Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

A Hybrid Optimization Framework for Early Alzheimer's Disease Detection Using MDKM- Segmented MRI and DenseNet169

Hemavathi U.[1*], Durai S.[2]

Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India.

E-mail: [1*]uhemav@gmail.com, [2]durais@veltech.edu.in

Abstract

One of the common neurodegenerative disorders is called Alzheimer's disease (AD), which gradually reduces human memory, thinking ability, and cognition. Deep learning (DL) approaches play a vital role in early AD prediction because they do not require manual feature learning like traditional machine learning (ML) techniques. However, further improvement is still possible in terms of detection accuracy by reducing training time and minimizing overfitting. This research study develops a DL technique for early AD prediction and includes a pre-trained convolutional neural network (CNN) based feature extraction. The proposed technique consists of three phases: (1) preprocessing, (2) segmentation, and (3) classification. Initially, the system collects the input MRI images from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset. Then, contrast-limited adaptive histogram equalization (CLAHE) is employed to enhance the contrast of the gathered data from the dataset. The model then employs the Mahalanobis distance-based k-means (MDKM) algorithm to segment the skull portions from the input images. Finally, the system utilizes the osprey-optimized densely connected network 169 (O2DenseNet169) to classify AD, with hyperparameters optimally selected using the osprey optimization algorithm (OOA). The proposed technique achieves 99.44% accuracy, demonstrating the effectiveness of O2DenseNet169 in automatically learning biomarkers from images to classify AD.

Keywords: Alzheimer's Disease Prediction, Segmentation, ADNI Dataset, K-Means, and DenseNet169.

1. Introduction

The human brain consists of about 86 billion neurons that communicate and pass information between various parts of the brain. A neurodegenerative brain disorder called Alzheimer's disease (AD) primarily affects elderly people. This illness of unidentified etiology causes cognitive deterioration [1, 2]. Worldwide, about 45 million people suffer from this irreversible disease. It gradually weakens communication, memory, daily mobility, and speech [3]. It is the most prevalent form of dementia, appearing in approximately 60–80% of all dementia cases [4]. Therefore, it is essential to detect AD at an earlier stage to enhance the quality of the patient's life. However, diverse medical examinations are required to diagnose AD. Here, a significant amount of multivariate, heterogeneous data was employed to diagnose

*** Corresponding Author**

DOI: https://doi.org/10.36548/jiip.2026.2.002
Received: 14.02.2026, received in revised form: 15.03.2026, accepted: 31.03.2026, published: 08.04.2026

AD. In contrast, manual evaluation and data integration are time-consuming and complex [5]. In recent times, several neuroimaging modalities, including positron emission tomography (PET), magnetic resonance imaging (MRI), and single-photon emission computed tomography (SPECT), have been employed to diagnose AD. These modalities are considered powerful diagnostic tools for AD [6, 7]. Among these, MRI is well-suited for AD detection because it clearly shows the anatomical structure of the brain. Particularly, it helps to identify a key feature of AD, namely, brain atrophy. On the other hand, PET and SPECT can clearly show brain activities and functional changes, but their accuracy for detection is lower compared to MRI. Hence, it is essential to classify the MRI scans to assess brain health. However, developing a robust automated system for AD detection is challenging due to the following reasons: (1) complex anatomy, (2) expert interpretation requirements, and (3) variability in image quality [8]. To overcome these problems, artificial intelligence (AI)-based Deep Learning (DL) approaches are widely used to help radiologists in analyzing MRI data to enhance diagnostic accuracy [9].

In medical image analysis, an AI-based DL approach is widely used, achieving a superior rate in image classification [10]. Compared to Machine Learning (ML) models, DL models provide several benefits: (1) automatic extraction of complex features from low- to high-level images, (2) handling of large and diverse datasets, and (3) elimination of the need for manual feature engineering is not required. These benefits increase the prediction accuracy [11, 12]. Among all DL models, convolutional neural networks (CNNs) are significantly used in medical image diagnosis. They can automatically learn high-level features from 2D or 3D MRI images, reducing classification errors caused by handcrafted engineering. However, training requires higher computational power and a substantial amount of annotated medical data. In practice, medical image datasets are generally small, and collecting them for training is expensive. To address the limitations of CNN, transfer learning (TL)-based models are employed in medical image processing. These TL models utilize pre-trained models to extract general features. The extracted features are then fine-tuned for specific medical tasks. Hence, this TL model improves accuracy and reduces training time compared to a CNN on small medical datasets [13].

Many pre-trained approaches, including AlexNet, ResNet-50, VGG-16, ResNet-34, SqueezeNet, DenseNet, and InceptionV3, are commonly used for medical image diagnosis [14]. These models extract meaningful features from images, thereby providing high detection accuracy and faster training [15]. Among these, the DenseNet model provides significant benefits when compared to other models. It enables feature reuse by significantly reducing the number of parameters without compromising accuracy and avoids vanishing gradient issues by allowing information to flow freely across all layers. However, optimization of the hyperparameters is essential for ensuring effective performance of the DenseNet model. This optimization ensures feature reuse and propagation through the network, avoids the overfitting problem, balances training time, and improves accuracy and generalization on unseen data. Hence, this research study proposes a hyperparameter-tuned DenseNet169 model to classify AD based on a cluster segmentation approach. The major findings of the manuscript are organized as follows:

- The system introduces the MDKM clustering model to segment the skull regions from the preprocessed images. The advanced distance metric is used instead of Euclidean distance to avoid issues with local optima.

- The system develops the O2DenseNet169 model to extract features and classify abnormalities of AD, where OOA optimally selects hyperparameters, namely learning rate, activation, and dropout, to improve model performance.

The subsequent portions of the manuscript are constructed as follows: Section 2 surveys recent ML and DL models for AD prediction and their limitations, focusing on these aspects. Section 3 discusses the proposed methodology. Section 4 evaluates the performance of the proposed and existing models in terms of several performance metrics. Lastly, Section 5 presents the conclusion and future research directions.

2. Related Works

This section reviews the existing works related to AD detection using ML and DL methods. M. Sudharsan and G. Thailambal proposed an AD detection system utilizing principal component analysis and machine learning classifiers [16]. Initially, essential features from the images were selected using PCA and ML classifiers, such as regularized extreme learning machine (RELM), Import Vector Machine (IVM), and support vector machine (SVM), for AD classification. The SVM reached 79.03% accuracy on the ADNI dataset, which was higher than other ML models. An extreme gradient boosting (XGBoost) network was recommended by Fuliang Yi et al. [17] to classify AD. Initially, Boruta, information gain, and elastic net were employed for feature selection, followed by the use of XGBoost for classification. The system achieved 88.74% accuracy. Vijeeta Patil et al. [18] recommended two types of CNN: a three-dimensional CNN (3DCNN) and an 18-layer CNN for AD prediction. The 18-layer CNN achieved 98% accuracy, which was higher than that of the 3DCNN. Weichen Huang [19] recommended an automated AD prediction using multimodal contrastive learning. Initially, the model extracted the multimodal contrast learning features from the image and tabular data from the ADNI database. Then, the salient features were amplified and ranked using the tabular attention module. The technique attained 83.8% accuracy in classifying AD severity.

An early AD prediction using VGG-TSwinformer was recommended by Zhentao Hu et al. [20]. The system extracted low-level spatial features from the ADNI dataset using VGG-16. Then, using sliding window attention, it fused spatially adjusted and distant spatial features. Finally, the fused feature representation was evolved using temporal attention, and the model attained an accuracy of 77.2%. Qiankun Zuo et al. [21] suggested a deep fusion network for AD prediction. The model learned topological features using a cross-modal transformer generative adversarial network (CT-GAN) and created multimodal connectivity from ADNI. Furthermore, standard features are aligned, and the AD was classified by employing a swapping bi-attention approach. The system attained 90.24% accuracy. S. Venkatasubramanian et al. [22] suggested a pre-trained CNN for AD prediction. Hippocampus segmentation was done using multi-task deep learning (MTDL). Then, AD classification was performed using the ResNet model, where deer hunting optimization (DHO) was employed for CNN hyperparameter optimization, and the approach achieved an accuracy of 97.1%.

M. Rajesh Khanna et al. [23] presented a deep CNN and deep ensemble learning (DEL) model for multi-level AD classification. The DEL model attained a maximum accuracy of 94%, surpassing that of DCNN. Sheng Liu et al. [24] suggested a 3D CNN for early AD detection. Initially, data quality was improved by preprocessing before AD classification. The system attained an AUC of 85.12. F M Javed Mehedi Shamrat et al. [25] utilized several pre-trained models for AD classification. Initially, the images' contrast was enhanced, and then pre-trained

models, such as MobileNetV2, InceptionV3, ResNet50, VGG16, and AlexNet, were utilized for AD prediction. The InceptionV3 model outperformed all other models, achieving the best accuracy of 98.67%. Likewise, Purushottam Kumar Pandey et al. [26] suggested ResNet-50, ResNet-101, ResNet-152, DenseNet-201, EfficientNet-B0, and other models to learn subtle patterns related to AD. Compared to others, the ResNet-101 model achieved the best accuracy, i.e., 98.21%.

Modupe Odusami et al. [27] recommended an explainable DL architecture for diagnosing AD. Initially, the system identified the mask using the fuzzy c-means (FCM) algorithm for the white matter in the original images. Then, the 3-channel ResNet18 model. Finally, the system obtained 73.90% accuracy. Dayananda Pruthviraja et al. [28] suggested a cloud-based DL technique for AD detection. The system extracted the features and classified the AD using the Google model. The model yielded an accuracy of 98%. S. Suchitra et al. [29] suggested an EfficientNetB7 for AD detection, achieving an accuracy of 98.2%. Gia Minh Hoang et al. [30] recommended an AD prediction framework using vision transformers. First, the FLIRT function was used for skull stripping, and then classification was performed using the vision transformer. The model obtained an accuracy of 83.27%.

2.1 Problem Statement

The aforementioned existing works provide better and improved outcomes. However, several challenges still remain, which are as follows:

- The major drawback of existing solutions is their lack of focus on skull stripping, which is crucial because it is essential for cortical reconstruction and brain volume measurement. The presence of non-brain tissues may cause incorrect cortical surface reconstruction and brain volume estimation, resulting in low prediction accuracy.
- The works [18-30] utilize pre-trained DL models, which automatically learn features from the data. However, these models require a significantly larger number of training parameters to understand complex data patterns, leading to training difficulties and an increased risk of overfitting. Additionally, it is crucial to select the hyperparameters optimally to enhance performance. Unfortunately, existing models fail to focus on this aspect.
- Also, the existing approaches do not fully explore the importance of data augmentation (DA). They rely solely on limited and fixed data without applying any transformations to the dataset to enhance its diversity, which leads to poor generalization and overfitting.

This paper proposes an optimized pre-trained DenseNet169 with DA and advanced segmentation procedures to overcome such limitations. Initially, CLAHE is used for contrast enhancement to increase the visibility of essential features in MRI scans. Then, DA is performed to improve the dataset's diversity. The MDKM is employed to segment non-brain tissues from the brain, facilitating more accurate disease classification. Finally, O2DenseNet169 was utilized to extract complex, meaningful features from the segmented images for classifying AD as NC, AD, LMCI, and EMCI classes, with the hyperparameters optimally chosen via the OOA algorithm to provide optimal outcomes. The O2DenseNet169 enables feature reuse, thereby minimizing the number of parameters and reducing overfitting.

3. Proposed Methodology

The workflow diagram of the proposed method is illustrated in Figure 1. It comprises three main phases: (1) preprocessing, (2) segmentation, and (3) classification. Each phase is comprehensively explained in the following subsections.

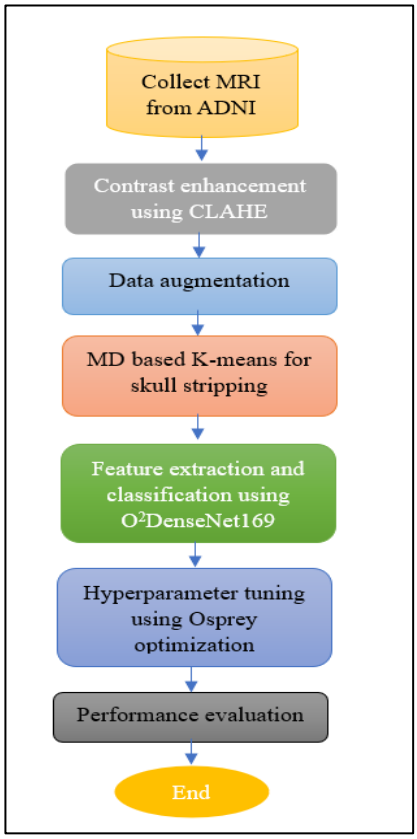


Figure 1. Proposed Technique's Workflow

3.1 Preprocessing

Initially, the input MRIs are collected from the publicly available ADNI database. Then, preprocessing is performed on the collected data to attain reliable classification outcomes. Here, the system performs image contrast enhancement to highlight specific tissues and structures within the brain. This process provides a more accurate prediction and characterization of AD, which is explained below.

3.1.1 Contrast Enhancement

The system uses the CLAHE algorithm to enhance the images' contrast. It is a variant of adaptive histogram equalization (AHE) that addresses the over-amplification issue inherent in AHE. It divides the image into smaller regions and individually improves the contrast of each region, while simultaneously limiting noise amplification. This results in a subtle region's improved visualization, which helps radiologists detect AD more accurately [25, 31]. The CLAHE's algorithmic procedures are explained as follows:

Step 1: Input the MRI $I\ (p, q)$ from the ADNI database, where $I\ (p, q)$ denotes the pixel intensities at spatial coordinates (p, q) of the MRI image.

Step 2: Partition each image into several 8x8 non-overlapping regions of size $R \times R$ each containing a neighborhood of 64 pixels. To analyze how many local regions are developed for contrast enhancement, the total number of tiles (N) for each image of size $X \times Y$ is described as follows:

$$N = \frac{X \times Y}{R \times R} \tag{1}$$

Where, $X \times Y$ indicates the height and width of the input image.

Step 3: Compute a histogram of pixel intensities for each region to estimate how frequently each intensity value is shown within a local region. The number of times the pixel intensity level pi occurs in the region r is denoted by $H_r(pi)$. The histogram for each region r is computed as follows

$$H_r(pi) = \sum_{p,q \in R} \alpha[pi - I_r(p,q)] \tag{2}$$

Where p, q indicates the spatial coordinates of a pixel within a region r, $I_r(p,q)$ symbolizes the pixel intensity value at position (p,q) in region r and $\alpha[pi - I_r(p,q)]$ denotes a counting function that adds 1 each time when the pi appears in a region r

Step 4: Set the clip limit for the histogram to avoid over-enhancement. Higher clip limits may over-enhance the contrast of the local image region, so they should be set to a minimum.

$$H_r^1(pi) = \min(H_r(pi), Cl) + \frac{\max(0, H_r(pi) - Cl)}{K} \tag{3}$$

Where Cl indicates the clip limit, K denotes the total number of intensity levels, $\max(0, H_r(pi) - Cl)$ indicate the maximum that was clipped from the histogram, and $H_r^1(pi)$ clipped histogram value after applying the clip limit.

Step 5: After clipping, the histogram values are normalized to maintain the total number of pixels within the region, which is mathematically expressed as.

$$H_r^2(pi) = \frac{H_r^1(pi)}{\sum_{pi} H_r^1(pi)} \tag{4}$$

Step 6: Compute the cumulative distribution function (CDF) from the normalized histogram to map the intensity distribution using Equation (5).

$$Cdf_r(pi) = \sum_{j=0}^{pi} H_r^2(j) \tag{5}$$

Step 7: Generate the new pixel intensity values (I_r^n) by mapping pixel intensities based on the CDF, which is given in Equation (6)

$$I_r^n(p,q) = rnd\big[Cdf_r\big(I_r(p,q)\big)\big].(Q-1) \tag{6}$$

Where rnd indicates the rounding function that converts the value into an integer, and Q denotes the total number of intensity levels.

Step 8: Merge the individual enhanced regions to generate a contrast-enhanced image, denoted as.

$$Ien\,(p,q) = I_r^n(p,q) \tag{7}$$

Where, Ien shows the final contrast-enhanced image. The CLAHE's hyper-space considered in experiments is presented in Table 1, which includes fixed parameters.

Table 1. Hyper-Space Used in CLAHE

Parameter	Value used	Type	Description
Region size	8 × 8	Fixed	It offers an effective local contrast enhancement without losing structural details
Clip limit	0.01	Fixed	Prevents over-amplification of noise in relatively homogeneous regions of the MRI image
Histogram bins	256	Fixed	Represents the distribution of pixel intensities within local MRI tiles

3.1.2 Data Augmentation

After contrast enhancement, DA minimizes overfitting and enhances generalization during neural network training. DA was applied after partitioning the database into training and testing sets, and after contrast enhancement. The method artificially expanded the dataset by making class-preserving changes, such as rotation, flipping, translation, etc., to the individual data without compromising the image's semantic meaning [32]. These techniques were applied only to the training dataset before model training to prevent data leakage. The augmentation procedures followed by the proposed system are listed below.

- **Rotation:** It uses random angles from 0 to 360 degrees to rotate an image, allowing the model to learn orientation-invariant features.
- **Flipping:** The images are horizontally flipped to learn features from different perspectives, improving data diversity by creating mirror images.
- **Translation:** It uses a vector to translate an image along an axis, preserving the relative positions between pixels. This prevents positional bias, where the features are slightly displaced in MRI because the brain's structures are not always perfectly aligned.
- **Resizing:** Applying resizing to an image ensures the uniformity of the dataset, where the images are resized to 224 × 224 to fit into a CNN model.
- **Normalization:** It scales each pixel value to the range of 0 to 1 to improve convergence and model performance.

Thus, applying the above DA procedures makes our model more robust for detecting AD, even when the MRI data contain variations.

3.2 Segmentation

After preprocessing, skull stripping or segmentation is performed for morphometric analysis and brain MRI studies. This procedure segments the cortex and cerebellum from non-brain and skull areas. The goal is to identify brain tissue associated with AD, namely gray matter, white matter, and cerebrospinal fluid, in the image for the learning process. An enhanced version of the k-means (KM) algorithm, named MDKM, was utilized for skull stripping, KM is a simple and computationally efficient unsupervised segmentation model compared to other complex clustering approaches because ADNI datasets generally do not contain pixel-level ground truth annotations for brain tissue extraction [33]. Hence,

unsupervised clustering methods offer a practical solution by grouping pixels with similar intensity characteristics without requiring manually labeled training data. However, in conventional KM clustering, the objective function minimizes the Euclidean distance between each data point and its corresponding cluster centroid. This is because the Euclidean distance is sensitive to outliers and less effective if the number of dimensions increases. It is also unsuitable for real-world applications because it treats each feature equally and linearly, whereas the MRI data contains a nonlinear relationship between the data patterns. In this work, the clustering objective is altered by replacing the Euclidean distance with the Mahalanobis distance, enabling the clustering process to consider the covariance structure of MRI intensity features. Furthermore, it improves separation between clusters by considering the data's full distribution, resulting in accurate segmentation. The steps of MDKM are as follows:

Step 1: Select the number of clusters k (here, 2 one is brain tissues, and the other is non-brain tissues) and randomly assign cluster centroids for the clusters.

Step 2: To calculate the similarity between a pixel and cluster centroid while considering the covariance among MRI intensity features, the Mahalanobis distance $\bar{D}_{MN}$ is computed using Equation (8).

$$D_{MN}(m_i,\ n_k) = \sqrt{(m_i - n_k)^T\ \Sigma^{-1}\ (m_i - n_k)} \tag{8}$$

Where, $D_{MN}(m_i,\ n_k)$ refers to the Mahalanobis distance between data points m_i and cluster centroid n_k, Σ^{-1} indicates the inverse covariance matrix of the data points, allowing clusters to form ellipsoidal boundaries rather than being restricted to spherical shapes and T denotes the transpose operator. A small constant value is included in the covariance matrix's diagonal element when necessary to enhance the stable computation of the inverse covariance matrix. This prevents singularity problems and stabilizes the Mahalanobis distance computation.

Step 3: Assign each data point to the nearest centroid in a cluster and recompute the centroid's new position using the equation below:

$$L_k = \frac{1}{Y_k}\sum_{m_i \in L_k} m_i \tag{9}$$

Where, L_k refers to the centroid of the cluster k, m_i denotes data points belonging to a cluster k, and Y_k indicates the number of data points in the cluster k.

Step 4: The clustering process continues until the grouping of pixels into the two clusters (brain tissues and non-brain tissues) becomes stable. This stability of the clustering process is computed using the following objective function (Equation 10).

$$\bar{D}_{AN}(m,\ n) = 1 - exp\ \left\{(-1)\ \left(\frac{log10(1+\beta C_E)}{C_E!}\right)^{1/2}\ \left(\frac{\|m-n\|^2}{\sigma^2}\right)\right. \tag{10}$$

Where, $\bar{D}_{AN}(m,\ n)$ indicate the similarity measure between data points, (m, n) denotes the data points of the image, C_E indicates the number of elements in the cluster, β denotes the constant parameter, and σ^2 denotes the variance of the data, respectively. The term $\|m - n\|^2$ indicates the squared Euclidean distance between data points. While the clustering results may

be influenced by the initial centroid selection in the existing KM algorithm, the use of Mahalanobis distance in the proposed MDKM method decreases this sensitivity by considering the covariance among MRI intensity features. According to the data variation, the distance measure adjusts the cluster boundaries by including feature variance, which helps the algorithm assign pixels more accurately even when the initial centroids are not optimal.

3.3 Classification

Finally, AD is classified using the O2DenseNet169 model initialized with weights pre-trained on the ImageNet dataset. The DenseNet169 is chosen due to its deep network structure (169 layers) and lower network parameters, which enable generalization and facilitate training on smaller datasets, such as ADNI [34]. Its higher effectiveness compared to other techniques, namely VGG16 and ResNet50, makes it an excellent choice for AD classification. During training, the initial convolutional layers of DenseNet169 were frozen to retain the pre-trained ImageNet feature representations and capture local features, while the higher dense blocks and the last classification layer were fine-tuned using the ADNI MRI dataset. This Transfer Learning strategy enriches training efficiency and classification reliability. The architecture of DenseNet169 is complex. Therefore, it is essential to select proper hyperparameters, namely batch size, learning rate, and number of layers. This research paper utilizes the OOA method to select the optimal hyperparameters. This algorithm effectively searches the complex hyperparameter space and creates a balance between exploration and exploitation enhancing model performance. The hyperparameter optimization improves training performance and provides accurate and reliable outcomes. The OOA-based DenseNet169 optimization is referred to as O2DenseNet169. Fig.2. shows the standard DenseNet169 structure.

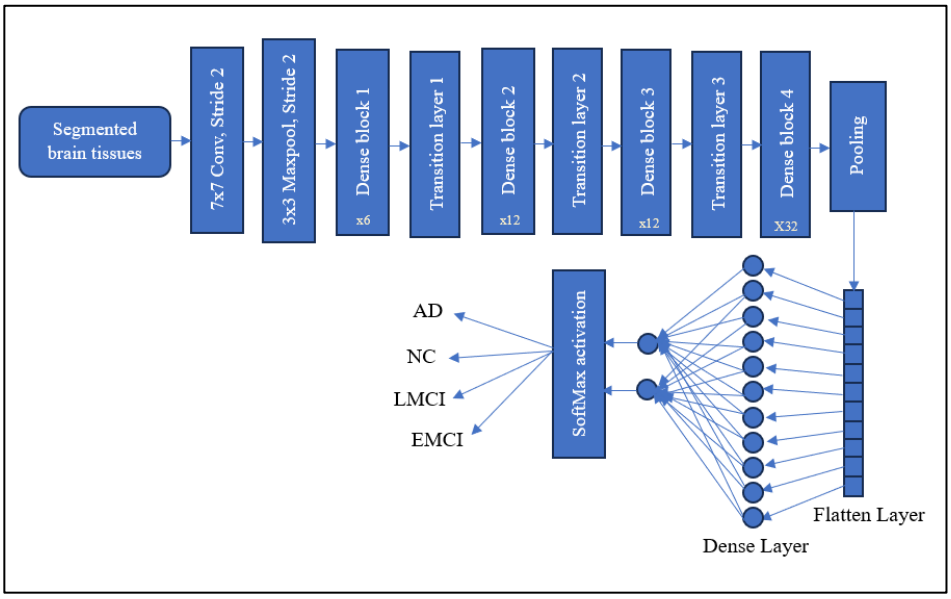


Figure 2. Architecture Diagram of DenseNet169

Figure 2 comprises 5 layers: a convolution, a pooling, a dense, a transition, and a classification layer, which are briefly discussed as follows:

- **Initial Convolution Layer:** Initially, the input MRI images were resized to 224 × 224 pixels and then input to the convolution layer, which is the core block of DenseNet169, This layer is where the majority of computation occurs, and is responsible for capturing feature maps from the input data using a filter or a kernel

(generally 3x3 or 5x5). The layer is vital for classification because the kernel selection determines which features the model focuses on, such as corners, textures, and edge. Before convolution, padding is applied, which adds extra pixels to the edge of the feature maps to avoid shrinking the feature map after applying several convolution layers. The non-linear relationship between the input and output is learned by applying a rectified linear unit (ReLU) activation function after each convolution layer. It returns the input if it is positive; otherwise, it returns 0. The function is computed as follows.

$$f\left(\hat{S}_I\right) = \max\left(0, \hat{S}_I\right) \tag{11}$$

Where, $\hat{S}_I$ refers to the input brain tissue segmented image.

- **Pooling Layer:** The features extracted from the convolutional layer are then passed to the max-pooling layer to decrease the feature map size. The layer applies a pooling filter over the feature map to aggregate the information within its region. The size of the feature map with $\left(k_h, k_w, k_c\right)$ dimensions can be reduced by applying the MaxPool technique as described in Equation (12) as follows:

$$\max\_ pool = \frac{k_c \times \left(k_h - s + 1\right) \times \left(k_w - s + 1\right)}{j^2} \tag{12}$$

Where, $h, w, and\ c$ denotes the height, width, and channel of the feature map, S denotes filter size, and S refers to the stride, respectively.

- **Dense Layer:** This layer comprises a set of neurons, where each layer is linked to the neurons of the previous layers, so-called dense layers, which enable feature reuse and improve information flow across the networks. The model contains four dense layers, each containing multiple convolution layers. Blocks 1 and 2 contain 6 and 12 convolution layers, while blocks 3 and 4 contain 32. All four blocks come with 32 output channels for each convolution layer they use. The filter sizes of the convolutions used in the dense blocks are generally 1x1 and 3x3.

- **Transition Layer:** A transition layer is placed after each dense block to reduce the feature maps' spatial dimensions. The network uses three transition layers after dense blocks 1, 2, 3, and 4. The layers use 1x1 convolution layers and 2x2 average pooling with a stride of 2 to lower the number of features.

- **Classification Layer:** This is the final layer, which employs the Softmax function to identify the type of AD. Softmax is generally used to classify DL models. It multiplies all the received values regardless of their nature and converts them into a total number between 0 and 1. It classifies the images into four classes: AD, NC, LMCI, and EMCI.

3.3.1 Hyperparameter Tuning

Before training, the hyperparameters, namely learning rate, batch size, dropout rate, and optimizer of the O²DenseNet169 model, need to be assigned optimally, which is done using OOA. It is a new optimization technique mimicking the osprey's behavior (hunting fish from

the sea) in nature. After the osprey detects its position, it hunts and eats the prey. The model's exploration and exploitation behavior are balanced based on the ospreys' natural behavior [35]. The model searches the hyperparameter space intelligently to get the optimal configurations to attain higher performance in prediction. As a result, training time and computational resources required for classification are reduced compared to manual tuning. The step-by-step working process of OOA is explained as follows:

Step 1: Population initialization

Equation (13) initiates or defines the population of ospreys (hyperparameters) within the search space. It indicates a set of possible values for each hyperparameter: learning rate, batch size, dropout rate, and optimizer.

$$\hat{M}_{a,b} = Rand \times \left(UB_b - LB_b\right) + LB_b, b = 1,2,.....d \tag{13}$$

Where, $\hat{M}_{a,b}$ denotes the a^{-th} individual position in b^{-th} dimension, $Rand$ indicates the arbitrary number between 0 and 1, UB_b and LB_b indicates the search space's upper and lower bounds, and d indicates the optimization problem's dimension.

Step 2: Fitness estimation

This function trains the model using a set of hyperparameters and outputs the performance metric. Estimate each individual's fitness $\left(\widetilde{F}\right)$ based on equation (14). The ospreys having minimal classification loss are considered the fittest ospreys of the current iteration. It is defined as follows:

$$\widetilde{F} = \frac{\overline{E}_{MI}}{T_{SP}} \times 100 \tag{14}$$

Where, $\overline{E}_{MI}$ represents the number of misclassified instances and T_{SP} represents the total number of instances, respectively.

Step 3: Exploration

The osprey randomly searches for a position to attack underwater fish. It evaluates the fitness function for a chosen set of random hyperparameters, known as exploration, where the relationship between the hyperparameters and model performance is learned. The osprey's new location has been identified based on the movement of the osprey toward the fish, as stated in equation (15).

$$\hat{M}_{a,b}^{P1} = \hat{M}_{a,b} + \hat{R}_1 \times \left(\widetilde{S}_{a,b} - \widetilde{G}_{a,b} \times \hat{M}_{a,b}\right), a = 1,2,.....N,\ b = 1,2,.....d \tag{15}$$

Where, $\widetilde{S}_{a,b}$ denotes the fish's position, chosen from the identified fishes, $\hat{R}_1$ indicates an arbitrary number between [0, 1], and $\widetilde{G}_{a,b}$ refers to an arbitrary integer between 1 and 2. If the new position's fitness value is better, the existing position is replaced by the new position, as defined in Equation (16).

$$\hat{M}_a = \begin{cases} \hat{M}_{a,b}^{P1}, & if\ \widetilde{F}_a^{P1} < \widetilde{F}_a \\ \hat{M}_a, & Otherwise \end{cases} \tag{16}$$

Where, $\widetilde{F}_a^{P1}$ refers to the newly constructed position's fitness.

Step 4: Exploitation

Once the exploration is completed, the model will look for an exploitation, where the hyperparameters likely to improve the model's performance are selected based on previous outcomes. i.e., the algorithm chooses a Suitable position after fish hunting, which makes the model carry out convergence and local search for the optimal solution. The mathematical form of the exploitation process is shown in Equation (17).

$$\hat{M}_{a,b}^{P2} = \hat{M}_{a,b} + \frac{LB_b + \hat{R}_2 \times \left(UB_b - LB_b\right)}{t}, \quad t = 1,2,\ldots MaxT \tag{17}$$

Where, $\hat{R}_2$ indicates an arbitrary value between [0, 1], t and $MaxT$ are the current and maximum iteration numbers, respectively. As in the previous phase, a new individual adopts a better solution by evaluating the individual's fitness, as defined in Equation (18).

$$\hat{M}_a = \begin{cases} \hat{M}_{a,b}^{P2}, & if\ \widetilde{F}_a^{P2} < \widetilde{F}_a \\ \hat{M}_a, & Otherwise \end{cases} \tag{18}$$

Where, $\widetilde{F}_a^{P2}$ refers to the fitness of the newly generated position.

Step 5: Based on performance, replace the existing solutions with a new one that maintains the best ones and avoids the worst ones.

Step 6: Until the stopping criteria are reached, continue steps 3 to 5 to attain the best possible solutions (hyperparameters) of the DenseNet169 for higher AD detection. The pseudocode of the proposed OOA is given in Fig. 3.

Input: Information of the optimization problems
Output: Optimal set of hyperparameters

Begin
Set the population size N
Set the current iteration t and maximum number of iteration $MaxT$
Initialize the osprey's position in the search space using Eqn (13)
Estimate each individual's fitness using Eqn (14)
For $t = 1$ *to* $MaxT$
For $a = 1$ *to* N
//Exploration
Compute the new position for the a^{-th} osprey using Eqn (15)
Check the boundary conditions and update a^{-th} osprey using Eqn (16)
//Exploitation
Compute the new position for the a^{-th} osprey using Eqn (17)
Check the boundary conditions and update a^{-th} osprey using Eqn (18)
End for
Save the best candidate solution found so far
End for
Output the best obtained solution
End

Figure 3. Pseudocode of the OOA

In addition, the OOA algorithm was implemented with a population size of 20 and a maximum of 30 iterations. Table 2 illustrates the search range used for hyperparameter optimization.

Table 2. Search Ranges Used in OOA

Hyperparameter	Search Range
Learning rate	0.00001 – 0.01
Batch size	16 – 64
Dropout rate	0.3 – 0.6
Optimizer	Adam and RMSprop

4. Results and Discussion

Here, the performance of several DL approaches is evaluated to prove the proposed technique's effectiveness for AD prediction. The implementation was performed using Python 3.7 software and the Google Collaboratory Pro platform, with the following configurations: Tesla K80 GPU, 12 GB of GDDR5 VRAM, and 16 GB of RAM. The libraries used include TensorFlow with Keras to build and train a DL model, SimpleITK to load the MRI images, Scikit-learn for skull stripping and clustering, NumPy and Pandas for data handling and manipulation, and Matplotlib for visualization. Table 3 shows the hyperparameter settings of the classifiers used in the evaluation. These hyperparameter values are selected based on the commonly used configurations in previous DenseNet studies and experimental trials. Furthermore, these ranges were selected to maintain stable network training while allowing the OOA algorithm to explore various hyperparameter combinations for enhanced AD classification performance. The model was trained using the Adam optimizer with a learning rate of 0.0001, a batch size of 32, and categorical cross-entropy loss for 50 epochs.

Table 3. Hyperparameters of the Proposed Model

Hyperparameters	Values
Learning rate	0.0001
Activation function	ReLU
Batch size	32
Epochs	50
Dropout	0.5
Optimizer	Adam
Loss function	Categorical cross entropy

4.1 Dataset Description

The system utilizes the publicly available ADNI repository [36]. The database was established by medical centers and universities in Canada and the United States to develop biomarker processes and standardized imaging approaches for AD. This dataset enables robust training and evaluation across early-to-advanced Alzheimer's stages, making it valuable for fine-grained classification tasks. It comprises multimodal neuroimaging data, including structural T1-weighted MRI scans, with labels derived from expert clinical assessments. The dataset contains four classes: Late Mild Cognitive Impairment (LMCI), AD, Normal Control (NC), and Early Mild Cognitive Impairment (EMCI). The repository consists of MRI scans from 300 subjects, with 75 subjects in each category (LMCI, AD, NC, and EMCI). The original MRI scans are provided as 3D volumes, from which multiple informative 2D slices were extracted to capture relevant brain structures for the analysis of AD. After preprocessing and data augmentation, the final dataset consists of 21,816 MRI slices distributed across the four classes. The images are 2D, and each image was resized to 224 × 224. Demographic details of the subjects are presented in Table 4.

Table 4. Dataset Distribution Across Classes

Class	Images
NC	6775
EMCI	5817
AD	5764
LMCI	3460
Total	21816

To avoid data leakage and ensure the authenticity and reproducibility of the experiments, the dataset was divided at the subject level. All MRI slices belonging to a particular subject were kept together and assigned either to the training set or the testing set. Approximately 70% of the subjects were employed for training, and the remaining 30% for testing. The sample images from the ADNI representing the four classes of AD are demonstrated in Fig. 4.

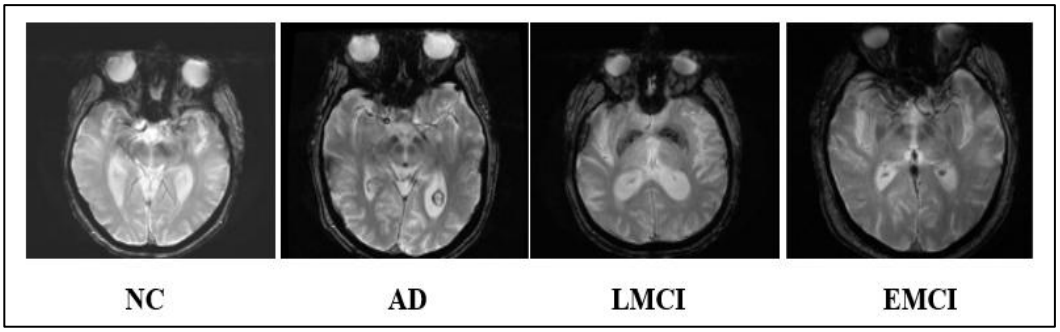


Figure 4. Sample Images from ADNI

4.2 Evaluation Metrics

We evaluate the segmentation and classification model used in our current methodology. The metrics used to assess the clustering performance include the Jaccard index (JSI) and Dice coefficient (DSC), and the metrics used to evaluate the classifier's performance consist of accuracy, precision, recall, and f1-score. These metrics are computed based on the four values of the confusion matrix: True Positive $(\hat{Q}_{TP})$, False Positive $(\hat{Q}_{FP})$, True Negative $(\hat{Q}_{TN})$, and False Negative $(\hat{Q}_{FN})$. The definitions are as follows:

- **True Positive:** Correctly predicted an image of the given class category.
- **False Positive:** Incorrectly recognizes a category when it belongs to another AD category.
- **True Negative:** Correctly predicted an image that does not belong to the given image category.
- **False Negative**: Incorrectly recognizes a category as not belonging to the target class when it belongs to that class.

These definitions are applicable to both segmentation and classification, but segmentation is measured at the pixel level instead of the image level. Since the ADNI dataset does not contain pixel-level ground truth masks for segmentation, the segmentation results given by the proposed MDKM method were compared with those obtained from existing segmentation methods such as KM, fuzzy c-means (FCM), region growing algorithm (RGA), and Otsu thresholding (OTH). The overlap between the segmented regions was analyzed based on DSC and JSI metrics. Equations (19)-(24) show the mathematical calculations of these metrics.

$$DSC = \frac{2\,\hat{Q}_{TP}}{\left(\hat{Q}_{TP} + \hat{Q}_{FP} + \hat{Q}_{FN}\right)} \tag{19}$$

$$JSI = \frac{\hat{Q}_{TP}}{\hat{Q}_{TP} + \hat{Q}_{FP} + \hat{Q}_{FN}} \tag{20}$$

$$Accuracy = \frac{\hat{Q}_{TP} + \hat{Q}_{TN}}{\hat{Q}_{TP} + \hat{Q}_{TN} + \hat{Q}_{FP} + \hat{Q}_{FN}} \tag{21}$$

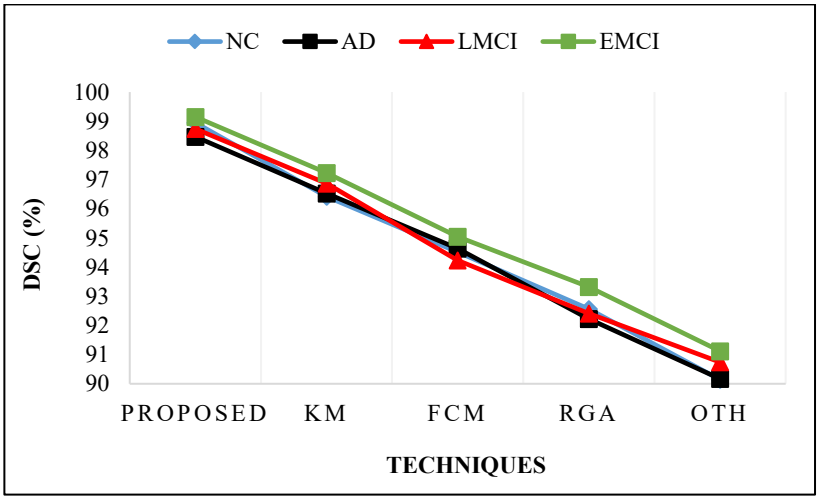
$$precision = \frac{\hat{Q}_{TP}}{\hat{Q}_{TP} + \hat{Q}_{FP}} \tag{22}$$

$$recall = \frac{\hat{Q}_{TP}}{\hat{Q}_{TP} + \hat{Q}_{FN}} \tag{23}$$

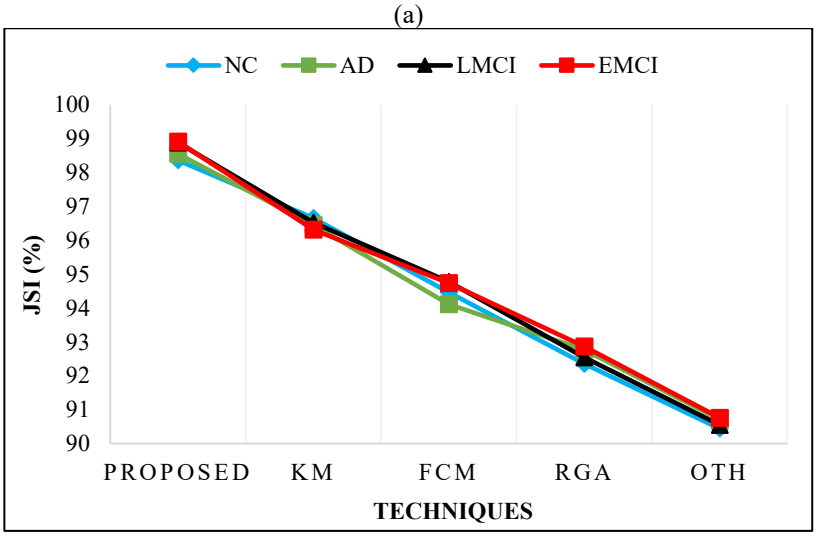
$$f1-score=2\times\frac{precision\times recall}{precision+recall} \tag{24}$$

4.3 Performance Analysis of Proposed Segmentation

This section provides the experimental outcomes of the proposed MDKM algorithm and the existing KM, CM, RGA, and OTH methods, regarding DSC and JSI, which are shown in the following figure 5(a) and 5(b).



(a)



(b)

Figure 5. Evaluation of the Clustering Models Based on DSC and JSI

The measures generally employed to evaluate the segmentation effectiveness of the model are DSC and JSI. The degree of overlap between the masks or bounding boxes is determined using JSI, and the similarity between the two masks is determined using DSC. Compared to existing techniques, the proposed approach reaches the best DSC and JSI of 98.95% and 98.36%, respectively, for the NC class. Similarly, for the AD, LMCI, and EMCI classes, the proposed model achieves the best DSC results of 98.47%, 98.74%, and 99.15%, respectively, and the best JSI of 98.55%, 98.88%, and 98.91%, respectively. The model captures feature correlations and handles complex data in a noisy environment effectively, making it suitable for skull stripping, where existing region-growing methods, such as FCM and Otsu, are prone to errors when the input data is noisy and complex. It also effectively

differentiates the several brain structures, including the skull, brain, and other tissues, making it suitable for complex MRI scans.

4.4 Evaluation of Classification Models

Here, the outcomes of the proposed O2DenseNet169 model are contrasted with the existing methods, namely, DenseNet121, ResNet50, VGG16, and GoogleNet, based on the above-mentioned classification metrics. All these baselines were evaluated using the same experimental settings, such as the same dataset split, preprocessing procedures, and DA strategy as the proposed O2DenseNet169 technique, which offers a fair comparison against the proposed technique.

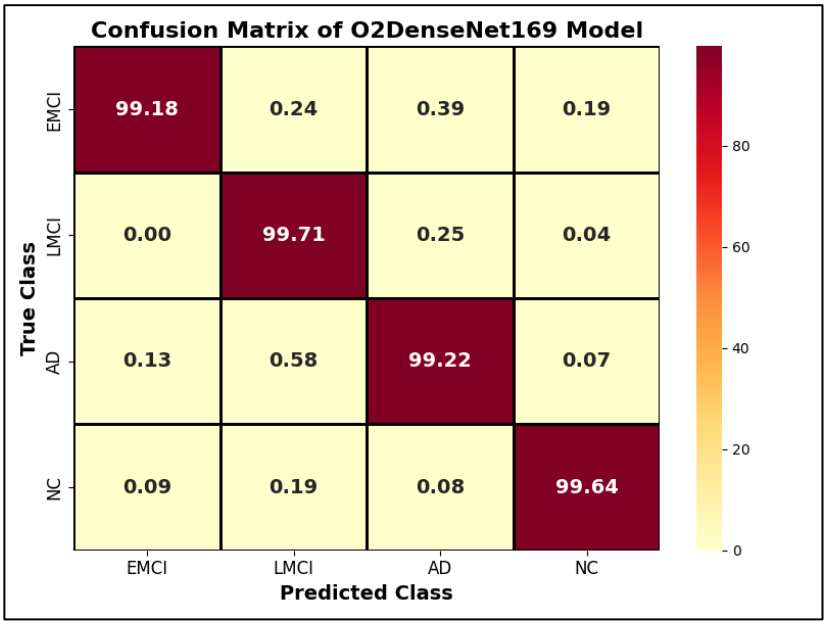
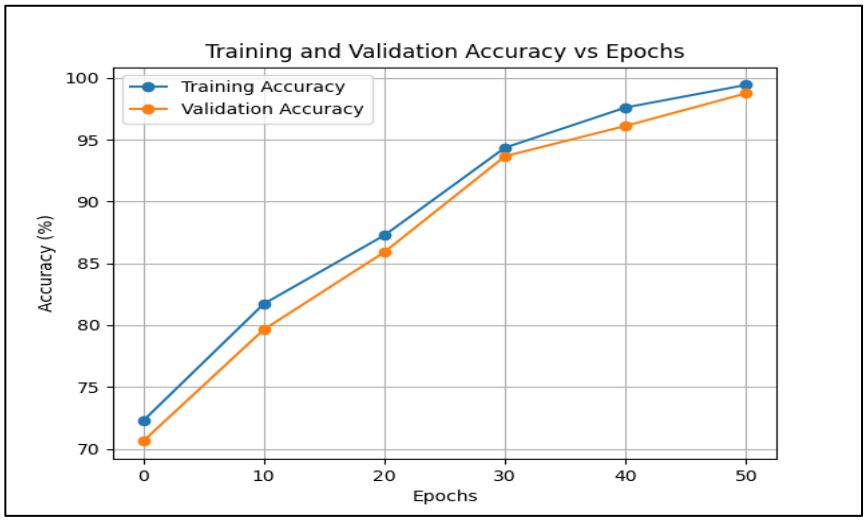


Figure 6. Confusion Matrix

Figure 6 depicts a confusion matrix of the proposed system for diagnosing AD with different classes. It was observed that our model achieves higher accuracy for all the classes of the database, such as 99.18% for EMCI, 99.71% for LMCI, 99.22% for AD, and 99.64% for NC, which demonstrates its superior learning performance for disease prediction.



(a)

(b)

Figure 7. Accuracy and Loss of the Proposed Model During Training and Testing

Figure 7 illustrates the accuracy and loss of an AD detection model during training and testing over 50 epochs. The training accuracy increases consistently, and the training loss consistently decreases, representing the model's ability to learn from the dataset. Also, the validation accuracy demonstrates a steady enhancement where the validation loss gradually declines, suggesting that the model generalizes well without significant overfitting. This performance enhancement is mainly due to the integration of OOA-based optimal hyperparameter selection of the O^2DenseNet169 model, which avoids overfitting and increases generalization ability by exploring the optimal configurations.

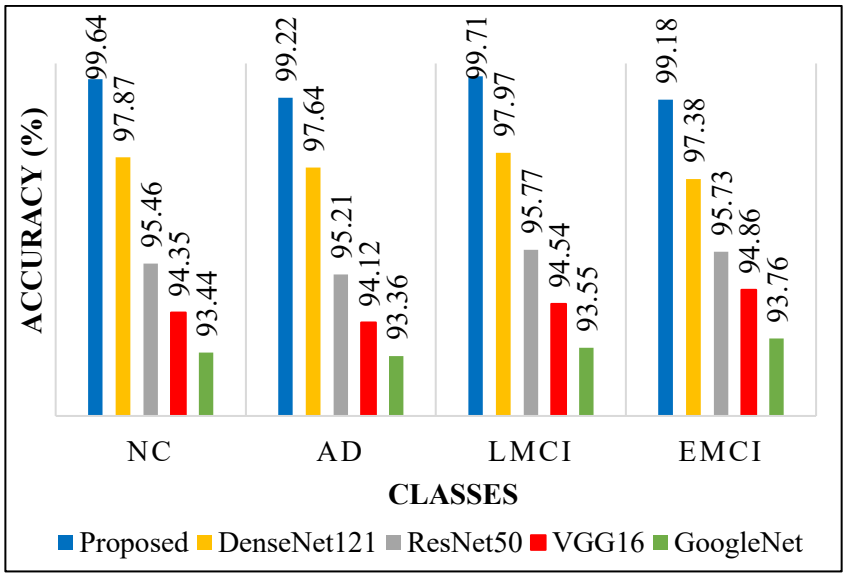


Figure 8. Accuracy Analysis

Figure 8 illustrates the outcomes of several classifiers regarding accuracy metrics. For the NC class, the existing methods, such as DenseNet-121, ResNet-50, VGG16, and GoogleNet, provide an accuracy of 97.87%, 95.46%, 94.35%, and 93.44%, respectively, which are less than the proposed technique, which reaches an accuracy of 99.64% for NC. Likewise, our model achieves 99.22% for AD, 99.71% for LMCI, and 99.18% for the EMCI class, which are better than the existing methods. Overall, the approach achieved a higher average accuracy of 99.44% compared to the existing models.

Table 5. (a) Precision, (b) Recall, and (a) F1-Score Analysis

(a)

Classes	Proposed	DenseNet121	ResNet50	VGG16	GoogleNet
NC	99.71	97.94	95.53	94.43	93.53
AD	99.62	97.68	95.29	94.47	93.37
LMCI	99.82	98.03	95.83	94.63	93.64
EMCI	99.23	97.44	95.81	94.97	93.87
Average	99.6	97.77	95.62	94.63	93.6

(b)

Classes	Proposed	DenseNet121	ResNet50	VGG16	GoogleNet
NC	99.52	97.76	95.38	94.26	93.35
AD	99.07	97.71	95.18	94.02	93.23
LMCI	99.65	97.88	95.69	94.44	93.48
EMCI	99.07	97.31	95.66	94.77	93.67
Average	99.33	97.67	95.48	94.37	93.43

(c)

Classes	Proposed	DenseNet121	ResNet50	VGG16	GoogleNet
NC	99.67	97.91	95.49	94.38	93.47
AD	99.25	97.67	95.24	94.16	93.39
LMCI	99.74	98.01	95.81	94.57	93.59
EMCI	99.22	97.41	95.76	94.89	93.79
Average	99.47	97.75	95.58	94.5	93.56

Table 5 presents the outcomes of the classifiers regarding (a) precision, (b) recall, and (c) F1-score. The proposed model yields precision rates of 99.71%, 99.62%, 99.82%, and 99.23% for the NC, AD, LMCI, and EMCI classes, respectively. The proposed model yields recall rates of 99.52%, 99.07%, 99.65%, and 99.07% for the same classes. Additionally, the proposed model attains F1-score rates of 99.67%, 99.25%, 99.74%, and 99.22% for the NC, AD, LMCI, and EMCI classes. Table 6 presents the inference time analysis of the proposed and baseline models. In this case, the proposed model takes less inference time, at 0.21 seconds per MRI, which is better than the baseline models. Thus, the overall performance analysis shows that the proposed technique achieves outstanding results compared to existing approaches. This is due to its use of an advanced deep learning structure to learn complex features from the MRI by improving image contrast using CLAHE, while augmentation helps enhance the model's generalization ability, leading to higher accuracy with better segmentation outcomes.

Table 6. Inference Time Analysis of the Proposed Model

Model	Inference Time (seconds per MRI)
Proposed model	0.21
DenseNet121	0.27
ResNet50	0.32
VGG16	0.36
GoogleNet	0.31

Figure 9 illustrates the real-time inference performance of the proposed technique, where the processing time remains consistent at around 0.21 sec with minor variations. This demonstaretes the model’s effectiveness and suitability for real-time applications.

Figure 9. Real Time Inference Performance of the Proposed Model

4.5 Comparative Analysis

Here, the proposed model's comparison outcomes (Table 7) with the existing literature works using the ADNI database are discussed in terms of accuracy.

Table 7. Comparative Analysis

Author's Name & Ref. No	Techniques Used	Accuracy (%)
Ours	O^2DenseNet169	99.44
M. Sudharsan and G. Thailambal [16]	SVM, RELM, and IVM	79.03
Fuliang Yi et al. [17]	XGBoost	88.74
Vijeeta Patil et al. [18]	3DCNN	98
Weichen Huang [19]	Multimodal contrastive learning	83.8
Zhentao Hu et al. [20]	VGG-TSwinformer	77.2
Qiankun Zuo et al. [21]	CT-GAN	90.24
S. Venkatasubramanian et al. [22]	MTDL	97.1
M. Rajesh Khanna et al. [23]	DCNN	94
Sheng Liu et al. [24]	3DCNN	85.12
F M Javed Mehedi Shamrat et al. [25]	InceptionV3	98.67
Purushottam Kumar Pandey et al. [26]	ResNet-101	98.21
Modupe Odusami et al. [27]	ResNet18	73.90
Dayananda Pruthviraja et al. [28]	GoogLeNet	98
S. Suchitra et al. [29]	EfficientNetB7	98.2
Gia Minh Hoang et al. [30]	Vision transformer	83.27

Our model provides an accuracy of 99.44%, which is comparatively higher than existing works. Most existing works use effective pre-trained CNN models for classification, which offer good predictive accuracy, but compared to our work, they show lower effectiveness. We are applying an MDKM that uses a Mahalanobis distance-based clustering process for segmenting the skull and non-brain tissues to focus on the brain. It can work on high-dimensional and complex data, which is significant because MRI consists of noise and different intensity levels across various regions. This segmentation provides fine-grained information about object boundaries, enabling higher accuracy in classification. Whereas traditional segmentation based on region growth, thresholding, and pre-trained CNN models struggles to handle these noisy and high-dimensional data. The O^2DenseNet169 model is well-suited for AD classification as it effectively captures the image's local and global descriptors through its dense connections and feature reuse characteristics. Combined with KM, it performs precise

skull stripping even in a noisy background. Also, the OOA fine-tunes the hyperparameters of the DenseNet169 model and further improves its performance for AD classification. Furthermore, the preprocessing procedures, such as contrast enhancement using CLAHE and DA, help the model identify essential features in the image, leading to improved segmentation while reducing overfitting risks by increasing the diversity of the dataset. These improvements over existing architectures, VGG16, ResNet, EfficientNetB7, and InceptionV3, without optimization and compelling segmentation, lead to higher outcomes for AD classification because the existing pre-trained models did not use the feature reuse mechanism, showing inefficiencies in learning subtle patterns from the image.

5. Conclusion

The study assesses the effectiveness of the O^2DenseNet169 model, MDKM segmentation, and OOA for skull removal and AD prediction using ADNI. Compared to the existing approaches, namely ResNet-50, VGG16, and GoogleNet, the proposed O^2DenseNet169 combined with MDKM achieved better outcomes for segmenting brain tissues and classifying AD. Furthermore, the OOA enhanced the model's performance by fine-tuning the network hyperparameters. Additionally, contrast enhancement using CLAHE improves the differentiation of brain structures. The model's robustness is increased by applying DA, which improves generalization ability. Therefore, the outcomes highlight the performance of the DL architecture called DenseNet169 in extracting complex hierarchical features required for AD classification. The combination of MDKM and OOA provides an effective solution for challenging tasks like skull removal. In the future, it is planned to extend this work by applying multimodal data, including functional and signal-based data, to attain even better performance and explore the effectiveness of other learning and segmentation procedures.

References

[1] Sharmili, K. C., G. P. Suja, E. Pandian, Md Abul Ala Walid, Sripriya Arunachalam, and G. Charles Babu. "An Effective Diagnosis of Alzheimer's Disease with the Use of Deep Learning Based CNN Model." In 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2023, 443-448.

[2] Yang, Qin, Xin Li, Xinyun Ding, Feiyang Xu, and Zhenhua Ling. "Deep Learning-Based Speech Analysis for Alzheimer's Disease Detection: A Literature Review." Alzheimer's Research & Therapy 14, no. 1 (2022): 186.

[3] De Silva, Kevin, and Holger Kunz. "Prediction of Alzheimer's Disease from Magnetic Resonance Imaging Using a Convolutional Neural Network." Intelligence-Based Medicine 7 (2023): 100091.

[4] Alsubaie, Mohammed G., Suhuai Luo, and Kamran Shaukat. "Alzheimer's Disease Detection Using Deep Learning on Neuroimaging: A Systematic Review." Machine Learning and Knowledge Extraction 6, no. 1 (2024): 464-505.

[5] Mohammed, Eqtidar M., Ahmed M. Fakhrudeen, and Omar Younis Alani. "Detection of Alzheimer's Disease Using Deep Learning Models: A Systematic Literature Review." Informatics in Medicine Unlocked 50 (2024): 101551.

[6] Hamdi, Mounir, Sami Bourouis, Kulhanek Rastislav, and Faizaan Mohmed. "Evaluation of Neuro Images for the Diagnosis of Alzheimer's Disease Using Deep Learning Neural Network." Frontiers in Public Health 10 (2022): 834032.

[7] Dara, Omer Asghar, Jose Manuel Lopez-Guede, Hasan Issa Raheem, Javad Rahebi, Ekaitz Zulueta, and Unai Fernandez-Gamiz. "Alzheimer's Disease Diagnosis Using Machine Learning: A Survey." Applied Sciences 13, no. 14 (2023): 8298.

[8] Liu, Ning, Kexue Luo, Zhenming Yuan, and Yan Chen. "A Transfer Learning Method for Detecting Alzheimer's Disease Based on Speech and Natural Language Processing." Frontiers in Public Health 10 (2022): 772592.

[9] Zhao, Zhen, Joon Huang Chuah, Khin Wee Lai, Chee-Onn Chow, Munkhjargal Gochoo, Samiappan Dhanalakshmi, Na Wang, Wei Bao, and Xiang Wu. "Conventional Machine Learning and Deep Learning in Alzheimer's Disease Diagnosis Using Neuroimaging: A Review." Frontiers in computational neuroscience 17 (2023): 1038636.

[10] Leela, M., K. Helenprabha, and L. Sharmila. "Prediction and Classification of Alzheimer Disease Categories Using Integrated Deep Transfer Learning Approach." Measurement: Sensors 27 (2023): 100749.

[11] Mohi ud din dar, Gowhar, Avinash Bhagat, Syed Immamul Ansarullah, Mohamed Tahar Ben Othman, Yasir Hamid, Hend Khalid Alkahtani, Inam Ullah, and Habib Hamam. "A Novel Framework for Classification of Different Alzheimer's Disease Stages Using CNN Model." Electronics 12, no. 2 (2023): 469.

[12] Raza, Noman, Asma Naseer, Maria Tamoor, and Kashif Zafar. "Alzheimer Disease Classification Through Transfer Learning Approach." Diagnostics 13, no. 4 (2023): 801.

[13] Sharma, Shagun, Kalpna Guleria, Sunita Tiwari, and Sushil Kumar. "A Deep Learning Based Convolutional Neural Network Model with VGG16 Feature Extractor for the Detection of Alzheimer Disease Using MRI Scans." Measurement: Sensors 24 (2022): 100506.

[14] Tajammal, Taliah, Syed Khaldoon Khurshid, Abdul Jaleel, Samyan Qayyum Wahla, and Riaz Ahmad Ziar. "Deep Learning-Based Ensembling Technique to Classify Alzheimer's Disease Stages Using Functional MRI." Journal of Healthcare Engineering 2023, no. 1 (2023): 6961346.

[15] Saleh, Ahmad Waleed, Gaurav Gupta, Surbhi B. Khan, Nora A. Alkhaldi, and Amit Verma. "An Alzheimer's Disease Classification Model Using Transfer Learning Densenet with Embedded Healthcare Decision Support System." Decision Analytics Journal 9 (2023): 100348.

[16] Sudharsan, M., and G. Thailambal. "Alzheimer's Disease Prediction Using Machine Learning Techniques and Principal Component Analysis (PCA)." Materials today: proceedings 81 (2023): 182-190.

[17] Yi, Fuliang, Hui Yang, Durong Chen, Yao Qin, Hongjuan Han, Jing Cui, Wenlin Bai, Yifei Ma, Rong Zhang, and Hongmei Yu. "XGBoost-SHAP-Based Interpretable Diagnostic Framework for Alzheimer's Disease." BMC medical informatics and decision making 23, no. 1 (2023): 137.

[18] Patil, Vijeeta, Manohar Madgi, and Ajmeera Kiran. "Early Prediction of Alzheimer's Disease Using Convolutional Neural Network: A Review." The Egyptian Journal of Neurology, Psychiatry and Neurosurgery 58, no. 1 (2022): 130.

[19] Huang, Weichen. "Multimodal Contrastive Learning and Tabular Attention for Automated Alzheimer's Disease Prediction." In Proceedings of the IEEE/CVF international conference on computer vision, 2023, 2473-2482.

[20] Hu, Zhentao, Zheng Wang, Yong Jin, and Wei Hou. "VGG-TSwinformer: Transformer-Based Deep Learning Model for Early Alzheimer's Disease Prediction." Computer Methods and Programs in Biomedicine 229 (2023): 107291.

[21] Zuo, Qiankun, Yanyan Shen, Ning Zhong, CL Philip Chen, Baiying Lei, and Shuqiang Wang. "Alzheimer's Disease Prediction Via Brain Structural-Functional Deep Fusing Network." IEEE Transactions on Neural Systems and Rehabilitation Engineering 31 (2023): 4601-4612.

[22] Venkatasubramanian, S., Jaiprakash Narain Dwivedi, S. Raja, N. Rajeswari, J. Logeshwaran, and Avvaru Praveen Kumar. "Prediction of Alzheimer's Disease Using DHO-Based Pretrained CNN Model." Mathematical Problems in Engineering 2023, no. 1 (2023): 1110500.

[23] Rajesh Khanna, M. "Multi-Level Classification of Alzheimer Disease Using DCNN And Ensemble Deep Learning Techniques." Signal, Image and Video Processing 17, no. 7 (2023): 3603-3611.

[24] Liu, Sheng, Arjun V. Masurkar, Henry Rusinek, Jingyun Chen, Ben Zhang, Weicheng Zhu, Carlos Fernandez-Granda, and Narges Razavian. "Generalizable Deep Learning Model for Early Alzheimer's Disease Detection from Structural MRIs." Scientific reports 12, no. 1 (2022): 17106.

[25] Shamrat, FM Javed Mehedi, Shamima Akter, Sami Azam, Asif Karim, Pronab Ghosh, Zarrin Tasnim, Khan Md Hasib, Friso De Boer, and Kawsar Ahmed. "AlzheimerNet: An Effective Deep Learning Based Proposition for Alzheimer's Disease Stages Classification from Functional Brain Changes in Magnetic Resonance Images." IEEE Access 11 (2023): 16376-16395.

[26] Pandey, Purushottam Kumar, Jyoti Pruthi, Saeed Alzahrani, Anshul Verma, and Benazeer Zohra. "Enhancing Healthcare Recommendation: Transfer Learning in Deep Convolutional Neural Networks for Alzheimer Disease Detection." Frontiers in Medicine 11 (2024): 1445325.

[27] Odusami, Modupe, Rytis Maskeliūnas, Robertas Damaševičius, and Sanjay Misra. "Explainable Deep-Learning-Based Diagnosis of Alzheimer's Disease Using Multimodal Input Fusion of PET And MRI Images." Journal of medical and biological engineering. 43, no. 3 (2023): 291-302.

[28] Pruthviraja, Dayananda, Sowmyarani C. Nagaraju, Niranjanamurthy Mudligiriyappa, Mahesh S. Raisinghani, Surbhi Bhatia Khan, Nora A. Alkhaldi, and Areej A. Malibari. "Detection of Alzheimer's Disease Based on Cloud-Based Deep Learning Paradigm." Diagnostics 13, no. 16 (2023): 2687.

[29] Suchitra, S., Lalitha Krishnasamy, and R. J. Poovaraghan. "A Deep Learning-Based Early Alzheimer's Disease Detection Using Magnetic Resonance Images." Multimedia tools and applications 84, no. 16 (2025): 16561-16582.

[30] Hoang, Gia Minh, Ue-Hwan Kim, and Jae Gwan Kim. "Vision Transformers for the Prediction of Mild Cognitive Impairment to Alzheimer's Disease Progression Using Mid-Sagittal sMRI." Frontiers in Aging Neuroscience 15 (2023): 1102869.

[31] Bhan, Brij, and Shailendra Patel. "Efficient Medical Image Enhancement Using CLAHE Enhancement and Wavelet Fusion." International Journal of Computer Applications 167, no. 5 (2017): 1-5.

[32] Awang, Mohd Khalid, Javed Rashid, Ghulam Ali, Muhammad Hamid, Samy F. Mahmoud, Dalia I. Saleh, and Hafiz Ishfaq Ahmad. "Classification of Alzheimer Disease Using DenseNet-201 Based on Deep Transfer Learning Technique." Plos one 19, no. 9 (2024): e0304995.

[33] Sabaghian, Sahar, Hamed Dehghani, Seyed Amir Hossein Batouli, Ali Khatibi, and Mohammad Ali Oghabian. "Fully Automatic 3D Segmentation of the Thoracolumbar Spinal Cord and the Vertebral Canal from T2-Weighted MRI Using K-Means Clustering Algorithm." Spinal Cord 58, no. 7 (2020): 811-820.

[34] Dalvi, Pooja Pradeep, Damodar Reddy Edla, and B. R. Purushothama. "Diagnosis of Coronavirus Disease from Chest X-ray Images Using DenseNet-169 Architecture." SN Computer Science 4, no. 3 (2023): 214.

[35] Dehghani, Mohammad, and Pavel Trojovský. "Osprey Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems." Frontiers in Mechanical Engineering 8 (2023): 1126450.

[36] https://adni.loni.usc.edu/data-samples/.