Optimized Hybrid Approach for Early Detection of Alzheimer's Disease Using Machine Learning and Deep Learning Technique

Eshat Ahmad Shuvo[a], Wahidur Rahman[a], Pabon Shaha[b], Md. Sabbir Hossen[a], Anichur Rahman[c,d,*], Amin Mohammad Wasiuzzaman[d], Fahmid Al Farid[e,f,*], Hezerul Abdul Karim[f,*], Abu Saleh Musa Miah[g]

[a]*Department of Computer Science and Engineering, Bangladesh University, Dhaka, BD*
[b]*Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University, Bangladesh, Dhaka, Tangail, 1902, BD*
[c]*Department of Computer Science and Engineering, NITER, Constituent Institute of Dhaka University, Bangladesh, Dhaka, Savar, 1350, BD*
[d]*College of Engineering and Computing, Georgia Southern University, Statesboro, GA 30458, Georgia, USA,*
[e]*Faculty of Computer Science and Informatics, Berlin School of Business and Innovation Karl-Marx-Straße 97-99, Berlin, 12043, Germany,*
[f]*Centre for Image and Vision Computing (CIVC), COE for Artificial Intelligence, Faculty of Artificial Intelligence and Engineering (FAIE), Multimedia University, Cyberjaya 63100, Selangor, Malaysia, Dhaka, Savar, 1350, BD*
[g]*School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu, Fukushima, Japan.,*

Abstract

Alzheimer's disease (AD) is an emerging and fatal neurological disorder that is characterized by slowly progressive cognitive deterioration and memory loss. Hence, early identification of the disease and timely intervention are key to managing AD and ensuring minimal deterioration of the condition. In the past, diagnosis was usually done clinically by neurologists with the help of physical examination, including neuro-imaging, which is also efficient; however, it is time-consuming and tiresome. In recent years, the use of AI-enabled computer vision as the main technique for image processing and analysis has received much interest. Prior studies using AI-ML approaches have noted improved accuracy, but they entail significant computational resources and massive data. This paper presents a cascaded network topology employing both DL and conventional ML methods, optimized by using LDA for AD diagnosis from neuroimaging data. It was established that the proposed approach provided an accuracy of 98.32% training the model on a small amount of data and with low hardware resources. The five AI models were chosen for the cascaded networks through a compilation of five well-known CNN models for feature extraction. LDA was used to find out the best features for classification and seven conventional machine learning models for AD detection. The best network among them was recognized to be the DenseNet201-LDA-KNN cascaded network in this study.

Keywords: Alzheimer's disease (AD), Linear Discriminant Analysis (LDA), Bacterial Foraging Optimization (BFO), Machine learning (ML), Deep Learning (DL).

1. Introduction

Alzheimer's disease (AD), a brain condition that stands globally as one of the fatal neuro's degenerative disorders; acting as a memory disorder causing decline and ultimately leading to a loss of independence. AD is characterized by the buildup of protein deposits in the brain causing widespread neuron death and the formation of distinct plaques and tangles. Addressing AD remains a health challenge today with an estimated 50 million individuals grappling with this enduring condition. As life expectancy continues to rise, the prevalence of Alzheimer's disease is projected to triple by 2050 placing a strain on healthcare and economic systems [1]. Early and precise diagnosis plays a role in facilitating treatment and care for AD; however, current diagnostic approaches rely heavily on subjective clinical evaluations and costly neuro-imaging technologies [2]. Alzheimer's disease is a type of dementia that is chronic and continues to worsen over time leading to disability and death affecting mainly the memory, cognitive abilities and psychosocial behavior of patients. It is undeniably a world health issue as it is a fatal disease that marks the appearance of amyloid

*Corresponding author: Fahmid Al Farid (fahmid.alfarid@berlinsbi.com); Anichur Rahman (ar36248@georgiasouthern.edu); Hezerul Abdul Karim (hezerul@mmu.edu.my).

plaques and neuro-fibrillary tangles in the brain leading to massive neuronal fatalities and cerebral tissue degeneration. AD is not only a profound and emotional issue for the patients and their families, but it is also a major issue for healthcare systems and economies worldwide. According to the present statistics, more than 50 million people suffer from Alzheimer's Disease and this figure will soon triple due to the increase of people's average life expectancy. Such a pessimistic prognosis urges for the methods required for AD treatment and diagnostics in the development of novel diagnostic and therapeutic tools [3]. The accurate and early diagnosis of AD is of great significance for the beginning of the required treatment and subsequent improvement of the patient's life quality. However, the existing diagnostic approaches present quite a few problems. For instance, the global application of standard clinical measures, including brief cognitive testing or neuro-imaging like MRI and PET scans are impractical, costly and mostly limited by time constraints and unavailability. These restraints make it increasingly crucial to develop cheaper, faster diagnostics and less reliant upon sophisticated equipment or highly trained individuals [4]. Over the last few years, AI has established itself as a progressive medium in the detection and segmentation of abnormalities and the classification of specific diseases in the medical imaging system. Artificial intelligence interventions take advantage of highly sophisticated analyzers of human ailments through complex algorithms in the images. In the case of Alzheimer's detection, a number of manuscripts have applied machine learning (ML) algorithms by using features such as the texture, shape and structure derived from images of the brain. Although the use of ML techniques is advantageous in terms of computational costs and relatively good performance for small and moderate-sized data, it is, in many cases, critically oriented to use features extracted manually, which leads to low accuracy and problems connected with absolutely high dimensionality. On the flip side, the DL such as CNN proved to be the most accurate because, unlike other methods; it learned features from the lower layer to the higher from the original inputs. However, these methods typically need annotated datasets for training, which is a major issue occurring in medical image analysis [5]. In an attempt to overcome the drawbacks of implementing these individual approaches, this research proposes a novel cascaded diagnostic model that integrates features of both machine and deep learning. The described framework uses existing CNN models for the analysis of the defined features from brain MRI images and fine-tunes this set of features by means of BFO and PCA algorithms. Furthermore, Linear Discriminant Analysis (LDA) using sample mean and scatter matrix for each class is applied to reduce the dimensions and maintain the algorithm's effectiveness in terms of computational time. To ensure comprehensive analysis, the study investigates a couple of experimental setups. The major contributions of our research are as follows:

- We introduce a cascaded network of integrated DL and ML approaches that would enhance the diagnosis of Alzheimer's Disease.
- Five pre-trained CNN models, including MobileNetV2, DaseNet201, InceptionV3, Xception and DaseNet201, were used to extract the raw feature from the data set and later to classify seven of the most dominant classifiers (Support Vector Classifier, Random Forest, Decision Tree, Gaussian naive bias, Extreme gradient boosting, K-nearest neighbors and Logistic Regression).
- We utilize bio-inspired methodology for bacterial foraging optimization for selection of features, linear discriminant analysis, and Principal component analysis for reduction to identify the most optimized features from the raw features.
- One of the major aims of this study was to build a low-weight model that can perform in minimal hardware resources and small datasets compared to DL methods.

The headings represent the rest of the section of our research: Section 2: Related work shows the review of the literature of existing research based on Alzheimer's disease. Section 3: Methodology This section explains the system architecture, related diagrams, and requirements. Section 4: Experimental Result shows the results with different tables, graphs, and figures. Section 5: Conclusion and Future Prospects presents the article's core idea, which is concluded in this section, and discusses potential future applications.

2. Literature Review

Rahmeh Ibrahim et al [6] presented a multi-architecture CNN approach combined with PSO to detect AD. After careful analysis, three different datasets were combined. CNN with PSO performed exceptionally well according to their findings, as seen by its astounding 98.50%, 98.83%, and 97.12% testing accuracy for three different datasets. Gargi Pant Shukla, et al. [7] employed three learning algorithms for AD classification: random forest, XGBoost and

Convolution Neural Networks (CNN) and their best out-come was 97.57% accuracy and 97.60% sensitivity. In fact, the study was not illustrated comprehensively rather than in brief. Ebrahimi et al. [8] used a pre-trained ResNet in their deep sequencing model for the classification of Alzheimer's Disease (AD). Their approach was able to get the accuracy of about 91.78% which improved the existing other CNN-based methods by 10%. Likewise, Kang et al. [9] designed an architecture of deep ensemble 2D CNN based on multiple models and image slides. The classification accuracy of the model was 90.36% for NC and AD, 77.19% for AD and MCI and 72.36% for NC and MCI discrimination. Hridhee et al. [10] suggested a 2D CNN model for the early diagnosis of AD utilizing magnetic resonance imaging (MRI) data. Their technique involved using different image processing and augmentation methods then training the data using VGG16, Xception and a unique model. The custom model had a higher accuracy with a level of 94.77% in contrast with the other models. Allada et al. [11] proposed a swarm multi-verse algorithm to improve the deep neuro-fuzzy network for classifying AD in different stages. Pre-processing was done using median filtering in order to improve the quality of images, while multichannel feature pyramid network was used for segmentation and CNN for feature extraction. The final model had an accuracy of 89.9% based on the k-fold validation. Gowhar et al. [12] proposed a deep learning framework using CNN for early detection of AD. The technique incorporated elaborative pre-processing and augmentation methods; feature extraction, dimensionality reduction and classification. Their model had a high accuracy, recorded to be 96.22% which was found to be better compared to other classical methods. Tufail et al. [13] developed an early-stage AD classifier employing 2D and 3D CNNs with PET at the beginning of this year. Their experiment performed well and their findings provided higher accuracy rates at 89.21% for the 3D CNN model in regards to the AD/NC binary classification. They used data enhancement and 5-fold cross-validation as techniques to improve the productivity of the algorithm. To this end, Alinsaif and Lang [14] proposed a new method, which applied 3D Shearlets Descriptors for data dimensionality reduction of MRI datasets with significant efficiency in identifying Alzheimer's Disease (AD). AD is discussed in detail in Mahendran and P. M. [15], was developed using a framework that has an accuracy of 88.7% even when the data used in developing the framework was limited. For instance, Zhang et al. [16] developed a multi-model based model involving PET and MRI data of the study cohort from ADNI showed better capability in early AD diagnosis. Spasov et al. [17] applied a parameter-efficient solution for the prevention of MCI to AD progression. This model required other types of data such as MRI, demographic, neuro-psychological, genetics data and was also shown to be expandable as newer imaging data such as PET scans could also be added and their accuracy was 95.89%. Liu et al. [18] proposed a CNN model specifically for AD detection with limited MRI data and with 78.02% accuracy with improved portability. Janghel and Rathore [19] used some basic data pre-processing operations on image data sets before applying CNN feature extraction. They noted a level of accuracy of 99.95% when compared with fMRI data and 73.46% for the PET data. Transfer learning was employed in 3D CNN by Bae et al. [20] to predict MCI conversion to AD with an accuracy of 82.4%, which stands higher than other existing models.

Table 1: Summary table of existing works on Alzheimer's disease detection using various CNN models.

Ref.	Author	Method	Dataset	Accuracy (%)	Limitations
[10]	Hridhee et al. [2023]	CNN	ADNI	94.77	No optimization, complexity and limited model used.
[11]	Allada et al. [2023]	Neuro-fuzzy	ADNI	89.9	Lower accuracy and no complexity.
[12]	Gowhar et al. [2023]	VGG16 & ResNet50	ADNI	96.22	No complexity.
[13]	Tufail et al. [2022]	CNN	ADNI	89.21	No complexity.
[14]	Alinsaif et al. [2022]	3D Shearlets	ADNI	88.7	Lower accuracy.
[16]	Zhang et al. [2022]	ResNet	ADNI	90	No complexity.
[17]	Spasov et al. [2022]	CNN	ADNI	95.89	No complexity.
[18]	J. Liu et al. [2021]	AlexNet & GoogLeNet	OASIS	78.02	Lower accuracy, no complexity and optimization.
[20]	Bae et al. [2021]	3D CNN	ADNI	82.4	Lower accuracy, no complexity and optimization.

In Table 1, a brief summary has been illustrated to summarize the overall literature review of the other AD existing models.

3. Methodology

The general layout of the proposed methodology is presented in a graphical form in Figure 1. This comprehensive approach is divided into four interconnected phases: acquiring the research dataset, data pre-processing, layout of the proposed system structure and choosing the best model for the proposed network.

Figure 1 Workflow diagram of the model success and credibility of the final model as well as to minimize the number of errors. For the first analysis, the Alzheimer's disease (AD) dataset was collected from a reliable source

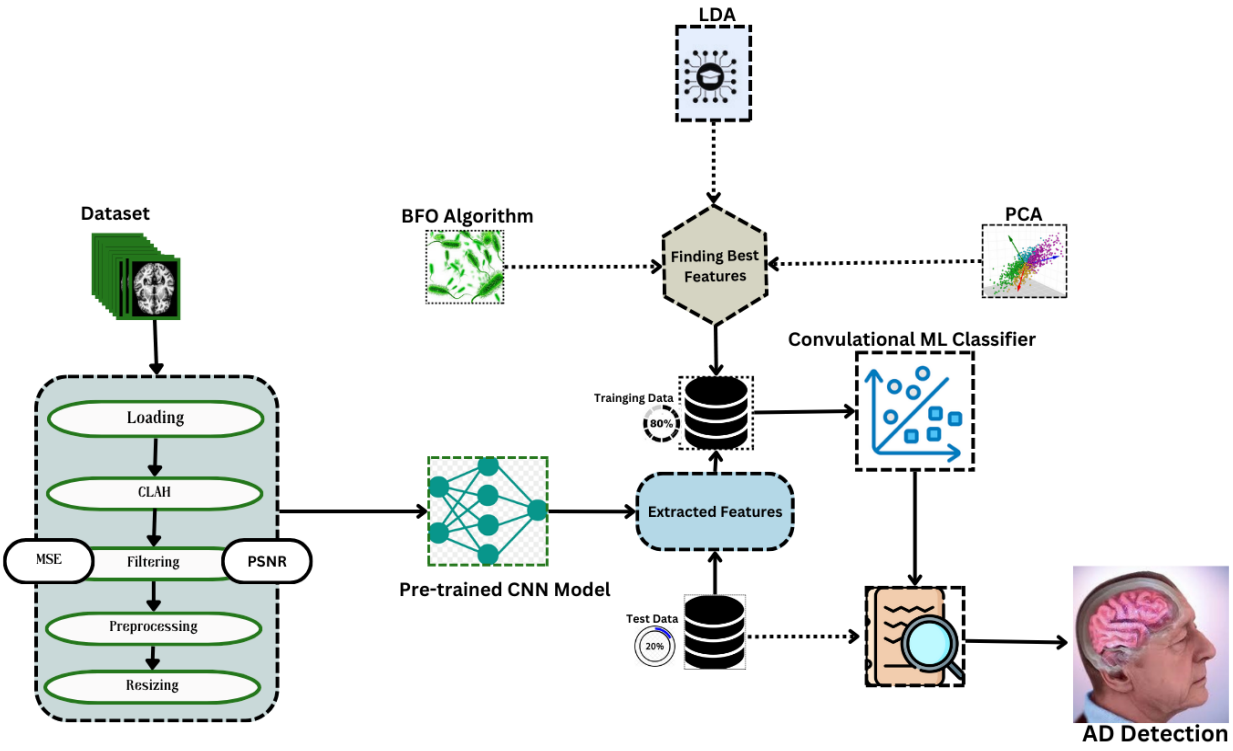


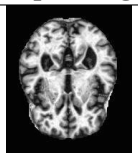
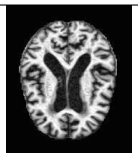
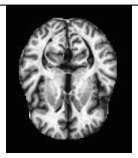
Figure 1: Workflow diagram of the proposed model.

of secondary data resource platforms. These data constitute the basis for the study; therefore, its preparation was deemed necessary to determine its suitability for further analysis. In the pre-processing stage, every advanced image pre-processing technique was applied to enhance the quality and formats of the data set. Among them, one of the methods used is the Contrast Limited Adaptive Histogram Equalization (CLAHE), which increases the contrast and excludes the noise in the images [21]; so, the dataset is improved for analysis. Moreover, any cases of missing values in the data set were taken to ensure that data was handled credibly throughout the process. When all the pre-processing was done, the enhanced and cleaned images were passed into different Convolutional Neural Network (CNN) models to obtain high-quality features. These pre-trained models use their layers to decompose the medical images to identify smaller features, that are essential for classification or analysis. Subsequently, feature selection and reduction methods were used to extract the most differentiating and significant features out of the obtained set. It helps to select features of high quality and informative for model enhancement, as well as minimize computational problems. After that, the dataset was divided into train and test splits to help progress the model building and assessment. Hence, the training set was used to train different types of learning classifiers, which have different characteristics well-suited for the task. The classifiers used in the model were:

3.1. Dataset

The brain MRI images on which this study was conducted were obtained from a public dataset [22]. Each of these images was thoroughly assessed by a professional neurosurgeon to determine the nature and stage of Alzheimer's disease (AD). The images for each patient were rated on a scale of 0 to 2; where 2 refers to Very Mild Demented and 1 refers to Mild Demented while code 0 was for Non-Demented. In the present study, only pictures that were assigned the rating of 2, 1 or 0 were used to remain fully focused on these particular stages. Consequently, the quality of the images used in this study was methodically scrutinized for effectiveness in the training of machine learning models. This evaluation was performed concerning the image quality assessment method proposed by Yagi et al. [16]; it estimates the sharpness and noise of images. Utilizing this method, we ensured that the dataset was comprised of high-quality images that could enable accurate model development and analysis. The clip collected dataset was of 1693 Non-Demented, 731 Mild Demented and 1833 Very Mild Demented MRI images which could be used for analysis and training datasets. Some of the contrasting examples from each category are presented in Table I to exemplify the variations, differences in image quality, and diagnostic labeling.

Table 2: Class distribution of the sample images.

Sample Images	Class Name	Class No.	No. of Sample Images
	Non-Demented	0	1693
	Mild Demented	1	731
	Very Mild Demented	2	1833

3.2. Pre-processing

For better Alzheimer's Disease (AD) classification, a number of image pre-processing strategies were applied in this research. Pre-processing provides valuable strategies in medical imaging analysis by improving the data quality, reducing the background noisiness levels and allowing the model to work effectively on its input. In this study, the

pre-processing techniques deliberately targeted the enhancement of brain MRI images to ensure appropriate extraction and classification of features. It is common to hear complaints of noise and low contrast as well as of various disturbances in medical images, all of which can impair the effectiveness of machine learning and deep learning models. In order to combat several concerns above, a number of pre-processing approaches was implemented. One of the foremost strategies employed was Contrast Limited Adaptive Histogram Equalization (CLAHE), which enhances image contrast through the manipulation of the intensity distribution. It specifically helps in underlining the smaller differences between healthy and unsound brain tissues; thus, making them easier to analyze since they will be more salient after processing. In addition to the contrast enhancement, several other image filters for noise suppression in the images were applied to the pictures to remove noise and other unnecessary features. Not only did these filters augment the quality of the images, but they also lowered the computation requirements of the classification models by highlighting the important structures and patterns. Additionally, image data that was either missing or corrupted was dealt with systematically to maintain the consistency of the dataset. There were also techniques developed for image quality assessment and reconstruction of missing data, paving the way for the models to be trained on a structured and reliable dataset. Employing these pre-processing procedures, the investigation determined that the quality of the MRI images was sufficient and there were no such artifacts, which may limit the performance of the model in the images. This infusion of pre-processing methods provides the necessary components for the optimal performance of the cascaded framework which combines machine learning and deep learning techniques for the effective detection of Alzheimer's Disease. These developments provide a platform for an effective diagnostic technique that incorporates the requirements of practical problems of medical imaging.

3.2.1. Image Filtering

Image filtering is one of the main pre-processing techniques in computer vision that plays a vital role in enhancing the quality and authenticity of images, used for image analysis tasks [23]. This is the case, especially in the field of medical imaging, where even the smallest of the details can be the deciding factor in the cases of detecting and classifying diseases. In the process of image filtering, noise is diminished and the important features are brought out so that images become sharper, more contrasted and clearer. Thus, images can be more suited for machine learning and deep learning models. In this work, several image filtering methods were used to increase the quality of brain MRI images. So, Gaussian filters, median filters and Contrast Limited Adaptive Histogram Equalization (CLAHE) were some of the methods used. Gaussian filtering is mainly used to get rid of random noise and preserve important edges in the image, making it smoother. Median filtering is indeed very good at removing impulse noise and at the same time keeping sharp transitions in images. Using CLAHE (Contrast Limited Adaptive Histogram Equalization), it is possible to heighten local contrast and thereby improve the detection in areas where the intensity differences are very low. Careful modification of the dataset through the use of these filtering methods was possible to show the important features while not fully distracting. In turn, this shows the improved model accuracy and performance. Image filtering is the first stage of the road from raw data to feature extraction in medical imaging applications. Among various image filtering techniques techniques are applied, including:

Average Filtering: Mean filtering is an image processing technique where a sliding window is an n x n kernel used to average out the images and remove pixel intensity change and noise. This is normally a square matrix (3 x 3, 5 x 5) which slides over the image; the value at the middle point of the window is replaced by the average intensity of the area. This process helps in cutting high-frequency noise which gives a smoother image but the edges may be smoothed [24]. However, for medical imaging and balancing the real pixel intensities, Gaussian or median filters are more appropriate if superior edge preservation is wanted. The equation of the average filtering is.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i + j] \tag{1}$$

Where x [ ] is the input signal, y [i] is the output signal and M is the number of points in the average.

Gaussian Filtering: Linear filtering is one of the most widely employed techniques in the field of digital image processing; of all types of filters, the most popular one is the Gaussian filter [25]. Using this technique is useful if blurring a picture or reducing the noise level of the picture is desired. The formula that corresponds to the Gaussian. The equation of this filtering technique is

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

Where G(x,y): Gaussian kernel value at position, σ : standard deviation, which controls the spread of the Gaussian curve and x,y: pixel coordinates relative to the center of the kernel.

Median Filtering: A popular method in digital image processing; median filtering is exceptionally suited for denoising, especially impulses such as salt-and-pepper noise [26]. While mean filtering applies the average replacement of a particular pixel by its neighbors, median filtering applies a middle neighbor value selection. This approach also maintains good edges; hence, the image quality is well maintained. The filter reduces noise figures but retains details of the image by dragging a window over the image, sorting pixel values within the window and taking the median of the pixel value. The essence of the proposed filter is that it works with any noise without assuming a prior noise distribution, providing a good practical application in such fields as medical imaging and satellite imagery. However, it has been determined that the application of median filtering may lead to the introduction of artifacts; especially at region boundaries where there is a rapid transition in intensity values and the effectiveness of the algorithm is determined by the window size used. The general equation that this filtering technique uses is:

$$I(x, y) = \text{median}\{I(x + i, y + j) \,|\, (x, y) \in W\} \tag{3}$$

Here, W is the set of coordinates in the k×k window centered at (x,y). This operation is repeated for all pixels in the image (excluding the boundary pixels, depending on the padding strategy).

Bilateral Filtering: The bilateral filter is one of the methods of image filtering which is non-linear; smooths the images while maintaining the edges of objects and suppresses noises. It just replaces the intensity of a pixel with the average of weights derived from the intensity of some other neighboring pixels. This filtering technique follows this formula to work properly

$$b(x, f(x)) = \int Rf(y)Gs(x - y)Gt(z - f(y))\, dy \int RGs(x - y)Gt(z - f(y))\, dy \tag{4}$$

Here, (b(x,z) is the bilateral stack and (fz) is the image, defined as fz(y)=Gt(z - f(y)).

Min Filtering: It is a kind of image enhancement technique in digital image processing where each pixel is replaced with the lowest pixel intensity from its surrounding area; often employed to keep edges sharp and reduce noise. The formula for min filtering is given below

$$I(x, y) = \min_{(i,j)\in W} I(x + i, y + j) \tag{5}$$

Here, I (x, y): The intensity value of the pixel at location (x, y) in the original image. I'(x,y): The intensity value of the pixel at location (x, y) in the filtered image. W: A window (or kernel) of size kXk centered around the pixel (x, y). min: The minimum intensity value within the window W.

Max Filtering: A technique of image processing wherein as many pixels as possible in a given region are replaced by the maximum value within the same region, often used in morphological operations and feature extraction in image processing analysis.

$$I(x, y) = \max_{(i,j)\in W} I(x + i, y + j) \tag{6}$$

Here, max is the maximum intensity value within the window W

3.3. Architecture of the proposed system

Figure 1 shows the architecture of the proposed system and Algorithm 1 explains the proposed method step by step. The pre-processing consists of the quality check of the image and the noise reduction of the images by using the Contrast Limited Adaptive Histogram Equalization (CLAHE) and the level adaptation of the contrast. The IQAC (Image Quality Assessment Cell) is the one that does the evaluation of the image quality. The IQAC and CLAHE ensure the quality of the image based on its sharpness and level of noise. The sharpness and noise index of the image were calculated according to the technique proposed by Yagi et al. [27]. The image quality was derived using a linear regression model. The coefficients of quality predictors were pre-calculated in a regression analysis where mean square errors (MSE) of the images were used. Therefore, a higher value of Q_i indicated the low quality of the image. If the quality value, Q_i of an image, is lower than a threshold Q_{ih}, then the proposed method utilized the image for further analysis. The threshold value was 10 for our study. After that, the IQAC-approved image

was transferred to sRGB space to normalize the color values. Finally, the image was resized to fit the input size of AI models. These pre-processing steps, especially the quality verification, are crucial to ensuring the practical use of the proposed system [25]. After the pre-processing, the image is processed using the pre-trained CNN model to extract features automatically. In this study, five different pre-trained models including ResNet50, DenseNet201, InceptionV3, MobileNetV2 and Xeption, had used for feature extraction. The DenseNet201 model was selected based on its performance and suitability for the proposed system, which is explained in the following section. Then, the BFO and LDA methods processed the extracted features to select the Optimized one for the classifier. Finally, the machine learning model utilized the BFO and LDA-selected features to predict the output class of the image as Non-demented, mild-demented and very-mild-demented. Later on, the best feature extractor and classifier for the proposed system selection is explained.

3.4. Algorithm For the Proposed Model

3.5. Feature Extraction Model

In this research, we proposed a broad approach to feature extraction and classification where the deep learning model is followed by the traditional machine learning model. Our feature extraction phase employed five popular including:

3.5.1. ***MobileNetV2***

It has been developed by the Google researchers [28]. The mobileNetV2 involves the following work process: First, it proposes a state-of-art network structure for convolutional neural networks that can be used in the embedded and mobile devices taking into consideration both accuracy and efficiency. The model goes through training on large data sets such as ImageNet so as to achieve feature learning and recognition in the input data set. Finally, the MobileNetV2 model is trained and optimised for real time application on the embedded and mobile devices in our model for the purpose of carrying out the task detection and image recognition.

3.5.2. ***DenseNet201***

Gao Huang and his team formulated DenseNet201 in 2017 [29], which is built to improve the feature propagation efficiency since each layer is connected to all the other subsequent layers. It removes the redundancy, minimizes the parameters and enhances the gradients flow which helps in training the deep networks. As observed in the analysis, the DenseNet201 provides high accuracy in classification and in segmentation of images with low computational complexity. The small size of this device is particularly suitable for low-power environments and for medical diagnostic imaging whereby power and speed play an important role.

3.5.3. ***InceptionV3***

InceptionV3 is built by Christian Szegedy and his team at Google in 2016 [30]. This extends prior architectures with factorized convolutions and label smoothing as well as auxiliary classifiers. These improvements allow the proposed method to extract multiple scales of features of objects through a single template bank and within reasonable computation time. InceptionV3 is used for image classification, especially when the scale is large such as ImageNet because it provides both high accuracy and a relatively light load.

3.5.4. ***Xception***

Xception, also known as Extreme Inception, was proposed by François Chollet in 2017 [31]. This model replaces traditional convection layers with depthwise separation convection layers, which significantly reduces the computational difficulty while increasing the accuracy. It partners spatial correlation with channel index in order to reduce dimensionality permitting feature selection. Xception is particularly beneficial for things such as image recognition and segmenting which make this more useful for high accuracy use.

3.5.5. ***ResNet50***

This is one of the models created by the ResNet family by Kaiming and his team [32]. The presented 50-layer model assists to solve the challenge, known as vanishing gradient, by implementing the skip connectivity that allows identity mapping between the layers. Therefore, ResNet50 has the highest accuracy in tasks like image classification and other tasks like object detection. Its scalability has therefore made it an important base for contemporary deep-learning architectures.

Algorithm 1: Algorithm for Alzheimer's Disease Detection

Initialize: P, B, C ; // Parameter sets for FE, BFO, and SVC
Input: I_{RGB} ;
Output: ψ ; // Predicted Dementia Class
while $I_{SRGB} \neq I_{NIL}$ **do**
    // Quality-Aware Conversion (QAC)
    $I_{\text{SRGB}} \leftarrow I_{\text{QAC}}(I_{\text{RGB}})$;
    // Resize Image
    $I_R \leftarrow \text{Resize}(I_{\text{SRGB}}, 224 \times 224)$;
    // Feature Extraction using DenseNet201
    $\text{FM}_i \leftarrow$ DenseNet201 (parameters P_i) ;
    $f_i \leftarrow \text{FM}_i(I_R)$;
    // BFO-Based Feature Optimization
    $\text{BFO}_i \leftarrow$ BFO (parameters B_i) ;
    $Sf_i \leftarrow \text{BFO}_i(f_i)$;
    // LDA for Feature Reduction
    $\text{LDA}_i \leftarrow$ Load LDA (parameters LDA_i) ;
    $R_{f_i} \leftarrow \text{LDA}_i(Sf_i)$;
    // Classification using Support Vector Classifier
    $\text{CM}_i \leftarrow$ SVC (parameters C_i) ;
    $y \leftarrow \text{CM}_i(R_{f_i})$;
    // Decision Mapping
    if $y == 0$ **then**
        $\psi \leftarrow$ Non-Demented ;
    else
        if $y == 0.5$ **then**
            $\psi \leftarrow$ Mild Demented ;
        else
            $\psi \leftarrow$ Very Mild Demented ;

Return: ψ ;

Function $I_{\text{QAC}}(I_{RGB})$;
Initialize: $\alpha, \beta, \gamma, Q_{ih}$;
Load QAC model with parameters (α, β, γ) ;
$Q_i = \alpha + \beta \times \text{Sharpness} + \gamma \times \text{Noise}$;
if $Q_i < Q_{ih}$ **then**
    return I_{NIL} ;
if $C_{linear} \leq 0.0031$ **then**
    $I_{\text{SRGB}} = 12.92 \times C_{\text{linear}}$;
else
    $I_{\text{SRGB}} = 1.0552 \times C_{\text{linear}}^{1/2.4}$;
return I_{SRGB} ;

Function $\text{BFO}(F_i)$;
Initialize: $a \leftarrow 2A - 1$ where $A = 1, 2, 3, \ldots$;
$P \leftarrow$ Input Image ;
$R_u \leftarrow$ Feature Vector ;
BestFilter $\leftarrow$ applyFilter(P) ;
$Q_a \leftarrow$ applyFilterToImage(P, BestFilter) ;
$A \leftarrow$ imageToArray(P, Q_a) ;
foreach A **do**
    Compute $R_a = \{V_0, V_1, \ldots, V_{14}\}$ using (P, Q_a) ;
$R_u \leftarrow R_a$;
return R_u ;

Function $\text{LDA}_i(Sf_i)$;
Load LDA model with parameters LDA_i ;
$R_{f_i} \leftarrow$ Apply LDA to Sf_i ;
return R_{f_i} ;

3.6. Machine Learning Classifier

*3.6.1. **K-Nearest Neighbor (KNN)***

KNN is the basic ML, used for categorizing the data. In order to identify the K points in the feature space that are the closest to a given observation, the algorithm defines a certain distance that has to be employed. In most cases, the Manhattan distance is the most appropriate distance that is often used because of its efficiency and simplicity. It is easy to understand and apply since KNN does not have to be trained explicitly to be able to make predictions. The value in KNN calculates the mean of these neighbors' goal values for regression problems and selects the most repeated label among them for the classification problems [33].

*3.6.2. **Decision Tree (DT)***

A popular supervised learning technique for classification and regression is the Decision Tree. It forms a structure like a tree, with each internal node assuming a split into sub-datasets depending on a feature value. A leaf node is associated with a class label or a prediction. At each node, the algorithm chooses the feature that results in maximum homogeneity of the sub-datasets via criteria such as information gain. This process is continued recursively until a stopping criterion is reached, including the maximum depth of the tree or reaching pure leaf nodes, given their high level of interpretability and the ability to visually represent their outputs [34].

*3.6.3. **Random Forest (RF)***

Random Forest is one of the modern machine learning technologies based on the ensemble learning approach, which in turn is built on decision trees. It is a procedure of employing some features absolutely at random and some samples for constructing each tree. This in turn makes the model less prone to being overtrained and less likely to overfit the training data. In the random forest, the prediction of the tree is a stage that involves the aggregation of all the individual trees to give the final outcome. This is normally done in one of two ways; averaging for regression problems and voting, for classification problems. Therefore, the resulting estimator from the Random Forest model is obtained. Random Forest has gained increasing recognition based on aspects such as accuracy, scalability and suitable execution of high dimensions and big data sets. This is, in fact, rather a general algorithm, that can be applied to a variety of machine learning tasks, including classification, regression and outliers' detection. As with the case of Random Forest the ensemble technique used allows for achieving promising high level of accurate predictions and good generality. Furthermore, it also offers the chance to interpretation and implementation compared to the other approaches of the natural language processing [35].

*3.6.4. **Support Vector Machine classifier (SVC)***

The primary goal of SVC algorithm is to build the ideal hyperplane which helps in data point classification and at the same time, the maximum space between them is built. To achieve this, SVM looks for an N-dimensional space to separate the data with the help of a hyperplane, where N corresponds to the number of features. Support vectors are chosen with SVM having deemed the extreme points from each of the classes. This method can be used to discretize an input space. It is noteworthy that with an SVM, both linear and non-linear problems can be effectively solved. SVM is preferred in many domains due to its high robustness, efficiency in high-dimensional problems and ability to solve high complexities and high-dimensional problems of data sets efficiently [36].

*3.6.5. **Extreme Gradient Boosting (XGB)***

One of the supervised learning algorithms that have gained popularity for its precision and efficiency in regression and classification tasks is Extreme Gradient Boosting. It is one of many advanced boosting methods that enhances the performance of weak learners mainly decision trees by minimizing a particular loss function through iterative fitting. However, XGBoost has unique features—an added regularized model that reduces the robustness of boosters and a novel spilt finding approach that attempts to find the splits that give the most significant growth that set it apart from other boosting methods. In addition, it has other innovative capabilities like parallel computed and out-of-core computing that allows it to operate on massive databases efficiently [37].

*3.6.6. **GaussianNB (GNB)***

GNB means Gaussian Naïve Bayes, which is a simple form of classification algorithm that employs the probability method. It depends on Bayes' theorem and provides the assumption of feature independence. GNB performs best in situations where classifiers are required to rank entities based on categories, which often involves the use of continuous-valued feature vectors. The model posits that the appearances of the features are normally distributed

and assigns a probability density function to each class, which is a Gaussian function. GNB maintains the mean and variance of each attribute for each class during training, which are then used for classification during the testing phase [38].

3.6.7. ***Logistic Regression (LR)***

Logistic Regression is one of the widely used algorithms used for binary classification. As the name suggests, it is a model belonging to the linear class of models. Finally, it calculates the possibility of the input data belonging to a class using the sigmoid function while the output values should range between 0 and 1. LR employs the Maximum Likelihood Estimation method of adjusting weights in a way that will produce the likelihood of the observed data. It then scales the weights of the features through the sigmoid function in order to predict the class using the decision threshold. Another reason why logistic regression is powerful and easy to interpret is that it is applied in linear separability. It is an easy-to-implement and robust approach that can be applied in various fields like finance, healthcare and marketing [39].

These classifiers were chosen to compare their performance effectively and thus determine the most suitable one for the proposed system.

3.7. Feature Optimization Techniques

These whole study were chosen because of their performance in different visual recognition benchmarks, providing different architectural features that contribute to extract important features from medical image data. For classification, we incorporated seven well-known machine learning algorithms: SVC, DT, RF, NB, XGB, KNN and LR is the list of feature selection approaches used in developing the model. To this end, we connected each CNN feature extractor to each of the machine learning classifiers to create cascaded networks to harness the benefits of both deep learning and traditional classification models; and also, accomplish a better feature representation and translation. This approach produced 35 different cascaded networks. In the same experiment, we also evaluated additional optimization and a feature reduction approach that might improve our results. In this study we utilize four different optimization technique including:

3.7.1. ***Bacterial Foraging Optimisation (BFO)***

Bacterial Foraging Optimization (BFO) is an optimization algorithm that is based on the foraging behavior of E. coli bacteria. Like the bacteria, BFO also moves from regions of low nutrient density towards high nutrient density and away from toxins. At the intra-population level, these include edge detection and chemotaxis (bacteria movement to optimize nutrient intake), swarming (collective behavior to converge on optimal solutions) and reproduction and elimination-dispersal (maintaining population diversity and avoiding local optima). In the model, we assume that the position update of a bacterium is

$$\mathbf{P}(t+1) = \mathbf{P}(t) + C \cdot \Delta \tag{7}$$

where C is the step size and Δ is the direction of the move. BFO is used to analyze and solve optimization problems within the machine learning and bioinformatics domains [40].

3.7.2. ***Principal Component Analysis (PCA)***

Principal Component Analysis (PCA) is a widely used technique, which reduces the dimensionality of the given dataset and saving as much variability as possible. It changes the data into a new system of indices designated as principal components in a rank order determined by their capacity to account for the variability in the data. The first being the maximum variation followed by the second with the next highest variation with a direction in space completely independent of the first. PCA is popular technique of feature extraction, filtering and dimensionality reduction for high density data. The principal components are derived by solving the eigenvalue equation of the covariance matrix C of the dataset:

$$\mathbf{Cw} = \lambda \mathbf{w} \tag{8}$$

Where, C: Covariance matrix of the data, λ: Eigenvalue representing the variance explained by the principal component, w: Eigenvector representing the direction of the principal component. By selecting a subset of principal components with the highest eigenvalues, PCA reduces the dataset's dimensionality while retaining its most significant features [41].

3.7.3. *Linear Discriminant Analysis (LDA)*

LDA is a supervised learning dimensionality reduction method, used to obtain features on which samples are most likely to be linearly separable. LDA achieves class distribution in this manner because it maps data on to a space, where the between class variance is maximized and within class variance is at a minimum. Unlike PCA, which is unsupervised and preserves overall variance, LDA uses class labels to focus on features that best distinguish the classes. As a real time technology, it proves beneficial to work in a classification algorithm which includes facial recognition and diagnosis [42].

$$\mathbf{w} = S_w^{-1}(\mu_1 - \mu_2) \tag{9}$$

Where, w: Direction of projection, S_w: Within-class scatter matrix, μ_1, μ_2: Difference in class means.

In this configuration, we set up two groups of seventy cascaded networks for a total of one hundred forty networks. More precisely, we constructed 70 without optimization and BFO-integrated networks, 70 LDA and PCA-improved networks. All of the these networks were trained and examined on an offline database in the ratio of 80 % and 20%. Where 80% was for training the model and 20% for testing the model. This split helped to evaluate model generalization and reliability on unseen data and then choose the best network configuration for the proposed system. In the cascaded structures, this systematic evaluation also facilitated an understanding of how each feature extractor, classifier and optimization technique contributes to the performance.

3.8. *Performance Evaluation Metrics*

To evaluate the model, this study employed four individual evaluation matrix for instance.
Accuracy offers an evaluation of how well the model makes prediction for every class in general.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{10}$$

The **precision** measures the number of positive predictions to the total of positive and false predictions and enables the evaluation of the model's capability of avoiding errors of the first kind.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

The ratio of true positives to the total of true positives and false negatives indicates **recall**, which measures the model's capacity to detect positive occurrences.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

Finally, The **F1 score** is defined as the harmonic mean of recall and accuracy; it is a percentage of accurate positive predictions.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

4. Results

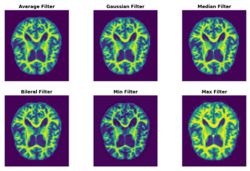
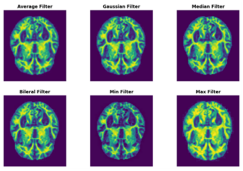
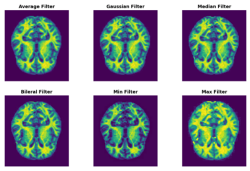
4.1. *Comparative Performance of Filtering Techniques Across Image Classes*

Among the six different filtering techniques mentioned in Table 2, Bilateral Filter was chosen as the best image-filtering scheme for further study. This decision is conducted by numerical results shown in the table above indicating that the Bilateral Filter provides higher image quality than the other techniques. In particular, it has the lowest Mean Squared Error (MSE) value and the highest Peak Signal-to-Noise Ratio (PSNR) in all the image classes within the Alzheimer's disease (AD) data set. These metrics demonstrate that it provides better noise attenuation while maintaining vital image features, which makes it ideal for pre-processing of medical imagery.

Images 1, 2 and 3 from table 2 illustrate the visual results obtained by applying these filtering techniques to three distinct categories of the AD dataset. Hence, the three groups will be referred to as Non-Demented, Mild-Demented and Very-Mild-Demented. The comparison shows that the Bilateral Filter is perfect in preserving the image quality and improving the contrast of the main details, which is necessary for subsequent deep learning analysis.

The filtering techniques tested are Average Filter, Gaussian Filter, Median Filter, Bilateral Filter, Min Filter and Max Filter. In comparison to other available methods like the Median Filter, moderate performance of other two methods was observed in terms of MSE and PSNR trade-off; however, Bilateral filter's remarkable effect of noise removal along with fine structural detail preservation makes it more effective and suitable for this particular task. Consequently, its application increases the reliability of the subsequent stages of the pipeline so that diagnostic accuracy and interpretability are improved in the context of early identification of Alzheimer's disease.

Table 3: MSE and PSNR of Filtering Techniques on Different Image Classes.

Filtered Sample Image	Class	Filtering Technique	MSE	PSNR
	Non-Demented	Average Filter	57.93	30.50
		Gaussian Filter	23.82	34.36
		Median Filter	17.76	35.63
		Bilateral Filter	**17.46**	**35.70**
		Min Filter	55.53	30.68
		Max Filter	57.14	30.56
	Mild-Demented	Average Filter	57.54	30.53
		Gaussian Filter	23.69	34.38
		Median Filter	18.28	35.50
		Bilateral Filter	**17.37**	**35.73**
		Min Filter	56.30	30.62
		Max Filter	57.99	30.49
	Very-Mild-Demented	Average Filter	57.38	30.54
		Gaussian Filter	27.02	33.81
		Median Filter	21.96	34.71
		Bilateral Filter	**20.97**	**34.91**
		Min Filter	56.45	30.61
		Max Filter	58.52	30.45

4.2. Feature, Parameter, MACS & FLOPS values of different CNN Architecture

Here, in Table 3 we compare the performance of five deep learning models (DenseNet201, MobileNetV2, ResNet50, InceptionV3 and Xception) based on the extracted features, number of trainable parameters and computational cost (MACS and FLOPS). The experimental result shows that feature optimization using BFO and PCA consistently outperforms the corresponding underdeveloped models and gives fewer feature numbers in most cases, with PCA trapping slightly more features. The best result regarding parameters, MACS and FLOPS was achieved by MobileNetV2, whose values are equal to 3,505 M; 327,487 M; and 654,973 M respectively, while the highest value of all belongs to InceptionV3 which is 5,749 M of MACS and 11,498 M of FLOPS. ResNet50 has the largest number of parameters: 25.557 M, which also reflects the greater complexity of the model. This comparison can help to decide on a model when certain characteristics of the problem require simplified feature size, minimal model complexity, or maximum computational efficiency.

Table 4: Feature, Parameter, MACS & FLOPS values of different models used in this study.

Model	Number of Features			Parameter	MACS	FLOPS
	Without Opt	BFO	PCA			
DenseNet201	1024	510	600	20.014 M	4.390 G	8.781 G
MobileNetV2	1024	511	600	3.505 M	327.487 M	654.973 M
ResNet50	2048	1014	800	25.557 M	4.134 G	8.267 G
InceptionV3	2048	994	800	23.835 M	5.749 G	11.498 G
Xception	2048	1020	800	22.855 M	4.146 G	8.292 G

4.3. Result Analysis of Alzeimer Disease (without Optimization and using BFO Optimizer)

On the other hand, table 4 compares the result of the combination of five pre-trained deep learning models (DenseNet201, InceptionV3, MobileNetV2, ResNet50, Xception) with seven classifiers (SVC, XGB, RF, DT, KNN,

LR and GNB) with accuracy, precision, recall and F1-Score. DenseNet201 with XGBoost (XGB) specifically reaches the highest accuracy of 94.71%, precision of 95.75% as well as the F1-score of 91.67%, which also demonstrates the model's effectiveness. As the table above shows, InceptionV3 is the best-performing architecture by having the KNN classifier yielding an accuracy of 89.49% and reasonable metrics thereafter. For KNN, MobileNetV2 and ResNet50 produce the highest accuracy as 86.34% and 85.04% and Xception gives the highest accuracy of 83.75%. The table illustrates how classifier selection can vastly affect a model's outcome and how frequently XGBoost and KNN excelled. Furthermore, Table 4 compares the performance of five pre-trained models (DenseNet201, InceptionV3, ResNet50, XeptionNet and MobileNetV2) with seven classifiers (SVC, XGB, RF, DT, KNN, LR and GNB) under two configurations, while in the second case it has been simulated with and without Bacterial Foraging Optimization (BFO). In so doing, DenseNet201 achieves the accuracy of 94.17% using the XGB model without further optimization; whereas, the accuracy is slightly lower when applying BFO: 92.42%. For all the classifiers, InceptionV3 and ResNet50 are slightly less accurate when using BFO but KNN and XGB retain their high result. The largest jumps for BFO are seen when: using XGB the accuracy increased from 78,92% to 89,00% and for KNN it increased from 85,04% to 89,12%. BFO is also employed in Mo-bileNetV2, the SVC accuracy rises from 61.48% to 73.73% and the GNB accuracy rises from 43.11% to 69.26%. When tested for all the model-classifier pairs, these results imply that BFO optimization yields positive impacts in some pairs while negative impacts in others.

Table 5: Comparison of results between Without Optimization (W.O) and BFO Optimization (BFO)

Pre-trained model	**Classifier**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
		W.O — BFO	**W.O — BFO**	**W.O — BFO**	**W.O — BFO**
DenseNet201	SVC	83.75 — 83.51	57.48 — 57.44	66.67 — 66.67	61.34 — 61.31
	XGB	**94.71 — 92.42**	**95.75 — 94.27**	**89.44 — 84.97**	**91.67 — 87.52**
	RF	89.05 — 88.57	92.52 — 93.01	77.68 — 76.89	79.44 — 78.40
	DT	85.20 — 84.00	78.78 — 77.50	77.89 — 77.25	78.27 — 77.37
	KNN	90.13 — 88.21	86.90 — 84.26	83.73 — 80.51	84.92 — 81.72
	LR	81.95 — 83.27	78.56 — 73.17	81.48 — 68.28	78.10 — 65.97
	GNB	84.84 — 80.63	78.02 — 75.14	72.87 — 76.60	73.21 — 75.34
InceptionV3	SVC	65.40 — 66.87	63.68 — 65.39	56.11 — 57.60	55.05 — 57.21
	XGB	88.51 — 88.88	87.58 — 89.03	87.69 — 86.41	87.64 — 87.50
	RF	82.15 — 84.60	83.99 — 86.66	77.45 — 79.68	79.39 — 81.77
	DT	74.82 — 76.41	72.72 — 74.42	72.76 — 73.28	72.69 — 73.77
	KNN	**89.49 — 89.10**	**89.43 — 89.45**	**88.20 — 88.83**	**88.74 — 89.10**
	LR	72.13 — 73.47	70.56 — 70.76	68.80 — 70.65	69.50 — 70.69
	GNB	61.49 — 57.46	67.24 — 64.36	63.97 — 60.37	59.28 — 55.41
ResNet50	SVC	76.44 — 60.90	81.52 — 75.10	80.80 — 50.50	80.83 — 49.25
	XGB	80.92 — 78.09	84.81 — 81.75	84.62 — 73.99	84.67 — 76.42
	RF	78.09 — 66.08	82.68 — 74.96	82.23 — 56.53	82.29 — 56.87
	DT	66.78 — 53.24	73.63 — 51.08	73.65 — 52.08	73.55 — 51.45
	KNN	**86.34 — 82.39**	**89.23 — 82.02**	**88.93 — 82.50**	**89.01 — 83.19**
	LR	74.91 — 83.39	79.94 — 84.02	79.90 — 82.50	79.91 — 83.19
	GNB	66.43 — 51.83	74.51 — 51.29	73.93 — 55.09	72.97 — 51.32
XceptionNet	SVC	63.37 — 66.63	77.06 — 64.31	51.03 — 57.53	48.81 — 57.22
	XGB	78.92 — 89.00	81.24 — 89.32	73.96 — 86.25	76.29 — 87.47
	RF	69.14 — 84.23	78.32 — 86.17	59.79 — 80.03	61.43 — 82.04
	DT	55.59 — 53.59	51.80 — 49.85	51.95 — 50.24	51.87 — 49.96
	KNN	**85.04 — 89.12**	**85.68 — 88.63**	**83.72 — 87.26**	**84.61 — 87.87**
	LR	70.79 — 72.49	69.03 — 69.49	68.86 — 69.20	68.94 — 69.34
	GNB	49.47 — 57.70	49.93 — 65.10	53.18 — 61.09	48.44 — 55.84
MobileNetV2	SVC	61.48 — 73.73	42.43 — 79.12	48.57 — 78.63	44.69 — 78.57
	XGB	76.21 — 78.09	79.21 — 82.35	73.15 — 82.22	75.34 — 82.16
	RF	66.43 — 74.32	75.69 — 79.45	57.17 — 79.17	58.62 — 79.16
	DT	55.12 — 67.49	52.21 — 73.80	52.56 — 73.75	52.37 — 73.74
	KNN	**83.75 — 83.22**	**83.23 — 86.09**	**82.90 — 86.79**	**83.06 — 85.85**
	LR	70.55 — 75.38	69.91 — 80.19	67.90 — 80.13	68.75 — 80.14
	GNB	43.11 — 69.26	52.53 — 75.60	51.85 — 75.47	43.31 — 75.24

4.4. Result Analysis of Alzheimer's Disease using PCA and LDA optimization

Table 5 compares the performance of five pre-trained models (DenseNet201, InceptionV3, ResNet50, XceptionNet and MobileNetV2) with seven classifiers (SVC, XGB, RF, DT, KNN, LR and GNB) using two feature reduction techniques: Among the popular techniques used in Intensity Transformation, two Methods can be named – PCA and LDA. The used performance metrics include the Accuracy, Precision, Recall, and F1 Score, all in percentage form. For all distributions, DenseNet201 excels with KNN using LDA of 98.32% and PCA of 97.47%, which are the most stable. When comparing the two optimization techniques, LDA produces the highest accuracy in four cases: DenseNet201 & KNN of 98.32%, combined model of DenseNet201 & XGB of 92.66%, ResNet50 & KNN of 89.63%, and XceptionNet & KNN of 89.63%. PCA achieves the highest accuracy in one case: the described architectures were MobileNetV2 with KNN (81.34%) and InceptionV3 with KNN (89.61%). The results presented above show that, in most cases, the LDA provides better classification accuracy than PCA, although PCA combined with SVC and RF classifiers is characterized by more stable performance.

Table 6: Comparison of results between LDA and PCA

Pre-trained model	**Classifier**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
		LDA — PCA	**LDA — PCA**	**LDA — PCA**	**LDA — PCA**
DenseNet201	SVC	85.92 — 85.92	90.46 — 90.46	72.48 — 72.48	71.90 — 71.90
	XGB	92.66 — 92.78	93.53 — 93.62	86.29 — 86.53	88.51 — 88.73
	RF	88.33 — 85.44	92.13 — 91.49	77.21 — 71.39	78.67 — 70.07
	DT	84.00 — 80.14	78.02 — 73.46	78.07 — 73.91	78.05 — 73.62
	KNN	98.32 — 97.47	98.38 — 97.41	96.95 — 95.60	97.61 — 96.42
	LR	91.34 — 91.58	87.94 — 88.25	88.80 — 89.13	88.33 — 88.66
	GNB	81.47 — 92.54	76.82 — 90.16	78.64 — 88.89	77.01 — 89.47
InceptionV3	SVC	85.70 — 85.45	84.87 — 84.67	82.85 — 82.65	83.70 — 83.51
	XGB	91.08 — 87.04	90.85 — 87.25	89.73 — 83.94	90.21 — 85.26
	RF	84.84 — 75.67	85.63 — 81.83	80.95 — 66.70	82.66 — 67.63
	DT	72.13 — 66.75	69.33 — 63.26	69.17 — 63.20	69.25 — 63.13
	KNN	89.12 — 89.61	88.06 — 88.23	88.29 — 88.67	88.13 — 88.42
	LR	87.04 — 84.23	86.54 — 83.64	85.92 — 83.29	86.21 — 83.44
	GNB	58.92 — 52.44	67.11 — 55.44	64.12 — 54.93	57.22 — 51.41
ResNet50	SVC	82.92 — 80.33	86.43 — 83.13	79.42 — 83.78	81.83 — 83.31
	XGB	79.27 — 78.92	82.51 — 82.80	75.89 — 82.75	78.15 — 82.74
	RF	68.55 — 74.79	77.99 — 79.59	58.43 — 79.31	58.99 — 79.37
	DT	55.95 — 64.43	52.28 — 71.13	52.51 — 71.11	52.38 — 71.12
	KNN	82.80 — 83.51	83.46 — 86.68	82.45 — 86.37	82.89 — 86.43
	LR	76.56 — 75.27	76.15 — 80.06	76.24 — 79.81	76.14 — 79.91
	GNB	46.53 — 65.14	48.60 — 66.79	50.96 — 68.99	45.72 — 65.91
XceptionNet	SVC	76.91 — 82.92	81.11 — 85.83	72.61 — 79.86	75.24 — 82.02
	XGB	74.56 — 73.26	76.01 — 76.56	69.19 — 68.20	71.20 — 70.55
	RF	65.61 — 63.02	76.86 — 75.48	56.27 — 51.15	57.13 — 48.38
	DT	52.53 — 47.70	48.34 — 43.54	48.63 — 43.24	48.44 — 43.35
	KNN	89.63 — 83.16	88.87 — 83.62	88.99 — 82.62	88.92 — 83.08
	LR	75.97 — 70.08	75.72 — 68.88	74.09 — 69.96	74.82 — 69.37
	GNB	40.87 — 51.24	52.00 — 50.52	49.74 — 53.55	40.95 — 50.27
MobileNetV2	SVC	80.68 — 77.15	83.44 — 80.63	84.06 — 72.88	83.60 — 75.34
	XGB	80.68 — 72.32	84.41 — 76.84	84.15 — 66.73	84.24 — 69.28
	RF	78.92 — 61.84	83.00 — 71.15	82.66 — 50.37	82.75 — 47.98
	DT	66.90 — 46.17	73.11 — 41.91	73.05 — 41.64	73.06 — 41.71
	KNN	84.22 — 81.39	87.20 — 82.02	87.00 — 79.52	87.04 — 80.58
	LR	76.44 — 68.32	80.95 — 66.99	81.04 — 67.77	80.97 — 67.35
	GNB	70.67 — 32.98	76.46 — 42.18	76.91 — 42.32	76.20 — 32.64

4.5. Performance of CNN Architectures with Time and Space Complexity

Table 6 shows the results including accuracy (A), precision (P) and recall (R) together with F1-score (F1), time complexity and space complexity of using deep learning models in combination with LDA and KNN. While

DenseNet201+LDA+KNN achieves the best result, with accuracy of 98.32%, precision of 98.38% and F1-score of 97.61%, the time complexity is highest at 309 ms. InceptionV3 with LDA and KNN is slightly lower than two previous strategies but remains at a high level of accuracy (89.12%) and F1-score (88.13%) and at the same time has the lower execution time 132 ms. MobileNetV2+LDA+KNN has the lowest space time complexity, while still achieving 84.22% accuracy and 87.04% F1-score. Xception+LDA+KNN achieved roughly 82%-83% and ResNet50+LDA+KNN is slightly slower while both have moderate results; yet, making more assumptions. DenseNet201 are more accurate on the test and InceptionV3 is faster.

Table 7: Best performance achieved by different CNN architectures with corresponding time and space complexity

Technique	A. (%)	P. (%)	R. (%)	F1. (%)	Time Complexity	Space Complexity
ResNet50+LDA+KNN	82.45	82.89	82.80	83.46	305 ms ± 54.4 ms	PM: 1960.76 MiB, INC: 791.12 MiB
Xception+LDA+KNN	82.80	83.46	82.45	82.89	177 ms ± 52 ms	PM: 1962.64 MiB, INC: 792.05 MiB
MobileNetV2+LDA+KNN	84.22	87.20	87.00	87.04	139 ms ± 5.84 ms	PM: 1582.88 MiB, INC: 602.19 MiB
InceptionV3+LDA+KNN	89.12	88.06	88.29	88.13	132 ms ± 10.2 ms	PM: 1789.20 MiB, INC: 705.63 MiB
DenseNet201+LDA+KNN	98.32	98.38	96.95	97.61	309 ms ± 128 ms	PM: 1493.54 MiB, INC: 557.73 MiB

4.6. Confusion Matrix of the proposed model

A confusion matrix is a performance measurement tool which uses actual and predicted outcomes to affirm a classification model's accuracy and yields True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). It helps calculate metrics like accuracy, precision, recall and F1-score to assess model performance. Also Figure 2 shows a normalized confusion matrix for the proposed model, assessing its performance in predicting three classes: Non-Demented, Mild Demented and Very Mild Demented. Classification is 100% correct for Non-Demented and almost 99% for Very Mild Demented with correct classification of 91% for Mild Demented. Most misclassifications are between Mild Demented and Very Mild Demented classes, 8.6% of samples from the experiment were misclassified from Mild Demented to Very Mild Demented and 0.57% from Very Mild Demented to Mild Demented. It also gives the viewer a V-shaped pattern where the performance is very impressive, especially at the per diagonals, suggesting high overall classification accuracy.

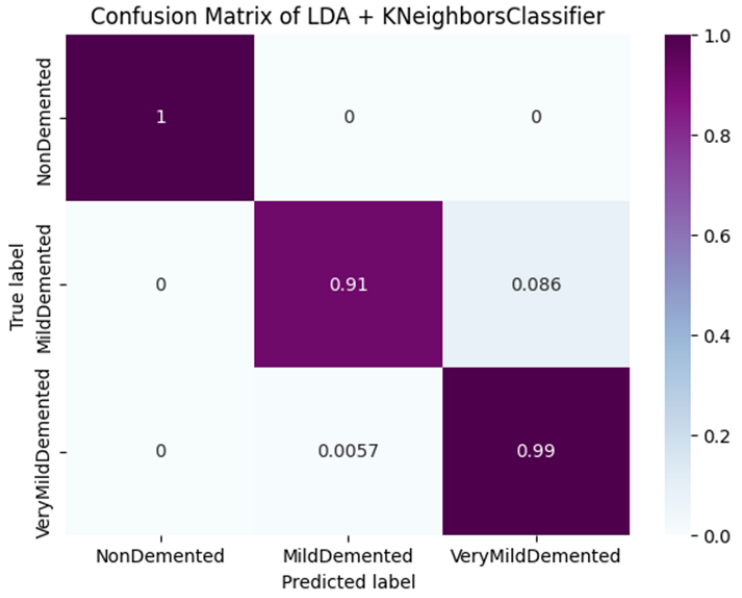


Figure 2: Confusion Matrix of the Proposed Model

4.7. Outcome Explanation with XAI

A comparison view of interpretability results of MRI scans as Non-Demented, Mild Demented and Very Mild Demented subjects is depicted in Table 8. The Actual Image, Grad-CAM++, and LIME visualizations of each class are shown to gain a complete insight into the capability of the red yellow) indicating higher activation region. Cooler colors(blue), however, correspond to regions with low contribution. The LIME maps visualize local super pixels that contribute most to the predicted class, highlighting interpretabilityof the model with respect to Artifacts relevance through interpretable boundaries. From the visual inspection,it can be seen that for Non-Demented subjects, the Grad-CAM++ activations are relatively diffuse and predominantly situated in non-critical regions, in comparison to Mild and Very-Mild Demented subjects where they localized more specifically to hippocampal and cortical areas experience early on-set Alzheimer's disease progression. Also, the LIME interpretations also overarchingly focus on the same anatomy regions and solidify the concordance between different XAI models. In summary, these visual explanations confirm that the suggested model not only reaches a very high level of accuracy in classification but also offers clear and clinically interpretable views of the neural areas that are involved in its diagnostic choices.

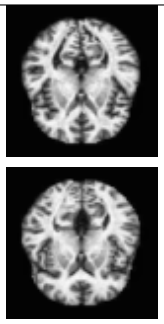
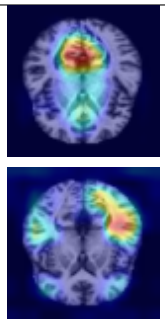
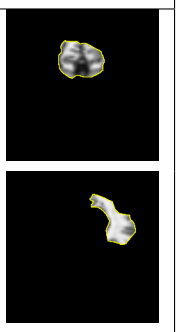
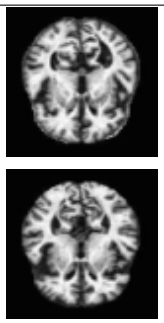
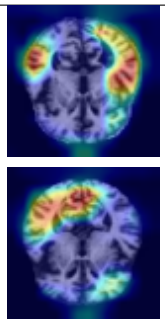
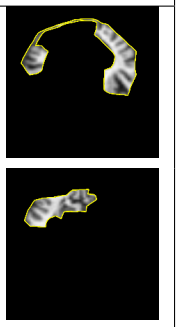
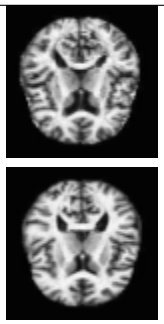
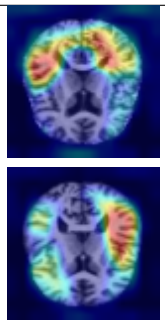
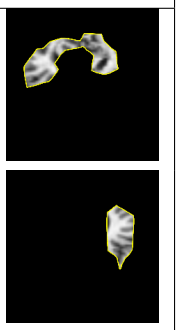
Sample Source	Actual Image	Grad-CAM++	LIME
Non-Demented			
Mild Demented			
Very Mild Demented			

Table 8: Visualization of Grad-CAM++ and LIME for different dementia classes.

4.8. Learning Curve of the proposed model

A learning curve is a curve that displays a model training and validation performance (e.g., accuracy or loss). It helps identify issues like overfitting or underfitting. Figure 2. shows the learning curve of the proposed model

(DaseNet201+LDA+KNN), which combines Linear Discriminant Analysis (LDA) and KNeighborsClassifier. Training accuracy (dashed black) remains high, near 1, consequently inferring high accuracy from the training pool while cross-validation accuracy initial curve (solid green) is low but gradually rises when the number of training instances increases. This shows the first signs of over-fitting, but in fact, the model learns better with increasing volumes of data. The shaping of the curves and the reduction of the area between the curves and the axis (variance) show better stability and generalization using a larger set. In general, the model looks promising: the quality of generalization grows rapidly as the amount of training data increases.

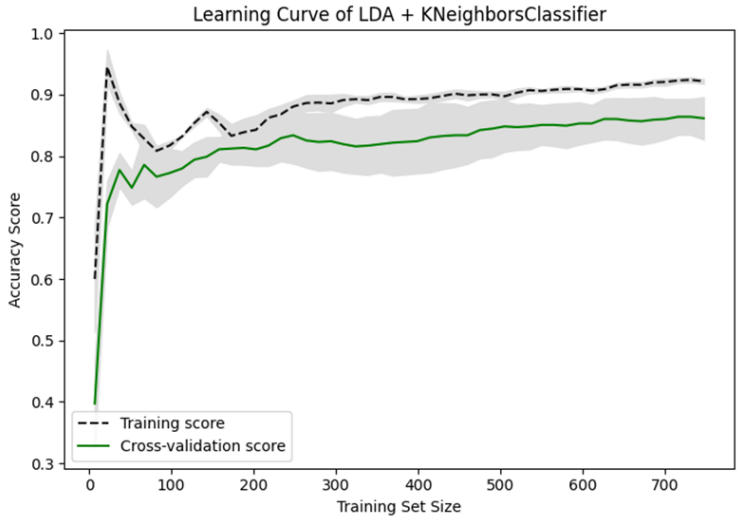


Figure 3: Learning Curve of the Proposed Model

4.9. ROC Curve of the proposed model

An ROC curve plots the True Positive Rate (TPR) vs. False Positive Rate (FPR) at different thresholds, exhibiting a model's ability to distinguish classes. Its Area Under the Curve (AUC) measures the overall performance. Figure 4 presents ROC curves for the classification model with the corresponding accuracy LDA and KNN depending on three classes. The x-axis and the y-axis signify False Positive Rate (FPR) and True Positive Rate (TPR), respectively. For class 0, class 1 and class 2, the color used is blue with AUC value of 1.00 (perfect classification), red with AUC value of 0.96 and green with AUC value of 0.98 respectively. The dotted black line represents chance level, area under curve (AUC) = 0.5. During classification, the proposed model shows promising results, especially for class 0 along with classes 1 and 2.

5. Discussion

A comparative analysis of the proposed method (DenseNet201 + LDA + KNN) against various other existing method for detecting AD is presented in Table 7 based on accuracy, optimization techniques and complexity. The proposed method achieves the highest accuracy of 98.32%, outperforming models such as Neuro-fuzzy (89.9%) by Allada et al. (2023), ResNet (90.0%) by Zhang et al. (2022) and CNN-based architectures (94.77%–96.22%) reported in studies by Hridhee et al. (2023), Spasov et al. (2022) and Gowhar et al. (2023). While other models employ optimization techniques like CSMVO, data augmentation, Adam optimizer and RMSProp; the proposed approach integrates advanced bio-inspired and dimensionality reduction methods, including Bacterial Foraging Optimization (BFO), Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to enhance performance. Furthermore, the proposed model presents a time complexity of about 309 ms ± 128 ms in addition to Peak memory (maximum amount of memory consumed during the model's execution) complexity of about 1493.54 MiB and incremental memory (memory usage during model execution, representing additional memory consumption beyond the primary memory) complexity of about 557.73 MiB. Such information is often omitted in existing works, where

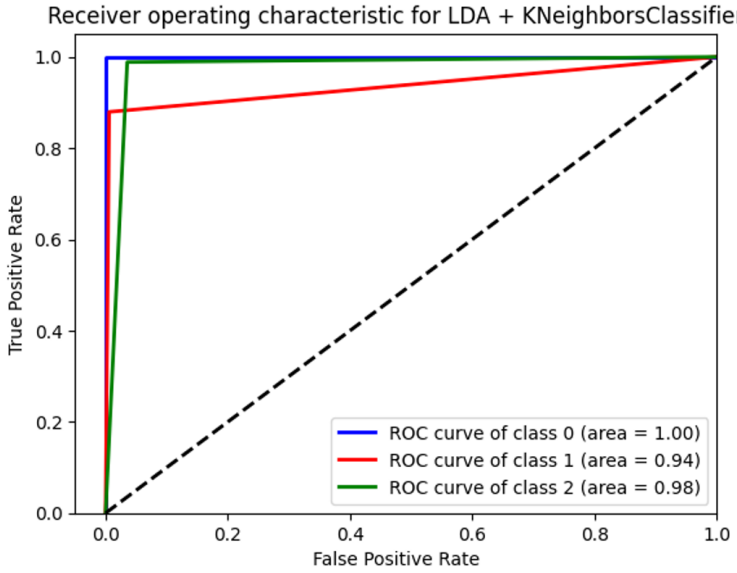


Figure 4: ROC Curve of the Proposed Model

complexity is generally unspecified. The results show that the proposed method not only provides better diagnostic classification accuracy, but it also analyzes the computational complexity aspect, giving the proposed method as highly reliable for diagnosis of Alzheimer's disease.

Table 9: Comparison of the proposed model with other existing models.

Ref.	Method	Accuracy	Optimization	Time	Space
10	Neuro-fuzzy	89.9%	CSMVO	X	X
15	ResNet	90.00%	Data augmentation	X	X
9	CNN	94.77%	X	X	X
16	CNN	95.89%	Adam Opti.	X	X
11	VGG16 & ResNet50	96.22%	RMSProp	X	X
Proposed method	**DenseNet201 + LDA + KNN**	98.32%	**BFO, PCA, & LDA**	**309ms ± 128ms**	**PM: 1493.54 MiB, INC: 557.73 MiB**

6. Conclusion

This work aims to fill the literature gap by developing an effective and reliable method for the diagnosis of Alzheimer's disease, a rapidly emerging neuro-degenerative disease. Through this research work, a new diagnostic model that incorporates the use of ML and DL methods is suggested to enhance the efficacy of the AD screening process. The proposed method mainly comprises of a cascaded network that consists of pre-trained CNNs for extracting deep features, BFO for feature selection, LDA and PCA for feature reduction; and finally, several conventional ML classifiers to detect the AD. This approach combines the advantages of the ML and the DL providing the balanced solution, which does not need heavy computational resources and large amounts of data that are limitations of the DL methods. The lengthy testing was carried out on a total of 4257 Brain MRI images including Non-Demented, Mild-Demented and Very Mildly Demented categories. Among different configurations, DenseNet201-LDA-KNN has higher measures of performance such as accuracy of about 98.32%, precision of 98.38%, recall of 96.95 and an F1-score of 97.61%. These results tell us that the proposed model could diagnose different stages of AD with high accuracy; it also point to the benefits of integrating DL and LDA methods with ML classifiers for feature extraction in medical imaging. It provides a feasible solution of early AD diagnosis with higher efficiencies that may be greatly benefiting the society and increasing patients' quality of life as well as reducing the healthcare burden. The study could be extended in the future by using a larger data set or data set from a variety of sources, comparing the performance of other optimization algorithms and testing the model on real patient data. Lastly, it will be useful to extend

the model to other neuro-degenerative disorders to showcase its potential applications in medical diagnosis.

References

[1] H. W. Querfurth, F. M. LaFerla, Alzheimer's disease, New England Journal of Medicine 362 (4) (2010) 329–344.
[2] K. Blennow, M. J. de Leon, H. Zetterberg, Alzheimer's disease, The Lancet 368 (9533) (2006) 387–403.
[3] J. S. Duncan, N. Ayache, Medical image analysis: Progress over two decades and the challenges ahead, IEEE transactions on pattern analysis and machine intelligence 22 (1) (2000) 85–106.
[4] J. Ker, L. Wang, J. Rao, T. Lim, Deep learning applications in medical image analysis, Ieee Access 6 (2017) 9375–9389.
[5] P. Scheltens, K. Blennow, M. M. Breteler, B. De Strooper, G. B. Frisoni, S. Salloway, W. M. Van der Flier, Alzheimer's disease, The Lancet 388 (10043) (2016) 505–517.
[6] R. Ibrahim, R. Ghnemat, Q. Abu Al-Haija, Improving alzheimer's disease and brain tumor detection using deep learning with particle swarm optimization, AI 4 (3) (2023) 551–573.
[7] G. P. Shukla, S. Kumar, S. K. Pandey, R. Agarwal, N. Varshney, A. Kumar, Diagnosis and detection of alzheimer's disease using learning algorithm, Big Data Mining and Analytics 6 (4) (2023) 504–512.
[8] A. Ebrahimi, S. Luo, R. Chiong, A. D. N. Initiative, et al., Deep sequence modelling for alzheimer's disease detection using mri, Computers in Biology and Medicine 134 (2021) 104537.
[9] W. Kang, L. Lin, B. Zhang, X. Shen, S. Wu, A. D. N. Initiative, et al., Multi-model and multi-slice ensemble learning architecture based on 2d convolutional neural networks for alzheimer's disease diagnosis, Computers in Biology and Medicine 136 (2021) 104678.
[10] R. A. Hridhee, B. Bhowmik, Q. D. Hossain, Alzheimer's disease classification from 2d mri brain scans using convolutional neural networks, in: 2023 International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE, 2023, pp. 1–6.
[11] A. Allada, R. Bhavani, K. Chaduvula, R. Priya, Alzheimer's disease classification using competitive swarm multi-verse optimizer-based deep neuro-fuzzy network, Concurrency and Computation: Practice and Experience 35 (21) (2023) e7696.
[12] G. Mohi ud din dar, A. Bhagat, S. I. Ansarullah, M. T. B. Othman, Y. Hamid, H. K. Alkahtani, I. Ullah, H. Hamam, A novel framework for classification of different alzheimer's disease stages using cnn model, Electronics 12 (2) (2023) 469.
[13] A. B. Tufail, N. Anwar, M. T. B. Othman, I. Ullah, R. A. Khan, Y.-K. Ma, D. Adhikari, A. U. Rehman, M. Shafiq, H. Hamam, Early-stage alzheimer's disease categorization using pet neuroimaging modality and convolutional neural networks in the 2d and 3d domains, Sensors 22 (12) (2022) 4609.
[14] S. Alinsaif, J. Lang, A. D. N. Initiative, et al., 3d shearlet-based descriptors combined with deep features for the classification of alzheimer's disease based on mri data, Computers in Biology and Medicine 138 (2021) 104879.
[15] N. Mahendran, D. R. V. PM, A deep learning framework with an embedded-based feature selection approach for the early detection of the alzheimer's disease, Computers in Biology and Medicine 141 (2022) 105056.
[16] F. Zhang, Z. Li, B. Zhang, H. Du, B. Wang, X. Zhang, Multi-modal deep learning model for auxiliary diagnosis of alzheimer's disease, Neurocomputing 361 (2019) 185–195.
[17] S. Spasov, L. Passamonti, A. Duggento, P. Lio, N. Toschi, A. D. N. Initiative, et al., A parameter-efficient deep learning approach to predict conversion from mild cognitive impairment to alzheimer's disease, Neuroimage 189 (2019) 276–287.
[18] J. Liu, M. Li, Y. Luo, S. Yang, W. Li, Y. Bi, Alzheimer's disease detection using depthwise separable convolutional neural networks, Computer Methods and Programs in Biomedicine 203 (2021) 106032.
[19] R. Janghel, Y. Rathore, Deep convolution neural network based system for early diagnosis of alzheimer's disease, Irbm 42 (4) (2021) 258–267.
[20] J. Bae, J. Stocks, A. Heywood, Y. Jung, L. Jenkins, V. Hill, A. Katsaggelos, K. Popuri, H. Rosen, M. F. Beg, et al., Transfer learning for predicting conversion from mild cognitive impairment to dementia of alzheimer's type based on a three-dimensional convolutional neural network, Neurobiology of aging 99 (2021) 53–64.
[21] A. M. Reza, Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement, Journal of VLSI signal processing systems for signal, image and video technology 38 (2004) 35–44.
[22] uraninjo, Augmented alzheimer mri dataset (2022).
URL https://www.kaggle.com/datasets/uraninjo/augmented-alzheimer-mri-dataset
[23] Y. AbdulAzeem, W. M. Bahgat, M. Badawy, A cnn based framework for classification of alzheimer's disease, Neural Computing and Applications 33 (16) (2021) 10415–10428.
[24] J. C. Church, Y. Chen, S. V. Rice, A spatial median filter for noise removal in digital images, in: IEEE SoutheastCon 2008, IEEE, 2008, pp. 618–623.
[25] N. Wiener, Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications, The MIT press, 1949.
[26] D. Marr, E. Hildreth, Theory of edge detection, Proceedings of the Royal Society of London. Series B. Biological Sciences 207 (1167) (1980) 187–217.
[27] N. Hashimoto, P. A. Bautista, M. Yamaguchi, N. Ohyama, Y. Yagi, Referenceless image quality evaluation for whole slide imaging, Journal of pathology informatics 3 (1) (2012) 9.
[28] K. Dong, C. Zhou, Y. Ruan, Y. Li, Mobilenetv2 model for image classification, in: 2020 2nd International Conference on Information Technology and Computer Application (ITCA), IEEE, 2020, pp. 476–480.
[29] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269. doi:10.1109/CVPR.2017.243.
[30] X. Xia, C. Xu, B. Nan, Inception-v3 for flower classification, in: 2017 2nd international conference on image, vision and computing (ICIVC), IEEE, 2017, pp. 783–787.
[31] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.
[32] B. Koonce, Resnet 50, in: Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization, Springer, 2021, pp. 63–72.
[33] M. Steinbach, P.-N. Tan, knn: k-nearest neighbors, in: The top ten algorithms in data mining, Chapman and Hall/CRC, 2009, pp. 165–176.

[34] B. De Ville, Decision trees, Wiley Interdisciplinary Reviews: Computational Statistics 5 (6) (2013) 448–455.
[35] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
[36] H. Xue, Q. Yang, S. Chen, Svm: Support vector machines, in: The top ten algorithms in data mining, Chapman and Hall/CRC, 2009, pp. 51–74.
[37] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, et al., Xgboost: extreme gradient boosting, R package version 0.4-2 1 (4) (2015) 1–4.
[38] C.-H. Lee, A gradient approach for value weighted classification learning in naive bayes, Knowledge-Based Systems 85 (2015) 71–79.
[39] T. G. Nick, K. M. Campbell, Logistic regression, Topics in biostatistics (2007) 273–301.
[40] K. M. Passino, Bacterial foraging optimization, in: Innovations and developments of swarm intelligence applications, IGI Global Scientific Publishing, 2012, pp. 219–234.
[41] H. Abdi, L. J. Williams, Principal component analysis, Wiley interdisciplinary reviews: computational statistics 2 (4) (2010) 433–459.
[42] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, L. Zhao, Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey, Multimedia tools and applications 78 (11) (2019) 15169–15211.