preprocess dataset

1. Load the pretrained ResNet-50 model and its layers. (The same steps are applied to the DenseNet-201, MobileNet-v2, ResNet-18, ResNet-101, and ShuffleNet architectures.)
2. Load AD MR images dataset and split into training (70%) and testing (30%) sets.

Step 2: Apply data augmentation

1. Define image augmentation parameters:
   Random rotation = [− 5 5],
   Random X reflection = 1,
   Random X shear = [− 0.05 0.05],
   Random Y shear = [− 0.05 0.05].
2. Apply augmentation to training images and resize all images to 224 × 224 pixels.

Step 3: Feature extraction from CNN

1. Extract features from a specified layer (e.g., "avg_pool") of the pretrained ResNet-50 model for both training and testing sets
2. Transpose the training features for further processing

Step 4: Implement NCA model

1. Create NCA model with parameters: lambda = 0.005, solver = SGD, gradient tolerance = 0.0001
2. Fit the NCA model to the training features and labels

Step 5: Feature selection using FTNCA

1. Define tolerance range (n = 0:0.01:0.9) and initialize loss array (L)
2. For each tolerance value:
   a. Select features with weights above the tolerance threshold
   b. Train classifier (e.g., KNN) with the selected features
   c. Calculate and store loss value
3. Identify the best tolerance value with minimum loss
4. Select the best features based on optimal tolerance

Step 6: Classification and evaluation

1. Train the final classifier (e.g., KNN) with the selected features
2. Predict labels for the test set and calculate accuracy

Step 7: Performance study

1. Perform multiclass confusion matrix analysis
2. Calculate and display additional performance metrics: sensitivity, specificity, precision, F1-score, MCC, kappa

Step 8: Save results

1. Save the final trained model for future use
```

2.4 Classification

The selected features based on the FTNCA were classified with ANN (using 50 neurons) [70], KNN (using a Euclidean distance and K value of 5) [71], naïve Bayes (using a multinomial distribution) [72], and SVM (with a Gaussian kernel function) [73].

2.5 Cross-Validation

In the present study, five fold cross-validation was applied to obtain reliable and credible results for classification purposes. Furthermore, the utilization of this approach effectively addresses the problem of overfitting during data analysis [7, 74]

2.6 Performance Evaluation

This study evaluated classifiers based on the following performance metrics: accuracy, sensitivity, specificity, precision, F1-score, MCC, and kappa coefficient [7, 11, 62, 75]. Furthermore, the following equations in Eqs. (4–11) were presented:

$$\text{Accuracy} = p_A = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \tag{4}$$

$$\text{F1 - score} = \frac{(2 \times \text{TP})}{(2 \times \text{TP} + \text{FP} + \text{FN})} \tag{5}$$

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{6}$$

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{8}$$

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} + \text{FN})}{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})} \tag{9}$$

$$p_E = \frac{(\text{TP} + \text{TN})(\text{TP} + \text{FP}) + (\text{FP} + \text{TN})(\text{TP} + \text{FN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})^2} \tag{10}$$

$$\text{kappa} = \max\left(\frac{p_A - p_E}{1 - p_E}; \frac{p_E - p_A}{1 - p_A}\right) \tag{11}$$

Here, TP: true positive, FP: false positive, TN: true negative, and FN: false negative were displayed. These metrics provide a comprehensive assessment of the classifiers' performance in detecting AD stages. Accuracy represents the proportion of correctly classified instances. The F1-score is a harmonic mean of precision and recall. Sensitivity (or recall) measures the true positive rate, while specificity measures the true negative rate. Precision is the ratio of true positive predictions to the total positive predictions. The MCC considers true and false positives and negatives. Additionally, the kappa coefficient measures the agreement between predicted and true labels, accounting for chance. These detailed metrics and their corresponding equations [Eqs. (4–11)] ensure a thorough evaluation of the proposed method, enabling a robust comparison with standard benchmarks and previous studies.

2.7 Experimental Setup

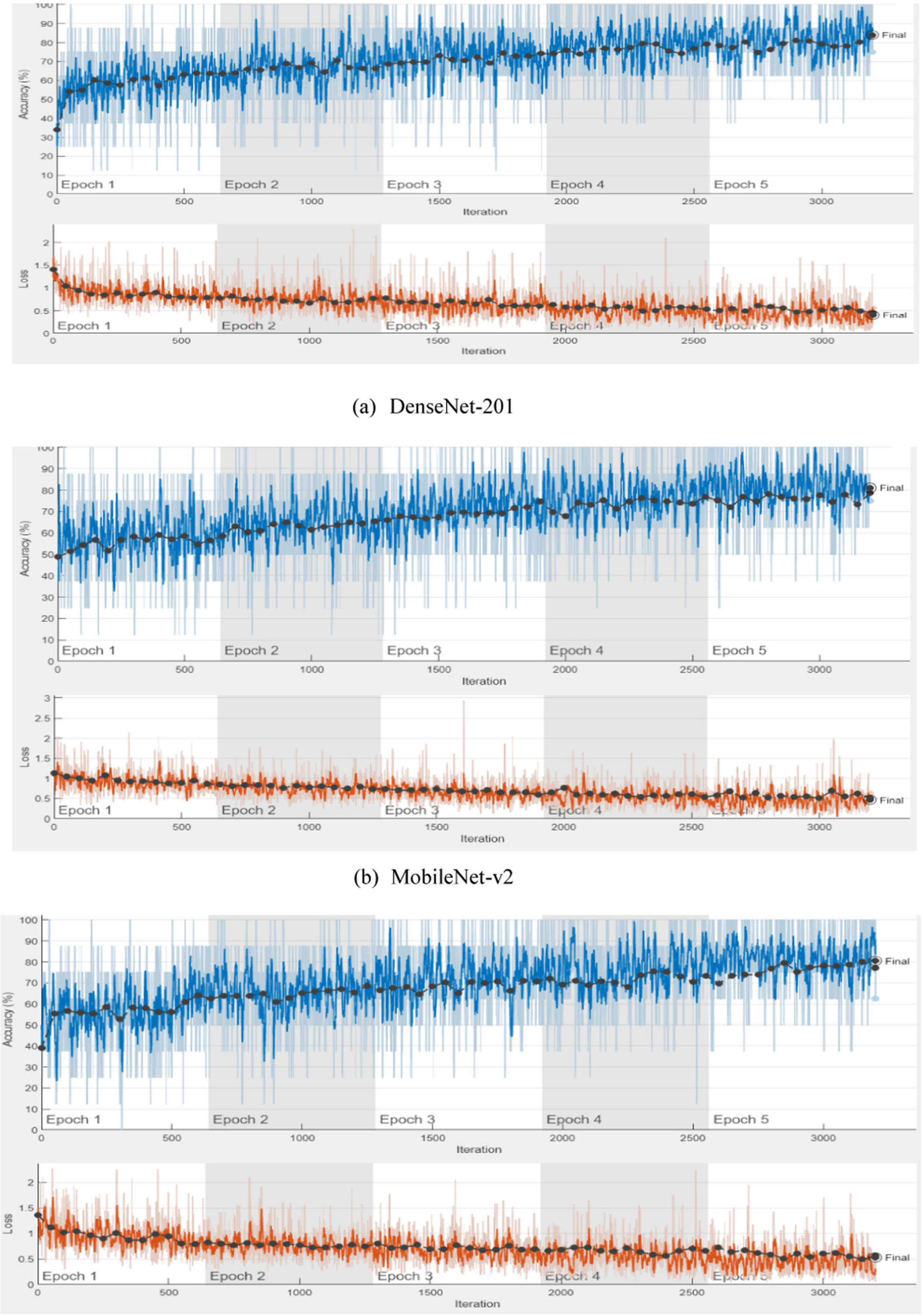
The experiments were conducted in a controlled hardware and software environment to ensure the reproducibility and reliability of the results. The hardware setup included an Intel Core i5-12450H processor, an NVIDIA GeForce RTX 1650 GPU with 8.00 GB RAM, providing the necessary computational power and storage for handling extensive data and complex neural network models. The software environment comprised Windows 11 as the operating system and MATLAB R2022b environment. CUDA version 11.2 was used to leverage GPU acceleration, expediting the training and inference processes.

The experimental procedure began with loading and preprocessing the MR images dataset using MATLAB functions, followed by applying data augmentation techniques including random rotations, reflections, and shears to increase the diversity of the training set. Pretrained CNN architectures (DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet) were fine-tuned to adapt to the AD detection task. The fine-tuning involved replacing the final fully connected and classification layers to suit the specific number of classes in the dataset. The Adam optimizer, chosen for its adaptive learning rate capabilities, was used with a mini-batch size of 8, and training was conducted over five epochs with an initial learning rate of 0.00001 to ensure effective learning without overfitting.

Features were extracted from the last layer of the CNN architectures, providing a rich set of data for further analysis. The FTNCA algorithm was then applied to select the most relevant features by iteratively adjusting the tolerance value and using NCA to maximize classification accuracy. This approach ensured that the most discriminative features were identified and used for training classifiers, including ANN, KNN, Naive Bayes, and SVM. The performance of each classifier was evaluated using metrics such as accuracy, sensitivity, specificity, precision, F1-score, MCC, and kappa coefficient, providing a comprehensive assessment of the model's effectiveness.

3 Experimental Results

The study employed DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet to classify AD MR images using a fivefold cross-validation method. Figure 3 illustrates the training process, indicating no overfitting. Performance metrics are detailed in Table 3. While the dataset is global, it effectively represents a real-world scenario; however, the unbalanced class distribution may have hindered the performance of pretrained architectures. Consequently, the study focused on hybrid methods to reduce

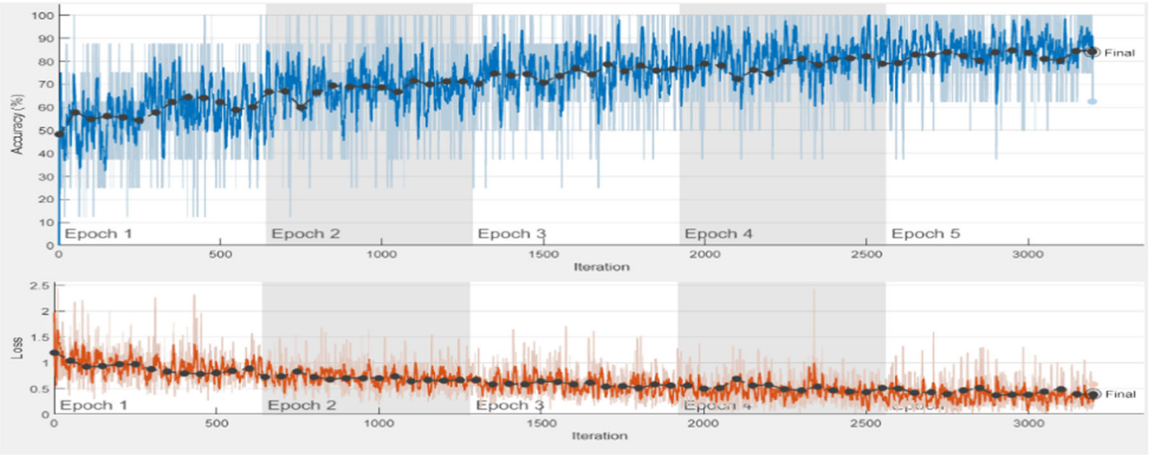


(a) DenseNet-201

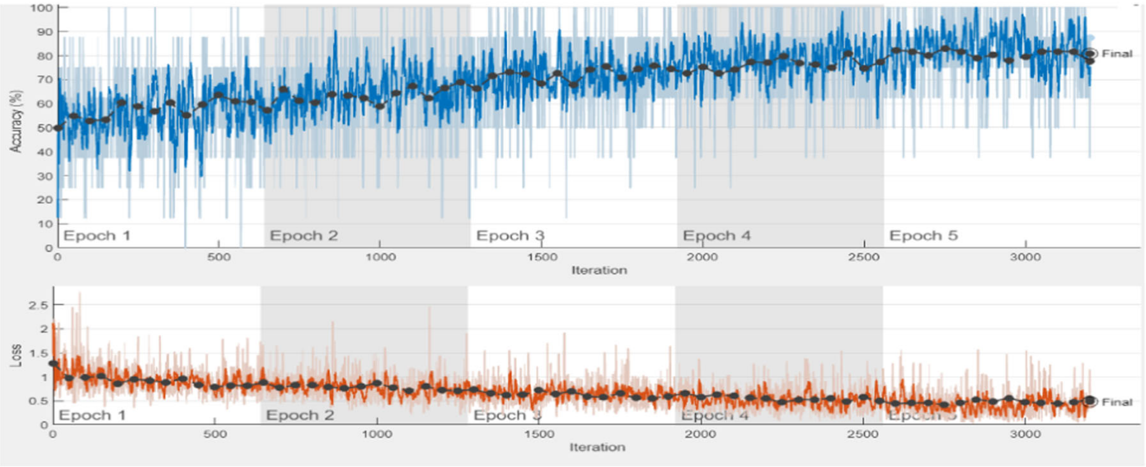
(b) MobileNet-v2

(c) ResNet-18

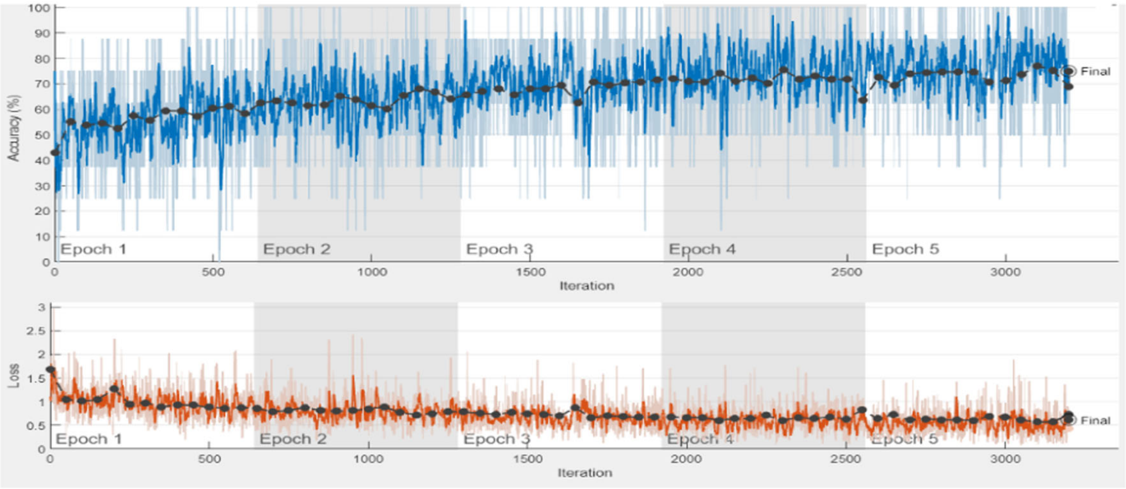
Fig. 3 The training plots for each pretrained architecture



(d) ResNet-50



(e) ResNet-101



(f) ShuffleNet

Fig. 3 continued

Table 3 Pretrained architecture classification results using five fold cross-validation

CNN	Accuracy	Sensitivity	Specificity	Precision	F1_score	MCC	Kappa	Average training time (s)
DenseNet-201	0.8211	0.8367	0.9229	0.8584	0.8440	0.7731	0.5229	30,797
MobileNet-v2	0.7947	0.7997	0.9138	0.8064	0.8021	0.7191	0.4525	2460
ResNet-18	0.8102	0.8166	0.9213	0.8395	0.8265	0.7491	0.4937	1490
ResNet-50	0.8116	0.7979	0.9176	0.8639	0.8262	0.7506	0.4975	3268
ResNet-101	0.7992	0.7411	0.9098	0.8660	0.7907	0.7139	0.4646	7436
ShuffleNet	0.7792	0.7206	0.9058	0.8197	0.7597	0.6750	0.4112	3428

Table 4 Results for the DenseNet-201-FTNCA hybrid algorithms

ML Algorithm	Tol_value	Features	Accuracy	Sensitivity	Specificity	Precision	F1_score	MCC	Kappa	Time (s)
ANN	0.02	334	0.9589	0.9542	0.9831	0.9539	0.9540	0.9371	0.8903	533
KNN	0.03	294	0.9969	0.9963	0.9987	0.9979	0.9971	0.9959	0.9917	443
Naïve Bayes	0.2	106	0.9073	0.8345	0.9627	0.9184	0.8659	0.8340	0.7528	514
SVM	0.02	334	0.9589	0.9685	0.9831	0.9550	0.9616	0.9446	0.8903	491

Table 5 Results for the MobileNet-v2-FTNCA hybrid algorithms

ML Algorithm	Tol_value	Features	Accuracy	Sensitivity	Specificity	Precision	F1_score	MCC	Kappa	Time (s)
ANN	0.05	164	0.8984	0.9190	0.9587	0.9152	0.9171	0.8753	0.7292	234
KNN	0.03	315	0.9807	0.9847	0.9922	0.9820	0.9833	0.9755	0.9486	179
Naïve Bayes	0.1	93	0.8495	0.7677	0.9396	0.8487	0.7950	0.7406	0.5986	139
SVM	0.04	185	0.9130	0.9276	0.9638	0.9086	0.9175	0.8817	0.7681	276

Table 6 Results for ResNet-18-FTNCA hybrid algorithms

ML Algorithm	Tol_value	Features	Accuracy	Sensitivity	Specificity	Precision	F1_score	MCC	Kappa	Time (s)
ANN	0.02	335	0.9333	0.9492	0.9713	0.9375	0.9430	0.9152	0.8222	137
KNN	0.08	180	0.9844	0.9852	0.9931	0.9896	0.9873	0.9809	0.9583	51
Naïve Bayes	0.43	23	0.8771	0.8965	0.9493	0.8954	0.8958	0.8458	0.6722	73
SVM	0.02	335	0.9323	0.9289	0.9712	0.9478	0.9381	0.9102	0.8194	63

Table 7 Results for ResNet-50-FTNCA hybrid algorithms

ML Algorithm	Tol_value	Features	Accuracy	Sensitivity	Specificity	Precision	F1_score	MCC	Kappa	Time (s)
ANN	0.04	1045	0.9776	0.9826	0.9916	0.9663	0.9743	0.9654	0.9403	248
KNN	0.13	344	1.00	1.00	1.00	1.00	1.00	1.00	1.00	339
Naïve Bayes	0.24	108	0.9016	0.8897	0.9616	0.8939	0.8914	0.8529	0.7375	364
SVM	0.03	1229	0.9823	0.9861	0.9930	0.9856	0.9858	0.9785	0.9528	347

Table 8 Results for ResNet-101-FTNCA hybrid algorithms

ML Algorithm	Tol_value	Features	Accuracy	Sensitivity	Specificity	Precision	F1_score	MCC	Kappa	Time (s)
ANN	0.04	980	0.9599	0.9341	0.9830	0.9693	0.9500	0.9338	0.8931	149
KNN	0.05	842	0.9969	0.9975	0.9988	0.9858	0.9915	0.9903	0.9917	123
Naïve Bayes	0.22	178	0.9073	0.8382	0.9613	0.9057	0.8650	0.8300	0.7528	126
SVM	0.02	1324	0.9682	0.9775	0.9864	0.9766	0.9771	0.9635	0.9153	146

Table 9 Results for ShuffleNet-FTNCA hybrid algorithms

ML Algorithm	Tol_value	Features	Accuracy	Sensitivity	Specificity	Precision	F1_score	MCC	Kappa	Time (s)
ANN	0.05	273	0.8146	0.7622	0.9238	0.8248	0.7898	0.7168	0.5056	108
KNN	0.02	404	0.9844	0.9887	0.9935	0.9879	0.9883	0.9817	0.9583	98
Naïve Bayes	0.2	169	0.7047	0.7278	0.8853	0.6789	0.6974	0.5848	0.2125	59
SVM	0.02	404	0.8385	0.7171	0.9342	0.7561	0.7337	0.6702	0.5694	91

computational complexity and temporal issues. The automatic feature extraction capability of pretrained architectures facilitated the easy acquisition of features from AD MR images without expert input, though not all features are equally important. Thus, a feature selection method called FTNCA was implemented to identify the most significant features. For classification, ANN, KNN, naïve Bayes, and SVM were utilized, and their performance in detecting AD was compared, as shown in the comparison graph. All experimental results are presented in Tables 3, 4, 5, 6, 7, 8 and 9.

Table 3 presents the performance evaluation of various pretrained CNN architectures for AD detection, focusing on several key performance metrics. DenseNet-201 achieves the highest overall accuracy (82.11%), sensitivity (83.67%), specificity (92.29%), precision (85.84%), F1-score (84.40%), and MCC (77.31%). Despite its superior performance, it has the longest average training time of 30,797 s. MobileNet-v2 balances performance and efficiency, with an accuracy of 79.47%, sensitivity of 79.97%, and a notably shorter training time of 2460 s, making it a practical choice for time-sensitive applications. ResNet-18 and ResNet-50 offer strong performances, with accuracies of 81.02% and 81.16%, respectively. ResNet-18 has a shorter training time (1490 s) compared to ResNet-50 (3268 s), while ResNet-50 provides slightly better precision (86.39%). ResNet-101 shows moderate performance with an accuracy of 79.92%, lower sensitivity (74.11%), but high precision (86.60%), and a relatively long training time of 7,436 s. ShuffleNet has the lowest performance metrics, with an accuracy of 77.92% and sensitivity of 72.06%, but maintains reasonable specificity (90.58%) and precision (81.97%). Its training time (3428 s) is moderate, but its overall lower performance may limit its applicability.

Briefly, DenseNet-201 is the most accurate and reliable model but requires significant computational resources and time. For applications where training efficiency is critical, MobileNet-v2 and ResNet-18 are preferable due to their shorter training times and competitive performance. Each model's strengths and weaknesses should be carefully considered in the context of specific application requirements and available resources.

In Sect. 2.4, the process of utilizing pretrained architectures as feature extractors to obtain a varying number of features is described. At this stage, the dataset is divided into two parts: 70% for training, consisting of 4480 images, and 30% for testing, consisting of 1920 images. Subsequently, the architectures are activated, and features from their final layers are extracted. A classical NCA model is created to calculate feature weights. Following the calculation of these feature weights, the FTNCA algorithm is applied to determine the optimal tolerance value that minimizes loss. Consequently, the best number of features for each architecture and classifier is identified. The primary objective is to determine the optimal number of features to achieve the most accurate detection of AD.

Table 4 presents the results of utilizing DenseNet-201 as a feature extractor, combined with FTNCA and machine learning algorithms, resulting in the extraction of 1,920 features per image from DenseNet-201's global average pooling layer.

Table 4 illustrates the performance of hybrid algorithms when combined with DenseNet-201. DenseNet-201-FTNCA-KNN stands out with an exceptional accuracy of 99.69%, sensitivity of 99.63%, and specificity of 99.87%, using 294 features with a tolerance value of 0.03. Its training time is relatively efficient at 443 s. DenseNet-201-FTNCA-ANN achieves a high accuracy of 95.89%, sensitivity of

95.42%, and specificity of 98.31%, with 334 features and a tolerance value of 0.02, and has a training time of 533 s. DenseNet-201-FTNCA-SVM mirrors the performance of ANN with an accuracy of 95.89%, but a higher sensitivity of 96.85% and similar specificity of 98.31%, with a training time of 491 s using the same number of features and tolerance value as ANN. DenseNet-201-FTNCA-Naïve Bayes, although demonstrating lower accuracy (90.73%) and sensitivity (83.45%) compared to others, maintains high specificity (96.27%) using 106 features with a tolerance value of 0.2. It completes training in 514 s.

Overall, KNN is the most accurate and efficient model, while ANN and SVM also perform robustly with longer training times. Naïve Bayes, despite its moderate performance, provides a viable option for less complex applications. The integration of DenseNet-201 and FTNCA enhances classification performance significantly compared to the baseline DenseNet-201 accuracy of 82.11%.

Table 5 presents the performance of various machine learning algorithms combined with MobileNet-v2-FTNCA for AD detection. MobileNet-v2-FTNCA-KNN classifier achieved the highest performance, with an accuracy of 98.07%, sensitivity of 98.47%, and specificity of 99.22%, using 315 features and a tolerance value of 0.03. The training time for MobileNet-v2-FTNCA-KNN was 179 s.

Other notable performances include the MobileNet-v2-FTNCA-ANN with an accuracy of 89.84% and training time of 234 s, and the MobileNet-v2-FTNCA-SVM with an accuracy of 91.30% and training time of 276 s. Although MobileNet-v2-FTNCA-Naïve Bayes showed moderate performance, it still achieved an accuracy of 84.95% with a significantly lower feature count of 93 and a training time of 139 s.

The integration of MobileNet-v2 with FTNCA allowed for an automatic reduction of features from 1280 to a more optimal number, significantly improving the classification accuracy and efficiency of all classifiers compared to the baseline MobileNet-v2 accuracy of 79.47%.

Table 6 presents the performance of various machine learning algorithms when combined with ResNet-18-FTNCA for AD identification. ResNet-18-FTNCA-KNN demonstrates the most superior performance, achieving an exceptional accuracy of 98.44%. With a tolerance value of 0.08 and 180 features, ResNet-18-FTNCA-KNN also shows high sensitivity (98.52%) and specificity (99.31%), with a training time of just 51 s. This notable reduction in features from the original 512, thanks to FTNCA, enhances the classification performance significantly. ResNet-18-FTNCA-ANN achieves an accuracy of 93.33%, sensitivity of 94.92%, and specificity of 97.13%, using 335 features with a tolerance value of 0.02. The training time is 137 s. ResNet-18-FTNCA-SVM exhibits similar performance to the ANN classifier with an accuracy of 93.23%, sensitivity of 92.89%, and specificity of 97.12%, also using 335 features and a tolerance value of 0.02, with a training time of 63 s. ResNet-18-FTNCA-Naïve Bayes, despite its lower performance, still achieves an accuracy of 87.71%, sensitivity of 89.65%, and specificity of 94.93%, with a significantly lower feature count of 23 and a tolerance value of 0.43. The training time is 73 s.

In conclusion, the KNN classifier using the ResNet-18-FTNCA model clearly exhibits the highest performance, with a remarkable accuracy rate and efficient training time. Other classifiers also show improved performance compared to the baseline ResNet-18 accuracy of 81.02%, demonstrating the effectiveness of the FTNCA method in feature selection and enhancement of classification accuracy.

Table 7 presents the performance of various machine learning algorithms when combined with ResNet-50-FTNCA for AD detection. ResNet-50-FTNCA-KNN demonstrates the highest level of performance, achieving a perfect accuracy rate of 100%. The optimal tolerance value for KNN was determined to be 0.13, with an ideal number of 344 features. This significant reduction in features from the original 2048, thanks to FTNCA, leads to outstanding classification performance. The training time for KNN is 339 s.

Other classifiers also benefit from the ResNet-50-FTNCA approach: ANN achieves an accuracy of 97.76%, sensitivity of 98.26%, and specificity of 99.16%, using 1045 features with a tolerance value of 0.04. The training time is 248 s. SVM exhibits high performance with an accuracy of 98.23%, sensitivity of 98.61%, and specificity of 99.30%, using 1229 features and a tolerance value of 0.03, with a training time of 347 s. Naïve Bayes, despite showing lower performance, still achieves an accuracy of 90.16%, sensitivity of 88.97%, and specificity of 96.16%, with a feature count of 108 and a tolerance value of 0.24. The training time is 364 s.

In summary, the KNN classifier using the ResNet-50-FTNCA model demonstrates the most effective performance for detecting AD from MR images, with exceptional accuracy and efficient training time. Other classifiers also show significant performance improvements compared to the baseline ResNet-50 accuracy of 81.16%, highlighting the effectiveness of FTNCA in enhancing classification accuracy through optimal feature selection. Figures 4 and 5 illustrate the confusion matrix and tolerance value graph for the ResNet-50-FTNCA-KNN model, respectively, alongside the corresponding loss values.

Figure 4 displays the confusion matrix for the ResNet-50-FTNCA-KNN model. The confusion matrix is a visual representation of the classification performance, illustrating the true positives, true negatives, false positives, and false negatives for each class (mild demented, moderate demented, non-demented, very mild demented). In this matrix, the diagonal elements represent correct predictions, while off-diagonal elements represent incorrect predictions.

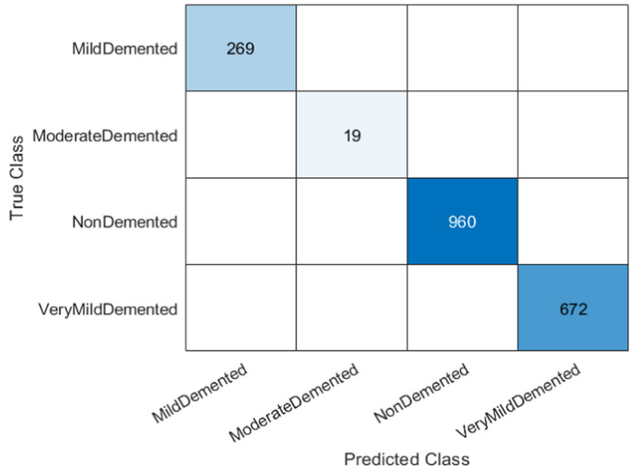


Fig. 4 Confusion matrix of ResNet-50-FTNCA-KNN

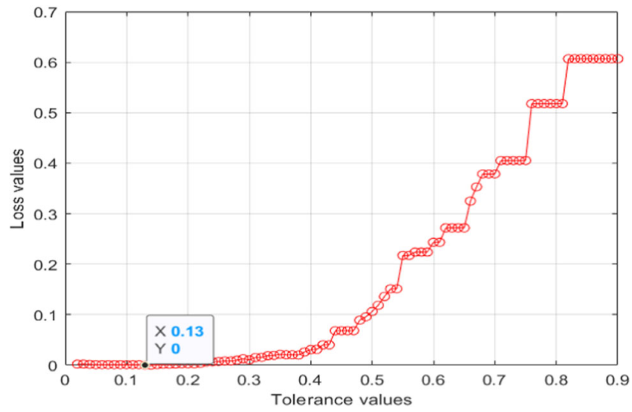


Fig. 5 The optimal tolerance value for ResNet-50-FTNCA-KNN is determined by minimizing the loss

This matrix indicates that the model achieved perfect classification with no errors, reflecting the 100% accuracy rate.

Figure 5 shows the relationship between tolerance values and loss values for the ResNet-50-FTNCA-KNN model. The x-axis represents tolerance values ranging from 0 to 0.9, while the y-axis represents loss values. The graph illustrates how different tolerance values affect the loss during the feature selection process. As seen in the graph, the loss remains minimal up to a tolerance value of 0.13, after which it increases significantly. This indicates that 0.13 is the optimal tolerance value for minimizing loss and maximizing classification performance.

Table 8 illustrates the performance metrics of various machine learning algorithms when integrated with ResNet-101-FTNCA for the detection of AD. ResNet-101-FTNCA-KNN exhibits the highest efficacy, attaining an impressive accuracy of 99.69%. The ideal tolerance value for KNN is identified as 0.05, with the optimal number of features being 842. This reduction from the initial 2048 features substantially enhances the classification performance. Notably, the training duration for KNN is 123 s. ResNet-101-FTNCA-ANN achieves a notable accuracy of 95.99%, with a sensitivity of 93.41% and specificity of 98.30%. Using 980 features with a tolerance value of 0.04, its training time stands at 149 s. Next, ResNet-101-FTNCA-SVM demonstrates a strong performance, achieving 96.82% accuracy, with a sensitivity of 97.75% and specificity of 98.64%. This model uses 1324 features and a tolerance value of 0.02, with a training time of 146 s. ResNet-101-FTNCA-Naïve Bayes, while exhibiting lower performance compared to others, still achieves an accuracy of 90.73%, sensitivity of 83.82%, and specificity of 96.13%. Utilizing 178 features and a tolerance value of 0.22, the training time for Naïve Bayes is 126 s.

In sum, the KNN classifier employing the ResNet-101-FTNCA model demonstrates superior performance with nearly perfect accuracy and efficient training. Other classifiers also exhibit notable improvements in accuracy and efficiency compared to the baseline ResNet-101 accuracy of 79.92%, underscoring the effectiveness of FTNCA in optimal feature selection and enhancement of classification accuracy.

Table 9 displays the performance results of various machine learning algorithms combined with ShuffleNet-FTNCA for AD detection. ShuffleNet-FTNCA-KNN is the most effective, achieving an accuracy of 98.44%. Its optimal tolerance value is 0.02, and it uses 404 features, significantly reducing the original 544 features and enhancing classification performance. The training time for ShuffleNet-FTNCA-KNN is 98 s. Next, ShuffleNet-FTNCA-ANN achieves 81.46% accuracy, with a sensitivity of 76.22% and specificity of 92.38%, using 273 features and a tolerance value of 0.05. The training time is 108 s. Moreover, ShuffleNet-FTNCA-SVM records an accuracy of 83.85%, sensitivity of 71.71%, and specificity of 93.42% with 404 features and a tolerance value of 0.02. The training time is 91 s. Naïve Bayes, while performing lower again, achieves 70.47% accuracy, 72.78% sensitivity, and 88.53% specificity, using 169 features and a tolerance value of 0.2. The training time is 59 s.

Briefly, the KNN classifier with the ShuffleNet-FTNCA model demonstrates superior performance with the highest accuracy and efficient training. Other classifiers also show improved results compared to the baseline ShuffleNet, which had an accuracy of 77.92%, highlighting the effectiveness of FTNCA in optimal feature selection and classification accuracy enhancement.

Based on the comprehensive analysis conducted, it can be concluded that the ResNet-50-FTNCA-KNN architecture demonstrates unparalleled performance in the detection of AD from MR images. This model achieves an exceptional accuracy of 100%, signifying its superior efficacy compared to other evaluated models. The optimal tolerance value for this architecture was empirically determined to be 0.13, and

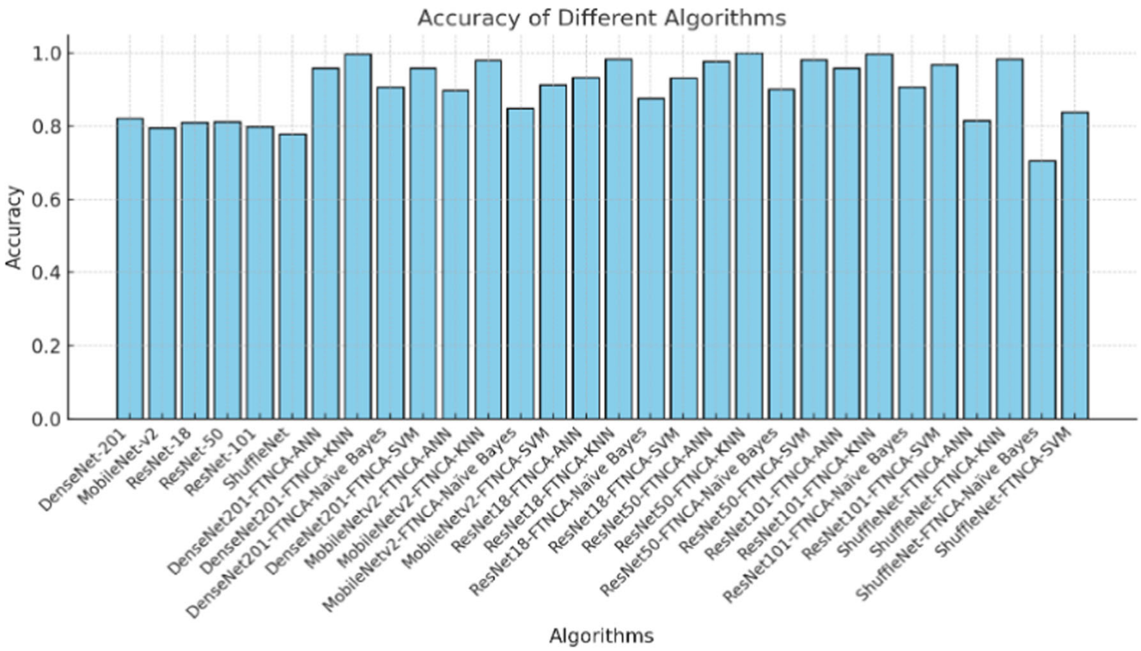


Fig. 6 Comparative of accuracy across different algorithms

the ideal number of features was identified as 344. This substantial reduction from the initial 2048 features to 344, facilitated by the FTNCA algorithm, significantly enhances the classification performance. The KNN classifier, integrated with the ResNet-50-FTNCA model, not only provides perfect accuracy but also ensures efficient processing with a training time of 339 s. This combination underscores the effectiveness of utilizing advanced feature selection techniques in conjunction with robust classification algorithms to achieve highly accurate and reliable outcomes in medical image analysis. Figure 6 presents a comparison of the analyses conducted.

Figure 6 presents a comparative analysis of the accuracy rates of various machine learning algorithms combined with FTNCA and different pretrained architectures, including DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet. The figure highlights the effectiveness of each algorithm in achieving high accuracy in AD detection from MR images, showcasing the superior performance of the ResNet-50-FTNCA-KNN model, which attained a perfect accuracy of 100%.

4 Discussion

Deep learning is recognized as fundamental to machine learning, particularly in addressing complex computer vision problems with CNNs [69]. This study employs pretrained CNN architectures (DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet) to classify stages of AD. Performance results from five fold cross-validation and average training times are summarized in Table 3. The study aims to distinguish four classes effectively, focusing on improving the efficiency and reducing the computational complexity of automatic AD detection. Therefore, pretrained architectures serve as feature extractors in this research.

The primary contribution of this work is the reduction of features extracted from the last layer of these networks using the FTNCA algorithm. Unlike traditional NCA, which relies on trial-and-error [76–78] for feature selection, FTNCA automates this process. The selected features vary based on the classifier (ANN, KNN, Naïve Bayes, SVM) and deep feature extractor, resulting in different error rates. The goal is to identify the optimal feature and algorithm structures to minimize errors in detecting AD from MR images.

Experimental results, particularly highlighted in Table 7, show that the ResNet-50-FTNCA-KNN model achieved a remarkable accuracy of 100%. Initially, the ResNet-50 model produced 2048 features from its last layer, which were reduced to 344 features using FTNCA, optimizing the tolerance value to minimize loss. This process is illustrated in Fig. 5. KNN's superior performance underscores its effectiveness in determining the best tolerance value and achieving minimal loss, as evidenced by the confusion matrix in Fig. 4. In a concise manner, the study's strengths, limitations, and time and memory complexity can be presented as follows:

Strengths: (i) The study utilizes a large, unbalanced dataset from Kaggle, comprising 896 mild dementia, 64 moderate dementia, 3200 nondementia, and 2240 very mild dementia

Table 10 Comparison of the proposed algorithm with state-of-the-art studies on identical data

Study	Dataset	Classes	Method	Accuracy (%)
Khalid et al. [33]	Kaggle	Mild dementia Moderate Nondementia Very mild dementia	DenseNet-121-handcrafted features-PCA-feedforward neural network (FFNN)	99.7
Lanjewar et al. [34]	Kaggle	Mild dementia Moderate Nondementia Very mild dementia	CNN-KNN	99.58
Roopa et al. [39]	Kaggle	Mild dementia Moderate Nondementia Very mild dementia	Deep CNN	95.80
Chabib et al. [40]	Kaggle	Mild dementia Moderate Nondementia Very mild dementia	Fast discrete curvelet transform-CNN	98.62
Proposed study	Kaggle	Mild dementia Moderate Nondementia Very mild dementia	ResNet-50-FTNCA-KNN	100

samples. (ii) The five fold cross-validation technique mitigates overfitting of CNN architectures. (iii) The FTNCA algorithm autonomously selects key features by optimizing tolerance values to minimize loss. (iv) As shown in Tables 3, 4, 5, 6, 7, 8, and 9, KNN consistently outperforms other classifiers. (v) The ResNet-50-FTNCA-KNN model is particularly effective, achieving 100% accuracy and aiding in the early detection of AD, benefiting healthcare professionals.

The study has limitations, notably that the proposed algorithm is tested on a publicly available dataset focusing solely on axial MR images. To ensure generalizability, it should be evaluated with various types of brain imaging. Additionally, reliance on publicly available datasets may limit data diversity and variability, potentially impacting the robustness of the findings.

The time complexity of the FTNCA algorithm primarily depends on the dataset size and the number of features. Memory complexity is influenced by the storage requirements for feature vectors and the distance matrix. Efficient implementation and dimensionality reduction significantly alleviate these complexities, making the algorithm suitable for large-scale datasets.

As a result, integrating FTNCA with pretrained CNN architectures demonstrates significant potential for enhancing the accuracy and efficiency of AD detection from MR images, as evidenced by the exceptional performance of the ResNet-50-FTNCA-KNN model. Table 10 presents a comparison proposed algorithm ResNet-50-FTNCA-KNN with previous studies.

Table 10 compares the proposed ResNet-50-FTNCA-KNN algorithm with various state-of-the-art studies using identical data from Kaggle, encompassing four classes of Alzheimer's disease detection: mild dementia, moderate dementia, nondementia, and very mild dementia. The proposed ResNet-50-FTNCA-KNN model outperforms all other methods, achieving a perfect accuracy of 100%. This underscores the efficacy of combining ResNet-50 with the FTNCA feature reduction technique and the KNN classifier, demonstrating superior performance in detecting AD from MR images.

In summary, the proposed ResNet-50-FTNCA-KNN algorithm not only surpasses the accuracy rates of current state-of-the-art models but also offers a robust and efficient approach for early detection of Alzheimer's disease, thereby potentially aiding healthcare professionals in making timely and accurate diagnoses.

5 Conclusion

The objective of this study is to develop a robust method for the automatic detection of AD by integrating deep learning, FTNCA, and machine learning algorithms. CNNs are utilized as deep feature extractors due to their proficiency in extracting features without manual intervention. Six pretrained architectures—DenseNet-201, MobileNet-v2, ResNet-18, ResNet-50, ResNet-101, and ShuffleNet—are employed. The last layers of these architectures generate deep features. The FTNCA algorithm, that is proposed in this study, is subsequently used to select the most informative features. These selected features are then classified using four different machine learning algorithms: ANN, KNN, Naïve Bayes, and SVM. Among these, the ResNet-50-FTNCA-KNN model stands out, reducing the feature set from 2048 to 344 features by optimizing the tolerance value to minimize loss. This model achieves exceptional performance metrics, including 100% accuracy, sensitivity, specificity, precision, F1-score, MCC, and kappa.

Additionally, when KNN is combined with FTNCA and other pretrained architectures, the accuracy exceeds 98.03%. This underscores the efficacy of the proposed methodology in classifying AD using MR images, as demonstrated by the highest accuracy rates achieved in this study. Future work should evaluate this approach using various types of brain imaging to ensure broader applicability and validation.

This integration of deep learning and feature selection techniques not only enhances classification accuracy but also significantly reduces computational requirements, providing a practical and efficient solution for early AD detection.

Author Contributions Öznur Özaltin designed this article and analyzed the data.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). There was no external funding for this article.

Data Availability The dataset was obtained from Kaggle at https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images (accessed on January 14, 2023).

Declarations

Conflict of Interest The author declares no competing financial interests or personal relationships in the presented article.

Ethical Approval This article contains no studies with human participants or animals performed by the author.

References

1. Orouskhani, M.; Zhu, C.; Rostamian, S.; Zadeh, F.S.; Shafiei, M.; Orouskhani, Y.: Alzheimer's disease detection from structural MRI using conditional deep triplet network. Neurosci. Inform. **2**(4), 100066 (2022)
2. Kaplan, E.; Baygin, M.; Barua, P.D.; Dogan, S.; Tuncer, T.; Altunisik, E.; Palmer, E.E.; Acharya, U.R.: ExHiF: Alzheimer's disease detection using exemplar histogram-based features with CT and MR images. Med. Eng. Phys. **115**, 103971 (2023)
3. Subasi, A.: Use of artificial intelligence in Alzheimer's disease detection. Artif. Intell. Precis. Health 257–278 (2020)
4. Subasi, A.; Kapadnis, M. N.; Bulbul, A. K.: Alzheimer's disease detection using artificial intelligence. In: Augmenting Neurological Disorder Prediction and Rehabilitation Using Artificial Intelligence. Elsevier (2022)
5. Jin, J.: Alzheimer disease. JAMA **313**(14), 1488–1588 (2015)
6. AbdulAzeem, Y.; Bahgat, W.M.; Badawy, M.: A CNN based framework for classification of Alzheimer's disease. Neural Comput. Appl. **33**, 10415–10428 (2021)
7. Ozaltin, O.; Yeniay, O.; Subasi, A.: Artificial intelligence-based brain hemorrhage detection. In: Accelerating Strategic Changes for Digital Transformation in the Healthcare Industry. Elsevier (2023)
8. Ozaltin, O.; Coskun, O.; Yeniay, O.; Subasi, A.: Classification of brain hemorrhage computed tomography images using OzNet hybrid algorithm. Int. J. Imaging Syst. Technol. **33**(1), 69–91 (2023)
9. Ozaltin, O.; Coskun, O.; Yeniay, O.; Subasi, A.: A deep learning approach for detecting stroke from brain CT images using OzNet. Bioengineering **9**(12), 783 (2022)
10. Ozaltin, O.; Yeniay, O.: Detection of monkeypox disease from skin lesion images using Mobilenetv2 architecture. Commun. Fac. Sci. Univ. Ankara Ser. A1 Math. Stat. **72**(2), 482–499 (2023)
11. Ozaltin, O.; Yeniay, O.: A novel proposed CNN–SVM architecture for ECG scalograms classification. Soft. Comput. **27**(8), 4639–4658 (2023)
12. Al-Fahoum, A.S.; Al-Fraihat, A.A.: Methods of EEG signal features extraction using linear analysis in frequency and time-frequency domains. Int. Sch. Res. Notices **2014**(1), 730218 (2014)
13. Khader, A.; Zyout, A.A.; Al Fahoum, A.: Combining enhanced spectral resolution of EMG and a deep learning approach for knee pathology diagnosis. PLOS ONE **19**(5), e0302707 (2024)
14. Al Fahoum, A.: Complex wavelet-enhanced convolutional neural networks for electrocardiogram-based detection of paroxysmal atrial fibrillation. Adv. Signal Process. Artif. Intell. **158** (2024)
15. Al, F.A.: Early detection of neurological abnormalities using a combined phase space reconstruction and deep learning approach. Intell. Based Med. **8**, 100123 (2023)
16. Al Fahoum, A.; Al Omari, A.; Al Omari, G.; Zyout, A. A.: PPG signal-based classification of blood pressure stages using wavelet transformation and pre-trained deep learning models. In: 2023 Computing in Cardiology Conference (CinC). of Conference Year.
17. Al, F.A.: Wavelet transform, reconstructed phase space, and deep learning neural networks for EEG-based schizophrenia detection. Int. J. Neural Syst. **34**(9), 2450046 (2024)

18. Kasmaiee, S.; Tadjfar, M.: Non-circular pressure swirl nozzles injecting into stagnant air. Int. J. Multiph. Flow **175**, 104798 (2024)
19. Ozaltin, O.; Yeniay, O.; Subasi, A.: OzNet: A new deep learning approach for automated classification of COVID-19 computed tomography scans. Big Data (2023)
20. Subasi, A.; Ozaltin, O.; Mitra, A.; Subasi, M. E.; Sarirete, A.: Trustworthy artificial intelligence in healthcare. In: Accelerating Strategic Changes for Digital Transformation in the Healthcare Industry. Elsevier (2023)
21. Kasmaiee, S.; Tadjfar, M.: Elliptical pressure swirl jet issuing into stagnant air. Phys. Fluids **36**(7) (2024)
22. Kasmaiee, S.; Tadjfar, M.; Kasmaiee, S.; Ahmadi, G.: Linear stability analysis of surface waves of liquid jet injected in transverse gas flow with different angles. Theoret. Comput. Fluid Dyn. **38**(1), 107–138 (2024)
23. Liu, J.; Li, D.; Shan, W.; Liu, S.: A feature selection method based on multiple feature subsets extraction and result fusion for improving classification performance. Appl. Soft Comput. **150**, 111018 (2024)
24. Sethi, M.; Rani, S.; Singh, A.; Mazón, J. L. V.: A CAD system for Alzheimer’s disease classification using neuroimaging MRI 2D slices. Comput. Math. Methods Med. **2022** (2022)
25. Ismail, M. M. B.; Alabdullatif, R.; Bchir, O.: Alzheimer’s disease detection using neighborhood components analysis and feature selection. Int. J. Adv. Comput. Sci. Appl. **11**(10) (2020).
26. Al-Fahoum, A.; Zyout, A.: Enhancing early detection of schizophrenia through multi-modal EEG analysis: a fusion of wavelet transform, reconstructed phase space, and deep learning neural networks. In: 5th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI’2023). of Conference (2023)
27. Jin, M.; Deng, W.: Predication of different stages of Alzheimer’s disease using neighborhood component analysis and ensemble decision tree. J. Neurosci. Methods **302**, 35–41 (2018)
28. Ozbay, F.A.; Ozbay, E.: An NCA-based Hybrid CNN model for classification of Alzheimer’s disease on grad-CAM-enhanced brain MRI images. Turk. J. Sci. Technol. **18**(1), 139–155 (2023)
29. Kaplan, E.; Dogan, S.; Tuncer, T.; Baygin, M.; Altunisik, E.: Feed-forward LPQNet based automatic alzheimer’s disease detection model. Comput. Biol. Med. **137**, 104828 (2021)
30. Asgharzadeh-Bonab, A.; Kalbkhani, H.; Azarfardian, S.: An Alzheimer’s disease classification method using fusion of features from brain Magnetic Resonance Image transforms and deep convolutional networks. Healthcare Anal. 100223 (2023)
31. AlSaeed, D.; Omar, S.F.: Brain MRI analysis for Alzheimer’s disease diagnosis using CNN-based feature extraction and machine learning. Sensors **22**(8), 2911 (2022)
32. Hazarika, R.A.; Kandar, D.; Maji, A.K.: An experimental analysis of different deep learning based models for Alzheimer’s disease classification using brain magnetic resonance images. J. King Saud Univ. Comput. Inf. Sci. **34**(10), 8576–8598 (2022)
33. Khalid, A.; Senan, E.M.; Al-Wagih, K.; Ali Al-Azzam, M.M.; Alkhraisha, Z.M.: Automatic analysis of MRI images for early prediction of Alzheimer’s disease stages based on hybrid features of CNN and handcrafted features. Diagnostics **13**(9), 1654 (2023)
34. Lanjewar, M.G.; Parab, J.S.; Shaikh, A.Y.: Development of framework by combining CNN with KNN to detect Alzheimer’s disease using MRI images. Multimed. Tools Appl. **82**(8), 12699–12717 (2023)
35. Shojaei, S.; Abadeh, M.S.; Momeni, Z.: An evolutionary explainable deep learning approach for Alzheimer’s MRI classification. Expert Syst. Appl. **220**, 119709 (2023)
36. Mohi ud din dar, G.; Bhagat, A.; Ansarullah, S.I.; Othman, M.T.B.; Hamid, Y.; Alkahtani, H.K.; Ullah, I.; Hamam, H.: A novel framework for classification of different Alzheimer’s disease stages using CNN model. Electronics **12**(2), 469 (2023)
37. Agarwal, D.; Berbís, M.Á.; Luna, A.; Lipari, V.; Ballester, J.B.; de la Torre-Díez, I.: Automated medical diagnosis of Alzheimer´s disease using an efficient net convolutional neural network. J. Med. Syst. **47**(1), 57 (2023)
38. Venkatasubramanian, S.; Dwivedi, J. N.; Raja, S.; Rajeswari, N.; Logeshwaran, J.; Praveen Kumar, A.: Prediction of Alzheimer’s disease using DHO-based pretrained CNN model. Math. Probl. Eng **2023** (2023)
39. Roopa, Y. M.; Reddy, B. B.; Babu, M. R.; Nayak, R. K.: Teaching learning-based brain storm optimization tuned Deep-CNN for Alzheimer’s disease classification. Multimed. Tools Appl. 1–24 (2023)
40. Chabib, C., Hadjileontiadis, L. J.; Al Shehhi, A.: DeepCurvMRI: deep convolutional curvelet transform-based MRI approach for early detection of Alzheimer’s disease. IEEE Access (2023)
41. Abbas, S.Q.; Chi, L.; Chen, Y.-P.P.: Transformed domain convolutional neural network for Alzheimer’s disease diagnosis using structural MRI. Pattern Recogn. **133**, 109031 (2023)
42. Kaggle. 14.01.2023 (2023). https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images
43. Ha, W.; Vahedi, Z.: Automatic breast tumor diagnosis in MRI based on a hybrid CNN and feature-based method using improved deer hunting optimization algorithm. Comput. Intell. Neurosci. **2021** (2021)
44. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.: Recent advances in convolutional neural networks. Pattern Recogn. **77**, 354–377 (2018)
45. LeCun, Y.A.; Bottou, L.; Orr, G.B.; Müller, K.-R.: Efficient BackProp. In: Montavon, G.; Orr, G.B.; Müller, K.-R. (Eds.) Neural Networks: Tricks of the Trade, 2nd edn. Springer, Berlin (2012)
46. Nair, V.; Hinton, G. E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). of Conference; Year.
47. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition of Conference; Year.
48. Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications (2017). arXiv:1704.04861
49. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition of Conference Year
50. He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition of Conference Year.
51. Zhang, X.; Zhou, X.; Lin, M.; Sun, J.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition of Conference Year
52. Ghosal, P., Nandanwar, L., Kanchan, S., Bhadra, A., Chakraborty, J., Nandi, D.: Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network. In: 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP) of Conference IEEE Year.
53. Chandola, Y.; Virmani, J.; Bhadauria, H.; Kumar, P.: Chapter 4-Deep Learning for Chest Radiographs: Computer-Aided Classification. Elsevier (2021)
54. Shang, Q.; Tan, D.; Gao, S.; Feng, L.: A hybrid method for traffic incident duration prediction using BOA-optimized random forest combined with neighborhood components analysis. J. Adv. Transp. **2019** (2019)
55. Koc, M.; Sut, S.K.; Serhatlioglu, I.; Baygin, M.; Tuncer, T.: Automatic prostate cancer detection model based on ensemble VGGNet

feature generation and NCA feature selection using magnetic resonance images. Multimed. Tools Appl. **81**(5), 7125–7144 (2022)

56. Tuncer, T.; Ertam, F.: Neighborhood component analysis and reliefF based survival recognition methods for Hepatocellular carcinoma. Physica A **540**, 123143 (2020)
57. Dogan, S.; Barua, P.D.; Kutlu, H.; Baygin, M.; Fujita, H.; Tuncer, T.; Acharya, U.R.: Automated accurate fire detection system using ensemble pretrained residual network. Expert Syst. Appl. **203**, 117407 (2022)
58. Barua, P.D.; Yildiz, A.M.; Canpolat, N.; Keles, T.; Dogan, S.; Baygin, M.; Tuncer, I.; Tuncer, T.; Tan, R.-S.; Fujita, H.: An accurate automated speaker counting architecture based on James Webb Pattern. Eng. Appl. Artif. Intell. **119**, 105821 (2023)
59. Goldberger, J.; Hinton, G. E.; Roweis, S.; Salakhutdinov, R. R.: Neighbourhood components analysis. Adv. Neural Inf. Process. Syst. **17** (2004)
60. Yang, W.; Wang, K.; Zuo, W.: Neighborhood component feature selection for high-dimensional data. J. Comput. **7**(1), 161–168 (2012)
61. Tuncer, T.; Dogan, S.; Özyurt, F.; Belhaouari, S.B.; Bensmail, H.: Novel multi center and threshold ternary pattern based method for disease detection method using voice. IEEE Access **8**, 84532–84540 (2020)
62. Karadal, C.H.; Kaya, M.C.; Tuncer, T.; Dogan, S.; Acharya, U.R.: Automated classification of remote sensing images using multileveled MobileNetV2 and DWT techniques. Expert Syst. Appl. **185**, 115659 (2021)
63. Akbal, E.; Barua, P.D.; Dogan, S.; Tuncer, T.; Acharya, U.R.: DesPatNet25: data encryption standard cipher model for accurate automated construction site monitoring with sound signals. Expert Syst. Appl. **193**, 116447 (2022)
64. Dogan, S.; Baygin, M.; Tasci, B.; Loh, H.W.; Barua, P.D.; Tuncer, T.; Tan, R.-S.; Acharya, U.R.: Primate brain pattern-based automated Alzheimer's disease detection model using EEG signals. Cogn. Neurodyn. **17**(3), 647–659 (2023)
65. Tuncer, T.; Dogan, S.; Acharya, U.R.: Automated accurate speech emotion recognition system using twine shuffle pattern and iterative neighborhood component analysis techniques. Knowl. Based Syst. **211**, 106547 (2021)
66. Tuncer, T.; Ozyurt, F.; Dogan, S.; Subasi, A.: A novel Covid-19 and pneumonia classification method based on F-transform. Chemom. Intell. Lab. Syst. **210**, 104256 (2021)
67. Tuncer, T.: A new stable nonlinear textural feature extraction method based EEG signal classification method using substitution Box of the Hamsi hash function: Hamsi pattern. Appl. Acoust. **172**, 107607 (2021)
68. Baygin, M.; Barua, P.D.; Chakraborty, S.; Tuncer, I.; Dogan, S.; Palmer, E.; Tuncer, T.; Kamath, A.P.; Ciaccio, E.J.; Acharya, U.R.: CCPNet136: automated detection of schizophrenia using carbon chain pattern and iterative TQWT technique with EEG signals. Physiol. Meas. **44**(3), 035008 (2023)
69. Poyraz, A.K.; Dogan, S.; Akbal, E.; Tuncer, T.: Automated brain disease classification using exemplar deep features. Biomed. Signal Process. Control **73**, 103448 (2022)
70. McCulloch, W.S.; Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4), 115–133 (1943)
71. Fix, E.; Hodges, J. L.: Discriminatory analysis, nonparametric discrimination: consistency properties. In: Field, R. (ed) Technical Report 4. USAF School of Aviation Medicine, Texas (1951)
72. McCallum, A.; Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization of Conference. Madison, WI Year
73. Cortes, C.; Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
74. Saber, A.; Sakr, M.; Abo-Seida, O.M.; Keshk, A.; Chen, H.: A novel deep-learning model for automatic detection and classification of breast cancer using the transfer-learning technique. IEEE Access **9**, 71194–71209 (2021)
75. Al, F.A.: Enhanced cardiac arrhythmia detection utilizing deep learning architectures and multi-scale ECG analysis. Tuijin Jishu/J. Propuls. Technol. **44**(6), 5539–5554 (2023)
76. Khatri, S., Bansal, P.: Hyperparameter tuning and validation of neural network model for software effort estimation with NCA based feature selection. In: 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS) of Conference: IEEE Year.
77. Bu, L.; Du, G.; Hou, Q.: Prediction of the compressive strength of recycled aggregate concrete based on artificial neural network. Materials **14**(14), 3921 (2021)
78. Lou, Z.; Wang, Y.: New nonlinear approach for process monitoring: Neural component analysis. Ind. Eng. Chem. Res. **60**(1), 387–398 (2020)