IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 March 2024, accepted 26 March 2024, date of publication 28 March 2024, date of current version 3 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3382947

RESEARCH ARTICLE

A Novel Deep Learning Architecture Optimization for Multiclass Classification of Alzheimer's Disease Level

MAHİR KAYA AND YASEMİN ÇETİN-KAYA
Department of Computer Engineering, Faculty of Engineering and Architecture, Tokat Gaziosmanpaşa University, 60250 Tokat, Turkey
Corresponding author: Mahir Kaya (mahir.kaya@gop.edu.tr)

ABSTRACT Alzheimer's disease is a neurodegenerative disorder prevalent in older adults, and early diagnosis is crucial for effective treatment. A deep learning model can automatically classify Alzheimer's disease from magnetic resonance imaging to aid clinicians in diagnosis. Convolutional Neural Networks (CNNs) are commonly used for disease detection in medical images, but their performance is limited due to inadequate labeled data, high inter-class similarity, and overfitting problems. Key hyperparameters influencing CNN performance include the number of convolution layers and filters assigned to each convolution layer. About other hyperparameters, numerous combinations exist. Since CNN models take a long time to train, it is quite costly to try all combinations to find the optimal model. Existing studies have optimized only a few hyperparameters, such as learning rate, batch size, and optimizer in custom and transfer learning models. In this study, we propose an algorithm based on particle swarm optimization to fine-tune the hyperparameters, including the number of convolution layers, filters, and other hyperparameters, in CNN architectures designed to classify Alzheimer's disease severity. Using the proposed lightweight model, Alzheimer's disease was accurately classified with an accuracy of 99.53% and an F1-score of 99.63% on a public dataset. Our model surpasses the performance of previous studies, offering the potential to alleviate the burden on doctors and expedite their decision-making processes. The developed framework can be accessed via the link: "https://ai.gop.edu.tr/alzheimer".

INDEX TERMS Deep learning, convolutional neural network, Alzheimer, optimization, hyperparameter.

I. INTRODUCTION

Alzheimer's disease (AD) is a chronic neurodegenerative disorder. Mild changes in the hippocampus in the brains of individuals with this disease are not detectable. Degenerative symptoms, such as memory loss and language impairment, are observed only as the process advances due to damage to certain nerve cells in the brain [1]. The exact cause of the disease is unknown, and there is no effective treatment. However, experts have observed that 10-15% of individuals with mild cognitive impairment each year will develop Alzheimer's disease in the future [2]. If individuals with mild cognitive impairment are screened early, the transition from mild cognitive impairment to Alzheimer's disease can be delayed or even prevented [3]. Magnetic resonance imaging (MRI) is a common brain imaging technique. This technique is often used to understand the physiological processes in Alzheimer's patients. To assist experts in the detection and classification of Alzheimer's disease in MRI images and to achieve high classification performance values, many researchers utilize artificial intelligence models. Thus, when the Alzheimer's level is detected with computer-aided systems at an early stage, the appropriate treatment process will be initiated.

Convolutional Neural Networks (CNNs), a type of deep learning, have been widely used in image classification and segmentation from medical images [4], [5], [6]. CNNs consist of convolution and pooling layers in the feature extraction and fully connected layers in the classification part [7], [8]. CNNs automatically perform end-to-end learning from raw

The associate editor coordinating the review of this manuscript and approving it for publication was Berdakh Abibullaev.

images in the training phase. Due to the large number of parameters in deep and wide CNN architectures, there is often an overfitting problem in the case of limited labeled data [9]. In addition, various issues such as limited labeled datasets, noises, unbalanced class distributions, and high inter-class similarity are encountered in medical image analysis [6], [10]. Transfer learning is widely used in the case of limited labeled datasets [11]. However, state-of-the-art CNN models often face the problem of overfitting because they contain too many parameters [10], [12], [13]. In CNN architectures, learning takes place by updating the initially randomized filter weights in the convolution layer and the weights in the fully connected layer with the backpropagation algorithm in the training phase [8]. The successful performance of CNNs depends strongly on the architectural design, the combination of hyperparameters, and the dataset. In CNN architecture, there are many hyperparameters such as the number of convolution layers, number of filters in each convolution layer, filter sizes, the number of fully connected layers and neuron size in each fully connected layer, learning rate, optimizer, batch size, dropout rate, and epoch. Since the training phase of CNNs takes a long time, it is quite difficult to manually tune all these hyperparameter combinations [14]. Some studies [15], [16] have only optimized a few hyperparameters, such as dropout, batch size, loss function, and learning rate. When the existing studies are examined, it is found that there is a lack of automatic optimization of the number of convolution layers and the number of filters in each convolution layer, which play an important role in CNN architectures for the successful classification of Alzheimer's disease.

In this study, CNN architectures are optimized with a Particle Swarm Optimization (PSO) based algorithm for the automatic classification of Alzheimer's disease in the early stage using MR images. In the feature extraction part of CNN models, we usually define the convolution, batch normalization, and pooling layers together as a block. A block responsible for feature extraction in input images results in a feature map. In CNN architectures, the first layers usually learn basic features such as line, edge, and color blobs. Subsequent layers learn more complex forms of problem-specific features, so more than one block is usually defined. However, increasing the number of blocks, i.e., deepening the model, causes overfitting of the training dataset [17], [18]. This causes the model to perform poorly on test data that it has not seen before. In addition, the number of filters in each convolution layer that will perform the feature extraction also expresses the width of the model. Increasing the filter size of the models too much causes overfitting in the training phase. Therefore, first of all, the number of convolution layers and the number of filters in a convolution layer should be determined in an optimum way to improve the model performance. In addition, the optimizer, learning rate, number of fully connected layers, and number of neurons in the fully connected layers affect the model performance. The proposed framework determined CNN architecture and hyperparameters optimally with a PSO-based algorithm. The proposed novel CNN architecture achieved classification more successfully than existing studies. This model, which successfully classifies the level of Alzheimer's disease, will reduce the workload of doctors and accelerate the decision-making process.

A. RELATED WORKS

In this section, the studies conducted using machine learning and deep learning methods in the field of Alzheimer's disease diagnosis are presented in two groups. The first group consists of studies that perform binary classification, for example, determining whether Alzheimer's disease is present or not. The second group consists of studies that perform multiclass classification, that is, studies that determine not only the presence but also the type of the disease.

The first group includes studies conducted to detect the presence of Alzheimer's disease. Hussain et al. [19] proposed a CNN-based model for binary classification (Alzheimer/healthy) of Alzheimer's disease using brain MRI data. The performance of the proposed 12-layer CNN model was evaluated on the Open Access Series of Imaging Studies (OASIS) dataset and compared with pre-trained InceptionV3, Xception, MobilenetV2, and VGG architectures. With the proposed 12-layer CNN model, 97.75% accuracy and 97.50% f1-score values were obtained. Erdogmus and Kabakus [20] proposed a CNN model with 12 layers. For 12 hyperparameters, hyperparameter optimization was performed. The DARWIN dataset was used to evaluate the model's performance. This dataset consists of 1D data. Therefore, the data was converted from 1D to 2D in order to transfer the data to the proposed model and other transfer learning models. The binary classification (patient vs healthy) accuracy of the model is 90.4%.

Cui et al. [21] proposed an adaptive logistic regression model for binary classification on the ADNI dataset based on particle swarm optimization (PSO). The PSO algorithm is used in the study to remove redundant features and reduce computational time. In this study, Alzheimer's disease types were compared in pairs. At the end of the study, the accuracy values for AD vs. HC, MCI vs. HC, and cMCI vs. sMCI were 96.27%, 84.81%, and 76.13%, respectively. In the study conducted by Lahmiri [22], a CNN model was developed using MRI images from the OASIS dataset. After feature extraction with CNN, binary classification was performed using the k nearest neighbor (KNN) algorithm. Bayesian optimization (BO) algorithm was used to adjust the parameters in the KNN algorithm. At the end of the study, 94.96 ± 0.0486% accuracy, 92.05 ± 0.0746% sensitivity, and 96.62 ± 0.0350% specificity values were obtained.

Francis and Pandian [23] used an ensemble model to improve the classification accuracy of cognitively normal (CN) and mild cognitive impairment convertible (MCIc) classes. The authors used Xception and MobileNet pre-trained network models with ensemble method. Evaluation of the performance of pre-trained and ensemble models was conducted using data from the Alzheimer's Disease Neuroimaging Initiative (ADNI). The Xception and MobileNet models achieved accuracy rates of 89.23% and 89.89%, respectively. On the other hand, the ensemble model achieves a higher classification accuracy of 91.3%. Li et al. [24] used the EfficientNetB2 model with an attention mechanism to classify mild cognitive impairment (MCI), normal control (NC), and Alzheimer's disease (AD) classes. The ADNI dataset was used in the study, and binary classification was performed. The suggested method's accuracy for AD/NC, AD/MCI, and MCI/NC is 93.30%, 92.42%, and 92.03%, respectively.

Wang et al. [25] stated in their study that existing CNN networks frequently have highly complicated topologies and require enormous data sets. Based on hippocampal segments, the study suggests an accurate and lightweight densely connected 3D convolutional neural network (DenseCNN) for Alzheimer's categorization. The ADNI dataset provides 746 training and 187 testing data points for DenseCNN. The proposed DenseCNN model attained an accuracy of 89.8%. A DenseCNN2 model that distinguishes between Alzheimer's disease vs. control normal classes was presented by Katabathula et al. [26]. Using the Hippmapp3r tool, the hippocampi were divided into left and right segments. While deep visual features were derived from the DenseCNN model, global shape features were obtained using the LB spectrum. These two feature types were then combined in the DenseCNN2 model, and classification was performed by transferring them to the joint model, which included fully connected layers and a softmax classifier. The proposed model's Alzheimer's disease vs. control normal classification accuracy, sensitivity, specificity, and AUC values are, respectively, 92.52, 88.20, 94.95, and 97.89.

The second group includes studies that perform multi-class classification for the diagnosis of Alzheimer's disease. Zeng et al. [27] proposed an SVM-based model for Alzheimer's diagnosis. In the study, the image processing stage was first applied for feature extraction, and then voxel features were obtained. Furthermore, Principal Component Analysis (PCA) was applied to reduce dimension. Then, the parameters of SVM are optimized with a new switching delay particle swarm optimization (SDPSO) algorithm. The proposed SDPSO-SVM model is tested with the ADNI dataset. Classification accuracies of 69.23%, 81.25%, 76.92%, 85.71%, 71.23%, and 57.14% were obtained for sMCI and pMCI, NC and AD, NC and sMCI, NC and pMCI, sMCI and AD and pMCI and AD, respectively.

Transfer learning, a method of using pre-trained networks, has also been used to diagnose Alzheimer's disease. Sun et al. [28] improved the ResNet50 model for Alzheimer's disease diagnosis by including a spatial transformer networks (STN) module and attention mechanisms. They also used the Mish activation function rather than Relu. The proposed model achieved 97.1% accuracy and 95.4% F1-score as a result of the test. Sharma et al. [29] proposed a VGG-16-based model for the diagnosis of Alzheimer's disease. The features to be used for disease diagnosis are determined by the VGG16 model, and classification is performed using an artificial neural network. The model was tested on two sets of data: Dataset1 (four-class) with 6400 images and Dataset2 (three-class) with 6330 MRI images. The accuracy of the four-class classification is 90.4%, compared to 71.1% for the three-class classification. In order to train the VGG-19 architecture, Manimurugan [30] employed the fine-tuning technique. In the study, training and testing were conducted in an 80/20 split using the OASIS dataset. 95.82% accuracy was attained with the VGG-19 model. Furthermore, a comparative analysis was conducted with alternative cutting-edge models, yielding accuracy results of 89.45% for AlexNet, 92.90% for GoogleNet, and 94.91% for VGG-16. Savas [31] conducted a study to measure the performance of different pre-trained models in Alzheimer's disease diagnosis. The ADNI dataset was split 90% for training and 10% for testing. 29 different models were trained and tested with the transfer learning method. The best accuracy value was found to be 92.98% with the EfficientNetB0 model.

Yıldırım and Çınar [32] suggest a hybrid technique in their investigation. The ResNet model is used in the study, and the last five levels are extracted. The model was then enhanced with 10 extra layers, yielding a hybrid model. The dataset was divided into two parts: training (%80) and testing (%20). The proposed hybrid model achieved 90% accuracy. The following are the accuracy values achieved on a class basis: MildDemented received 96.6%, ModerateDemented received 70%, NonDemented received 90%, and VeryMildDemented received 90%. Thangavel et al. [33] proposed the Early Alzheimer's disease - Deep Neural Network (EAD-DNN) model for Alzheimer's disease diagnosis. The study employed two data sets: MRI and comma-separated values (CSV). The Alzheimer_ResNet model was trained after the noise in the images was eliminated. To find the optimal features, Modified Adam's Optimization (MAO) was employed. The multi-classification data were partitioned into two classification matters using the one-versus-the-rest approach. The proposed model achieves a 98% accuracy rate.

Ensemble learning is one of the preferred methods for Alzheimer's disease classification. Sadat et al. [34] proposed an ensemble learning technique using five pre-trained architectures and a scratch model. VGG19, Inception- ResNetv2, ResNet152v2, EfficientNetB5, and EfficientNetB6 architectures were trained with the fine-tuning approach. In the study, the OASIS dataset was used for training, validation, and testing with 60%, 20%, and 20%, respectively. Then, the ensemble process was performed with a weighted average technique. An accuracy of 96% was obtained with the ensemble model. Wang et al. [35] proposed the 3D-DenseNets model that adds dense connections to CNN. It was found that the performance of 3D-DenseNet varies depending on the hyperparameters. As a result, five base 3D-DenseNets with different architectures and hyperparameters were selected after the experiments. A probability-based ensemble method

TABLE 1. Comparison of the studies.

References	Method	Classification Type	Dataset	Accuracy (%)	F1-Score (%)
Hussain et al. [19]	Scratch Model	Binary Classification	OASIS	97.75	97.50
Erdogmus and Kabakus [20]			DARWIN	90.4	90.4
Wang et al. [25]		Multiclass Classification	ADNI	89.8	-
Katabathula et al. [26]			ADNI	92.50	-
Li et al. [24]	Transfer Learning	Binary Classification	ADNI (AD vs. NC)	93.30	93.16
			ADNI (AD vs. MCI)	92.42	92.64
			ADNI (MCI vs.NC)	92.03	92.30
Manimurugan [30]		Multiclass Classification	OASIS	94.82	94.10
Sun et al. [28]			ADNI	97.10	95.4
Sharma et al. [29]			Kaggle	90.4	90.4
Kumar et al. [36]			OASIS	98.35	-
Savaş [31]			ADNI	92.98	-
Thangavel et al. [33]			Kaggle	98	90
Yıldırım and Çınar [32]			Kaggle	90	-
Francis and Pandian [23]	Ensemble Model	Binary Classification	ADNI	91.3	-
Sadat et al. [34]		Multiclass Classification	OASIS	96	95
Wang et al. [35]			ADNI	97.52	97.1
Cui et al. [21]	With the help of Optimization Algorithms	Binary Classification	ADNI (AD vs. HC)	96.27	-
			ADNI (MCI vs. HC)	84.81	-
			ADNI (cMCI vs. sMCI)	76.13	-
Lahmiri [22]			OASIS	94.96	-
Deepa and Chokkalingam [16]		Multiclass Classification	ADNI	97	95.78
Baghdadi et al. [15]			Kaggle	96.65	96.65
			ADNI	96.25	96.22

was applied to combine the results from these base networks.

With the proposed model, they obtained 93.61% accuracy, 92.45% recall, and 94.59% precision in AD/MCI classification. The multi-class (three-class) classification performance of the model is 97.52%.

Another method used in the multi-class classification of Alzheimer's disease is the use of optimization algorithms to optimize hyperparameters. Deepa and Chokkalingam [16] proposed the Optimized VGG-16 model that classifies Alzheimer's disease into three classes. Arithmetic Optimization Algorithm (AOA) was used to optimize the dropout rate and batch size. The study employed three datasets: ADNI, OASIS, and Single Individual Volunteer for Multiple Observations across Networks (SIMON). The study started with the preprocessing of the images. Then, image segmentation was applied. Finally, classification is performed with the proposed Optimized-VGG16 model. The proposed model achieved 97% accuracy and 95.78% F1-score. Baghdadi et al. [15] suggested a system that incorporates transfer learning and the gorilla troops optimizer (GTO). Transfer learning is accomplished by the use of eight cutting-edge models. GTO is used for hyperparameter optimization. Two datasets were used to test the proposed framework. The best accuracy result in the Alzheimer's Dataset was 96.65% with MobileNet. The Xception model achieved the highest accuracy with the ADNI dataset (96.25%).

Table 1 presents a comparison of studies on the classification of Alzheimer's disease.

B. MOTIVATION

Alzheimer's disease is increasingly becoming a major public health problem as the elderly population increases worldwide [16]. This disease causes individuals to lose their cognitive abilities over time, reducing their quality of life. The effects of Alzheimer's disease have profound impacts not only on the individuals affected but also on families and health systems [33], [37]. This study aims to go beyond traditional methods in the classification and diagnosis of Alzheimer's disease. The novel model we have developed aims to improve early diagnosis and classification of the disease. Although there are studies using custom/scratch models and transfer learning in the classification of Alzheimer's disease, we can list the shortcomings in existing studies as follows.

1. In custom and transfer learning models, a small number of hyperparameters such as dropout, batch size, learning rate, and optimizer are generally optimized. This

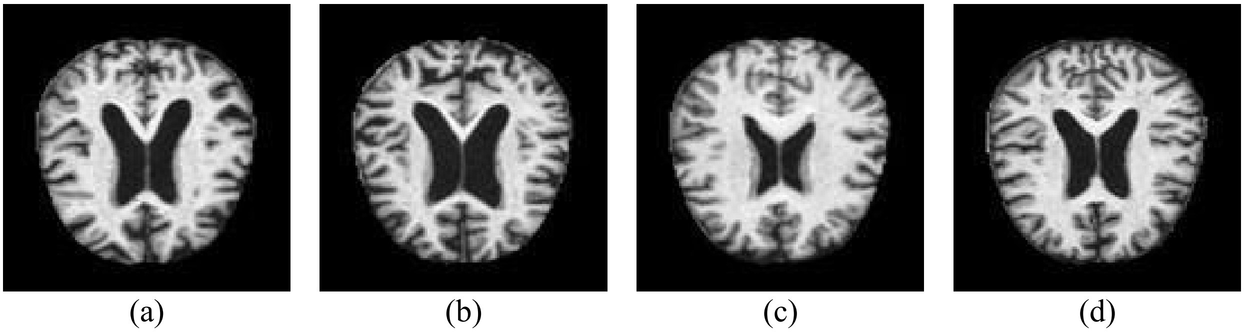


FIGURE 1. Classes included in the Alzheimer's MRI Dataset: (a) Mild Demented, (b) Moderate Demented, (c) Non Demented, (d) Very Mild Demented.

is not enough to obtain a successful result on various datasets.

2. Most existing models face the problem of overfitting in the case of a limited labeled dataset. Optimizing the number of convolution layers and the number of filters in each convolution layer helps to overcome the overfitting problem. The impact of architecturally optimizing CNN needs to be examined.
3. Since state-of-the-art models such as VGG, ResNet, and DenseNet are used in transfer learning, they contain many parameters. The training phases of the models take time, and they are also prone to overfitting.
4. Investigating the effect of residual connections (concatenate) in architectural optimization of CNN.

C. CONTRIBUTIONS

In this study, we proposed a framework that optimizes the CNN architecture for successful classification of the level of Alzheimer's disease. We developed a PSO-based algorithm that optimizes the number of convolution layers and the number of filters in each convolution layer, which are the keystones of CNN architecture for any given dataset and creates a lightweight model. The important contributions of this paper are:

1. Our novelty in this work is to find the lightweight optimal CNN architecture in an iterative method. The architectures created by our proposed framework outperformed existing studies.
2. The suggested method determines the least parameterized and most effective CNN architecture for a given dataset. The most crucial elements of CNN architecture—the number of convolution layers and the number of filters in each convolution layer—are optimized through the use of a PSO-based algorithm.
3. CNN hyperparameters such as learning rate, optimizer, number of fully connected layers, and number of neurons in fully connected layers were also tuned and Alzheimer's level was classified with the highest performance.
4. The lightweight of the model and the short test period increase the applicability of the model in accordance with the purpose of the study.
5. The exploration capability of the PSO algorithm was improved by adaptively changing the number of filters in each iteration by local search.

II. MATERIALS AND METHOD

A. DATASET

In the proposed study, a publicly available dataset called Alzheimer's MRI Dataset [38], published in the Kaggle, is used. It is a dataset created by collecting data from various websites, hospitals, and public databases. It consists of preprocessed MRI images. There are a total of 6400 images in the dataset consisting of Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented categories. The mild dementia class consists of 869 images, the moderate dementia class consists of 64 images, the non-demented class consists of 3200 images, and the very mild dementia class consists of 2240 images. Each image in the dataset is sized as 128 × 128 pixels. Sample images of the data set are shown in Fig. 1.

The data set used in this study has an imbalanced distribution. A number of drawbacks are associated with imbalanced datasets, such as skewed model performance, overfitting to the majority class, poor generalizability, difficulties in identifying within the minority class, and a deficiency of knowledge about the minority class [39]. Since the dataset used in the study has an imbalanced distribution, class weighting was used to minimize these negative effects on performance [39], [40]. The following formula was used to calculate class weights for each class. Mj denotes the number of images in each class j, and c represents the number of classes.

$$class_{weighti} = \frac{\sum_{j=1}^{c} M_j}{c \times M_i} \tag{1}$$

The class weights determined for the classes in the study are as follows: mild dementia: 1.79; moderate dementia: 25. non-dementia: 2; very mild dementia: 2.86.

We used the ADNI dataset as the second dataset for the proposed model. ADNI dataset comprises 2D axial images obtained from the ADNI baseline dataset, which originally comprised NIfTI images. It encompasses three categories: AD (Alzheimer's Disease), CI (Mild Cognitive Impaired), and CN (Common Normal) individuals. We split the dataset into 80% training and 20% testing. In addition, 10% of the training dataset was used for validation during the training phase.

B. PARTICLE SWARM OPTIMIZATION ALGORITHM

Heuristic algorithms can yield faster and more effective results for hyperparameter optimization because they are typically based on experience and intuition in the problem domain. Compared to manual tuning or conventional optimization techniques, this approach saves time and labor. Random and Grid search-based algorithms generally do not act according to the values found in previous iterations. However, heuristic approaches (especially population-based ones) determine and position the new search space according to the best values found in the previous iteration. Heuristics are widely used in many different fields, such as image stitching [41], video analysis [42], and hyperparameter optimization of CNN [15], [16], [21], [22]. CNN models frequently have complex, multidimensional hyperparameter spaces. Such vast search spaces can be efficiently navigated by heuristic algorithms, which also find the ideal hyperparameter values. The optimization process can be improved and balanced with the help of heuristic algorithms. These algorithms can modify the parameters to get a better result by drawing on the knowledge they have learned from earlier iterations. These algorithms can also be easily adjusted to different kinds of problems. To avoid local minima and converge to global minima, they can operate across a wide search space.

In a given problem domain, it is crucial to ascertain which kind of heuristic algorithm to apply. This decision can be affected by the type of data set, the structure of the problem, and the amount of computing power available.

The PSO, which falls under the category of swarm intelligence algorithms, was first presented by [19]. It is composed of a collective unit called a swarm, consisting of individual elements named particles. At the outset, the algorithm generates a group of particles randomly, and through successive iterations, updates are applied to these particles to identify the optimal value. During each iteration, every particle undergoes updates based on two values. The first, denoted as $X_{i,p_{best}}$, represents the best fitness value attained by a particle thus far. The second value, X_{gbest}, signifies the best fitness value achieved by any particle in the entire population. Following the identification of optimal instances for both values, particle velocities, and positions are updated using Equations 2 and 3.

$$V_{i,new} = \omega V_{i,j} + c_1 r_1 \left(X_{i,pbest} - X_{i,j}\right) + c_2 r_2 (X_{g_{best}} - X_{i,j}) \tag{2}$$

$$X_{i,new} = X_{i,j} + V_{i,new} \tag{3}$$

The acceleration factors, denoted as c_1 and c_2, serve as guiding influences for movement toward $X_{i,pbest}$ and X_{gbest},, respectively. The role of c_1 is to direct movement based on the particle's individual experience, while c_2 directs movement based on the collective experience of other particles within the swarm. In every iteration, r_1 and r_2 are assigned random coefficients within the range of 0 to 1 and updated. The inertia weight ω is usually selected to fluctuate between 0.1 and 1.

In the training phase, fitness values were calculated in each iteration of the PSO algorithm based on a randomly generated hyperparameter combination for each particle. First, a CNN architecture was created according to the hyperparameter combination for a particle in an iteration. With this CNN architecture, this model was trained on the training data of the Alzheimer's dataset (this is the most time-consuming part of the optimization of CNN architectures). After CNN training was over, this model was tested on the test dataset, and the test accuracy was calculated. For a particle, this test accuracy is the fitness value. This process is performed with all particles in one iteration. In the iteration, the particle with the best fitness value is found. With local search, the best particle is improved again according to filter numbers. The hyperparameter values of a particle are updated according to the previous best values of the same particle and the values of the best particle in the swarm. In this update, each hyperparameter value is determined according to the closest value according to the specified ranges of each value in the hyperparameter list.

PSO is an optimization technique that is commonly used to solve complex, multidimensional, and nonlinear problems. PSO tends to converge faster than genetic algorithms or other evolutionary computing techniques. Meta-heuristics such as PSO are used to optimize hyperparameters in deep learning models such as CNN (Convolutional Neural Network), which have large and complex search spaces. In our study, we used PSO to optimize our models' hyperparameters because it converges quickly, is straightforward to apply, and works well in high-dimensional and complicated domains. In general, PSO converges more quickly than other optimization techniques. Compared to other optimization techniques, it can have fewer parameters, which facilitates speedy algorithm usage. CNN hyperparameter optimization issues are frequently encountered in complex, multidimensional domains. In such environments, population-based techniques like PSO can demonstrate effectiveness.

C. PROPOSED METHOD

CNN architectures usually consist of many layers of convolution. They are defined sequentially, except for the parallel case in the Inception structure. There are many hyperparameters, such as the number of convolution layers, the number of filters in each convolution layer, learning rate, epoch, optimizer, number of fully connected layers, number of neurons in fully connected layers, and dropout. Since the training time is long, it is almost impossible to try all combinations of hyperparameters. Grid and Random search algorithms also do not work as efficiently as evolutionary algorithms because they do not keep historical information in memory. In order to achieve the best performance of the model, the optimal values of the hyperparameters need to be determined. Many works in the literature do this manually. However, in our study, we performed this process with the PSO-based algorithm, which is one of the evolutionary algorithms. The pseudo-code of the algorithm is shown in Algorithm 1.

The fitness of the objective function is defined as the model's classification accuracy. First, the optimization parameters such as max_iteration, num_particles, c_1, c_2 and ω need to be determined. These parameters are used to control the behavior of the algorithm. Then, two empty lists called best_positions and best_fitness are created. These lists will be used to store the best position each particle finds and the corresponding fitness value. For the initial position and velocity of each particle, two lists called positions and velocities are created and filled with random values. Then, a variable called gBestVal is initialized with a negative infinite value and a list called gBestPos is defined as empty. These variables will be used to store the best position and fitness value found so far.

Algorithm 1 PSO-Based Algorithm

```
for iteration ← 1 to max_iteration
    for i ← 1 to num_particles
        fitness ← build_CNN(positions[i])
        If fitness > best_fitness[i] Then
            best_positions[i] ← positions[i]
            best_fitness[i] ← fitness
        end if
    end for
    If max(best_fitness) > gBestVal Then
        gBestVal ← max(best_fitness)
        gBestPos ←
          best_positions[best_fitness.index(max(best_fitness))]
    end if
    #local search, create a new filter combination for the
    best particle
     for i ← to new_filter_combination
        fitness ← build_CNN(best_particle, new_comb)
        # If fitness is improved, assign new filters
     for i ← 1 to num_particles
        r1, r2 ← Random Numbers
        cognitive_component ← c1×r1×(best_positions[i]
          - positions[i])
        social_component ← c2 × r2 × (gBestPos -
          positions[i])
        velocities[i] ← w × velocities[i] +
          cognitive_component + social_component
        positions[i] ← positions[i] + velocities[i]
        positions[i] ←
          find_closest_hyperparameters(positions[i],
          hyperparameter_list)
    end for
end for
```

Then, within a loop, the maximum number of iterations (max_iteration) is looped. Each iteration represents an update of the particle population. With two nested loops, the fitness value for each particle is calculated and the best found position and fitness value are updated. The best fitness value (gBestVal) and the corresponding position (gBestPos) are updated by selecting the best among the particles. Particle motion and hyperparameter updates are performed. The velocity and position of each particle are updated, taking into account individual and collective experiences.

The computation complexity of Algorithm 1 is O(N × (M × (f+M+h))). N is the number of iterations, M is the number of particles, f is the complexity of building the CNN and calculating the fitness, and h is the complexity of finding the closest hyperparameters.

Another function is defined as find_closest _hyperparameters, shown in Algorithm 2. This function is used to find the closest valid hyperparameters for a given set of hyperparameters.

Algorithm 2 Find_closest_hyperparameters Function

```
procedure find_closest_hyperparameters (values,
  hyperparameter_list)
______________________________________________
closest_list ← Empty List
for value, hyperparameter in zip (values,
  hyperparameter_list)
    min_distance ← Positive Infinity (float('inf'))
    closest_hyperparameter ← None
    for h in htuple
        distance ← abs(v-h)
        if distance < min_distance
            min_distance ← distance
            closest_hyperparameter ← h
        end if
    end for
    append "closest_hyperparameter" to the "closest_list"
end for
return closest_list
```

As shown in Fig. 2, first the random hyperparameter combination is determined for each particle. For each CNN architecture, training is performed, and accuracy and loss values are calculated on the test data. Then, the best hyperparameter combination and the best performance are determined for each particle. The best performance and position of the group are also determined according to all particles. Based on this information, the hyperparameter combinations of the particles are updated, and the iteration continues. The computation complexity of Algorithm 2 is O(n ×m). n is the number of elements in the hyperparameter list of a particle, and m is the number of elements in the hyperparameter_tuple.

In the PSO algorithm, we performed a CNN architecture-specific local search at each iteration. After finding the best particle (CNN architecture) at each iteration, we tried to improve the current best particle by changing the architecture width (the number of filters in each layer), which is an important hyperparameter in CNN architectures, at certain intervals. In CNN architectures, the first layers usually try to extract general features such as edges, corners, and color blobs, while the next layers try to extract problem-specific features according to the number of filters defined. For this reason, we attempted to improve the exploration capability of the PSO algorithm by adaptively changing the number of filters in each iteration by local search. We also defined hyperparameter ranges for each hyperparameter and assigned the closest value in this hyperparameter list when updating the particle velocities at the end of each iteration so that the algorithm can reach the optimum result by trying different combinations more quickly.

D. PERFORMANCE METRICS

Accuracy, recall, precision/positive predictive value(PPV), F1-Score, specificity, false positive rate (FPR), false negative rate (FNR), and negative predictive value (NPV) performance metrics were calculated from a confusion matrix. These performance metrics are used to evaluate the performance of the proposed CNN models.

Accuracy is the ratio of the sum of True-Positive (TP) and True-Negative (TN) values to all data as shown in (4). Recall is the ratio of TP values to the sum of TP and False-Negative (FN) values as shown in (5). Precision is the ratio of TP values to the sum of TP and False-Positive (FP) values as shown in (6). The F1-Score value is the harmonic mean of the precision and sensitivity values as shown in (7). In cases where the class distributions in the dataset are unbalanced, the accuracy should be evaluated together with the F1-score value to determine the classification performance [14]. Specificity is defined as the ratio of TN values to the sum of TP and FP, as seen in (8). False positive rate, as expressed in (9), is the ratio of FP values to the total of FP and TN. The ratio of FN values to the total of FN and TP is known as the false negative rate, as seen in (10). NPV shows the likelihood that a negative test result is actually negative. The calculation formulas of these metrics are as follows:

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

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

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

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{7}$$

$$Specificity = \frac{TN}{TN + FP} \tag{8}$$

$$FalsePositiveRate = \frac{FP}{FP + TN} \tag{9}$$

$$FalseNegativeRate = \frac{FN}{FN + TP} \tag{10}$$

$$NegativePredictiveValue = \frac{TN}{TN + FN} \tag{11}$$

III. EXPERIMENT AND RESULTS

Within this section, the study outlines the experiments and evaluation methodologies utilized for appraising the efficiency of the suggested model. The experiments made use of the MRI dataset referenced in [38]. TensorFlow and Keras deep learning libraries were used to build and train CNN models. All calculations and processes were executed on a regular PC equipped with 16 GB of RAM, an NVIDIA GeForce GTX 1080 Ti GPU featuring 11 GB of memory, and an Intel i5-8400 processor.

The dataset was split 80/20 for training and testing. 10% of the training dataset was set aside for validation.

Data augmentation was used to prevent overfitting. Shear range (0.2), zoom range (0.2), and horizontal flip methods were applied. Each model was trained for 100 epochs. In our preliminary study, we found that the training curve flattened around 100 epochs, indicating that learning did not increase. The number of epochs with the highest accuracy and the lowest loss is set to 100.

The literature and the outcomes of our earlier research were examined to determine which hyperparameters needed to be optimized. Our goal in this study is to create a lightweight model. As a result, we focused primarily on optimizing model architecture parameters. In addition to the optimization algorithms and learning rate that are frequently discussed in the literature, the study includes hyperparameters such as the number of layers, number of filters, and number of neurons that determine the model architecture but have not been thoroughly investigated in the literature. In Table 2, the optimized hyperparameters and the range of values of each hyperparameter are specified to build the CNN models. A preliminary study determined these hyperparameter value ranges as a result of manual optimization. According to the number of convolution layers to be used in the proposed CNN models, the number of filters in each convolution layer is selected from the specified range.

In CNN models, one or two dense layers are selected. If the number of neurons for the second dense layer is zero, the model is built with a single dense layer. The range of the number of neurons in the first dense layer was chosen wider to ensure that the number of neurons in the first dense layer is likely to be higher than in the second dense layer. However, we have not set a strict limitation for this rule.

TABLE 2. The hyperparameter set.

Parameters	Range
Number of Convolution Layers	3, 4, 5, 6, 7, 8, 9, 10
Number of Filters	16, 32, 48, 64, 96, 128, 144, 160, 176, 192, 256
Optimization Algorithms	Adam, SGD with Nesterov, Nadam
Learning Rate	0.0001, 0.001
Number of Neurons (Dense 1 – D1)	32, 64, 96, 128, 144, 160, 176, 192, 256
Number of Neurons (Dense 2 – D2)	0, 16, 32, 64, 96, 112, 128

A. PROPOSED CNN MODELS

In order to create different CNN architectures, the number of convolution layers was first determined. Batch normalization was used after each convolution layer in order to avoid

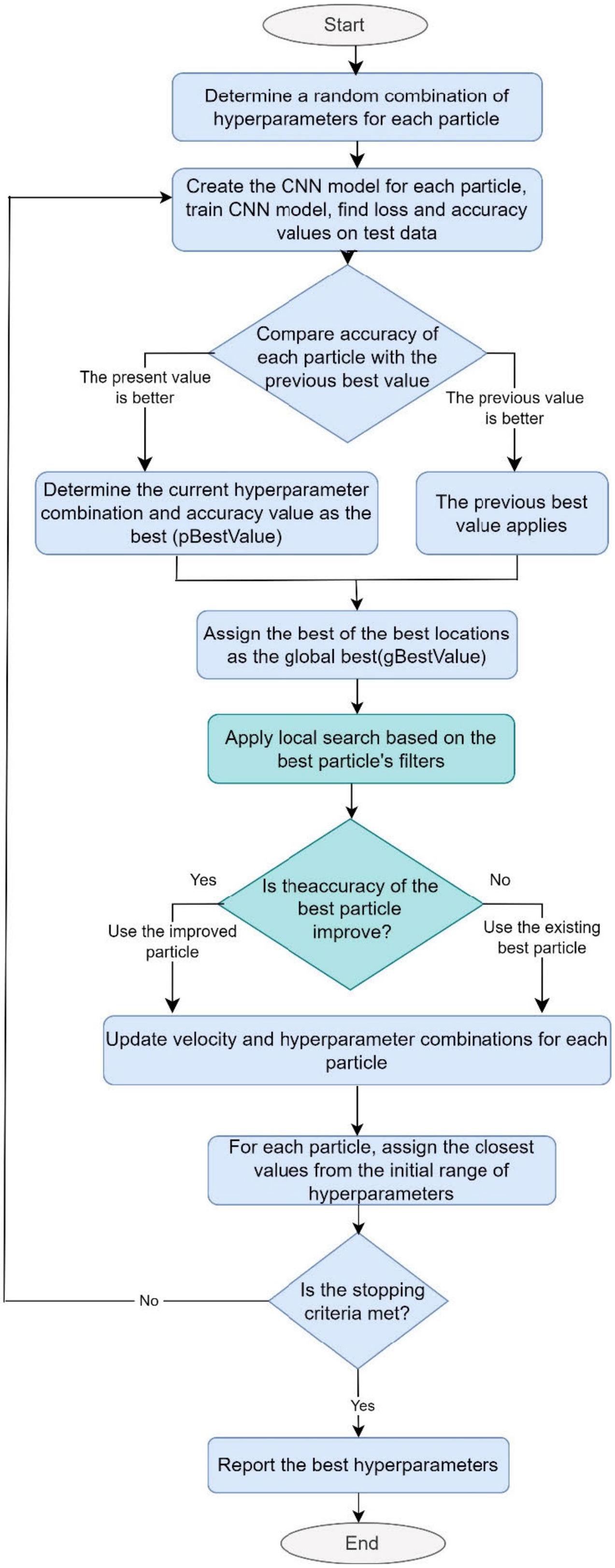

Start
Determine a random combination of hyperparameters for each particle
Create the CNN model for each particle, train CNN model, find loss and accuracy values on test data
Compare accuracy of each particle with the previous best value
The present value is better
The previous value is better
Determine the current hyperparameter combination and accuracy value as the best (pBestValue)
The previous best value applies
Assign the best of the best locations as the global best(gBestValue)
Apply local search based on the best particle's filters
Yes
No
Is theaccuracy of the best particle improve?
Use the improved particle
Use the existing best particle
Update velocity and hyperparameter combinations for each particle
For each particle, assign the closest values from the initial range of hyperparameters
No
Is the stopping criteria met?
Yes
Report the best hyperparameters
End

FIGURE 2. Hyperparameters optimization algorithm flow.

overfitting and for the models to reach the optimum faster. In the models, after the convolution layer, the batch normalization and max-pooling layers are considered as a block. Since the input image size is 128 × 128, max-pooling is limited to a maximum of five. With max-pooling, the feature map size is reduced by subsampling.

Fig. 4 shows the second-best model found by the optimization algorithm. To increase the number of convolution layers after a maximum of 5 blocks, the number of convolution layers is increased starting from the last block. Namely, two convolution layers are defined in the last layer, a batch normalization layer is defined after the convolution layers and a max-pooling layer is defined at the end of the block. Since CNN models extract features with more complex forms specific to the available images in the last layers, the method of increasing the number of convolution layers in the last blocks was chosen. In this CNN architecture, a single dense layer with 128 neurons was found. Fig. 3 shows the best model architecture. After five blocks, the number of convolution layers is increased by two starting from the last block to the third block.

We also investigated the effect of residual connection for each architecture. Fig. 5 shows the architecture with the fifth-best performance. Residual connections have been proposed to solve the problem of gradient vanishing in deep architectures. In ResNet architecture, the architecture is created by aggregating the values of the previous feature maps. However, in DenseNet architecture, it continues by concatenating the output feature map of all previous blocks. Like the ResNet architecture, we created an architecture in which the outputs of the previous block are added, but we chose the concatenate operation in DenseNet instead of the addition operation in ResNet. This allows more information to be stored in the feature maps.

The optimal performance values of CNN models generated according to the hyperparameter ranges given in Table 2 are shown in Table 3. In this study, the PSO-based algorithm is run for two different approaches. In the first approach, the best CNN models are obtained without adding residual connections. In the second approach, the option to add residual connections was added to the algorithm. For each approach, the algorithm ran for 10 iterations and generated hundreds of models. The 10 best performing models are given in Table 3. The models are named in order of performance, with Model 1 achieving the best results.

When the number of convolution layer parameter is examined, it is seen that the models require at least five convolution layers for optimal performance. However, it has been observed that increasing the number of layers has little effect on performance after a certain threshold. The three best results are obtained with 6-8 layers rather than 10 layers. When the number of filters in the convolutional layers is examined, it is apparent that different numbers of filters are chosen in each layer. The traditional strategy of utilizing an equal number of filters in convolutional blocks or increasing the number of filters as the layers deepen was not detected in the models produced by the optimization we performed. When dense layers are analyzed, models with two dense

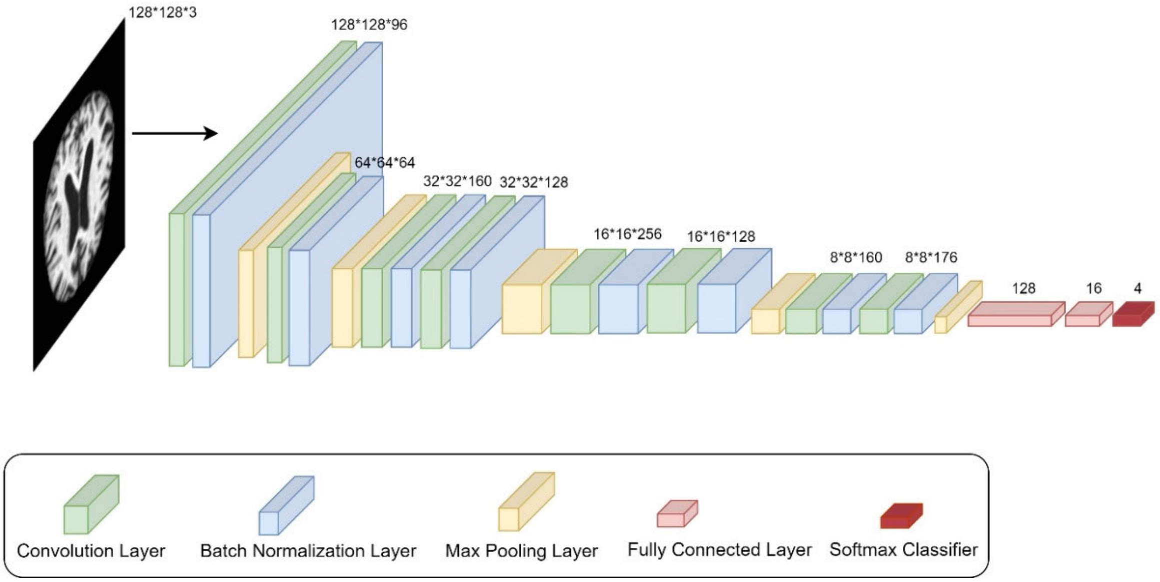


FIGURE 3. CNN architecture of the best model (with the number of filters in each conv layer).

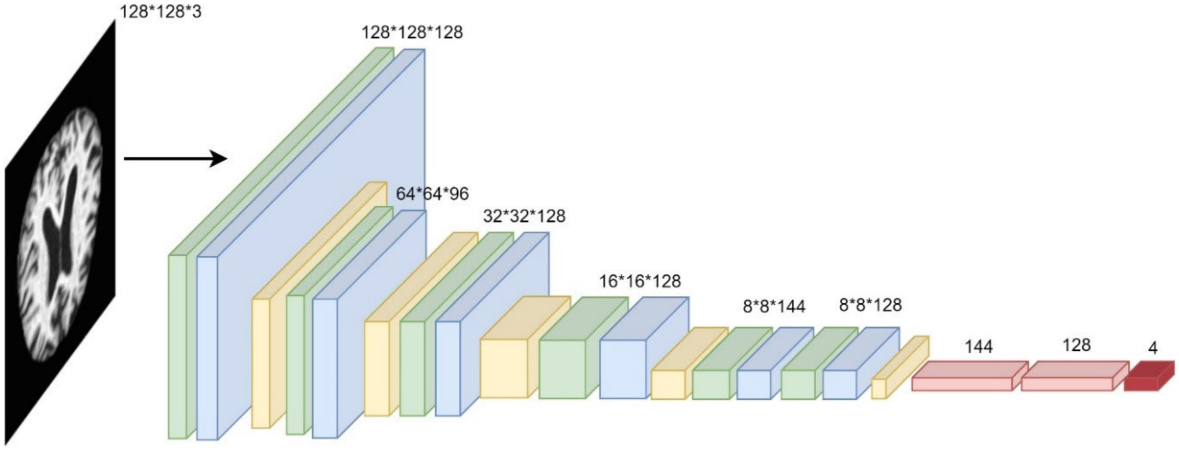


FIGURE 4. CNN architecture of the second best model.

layers have more neurons in the first layer than in the second layer. In addition, models with a single neuron have 64 or more neurons. The models used both of the selected learning rates. Similarly, no single optimization algorithm was found to be superior, with Nadam and SGD-N used in roughly equal proportions.

Table 4 represents the performance metrics of the models. As seen in Table 4, the models achieved high performance in all performance metrics in the classification of Alzheimer's disease level. The results show that the models performed well across all performance metrics in the classification of Alzheimer's disease level.

The architecture in Fig. 3 with eight convolution layers and varying numbers of filters gave the best performance. Increasing the number of convolution layers did not lead to better performance. Nadam as an optimizer, learning rate 0.0001, and two dense layers with 128 and 16 neurons was the best combination. With Model 1, 99.53% accuracy and 99.63% F1-score values were achieved in the classification of Alzheimer's disease.

There is a residual connection in four out of the ten best performing models (Model 5, Model 7, Model 8, and Model 10). Among the models with residual connection, Model 5 performed the best with 98.60% accuracy and 99.05% F1-score. However, the intended performance improvement was not realized by Model 5. When it comes to diagnosing Alzheimer's, the four models (Model 1-4) lacking residual connectivity perform better than this one.

B. ACCURACY/LOSS GRAPHS

Fig. 6 shows the accuracy and loss graphs of the models in Table 3 during training and validation. By looking at the

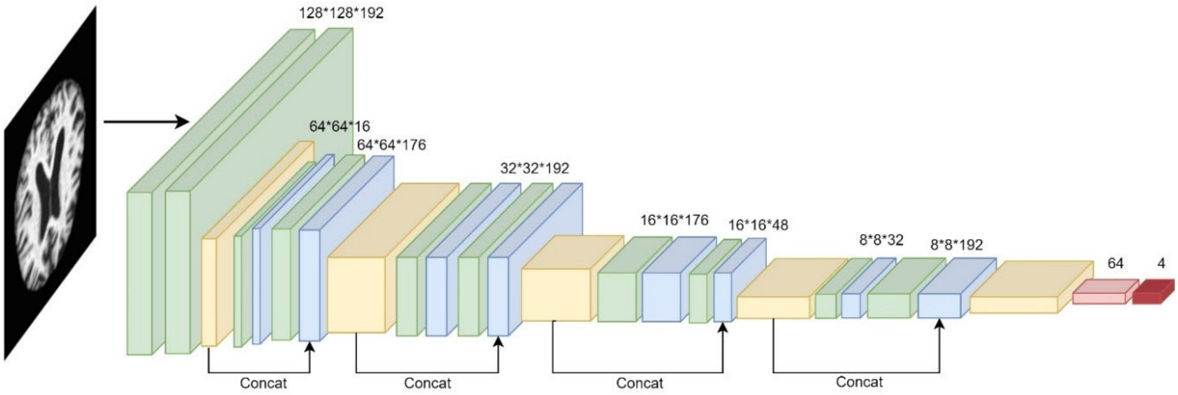


FIGURE 5. CNN architecture of the best model with residual connections.

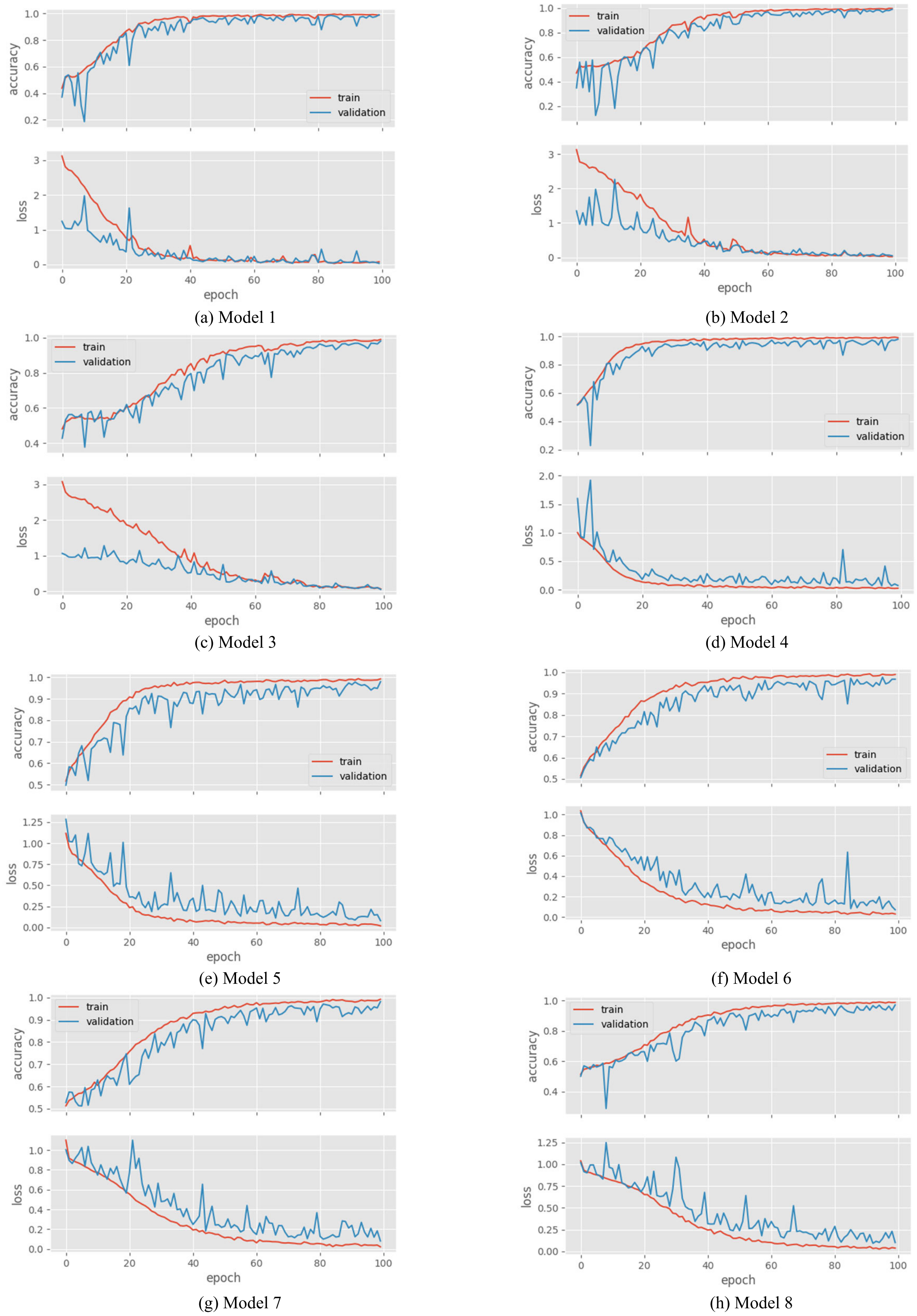
TABLE 3. PSO-based algorithm results for the 10 best performing models.

Model	Number of Convolution Layers	Number of Filters	Number of Dense Layers (Number of Neurons D1, D2)	Learning Rate	Optimization Algorithm	Accuracy (%)
Model 1	8	96, 64, 160, 128, 256, 128, 160, 176	2 (128, 16)	0.0001	Nadam	99.53
Model 2	6	128 ,96, 128, 128, 144, 128	2 (144, 128)	0.001	SGD-N	99.53
Model 3	7	96, 16, 32, 176, 32, 16, 48	2 (192, 32)	0.001	SGD-N	99.38
Model 4	6	128, 160, 16, 160, 256, 16	1 (256)	0.001	Nadam	98.91
Model 5	9	192, 16, 176, 192, 192, 176, 48, 32, 192	1 (64)	0.0001	Nadam	98.60
Model 6	7	128, 128, 64, 128, 32, 128, 64	2 (128, 64)	0.0001	Adam	98.60
Model 7	10	144, 64, 32, 32, 16, 176, 176, 32, 128, 160	1 (128)	0.001	SGD-N	98.29
Model 8	9	128, 256, 256, 32, 64, 16, 256, 256, 128	2 (128, 16)	0.001	SGD-N	98.29
Model 9	5	128 ,32, 128, 128, 64	2 (128, 32)	0.001	Nadam	98.13
Model 10	6	256, 256, 192, 16, 16, 16	1 (112)	0.0001	Nadam	97.98

TABLE 4. Performance metrics of the models.

Model	Accuracy (%)	Precision / PPV (%)	Recall/ Sensitivity (%)	F1-score (%)	Specificity (%)	NPV (%)	FPR (%)	FNR (%)
Model 1	99.53	99.51	99.77	99.63	99.84	99.77	0.16	0.23
Model 2	99.53	99.38	99.70	99.54	99.83	99.80	0.17	0.30
Model 3	99.38	99.39	99.26	99.32	99.78	99.77	0.22	0.74
Model 4	98.91	98.85	99.09	98.96	99.57	99.58	0.43	0.91
Model 5	98.60	99.16	98.93	99.05	99.37	99.42	0.63	1.07
Model 6	98.60	98.85	99.06	98.95	99.40	99.43	0.60	0.94
Model 7	98.29	98.58	98.42	98.50	99.30	99.31	0.70	1.58
Model 8	98.29	98.32	98.74	98.52	99.31	99.29	0.69	1.26
Model 9	98.13	98.61	98.33	98.46	99.17	99.32	0.83	1.67
Model 10	97.98	98.28	98.68	98.47	99.16	99.16	0.84	1.32

accuracy/loss graphs of the models in the training and validation phase, we can see whether there is overfitting or not. In Models 1 (Fig. 6a) and 2 (Fig. 6b), the training and validation acc/loss plots continue to overlap after a certain epoch. There is no overfitting in these models, and the models have reached their full learning capacity as the graphs continue to flatten after a certain epoch. If the models were overfitting, the training accuracy would increase while the validation



(a) Model 1

(b) Model 2

(c) Model 3

(d) Model 4

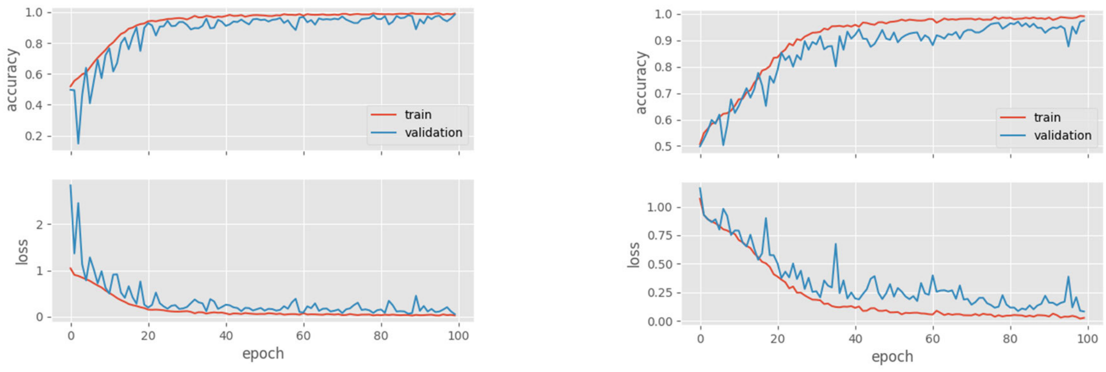
(e) Model 5

(f) Model 6

(g) Model 7

(h) Model 8

FIGURE 6. Train and validation accuracy/loss graphs of the CNN models in Table 3.



(i) Model 9

(j) Model 10

FIGURE 6. ***(Continued.)*** **Train and validation accuracy/loss graphs of the CNN models in Table 3.**

accuracy would start to decrease after a certain epoch. Although Models 4 (Fig. 6d), 5 (Fig. 6e), and 6 (Fig. 6f) learn faster, the validation accuracy curve generally follows the training accuracy curve slightly downwards. In subsequent models, the validation accuracy follows the training accuracy slightly below. Since these models slightly overfitted in the training phase, they were not able to generalize the dataset completely. The same can be said for the loss graphs. Validation loss graphs followed the training loss graphs slightly above. All models generally achieved most of the learning up to 80 epochs.

C. CONFUSION MATRIX

Fig. 7 shows the confusion matrix results of the CNN models in Table 3. In the confusion matrix, row values represent the actual values, and column values represent the predictions of the models. Fig. 7.a shows that all 91 Mild images are correctly classified in the first row of the confusion matrix according to the test dataset. When we look at the first column values, 1 normal (Non) image was incorrectly classified as Mild. In addition, 2 normal (Non) images were misclassified as Very Mild. When Fig. 7.c is examined, 2 Mild images were incorrectly classified as Very Mild. 1 Very Mild image was misclassified as Mild. In Fig.s 7.e and 7.f, five Very Mild images were incorrectly classified as Normal (Non). The absence of these errors is very important for us. It is not desirable to classify Very Mild images as normal. In addition, there are misclassifications between Mild and Very Mild due to intra-class similarity.

D. ITERATION PROCESS

Fig. 8 shows the iteration process of Algorithm 1 with and without the residual link and the change in the performance of the particles in each iteration. In both approaches, the algorithm reached its optimum at the sixth iteration.

E. NUMBER OF PARAMETERS AND PREDICTION TIMES

Fig. 9 shows the number of parameters for each CNN model in Table 3. Model 3 achieved 99.38% accuracy with the lowest number of parameters (289956). Model 1 has 1730500 parameters. Models 5, 7, 8, and 10 are models with residual connections. It is seen that the number of parameters increases in these models. The number of parameters is quite low compared to state-of-the-art CNN models.

The models with residual connections have more parameters than the other models, but they do not perform better. The results show that with proper hyperparameter optimization, we can achieve high performance without using residual connections. In other words, instead of using a fixed number of filters in each layer, specifying the number of filters unique to that layer improves performance.

Fig. 10 shows the time it takes for the proposed CNN models to classify an image. Model 3 gave the best result with 2.88 ms. Model 1 performed quite well with a prediction time of 3.16 ms.

F. COMPARISON OF PROPOSED MODELS WITH STATE-OF-THE-ART CNN MODELS

Fig. 11 shows the accuracy and F1-score values of the state-of-the-art CNN models on the test dataset after being trained on the Alzheimer's dataset under the same experimental conditions using the transfer learning method. Our proposed models outperform the existing CNN models. In addition, the proposed models are lightweight.

Fig. 12 depicts the number of parameters for the study's proposed models and the state-of-the-art models included in the study. The top three models, which show the best accuracy performance among the proposed models, have fewer parameters than the state-of-the-art models.

G. COMPARISON WITH GRAY WOLF OPTIMIZATION ALGORITHM

The results of hyperparameter optimization of CNN models with the Gray Wolf Optimization (GWO) algorithm and the best accuracy and overall accuracy values at each iteration are shown in Fig. 13.a and 13.b. Although the exploration capability of the GWO algorithm is high, the algorithm is

	Mild	Moderate	Non	Very Mild
Mild	91	0	0	0
Moderate	0	7	0	0
Non	1	0	317	2
Very Mild	0	0	0	224

(a) Model 1

	Mild	Moderate	Non	Very Mild
Mild	91	0	0	0
Moderate	0	7	0	0
Non	1	0	319	0
Very Mild	1	0	1	222

(b) Model 2

	Mild	Moderate	Non	Very Mild
Mild	89	0	0	2
Moderate	0	7	0	0
Non	0	0	319	1
Very Mild	1	0	0	223

(c) Model 3

	Mild	Moderate	Non	Very Mild
Mild	90	0	0	1
Moderate	0	7	0	0
Non	1	0	319	0
Very Mild	2	0	3	219

(d) Model 4

	Mild	Moderate	Non	Very Mild
Mild	90	0	0	1
Moderate	0	7	0	0
Non	0	0	317	3
Very Mild	0	0	5	219

(e) Model 5

	Mild	Moderate	Non	Very Mild
Mild	91	0	0	0
Moderate	0	7	0	0
Non	0	0	318	2
Very Mild	2	0	5	217

(f) Model 6

	Mild	Moderate	Non	Very Mild
Mild	88	0	1	2
Moderate	0	7	0	0
Non	1	0	316	3
Very Mild	1	0	3	220

(g) Model 7

	Mild	Moderate	Non	Very Mild
Mild	90	0	0	1
Moderate	0	7	0	0
Non	3	0	316	1
Very Mild	1	0	5	218

(h) Model 8

	Mild	Moderate	Non	Very Mild
Mild	89	0	0	2
Moderate	0	7	0	0
Non	0	0	320	0
Very Mild	2	0	8	214

(i) Model 9

	Mild	Moderate	Non	Very Mild
Mild	91	0	0	0
Moderate	0	7	0	0
Non	0	0	316	4
Very Mild	3	0	6	215

(j) Model 10

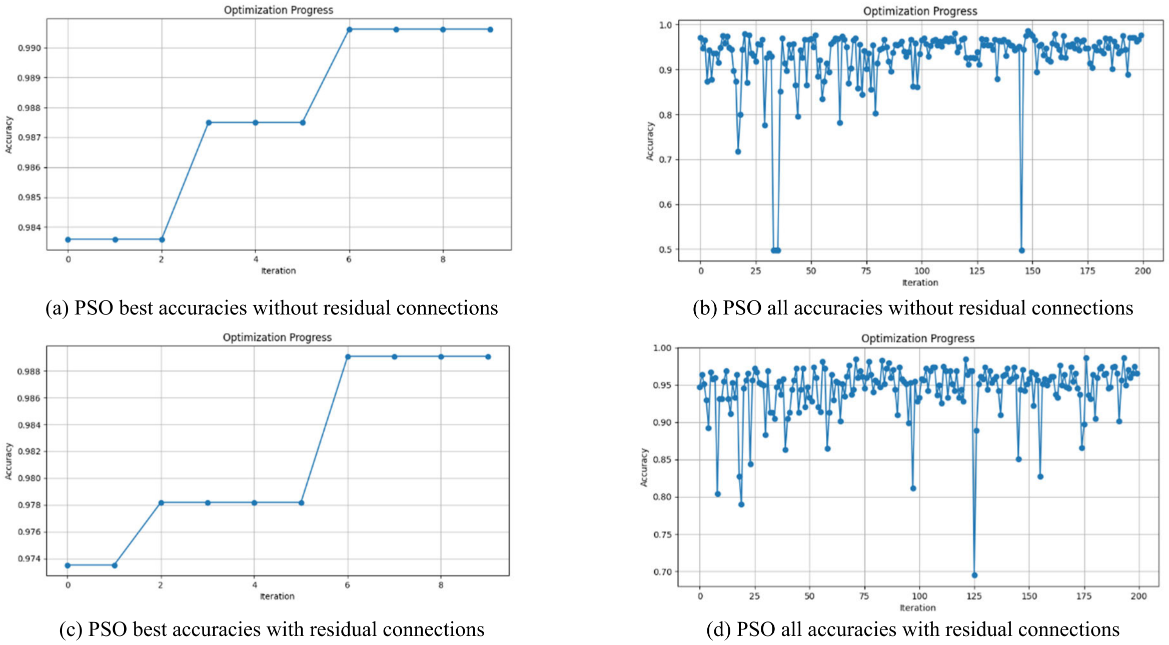
FIGURE 7. **Confusion matrix of the CNN models in Table 3.**

stuck at the local optimum in the first iterations and cannot improve further.

On the other hand, the PSO algorithm found the best result by achieving a good balance between exploration and exploitation. In the PSO algorithm, local search was used to improve the PSO algorithm based on the best particle in each iteration. In each iteration, the best particle was identified, and the filter numbers of this particle were searched again to find the optimum value of the filter numbers based on a narrower range in the first layers and a broader range in the last layers in accordance with the CNN architecture. If a better-performing model was found in the local search, it was replaced with the existing particle in each iteration, and the next iteration was started. Although this additional local search increases the algorithm time in the PSO, this can be tolerated since finding the most accurate model is more important in the health domain.

H. CLASSIFICATION WITH ADNI DATASET

In this study, we retrained our best lightweight model obtained with the PSO-based optimization algorithm on the ADNI dataset with transfer learning. Since we perform four-class classification on the Alzheimer's dataset, there are three classes in the ADNI dataset, namely AD, CI, and CN. For this reason, we changed the softmax layer in the last layer of model-1 to have three classes. We trained with batch size 16, learning rate 0.0001, and epoch 100. As a result of this training, training and validation accuracy/loss curves are given in Figure 14. It can be seen that there is no



(a) PSO best accuracies without residual connections

(b) PSO all accuracies without residual connections

(c) PSO best accuracies with residual connections

(d) PSO all accuracies with residual connections

FIGURE 8. **Iteration process of the proposed method (a-b) without and (c-d) with the residual connections.**

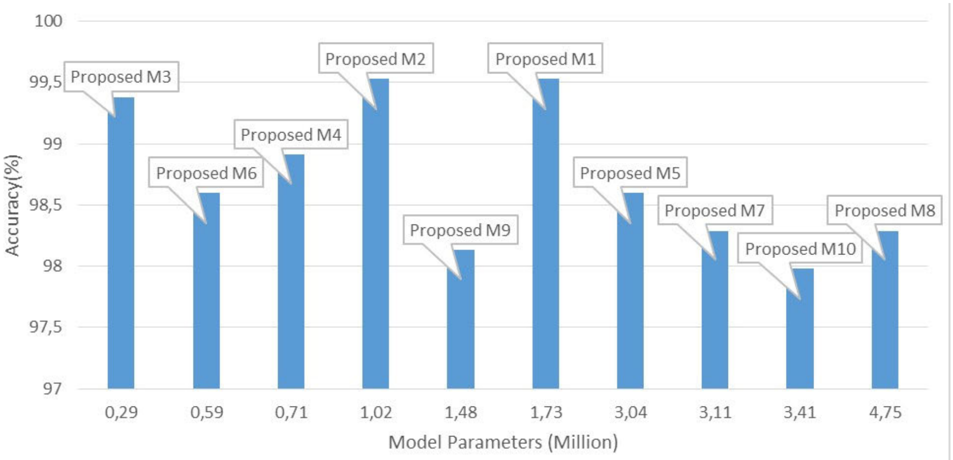


FIGURE 9. **The number of parameters of the CNN models.**

overfitting since the training and validation curves overlap in the accuracy/loss graphs throughout the training.

When the confusion matrix is analyzed in Figure 15, although the true positive predictions are very good, most errors are made between CI and AD. In the second row of the confusion matrix, 6 images that were actually in the CI class were incorrectly predicted by our model and classified as AD. Likewise, 1 CN image was misclassified as AD.

The performance metrics values of the model are presented in Table 5. The model showed high performance with 99.32% accuracy and 99.24% F1-score. The results show that the proposed model can accurately classify Alzheimer's disease across multiple datasets. This demonstrates the proposed model's strong generalization capabilities. It also outperformed the multi-class classification studies conducted in the literature using the ADNI dataset (see Table 1). This suggests that our model can be used successfully in Alzheimer's disease studies instead of state-of-the-art models. As a result, using the suggested method, high performance can be attained by training fewer parameters.

TABLE 5. **Performance metrics of model 1 with the ADNI dataset.**

Accuracy (%)	Precision / PPV (%)	Recall/ Sensitivity (%)	F1-score (%)	Specificity (%)	NPV (%)	FPR (%)	FNR (%)
99.32	98.99	99.32	99.24	99.71	99.57	0.29	0.50

1) END-TO-END ONLINE SYSTEM FOR MULTICLASS CLASSIFICATION OF ALZHEIMER'S DISEASE

Fig. 16 shows an end-to-end web-based online system for classifying levels of Alzheimer's disease. Since the models

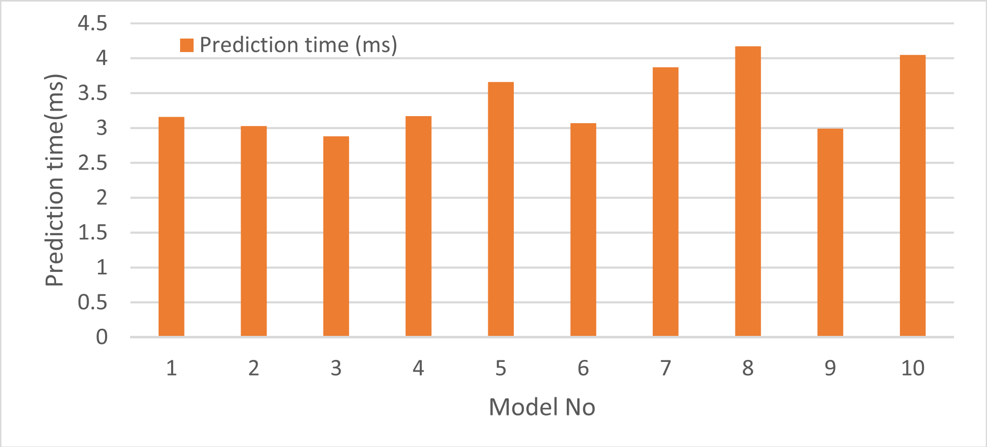


FIGURE 10. Prediction time (ms) of the CNN models for one image.

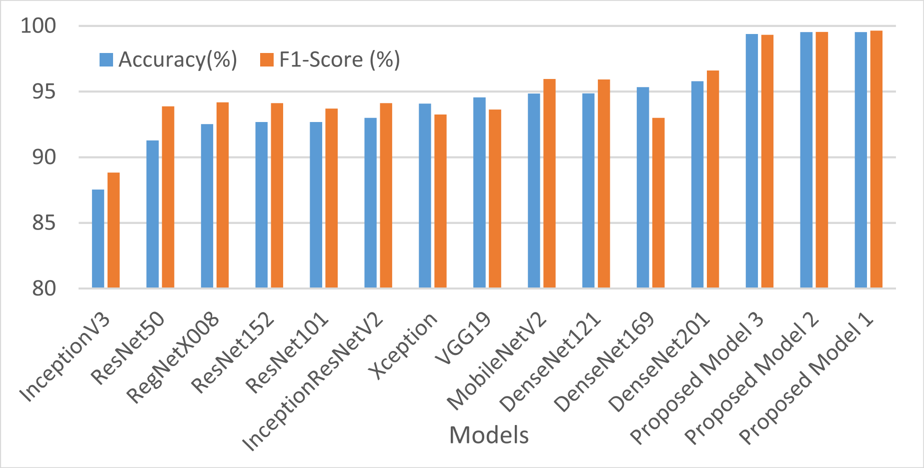


FIGURE 11. Accuracy comparison of proposed models with state-of-the-art CNN models.

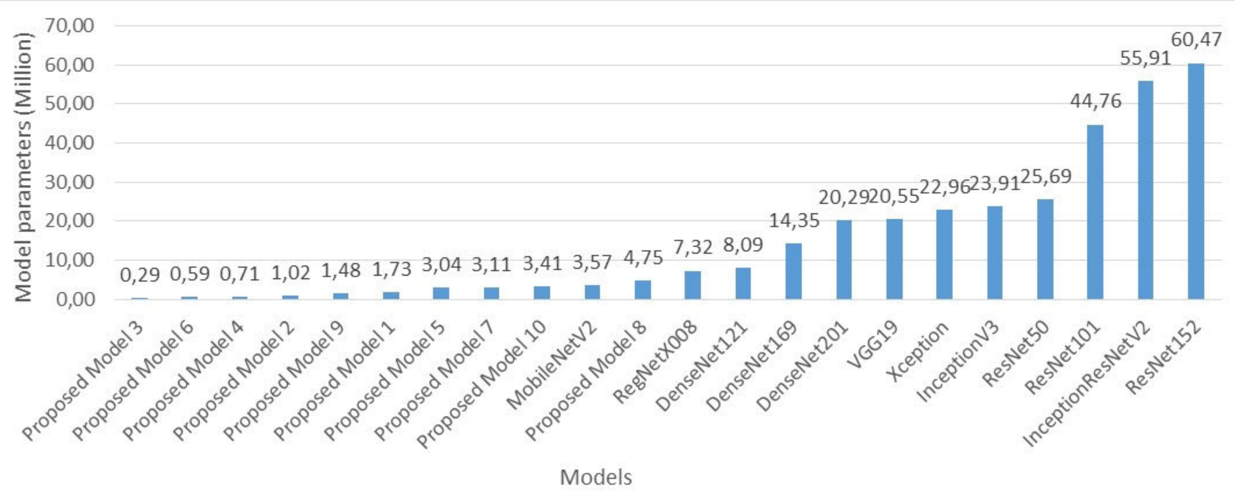
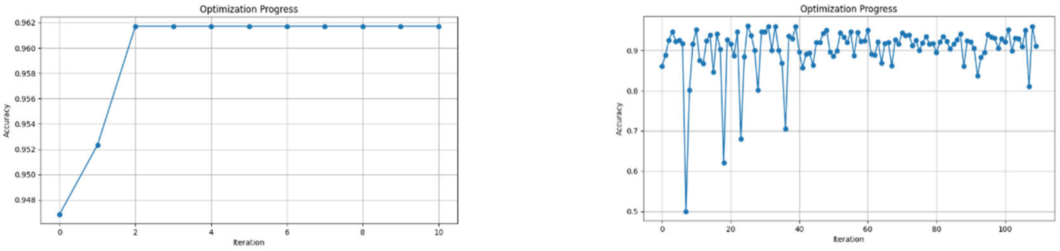


FIGURE 12. Number of parameter comparisons of proposed models with state-of-the-art CNN models.

proposed in the literature are usually not implemented in an online system, end users do not have the opportunity to test the systems by uploading relevant images. For this reason, we have integrated our proposed model into a web-based



(a) GWO best accuracies

(b) GWO all accuracies

FIGURE 13. Results of the GWO algorithm.

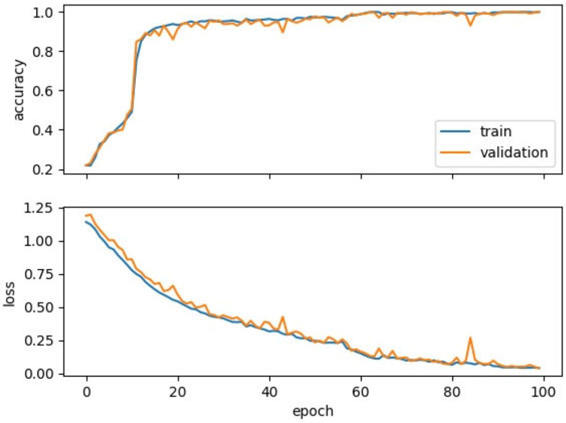


FIGURE 14. Train and Validation accuracy/loss graphs of Model 1 with the ADNI datase.

system so that end-users can upload the images they want in a practical way and get the results. This will speed up the decision-making processes of doctors and alleviate the excessive workload on doctors in the health field.

IV. DISCUSSION

In this study, a lightweight model for diagnosing Alzheimer's disease was developed. Researchers propose custom models for Alzheimer's disease in the literature. However, it is difficult to develop custom models with successful performance due to the limited number and imbalanced data sets. To address this issue, transfer learning is used to train the final layers of deep networks that have already been trained for another problem domain. In this technique, while training the last layers is sufficient for daily life problems, many layers need to be retrained to solve medical problems [13]. Deep tuning (training these models from scratch) takes a long time due to the large number of layers and parameters to complete. In our study, we demonstrated that lightweight models can also perform well in Alzheimer's disease diagnosis. Our findings have demonstrated that it is possible to outperform transfer learning models with few parameters through a well-designed optimization process. This shortens the training time and reduces resource requirements.

	AD	CI	CN
AD	225	0	0
CI	6	512	0
CN	1	0	287

FIGURE 15. Confusion matrix of Model 1 with the ADNI datase.

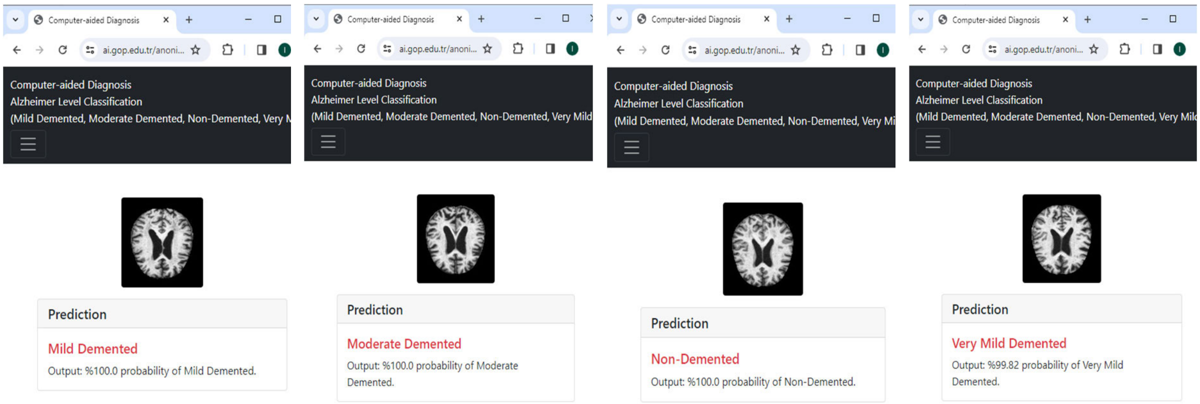
By providing an accurate assessment of Alzheimer's disease, the deep learning model we have developed can enhance patients' quality of life, boost the efficacy of healthcare services, and lead to advancements in treatment and research.

A. COMPARISON OF THE PREVIOUS STUDIES

In the investigations for Alzheimer's disease diagnosis, transfer learning [24], [28], [29], [30], [31], [32], [33], [36], ensemble model [23], [34], [35], optimization algorithms [15], [16], [21], [22], and scratch model creation [19], [20], [25], [26], [32] methods were utilized.

The transfer learning method is the most preferred method in the literature. Except one study [24], this method was used for multiclass classification. The best accuracy value obtained using the transfer learning method [36] is 98.35%. When ensemble models in which the results of more than one model are combined, the accuracy of the most successful model [35] is 97.52%. Five different models were trained for this study, and the predictions were combined using a probability-based ensemble. These models have a large number of parameters and a lengthy training time. Despite using fewer parameters, our proposed model outperforms transfer learning-based models in terms of accuracy, precision, recall, and f1-score performance.

Optimization algorithms have also been used to detect Alzheimer's disease. Machine learning algorithms were optimized with optimization algorithms in studies [21] and [22] while hyperparameter optimization was performed in studies [15] and [16]. In [21], optimization was used to determine the features to be used in the logistic regression algorithm, and in [22], the parameters of the kNN algorithm were tuned using the Bayesian optimization (BO) algorithm. Furthermore, only a few hyperparameters, such as learning rate, batch size, dropout, and optimizer, were optimized in these studies. The study [16] employed the Arithmetic Optimization Algorithm (AOA) to optimize the batch size



Computer-aided Diagnosis
Alzheimer Level Classification
(Mild Demented, Moderate Demented, Non-Demented, Very M

Prediction
Mild Demented
Output: %100.0 probability of Mild Demented.

Computer-aided Diagnosis
Alzheimer Level Classification
(Mild Demented, Moderate Demented, Non-Demented, Very Mild

Prediction
Moderate Demented
Output: %100.0 probability of Moderate Demented.

Computer-aided Diagnosis
Alzheimer Level Classification
(Mild Demented, Moderate Demented, Non-Demented, Very Mi

Prediction
Non-Demented
Output: %100.0 probability of Non-Demented.

Computer-aided Diagnosis
Alzheimer Level Classification
(Mild Demented, Moderate Demented, Non-Demented, Very Mild

Prediction
Very Mild Demented
Output: %99.82 probability of Very Mild Demented.

FIGURE 16. End-to-end online system for multiclass classification of Alzheime's diseas.

TABLE 6. A comparison of the proposed model's performance results with the results of similar studies.

Method	References	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)
Transfer Learning	Manimurugan [30]	94.82	93.24	93.13	94.10	-
	Sun et al. [28]	97.10	95.5	95.3	95.4	-
	Sharma et al. [29]	90.4	90.5	90.4	90.4	-
	Kumar et al. [36]	98.35	-	-	-	-
	Savaş [31]	92.98	-	-	-	-
	Thangavel et al. [33]	98	-	-	90	-
	Yıldırım and Çınar [32]	90	-	-	-	-
Ensemble Model	Sadat et al. [34]	96	95	95	95	-
	Wang et al. [35]	97.52	-	-	97.1	-
With the help of Optimization Algorithms	Lahmiri [22]	94.96	-	92.05	-	96.62
	Deepa and Chokkalingam [16]	97	96.90	96.21	95.78	96.78
	Baghdadi et al. [15]	96.65	96.69	96.62	96.65	98.90
		96.25	96.72	95.77	96.22	98.37
Scratch Model	Wang et al. [25]	89.8	-	98.5	-	85.2
	Katabathula et al. [26]	92.52	-	88.20	-	94.95
	Proposed model	99.53	99.51	99.77	99.63	99.84

and drop rate. Among these studies, [15] performed the best, with 96.65% accuracy. In this study, they performed a transfer learning study with various versions of DenseNet, MobileNet, and VGG models and used Gorilla Troops Optimizer for hyperparameter optimization. Our lightweight model outperformed the studies with 99.53% accuracy. Regarding all of the performance metrics listed in Table 4, our lightweight model performed better than previous research employing optimization algorithms.

Studies on scratch model development have used both binary [19], [20] and multiclass classification [25], [26]. By having the highest accuracy, recall, and specificity among these studies, our proposed new lightweight model outperformed the existing studies that proposed scratch models. Furthermore, despite the fact that the dataset is multi-class, only Alzheimer's disease and control normal classes are classified in [25] and [26]. Our proposed model, on the other hand, completes a more difficult task by classifying four different Alzheimer's classes.

Other studies using the same dataset as ours reported accuracy values of 90.4% [29], 90% [32], 98% [33], and 96.65% [15], whereas our proposed model produced an accuracy value of 99.53%.

Table 6 presents a comparison of the performance results of the proposed model with the results of similar studies that perform multi-class classification. Among all methods, our proposed new lightweight model has better performance indicators in terms of accuracy, precision, recall, F1-score, and specificity values than all the studies in Table 6. This shows the power of the proposed model. A high recall value is crucial because it keeps a patient diagnosed with dementia from being classified as not having dementia. A high precision value is crucial for Alzheimer's diagnosis because false positive values can result in needless treatment and expenses. Our proposed model further demonstrates the strength of our model by providing the best F1-score for the balanced relationship between precision and recall. Our model performs well in identifying patients who are not actually demented, as evidenced by its high specificity. In conclusion, the proposed model outperforms the studies in the literature in terms of accurately identifying Alzheimer's disease from MR images.

We performed 10-fold cross validation on the Alzheimer dataset with proposed Model 1, which has the best architecture obtained as a result of optimization. As a result of 10-fold cross validation, the average validation accuracy value was found to be 99.29%. We applied One-Sample Wilcoxon's Signed Rank test between the 10 validation accuracy values we obtained as a result of 10-fold cross validation and existing studies.

By employing the One-Sample Wilcoxon's Signed Rank test, we determined the p-value to assess if there's a noteworthy divergence between the proposed methodology and previous investigations, aiming to establish statistical significance. This assessment entails a comparison of the proposed method's performance on the Alzheimer dataset with existing studies to maintain fairness. Noteworthy superiority is indicated by a p-value below 0.05 in the comparison between the proposed method and each existing study. Conversely, if the p-value exceeds 0.05, no significant disparity is evident. The statistical test was applied to existing studies [12], [26], [29], [30] using the same dataset as ours. Our method demonstrates significant superiority as evidenced by p-values below 0.01 when compared individually with each existing study.

V. CONCLUSION AND FUTURE DIRECTION

The growth in Alzheimer's disease has increased the importance of early diagnosis. Improvements in the treatment process can be achieved by increasing the classification success in early diagnosis. Although CNN models are widely used in disease detection from medical images, optimized architectures need to be created to improve classification performance. The most important hyperparameters affecting the performance of CNN architectures are the number of convolution layers and the number of filters to be used in each convolution layer. When we consider other hyperparameters, there are many combinations of hyperparameters. In this study, we propose a PSO-based algorithm to optimize the number of convolution layers, filters, and other hyperparameters in CNN architectures for Alzheimer's disease severity classification. With the proposed lightweight model, Alzheimer's disease was classified with 99.53% accuracy and 99.63% F1-score in a publicly available dataset. Our model outperforms existing studies and will reduce the workload of doctors and speed up their decision-making processes.

A. LIMITATIONS

Our study has some limitations in addition to its contributions. The quality and quantity of the datasets utilized directly affect how well CNN models detect diseases. Limited or imbalanced datasets can cause the model to perform less well overall and produce inaccurate findings. Furthermore, even though our work optimized the fundamental and significant CNN architecture-related hyperparameters, additional hyperparameters exist that can be adjusted to develop a model for Alzheimer's disease diagnosis. To address these limitations, larger and different data sets are planned to be used in future studies. This may improve the model's generalizability and ability to recognize various disease types. In addition, the effect of various hyperparameters on model performance is also planned to be investigated. In future studies, inception and attention modules will be included in the optimization process. It is critical to thoroughly assess the model's efficacy and dependability in disease identification. The model's practical applicability and impact on patient outcomes can be studied in greater depth.

The suggested model's capacity to precisely identify the degree of Alzheimer's disease may make it easier for medical practitioners to diagnose patients with the condition and assist them in managing it. Our suggested model can facilitate patient monitoring and improve the diagnostic process in the early stages of the disease. Managing symptoms through early interventions and administering appropriate treatments can help patients lead their daily lives more effectively. The severity of Alzheimer's disease is an important factor in treatment planning. This model can help medical professionals determine the best treatment options for patients based on their severity level. This can help patients better manage their symptoms. The model can provide data on the progression of Alzheimer's disease in patients. Additionally, the model we suggest can aid in the efficient management of healthcare services. This can increase the health system's efficiency by guaranteeing that resources are distributed properly.

REFERENCES

[1] D. AlSaeed and S. F. Omar, "Brain MRI analysis for Alzheimer's disease diagnosis using CNN-based feature extraction and machine learning," *Sensors*, vol. 22, no. 8, p. 2911, Apr. 2022. [Online]. Available: https://mdpi-res.com/d_attachment/sensors/sensors-22-02911/article_deploy/sensors-22-02911.pdf?version=1649643815

[2] M. Grundman, "Mild cognitive impairment can be distinguished from Alzheimer disease and normal aging for clinical trials," *Arch. Neurol.*, vol. 61, no. 1, p. 59, Jan. 2004. [Online]. Available: https://jamanetwork.com/journals/jamaneurology/fullarticle/785241

[3] J. Weller and A. Budson, "Current understanding of Alzheimer's disease diagnosis and treatment," *F1000Research*, vol. 7, p. 1161, Jul. 2018.

[4] K. Barrera, A. Merino, A. Molina, and J. Rodellar, "Automatic generation of artificial images of leukocytes and leukemic cells using generative adversarial networks (syntheticcellgan)," *Comput. Methods Programs Biomed.*, vol. 229, Feb. 2023, Art. no. 107314.

[5] K. Barrera, J. Rodellar, S. Alférez, and A. Merino, "Automatic normalized digital color staining in the recognition of abnormal blood cells using generative adversarial networks," *Comput. Methods Programs Biomed.*, vol. 240, Oct. 2023, Art. no. 107629.

[6] G. Litjens, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classification with deep convolutional neural networks,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–12.

[8] Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] V. Feldman, ‘‘Does learning require memorization? A short tale about a long tail,’’ in *Proc. 52nd Annu. ACM SIGACT Symp. Theory Comput.*, Jun. 2020, pp. 954–959.

[10] M. Kaya, ‘‘Feature fusion-based ensemble CNN learning optimization for automated detection of pediatric pneumonia,’’ *Biomed. Signal Process. Control*, vol. 87, Jan. 2024, Art. no. 105472.

[11] K. Barrera-Llanga, J. Burriel-Valencia, Á. Sapena-Bañó, and J. Martínez-Román, ‘‘A comparative analysis of deep learning convolutional neural network architectures for fault diagnosis of broken rotor bars in induction motors,’’ *Sensors*, vol. 23, no. 19, p. 8196, Sep. 2023.

[12] A. Mabrouk, R. P. D. Redondo, A. Dahou, M. A. Elaziz, and M. Kayed, ‘‘Pneumonia detection on chest X-ray images using ensemble of deep convolutional neural networks,’’ *Appl. Sci.*, vol. 12, no. 13, p. 6448, Jun. 2022.

[13] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, ‘‘Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning,’’ *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.

[14] M. Kaya, S. Ulutürk, Y. Ç. Kaya, O. Altintaş, and B. Turan, ‘‘Optimization of several deep CNN models for waste classification,’’ *Sakarya Univ. J. Comput. Inf. Sci.*, vol. 6, no. 2, pp. 91–104, Aug. 2023.

[15] N. A. Baghdadi, A. Malki, H. M. Balaha, M. Badawy, and M. Elhosseini, ‘‘A3C-TL-GTO: Alzheimer automatic accurate classification using transfer learning and artificial gorilla troops optimizer,’’ *Sensors*, vol. 22, no. 11, p. 4250, Jun. 2022.

[16] N. Deepa and S. P. Chokkalingam, ‘‘Optimization of VGG16 utilizing the arithmetic optimization algorithm for early detection of Alzheimer’s disease,’’ *Biomed. Signal Process. Control*, vol. 74, Apr. 2022, Art. no. 103455.

[17] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, ‘‘How transferable are features in deep neural networks?’’ in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–11.

[18] M. D. Zeiler and R. Fergus, ‘‘Visualizing and understanding convolutional networks,’’ in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.

[19] E. Hussain, M. Hasan, S. Z. Hassan, T. H. Azmi, M. A. Rahman, and M. Z. Parvez, ‘‘Deep learning based binary classification for Alzheimer’s disease detection using brain MRI images,’’ in *Proc. 15th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Nov. 2020, pp. 1115–1120.

[20] P. Erdogmus and A. T. Kabakus, ‘‘The promise of convolutional neural networks for the early diagnosis of the Alzheimer’s disease,’’ *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106254.

[21] X. Cui, R. Xiao, X. Liu, H. Qiao, X. Zheng, Y. Zhang, and J. Du, ‘‘Adaptive LASSO logistic regression based on particle swarm optimization for Alzheimer’s disease early diagnosis,’’ *Chemometric Intell. Lab. Syst.*, vol. 215, Aug. 2021, Art. no. 104316.

[22] S. Lahmiri, ‘‘Integrating convolutional neural networks, kNN, and Bayesian optimization for efficient diagnosis of Alzheimer’s disease in magnetic resonance images,’’ *Biomed. Signal Process. Control*, vol. 80, Feb. 2023, Art. no. 104375.

[23] A. Francis and I. Alex Pandian, ‘‘Early detection of Alzheimer’s disease using ensemble of pre-trained models,’’ in *Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, Mar. 2021, pp. 692–696.

[24] H. Li, Y. Tan, J. Miao, P. Liang, J. Gong, H. He, Y. Jiao, F. Zhang, Y. Xing, and D. Wu, ‘‘Attention-based and micro designed EfficientNetB2 for diagnosis of Alzheimer’s disease,’’ *Biomed. Signal Process. Control*, vol. 82, Apr. 2023, Art. no. 104571.

[25] Q. Wang, Y. Li, C. Zheng, and R. Xu, ‘‘DenseCNN: A densely connected CNN model for Alzheimer’s disease classification based on hippocampus MRI data,’’ in *Proc. AMIA Annu. Symp.*, 2020, p. 1277.

[26] S. Katabathula, Q. Wang, and R. Xu, ‘‘Predict Alzheimer’s disease using hippocampus MRI data: A lightweight 3D deep convolutional network model with visual and global shape representations,’’ *Alzheimer’s Res. Therapy*, vol. 13, no. 1, pp. 1–9, Dec. 2021.

[27] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, and Y. Li, ‘‘A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer’s disease,’’ *Neurocomputing*, vol. 320, pp. 195–202, Dec. 2018.

[28] H. Sun, A. Wang, W. Wang, and C. Liu, ‘‘An improved deep residual network prediction model for the early diagnosis of Alzheimer’s disease,’’ *Sensors*, vol. 21, no. 12, p. 4182, Jun. 2021.

[29] S. Sharma, K. Guleria, S. Tiwari, and S. Kumar, ‘‘A deep learning based convolutional neural network model with VGG16 feature extractor for the detection of Alzheimer disease using MRI scans,’’ *Meas., Sensors*, vol. 24, Dec. 2022, Art. no. 100506.

[30] S. Manimurugan, ‘‘Classification of Alzheimer’s disease from MRI images using CNN based pre-trained VGG-19 model,’’ *J. Comput. Sci. Intell. Technol.*, vol. 1, no. 2, pp. 34–41, 2020.

[31] S. Savaş, ‘‘Detecting the stages of Alzheimer’s disease with pre-trained deep learning architectures,’’ *Arabian J. Sci. Eng.*, vol. 47, no. 2, pp. 2201–2218, 2022.

[32] M. Yildirim and A. Cinar, ‘‘Classification of Alzheimer’s disease MRI images with CNN based hybrid method,’’ *Ingénierie des Systèmes d Inf.*, vol. 25, no. 4, pp. 413–418, Sep. 2020.

[33] P. Thangavel, Y. Natarajan, and K. R. S. Preethaa, ‘‘EAD-DNN: Early Alzheimer’s disease prediction using deep neural networks,’’ *Biomed. Signal Process. Control*, vol. 86, Sep. 2023, Art. no. 105215.

[34] S. U. Sadat, H. H. Shomee, A. Awwal, S. N. Amin, M. T. Reza, and M. Z. Parvez, ‘‘Alzheimer’s disease detection and classification using transfer learning technique and ensemble on convolutional neural networks,’’ in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 1478–1481.

[35] H. Wang, Y. Shen, S. Wang, T. Xiao, L. Deng, X. Wang, and X. Zhao, ‘‘Ensemble of 3D densely connected convolutional network for diagnosis of mild cognitive impairment and Alzheimer’s disease,’’ *Neurocomputing*, vol. 333, pp. 145–156, Mar. 2019.

[36] L. Sathish Kumar, S. Hariharasitaraman, K. Narayanasamy, K. Thinakaran, J. Mahalakshmi, and V. Pandimurugan, ‘‘AlexNet approach for early stage Alzheimer’s disease detection from MRI brain images,’’ *Mater. Today, Proc.*, vol. 51, pp. 58–65, Jan. 2022.

[37] C. Patterson, ‘‘The state of the art of dementia research: New frontiers,’’ *World Alzheimer Report*, 2018. Accessed: Mar. 28, 2024. [Online]. Available: https://www.alzint.org/u/WorldAlzheimerReport2018.pdf

[38] S. Kumar and S. Shastri, ‘‘Alzheimer MRI preprocessed dataset,’’ Tech. Rep. Accessed: Mar. 28, 2024. [Online]. Available: https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset

[39] Y. Sun, A. K. Wong, and M. S. Kamel, ‘‘Classification of imbalanced data: A review,’’ *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, Jun. 2009.

[40] H. He and E. A. Garcia, ‘‘Learning from imbalanced data,’’ *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[41] K. Prokop and D. Połap, ‘‘Heuristic-based image stitching algorithm with automation of parameters for smart solutions,’’ *Expert Syst. Appl.*, vol. 241, May 2024, Art. no. 122792.

[42] D. Połap, ‘‘Neuro-heuristic analysis of surveillance video in a centralized IoT system,’’ *ISA Trans.*, vol. 140, pp. 402–411, Sep. 2023.

MAHİR KAYA received the M.S. and Ph.D. degrees from the Department of Information Systems, Middle East Technical University, in 2010 and 2016, respectively. He is currently an Assistant Professor with the Department of Computer Engineering, Tokat Gaziosmanpaşa University. His research interests include machine learning, deep learning, mobile cloud computing, and optimization.

YASEMİN ÇETİN-KAYA received the M.Sc. and Ph.D. degrees in information systems from Middle East Technical University, Ankara, Turkey, in 2006 and 2014, respectively. She is currently a Faculty Member with the Department of Computer Engineering, Tokat Gaziosmanpaşa University. Her current research interests include deep learning, machine learning, and artificial intelligence.