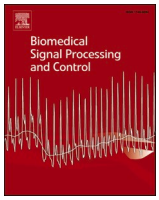
Biomedical Signal Processing and Control 103 (2025) 107422

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc



FiboNeXt: Investigations for Alzheimer's Disease detection using MRI

Turker Tuncer [a], Sengul Dogan [a], Abdulhamit Subasi [b,c,*]

[a] Department of Digital Forensics Engineering, Technology Faculty, Firat University, Elazig, Turkey
[b] Institute of Biomedicine, Faculty of Medicine, University of Turku, Turku 20520, Finland
[c] Department of Computer Science, Effat University, Jeddah 21478, Saudi Arabia

ARTICLE INFO

Keywords:
FiboNeXt
Alzheimer's disease detection
Attention network
Deep learning

ABSTRACT

Background: Deep learning models are currently at the forefront of machine learning. Researchers have proposed and used various deep-learning models. In this research, our primary objective is to introduce a next-generation convolutional neural network inspired by the Fibonacci sequence.
Materials and Methods: We utilized a public Alzheimer's disorder (AD) magnetic resonance imaging (MRI) dataset for this model. This dataset is divided into four categories and includes both augmented and original versions. To detect the AD type, we proposed a new lightweight Fibonacci network, incorporating the structure of ConvNeXt. We also integrated attention and concatenation layers. As a result, we named the proposed convolutional neural network FiboNeXt. The primary goal of FiboNeXt is to achieve high classification capability with fewer trainable parameters, making it a competitive CNN.
Results: The proposed FiboNeXt model was tested on two open-access MRI image datasets comprising both augmented and original versions. The augmented versions were utilized for training, while the original dataset was used for testing. The model achieved 95.40% and 95.93% validation accuracies for the first and second datasets, respectively. Furthermore, it attained test accuracies of 99.66% and 99.63% on the two utilized AD MR image datasets, respectively.
Conclusions: The results and findings unequivocally demonstrate that FiboNeXt is a potent deep-learning model. It holds the potential for addressing other computer vision challenges.

1. Introduction

Dementia encompasses symptoms typified by cognitive decline, originating from various neurologically compromising diseases and injuries [1,2]. Globally, over 55 million individuals struggle with dementia, with 60 % of these cases found in low-to-middle-income countries [3,4]. Annually, around 10 million new instances emerge. Key identifiers include memory lapses, linguistic challenges, directional disorientation, mood, and personality shifts. Alzheimer's Disease (AD) stands as the predominant causal factor, representing 60–70 % of cases [4]. It emerges from a confluence of genetic, lifestyle, and environmental determinants characterized by systematic neuronal degradation [5]. This disorder impairs memory, cognition, and behavioral patterns. Though typically manifesting post-65, early-onset cases are not uncommon [6,7]. While neuropsychological assessments, imaging, and physical evaluations offer preliminary diagnosis, conclusive confirmation often necessitates post-mortem cerebral tissue analysis [8,9]. Presently, no curative or reversive treatments for AD exist; however, current interventions can mitigate symptoms or decelerate disease progression [10].

Recognizing AD at its inception remains paramount [11]. Numerous machine learning methodologies have emerged for this preemptive diagnosis, providing diagnostic support and augmenting specialist consultations [12–15]. This paper introduces a novel machine-learning approach for early AD detection, elaborated in subsequent sections. In this context, this paper introduces a novel machine-learning approach tailored for early AD detection, poised to contribute to the ongoing efforts in advancing diagnostic capabilities and improving patient care. Leveraging cutting-edge technology and innovative methodologies, this approach holds promise in transforming the landscape of dementia diagnosis and management, ultimately fostering better outcomes for individuals affected by this debilitating condition and their families.

* Corresponding author at: Institute of Biomedicine, Faculty of Medicine, University of Turku, Turku 20520, Finland, Department of Computer Science, Effat University, Jeddah 21478, Saudi Arabia.
E-mail addresses: turkertuncer@firat.edu.tr (T. Tuncer), sdogan@firat.edu.tr (S. Dogan), abdulhamit.subasi@utu.fi, absubasi@effatuniversity.edu.sa (A. Subasi).

https://doi.org/10.1016/j.bspc.2024.107422
Received 16 December 2023; Received in revised form 9 October 2024; Accepted 17 December 2024
Available online 25 December 2024

1.1. Literature review

Various machine-learning techniques have been developed in the literature to diagnose many diseases. A brief literature summary for AD is presented in Table 1.

MCI: Mild Cognitive Impairment; HC: healthy controls; CNN: convolutional neural networks; SVM:support vector machine; MLP:Multi-Layer Perceptron; kNN: k-nearest neighbour.

MildD: Mild demented; Moderate demented:ModD; VMD: Very mild demented.

From our review of the literature, we identified several gaps, which are listed below:

- Despite the success of CNNs in medical image analysis, research is scarce exploring the potential of sequence-inspired architectures, such as those based on the Fibonacci sequence, in enhancing model performance and efficiency. The unique properties of the Fibonacci sequence, such as its inherent scalability and efficiency in representing natural phenomena, have not been fully leveraged in the design of deep learning models for AD classification.

Table 1
Literature review.

Study	Method	Classifier	Data	Split ratio	Class	The result(s) %	Limitations
Illakiya et al. [16]	Adaptive Hybrid Attention Network	Softmax	ADNI dataset	60:20:20	3 classes (AD, MCI, HC)	Accuracy: 98.53	There are only three classes.
De Mendonça et al. [17]	Texture extraction	Support vector machine	ADNI dataset	10-fold CV	3 classes (AD, MCI, HC)	Accuracy: 83.80 for HCxAD 74.10 for HCxMCI 75.40 MCIxAD	They used a feature engineering model and they attained limited classification performances
Sethi et al. [18]	CNN	SVM	OASIS dataset	80:20	3 classes (AD, MCI, HC)	Accuracy: 86.20	There are only three classes and their classification accuracy is relatively low.
Cobbinah et al. [19]	Convolutional adversarial autoencoder, Convolutional attention network	Softmax	ADNI dataset	5-fold CV	3 classes (AD, MCI, HC)	Accuracy: 91.80 for HCxAD 88.10 for HCxMCI 90.05 MCIxAD	Their classification accuracy is relatively low.
Yan et al. [20]	Pyramid squeeze attention mechanism, MLP	Softmax	ADNI dataset	5-fold CV	3 classes (AD, MCI, HC)	Accuracy: 98.85 %	They used a well-known deep learning model.
Dogan et al. [21]	Primate brain pattern	kNN	The dataset created by Alzheimer's Patients' Relatives Association of Valladoli	1. 10-fold CV 2. Leave-one subject-out	2 classes (AD, HC)	Accuracy 1. 100.0 2. 92.01	Their dataset only contains two classes.
Kaplan et al. [22]	Feed-forward LPQNet	kNN	1. Collected AD image dataset 2. The Harvard Brain Atlas AD dataset 3. The Kaggle AD dataset	10-fold CV	1. 2 classes (AD, HC) 2. 2 classes (AD, HC) 3. 4 classes (MildD, ModD, VMD, Healthy)	Accuracy: 1. 99.68 2. 100.0 3. 99.64	They used relatively small datasets.
Kaplan et al. [23]	ExHiF model	kNN	1. Collected dataset 2. The Kaggle AD dataset	10-fold CV	1. 2 classes (AD, HC) 2. 4 classes (MildD, ModD, VMD, Healthy)	Accuracy: 1. 100.0 2. 100.0	They used relatively small datasets.
Sorour et al. [24]	CNN	Softmax	ADNI dataset	80:20	3 classes (AD, MCI, HC)	Accuracy: 99.92	There are only three classes.
Akan et al. [25]	CNN, Vision transformer	Softmax	ADNI dataset	80:20	3 classes (AD, MCI, HC)	Accuracy: 95.67	There are only three classes.
Assmi et al. [26]	CNN	Softmax	The Kaggle AD dataset	80:20	4 classes (MildD, ModD, VMD, Healthy)	Accuracy: 92.86	They used a well-known deep learning model.
Goyal et al. [27]	Long short-term memory, generative adversarial network	Softmax	ADNI dataset	80:20	3 classes (AD, MCI, HC)	Accuracy: 96.83	There are only three classes.
Arafa et al. [28]	CNN	Softmax	The Kaggle AD dataset	10-fold CV	4 classes (MildD, ModD, VMD, Healthy)	Accuracy: 97.44	They used a well-known deep learning model.
Adarsh et al. [29]	CNN	Softmax	ADNI dataset	10-fold CV	3 classes (AD, MCI, HC)	Accuracy: 98.27	They used a well-known deep learning model.
Mahmud et al. [30]	CNN	Softmax	OASIS dataset	80:20	4 classes (MildD, ModD, VMD, Healthy)	Accuracy: 96.00	They used a well-known deep learning models.
Prasath and Sumathi [31]	CNN	Softmax	The Kaggle AD dataset	10-fold CV	4 classes (MildD, ModD, VMD, Healthy)	Accuracy: 99.50	They used relatively small datasets.

- Many existing models achieve high accuracy at the expense of many parameters, leading to increased computational cost and memory requirements. There is a gap in the literature regarding developing models that maintain or improve classification performance while significantly reducing the number of trainable parameters. This aspect is crucial for deploying models in resource-constrained environments or for real-time analysis.
- While attention mechanisms have been applied in deep learning models to improve focus on relevant features of input data, their combination with novel architectural designs, such as those inspired by the Fibonacci sequence, remains underexplored. Such integration could potentially enhance the model's ability to identify subtle patterns associated with different stages of Alzheimer's Disease, improving classification accuracy and model interpretability.

The primary objective of this research is to introduce a highly accurate deep learning model (CNN) with fewer learnable parameters. Currently, there is significant competition between Transformers and CNNs in the field of computer vision. After 2020, transformers began being used for computer vision tasks, achieving impressive classification capabilities. Notably, the Vision Transformer (ViT) [32] and Swin Transformer [33] have delivered outstanding classification performances in computer vision tasks. Researchers have introduced new blocks to design next-generation models in response to this development. One of the widely recognized competitive CNNs is ConvNeXt. The two crucial requirements for CNNs are (i) high classification accuracy and (ii) fewer trainable parameters. To address these needs, we have proposed a new ConvNeXt-based CNN model.

In this model, we employed depth-concatenation and self-attention blocks. The depth-concatenation block was used to increase the number of filters, while attention blocks were utilized to focus on regions of interest (ROI) to enhance classification capabilities. We concatenated the outputs of one block to serve as the input for the subsequent block. As a result, our model is named FiboNeXt. We have integrated a mathematical model to devise a new computer vision model in this context. To evaluate our model quantitatively, we utilized an Alzheimer's Disease (AD) MRI image dataset.

The primary aim of the FiboNeXt model is to harness the mathematical elegance and structural efficiency of the Fibonacci series to revolutionize the architecture of CNNs. The Fibonacci series serves as a blueprint for designing the network's layers by dictating the progression of layers and allocating resources within the network. This design strategy ensures that the number of neurons or filters in successive layers adheres to the Fibonacci sequence, optimizing the network's ability to process and recognize complex patterns within data. A critical component of this innovative approach is the enhancement of the model's attention mechanisms. Leveraging the Fibonacci series, FiboNeXt improves the network's focus on the most relevant parts of the input data, thus enhancing its learning from critical features while minimizing distractions from irrelevant information. This Fibonacci-inspired structure promises more efficient distribution of attention across the network, enabling superior management of hierarchical information present in the input data.

Furthermore, applying the Fibonacci series contributes significantly to the model's efficiency and scalability, crucial for processing large datasets and applications requiring real-time analysis. The unique integration of the Fibonacci series with attention-based mechanisms within a CNN framework markedly enhances the model's pattern recognition capabilities, particularly in analyzing complex datasets such as those found in medical imaging for Alzheimer's classification. This approach leads to optimal resource distribution, reduces redundancy and, focuses computational power where most needed, and introduces an innovative method for designing deep learning models. By merging mathematical principles with cutting-edge AI research, the FiboNeXt model aims to refine the network's attention capabilities and structural efficiency, paving the way for significant advancements in the performance of deep learning models across various applications. This pioneering step in the application of the Fibonacci series to the CNN method symbolizes a blend of mathematical rigor and AI innovation, promising to open new avenues for research and development in artificial intelligence, where mathematical principles can guide the creation of more effective and efficient neural networks.

The presented FiboNeXt is a novel CNN. Therefore, it has variable innovations and contributions. The innovations and contributions have been listed below.

Innovations:

- Our research introduces a novel approach by specifically designing a CNN architecture inspired by the Fibonacci series. While various studies have explored the application of CNNs, to our knowledge, no prior work has integrated the Fibonacci sequence as a fundamental design principle within a CNN framework. We have conducted a comprehensive literature review and found no documented evidence of similar methodologies being employed in existing research. However, we acknowledge the dynamic nature of research and welcome any references to prior work that might align with our approach.
- We introduced a novel attention-based CNN aimed at achieving superior classification performance.

Contributions:

- This research presents a new CNN model termed FiboNeXt, inspired by the Fibonacci sequence. By integrating attention mechanisms within a lightweight framework, FiboNeXt achieves enhanced feature recognition capabilities. The model's architecture is designed with fewer than 10 million learnable parameters, setting a new benchmark for efficient yet powerful neural networks.
- The presented FiboNeXt attained over 99 % accuracy in the test scenarios. In this aspect, the recommended FiboNeXt is a highly accurate CNN model.
- Nowadays, researchers generally focus on large models due to the effectiveness of large language models. In this research, we have proposed a lightweight CNN model to contribute to lightweight learning.
- By integrating attention mechanisms into FiboNeXt, this CNN's classification capability and explainability have increased. In this aspect, this model is a highly accurate CNN model and an explainable CNN.
- The proposed FiboNeXt can be adapted to other image classification models. Additionally, by changing the parameters of the presented FiboNeXt, larger versions of this CNN can be achieved.

2. Datasets

In this research, we utilized an open-access dataset related to AD [34]. This dataset is comprised of two types of images: (i) augmented images and (ii) original images. We employed the original images for testing, while the augmented images were used for training. Additionally, we used a public image dataset to obtain comparative results. The AD MRI image dataset we used consists of four classes: (i) mild demented, (ii) moderate demented, (iii) non-demented, and (iv) very mild demented. This dataset can also be downloaded from https://www.kaggle.com/datasets/uraninjo/augmented-alzheimer-mri-dataset URL. Moreover, we have used second version of this dataset and this dataset is a publicly available dataset and researchers can download the AD_v2 dataset using https://www.kaggle.com/datasets/uraninjo/augmented-alzheimer-mri-dataset-v2 URL. The characteristics of the used datasets are defined in Table 2.

According to Table 2, the datasets used contain 40,384 images. To train the proposed FiboNeXt, we utilized 33,984 augmented images and 6,400 original images from 40,384. They only used different

Table 2
The characteristics of the used AD image datasets.

Class	Augmented images	Original images	Total
Mild demented	8960	896	9856
Moderate demented	6464	64	6528
Non demented	9600	3200	12,800
Very mild demented	8960	2240	11,200
Total	33,984	6400	40,384

augmentation techniques.

3. The proposed FiboNeXt

In this work, we introduce FiboNeXt, a novel model tailored for high classification performance by using a next-generation approximation for convolutional neural networks (CNN). Drawing inspiration from the architecture of ConvNeXt, our model integrates addition blocks reminiscent of ResNets to mitigate the vanishing gradient issue. Furthermore, we incorporate a depth-concatenation block, architecting a structure analogous to the Fibonacci series. The proposed FiboNeXt draws inspiration from the ConvNeXt architecture, incorporating state-of-the-art design elements such as addition blocks to mitigate the vanishing gradient problem—a common issue in deep neural networks that hampers effective learning. This innovative synthesis also includes depth-concatenation blocks that emulate the natural efficiency and scalability of the Fibonacci series, setting a new precedent in neural network design for complex pattern recognition tasks.

A comprehensive illustration of the proposed FiboNeXt can be found in Fig. 1.

Fig. 1 shows that FiboNeXt comprises four distinct components: the stem block, the FiboNeXt block, the downsampling block, and the output blocks. In the subsequent sections, we elucidate these components' characteristics and general attributes within the proposed convolutional neural network.

Stem block: In the proposed FiboNeXt model, the stem block is the initial component. This block is meticulously constructed to set the groundwork for the following layers. The stem block's architecture incorporates a convolutional layer characterized by a 5×5 filter size. Subsequent to the convolution, batch normalization is integrated to ensure stability and enhance the efficiency of the training phase. We finalize the operations in this block with the swish activation function, a choice motivated by its demonstrated performance in numerous deep-learning contexts. We will direct readers interested in the stem block's intricate details and mathematical underpinnings to the following section.

$$X^1 = S(B(Cnv(Im))) \tag{1}$$

Herein, X^1 defines the first feature map, $Cnv(.)$ means the convolution,

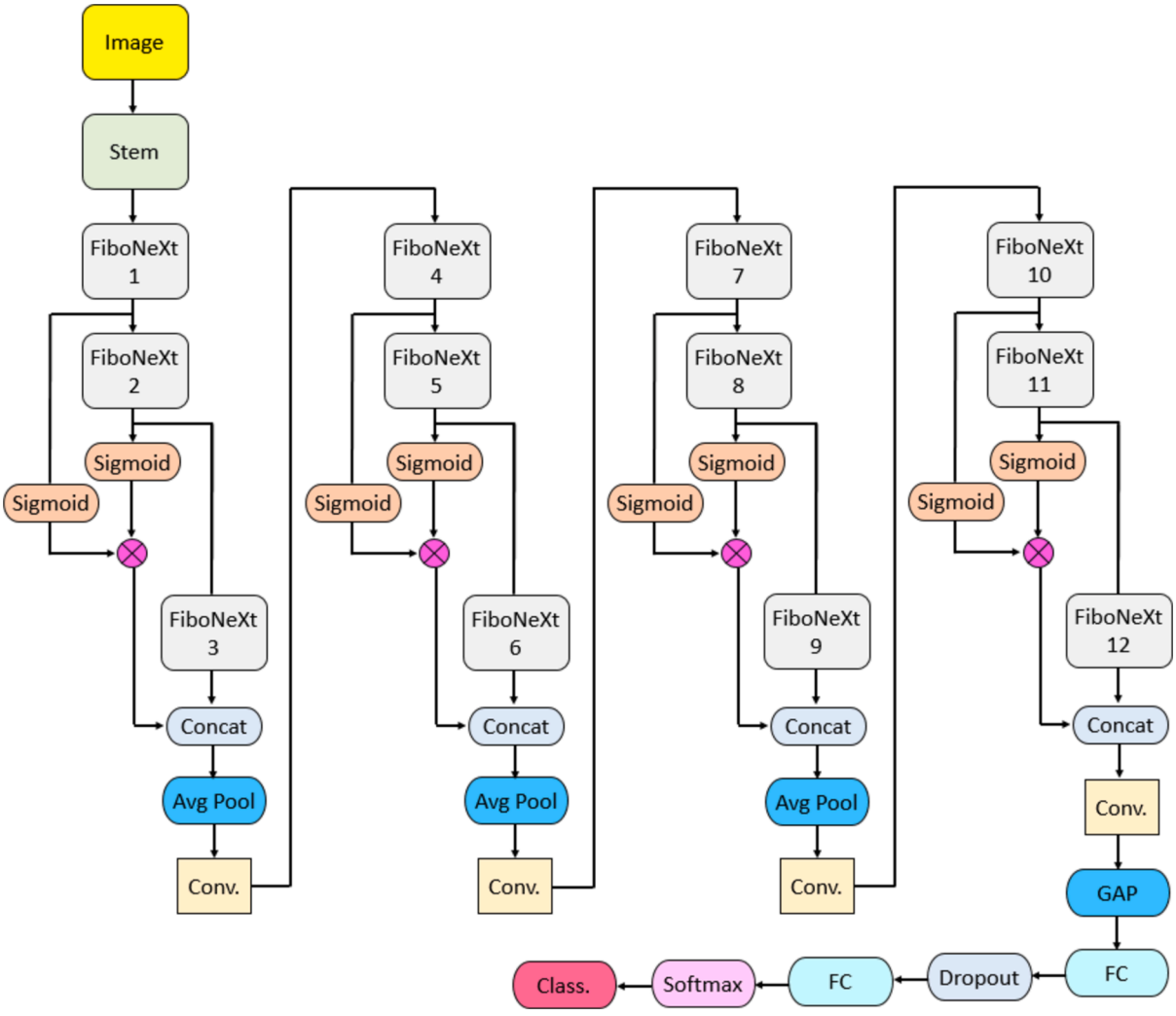


Fig. 1. The general block diagram of the FiboNeXt. The abbreviations of this figure are explained as follows. Concat: depth concatenation, Avg Pool: average pooling, Conv.: convolution, FC: fully connected, GAP: global average pooling, Class.: classification layer.

$S(.)$ is the swish function, $B(.)$ represents the batch normalization, and Im is the image with a size of 224 × 224. The attributes of the used convolution are given as follows. The used filter size is 5 × 5, the number of filters is selected as 96, and the stride is selected as 4 × 4.

FiboNeXt: The cornerstone of the proposed model is the 'FiboNeXt block', which takes inspiration from the architecture of the ConvNeXt block. This block is intricately designed with five convolutional layers. Among these layers, four feature 1 × 1 sized convolution operations serve specific purposes in feature extraction and spatial representations. However, the initial layer in this sequence employs a 3 × 3 convolution, setting the stage for subsequent operations. We introduce an addition block to address and counteract the ubiquitous vanishing gradient problem. Further enhancing the design, an inverse bottleneck structure is embedded within this block, optimizing the flow of information and ensuring efficient feature refinement. A comprehensive illustration of the FiboNeXt block, detailing its layers and their interconnections, can be found in Fig. 2.

The aforementioned FiboNeXt block serves as the core component of the proposed FiboNeXt model. In our design, attention blocks (see Fig. 1) are constructed using multiplication operations. In tandem, the Fibonacci block is crafted by integrating concatenation techniques with this main block. A comprehensive mathematical formulation of this particular block can be found in the subsequent section.

$$X^t = Cnv(S(Cnv(B(Cnv(S(Cnv(B(Cnv(X^{t-1})))))))))) + X^{t-1} \quad (2)$$

Herein, X^t defines output and X^{t-1} is input. We have defined this equation using $Fibo(.)$ function. By using $Fibo(.)$ function, we have created our attention and Fibonacci-like mathematical approximation. This approximation is defined below.

$$X^t = concat\left(sigmoid\left(Fibo\left(X^{t-3}\right)\right) \times sigmoid\left(Fibo\left(X^{t-2}\right)\right), Fibo\left(X^{t-1}\right)\right) \quad (3)$$

Herein, $concat(.)$ defines the concatenation function.

Downsampling block: In our design, we have incorporated a downsampling block reminiscent of the approach used in DenseNet. Specifically, we employ average pooling, utilizing a 2 × 2 pool size coupled with a stride of 2 × 2.

Output block: In the proposed FiboNeXt, we have implemented a standard output block. This block is structured with several layers, from global average pooling to spatially aggregating features. We introduce a fully connected layer to generate a rich feature representation. We have set its output size to 256 to capture essential patterns and nuances. A dropout layer is incorporated to mitigate potential overfitting and enhance generalization, with a dropout probability 0.5. After this, another fully connected layer is added, culminating in applying a softmax function to produce probabilistic outcomes. The final output of the block provides the classification results.

The general attributes of the proposed FiboNeXt: In this section, we have given an overview of the FiboNeXt. The formulation of the used FiboNeXt is: $C = (96, 192, 384, 768), B = (3, 3, 3, 3)$. where C implies the number of filters and B represents the number of repeats. The detailed description of the proposed FiboNeXt is also tabulated in Table 3.

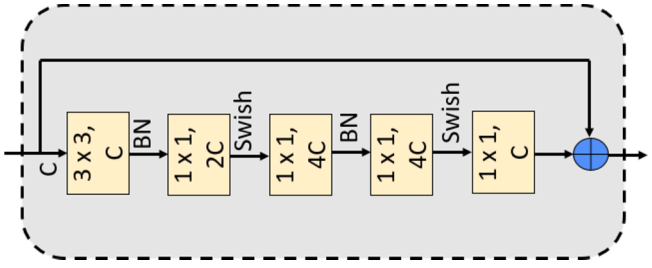


Fig. 2. FiboNeXt block.

Table 3
Details of the proposed FiboNeXt.

Layer	Operation	Input	Output
Stem	7 × 7, 64, BN + Swish, stride: 4.	224 × 224	56 × 56
FiboNeXt 1	$\begin{bmatrix} 3\times3, 96 \\ 1\times1, 192 \\ 1\times1, 384 \\ 1\times1, 384 \\ 1\times1, 96 \end{bmatrix} \times 3$	56 × 56	28 × 28
FiboNeXt 2	$\begin{bmatrix} 3\times3, 192 \\ 1\times1, 384 \\ 1\times1, 768 \\ 1\times1, 768 \\ 1\times1, 192 \end{bmatrix} \times 3$	28 × 28	28 × 28
FiboNeXt 3	$\begin{bmatrix} 3\times3, 384 \\ 1\times1, 768 \\ 1\times1, 1536 \\ 1\times1, 1536 \\ 1\times1, 384 \end{bmatrix} \times 3$	14 × 14	7 × 7
FiboNeXt 4	$\begin{bmatrix} 3\times3, 768 \\ 1\times1, 1536 \\ 1\times1, 3072 \\ 1\times1, 3072 \\ 1\times1, 768 \end{bmatrix} \times 3$	7 × 7	7 × 7
Output	GAP, fully connected layer, Dropout, fully connected layer, softmax, classification	7 × 7	Number of classes
Total learnable parameters			9.9 Million

Per Table 3, the general equation of the proposed model is given below.

$$FiboNeXt = R : [3, 3, 3, 3], F : [96, 192, 384, 768] \quad (4)$$

Herein, R: number of repition and F: number of filters. We have adopted the structure of ConvNeXt, utilizing filters [96,192,384,768] to leverage its architectural efficiency. Furthermore, we have implemented [3,3,3,3] as the number of repetitions to incorporate the Fibonacci logic into our model. Moreover, we have used attention blocks in this model. The equation of the attention block is defined below.

$$Att = Sigmoid(F^{t-2}) \times Sigmoid\left(F^{t-1}\right) \quad (5)$$

Herein, Att: Attention output, F: FiboNeXt block outputs. We have used multiplication-based attention. Equation (5) defines a Fibonacci-like attention. After that, we used depth concatenation to create a general output.

$$Out = concat(Att, F^t) \quad (6)$$

where Out: defines the concatenated output.

The choice of the FiboNeXt model for classification tasks is strongly justified by its structured layers and operations, as detailed in the table, which collectively contribute to its superior performance and efficiency. Starting with the stem block that employs a 7 × 7 convolution with batch normalization and Swish activation on a standard input size, FiboNeXt immediately sets the stage for high-level feature processing, effectively reducing dimensionality while maintaining crucial information.

The core of the FiboNeXt model comprises four distinct blocks, each designed to progressively refine and enhance the feature maps extracted from the input images. These blocks utilize a combination of 3 × 3 and 1 × 1 convolutions, increasing the depth and width of the network in a manner inspired by the Fibonacci series. This unique design allows for a significant expansion in the model's capacity to capture complex patterns and relationships within the data without an exponential increase in computational demand. The repetition of these blocks, with each subsequent block increasing the complexity and depth of the network, ensures comprehensive feature extraction and analysis, crucial for accurate classification tasks.

The final output block, which includes global average pooling followed by fully connected layers and a softmax classification layer, is designed to synthesize the refined features into actionable insights,

providing precise classification outputs across various classes. This structure demonstrates a careful balance between depth and efficiency, ensuring that the model remains lightweight with a total of only 9.9 million learnable parameters. This efficiency is particularly important for deploying the model in real-world scenarios, where computational resources may be limited or where rapid processing of large datasets is required. Moreover, this CNN has an attention structure.

The selection of the FiboNeXt model is underpinned by its innovative architectural design, which leverages depth-concatenation blocks inspired by the Fibonacci series to optimize information flow and feature processing. The model's ability to maintain high classification accuracy with a relatively low number of parameters addresses the critical need for efficient, scalable, and accurate deep learning solutions in the field of medical image analysis. As outlined, the detailed layer and operation design of FiboNeXt showcases its potential to significantly enhance diagnostic processes and improve patient outcomes by providing a powerful tool for accurately classifying complex patterns in medical images.

4. Results and Discussions

This section delves into the experimental results achieved with our proposed FiboNeXt model. Our implementation leveraged the MATLAB programming environment, using its robust tools. Specifically, the MATLAB Deep Network Designer was instrumental in bringing the FiboNeXt architecture to life. The design encompasses 161 layers interconnected by 180 connections, underscoring its depth and complexity.

4.1. Experimental results

The computational environment for the model's design and testing was a standard personal computer. Its specifications included 64 gigabytes of main memory, an NVIDIA GeForce RTX 2070 graphics card, and a central processing unit clocked at 3.6 GHz. This setup ran on the Windows 11 operating system, ensuring smooth operation and efficient computational throughput.

During the initial phase of our experiment, we directed our efforts toward training using the augmented dataset. We allocated data into training and validation sets, adhering to a split ratio of 70:30. The parameters chosen for the training phase of the FiboNeXt were as follows:

Solver: Stochastic Gradient Descent with Momentum (sgdm),
Momentum Coefficient: 0.9,
Initial Learning Rate: 0.01,
Mini-batch Size: 32,
Maximum Epochs: 50,
L2 Regularization Factor: 10^{-4}.

For a comprehensive view of the training progression and validation performance, we charted both the training and validation curves, which can be visualized in Fig. 3.

After completing the training phase, the model achieved a final validation accuracy of 95.40 % for AD augmented dataset and our proposed FiboNeXt reached 95.93 % final validation accuracy for AD augmented dataset v2. This commendable performance showcases the model's ability to generalize well on unseen data during training.

To further validate the trained model's effectiveness, we tested it using the original images. Remarkably, our proposed FiboNeXt exhibited an impressive test accuracy of 99.66 % on this original image dataset. Such high accuracy underscores the robustness and adaptability of our model in real-world scenarios.

We have charted a confusion matrix to offer a more granular insight into the model's classification prowess across different categories. This matrix visually represents true positive, false positive, true negative, and false negative classifications, enabling an intuitive understanding of where the model excels and might have faltered. The computed confusion matrix is illustrated in Fig. 4.

From Fig. 4, we have computed the standard metrics such as precision, recall, and F1-scores, including accuracy, providing a comprehensive view of our model's performance on the test data. These results are systematically tabulated in Table 4. Moreover, we have given brief information about the used performance evaluation metrics, and these are:

Precision: This metric evaluates the number of correctly predicted positive observations from the total predicted positives. High precision indicates that an algorithm returned substantially more relevant results than irrelevant ones [35].

Recall: Recall captures the number of correctly predicted positive observations out of all the actual positives. A high recall indicates that the class was correctly recognized to a large extent [36].

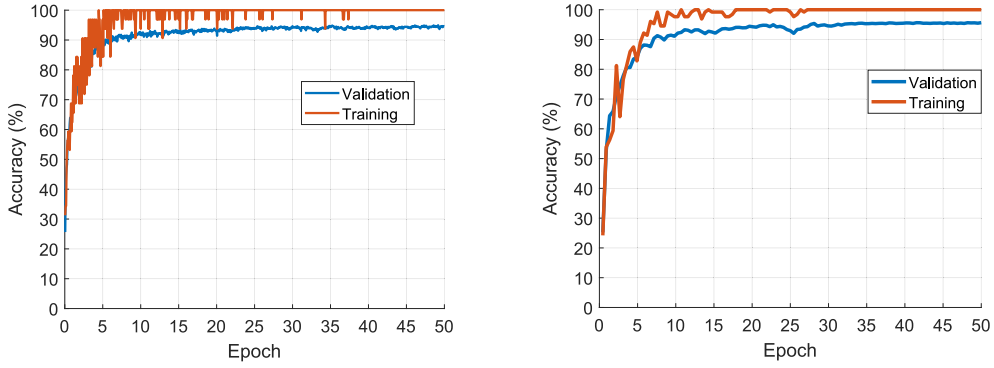
F1-Score: This is the weighted average of precision and recall. Thus, it takes both false positives and false negatives into account. An F1 score is a better metric for uneven class distributions, as it seeks a balance between precision and recall [37].

Accuracy: This metric calculates the ratio of correctly predicted observations to the total observations. While accuracy is intuitive, it can be misleading if the class distribution is imbalanced [38].

The mathematical definitions of the performance evaluation metrics utilized in this study are provided below. We have employed the variables tp (true positive), tn (true negative), fp (false positive), and fn (false negative) to delineate these parameters.

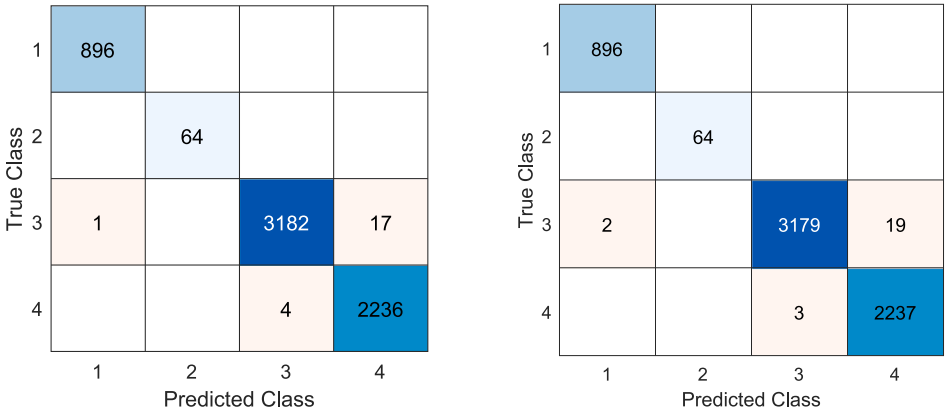
$$Precision = \frac{tp}{tp + fp} \tag{7}$$

$$Recall = \frac{tp}{tp + fn} \tag{8}$$



(a) AD augmented dataset (b) AD augmented dataset v2

Fig. 3. Validation and training curve of the proposed FiboNeXt on the augmented dataset.



(a) Augmented AD dataset (b) Augmented AD dataset v2

Fig. 4. Test confusion matrix of the proposed FiboNeXt. Herein, 1: Mild demented, 2: Moderate demented, 3: Non demented, 4: Very mild demented.

Table 4
The computed performance evaluation results (%).

Class	AD dataset				AD dataset v2			
	Rec.	Pre.	F1	Acc.	Rec.	Pre.	F1	Acc.
Mild demented	100	99.89	99.94	–	100	99.78	99.89	–
Moderate demented	100	100	100	–	100	100	100	–
Non demented	99.44	99.87	99.66	–	99.35	99.81	99.58	–
Very mild demented	99.82	99.25	99.53	–	99.87	99.16	99.51	–
Overall	99.81	99.75	99.78	99.66	99.81	99.69	99.75	99.63

** Rec.: Recall, Pre.: Precision, F1: F1-score, Acc.: Accuracy.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{10}$$

By analyzing metrics like precision, recall, F1-score, and accuracy, we can have a holistic view of our model's performance, identifying its strengths and potential areas of improvement.

As evidenced by Table 4, the FiboNeXt model has showcased exemplary performance across different classes of dementia. Here are some key observations:

Mild Demented Class: The model achieved perfect recall (100 %) in both dataset versions, indicating its exceptional capability to identify all relevant cases of mild dementia accurately. A slight decrease in precision from 99.89 % to 99.78 % in the second dataset version suggests a marginal increase in false positives. However, the F1 scores remain exceptionally high (above 99.89 %), demonstrating robust classification performance.

Moderate Demented Class: This category shows an outstanding 100 % performance across all metrics in both dataset versions, indicating the model's flawless identification and classification capabilities for moderate dementia cases.

Non Demented Class: There's a slight decrease in recall (from 99.44 % to 99.35 %) and precision (from 99.87 % to 99.81 %) when moving from the first to the second dataset version. These changes are minor but indicate a slightly reduced ability to correctly identify all non-demented cases and a minimal increase in false positives in the AD dataset v2. Despite this, the F1-scores remain high, showcasing strong classification accuracy.

Very Mild Demented Class: The recall slightly increased from 99.82 % to 99.87 %, while precision decreased from 99.25 % to 99.16 % when transitioning between dataset versions. These changes suggest a minor improvement in identifying very mild demented cases but at the cost of a slight increase in false positives in the second dataset version.

Overall Performance: The overall metrics show a consistent performance, with a slight decrease in precision and F1-score in the AD dataset v2, leading to a marginal drop in overall accuracy from 99.66 % to 99.63 %. This indicates that while the model performs exceptionally well across both datasets, there is a subtle variation in its effectiveness in differentiating between classes in the second dataset version.

The proposed FiboNeXt model demonstrates high efficiency in classifying different stages of dementia, with especially commendable results for the mild and moderate demented classes.

4.2. Explainable results

In order to explain the high classification performance of the proposed FiboNeXt, we have presented explainable results using heatmaps. We have used the Gradient-weighted Class Activation Mapping (Grad-CAM) model to generate heatmaps. Our proposed model is an attention CNN since we have used a multiplication operator to generate feature maps, and one of our ideas is to generate ROIs. The visualization of some images by deploying Grad-CAM per the classes is depicted in Fig. 5.

Fig. 5 demonstrates that the ROIs use darker colors (red tones), and per these results, the ROIs of these images per the classes are similar. Fig. 5 illustrates that this model has attention ability.

4.3. Ablation studies

We have introduced an ablation section to assess validation accuracies, defining various cases to obtain comparative results for the utilized FiboNeXt model. These cases, all based on the presented FiboNeXt framework, are described as follows:

Case 1: Femto FiboNeXt.
Filters: [32, 64, 128, 256],
Parameters: Approximately 1.1 million.
Case 2: Piko FiboNeXt.
Filters: [40, 80, 160, 320],

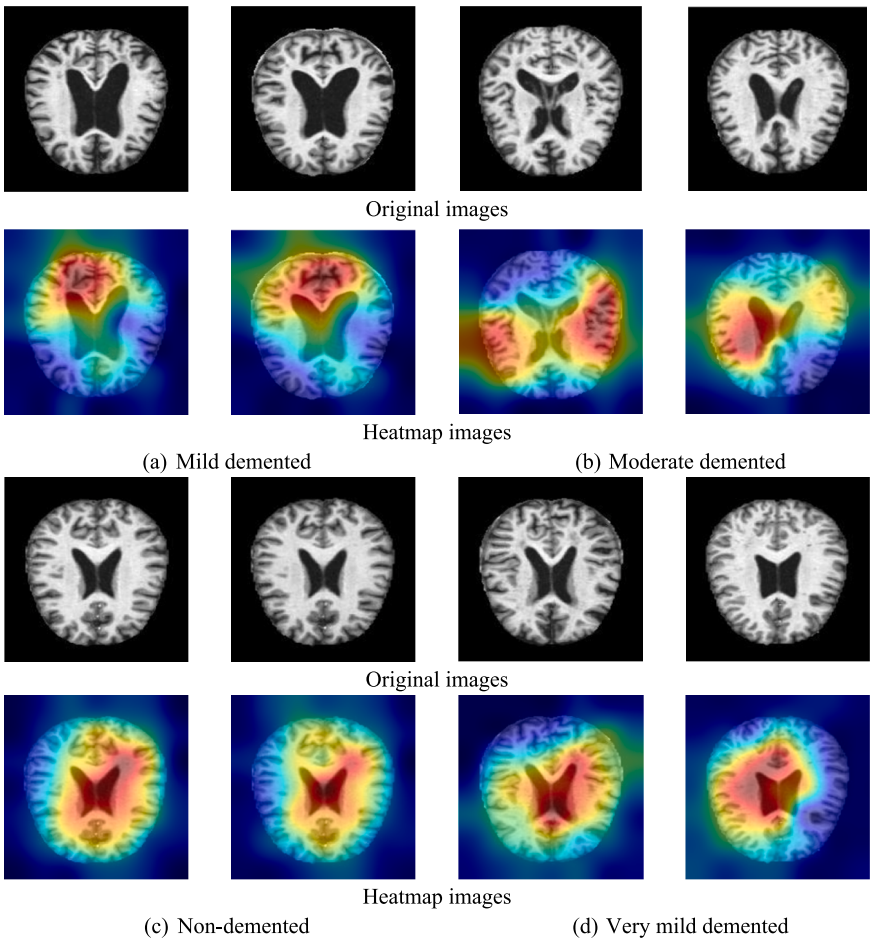


Fig. 5. Heatmaps of the proposed FiboNeXt using sample AD MR images.

Parameters: Approximately 1.8 million.

Case 3: Nano FiboNeXt.

Filters: [64, 128, 256, 512],

Parameters: Approximately 4.5 million.

Case 4: Small FiboNeXt.

Filters: [80, 160, 320, 640],

Parameters: Approximately 7 million.

Case 5: Big FiboNeXt.

Filters: [256, 512, 1024, 2048],

Parameters: Approximately 68.4 million.

Case 6: FiboNeXt without Attention Blocks.

Excludes the attention blocks to evaluate their impact on the model's performance. We have used Base FiboNeXt to create this case.

Case 7: Base FiboNeXt.

Description: The original presented model,

Parameters: Approximately 9.9 million.

The configurations were applied to the second Augmented AD Dataset (version 2), and their validation accuracies were computed and illustrated in Fig. 6.

Fig. 6 shows Case 5 (Big FiboNeXt) achieved the highest validation accuracy of 96.32 %, demonstrating the potential of using larger filter sizes. However, this configuration also has the highest number of learnable parameters (68.4 million).

Case 7 (Base FiboNeXt) achieved a validation accuracy of 95.93 % with significantly fewer parameters (9.9 million), indicating a balance between model complexity and performance.

Case 6 (FiboNeXt without Attention Blocks) decreased validation accuracy to 94.77 %, highlighting the importance of attention blocks in enhancing the model's performance. Including attention blocks

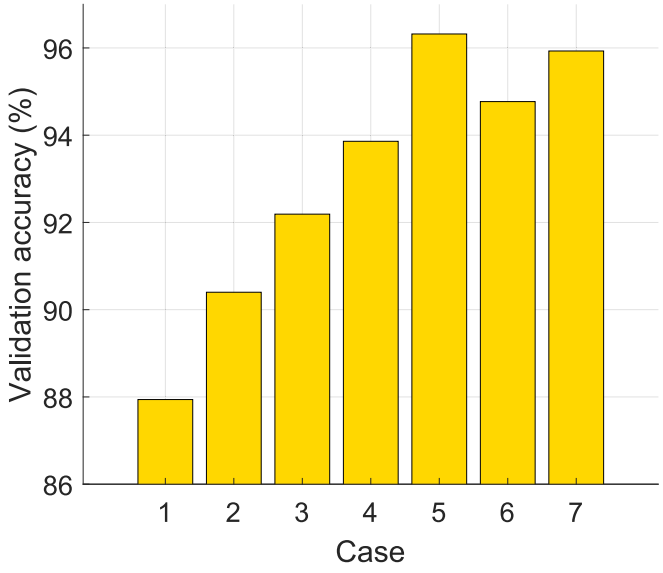


Fig. 6. Validation accuracies of the created cases.

contributed to an increase of 1.16 % in classification accuracy.

Per this ablation study, the detected findings are:

- Increasing the number of filters can improve validation accuracy and significantly increase the number of learnable parameters, affecting computational efficiency.
- Attention blocks are the effective blocks for FiboNeXt.
- The Base FiboNeXt model strikes a good balance between performance and computational efficiency.

4.4. Discussions

In this research, we have developed a new approximation for CNN, and we have used a Fibonacci-based approximation. In order to show the high classification performance of the proposed model, we have used an open-access AD image dataset and obtained 95.40 % validation and 99.66 % test classification accuracies. To better show the performance of our proposal, we have tabulated the comparative results in Table 5.

Per the comparative results, our proposal attained satisfactory classification performance on the used dataset. Moreover, by using the comparative results, we have demonstrated the superiority of the proposed FiboNeXt.

Furthermore, we compared the proposed FiboNeXt model with MobileNetV2, ResNet50, and DarkNet53. To evaluate test accuracies, we utilized the Alzheimer's Disease dataset version 2 (AD dataset v2), and the comparative results are illustrated in Fig. 7.

Table 5
Comparative results (%).

Study	Method	Classifier	Dataset	Accuracy
Rezaee et al. [39]	CNN– Cascade-ResNet	Softmax	9600 non-dementia 8960 very mild dementia 8960 mild dementia 6464 moderate dementia	99.02
Jha et al. [40]	InceptionV3	Softmax	9600 non-dementia 8960 very mild dementia 8960 mild dementia 6464 moderate dementia	97.00
Anitha et al. [41]	SegNet, Restricted Boltzmann Machine	Softmax	9600 non-dementia 8960 very mild dementia 8960 mild dementia 6464 moderate dementia	98.23
Elgendy and Nassif [42]	VGG16	Softmax	6500 MRI images for four classes 40,000 MRI augmented images for four classes	97.00
Taspinar [43]	CNN	Artificial neural network	3200 non-dementia 8960 very mild dementia 8960 mild dementia 6464 moderate dementia	90.00
Our study	FiboNeXt	A new CNN model	Augmented AD dataset Augmented AD dataset v2	99.66 99.63

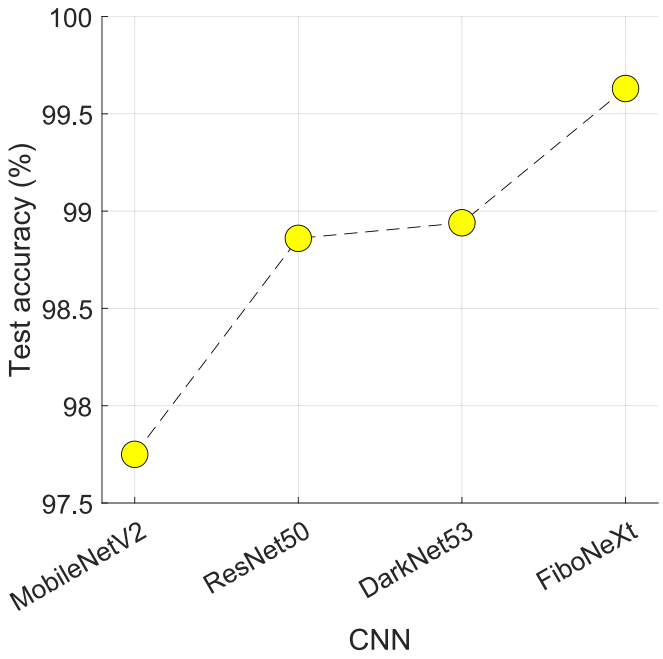


Fig. 7. Comparative test accuracies.

In Fig. 7, the test classification accuracies for MobileNetV2, ResNet50, DarkNet53, and FiboNeXt are presented, achieving 97.75 %, 98.86 %, 98.94 %, and 99.63 % respectively. These outcomes distinctly showcase the superior test classification accuracy of FiboNeXt on the utilized AD image dataset, underlining its effectiveness in comparison to the other CNN architectures.

We have discussed the findings and advantages of the proposed model, which are given below.

Findings:

- The presented FiboNeXt model demonstrated high accuracy, achieving over 99.60 % on the both used datasets. Moreover, this model attained high classification performance for two AD image datasets.
- With stellar recall, precision, and F1-score values across different classes of dementia, FiboNeXt has proven its efficacy in predictive modeling, especially in achieving perfect scores in identifying and classifying certain classes.
- Despite inherent challenges in distinguishing between non-demented and very mild demented classes, the model managed commendable recall rates, showcasing its robustness in handling subtle class variations.
- Inspired by the Fibonacci series, FiboNeXt integrates depth-concatenation blocks and addition blocks similar to ResNets but with a unique twist. This structure allows for a scalable and efficient increase in network depth and complexity directly influenced by the mathematical properties of the Fibonacci sequence. The model specifically addresses the vanishing gradient problem through these addition blocks and enhances feature extraction through depth concatenation.
- Despite its complexity, FiboNeXt remains remarkably efficient, with fewer than 10 million learnable parameters. This efficiency does not compromise the model's depth or performance, making it highly scalable and suitable for a wide range of applications, including those with limited computational resources.
- Incorporates a sophisticated attention mechanism by design, allowing the model to prioritize and focus on ROIs more effectively. This mechanism is further enhanced by the model's Fibonacci-inspired

structure, leading to a more nuanced and efficient analysis of medical images.

Advantages:

- The innovative utilization of a Fibonacci series-inspired structure in the FiboNeXt block enables a meticulous channel-wise concatenation, which may contribute to the richness in feature representation and propagation throughout the network.
- The integration of attention blocks allows the model to prioritize relevant information selectively and focus on specific regions of interest in the input data, enhancing its sensitivity to critical features pertinent to classification tasks.
- The strategic employment of addition blocks, akin to the ResNets, mitigates the vanishing gradient problem, facilitating deeper model training without hindering the learning process.
- FiboNeXt block, an innovative variant of ConvNeXt-like blocks, deploys concatenation and attention mechanisms creatively, yielding a distinctive architecture encompassing depth and feature selectivity.
- With less than 10 million learnable parameters, FiboNeXt ensures computational efficiency and illustrates a scalable model architecture that may be deployed in environments with varying computational resources.
- The model did not merely memorize the training data but effectively generalized its learning, ensuring reliable predictions on unseen data, as evidenced by the impressive performance metrics on the test data.
- FiboNeXt's architecture and principles could be adapted or transferred to similar medical image classification tasks, demonstrating its versatile applicability across a spectrum of use cases in medical diagnostics.

In light of these findings and advantages, FiboNeXt has proven itself to be a robust and reliable model for the classification of Alzheimer's Disease stages. It stands out in terms of high accuracy and commendable performance metrics, and the innovative underpinning architectural principles also set a promising precedent for future explorations in medical image classification.

5. Limitations and future works

While the FiboNeXt model exhibits innovative architecture and impressive performance metrics, it also has its own limitations. One of the primary limitations stems from dataset specificity. The performance demonstrated is based on a specific dataset and raises questions about its adaptability and consistency across datasets with different image resolutions, quality, or different patient demographics. This is attributed to another limitation regarding its generalization across different imaging modalities, as it was specifically trained on AD MR images. We have used two AD image datasets. Additionally, although novel, the inherent complexity of the FiboNeXt block structure may pose challenges to interpretability; this is a critical factor when considering its application in clinical settings. Given the lack of additional datasets for validation, there are also concerns about the model's ability to handle images with artifacts, noise, or non-standard capture angles, as well as potential overfitting issues.

Our future plans to improve the proposed FiboNeXt are given as follows. An important step will be to ensure its robustness by testing the model in a variety of imaging modalities, from CT to PET scans. Combining larger and more diverse data sets representing various regions and patient groups could make the model universally applicable and more robust. Since the field of medical imaging relies heavily on clinician expertise, making the FiboNeXt model's predictions and studies more explainable could bridge the gap between AI insights and clinical decision-making. Application of the model to real-world clinical scenarios will provide invaluable feedback and insights into real-time diagnostic capabilities. Given the potential of its architecture, the adaptation of FiboNeXt to other medical or neurodegenerative disorders could expand its range of applications. Finally, addressing potential overfitting with advanced regularization techniques and noise-augmented training data can refine the model's predictions. In the dynamic field of medical image analysis, the FiboNeXt model provides a solid foundation with ample opportunities to improve and expand its clinical applications.

FiboNeXt's unique architecture, inspired by the Fibonacci series, positions it as a prime candidate for tasks ranging from environmental monitoring, where it could analyze satellite imagery to track deforestation and climate change, to autonomous vehicle navigation, requiring precise, real-time interpretation of sensor data. In manufacturing, FiboNeXt's precision in identifying defects could revolutionize quality control processes, while in agriculture, it could analyze crop health from drone-captured images, enhancing yield predictions and pest management. The retail sector could benefit from its ability to analyze customer behavior optimizing store layouts and shopping experiences. Meanwhile, FiboNeXt could automate video editing and content categorization in entertainment and media, offering new interactive experiences. Additionally, its application in security and surveillance could support real-time threat detection, necessitating advancements in anomaly detection. Each of these applications would not only leverage FiboNeXt's capacity for complex pattern recognition but also necessitate model adaptations, including the development of specialized input layers for multispectral imagery, enhancements in real-time processing, and improvements in model resilience to diverse environmental conditions. This broadened application spectrum underscores FiboNeXt's potential to impact various industry sectors by harnessing deep learning for innovative solutions.

6. Conclusions

In this work, we introduced the novel FiboNeXt model, a deep-learning architecture inspired by Fibonacci sequences and tailored specifically for high-performance classification in the realm of medical imaging. The uniqueness of FiboNeXt lies in its intricate block structures, synergizing attention mechanisms with ConvNeXt-like designs, thus resulting in a robust model capable of capturing intricate patterns in AD MR images.

Our experiments, conducted within the MATLAB environment, validate the efficacy of FiboNeXt. The model's performance metrics, especially a test accuracy of over 99.60 % (99.66 % and 99.63 %) on the original image dataset, indicate its potential in real-world medical scenarios. More impressively, certain classes, such as mild and moderate demented, showed impeccable recall rates of 100 %, underscoring the model's sensitivity. Furthermore, an F1-score of 100 % for the moderate demented class signifies balanced precision and recall, emphasizing the model's reliability in classification tasks.

While the FiboNeXt model demonstrates promising results, viewing these outcomes in the context of the data it was trained and tested on is crucial. The model's high precision, recall, and F1 score across classes highlight its potential in detecting and diagnosing neurodegenerative disorders, particularly AD. However, it's important to note that while numerical results are impressive, the model's real-world clinical utility will be determined by its adaptability, interpretability, and generalizability across diverse datasets and imaging modalities.

In conclusion, the FiboNeXt model stands as a testament to the power of innovative neural network designs in addressing complex classification challenges in the medical domain. Its numerical prowess offers hope for its utility in practical scenarios, paving the way for future enhancements and broader applications.

CRediT authorship contribution statement

Turker Tuncer: Writing – original draft, Validation, Methodology,

Conceptualization. **Sengul Dogan:** Writing – original draft, Methodology, Formal analysis. **Abdulhamit Subasi:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This study is supported by Effat University with a grant number UC#9/12June2023/7.1-21(4)8.

Declaration: We corrected the grammatical errors and rephrase some sentences by using large language model (ChatGPT4o).

Appendix

The MATLAB code of the proposed FiboNeXt is given below.

i Create Layer Graph

Create the layer graph variable to contain the network layers.

```
lgraph = layerGraph();
```

ii Add Layer Branches

Add the branches of the network to the layer graph. Each branch is a linear array of layers.

```
tempLayers = [
  imageInputLayer([224 224 3],"Name","imageinput")
  convolution2dLayer([5 5],96,"Name","conv","Padding","same","Stride",[4 4])
  batchNormalizationLayer("Name","batchnorm")
  swishLayer("Name","swish")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  groupedConvolution2dLayer([3 3],1,"channel-wise","Name","groupedconv","Padding","same")
  batchNormalizationLayer("Name","batchnorm_1")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_1","Padding","same")
  swishLayer("Name","swish_1")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_2","Padding","same")
  batchNormalizationLayer("Name","batchnorm_2")
  groupedConvolution2dLayer([1 1],1,"channel-wise","Name","groupedconv_3","Padding","same")
  swishLayer("Name","swish_2")
  convolution2dLayer([1 1],96,"Name","conv_1","Padding","same")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = additionLayer(2,"Name","addition");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  groupedConvolution2dLayer([3 3],1,"channel-wise","Name","groupedconv_4","Padding","same")
  batchNormalizationLayer("Name","batchnorm_3")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_5","Padding","same")
  swishLayer("Name","swish_3")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_6","Padding","same")
  batchNormalizationLayer("Name","batchnorm_4")
  groupedConvolution2dLayer([1 1],1,"channel-wise","Name","groupedconv_7","Padding","same")
  swishLayer("Name","swish_4")
  convolution2dLayer([1 1],96,"Name","conv_2","Padding","same")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = additionLayer(2,"Name","addition_1");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  groupedConvolution2dLayer([3 3],1,"channel-wise","Name","groupedconv_8","Padding","same")
  batchNormalizationLayer("Name","batchnorm_5")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_9","Padding","same")
  swishLayer("Name","swish_5")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_10","Padding","same")
  batchNormalizationLayer("Name","batchnorm_6")
  groupedConvolution2dLayer([1 1],1,"channel-wise","Name","groupedconv_11","Padding","same")
  swishLayer("Name","swish_6")
  convolution2dLayer([1 1],96,"Name","conv_3","Padding","same")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = additionLayer(2,"Name","addition_2");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = sigmoidLayer("Name","sigmoid");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = sigmoidLayer("Name","sigmoid_1");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = multiplicationLayer(2,"Name","multiplication");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  depthConcatenationLayer(2,"Name","depthcat")
  averagePooling2dLayer([2  2],"Name","avgpool2d","Padding","same","Stride",[2 2])
  groupedConvolution2dLayer([1 1],1,"channel-wise","Name","groupedconv_12","Padding","same")
  batchNormalizationLayer("Name","batchnorm_7")
  swishLayer("Name","swish_7")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  groupedConvolution2dLayer([3 3],1,"channel-wise","Name","groupedconv_13","Padding","same")
  batchNormalizationLayer("Name","batchnorm_8")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_14","Padding","same")
  swishLayer("Name","swish_8")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_15","Padding","same")
  batchNormalizationLayer("Name","batchnorm_9")
  groupedConvolution2dLayer([1 1],1,"channel-wise","Name","groupedconv_16","Padding","same")
  swishLayer("Name","swish_9")
  convolution2dLayer([1 1],192,"Name","conv_4","Padding","same")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = additionLayer(2,"Name","addition_3");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  groupedConvolution2dLayer([3 3],1,"channel-wise","Name","groupedconv_17","Padding","same")
  batchNormalizationLayer("Name","batchnorm_10")
  groupedConvolution2dLayer([1 1],2,"channel-wise","Name","groupedconv_18","Padding","same")
  swishLayer("Name","swish_10")
```

```
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_19","Padding","same")
    batchNormalizationLayer("Name","batchnorm_11")
    groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_20","Padding","same")
    swishLayer("Name","swish_11")
    convolution2dLayer([1
1],192,"Name","conv_5","Padding","same")];
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = additionLayer(2,"Name","addition_4");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = [
    groupedConvolution2dLayer([3 3],1,"channel-wise",
"Name","groupedconv_21","Padding","same")
    batchNormalizationLayer("Name","batchnorm_12")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_22","Padding","same")
    swishLayer("Name","swish_12")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_23","Padding","same")
    batchNormalizationLayer("Name","batchnorm_13")
    groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_24","Padding","same")
    swishLayer("Name","swish_13")
    convolution2dLayer([1
1],192,"Name","conv_6","Padding","same")];
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = additionLayer(2,"Name","addition_5");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = sigmoidLayer("Name","sigmoid_2");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = sigmoidLayer("Name","sigmoid_3");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = multiplicationLayer(2,"Name","multipli
cation_1");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = [
    depthConcatenationLayer(2,"Name","depthcat_1")
    averagePooling2dLayer([2 2],"Name","avgpool2d_1",
"Padding","same","Stride",[2 2])
    groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_25","Padding","same")
    batchNormalizationLayer("Name","batchnorm_14")
    swishLayer("Name","swish_14")];
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = [
    groupedConvolution2dLayer([3 3],1,"channel-wise",
"Name","groupedconv_26","Padding","same")
    batchNormalizationLayer("Name","batchnorm_15")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_27","Padding","same")
    swishLayer("Name","swish_15")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_28","Padding","same")
    batchNormalizationLayer("Name","batchnorm_16")
    groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_29","Padding","same")
    swishLayer("Name","swish_16")
    convolution2dLayer([1
1],384,"Name","conv_7","Padding","same")];
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = additionLayer(2,"Name","addition_6");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = [
    groupedConvolution2dLayer([3 3],1,"channel-wise",
"Name","groupedconv_30","Padding","same")
    batchNormalizationLayer("Name","batchnorm_17")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_31","Padding","same")
    swishLayer("Name","swish_17")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_32","Padding","same")
    batchNormalizationLayer("Name","batchnorm_18")
    groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_33","Padding","same")
    swishLayer("Name","swish_18")
    convolution2dLayer([1
1],384,"Name","conv_8","Padding","same")];
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = additionLayer(2,"Name","addition_7");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = [
    groupedConvolution2dLayer([3 3],1,"channel-wise",
"Name","groupedconv_34","Padding","same")
    batchNormalizationLayer("Name","batchnorm_19")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_35","Padding","same")
    swishLayer("Name","swish_19")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_36","Padding","same")
    batchNormalizationLayer("Name","batchnorm_20")
    groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_37","Padding","same")
    swishLayer("Name","swish_20")
    convolution2dLayer([1
1],384,"Name","conv_9","Padding","same")];
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = additionLayer(2,"Name","addition_8");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = sigmoidLayer("Name","sigmoid_4");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = sigmoidLayer("Name","sigmoid_5");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = multiplicationLayer(2,"Name","multipli
cation_2");
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = [
    depthConcatenationLayer(2,"Name","depthcat_2")
    averagePooling2dLayer([2 2],"Name","avgpool2d_2",
"Padding","same","Stride",[2 2])
    groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_38","Padding","same")
    batchNormalizationLayer("Name","batchnorm_21")
    swishLayer("Name","swish_21")];
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = [
    groupedConvolution2dLayer([3 3],1,"channel-wise",
"Name","groupedconv_39","Padding","same")
    batchNormalizationLayer("Name","batchnorm_22")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_40","Padding","same")
    swishLayer("Name","swish_22")
    groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_41","Padding","same")
    batchNormalizationLayer("Name","batchnorm_23")
    groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_42","Padding","same")
    swishLayer("Name","swish_23")
    convolution2dLayer([1
1],768,"Name","conv_10","Padding","same")];
  lgraph = addLayers(lgraph,tempLayers);
  tempLayers = additionLayer(2,"Name","addition_9");
```

```
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  groupedConvolution2dLayer([3 3],1,"channel-wise",
"Name","groupedconv_43","Padding","same")
  batchNormalizationLayer("Name","batchnorm_24")
  groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_44","Padding","same")
  swishLayer("Name","swish_24")
  groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_45","Padding","same")
  batchNormalizationLayer("Name","batchnorm_25")
  groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_46","Padding","same")
  swishLayer("Name","swish_25")
  convolution2dLayer([1
1],768,"Name","conv_11","Padding","same")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = additionLayer(2,"Name","addition_10");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  groupedConvolution2dLayer([3 3],1,"channel-wise",
"Name","groupedconv_47","Padding","same")
  batchNormalizationLayer("Name","batchnorm_26")
  groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_48","Padding","same")
  swishLayer("Name","swish_26")
  groupedConvolution2dLayer([1 1],2,"channel-wise",
"Name","groupedconv_49","Padding","same")
  batchNormalizationLayer("Name","batchnorm_27")
  groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_50","Padding","same")
  swishLayer("Name","swish_27")
  convolution2dLayer([1
1],768,"Name","conv_12","Padding","same")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = additionLayer(2,"Name","addition_11");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = sigmoidLayer("Name","sigmoid_6");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = sigmoidLayer("Name","sigmoid_7");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = multiplicationLayer(2,"Name","multipli
cation_3");
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
  depthConcatenationLayer(2,"Name","depthcat_3")
  groupedConvolution2dLayer([1 1],1,"channel-wise",
"Name","groupedconv_51","Padding","same")
  batchNormalizationLayer("Name","batchnorm_28")
  swishLayer("Name","swish_28")
  globalAveragePooling2dLayer("Name","gapool")
  fullyConnectedLayer(256,"Name","fc")
  dropoutLayer(0.5,"Name","dropout")
  fullyConnectedLayer(4,"Name","fc_1")
  softmaxLayer("Name","softmax")
  classificationLayer("Name","classoutput")];
lgraph = addLayers(lgraph,tempLayers);
% clean up helper variable
clear tempLayers;
```

iii Connect Layer Branches

Connect all the branches of the network to create the network graph.

```
lgraph = connectLayers(lgraph,"swish","groupedconv
");
lgraph = connectLayers(lgraph,"swish","addition/
in1");
lgraph = connectLayers(lgraph,"conv_1","addition/
in2");
lgraph = connectLayers(lgraph,"addition","grouped
conv_4");
lgraph = connectLayers(lgraph,"addition","addi-
tion_1/in2");
lgraph = connectLayers(lgraph,"addition",
"sigmoid");
lgraph = connectLayers(lgraph,"conv_2","addition_1/
in1");
lgraph = connectLayers(lgraph,"addition_1","grouped
conv_8");
lgraph = connectLayers(lgraph,"addition_1","addi-
tion_2/in2");
lgraph = connectLayers(lgraph,"addition_1",
"sigmoid_1");
lgraph = connectLayers(lgraph,"conv_3","addition_2/
in1");
lgraph = connectLayers(lgraph,"addition_2","depth-
cat/in1");
lgraph = connectLayers(lgraph,"sigmoid","multipli-
cation/in1");
lgraph = connectLayers(lgraph,"sigmoid_1","multi-
plication/in2");
lgraph = connectLayers(lgraph,"multiplica-
tion","depthcat/in2");
lgraph = connectLayers(lgraph,"swish_7","grouped
conv_13");
lgraph = connectLayers(lgraph,"swish_7","addi-
tion_3/in1");
lgraph = connectLayers(lgraph,"conv_4","addition_3/
in2");
lgraph = connectLayers(lgraph,"addition_3","grouped
conv_17");
lgraph = connectLayers(lgraph,"addition_3","addi-
tion_4/in2");
lgraph = connectLayers(lgraph,"addition_3","sigmoid
_2");
lgraph = connectLayers(lgraph,"conv_5","addition_4/
in1");
lgraph = connectLayers(lgraph,"addition_4","grouped
conv_21");
lgraph = connectLayers(lgraph,"addition_4","addi-
tion_5/in2");
lgraph = connectLayers(lgraph,"addition_4","sigmoid
_3");
lgraph = connectLayers(lgraph,"conv_6","addition_5/
in1");
lgraph = connectLayers(lgraph,"addition_5","depth-
cat_1/in1");
lgraph = connectLayers(lgraph,"sigmoid_2","multi-
plication_1/in1");
lgraph = connectLayers(lgraph,"sigmoid_3","multi-
plication_1/in2");
lgraph = connectLayers(lgraph,"multiplica-
tion_1","depthcat_1/in2");
lgraph = connectLayers(lgraph,"swish_14","grouped
conv_26");
lgraph = connectLayers(lgraph,"swish_14","addi-
tion_6/in1");
lgraph = connectLayers(lgraph,"conv_7","addition_6/
in2");
lgraph = connectLayers(lgraph,"addition_6","grouped
conv_30");
lgraph = connectLayers
```

```
(lgraph,"addition_6","addition_7/in2");
   lgraph = connectLayers(lgraph,"addition_6",
"sigmoid_4");
   lgraph = connectLayers(lgraph,"conv_8","addition_7/
in1");
   lgraph = connectLayers(lgraph,"addition_7","grouped
conv_34");
   lgraph = connectLayers(lgraph,"addition_7","addi-
tion_8/in2");
   lgraph = connectLayers(lgraph,"addition_7","sigmoid
_5");
   lgraph = connectLayers(lgraph,"conv_9","addition_8/
in1");
   lgraph = connectLayers(lgraph,"addition_8","depth-
cat_2/in1");
   lgraph = connectLayers(lgraph,"sigmoid_4","multi-
plication_2/in1");
   lgraph = connectLayers(lgraph,"sigmoid_5","multi-
plication_2/in2");
   lgraph = connectLayers(lgraph,"multiplica-
tion_2","depthcat_2/in2");
   lgraph = connectLayers(lgraph,"swish_21","grouped
conv_39");
   lgraph = connectLayers(lgraph,"swish_21","addi-
tion_9/in1");
   lgraph = connectLayers(lgraph,"conv_10","addi-
tion_9/in2");
   lgraph = connectLayers(lgraph,"addition_9","grouped
conv_43");
   lgraph = connectLayers(lgraph,"addition_9","addi-
tion_10/in2");
   lgraph = connectLayers(lgraph,"addition_9",
"sigmoid_6");
   lgraph = connectLayers(lgraph,"conv_11","addi-
tion_10/in1");
   lgraph = connectLayers(lgraph,"addition_10",
"groupedconv_47");
   lgraph = connectLayers(lgraph,"addition_10","addi-
tion_11/in2");
   lgraph = connectLayers(lgraph,"addition_10",
"sigmoid_7");
   lgraph = connectLayers(lgraph,"conv_12","addi-
tion_11/in1");
   lgraph = connectLayers(lgraph,"addition_11","depth-
cat_3/in1");
   lgraph = connectLayers(lgraph,"sigmoid_6","multi-
plication_3/in1");
   lgraph = connectLayers(lgraph,"sigmoid_7","multi-
plication_3/in2");
   lgraph = connectLayers(lgraph,"multiplica-
tion_3","depthcat_3/in2");
```

iv Plot Layers

```
   plot(lgraph);
```

Data availability

We used open data from Kaggle

References

[1] R.C. Petersen, J.C. Morris, Mild cognitive impairment as a clinical entity and treatment target, Arch. Neurol. 62 (2005) 1160–1163.

[2] G. Barisano, A. Montagne, K. Kisler, J.A. Schneider, J.M. Wardlaw, B.V. Zlokovic, Blood–brain barrier link to human cognitive impairment and Alzheimer's disease, Nat. Cardiovasc. Res. 1 (2022) 108–115.

[3] E. Anderson, J.L. Durstine, Physical activity, exercise, and chronic diseases: A brief review, Sports Med. Health Sci. 1 (2019) 3–10.

[4] WHO. World Health Organization, Dementia, https://www.who.int/news-room/fact-sheets/detail/dementia. 2023.

[5] A. Subasi, Applications of Artificial Intelligence in Medical Imaging, Academic Press, 2022.

[6] J.H. Kramer, J. Jurik, J.S. Sharon, K.P. Rankin, H.J. Rosen, J.K. Johnson, et al., Distinctive neuropsychological patterns in frontotemporal dementia, semantic dementia, and Alzheimer disease, Cogn. Behav. Neurol. 16 (2003) 211–218.

[7] A.S. Pillai, B. Menon, Augmenting Neurological Disorder Prediction and Rehabilitation Using Artificial Intelligence, Academic Press, 2022.

[8] S.B. Shively, I. Horkayne-Szakaly, R.V. Jones, J.P. Kelly, R.C. Armstrong, D.P. Perl, Characterisation of interface astroglial scarring in the human brain after blast exposure: a post-mortem case series, The Lancet Neurology. 15 (2016) 944–953.

[9] G. Waldemar, B. Dubois, M. Emre, J. Georges, I. McKeith, M. Rossor, et al., Recommendations for the diagnosis and management of Alzheimer's disease and other disorders associated with dementia: EFNS guideline, Eur. J. Neurol. 14 (2007) e1–e26.

[10] B. Ibach, E. Haen, Acetylcholinesterase inhibition in Alzheimer's Disease, Curr. Pharm. Des. 10 (2004) 231–251.

[11] M.W. Weiner, D.P. Veitch, P.S. Aisen, L.A. Beckett, N.J. Cairns, J. Cedarbaum, et al., 2014 Update of the Alzheimer's Disease Neuroimaging Initiative: a review of papers published since its inception, Alzheimers Dement. 11 (2015) e1–e120.

[12] M. Baygin, O. Yaman, T. Tuncer, S. Dogan, P.D. Barua, U.R. Acharya, Automated accurate schizophrenia detection system using Collatz pattern technique with EEG signals, Biomed. Signal Process. Control 70 (2021) 102936.

[13] S.G. Kobat, N. Baygin, E. Yusufoglu, M. Baygin, P.D. Barua, S. Dogan, et al., Automated diabetic retinopathy detection using horizontal and vertical patch division-based pre-trained DenseNET with digital fundus images, Diagnostics. 12 (2022) 1975.

[14] P.D. Barua, N.F. Muhammad Gowdh, K. Rahmat, N. Ramli, W.L. Ng, W.Y. Chan, et al., Automatic COVID-19 detection using exemplar hybrid deep features with X-ray images, Int. J. Environ. Res. Public Health 18 (2021) 8052.

[15] D. Barh, Artificial Intelligence in Precision Health: From Concept to Applications, Academic Press, 2020.

[16] T. Illakiya, K. Ramamurthy, M. Siddharth, R. Mishra, A. Udainiya, AHANet: Adaptive Hybrid Attention Network for Alzheimer's Disease Classification Using Brain Magnetic Resonance Imaging, Bioengineering 10 (2023) 714.

[17] L.J.C. de Mendonça, R.J. Ferrari, I. AsDN, Alzheimer's disease classification based on graph kernel SVMs constructed with 3D texture features extracted from MR images, Expert Syst. Appl. 211 (2023) 118633.

[18] M. Sethi, S. Rani, A. Singh, J.L.V. Mazón, A CAD System for Alzheimer's Disease Classification Using Neuroimaging MRI 2D Slices, Comput. Math. Methods Med. 2022 (2022).

[19] B.M. Cobbinah, C. Sorg, Q. Yang, A. Ternblom, C. Zheng, W. Han, et al., Reducing variations in multi-center Alzheimer's disease classification with convolutional adversarial autoencoder, Med. Image Anal. 82 (2022) 102585.

[20] B. Yan, Y. Li, L. Li, X. Yang, T.-q. Li, G. Yang, et al., Quantifying the impact of Pyramid Squeeze Attention mechanism and filtering approaches on Alzheimer's disease classification, Comput. Biol. Med. 148 (2022) 105944.

[21] S. Dogan, M. Baygin, B. Tasci, H.W. Loh, P.D. Barua, T. Tuncer, et al., Primate brain pattern-based automated Alzheimer's disease detection model using EEG signals, Cogn. Neurodyn. 17 (2023) 647–659.

[22] E. Kaplan, S. Dogan, T. Tuncer, M. Baygin, E. Altunisik, Feed-forward LPQNet based automatic alzheimer's disease detection model, Comput. Biol. Med. 137 (2021) 104828.

[23] E. Kaplan, M. Baygin, P.D. Barua, S. Dogan, T. Tuncer, E. Altunisik, et al., ExHiF: Alzheimer's disease detection using exemplar histogram-based features with CT and MR images, Med. Eng. Phys. 115 (2023) 103971.

[24] S.E. Sorour, A.A. Abd El-Mageed, K.M. Albarrak, A.K. Alnaim, A.A. Wafa, E. El-Shafeiy, Classification of Alzheimer's disease using MRI data based on Deep Learning Techniques, Journal of King Saud University-Computer and Information Sciences. 36 (2024) 101940.

[25] Akan T, Alp S, Bhuiyanb MA. Vision Transformers and Bi-LSTM for Alzheimer's Disease Diagnosis from 3D MRI. arXiv preprint arXiv:240103132. 2024.

[26] A. Assmi, K. Elhabyb, A. Benba, A. Jilbab, Alzheimer's disease classification: a comprehensive study, Multimed. Tools Appl. 1–24 (2024).

[27] P. Goyal, R. Rani, K. Singh, A multilayered framework for diagnosis and classification of Alzheimer's disease using transfer learned Alexnet and LSTM, Neural Comput. & Applic. 36 (2024) 3777–3801.

[28] D.A. Arafa, H.-E.-D. Moustafa, H.A. Ali, A.M. Ali-Eldin, S.F. Saraya, A deep learning framework for early diagnosis of Alzheimer's disease on MRI images, Multimed. Tools Appl. 83 (2024) 3767–3799.

[29] V. Adarsh, G. Gangadharan, U. Fiore, P. Zanetti, Multimodal classification of Alzheimer's disease and mild cognitive impairment using custom MKSCDDL kernel over CNN with transparent decision-making for explainable diagnosis, Sci. Rep. 14 (2024) 1774.

[30] T. Mahmud, K. Barua, S.U. Habiba, N. Sharmen, M.S. Hossain, K. Andersson, An Explainable AI Paradigm for Alzheimer's Diagnosis Using Deep Transfer Learning, Diagnostics. 14 (2024) 345.

[31] T. Prasath, V. Sumathi, Pipelined deep learning architecture for the detection of Alzheimer's disease, Biomed. Signal Process. Control 87 (2024) 105442.

[32] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:201011929. 2020.

[33] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF international conference on computer vision2021. p. 10012-22.
[34] Uraninjo. Augmented Alzheimer MRI Dataset, https://www.kaggle.com/datasets/uraninjo/augmented-alzheimer-mri-dataset. 2022.
[35] S. Szabó, I.J. Holb, V.É. Abriha-Molnár, G. Szatmári, S.K. Singh, D. Abriha, Classification assessment tool: a program to measure the uncertainty of classification models in terms of class-level metrics, Appl. Soft Comput. 155 (2024) 111468.
[36] E. Casas, L. Ramos, E. Bendek, F. Rivas-Echeverría, Assessing the effectiveness of YOLO architectures for smoke and wildfire detection, IEEE Access (2023).
[37] M.S.H. Talukder, A.K. Sarkar, Nutrients deficiency diagnosis of rice crop by weighted average ensemble learning, Smart Agric. Technol. 4 (2023) 100155.
[38] J. Deng, Z. Yang, H. Wang, I. Ojima, D. Samaras, F. Wang, A systematic study of key elements underlying molecular property prediction, Nat. Commun. 14 (2023) 6395.
[39] K. Rezaee, M.R. Khosravi, M. Sabri, K. Al-Qawasmi, A Hybrid Deep Cascade-ResNet Model for Detecting Alzheimer's Stages in MR Images, in: 2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI): IEEE, 2022, pp. 1–6.
[40] S. Jha, A.D. Mahajan, P.G. Thenraj, S. Karuppasamy, S. Lakshminarayanan, Comparative Evaluation of Transfer Learning Models on Dementia Prediction, in: 2022 6th International Conference on Electronics, Communication and Aerospace Technology: IEEE, 2022, pp. 1520–1526.
[41] R. Anitha, D.B. Dasari, P. Vivek, N.M.L. Kakarla, M.S. Kumar, A novel adaptive dual swarm intelligence based image quality enhancement approach with the modified SegNet-RBM-based Alzheimer Segmentation and classification, Multimed. Tools Appl. 1–28 (2023).
[42] O. Elgendy, A.B. Nassif, Alzheimer Detection using Different Deep Learning Methods with MRI Images, in: 2023 Advances in Science and Engineering Technology International Conferences (ASET): IEEE, 2023, pp. 1–6.
[43] Taspinar YS. Classification of Alzheimer MRI Images with Machine Learning Methods Using Deep Features. International Conference on Advanced Technologies (ICAT'22). Van,Turkey2022.