

# **FINE-TUNING ALZHEIMER'S DISEASE DIAGNOSIS USING IMPROVED WAVELET CONVOLUTION NEURAL NETWORK [IWCNN]**

*Report submitted to the SASTRA Deemed to be University  
as the requirement for the course  
CSE300 : MINI PROJECT*

*Submitted by*  
**Bhuvana B**  
(Reg No : 226003022, B.Tech-CSE)  
**Harini J**  
(Reg No : 226003170, B.Tech-CSE)  
**Chitte Renuka Reddy**  
(Reg No : 226003034, B.Tech-CSE)

**May 2025**



**SRINIVASA RAMANUJAN CENTRE**

**KUMBAKONAM, TAMIL NADU INDIA - 612001**



**SASTRA**  
ENGINEERING • MANAGEMENT • LAW • SCIENCES • HUMANITIES • EDUCATION  
DEEMED TO BE UNIVERSITY  
(U/S 3 of the UGC Act, 1956)



THINK MERIT | THINK TRANSPARENCY | THINK SASTRA  
T H A N J A V U R | K U M B A K O N A M | C H E N N A I

### SRINIVASA RAMANUJAN CENTRE

KUMBAKONAM, TAMIL NADU INDIA – 612001

#### Bonafide Certificate

This is to certify that the report titled “**Fine-Tuning Alzheimer's Disease Diagnosis Using Improved Wavelet Convolution Neural Network [IWCNN]**” submitted as a requirement for the course, CSE300 : **MINI PROJECT** for B.Tech. is a bonafide record of the work done by **Ms B. Bhuvana (226003022, B.Tech-CSE)**, **Ms J. Harini (226003170, B.Tech-CSE)**, **Ms Chitte Renuk Reddy (226003034, B.Tech-CSE)** during the academic year 2024-25, in the Srinivasa Ramanujan Center, under my supervision.

#### **Signature of Project Supervisor :**

**Name with Affiliation** : Dr Thanuja R/AP-II/CSE/SRC/SASTRA

**Date** :

Mini Project Viva voce held on \_\_\_\_\_

**Examiner 1**

**Examiner 2**

## Acknowledgements

We would like to express our sincere gratitude to our Honorable Chancellor, **Prof. R. Sethuraman**, for providing us with the opportunity and the necessary infrastructure to carry out this project as part of our curriculum

We would like to thank our Honorable Vice-Chancellor, **Dr. S. Vaidhyasubramaniam, and Dr. S. Swaminathan**, Dean of Planning & Development, for their constant encouragement and strategic support throughout our college journey. We also extend our sincere gratitude to **Dr. R. Chandramouli, Registrar**, SASTRA Deemed to be University, for providing us with the opportunity to pursue this project.

We extend our heartfelt thanks to **Dr.V. Ramaswamy**, Dean, **Dr. A. Alli Rani**, Associate Dean, Srinivasa Ramanujan Centre, SASTRA Deemed to be University. We exhibit our pleasure in expressing our thanks to **Dr. V. Kalaichelvi**, Associate Professor, Department of Computer Science and Engineering, Srinivasa Ramanujan Centre, for her encouragement during our project work.

Our guide **Dr. Thanuja R., Assistant Professor -II** Department of Computer Science and Engineering, Srinivasa Ramanujan Centre was the driving force behind this whole idea from the start. Her deep insight in the field and invaluable suggestions helped us in making progress throughout our project work.

We also thank the project review panel members **Dr. S. Meganathan** and **Mrs. D. Rekha** for their valuable comments and insights which made this project better.

We would like to extend our gratitude to **Dr. M. Vanitha, Assistant Professor -II & Project Co-ordinator**, Department of Computer Science and Engineering, Srinivasa Ramanujan Centre for organizing and helping us in the completion of the project.

We gratefully acknowledge all the contributions and encouragement from my family and friends resulting in the successful completion of this project. We thank you all for giving us the opportunity to showcase our skills through this project.

## List Of Figures

Figure No.	Title	Page No.
1.1	Architecture diagram	4
1.2	Image before preprocessing	6
1.3	Image after preprocessing	6
1.4	Grad-CAM Heatmap Visualization	10
1.5	Prediction using Grad CAM	17
5.1	Class distribution	32
5.2	Confusion matrix of CNN	33
5.3	Confusion matrix of DCNN	33
5.4	Confusion matrix of WCNN	33
5.5	CNN training and validation performance curves	34
5.6	DCNN training and validation performance curves	34
5.7	WCNN training and validation performance curves	34
5.8	Class Prediction using WCNN	35
5.9	WCNN Precision, Recall & F1-Score per Class	36
5.10	ROC curve of WCNN	36

## **List Of Tables**

<b>Table No.</b>	<b>Title</b>	<b>Page No.</b>
1.1	Dataset Split	6
3.1	Accuracies obtained for different models	16

## **Abbreviations**

<b>AD</b>	Alzheimer's disease
<b>CNN</b>	Convolutional Neural Networks
<b>DCNN</b>	Deep convolutional neural network
<b>WCNN</b>	Wavelet Convolutional Neural Network
<b>SMOTE</b>	Synthetic Minority Oversampling Technique
<b>SE</b>	Squeeze-and-Excitation
<b>Grad-CAM</b>	Gradient-weighted Class Activation Mapping

## **Abstract**

Alzheimer's disease (AD) is a progressive neurodegenerative disorder marked by memory loss and cognitive decline. Traditional diagnostics are often subjective, prompting the need for automated solutions. This project applies Discrete Wavelet Transform (DWT) to MRI scans, enhancing structural features. These are used to train an Improved Wavelet CNN for multi-class classification (AD, NC, EMCI, LMCI). Synthetic Minority Oversampling Technique (SMOTE) handles class imbalance by generating synthetic samples, while Gradient-weighted Class Activation Mapping (Grad-CAM) improves model interpretability by highlighting key brain regions. The proposed WCNN achieves 95.62% test accuracy, outperforming baseline and deeper CNN models. This approach offers a reliable and interpretable tool for early Alzheimer's detection.

**KEY WORDS:** Alzheimer's Disease, Deep Learning, Improved Wavelet Convolutional Neural Network, SMOTE, Grad-CAM, MRI Classification 4.

## Table of Contents

<b>Title No</b>	<b>Page No.</b>
Bonafide Certificate	ii
Acknowledgements	iii
List Of Figures	iv
List Of Tables	v
Abbreviations	vi
Abstract	vii
1. Summary of the base paper	1
2. Merits and Demerits of the base paper	12
3. Proposed Work	14
4. Source Code	19
5. Snapshots	32
6. Conclusion and Future Plans	37
7. References	39
8. Appendix-Base Paper	41

# CHAPTER 1

## SUMMARY OF THE BASE PAPER

**Title:** Advancing early diagnosis of Alzheimer's disease with next-generation deep learning methods

**Journal Name:** Biomedical Signal Processing and Control

**Publisher:** Elsevier

**Authors:** Cuneyt Ozdemir, Yahya Dogan

**Volume:** Volume 96

**Publication Date:** July 2024

**Indexing:** SCI-E Q1

### 1.1 Introduction

Alzheimer's disease (AD) is a degenerative neurological disorder that predominantly impacts older adults, causing a steady decline in memory, cognitive function, and behaviour. Since there is currently no definitive cure, early and accurate diagnosis is essential for slowing disease progression and improving patient outcomes. The research paper titled "Advancing Early Diagnosis of Alzheimer's Disease with Next-Generation Deep Learning Methods" by Cuneyt Ozdemir and Yahya Dogan proposes a deep learning-based model that effectively classifies various stages of Alzheimer's using MRI scans. Importantly, the approach is designed to be computationally efficient, supporting its practical use in clinical environments.

### 1.2 Problem Statement

- Alzheimer's disease remains challenging to detect in its early stages, limiting timely intervention and effective treatment. Existing diagnostic methods, including traditional assessments and AI-based models, often struggle with accuracy and reliability.
- One major challenge is the imbalance in MRI datasets, where fewer cases of severe Alzheimer's leads to biased models that underperform on underrepresented categories. Additionally, conventional CNNs primarily extract spatial features but struggle to capture crucial frequency-based patterns, reducing classification accuracy. Moreover, deep learning models often function as black boxes, offering little transparency in their decision-making process, which makes clinical adoption difficult.

### 1.3 Literature Survey

The 2024 study titled "Pipelined Deep Learning Architecture for the Detection of Alzheimer's Disease", published in *Biomedical Signal Processing and Control* (Science Direct) by T. Prasath and V. Sumathi, introduces a Pipelined LeNet (PLN) architecture. This model integrates image fusion and MRI preprocessing techniques to achieve an impressive classification accuracy of 99.5%, with a rapid execution time of just 0.65 milliseconds, highlighting its performance and efficiency. However, a limitation of this approach is its limited evaluation on real-world and diverse datasets, which raises concerns about its generalizability and robustness in practical scenarios.

In another 2024 publication from the journal *Measurement* (Science Direct), titled "Machine and Deep Learning Approaches for Alzheimer's Disease Detection Using Magnetic Resonance Images: An Updated Review", authors M. Menagadevi, Somasundaram Devaraj, and colleagues provide a comprehensive review of machine learning and deep learning approaches for Alzheimer's detection using MRI data. Covering literature from 2013 to 2023, the article discusses key techniques such as CNN, SVM, transfer learning, and preprocessing strategies like histogram equalization and contrast stretching. While the review is valuable for its breadth and informative synthesis, it lacks experimental validation, which limits its practical application in real-world diagnosis.

A customized dynamically ensemble CNN (PDECNN) model with an integrated attention mechanism is presented in the 2024 paper "Bio-Inspired Deep Learning – Personalized Ensemble Alzheimer's Diagnosis Model for Mental Well-Being" by Ajmeera Kiran, Mahmood Alsaadi, and colleagues, which was published in *SLAS Technology* (Science Direct). This model, which takes its cues from bio-intelligence, adjusts to the different ways that people's brains deteriorate in an effort to improve mental health. It shows an improvement in categorization accuracy of 4% to 11%. However, scalability and practical application are hindered by its high computational complexity.

Published in 2023 in *IEEE Transactions on Biomedical Engineering*, the study titled "Fine-Grained and Multimodal Classification of Alzheimer's Disease With Wavelet Convolutional Unit Network" by Jinyu Wen, Yang Li, Meic Fang, and colleagues presents the Wavelet Convolutional Unit Network (WCUN). By incorporating multi-scale wavelet decomposition and diffusion tensor imaging (DTI), the model achieves a fine-grained classification accuracy of 97.89%. Although this demonstrates state-of-the-art performance, the approach is computationally intensive, posing difficulties for deployment in resource-limited environments.

"Detection of Alzheimer's Disease Using Deep Learning Models: A Systematic Literature Review" by Eqtidar M. Mohammed, Ahmed N. Fakhrueldeen, and associates, published in *Informatics in Medicine Unlocked* in 2024, reviews more than 45 studies on deep learning-based Alzheimer's disease detection. In addition to exploring RNNs and Deep Belief Networks, it assesses a number of CNN architectures, such as ResNet, AlexNet, GoogleNet, and EfficientNet-B7. Although the study offers a thorough summary

of the area, it noticeably ignores non-imaging indicators, which are becoming more and more important for both early diagnosis and thorough assessment of Alzheimer's disease.

A deep learning framework that combines CNN, transfer learning, and multimodal fusion (MRI + PET) is proposed in the 2024 paper "Deep Learning-based Alzheimer's Disease Classification Using MRI Images" by R.S. Karthik, M. Sharma, and their colleagues, which was published in Neurocomputing. The model is excellent at extracting information from several modalities and has a high classification accuracy of 98.26%. Its need on sizable labeled datasets for training, which can be a barrier in real-world situations where annotated data is limited, is a significant drawback despite its efficacy.

L. Chen, H. Wang, and colleagues' 2024 paper "Alzheimer's Disease Diagnosis Using 3D Convolutional Neural Networks and MRI Scans" was published in Medical Image Analysis. It uses pretrained ResNet-50 models, data augmentation, and 3D Convolutional Neural Networks (3D-CNN) to diagnose Alzheimer's disease. Improved diagnostic accuracy results from this method's efficient handling of 3D brain scan data and robust feature extraction. Its high processing power requirements, however, might make it unusable in settings with constrained computing power.

N. Verma, A. Gupta, and colleagues' 2023 work "Hybrid Deep Learning Model for Early Detection of Alzheimer's Disease" in IEEE Access presents a hybrid architecture that combines CNN and RNN models with an attention mechanism. By extracting both temporal and spatial information from medical data, this model seeks to improve early detection and diagnostic results. Its primary flaw is that it requires a lot of memory and training time, which may prevent it from being used in actual clinical situations.

## 1.4 Architecture Diagram

In Fig. 1.1, the architecture diagram illustrates a pipeline for Alzheimer's disease classification using deep learning models. It starts with a dataset of MRI images, which undergoes data preprocessing including image resizing, balancing, and normalization. The data is then split into training, validation, and test sets. Multiple models—CNN, DCNN, and WCNN—are trained, and the best-performing model is selected. The chosen model is optimized, and its output is evaluated using metrics like accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix. Finally, Grad-CAM is used for visual interpretation of the model's decisions.

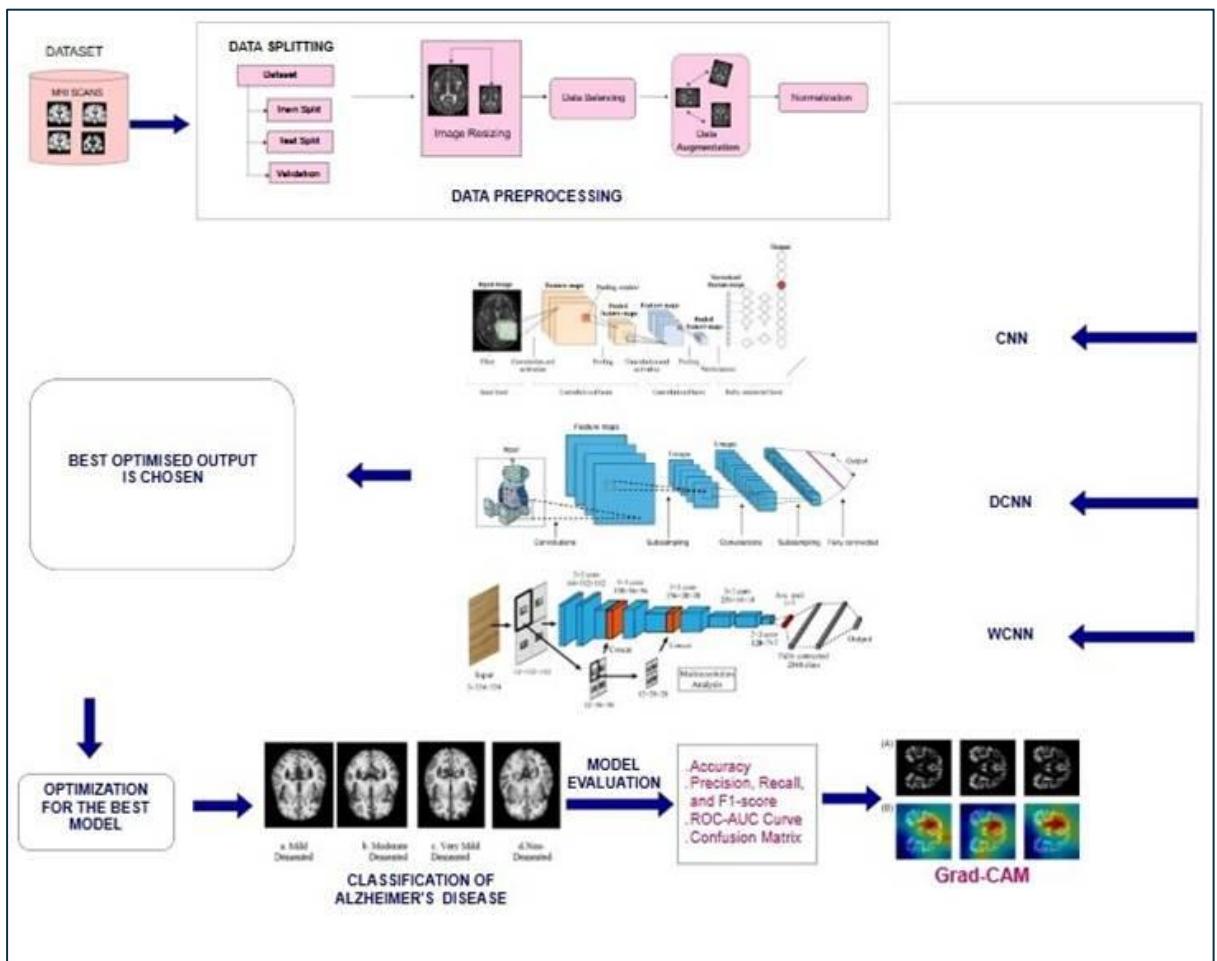


Fig. 1.1 Architecture diagram

## 1.5 Hardware and Software Requirements

### Hardware requirements

The hardware configuration required to support the project includes a processor such as Intel Core i3/i5 or AMD Ryzen 3/5. A minimum of 8 GB RAM is recommended to ensure efficient performance. The system should have at least 128 GB of SSD storage. A dedicated GPU is not necessary on the local machine, as cloud-based GPUs, such as those

provided by Kaggle, are utilized for computation. Additionally, a stable internet connection is essential to access and operate on cloud platforms like Kaggle effectively.

## Software Requirements

The project was developed and executed using Kaggle Notebooks, a cloud-based platform that supports all stages including preprocessing, training, evaluation, and prediction. Kaggle provides free access to powerful GPUs such as the NVIDIA Tesla T4 or P100, which significantly accelerates deep learning tasks. The implementation was carried out using Python version 3.7 or above. The primary development environment was the Jupyter Notebook, which is integrated within the Kaggle platform. Various Python libraries and tools were employed to accomplish different stages of the project. For data handling, NumPy and Pandas were used. Visualization was carried out using Matplotlib and Seaborn. Image processing tasks were handled with OpenCV. Machine learning models were implemented using Scikit-learn, while deep learning models were developed using TensorFlow and/or Keras or PyTorch. Evaluation of models was conducted using the `sklearn.metrics` module.

## 1.6 Modules and descriptions

### 1.6.1 Dataset Description

The Alzheimer's MRI Disease Classification dataset is a crucial resource for advancing research in medical imaging and neurodegenerative disease diagnosis. It comprises a total of 6,400 T1-weighted brain MRI images, divided into 5,120 training images and 1,280 testing images. Each image is categorized into one of four classes: Mild Demented (label '0'), Moderate Demented (label '1'), Non-Demented (label '2'), and Very Mild Demented (label '3'). The dataset was obtained from the publicly available source on Kaggle:<https://www.kaggle.com/datasets/borhanittrash/alzheimer-mri-disease-classification-dataset>.

### 1.6.2 Data Preprocessing

#### 1. Dataset Loading and Structure

The dataset used for this project was provided in. parquet format and included separate files for training and testing. Each file contained MRI scan data stored as byte-encoded images along with corresponding class labels. After loading the datasets into data frames using pandas, the structure was examined. Each row consisted of an image and a label, where the labels represented the stages of Alzheimer's disease as numerical classes: 0 (Non-Demented), 1 (Very Mild Demented), 2 (Mild Demented), and 3 (Moderate Demented).

## 2. Image Decoding and Processing

The image data in the dataset was stored as raw byte streams. Each image was decoded using the Python Imaging Library (PIL) to convert the byte format into a viewable image. Once decoded, the images were uniformly resized to 128 by 128 pixels to ensure consistency across all samples and to reduce computational overhead. Following resizing, pixel values were normalized to the range [0, 1] by dividing by 255. This normalization step helps in accelerating model convergence and improving training stability. All processed images were then converted into NumPy arrays to facilitate input into deep learning models.

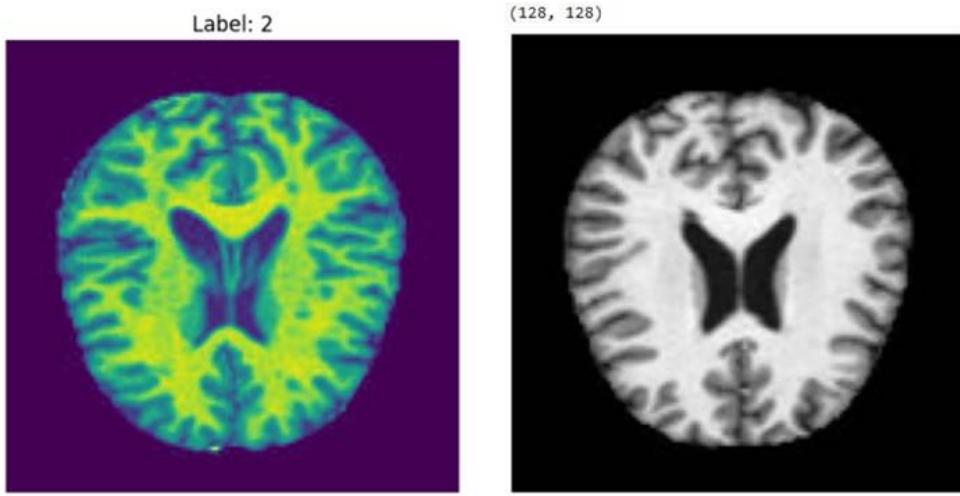


Fig. 1.2 Image before preprocessing

Fig. 1.3 Image after preprocessing

## 3. Train-Validation Split

After preprocessing the images, the dataset was split into training and validation subsets using an 80-20 ratio. A stratified split was employed to maintain the original class distribution in both subsets. This ensured that each class was proportionally represented in the validation data, providing a more reliable estimate of model performance.

Class	Training Set	Validation Set	Test Set
Mild Demented (0)	579 images	145 images	172 images
Moderate Demented (1)	39 images	10 images	15 images
Non-Demented (2)	2053 images	513 images	634 images
Very Mild Demented (3)	1425 images	356 images	459 images

Table 1.1 Dataset Split

#### 4. Addressing Class Imbalance with SMOTE

To address the significant class imbalance in the training data, the Synthetic Minority Oversampling Technique (SMOTE) was applied. Prior to applying SMOTE, each image was flattened from a 2D array ( $128 \times 128$ ) to a 1D vector, as SMOTE requires tabular input. The algorithm then generated synthetic samples for the minority classes to match the count of the majority classes. After oversampling, the data was reshaped back to its original 2D format. This balancing step helped in mitigating bias toward overrepresented classes and promoted better generalization during training.

SMOTE is a widely used data augmentation method designed to address the issue of imbalanced datasets in machine learning. When one class (the majority class) significantly outnumbers the other (the minority class), it can lead to biased models that perform poorly on the minority class. SMOTE works by generating synthetic samples for the minority class to balance the dataset and improve model performance. The SMOTE process begins by selecting a random sample from the minority class as a template. The algorithm then identifies the k-nearest neighbors of this template within the feature space. For each neighbor, a synthetic sample is created by linearly interpolating between the template sample and its selected neighbor. The formula for generating a synthetic sample is as follows:

$$P_{ij} = x_i + \text{rand}(0,1) * (x_{ij} - x_i) \quad (1.1)$$

where  $x_i$  is the template sample,  $x_{ij}$  is one of its nearest neighbors,  $\text{rand}(0,1)$  is a random number between 0 and 1,  $P_{ij}$  and is the generated synthetic sample.

This interpolation is repeated until the desired number of synthetic samples is created. SMOTE operates in the feature space, which means it generates new data points based on combinations of existing ones, rather than simply duplicating the data. This allows for the creation of more diverse and realistic synthetic examples, improving the ability of machine learning models to learn better decision boundaries. In the original training set used in this study, the class distribution was highly skewed: Mild\_Demented (579 images), Moderate\_Demented (39 images), Non\_Demented (2053 images), and Very\_Mild\_Demented (1425 images), leading to an imbalanced total of 4096 training images and 1024 validation images. After applying SMOTE, the training set becomes balanced with a total of 8212 images and corresponding labels, effectively equalizing class representation and enabling the model to better learn decision boundaries.

#### 5. Label Encoding

The categorical labels were transformed using one-hot encoding to prepare them for multi-class classification. This encoding converted each class label into a binary vector of length four, where only the position corresponding to the class was set to one and the rest were zeros. One-hot encoding was applied to the training, validation, and test labels to ensure consistency across all stages of model development.

## **6. Final Reshaping of Image Data**

As a final step, an additional dimension was added to the image data to represent the grayscale channel, transforming the image shape from (128, 128) to (128, 128, 1). This was necessary because convolutional neural networks typically expect a channel dimension in their input. By including this final reshaping, the data was made fully compatible with the input layer of CNN-based models.

### **1.6.3 Synopsis of the algorithms employed**

#### **Convolutional Neural Network (CNN)**

A **Convolutional Neural Network (CNN)**, also known as a ConvNet, is a specialized deep learning architecture designed to process grid-structured data such as images. CNNs are particularly well-suited for a wide range of computer vision applications, including image classification, object detection, and segmentation. Unlike traditional machine learning techniques that require handcrafted feature extraction, CNNs automatically learn hierarchical representations from raw input data. They extract features ranging from low-level patterns like edges and textures to high-level structures such as shapes and objects.

The core architecture of a CNN typically consists of convolutional layers, activation functions (e.g., ReLU), pooling layers, and fully connected layers. These components work in unison to detect patterns, reduce spatial dimensions, and perform final classification. This design enables CNNs to effectively manage spatial variations in data and scale well to large datasets, making them a powerful tool not only in image analysis but also in fields like audio processing and natural language processing.

However, CNNs come with certain limitations, including the need for large labeled datasets and substantial computational resources (often GPUs) due to their complex architecture. While they excel at image-based tasks, they are not as effective for sequential data, where Recurrent Neural Networks (RNNs) are more suitable. Nevertheless, CNNs remain a foundational technique in deep learning due to their versatility and high accuracy in pattern recognition tasks.

### **1.6.4 Feature Extraction**

In this study, feature extraction is enhanced through the integration of two advanced techniques: Squeeze-and-Excitation (SE) blocks and Avg-TopK Pooling. These components are incorporated into both the Convolutional Neural Network (CNN) and Deep Convolutional Neural Network (DCNN) architectures to improve the network's ability to focus on salient and class-discriminative features.

## Squeeze-and-Excitation (SE) Blocks

Squeeze-and-Excitation (SE) blocks are attention-based mechanisms designed to improve the representational power of convolutional neural networks by explicitly modeling the interdependencies between feature channels. Each SE block operates in three main stages:

- **Squeeze:** The spatial dimensions of the feature maps are compressed using Global Average Pooling, resulting in a single scalar value per channel. This captures the global contextual information of each feature map.
- **Excitation:** The squeezed outputs are passed through two fully connected (Dense) layers. The first layer reduces the channel dimension (as a bottleneck) using ReLU activation, and the second restores it back using a sigmoid activation function. This produces a set of attention weights between 0 and 1 for each channel.
- **Scaling:** The original feature maps are then scaled (channel-wise multiplication) by these learned weights, enabling the network to emphasize more relevant features and suppress less informative ones.

The SE block thus recalibrates the feature maps adaptively, helping the model to attend to the most informative regions, which is particularly useful in medical imaging where disease-specific cues can be subtle.

## Avg-TopK Pooling Layer

Traditional pooling methods such as max pooling or average pooling can lead to loss of critical information. To address this, an Avg-TopK Pooling layer is introduced in both CNN and DCNN architectures. This custom pooling mechanism selects the top-k highest activations across the spatial dimensions of a feature map and computes their average. By focusing on the top k responses rather than all activations, Avg-TopK Pooling strikes a balance between sensitivity (as in max pooling) and robustness (as in average pooling). It ensures that the most salient features—those most indicative of Alzheimer's pathology are preserved during downsampling, thereby improving classification performance.

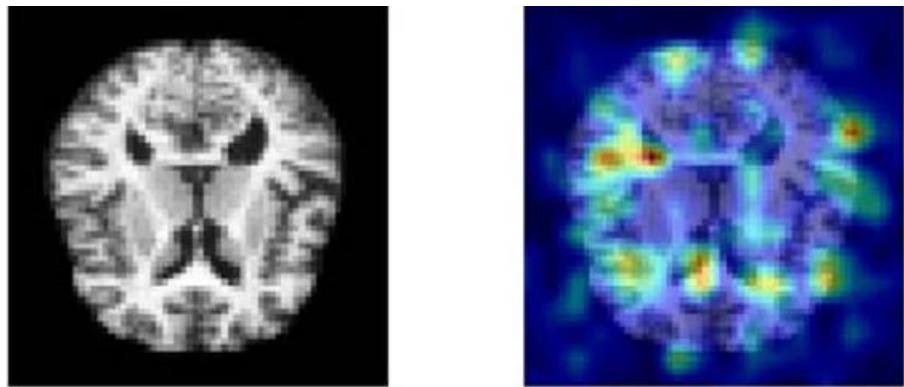
### 1.6.5 Visualization

#### Grad-CAM

**Gradient-weighted Class Activation Mapping (Grad-CAM)** is a technique developed to help interpret and visualize the decision-making process of convolutional neural networks (CNNs), particularly in image classification tasks. The core idea behind Grad-CAM is to generate a heatmap that highlights the regions of an input image that are most important for the model's prediction. This is achieved by leveraging the gradients of the target class (the class the model is predicting) with respect to the feature maps of the final convolutional layer.

To compute the Grad-CAM, the first step is to back propagate the gradients of the target class through the network and into the final convolutional layer, where the feature maps hold high-level spatial information. These gradients provide insight into how much each feature map contributes to the decision-making process for that specific class. The gradients are then pooled across all the spatial locations of the feature maps to generate a set of weights that represent the importance of each feature map. These weights are then used to create a weighted combination of the feature maps, resulting in a coarse heatmap that indicates which regions of the image were most influential for the model's classification.

Finally, this heatmap is overlaid on the original image to provide a visual explanation of the model's focus areas when making a prediction. The warmer regions of the heatmap (often red or yellow) indicate areas where the model paid more attention, while cooler regions (blue or green) suggest areas with less significance. Grad-CAM offers an intuitive way to understand what features or parts of the image are being used by the network to make its decisions, enhancing the interpretability of deep learning models. This method is particularly useful for detecting potential biases, improving model transparency, and gaining trust in automated systems, especially in applications like medical imaging or autonomous driving where interpretability is critical.



[Fig. 1.4](#) Grad-CAM Heatmap Visualization

In [Fig. 1.4](#) the Grad-CAM heatmap highlights the regions within the brain MRI scan that the Wavelet-CNN model considered most important for classifying this image as Very Mild Demented. In this specific example, the activated areas (in red and yellow) are prominently distributed across the left and right cortical regions, with additional focus around the parietal lobes and near ventricular boundaries—zones commonly linked to early neurodegenerative changes in Alzheimer's disease. This suggests that the model is attending to anatomically and clinically relevant features, rather than arbitrary patterns. Since the true label and predicted label match (Very\_Mild\_Demented), and the highlighted regions align with known biomarkers of early dementia, this reinforces both the accuracy and interpretability of the model's decision-making process.

### **1.6.6 Result and discussion**

The proposed lightweight CNN model, enhanced with SE blocks, the Avg-TopK pooling layer, and SMOTE for class balancing, achieved state-of-the-art performance in Alzheimer's disease classification, reaching a remarkable 99.84% test accuracy. Compared to 14 well-known transfer learning models, the proposed architecture not only outperformed high-capacity models like ResNet101V2 and DenseNet201 in terms of accuracy but also demonstrated significantly lower computational complexity. Ablation studies validated the individual and combined effectiveness of SE blocks, Avg-TopK pooling, and SMOTE, showing progressive accuracy improvements from 96.9% to 99.84%. Grad-CAM visualization confirmed that the model focused primarily on cortical regions—clinically relevant areas affected in Alzheimer's—highlighting the model's interpretability and diagnostic relevance. These results demonstrate the proposed model's capability as a reliable and efficient tool for early detection of Alzheimer's disease, offering strong potential for integration into clinical settings.

## CHAPTER 2

### MERITS AND DEMERITS OF THE BASE PAPER

#### Merits

1. **High Diagnostic Accuracy:** The base CNN model demonstrates strong classification performance on the Alzheimer's dataset, reaching near state-of-the-art accuracy.
2. **Lightweight Architecture:** Unlike conventional transfer learning models with millions of parameters, the custom CNN model is computationally efficient and suitable for deployment on low-resource systems.
3. **Effective Use of SMOTE:** The integration of the SMOTE technique successfully addresses class imbalance, a common issue in medical imaging datasets, resulting in improved generalization and fair performance across all classes.
4. **Innovative Pooling Strategy:** The use of Avg-TopK pooling contributes to better feature representation by selectively averaging the top K activations, which balances the trade-off between max and average pooling.
5. **Interpretability with Grad-CAM:** The implementation of Grad-CAM adds transparency to the model's predictions by highlighting the specific brain regions involved in classification, confirming alignment with medically relevant features.
6. **Balanced Performance Metrics:** The model not only delivers high accuracy but also demonstrates strong precision, recall, and F1-scores across all four Alzheimer's stages, including the underrepresented Moderate Demented class.

#### Demerits

1. **CNNs Only Capture Spatial Features, Not Frequency Information:** CNNs inherently lack the ability to analyse frequency-based patterns, which are critical in brain MRI where minute texture variations across frequencies might indicate early disease onset.
2. **No Multi-Resolution Analysis:** CNN filters operate at a single resolution, missing out on the multi-scale detail extraction that could benefit Alzheimer's diagnosis (which typically involves both large-scale atrophy and fine-grained textural changes).
3. **Loss of Fine-Grained Information Through Pooling:** Despite improvements like Avg-TopK, pooling operations can still result in some information loss, especially in early layers where subtle features may be prematurely down sampled.
4. **Requires Moderate-to-High Computational Resources:** Though lighter than large transfer learning models, training the CNN model for 200 epochs even on  $128 \times 128$  images still require a moderate amount of computational power. This may not be ideal for deployment in low-resource clinical environments or on devices

with limited processing capabilities, especially when scaling to larger datasets or higher-resolution inputs.

5. **Limited Robustness to Low-Contrast or Noisy Inputs:** CNNs may underperform in MRI images with poor contrast or noise, since they lack inherent mechanisms to adapt to intensity variations—unlike frequency-based approaches which can isolate meaningful signal components.

## CHAPTER 3

### PROPOSED WORK

#### **3.1 Methodology**

Before exploring specific deep learning techniques in AD prognosis, it is important to understand what deep learning entails and its relevance to this domain. Deep learning is a branch of machine learning that utilizes multi-layered neural networks to learn complex patterns and data representations. It is particularly effective when working with large and high-dimensional datasets, such as medical images and sequential health data. Unlike traditional machine learning, deep learning models can automatically extract and learn features from raw data, minimizing the need for manual intervention. These capabilities make deep learning well-suited for tasks like early detection, classification, and monitoring of disease progression in Alzheimer's Disease. Despite its advantages, deep learning requires substantial data, computational resources, and careful hyperparameter tuning. When appropriately applied and thoroughly validated, deep learning models can produce accurate and reliable results, aiding clinicians in making more informed decisions regarding AD prognosis.

#### **3.2 Synopsis of the algorithms employed**

##### **3.2.1 Deep Convolutional Neural Network (DCNN)**

**Deep Convolutional Neural Networks (DCNNs)** are an advanced extension of CNNs characterized by a deeper architecture comprising many convolutional and non-linear layers. The increased depth allows these networks to learn more complex and abstract features, making them particularly effective for tasks such as image recognition, object detection, and even text analysis. DCNNs process information in a hierarchical manner—early layers detect basic patterns like edges, while deeper layers recognize complex structures like objects or scenes. Their strength lies in their ability to learn directly from raw input data, eliminating the need for manual feature engineering and improving generalization on large-scale datasets.

The architecture of a DCNN typically includes convolutional layers for feature extraction, ReLU layers for introducing non-linearity, pooling layers for spatial dimensionality reduction, and fully connected layers for final prediction. To enhance generalization and prevent overfitting, dropout layers are also commonly used. When applied to image analysis, DCNNs utilize localized filters to identify features at different levels and combine them progressively for accurate classification. DCNNs have significantly influenced fields such as medical imaging, video surveillance, and autonomous systems due to their high accuracy and scalability. However, their complexity demands considerable computational resources and large annotated datasets, which can be

challenging in resource-constrained environments. In this work, the preprocessing steps used for the DCNN model are the same as those applied in the CNN model.

### 3.2.2 Wavelet Convolutional Neural Network (WCNN)

**Wavelet Convolutional Neural Networks (WCNNs)** are a sophisticated variant of CNNs that incorporate wavelet transforms to improve feature extraction and enhance robustness to noise. Unlike conventional CNNs that rely solely on max pooling or strided convolutions for down-sampling, WCNNs utilize the Discrete Wavelet Transform (DWT) to decompose feature maps into low-frequency and high-frequency components. Low-frequency components retain critical structural information, while high-frequency components, which often carry noise, are discarded—leading to more accurate and noise-resilient classifications. The performance of WCNNs is significantly influenced by the choice of wavelet functions, such as Haar, Daubechies, Symlet, or Coiflet. Each wavelet type possesses unique properties that affect how data is decomposed and analyzed. Selecting the appropriate wavelet based on the characteristics of the dataset is crucial for optimal performance and accurate feature learning.

WCNN architectures typically consist of wavelet-based down-sampling, traditional convolutional layers, and, in some implementations, Inverse DWT (IDWT) to reconstruct data after processing. By replacing pooling operations with wavelet-based alternatives, WCNNs offer better spatial localization and support multi-resolution analysis, making them ideal for tasks like image restoration, medical diagnosis, and signal processing. Particularly effective in noisy environments, WCNNs can improve classification accuracy without adding extra learnable parameters. However, they also introduce challenges, such as increased architectural complexity and sensitivity to wavelet selection, which can influence their performance across different datasets.

## 3.3 Feature Extraction in Improved WCNN

The Improved Wavelet Convolutional Neural Network (WCNN) begins by leveraging the Discrete Wavelet Transform (DWT) as a preprocessing step to extract meaningful features from input images. Unlike raw pixel values or traditional image transformations, DWT offers a multiresolution analysis framework that captures both spatial and frequency information, making it particularly effective for complex image data. The Haar wavelet, a commonly used orthogonal wavelet, is employed for its simplicity and computational efficiency. When DWT is applied to an image, it decomposes it into four components: LL (approximation), LH (horizontal detail), HL (vertical detail), and HH (diagonal detail). Among these, the LL component retains the low-frequency information that represents the general structure and coarse features of the image, while suppressing noise and high-frequency variations. This approximation image is then selected for further processing, reducing the dimensionality of the input and enhancing computational efficiency without significantly compromising feature richness. The transformed images, being smaller in size, also lead to faster training and better generalization.

### 3.4 Comparison of CNN, DCNN and WCNN

In this study, the effectiveness of frequency-aware feature extraction through wavelet-based downsampling is clearly demonstrated by the superior performance of the WCNN model. The CNN model, integrated with SE blocks and Avg-TopK pooling, achieved a test accuracy of 84.53%, indicating moderate performance. The deeper DCNN model, which incorporated strided convolutions and SE blocks, significantly enhanced generalization capability and reached a test accuracy of 93.44%. The proposed WCNN model, utilizing discrete wavelet transform (Haar) for downsampling instead of traditional pooling operations, outperformed the others with a test accuracy of 95.63%. Despite having fewer layers and no SE blocks, WCNN effectively preserved essential frequency and spatial features from the MRI scans, enabling more accurate classification. These results demonstrate the robustness of WCNN and establish it as the most suitable and efficient model for Alzheimer's disease classification in this research. As shown in [Table 3.1](#), the WCNN model outperformed the CNN and DCNN models across all datasets, achieving the highest train, validation, and test accuracies.

Models	Train Accuracy	Validation Accuracy	Test Accuracy
CNN	92.46%	85.55%	84.53%
DCNN	99.54%	93.65%	93.44%
WCNN	99.74%	95.80%	95.63%

**Table 3.1** Accuracies obtained for different models

### 3.5 Model Analysis using Grad-CAM

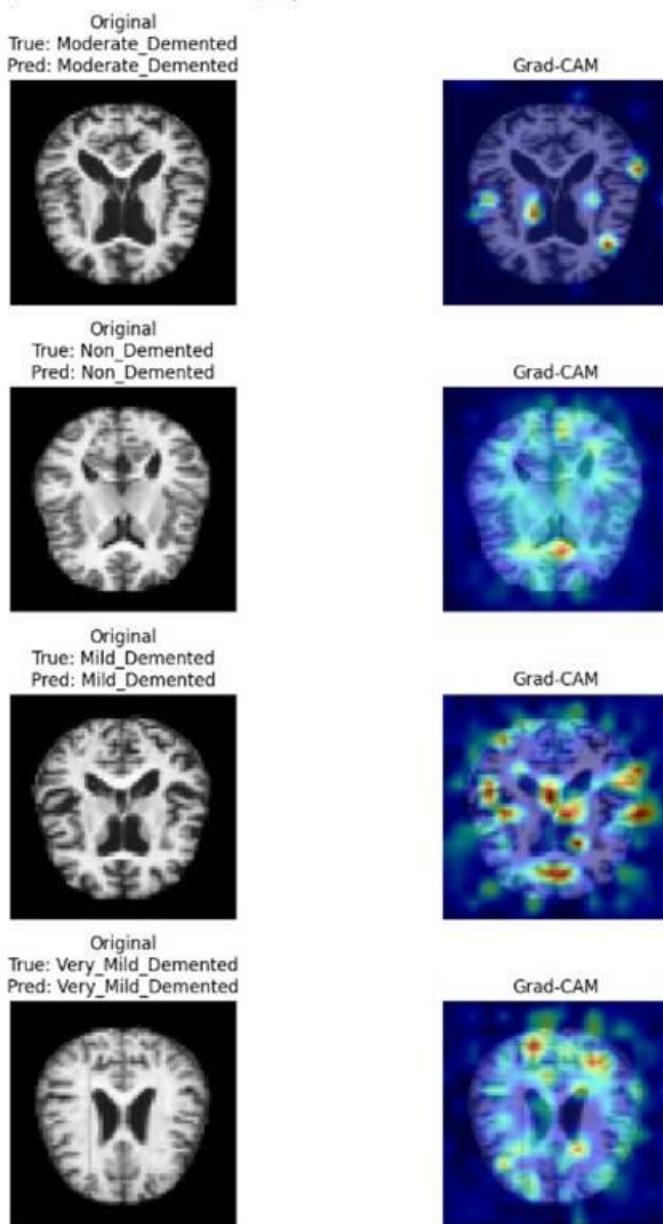


Fig. 1.5 Prediction using Grad CAM

1. **Moderate\_Demented (True & Predicted)** The Grad-CAM highlights symmetrical regions around the lateral ventricles and some deeper brain structures. This suggests the model is focusing on atrophy or shape changes commonly associated with moderate dementia.
2. **Non\_Demented (True & Predicted)** The model focuses more on the central and parietal areas, with less intense activation. This wide and dispersed attention might reflect the model confirming the absence of abnormalities, supporting a “Non\_Demented” classification.

3. Mild\_Demented (True & Predicted) Strong activations are seen in scattered cortical and subcortical regions. This indicates the model is detecting subtle variations in brain texture or structure, which are characteristic of early dementia stages.
4. Very\_Mild\_Demented (True & Predicted) The model concentrates on a few small, localized areas near the center and lower part of the brain. This focused attention suggests it's picking up on very early and subtle signs of degeneration typical of very mild dementia.

### 3.6 WCNN pseudocode

**Input:**

d : dataset,  
t : target size

**Output:**

Classified stage of Alzheimer's Disease on MRI scan

For i in dataset, do

    Apply wavelet transform to i using Haar DWT  
    Retain only approximation coefficients (LL)

Reshape input to (64, 64, 1)

S ← WCNN (X\_train, y\_train)

    For j in layers do

        Conv2D (filters, (3,3), padding="same", activation='relu')  
        BatchNormalization()  
        MaxPooling2D ((2,2))

Flatten → Dense (256, 'relu') → Dropout (0.5) → Dense (4, 'softmax')

Evaluate (DIR, X\_test, S)

Return result

# CHAPTER 4

## SOURCE CODE

### 4.1 Loading MRI Dataset for Alzheimer's Classification

```
import pandas as pd

train_path = "/kaggle/input/alzheimer-mri-disease-classification-dataset/Alzheimer MRI Disease Classification Dataset/Data/train-00000-of-00001-c08a401c53fe5312.parquet"
test_path = "/kaggle/input/alzheimer-mri-disease-classification-dataset/Alzheimer MRI Disease Classification Dataset/Data/test-00000-of-00001-44110b9df98c5585.parquet"

train_df = pd.read_parquet(train_path) test_df = pd.read_parquet(test_path)
print("Train Data:") display(train_df.head())
print("\nTest Data:") display(test_df.head())
```

Train Data:

	image	label
0	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	2
1	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	0
2	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	3
3	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	3
4	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	2

Test Data:

	image	label
0	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	3
1	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	0
2	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	2
3	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	3
4	{"bytes": b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x...}	0

### 4.2 Class Distribution

```
print("Unique Labels in Train Dataset:")
print(train_df["label"].unique())
print("\nClass Distribution:")
print(train_df["label"].value_counts())
```

Unique Labels in Train Dataset:

[2 0 3 1]

```

Class Distribution:
label
2 2566
3 1781
0 724
1 49
Name: count, dtype: int64

```

### 4.3 Decoding Image Data

```

import io import numpy as np
import matplotlib.pyplot as plt from PIL
import Image

def decode_image(image_data):
    return Image.open(io.BytesIO(image_data))
first_image_data = train_df["image"].iloc[0]['bytes']
first_image = decode_image(first_image_data)

plt.imshow(first_image)
plt.axis("off")
plt.title(f"Label: {train_df['label'].iloc[0]}")
plt.show()

```

### 4.4 Image preprocessing and array conversion for model training

```

import tensorflow as tf

def preprocess_image(image_data):
    img = decode_image(image_data)    # Decode
    img = img.resize((128, 128)) # Resize img = np.array(img) / 255.0 # Normalize
    return img

X_train = np.array([preprocess_image(i['bytes']) for i in train_df["image"]]) y_train =
np.array(train_df["label"])

X_test = np.array([preprocess_image(i['bytes']) for i in test_df["image"]]) y_test =
np.array(test_df["label"])

print(f"Processed Train Data: {X_train.shape}, Labels: {y_train.shape}")
print(f"Processed Test Data: {X_test.shape}, Labels: {y_test.shape}")

```

```
Processed Train Data: (5120, 128, 128), Labels: (5120,)  
Processed Test Data: (1280, 128, 128), Labels: (1280,)
```

#### 4.5 Splitting training data into train and validation sets

```
from sklearn.model_selection import train_test_split  
  
X_train_res, X_val, y_train_res, y_val = train_test_split( X_train, y_train, test_size=0.2,  
stratify=y_train, random_state=42 )  
print(f"Train Set: {X_train_res.shape}, Labels: {y_train_res.shape}")  
print(f"Validation Set: {X_val.shape}, Labels: {y_val.shape}")
```

```
Train Set: (4096, 128, 128), Labels: (4096,)  
Validation Set: (1024, 128, 128), Labels: (1024,)
```

```
Train Set: (4096, 128, 128), Labels: (4096,)  
Validation Set: (1024, 128, 128), Labels: (1024,)
```

#### 4.6 Balancing training data using SMOTE

```
from imblearn.over_sampling import SMOTE  
  
X_train_flat = X_train_res.reshape(X_train_res.shape[0], -1)  
smote = SMOTE(random_state=42) X_train_balanced, y_train_balanced =  
smote.fit_resample(X_train_flat, y_train_res)  
X_train_balanced = X_train_balanced.reshape(-1, 128, 128)  
print(f"Balanced Train Set: {X_train_balanced.shape}, Labels: {y_train_balanced.shape}")
```

```
Balanced Train Set: (8212, 128, 128), Labels: (8212,)
```

#### 4.7 One-hot encoding of class labels

```
from tensorflow.keras.utils import to_categorical  
  
y_train_balanced = to_categorical(y_train_balanced, num_classes=4)  
y_val = to_categorical(y_val, num_classes=4)  
y_test = to_categorical(y_test, num_classes=4)  
print(f"One-Hot Encoded Labels: {y_train_balanced.shape}")
```

```
One-Hot Encoded Labels: (8212, 4)
```

#### 4.8 Function of Squeeze & Excitation (SE) Block

```
import tensorflow as tf from tensorflow.keras.layers  
import Conv2D, Dense, Dropout, GlobalAveragePooling2D, Multiply, Reshape
```

```
def se_block(input_tensor, ratio=16):
```

```

filters = input_tensor.shape[-1]

# Squeeze: Global Average Pooling
se = GlobalAveragePooling2D()(input_tensor)
se = Dense(filters // ratio, activation="relu")(se)
se = Dense(filters, activation="sigmoid")(se)

# Excite: Scale feature maps
se = Reshape((1, 1, filters))(se)

return Multiply()([input_tensor, se])

```

#### 4.9 Function of AvgTopKPooling

```

import tensorflow.keras.backend as K
from tensorflow.keras.layers import Layer

class AvgTopKPooling(Layer):

    def __init__(self, k=3, **kwargs):
        super(AvgTopKPooling, self).__init__(**kwargs)
        self.k = k

    def call(self, inputs):
        top_k, _ = tf.math.top_k(inputs, k=self.k, sorted=False)
        return K.mean(top_k, axis=-1)

    def compute_output_shape(self, input_shape):
        return input_shape[:-1]

```

#### 4.10 Training of CNN Model

```

from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, ReLU, Flatten, Dense, Dropout

# Define CNN Model
def build_model(input_shape=(128, 128, 1), num_classes=4):
    inputs = Input(shape=input_shape)

    # Block 1
    x = Conv2D(32, (3, 3), padding="same")(inputs)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = se_block(x)

```

```

# Block 2
x = Conv2D(64, (3, 3), padding="same")(x)
x = BatchNormalization()(x)
x = ReLU()(x)
x = se_block(x)

# Block 3
x = Conv2D(128, (3, 3), padding="same")(x)
x = BatchNormalization()(x)
x = ReLU()(x)
x = se_block(x)

# Block 4
x = Conv2D(256, (3, 3), padding="same")(x)
x = BatchNormalization()(x)
x = ReLU()(x)
x = se_block(x)

# Avg-TopK Pooling
x = AvgTopKPooling(k=3)(x)

# Flatten & Fully Connected Layers
x = Flatten()(x)
x = Dense(128, activation="relu")(x)
x = Dropout(0.3)(x)
x = Dense(64, activation="relu")(x)
x = Dropout(0.3)(x)

# Output Layer
outputs = Dense(num_classes, activation="softmax")(x)
model = Model(inputs, outputs)
return model

Epoch 1/200
129/129 ————— 0s 754ms/step - accuracy: 0.2831 - loss: 2.9922
Epoch 1: val_loss improved from inf to 1.30846, saving model to best_model.keras
129/129 ————— 148s 800ms/step - accuracy: 0.2835 - loss: 2.9831 - val_accuracy: 0.1416 - val_loss: 1.3085 - learning_rate: 0.0010
Epoch 2/200
129/129 ————— 0s 635ms/step - accuracy: 0.4505 - loss: 1.0985
Epoch 2: val_loss improved from 1.30846 to 1.24140, saving model to best_model.keras

```

```

129/129 ----- 86s 669ms/step - accuracy:
0.4507 - loss: 1.0981 - val_accuracy: 0.3477 - val_loss: 1.2414 - learning_rate: 0.0010
.....  

.....  

Epoch 42/200
129/129 ----- 0s 633ms/step - accuracy:
0.9281 - loss: 0.2485
Epoch 42: val_loss did not improve from 0.52069
129/129 ----- 86s 666ms/step - accuracy:
0.9281 - loss: 0.2486 - val_accuracy: 0.7900 - val_loss: 0.7061 - learning_rate: 8.0000e-04
Epoch 43/200
129/129 ----- 0s 633ms/step - accuracy:
0.9241 - loss: 0.2717
Epoch 43: val_loss did not improve from 0.52069
129/129 ----- 86s 666ms/step - accuracy:
0.9241 - loss: 0.2716 - val_accuracy: 0.8555 - val_loss: 0.6389 - learning_rate: 8.0000e-04
Epoch 43: early stopping
Restoring model weights from the end of the best epoch: 28.

```

#### 4.11 Training of DCNN Model

```

from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, ReLU, Flatten,
Dense, Dropout

# Define Deep CNN Model
def build_model(input_shape=(128, 128, 1), num_classes=4):
    inputs = Input(shape=input_shape)

    # Block 1
    x = Conv2D(32, (3, 3), padding="same")(inputs)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = se_block(x)

    # Block 2
    x = Conv2D(64, (3, 3), padding="same", strides=2)(x)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = se_block(x)

```

```

# Block 3
x = Conv2D(128, (3, 3), padding="same")(x)
x = BatchNormalization()(x)
x = ReLU()(x)
x = se_block(x)

# Block 4
x = Conv2D(256, (3, 3), padding="same", strides=2)(x)
x = BatchNormalization()(x)
x = ReLU()(x)
x = se_block(x)

# Block 5
x = Conv2D(512, (3, 3), padding="same")(x)
x = BatchNormalization()(x)
x = ReLU()(x)
x = se_block(x)

# Avg-TopK Pooling
x = AvgTopKPooling(k=3)(x)

# Flatten & Fully Connected Layers
x = Flatten()(x)
x = Dense(256, activation="relu")(x)
x = Dropout(0.4)(x)
x = Dense(128, activation="relu")(x)
x = Dropout(0.3)(x)
x = Dense(64, activation="relu")(x)
x = Dropout(0.3)(x)

# Output Layer
outputs = Dense(num_classes, activation="softmax")(x)
model = Model(inputs, outputs)
return model

```

Epoch 1/200

**129/129** ————— **0s** 289ms/step - accuracy:

0.2706 - loss: 1.5484

Epoch 1: val\_loss improved from inf to 1.49368, saving model to best\_model.keras

**129/129** ————— **69s** 313ms/step - accuracy:

0.2709 - loss: 1.5471 - val\_accuracy: 0.1416 - val\_loss: 1.4937 - learning\_rate: 0.0010

Epoch 2/200

**129/129** ————— **0s** 185ms/step - accuracy:

```

0.5218 - loss: 1.0030
Epoch 2: val_loss improved from 1.49368 to 1.37538, saving model to best_model.keras
129/129 ————— 25s 196ms/step - accuracy:
0.5222 - loss: 1.0022 - val_accuracy: 0.1416 - val_loss: 1.3754 - learning_rate: 0.0010
.....  

.....  

Epoch 40/200
129/129 ————— 0s 188ms/step - accuracy:
0.9956 - loss: 0.0116
Epoch 40: val_loss did not improve from 0.20372
129/129 ————— 26s 198ms/step - accuracy:
0.9956 - loss: 0.0116 - val_accuracy: 0.9199 - val_loss: 0.4310 - learning_rate: 6.4000e-04
Epoch 41/200
129/129 ————— 0s 188ms/step - accuracy:
0.9955 - loss: 0.0166
Epoch 41: val_loss did not improve from 0.20372
129/129 ————— 26s 198ms/step - accuracy:
0.9955 - loss: 0.0166 - val_accuracy: 0.9365 - val_loss: 0.2855 - learning_rate: 6.4000e-04
Epoch 41: early stopping
Restoring model weights from the end of the best epoch: 26

```

#### 4.12 Training of WCNN model

```

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,
BatchNormalization
import pywt
import numpy as np

def wavelet_transform(image):
    coeffs2 = pywt.dwt2(image, 'haar') # Apply 2D Haar wavelet transform
    LL, (LH, HL, HH) = coeffs2 # Extract coefficients
    return LL # Keep only approximation coefficients

# Apply Wavelet Transform to dataset
X_train_wcnn = np.array([wavelet_transform(img.squeeze()) for img in
X_train_balanced])
X_val_wcnn = np.array([wavelet_transform(img.squeeze()) for img in X_val])
X_test_wcnn = np.array([wavelet_transform(img.squeeze()) for img in X_test])

```

```

# Reshape to match input shape (adding channel dimension back)
X_train_wcnn = X_train_wcnn.reshape(-1, 64, 64, 1) # Wavelet reduces size by half
X_val_wcnn = X_val_wcnn.reshape(-1, 64, 64, 1)
X_test_wcnn = X_test_wcnn.reshape(-1, 64, 64, 1)

# Define WCNN Model
wcnn_model = Sequential([
    Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(64, 64, 1)),
    BatchNormalization(),
    MaxPooling2D((2, 2)),

    Conv2D(64, (3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D((2, 2)),

    Conv2D(128, (3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D((2, 2)),

    Flatten(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax') # 4 output classes
])

# Compile the model
wcnn_model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),
loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
wcnn_history = wcnn_model.fit(
    X_train_wcnn, y_train_balanced,
    validation_data=(X_val_wcnn, y_val),
    epochs=30,
    batch_size=32
)

Epoch 1/30
257/257 ————— 9s 15ms/step - accuracy:
0.5208 - loss: 1.2969 - val_accuracy: 0.3486 - val_loss: 1.7682
Epoch 2/30
257/257 ————— 2s 6ms/step - accuracy:
0.7361 - loss: 0.5848 - val_accuracy: 0.5361 - val_loss: 1.0317

```

Epoch 3/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.8330 - loss: 0.4108 - val\_accuracy: 0.7617 - val\_loss: 0.5658

Epoch 4/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.8870 - loss: 0.2913 - val\_accuracy: 0.8174 - val\_loss: 0.4583

Epoch 5/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9282 - loss: 0.1940 - val\_accuracy: 0.8613 - val\_loss: 0.3448

Epoch 6/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9561 - loss: 0.1216 - val\_accuracy: 0.9082 - val\_loss: 0.2580

Epoch 7/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9726 - loss: 0.0816 - val\_accuracy: 0.9082 - val\_loss: 0.2595

Epoch 8/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9805 - loss: 0.0620 - val\_accuracy: 0.9092 - val\_loss: 0.2346

Epoch 9/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9862 - loss: 0.0442 - val\_accuracy: 0.9297 - val\_loss: 0.1962

Epoch 10/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9883 - loss: 0.0356 - val\_accuracy: 0.9102 - val\_loss: 0.2271

Epoch 11/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9913 - loss: 0.0285 - val\_accuracy: 0.9492 - val\_loss: 0.1604

Epoch 12/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9959 - loss: 0.0170 - val\_accuracy: 0.9561 - val\_loss: 0.1567

Epoch 13/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9959 - loss: 0.0174 - val\_accuracy: 0.9336 - val\_loss: 0.1864

Epoch 14/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9956 - loss: 0.0182 - val\_accuracy: 0.9453 - val\_loss: 0.1739

Epoch 15/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9970 - loss: 0.0145 - val\_accuracy: 0.9160 - val\_loss: 0.2239

Epoch 16/30  
**257/257**  **2s** 6ms/step - accuracy:  
0.9914 - loss: 0.0305 - val\_accuracy: 0.9531 - val\_loss: 0.1717

Epoch 17/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9939 - loss: 0.0177 - val\_accuracy: 0.9551 - val\_loss: 0.1483

Epoch 18/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9969 - loss: 0.0106 - val\_accuracy: 0.9404 - val\_loss: 0.1625

Epoch 19/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9953 - loss: 0.0149 - val\_accuracy: 0.9521 - val\_loss: 0.1791

Epoch 20/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9932 - loss: 0.0271 - val\_accuracy: 0.9434 - val\_loss: 0.1987

Epoch 21/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9909 - loss: 0.0245 - val\_accuracy: 0.9248 - val\_loss: 0.2199

Epoch 22/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9968 - loss: 0.0111 - val\_accuracy: 0.9307 - val\_loss: 0.2261

Epoch 23/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9977 - loss: 0.0068 - val\_accuracy: 0.9404 - val\_loss: 0.2254

Epoch 24/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9952 - loss: 0.0147 - val\_accuracy: 0.9521 - val\_loss: 0.1830

Epoch 25/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9987 - loss: 0.0055 - val\_accuracy: 0.9639 - val\_loss: 0.1154

Epoch 26/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9977 - loss: 0.0056 - val\_accuracy: 0.9336 - val\_loss: 0.2990

Epoch 27/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9948 - loss: 0.0128 - val\_accuracy: 0.9541 - val\_loss: 0.1973

Epoch 28/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9955 - loss: 0.0148 - val\_accuracy: 0.8691 - val\_loss: 0.5151

Epoch 29/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9901 - loss: 0.0292 - val\_accuracy: 0.9648 - val\_loss: 0.1385

Epoch 30/30

**257/257** ————— **2s** 6ms/step - accuracy:  
0.9978 - loss: 0.0067 - val\_accuracy: 0.9580 - val\_loss: 0.1706

**40/40** ————— **0s** 2ms/step - accuracy: 0.9523 -  
loss: 0.1757  
WCNN Test Accuracy: 0.9563

#### 4.13 Grad CAM Visualization

```
from tensorflow.keras import Model, Input
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,
BatchNormalization
from tf_keras_vis.gradcam import Gradcam
from tf_keras_vis.utils.model_modifiers import ReplaceToLinear
from tf_keras_vis.utils.scores import CategoricalScore
import matplotlib.pyplot as plt import numpy as np

input_layer = Input(shape=(64, 64, 1))
x = Conv2D(32, (3, 3), activation='relu', padding='same')(input_layer)
x = BatchNormalization()(x) x = MaxPooling2D((2, 2))(x)
x = Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = BatchNormalization()(x) x = MaxPooling2D((2, 2))(x)
x = Conv2D(128, (3, 3), activation='relu', padding='same')(x)
x = BatchNormalization()(x) x = MaxPooling2D((2, 2))(x)
x = Flatten()(x) x = Dense(256, activation='relu')(x)
x = Dropout(0.5)(x)

output_layer = Dense(4, activation='softmax')(x)
wcnn_model_func = Model(inputs=input_layer, outputs=output_layer)
wcnn_model_func.set_weights(wcnn_model.get_weights())

gradcam = Gradcam(model=wcnn_model_func, model_modifier=ReplaceToLinear(),
clone=True)
num_images = 4
indices = [435, 77, 4, 1200] # Indices of images you want to visualize
plt.figure(figsize=(10, num_images * 3))

for i, idx in enumerate(indices):
    input_image = X_test_wcnn[idx:idx+1].astype(np.float32)
    true_label = np.argmax(y_test[idx])
    pred_label = np.argmax(wcnn_model_func.predict(input_image))
    score = CategoricalScore([true_label])
    cam = gradcam(score, input_image, penultimate_layer=-1)
    heatmap = cam[0]

    # Original
```

```
plt.subplot(num_images, 2, 2 * i + 1)
plt.imshow(input_image[0].squeeze(), cmap='gray')
plt.title(f"Original\nTrue: {label_map[true_label]}\nPred:{label_map[ pred_label ]}")
plt.axis('off')

# Heatmap
plt.subplot(num_images, 2, 2 * i + 2)
plt.imshow(input_image[0].squeeze(), cmap='gray')
plt.imshow(heatmap, cmap='jet', alpha=0.5)
plt.title("Grad-CAM")
plt.axis('off')

plt.tight_layout()
plt.show()
```

## CHAPTER 5

### SNAPSHOTS

In Fig. 5.1 the bar chart illustrates the class distribution in the training set of an Alzheimer's MRI dataset. It shows that the 'Non\_Demented' class has the highest number of images, followed by 'Very\_Mild\_Demented'. The 'Mild\_Demented' class has significantly fewer images, and 'Moderate\_Demented' has the lowest representation in the dataset.

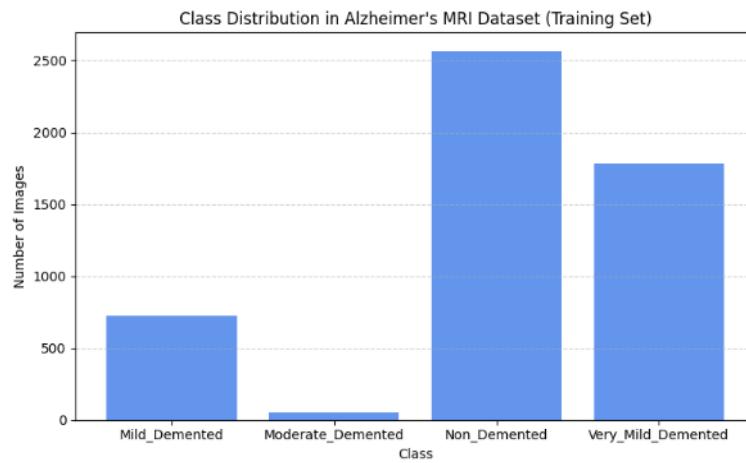


Fig. 5.1 Class distribution

#### Output of CNN

```
40/40 ━━━━━━━━ 12s 135ms/step - accuracy: 0.8413 - loss: 0.5900  
Test Accuracy: 84.53%  
Test Loss: 0.5134
```

#### Output of DCNN

```
40/40 ━━━━━━━━ 6s 35ms/step - accuracy: 0.9311 - loss: 0.3274  
Test Accuracy: 93.44%  
Test Loss: 0.2707
```

#### Output of WCNN

```
Epoch 30/30  
257/257 ━━━━━━━━ 2s 6ms/step - accuracy: 0.9978 - loss: 0.0067 - val_accuracy: 0.  
9580 - val_loss: 0.1706  
40/40 ━━━━━━━━ 0s 2ms/step - accuracy: 0.9523 - loss: 0.1757  
WCNN Test Accuracy: 0.9563
```

The results and outcomes of the models that were put into practice are displayed in this part. Among the three models evaluated (CNN, DCNN, and WCNN) -WCNN demonstrated the most effective performance in classifying Alzheimer's disease into four categories. The corresponding confusion matrices for CNN, DCNN, and WCNN are illustrated in Fig. 5.2, Fig. 5.3, and Fig. 5.4. As shown in the graphs below Fig. 5.5, Fig. 5.6, and Fig. 5.7, WCNN demonstrates a smoother and more consistent learning curve compared to CNN and DCNN, indicating better generalization and stability during training. While CNN and DCNN show fluctuations in validation loss, WCNN maintains low and stable values, confirming its robustness in learning complex patterns from MRI images. These graphical results further validate the effectiveness of the WCNN model in comparison to traditional CNN architectures for Alzheimer's disease classification

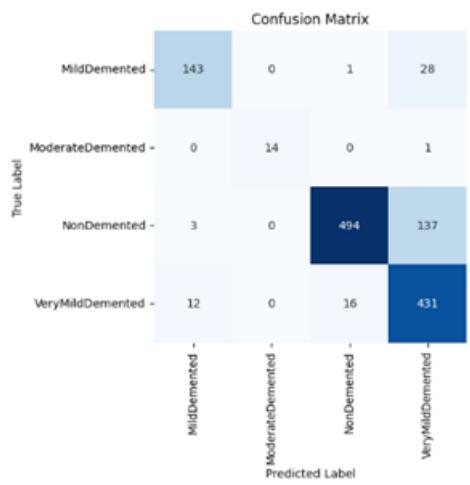


Fig. 5.2 Confusion matrix of CNN

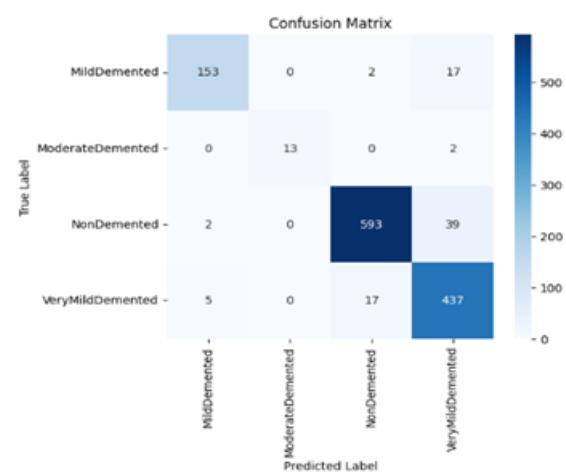


Fig. 5.3 Confusion matrix of DCNN

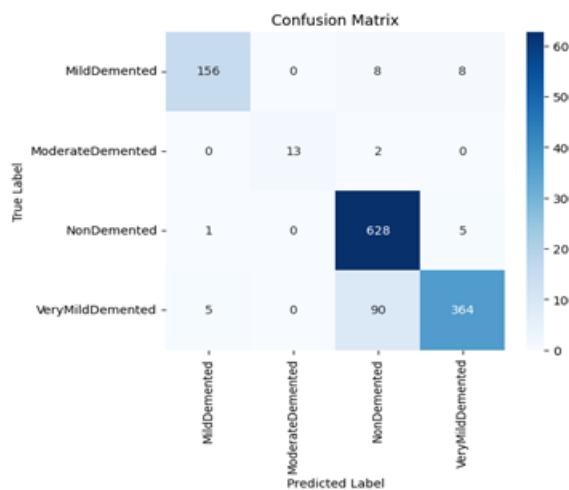


Fig. 5.4 Confusion matrix of WCNN.

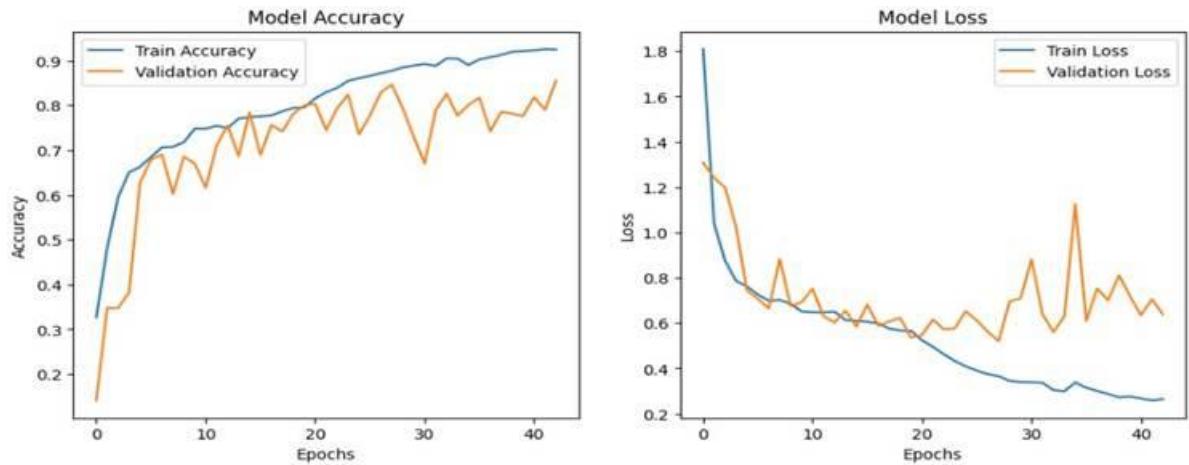


Fig. 5.5 CNN training and validation performance curves

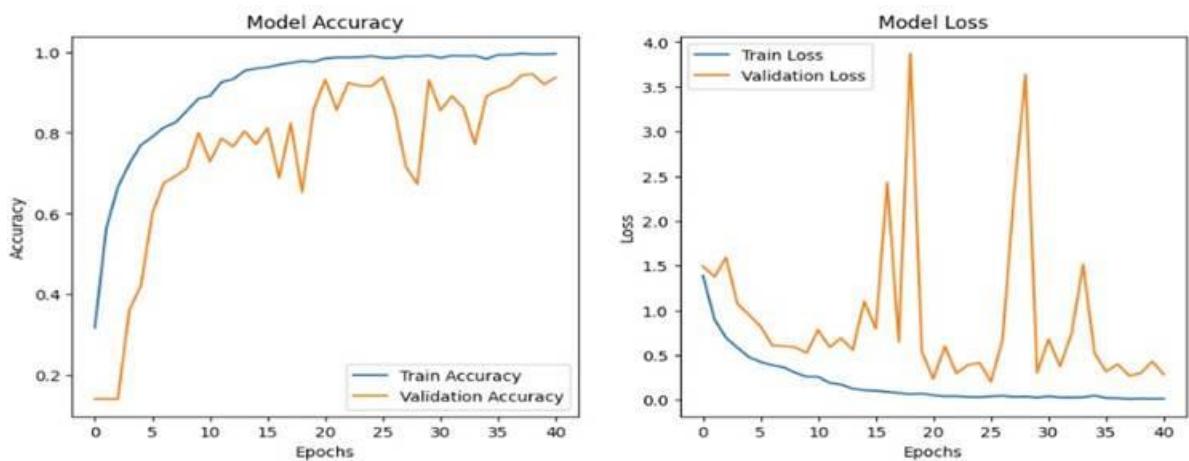


Fig. 5.6 DCNN training and validation performance curves

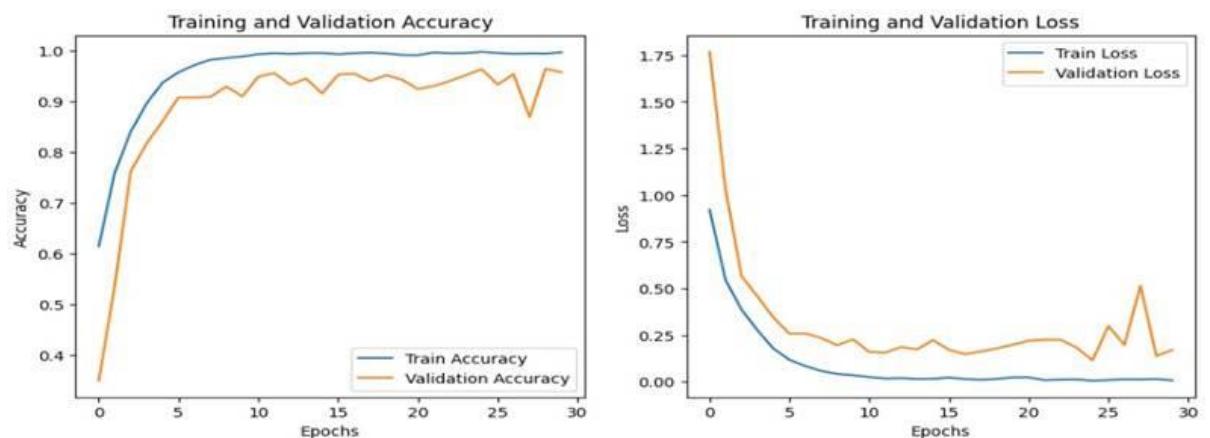


Fig. 5.7 WCNN training and validation performance curves

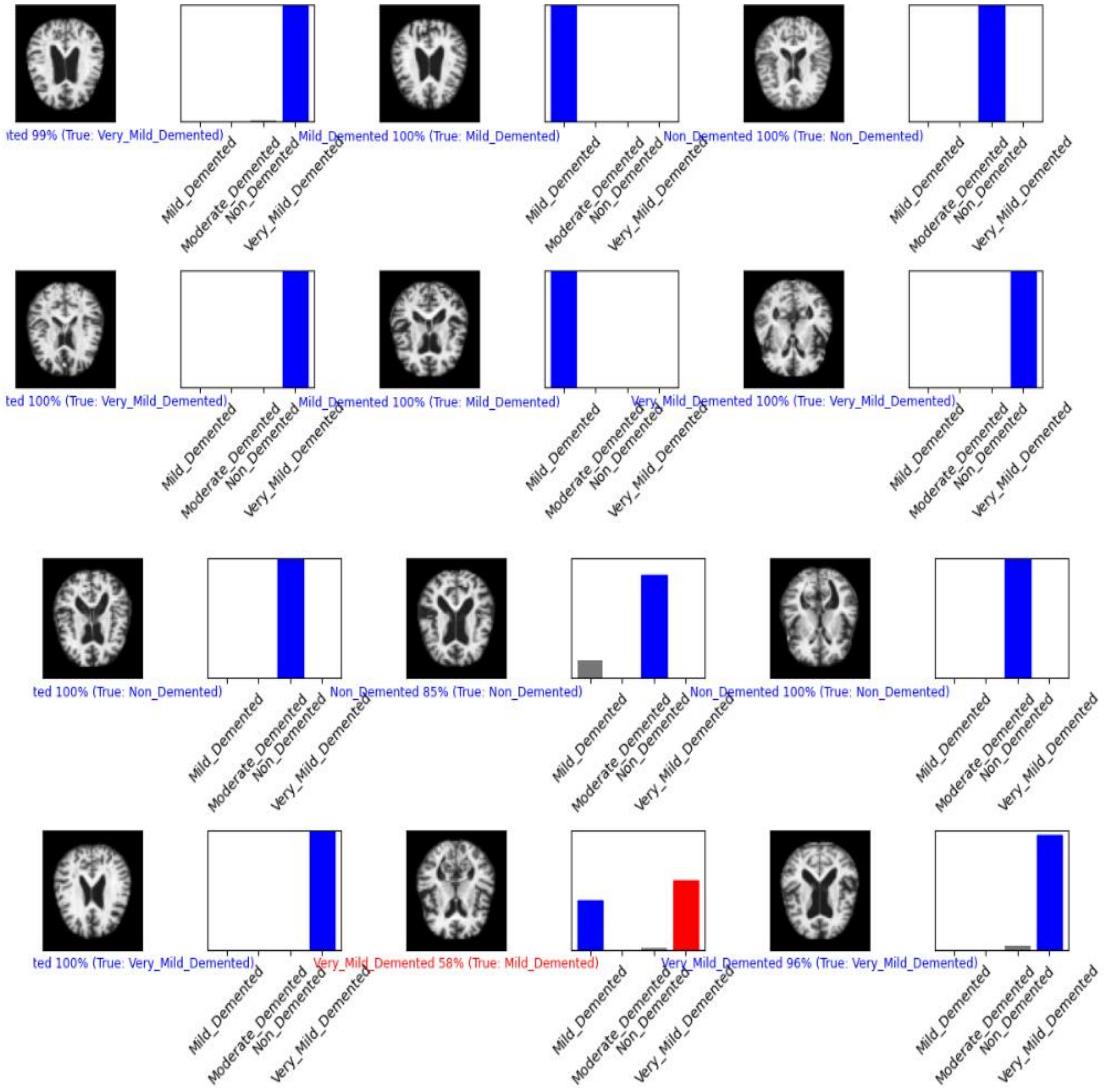


Fig. 5.8 Class Prediction using WCNN

In Fig. 5.8 the visualization demonstrates that the model performs strongly across all four classes of Alzheimer's MRI classification. Most predictions are made with high confidence and match the ground truth labels, indicating the model's reliability. Only one instance shows misclassification, suggesting minimal overfitting and good generalization. The model shows particularly strong performance in distinguishing “Non\_Demented” and “Very\_Mild\_Demented” categories, while maintaining respectable accuracy for “Mild\_Demented” and “Moderate\_Demented” as well.

The precision, recall, and F1-score of the WCNN model for each of the four Alzheimer's classification classes are shown in Fig. 5.9. The WCNN model has outperformed the others in every evaluation metric.

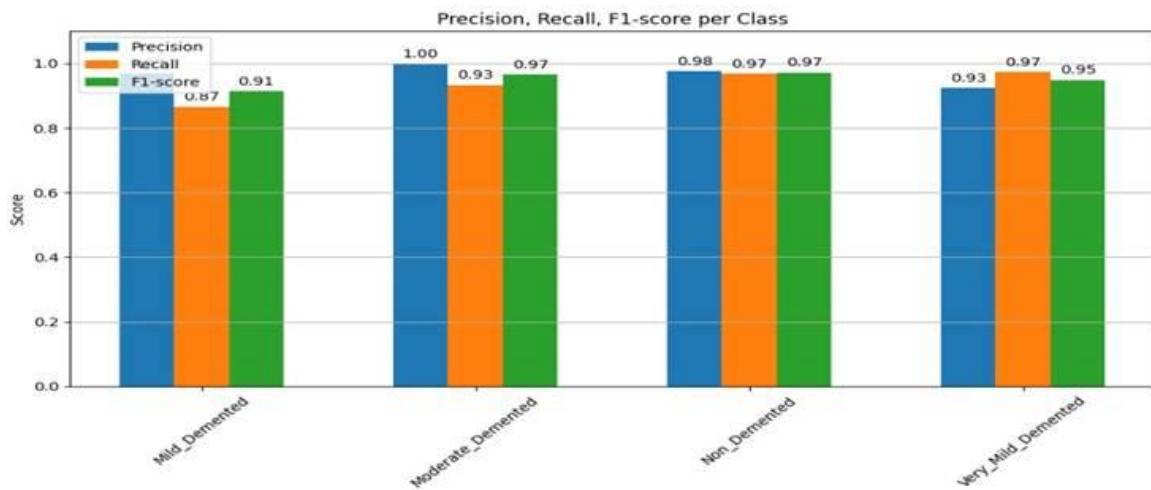


Fig. 5.9 WCNN Precision, Recall & F1-Score per Class

The ROC Figure 5.10 curve further supports this by showing excellent class-wise separation. This demonstrates the model's potential to serve as a reliable tool for early-stage Alzheimer's diagnosis.

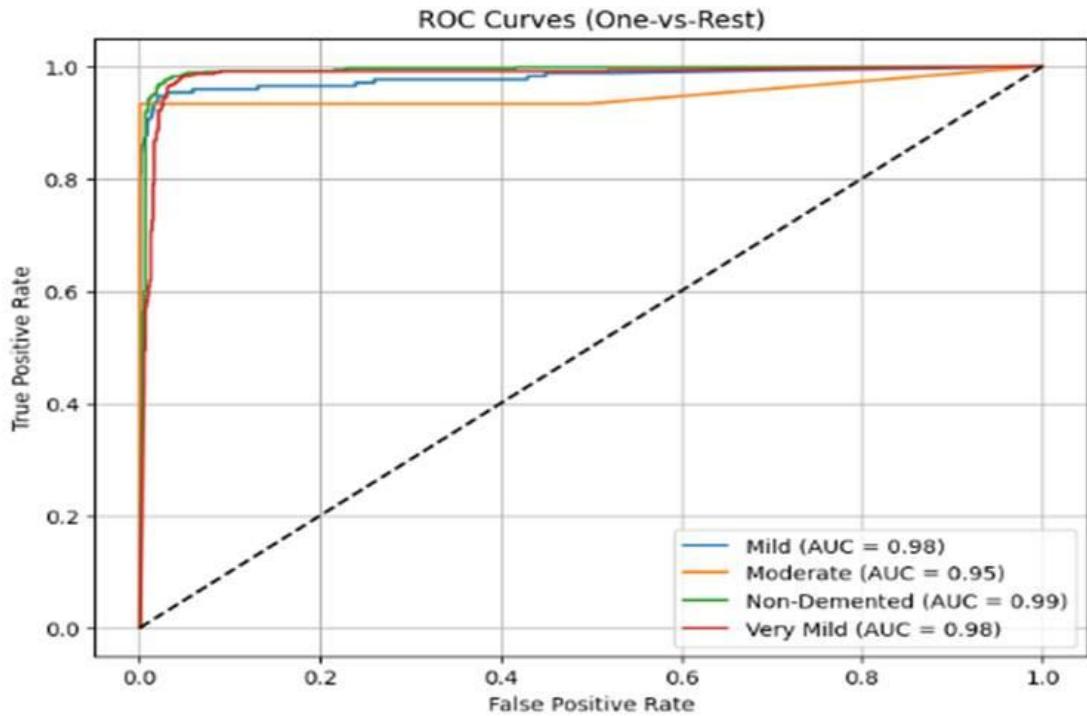


Fig. 5.10 ROC curve of WCNN

## CHAPTER 6

### CONCLUSION AND FUTURE PLANS

#### **Conclusion**

Early and accurate prediction of Alzheimer's disease is critical in the medical field, as it enables timely intervention, slows disease progression, and improves the quality of life for patients. With the growing prevalence of Alzheimer's worldwide, developing efficient and reliable diagnostic tools has become more important than ever. The research shows how well the suggested Wavelet-based Convolutional Neural Network (WCNN) performs in classifying Alzheimer's disease based on MRI scans. Through the substitution of conventional pooling operations with discrete wavelet transform (Haar), the WCNN managed to preserve essential frequency and spatial characteristics, resulting in enhanced prediction accuracy. The WCNN, despite its simpler architecture with fewer layers and no SE blocks, outperformed both the baseline CNN and the deeper DCNN models across all performance metrics. With a test accuracy of 95.63%, it demonstrated greater accuracy and efficiency. The results emphasize the promise of frequency-aware downsampling methods and establish WCNN as a dependable and resilient model for the clinical diagnosis of Alzheimer's disease. This strategy opens avenues for additional investigation of high-performance, lightweight models in the realm of medical image analysis.

#### **Future Plans**

While the current model demonstrates high accuracy and strong potential, there remains considerable scope for further enhancement to make it more practical, scalable, and suitable for real-world therapeutic applications.

##### **1. Advanced Wavelet Techniques**

Future research can explore the integration of more advanced wavelet families such as Daubechies and Symlets, in place of the basic Haar wavelet. These wavelets provide superior frequency localization and smoother reconstruction, enabling the model to capture finer structural details in MRI scans. Such improvements could enhance the model's sensitivity, particularly in detecting subtle changes associated with early-stage Alzheimer's disease.

##### **2. 3D Wavelet-CNN Architecture**

To leverage the full spatial information available in MRI data, the current 2D WCNN framework can be extended to a 3D WCNN architecture. By incorporating 3D discrete wavelet transforms alongside 3D convolutional layers, the model can learn from inter-slice dependencies and volumetric features. This approach would allow for more comprehensive pattern recognition, thereby improving the classification of Alzheimer's stages through a holistic analysis of brain structures.

### **3. Transformer-Wavelet Hybrid Models**

Introducing Transformer-based self-attention mechanisms into the WCNN architecture presents another promising direction. Transformers are well-suited for modeling long-range dependencies, and when combined with wavelet-based embeddings, they can enhance both feature selection and diagnostic accuracy. Such hybrid models could offer a balance between local detail extraction and global contextual awareness, improving both performance and interpretability.

### **4. Real-Time Inference and Edge Deployment**

For practical clinical deployment, particularly in real-time and remote settings, the model can be optimized for lightweight, real-time inference. Techniques such as model pruning, quantization, and conversion to formats like TensorFlow Lite or ONNX would enable the system to run on edge devices, including portable scanners and mobile diagnostic platforms. This would significantly improve accessibility and speed in clinical workflows.

### **5. Clinical Validation and Regulatory Compliance**

Transitioning from research to clinical practice requires rigorous validation in real-world settings, conducted in partnership with healthcare professionals. This includes testing the model's reliability, explainability through tools like Grad-CAM, and aligning its implementation with regulatory standards such as FDA or CE certifications. These efforts are essential for establishing clinical trust and ensuring safe adoption in diagnostic processes.

### **6. Multi-Modal Data Fusion**

An additional avenue for improvement involves the integration of multi-modal data, combining MRI scans with other relevant clinical information such as PET imaging, genetic profiles, or cognitive assessment scores. Utilizing architectures that support multi-input fusion—potentially through attention mechanisms or wavelet-aligned embeddings—can enhance diagnostic robustness and improve the model's effectiveness in handling complex or ambiguous cases.

## CHAPTER 7

### REFERENCES

1. Mehmood A, Yang S, Feng Z, Wang M, Ahmad AS, Khan R, et al. A transfer learning approach for early diagnosis of Alzheimer's disease on MRI images. *Neuroscience*. 2021;460:43–52.  
<https://doi.org/10.1016/j.neuroscience.2021.01.002>.
2. Brookmeyer R, Johnson E, Zieglergraham K. Forecasting the global burden of Alzheimer's disease. *J Alzheimers Dis.* 2007;3(3):186–91.  
<https://doi.org/10.1016/j.jalz.2007.04.381>.
3. World Alzheimer Report 2016. <https://www.alz.co.uk/research/WorldAlzheimerReport2016.pdf>. 2016.
4. World Health Organization. <https://www.who.int/news-room/fact-sheets/detail/dementia>.
5. Mehmood A, Maqsood M, Bashir M, Shuyuan Y. A deep siamese convolution neural network for multi-class classification of Alzheimer disease. *Brain Sci.* 2020;10(2):84.
6. National Institute on Aging. <https://www.nia.nih.gov/health/what-happens-brain-alzheimers-disease>.
7. Bi X, Li S, Xiao B, Li Y, Wang G, Ma X. Computer aided Alzheimer's disease diagnosis by an unsupervised deep learning technology. *Neurocomputing*. 2019;21:1232–45. <https://doi.org/10.1016/j.neucom.2018.11.111>.
8. Tanaka M, Toldi J, Vécsei L. Exploring the etiological links behind neurodegenerative diseases: inflammatory cytokines and bioactive kynurenes. *Int J Mol Sci.* 2020;21(7):2431. <https://doi.org/10.3390/ijms21072431>.
9. Tanaka M, Vécsei L. Editorial of special issue dissecting neurological and neuropsychiatric diseases neurodegeneration and neuroprotection. *Int J Mol Sci.* 2022;23:1–6. <https://doi.org/10.3390/ijms23136991>.
10. Shamrat FJ, Akter S, Azam S, Karim A, Ghosh P, Tasnim Z, et al. AlzheimerNet: An effective deep learning based proposition for Alzheimer's disease stages classification from functional brain changes in magnetic resonance images. *IEEE Access*. 2023;11:16376–16395.
11. Illakiya T, Ramamurthy K, Siddharth M, Mishra R, Udainiya A. AHANet: Adaptive hybrid attention network for Alzheimer's disease classification using brain magnetic resonance imaging. *Bioengineering*. 2023;10(6):714.
12. Illakiya T, Karthik R. Automatic detection of Alzheimer's disease using deep learning models and neuro-imaging: current trends and future perspectives. *Neuroinformatics*. 2023;21(2):339–364.
13. Terry RD, Masliah E. Synaptic pathology in the pathogenesis of Alzheimer dementia. In: Carlsson A, Riederer P, Beckmann H, Nagatsu T, Gershon S, Jellinger KA, et al., editors. *New trends in the diagnosis and therapy of Alzheimer's disease*. Springer Vienna; 1994. p. 1–8.

14. Sisodia PS, Ameta GK, Kumar Y, Chaplot N. A review of deep transfer learning approaches for class-wise prediction of Alzheimer's disease using MRI images. *Arch Comput Methods Eng.* 2023;30(4):2409–2429.
15. Xu J, Qiu C. Worldwide economic costs and societal burden of dementia. In: Perneczky R, editor. *Biomarkers for Preclinical Alzheimer's Disease*. Springer New York; 2018. p. 3–13. [https://doi.org/10.1007/978-1-4939-7674-4\\_1](https://doi.org/10.1007/978-1-4939-7674-4_1).
16. Falahati F, Westman E, Simmons A. Multivariate data analysis and machine learning in Alzheimer's disease with a focus on structural magnetic resonance imaging. *J Alzheimer's Dis.* 2014;41(3):685–708.
17. Sisodia PS, Ameta GK, Kumar Y, Chaplot N. A review of deep transfer learning approaches for class-wise prediction of Alzheimer's disease using MRI images. *Arch Comput Methods Eng.* 2023;30(4):2409–2429.
18. Hasib KM, Tanzim A, Shin J, Faruk KO, Al Mahmud J. BMNet 5: A novel approach of neural network to classify the genre of Bengali music based on audio features. *IEEE Access.* 2022;10:108545–108563.
19. Illakiya T, Karthik R. A dimension centric proximate attention network and swin transformer for age-based classification of mild cognitive impairment from brain MRI. *IEEE Access.* 2023;11:128018–128031.
20. Illakiya T, Karthik R, Alzheimer's Disease Neuroimaging Initiative, et al. A deep feature fusion network with global context and cross-dimensional dependencies for classification of mild cognitive impairment from brain MRI. *Image Vis Comput.* 2024;104967.
21. Rathore S, Guo Y, Hong Y, Zhou Y, Yan L, Zhang Y. Wavelet transform-based deep learning for MRI-based Alzheimer's disease classification. *Neurocomputing.* 2020;406:202–215. <https://doi.org/10.1016/j.neucom.2020.03.024>.
22. Shamrat FJ, Maqsood M, Bashir M. Wavelet transform and convolutional neural network hybrid approach for early diagnosis of Alzheimer's disease from MRI images. *Comput Methods Programs Biomed.* 2021;198:105714. <https://doi.org/10.1016/j.cmpb.2020.1057>

# CHAPTER 8

## APPENDIX-BASE PAPER

Biomedical Signal Processing and Control 96 (2024) 106614



Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: [www.elsevier.com/locate/bspc](http://www.elsevier.com/locate/bspc)



### Advancing early diagnosis of Alzheimer's disease with next-generation deep learning methods

Cuneyt Ozdemir\*, Yahya Dogan

Computer Engineering, Engineering Faculty, Siirt University, 56100, Siirt, Turkey

#### ARTICLE INFO

Dataset link: <https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-image-s>

**Keywords:**

Alzheimer's disease  
Deep learning  
SMOTE  
Squeeze and excitation block  
Avg-TopK

#### ABSTRACT

Alzheimer's disease, characterized by cognitive decline and memory impairment, poses a significant healthcare challenge. This study presents a specially designed CNN model, utilizing contemporary approaches, to distinguish between various types of Alzheimer's disease. This model can serve as an early diagnostic tool to prevent the disease from progressing towards more pronounced and severe dementia symptoms. In this context, the performance of various transfer learning models has been examined, leading to the development of a specialized model integrating compression and excitation blocks, an innovative Avg-TopK pooling layer, and the SMOTE technique to handle data imbalance. The ablation study results demonstrate the critical role of these components, highlighting the model's effectiveness and innovative design. This study is novel in that it combines modern methodologies for detecting Alzheimer's disease, resulting in a model with state-of-the-art accuracy of 99.84% and improved computing efficiency. Grad-CAM analysis further demonstrates that the model focuses on cortical areas during classification, underscoring its potential as a robust diagnostic tool. These innovations represent a significant advancement over existing models, positioning this study as a pioneering effort in the early diagnosis of Alzheimer's disease. This study aims to contribute significantly to both academic research and medical applications by focusing on integrating artificial intelligence methodologies into medical diagnosis.

#### 1. Introduction

Alzheimer's disease (AD) is a neurodegenerative disease primarily observed in the elderly population, characterized by cognitive decline, memory impairment, and behavioral changes [1,2]. As the disease progresses, cognitive abilities, including thinking skills and the ability to recall information, begin to deteriorate. AD is a leading cause of death worldwide, affecting individuals, families, and healthcare systems [3,4]. AD was initially identified by Dr. Alois Alzheimer in 1907. The disease was diagnosed in a 51-year-old female patient who was undergoing treatment at the psychiatric hospital in Frankfurt am Main. As the disease progressed, the patient exhibited various complex symptoms and experienced significant memory loss. It was observed that she could correctly name a series of objects when shown to her, but shortly after, she would forget everything. Additionally, difficulties in comprehension, hallucinations, and reading problems were detected. The disease process lasted for approximately four and a half years, and the patient passed away at the end of this period [5]. The World Health Organization (WHO) has indicated that approximately 10 million new cases are reported each year, and globally, more than 55 million people

live with dementia, with over 60% of these residing in low and middle-income countries. Dementia is currently the seventh leading cause of death and leads to disability and dependency among the elderly population worldwide [6].

AD is one of the most common diseases in elderly individuals, characterized by a gradual slowing and deterioration of cognitive abilities. Early symptoms of the disease include difficulties in recent memory recall, word-finding difficulties, repetition, multitasking impairments, and mood or behavioral changes. As the disease progresses, individuals may experience more pronounced memory loss, lose simple functions such as dressing or bathing, encounter difficulties in language and comprehension, exhibit aggression, engage in wandering behavior, and even experience hallucinations. The severity of symptoms corresponds to the extent of nerve cell damage in the brain [7–9].

The rapid and accurate diagnosis of AD is crucial for effective disease management and treatment. Recently, CNN models have achieved notable success in various fields, including healthcare services [10–16]. Despite their success, current methodologies often face constraints such as high computational requirements and inefficiencies when dealing with imbalanced datasets. These challenges highlight the need for innovative approaches that are both effective and resource-efficient.

\* Corresponding author.

E-mail addresses: [cozdemir@siirt.edu.tr](mailto:cozdemir@siirt.edu.tr) (C. Ozdemir), [yahyadogan@siirt.edu.tr](mailto:yahyadogan@siirt.edu.tr) (Y. Dogan).

<https://doi.org/10.1016/j.bspc.2024.106614>

Received 23 August 2023; Received in revised form 29 May 2024; Accepted 25 June 2024

Available online 6 July 2024

1746-8094/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

In this study, we propose an innovative deep-learning methodology using magnetic resonance imaging (MRI) data for the early diagnosis of AD. Instead of transfer learning models with high model capacity, this study focuses on developing a lightweight CNN model by integrating compression and excitation (SE) blocks and a state-of-the-art pooling technique. This model aims to significantly improve diagnostic accuracy in AD classification and reduce training time.

The dataset exhibits class imbalance. In this context, the study is divided into two main sections. Firstly, the performance of innovative methods was investigated using the current dataset. In this regard, the impact of integrating SE blocks, effective in feature representation development, and the innovative Avg-TopK method into the model was examined. We have provided comparisons with traditional methods at this stage. Secondly, the performance of the Synthetic Minority Over-sampling Technique (SMOTE) was explored to address the intrinsic class imbalance in the dataset.

To provide a comprehensive analysis, there are potential benefits to a mixed-methods strategy that combines quantitative data obtained from CNN model performance with qualitative insights from domain experts. This approach can enhance the robustness and interpretability of the findings, allowing for a more nuanced understanding of model effectiveness and the identification of potential areas for improvement. Although the primary focus of this study is on quantitative analysis using CNN models, incorporating expert evaluations and clinical observations is crucial for assessing model performance. In this study, we utilized expert opinions to evaluate the Grad-Cam results, which show the areas the model focuses on, to determine whether the model is focusing on the correct regions. However, the implementation of a mixed-methods strategy is limited by the availability of qualitative data and the scope of the current research design.

Alzheimer's disease not only imposes a significant burden on affected individuals and their families but also places substantial pressure on healthcare systems worldwide. The economic impact of AD is enormous, with billions of dollars spent annually on care and treatment. Early diagnosis is essential to manage the disease effectively, delay progression, and improve the quality of life for patients. Innovations in diagnostic techniques, particularly through the application of deep learning models to neuroimaging data, hold the promise of more accurate, efficient, and early detection of AD. This research is significant because it addresses the limitations of current diagnostic methods, offering a potentially transformative approach that could be integrated into clinical practice. By improving the accuracy and reducing the computational cost of AD diagnosis, this study contributes to better healthcare outcomes and resource allocation.

In addition, the main contributions of the study can be summarized as follows:

- We evaluated the performance of state-of-the-art transfer learning models for Alzheimer's disease classification. This provided insights into the strengths and limitations of these relevant models.
- We utilized SE blocks in our model to enhance feature representation, thus enhancing the model's ability to capture intricate details related to Alzheimer's disease.
- We have replaced traditional pooling layers with the recently proposed Avg-TopK method in our proposed model. This method aids in effectively extracting features crucial for Alzheimer's disease classification.
- To address the difficulties stemming from imbalanced datasets typically present in health data, we utilized the SMOTE technique. This approach enhanced the model's robustness, resulting in a more balanced representation of classes related to Alzheimer's disease.
- Despite its lighter, faster, and simpler nature, the proposed model outperforms current transfer learning models, delivering state-of-the-art results. These findings provide valuable insights that could pave the way for more refined and effective diagnostic tools in the field of neurodegenerative diseases.

## 2. Related works

Magnetic resonance imaging (MRI) is vital for exploring pathological brain changes in AD [17]. Over the past decades, combining neuroimaging data with machine learning (ML) techniques has shown promise for individualized AD diagnosis [18,19]. However, traditional methods requiring manual feature extraction are time-consuming and subjective. Deep learning, especially convolutional neural networks (CNNs), has transformed this field by automating feature extraction, thus enhancing efficiency and achieving significant success in AD diagnosis [20,21]. Studies using end-to-end CNNs for AD classification utilize various input types: 2D slice-level, 3D patch-level, ROI-based, and 3D subject-level [22].

In the 2D slice-level category, 2D CNN architectures use a series of 2D slices from the 3D MRI volume [23–31]. This method integrates well with established CNNs known for their efficacy in natural image classification, allowing for transfer learning. Additionally, it increases the training data volume by extracting multiple slices from one 3D image. Several studies exemplify this approach. [23] used the LeNet-5 model with the ADNI dataset, achieving 96.86% accuracy in distinguishing Alzheimer's from normal controls. [24] adapted VGG16 and InceptionV4 architectures through transfer learning, finding that pre-trained models outperformed those trained from scratch. [25] developed a multi-modal early stopping algorithm (ESA) using GoogleNet, ResNet-18, and ResNet-152, achieving 98.88% accuracy. [27] used cross-sectional MRI scans with an ESA model to diagnose AD and predict conversion from mild cognitive impairment to AD, achieving over 98% accuracy. These studies highlight the potential of deep learning, transfer learning, and innovative architectures in improving AD diagnostic accuracy. However, the main limitation of the 2D slice-level approach is that it analyzes each slice independently, not fully leveraging the 3D nature of MRI scans.

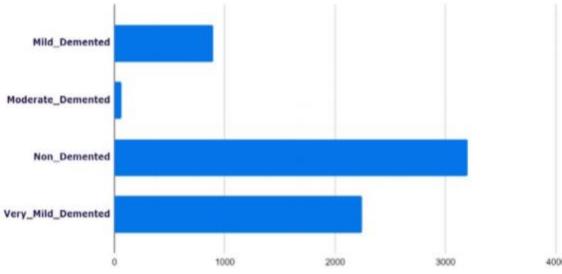
In the 3D patch-level category, to incorporate 3D information, the input consists of sequences of 3D patches extracted from an image [32–34]. This method can generate a larger sample size, as samples are based on the number of patches rather than subjects. Benefits include decreased memory usage and fewer learning parameters, provided the same network is used for all patches. [32,33] utilized large patches, extracting 27 overlapping 3D patches sized  $50 \times 41 \times 40$  voxels to cover the entire MR image volume ( $100 \times 81 \times 80$  voxels). Each patch was used to train individual convolutional networks. Subsequently, an ensemble CNN was trained to make decisions at the subject level, partly initialized with weights from the previously trained CNNs. [33] adopted the same architecture as [32] but enhanced it with a fusion of PET and MRI inputs. On the other hand, [34] employed smaller patches ( $32 \times 32 \times 32$ ). To address potential discrepancies between patches taken at the same coordinates for different subjects, they clustered their patches using k-means without nonlinear registration. One CNN was then trained per cluster, and the features obtained at the cluster level were assembled similarly to [32,33]. The main limitation of this approach is its complexity, as each patch position requires a separate network, which is then integrated and retrained at various representation levels, such as region-level and subject-level.

In the ROI-based category, methods focus on specific brain regions known to be informative, addressing the issue of uninformative patches in 3D patch-level approaches. This reduces the framework's complexity by using fewer inputs for training. The hippocampus, affected early in AD, is a common ROI. Studies vary in defining the hippocampal ROI. Aderghal et al. [35–37] conducted multiple studies employing linear registration and delineating a 3D bounding box encompassing all hippocampal voxels based on segmentation with the AAL atlas. Adopting a "2D +  $\epsilon$  approach", these studies crafted patches comprising three neighboring 2D slices within the hippocampus. However, the entire region is not well covered because they only use one or three patches per patient. In their initial study [35], only the sagittal view was utilized, with a classification of one patch per patient. In the subsequent

**Table 1**

The number of examples for each class in the training, validation, and test sets.

	Training	Validation	Test	Total
Mild_Demented	672	134	90	896
Moderate_Demented	48	9	7	64
Non_Demented	2400	480	320	3200
Very_Mild_Demented	1680	336	224	2240

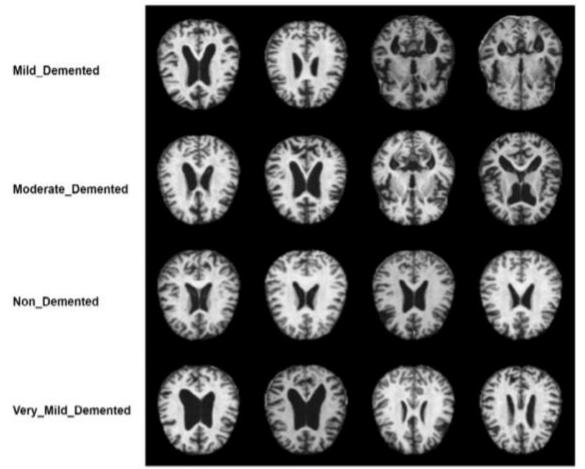
**Fig. 1.** The distribution of the number of examples across classes in the dataset.

study [36], all views, i.e., sagittal, coronal, and axial, were employed to generate patches. Consequently, three patches were generated per subject, and three networks were trained for each view before being fused. Aderghal et al.'s latest study in 2018 focused on transfer learning from anatomical MRI to diffusion MRI.

In the 3D subject-level category, entire MRI volumes are classified at the subject level [38–40]. However, these studies require high-performance computing resources. This approach fully integrates spatial information. When examining the studies conducted in this context, it is observed that some utilize original transfer learning models, while others have re-adapted them to fit all MRI data. In contrast, [41, 42] stated that they investigated the impact of pre-training with an autoencoder and concluded that it improved their results. However, this method risks overfitting due to the small number of samples compared to the number of parameters, as each subject contributes only one sample, and datasets typically include only a few hundred to thousands of subjects.

### 3. Dataset

In this study, we utilized an Alzheimer's disease brain MRI dataset containing 6,400 images [43]. These images have been divided into four distinct classes: "Mild Demented", "Moderate Demented", "Non-Demented", and "Very Mild Demented". To ensure uniformity, all images were standardized to a resolution of  $128 \times 128$  pixels. Fig. 1 depicts the number of examples per class in the dataset. The dataset was initially divided into two parts, with 90% allocated for training and 10% for testing. Subsequently, 85% of the training set was used for model training, while the remaining 15% was utilized for validation during parameter optimization. We provided randomly generated example images for each class in the dataset in Fig. 2. As shown in Table 1, there appears to be a notable imbalance in class distribution, which could adversely affect model training and subsequent performance. This inherent asymmetry has the potential to compromise the robustness of the model. To overcome this challenge and address data imbalance, we applied the SMOTE method to the dataset. In this context, an investigation of the model's performance before and after the use of SMOTE was performed. This analytical approach has allowed for a full assessment of the effects of generating new synthetic examples on model effectiveness.

**Fig. 2.** Representative images from each class in the dataset.

#### 3.1. Synthetic Minority Over-sampling Technique (SMOTE)

The SMOTE is a technique used to balance datasets with a small number of samples in certain classes. It achieves this by generating new synthetic examples, thereby reducing the class imbalance. By augmenting the representation of classes with fewer instances in the dataset, SMOTE helps to create a more balanced representation of each class during model training, potentially leading to improved classification performance. The SMOTE algorithm identifies the nearest neighbors for each minority class example using the k-nearest neighbors (k-NN) algorithm and generates new synthetic examples by using the vector difference between a randomly selected neighbor and the minority class example. This process is repeated until the representation of the minority class reaches the desired level. Mathematically, a new synthetic example  $x_{new}$  between a minority class instance  $x_i$  and one of its randomly selected neighbors  $x_{nn}$  is generated using the following formula:

$$x_{new} = x_i + \lambda \times (x_{nn} - x_i) \quad (1)$$

Where  $\lambda$  is a random coefficient between 0 and 1, which allows the generated synthetic examples to exhibit diversity among minority class instances. With this methodology, a total of 12,800 images were obtained, with 3,200 for each class, enhancing the representation of each class and creating a more balanced dataset. The advantages provided by SMOTE include enhancing model performance on the minority class and effectively reducing class imbalance. However, disadvantages such as the tendency of the method to cause overfitting by oversampling and the inability of synthetic examples to fully reflect the real data distribution also exist.

The problem of class imbalance affects many real-world applications, where most examples belong to the majority class and fewer to the minority class. Various methods, such as SMOTE, cost-sensitive analysis, and ensemble methods like bagging and boosting, have been proposed to address this issue. According to [44], these approaches have been extensively researched, providing a comprehensive overview of methods to improve classification in imbalanced datasets.

#### 4. Proposed methodology

In this study, we aimed to create a simple, low-capacity, and high-performance model for Alzheimer's disease classification using current deep learning methods. In this context, we comprehensively investigated the Alzheimer's disease classification performance of 14 transfer learning models commonly used in the literature. These models include

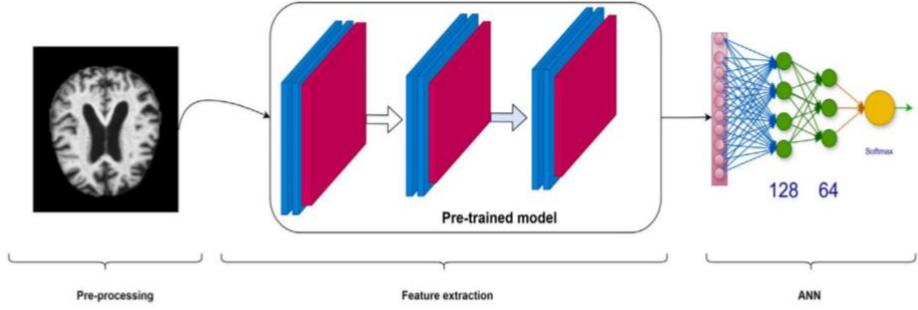


Fig. 3. The architecture created for performance comparison of transfer learning models.

InceptionV3, ResNet101, ResNet101V2, ResNet152V2, ResNetRS50, DenseNet121, DenseNet169, DenseNet201, InceptionResNetV2, RegNetX002, RegNetX320, MobileNetV2, MobileNetV3Large, EfficientNetB0, EfficientNetB7, and NASNetLarge. Transfer learning is the utilization of knowledge acquired from one task to solve a different but related task. This method is commonly employed when there is an insufficient amount of data available for a specific task, or when the time or computational resources required to train a model for a task are limited. In the transfer learning approach, a pre-trained model (typically a deep learning model trained on a large dataset) is obtained, and its feature extraction layers are extracted. Subsequently, these extracted features are utilized in a new model (often a smaller neural network) that will be employed to solve a new task. In this manner, the pre-trained model learns feature extraction for the new task. Despite exhibiting high success rates in various problem domains, these models generally possess high capacity and may pose challenges on devices with limited resources. These transfer learning models have consistently demonstrated exceptional performance across diverse domains, underscoring their effectiveness in feature extraction and representation.

In Fig. 3, we provide the architecture used to compare the performance of transfer learning models for AD classification. Firstly, the images in the dataset were resized to dimensions of  $128 \times 128$ . The dataset was divided into three separate partitions for training, validation, and testing, each consisting of batches of 64 images processed at once. Additionally, the data was rescaled to ensure appropriate value ranges before being fed into the model. Subsequently, image features were extracted using transfer learning models, capitalizing on pre-trained model knowledge to extract features from the new dataset. The extracted features were channeled into an artificial neural network architecture comprising two layers, with 128 and 64 neurons each. To counteract the risks of overfitting, dropout techniques were integrated after each layer. Thus, after data preprocessing, Alzheimer's disease classification was executed using transfer learning models and an artificial neural network.

In Fig. 4, we present our proposed model, strategically designed to integrate SE Blocks and Avg-TopK methods to optimize feature learning and enhance classification performance. The proposed model consists of four blocks, each comprising Convolution, Activation (ReLU), and Pooling layers. Dropout layers were strategically placed between these blocks to enhance generalization and prevent overfitting. Additionally, an SE block was applied to the output of the fourth block to intensify feature map importance and promote effective learning. The channel-level SE block enriched the significance of feature maps. Finally, the combined outputs from the fourth block and the SE block [45] were channeled into an ANN. The initial layer incorporates a convolution layer, essential for feature extraction from input data. These learnable filters discern distinctive features, generating feature maps that encapsulate pertinent input features. Subsequently, feature maps are subjected to the ReLU activation function, injecting non-linearity to

capture intricate feature relationships. Pooling layers reduce feature map dimensions. Alongside conventional pooling layers, such as maximum and average pooling, the efficacy of the Avg-TopK pooling layer, introduced by Ozdemir [46], was assessed and compared.

#### 4.1. Squeeze and excitation block

Squeeze and Excitation (SE) blocks were introduced by Hu et al. [45] as a new method. Fig. 5 illustrates the working procedure of SE blocks and the layers employed to construct such blocks. These blocks are capable of being embedded within large CNNs and are instrumental in establishing attention between features. The primary purpose of an SE block is to emphasize the most important features while suppressing those that are relatively less significant. By incorporating SE blocks into a deep learning model, it becomes possible to enhance its performance by mitigating the influence of irrelevant or insignificant features that could otherwise introduce noise and adversely impact the overall performance of the model.

As depicted in Fig. 5, the inclusion of the ratio term in the SE blocks serves to manage both the computational requirements and the capacity of the blocks. These SE blocks, through their feature recalibration mechanism, significantly improve the representation quality of CNN models for image processing. By incorporating global information, they enable the model to emphasize important features while de-emphasizing less significant ones.

The squeeze block plays a crucial role in reducing the data size while also expressing the features in a more distinctive manner. It effectively compresses the information without losing essential characteristics, leading to more efficient processing.

On the other hand, the Excitation block introduces an attention mechanism, which allows the model to focus on specific regions or features. By selectively highlighting relevant information, the model can improve its discriminative ability and improve performance in tasks that require precise localization or feature extraction.

#### 4.2. Avg-TopK pooling layer

The Avg-TopK [46] pooling method is a new pooling method that combines the strengths of average pooling and maximum pooling techniques. It takes a weighted average by selecting the K highest pixels from the pooling layer. This allows the model to reduce noise while retaining the most important features.

Fig. 6 presents a visual comparison of the Max, Avg, and Avg-TopK pooling methods on a dataset. In particular, the Avg-TopK method yields values that lie between those obtained from the Max and Avg methods. This signifies that the Avg-TopK method considers the highest energy values, ensuring their inclusion while also incorporating the average values, thus striking a balance between preserving important information and accounting for noise.

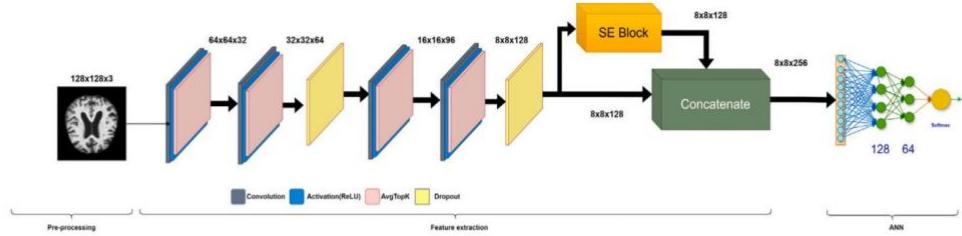


Fig. 4. Proposed model.

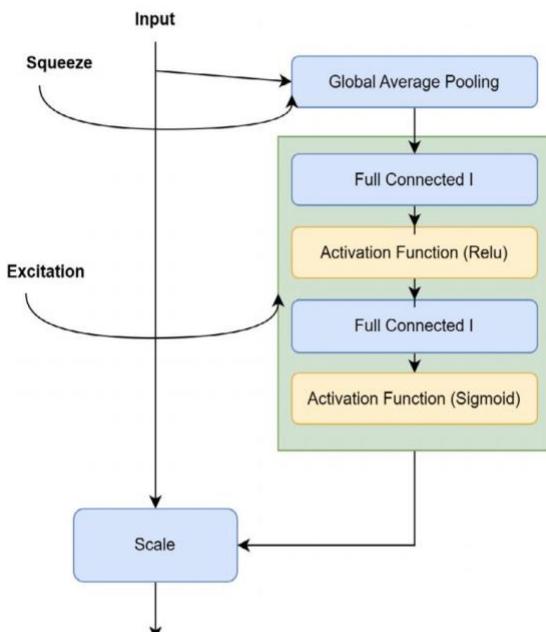


Fig. 5. Squeeze and excitation block.

Table 2  
Hyperparameters.

Batch Size	64
Learning Rate	0.001
Input Size	128 × 128x3
Optimizer	Adam
Loss Function	Sparse Categorical Cross-Entropy Loss
Epochs	200
Learning Rate Decay Factor	0.8

Table 3  
Performance of transfer learning models.

	Test accuracy (%)	Precision score (%)	Recall score (%)	F1 score (%)
InceptionV3	96.72	96.72	96.72	96.72
ResNet101	97.97	97.97	97.97	97.97
<b>ResNet101V2</b>	<b>98.28</b>	98.28	98.28	98.28
ResNet152V2	94.85	94.85	94.85	94.85
ResNetRS50	97.97	97.97	97.97	97.97
DenseNet121	97.50	97.50	97.50	97.50
DenseNet169	97.82	97.82	97.82	97.82
<b>DenseNet201</b>	<b>98.13</b>	98.13	98.13	98.13
InceptionResV2	97.82	97.82	97.82	97.82
RegNetX002	92.51	92.51	92.51	92.51
RegNetX320	95.48	95.48	95.48	95.48
MobileNetV2	40.87	40.87	40.87	40.87
MobileNetV3	47.89	47.89	47.89	47.89
EfficientNetB0	95.79	95.79	95.79	95.79
EfficientNetB7	97.04	97.04	97.04	97.04
NASNetLarge	49.92	49.92	49.92	49.92
<b>ProposedModel</b>	<b>98.9</b>	98.9	98.9	98.9

the sparse categorical cross-entropy loss function to update the model weights. The hyperparameters employed in the training of the overall model are summarized in Table 2.

## 5. Experimental and results

In this section, we provided the experimental studies conducted to evaluate the performance of the proposed model and existing models in the literature. Initially, we conducted experimental studies to examine the performance of 14 different transfer learning models commonly used in the literature on the relevant dataset. In this scope, we utilized the architecture shown in Fig. 3. We also presented the hyperparameter values used in all experimental iterations in Table 2. At this stage, experiments were conducted without using SMOTE to address the inherent imbalance in the dataset. The purpose of these preliminary experiments is to measure the performance of the models under basic conditions without using any data balancing technique. The results of these initial experiments are summarized in Table 3, providing information on the accuracy, precision, recall, and F1 metrics of the models.

Upon examining the model performances presented in Table 3, we observed that among the 14 different transfer learning models, ResNet101V2 (98.28%), DenseNet201 (98.13%), and InceptionResNetV2 (97.82%) models achieved higher performance. On the other

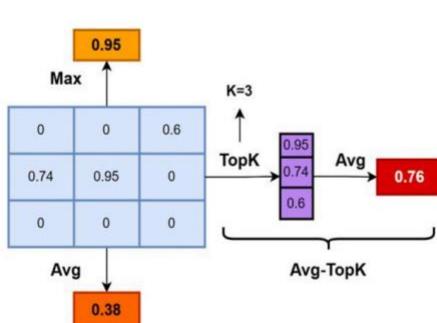


Fig. 6. Avg-TopK, Max and Average pooling with 3 × 3 pooling method.

### 4.3. Training details

In our training process, we employ the Adam optimizer with a batch size of 64 and conduct training from scratch for 200 epochs, setting the learning rate at 0.001. To dynamically adapt the learning rate based on validation loss, we implement the ReduceLROnPlateau strategy. This strategy entails reducing the learning rate by a factor of 0.8 when the minimum validation loss ceases to improve after 10 epochs. Ensuring the preservation of the best model, we utilize model checkpointing based on validation loss. Throughout the training iterations, we utilize

**Table 4**  
Trained parameter values of the models.

Model	Total params	Trainable params	Non-trainable params
ResNet101V2	42,626,560	42,528,896	97,664
DenseNet201	18,321,984	18,092,928	229,056
Our Model	2,299,652	2,299,652	0

hand, MobileNetV2 (40.87%), MobileNetV3Large (47.89%), and NASNetLarge (49.92%) models exhibited the lowest performance on the test set. MobileNet and its derivatives are generally low-capacity models. Our reason for examining the performance of these models was to evaluate the performance of low-capacity models. However, despite their reputation for computational efficiency, these models (MobileNetV2, MobileNetV3Large, NASNetLarge) do not seem suitable for the classification task of Alzheimer's disease. The proposed model demonstrates superior performance compared to the transfer learning models. Data balancing and Avg-Topk operations have not yet been integrated into the relevant model. In this model, the performance of a simple model architecture with SE block integration has been examined. Additionally, a comparison between the proposed simple model and the best-performing models from these transfer learning models in terms of total parameters, trainable parameters, and non-trainable parameters is provided in **Table 4**. Upon examining the parameter counts, it is evident that the proposed model's lightweight and fast nature. This contemporary technique we propose enhances CNN performance by recalibrating channel-based feature responses. In our proposed model, a global average pooling layer precedes the classification layers. This layer captures comprehensive global information, forming the basis for channel weight computation and feature map rescaling. This recalibration process strengthens informative channels while weakening less important ones, enhancing the robustness of feature representation. The ResNet101V2 model, one of the highest-performing models, has the highest total parameter count. The number of parameters significantly affects the model's training time, memory requirements, and inference. Compared to these models, the proposed model has substantially fewer total parameters. In comparison to the ResNet101V2 model, the proposed model exhibits a 94.59% reduction in trainable parameters and a significant 87.44% reduction in parameters compared to the DenseNet201 model. This demonstrates that our model is lighter and less complex.

In continuation of the study, experimental investigations were conducted utilizing the SMOTE technique to address data imbalance. Through the integration of the SMOTE methodology, the dataset's dimensionality was expanded, and the resulting synthetic dataset was subsequently used to retrain the proposed model. Accordingly, models were retrained using the configurations that exhibited the highest performance. The obtained results are presented in **Table 5**. It is noted that with the SMOTE method, the performance of the ResNet101V2 model increased from 98.28% to 98.91%, while the performance of the DenseNet201 model decreased from 98.13% to 97.11%, and the performance of the proposed model increased from 98.9% to 99.61%. While the SMOTE method improved the performance of the ResNet101V2 and proposed models, it adversely affected the performance of the DenseNet201 model.

In a further investigation, the impact of using different pooling layers in the proposed architecture on the overall model performance was investigated. In this context, the currently employed max-pooling layers were replaced with average pooling and the recently proposed Avg-Topk pooling layers to examine their performance. The primary aim of this research was to observe the potential consequences of changes in pooling layers on architectural dynamics. Upon examination of **Table 4**, it was observed that when the max pooling layers in the model were replaced with average pooling layers, the performance of the proposed model decreased from 99.61% to 98.12%. However,

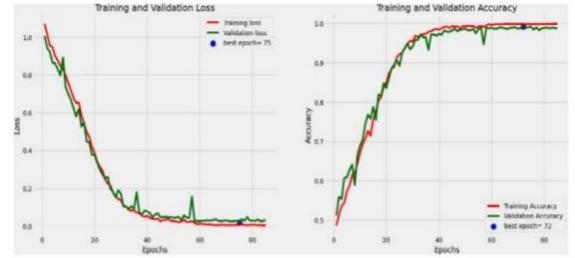


Fig. 7. Accuracy and loss visualization for the proposed model.

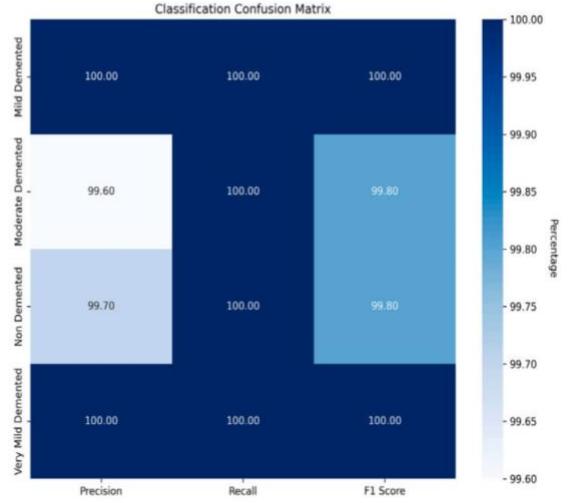


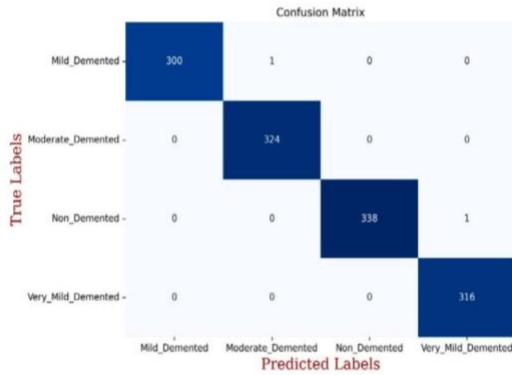
Fig. 8. Classification report for the proposed model.

when the pooling layers in the proposed model were replaced with the contemporary method of Avg-Topk, the model's performance increased from 99.61% to 99.84%, achieving state-of-the-art results. In Avg-TopK usage, a value of  $k = 3$  was employed; in other words, the average of the top 3 pixel values within the respective pooling area was utilized. **Fig. 7** illustrates the training and validation loss and accuracy graphs obtained during the training phase of the final model, namely SMOTE + SE block + Avg-TopK. In **Fig. 8**, a comprehensive classification report is provided for the proposed model. This report offers a clear explanation of the performance of our final model when subjected to a dataset encompassing four different classes associated with Alzheimer's disease, i.e., Non-Demented, Mild Demented, Moderate Demented, and Very Mild Demented. The visualization presents a detailed breakdown of various performance metrics, enabling a clear understanding of the model's discriminative capabilities across different classes. These findings unequivocally highlight the effectiveness of the proposed model in distinguishing between different disease stages and its capability to provide diagnostic evaluations. In **Fig. 9**, the visual representation of the confusion matrix between predicted and true labels is provided. This matrix serves as an illustrative depiction of the distribution dynamics of predictions and true labels, thereby demonstrating the diagnostic classification of the proposed model. According to the confusion matrix shown in **Fig. 9**, the proposed model misclassified 1 example from the Mild Demented class, correctly classified all examples from the Moderate Demented class, misclassified 1 example from the Non-Demented class, and correctly classified all examples from the Very Mild Demented class. These results indicate that the classification model demonstrated overall high accuracy rates and effectively classified nearly all examples. In **Fig. 10**, several randomly selected examples from the test set are depicted to illustrate how the proposed model

**Table 5**

The impact of employing the SMOTE technique and different pooling layers on model performance.

Model	Test accuracy (%)	Precision score (%)	Recall score (%)	F1 score (%)
ResNet101V2	98.91	98.94	98.89	98.90
DenseNet201	97.11	97.41	97.21	97.21
Model-Avg	98.12	98.12	98.12	98.12
Model-Max	99.61	99.61	99.61	99.61
<b>Model-Avg-TopK</b>	<b>99.84</b>	<b>99.84</b>	<b>99.84</b>	<b>99.84</b>

**Fig. 9.** Confusion matrix for the proposed model.**Fig. 10.** Actual vs. model predicted for random sample images.

predicts them. It is evident from these images that the proposed model performs classification with high accuracy.

By conducting a thorough comparison with results documented in the extant literature, an evaluation of the effectiveness of the proposed model is conducted. This involves an exhaustive literature review, where performance metrics obtained from studies conducted on the identical dataset are juxtaposed with the scores achieved through our innovative model. The findings are summarized in **Table 6**.

In **Table 6**, we compared our proposed model with studies conducted on the same dataset. The capacity of the model to surpass established accuracy criteria in the current literature underscores its effectiveness, particularly in achieving high sensitivity and reliability in

**Table 6**

Comparing the performance of the proposed model with studies conducted using the same dataset in the literature.

	Accuracy (%)
[9]	96.59
[47]	96.35
[48]	99.38
[49]	97.2
[50]	98.70
[51]	99.6
[52]	91.21
[53]	95
[54]	97.6
[55]	83.47
[56]	79.12
[57]	90.23
[58]	98.4
<b>Proposed Model</b>	<b>99.84</b>

the early detection and diagnosis of Alzheimer's disease. The significant increase in accuracy achieved by the proposed model emphasizes its innovative approach and departure from traditional methodologies. This notable progress positions the model as an advanced solution in reshaping and influencing the classification of Alzheimer's disease. These findings serve as evidence that the model has reached state-of-the-art results in the field.

To observe which areas of the images the proposed model focuses on, we utilized the Grad-CAM method. Grad-CAM is a visualization technique employed to make the predictions of deep learning models more transparent and interpretable. This method helps identify the features that the model pays attention to and highlights the visual characteristics that support the classification decisions. In Grad-CAM, regions, where the model focuses less, are represented by blue and green tones, while areas that are highly influential in classification are represented by yellow and red tones. Some Grad-CAM results for random images are provided in **Fig. 11**. Evaluation of the Grad-CAM heat maps by two expert radiologists revealed that the model primarily prioritized cortical regions and exhibited less focus on central parenchyma areas in T1-weighted axial magnetic resonance images. The cortical regions pertain to the outer peripheral section of the cerebral cortex, where alterations associated with Alzheimer's disease manifest prominently. Within these regions, the presence of gray matter, cerebrospinal fluid, and white matter is highlighted, with particular emphasis placed on changes attributed to cortical atrophy. The expert observations on the model's heatmap indicate the following: "The model's inclination towards cortical regions implies its emphasis on cortical atrophy-related changes, which serve as a significant marker for Alzheimer's disease diagnosis. Cortical atrophy denotes a condition characterized by diminished brain parenchyma, widened sulcus, and increased cerebrospinal fluid. Consequently, the model's focus on cortical regions suggests its capability to accurately discern and classify pathological processes specific to Alzheimer's disease". These findings indicate the potential of the model to serve as a valuable tool for early detection of Alzheimer's disease and its potential contribution in various clinical applications.

In **Table 7**, we provide the results of an ablation study examining the impact of various methods on the model's performance, assessed using metrics such as Test Accuracy, Precision, Recall, and F1 Score. Utilizing only the AvgTopK pooling layer, the model achieves 97.9% across all metrics, highlighting its significant contribution when used independently. When configured with only SE blocks, the model attains 96.9%, indicating that SE blocks improve performance but are slightly less effective than the AvgTopK layer. Integrating SE blocks with the AvgTopK layer results in a notable enhancement, achieving 98.9% across all metrics, suggesting a synergistic effect. Incorporating SE blocks with the SMOTE further improves performance to 99.24%,

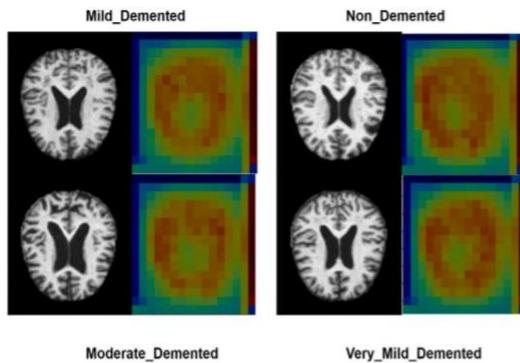


Fig. 11. The Grad-CAM results showing the areas the proposed model focused on in randomly selected images.

**Table 7**  
Ablation study results demonstrating the impact of various techniques on the model's performance.

Model	Test accuracy (%)	Precision score (%)	Recall score (%)	F1 score (%)
AvgTopK	97.9	97.9	97.9	97.9
SE Blocks	96.9	96.9	96.9	96.9
SE Blocks+AvgTopK	98.9	98.9	98.9	98.9
SE Blocks+SMOTE	99.24	99.24	99.24	99.24
AvgTopK+SMOTE	99.34	99.34	99.34	99.34
SE Blocks+AvgTopK+SMOTE	99.84	99.84	99.84	99.84

underscoring the efficacy of SMOTE in mitigating data imbalance. Combining the AvgTopK layer with SMOTE results in 99.34%, closely matching the performance with SE blocks and SMOTE. The highest performance is achieved when SE blocks, AvgTopK pooling layer, and SMOTE are integrated, with the model attaining 99.84% across all metrics. This comprehensive integration yields optimal performance, showcasing the substantial contributions of each component to the overall efficacy of the model. This study elucidates each component's critical role, validating the proposed model's innovative architecture in Alzheimer's disease classification and underscoring its potential as a robust diagnostic tool.

## 6. Conclusion

Alzheimer's disease is a progressive degenerative disease in the brain and is one of the most common types of dementia. Given that there is no definitive treatment for Alzheimer's disease, the ability to predict its early stages is of great importance in mitigating its progression towards broader dementia symptoms. In this study, a model is proposed that goes beyond previous methodologies by successfully applying deep learning models in the classification and early prediction of Alzheimer's disease. Comprehensive experimental research was conducted, covering 14 different transfer learning models commonly used in the literature, and ultimately, a model structure for AD classification was proposed using current techniques. This innovative model incorporates state-of-the-art SE blocks and specialized neural network layers that effectively enhance the representation of important details in feature maps. Impressively, the proposed model achieved a commendable success rate of approximately 98.9% on an imbalanced dataset, thus demonstrating strong performance even amidst natural data imbalances. Subsequently, integration of the new pooling methodology, Avg-TopK, and the application of the SMOTE technique to address data imbalance resulted in an accuracy rate of 99.84% on the test dataset. This significant performance improvement highlights the effectiveness of the model in categorizing examples of Alzheimer's

disease, thereby solidifying its position as an innovative and state-of-the-art solution for early diagnosis and prediction. When examining the classification report, it is evident that the model demonstrates notable proficiency in distinguishing between different classes. Only one misclassification occurred among the two classes, while all examples from the remaining two classes were correctly classified. This observation emphasizes the effectiveness of the model in discerning different classes with considerable precision.

The results of the ablation study presented in Table 7 validate the critical role of each component in enhancing the model's performance. The integration of SE blocks, AvgTopK pooling layer, and SMOTE demonstrates the substantial contributions of each component, resulting in the highest performance observed, thus underscoring the model's innovative architecture and its potential as a robust diagnostic tool.

In the conducted study, the utilization of Grad-CAM results to visualize the areas of focus of the proposed model within images emerges as a crucial aspect. The Grad-CAM heat map, assessed by expert radiologists, has revealed the model's focus on cortical regions within T1-weighted axial magnetic resonance images. The model's emphasis on alterations associated with cortical atrophy highlights its capability to discern and categorize pathological processes closely linked with diagnostic indicators of Alzheimer's disease. These findings validate the model's potential as a valuable tool for the early detection of Alzheimer's disease, thus making significant contributions to various clinical applications.

The proposed model has another advantage in that its capacity, or in other words, the number of parameters, is low. This contributes to high computational efficiency, making it faster and lighter compared to similar models. As a result, the model can effectively operate even with limited resources and becomes a significant choice, especially in devices where resources are constrained. The compact size of the model and its high computational efficiency position it as the most suitable option for integration into clinical settings. Additionally, the model's remarkable ability to learn complex and specific features contributes to its superior accuracy. Finally, even when trained on limited datasets, the model consistently achieves high accuracy, thereby enhancing its effectiveness in early diagnosis of Alzheimer's disease.

As illustrated in Table 6, our model demonstrates superior performance with an accuracy rate of 99.84% compared to studies conducted using the same dataset in the existing literature. This significant advancement underscores the innovative approach of our model and its departure from traditional methodologies. The findings obtained indicate that the model offers an advanced solution for the classification of Alzheimer's disease and has achieved state-of-the-art results in the field. Consequently, it can be concluded that the proposed model has the potential to reshape and influence current classification methods in this domain.

As a result, this study presents compelling evidence for effectively implementing deep learning models in the classification and early prediction of Alzheimer's disease, replacing previous methodologies and yielding significant results. However, further research is needed, particularly to rigorously assess the accuracy of the model and its generalizability across larger datasets. Additionally, investigating the integration of the model into clinical settings and evaluating its performance under real-world conditions is of paramount importance for future research endeavors.

## CRediT authorship contribution statement

**Cuneyt Ozdemir:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis. **Yahya Dogan:** Writing – review & editing, Writing – original draft, Visualization, Methodology.

## Declaration of competing interest

There is no 'Conflict of Interest' in the publication of the manuscript

## Data availability

Data available in a public (Kaggle) repository that does not issue datasets with DOIs (non-mandated deposition). The dataset that supports the findings of the study is available from <https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images>.

## Acknowledgments

In this study, the expertise and experience of expert radiologists Dr. Mehmet Ali GEDIK and Dr. Şahinde ATLANOĞLU in interpreting Grad-Cam heatmaps to assess the accuracy of the model's focused areas have greatly contributed to enhancing the quality, reliability, and validity of this research. We would like to express our sincere gratitude to them for their valuable contributions to this study.

## References

- [1] F.J.M. Shamrat, S. Akter, S. Azam, A. Karim, P. Ghosh, Z. Tasnim, K.M. Hasib, F. De Boer, K. Ahmed, AlzheimerNet: An effective deep learning based proposition for Alzheimer's disease stages classification from functional brain changes in magnetic resonance images, *IEEE Access* 11 (2023) 16376–16395.
- [2] 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 (6) (2023) 714.
- [3] The top 10 causes of death, 2020, URL: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>, publisher: World Health Organization.
- [4] T. Illakiya, R. Karthik, Automatic detection of Alzheimer's disease using deep learning models and neuro-imaging: current trends and future perspectives, *Neuroinformatics* 21 (2) (2023) 339–364.
- [5] A. Alzheimer, R.A. Stelzmann, H.N. Schnitzlein, F.R. Murtagh, An English translation of Alzheimer's 1907 paper, "Über eine eigenartige Erkrankung der Hirnrinde", *Clin. Anat. (New York, NY)* 8 (6) (1995) 429–431.
- [6] Dementia, 2023, URL: <https://www.who.int/news-room/fact-sheets/detail/dementia>, publisher: World Health Organization.
- [7] J. Xu, C. Qiu, Worldwide economic costs and societal burden of dementia, in: R. Pernecczyk (Ed.), in: *Biomarkers for Precclinical Alzheimer's Disease*, vol. 137, Springer New York, New York, NY, 2018, pp. 3–13, [http://dx.doi.org/10.1007/978-1-4939-7674-4\\_1](http://dx.doi.org/10.1007/978-1-4939-7674-4_1), URL: [http://link.springer.com/10.1007/978-1-4939-7674-4\\_1](http://link.springer.com/10.1007/978-1-4939-7674-4_1).
- [8] R.D. Terry, E. Masliah, Synaptic pathology in the pathogenesis of Alzheimer dementia, in: A. Carlsson, P. Riederer, H. Beckmann, T. Nagatsu, S. Gershon, K.A. Jellinger, K.A. Jellinger, G. Ladurner, M. Windisch (Eds.), *New Trends in the Diagnosis and Therapy of Alzheimer's Disease*, Springer Vienna, Vienna, 1994, pp. 1–8, [http://dx.doi.org/10.1007/978-3-7091-9376-1\\_1](http://dx.doi.org/10.1007/978-3-7091-9376-1_1), URL: [http://link.springer.com/10.1007/978-3-7091-9376-1\\_1](http://link.springer.com/10.1007/978-3-7091-9376-1_1).
- [9] P.S. Sisodia, G.K. Ameta, Y. Kumar, N. Chaplot, A review of deep transfer learning approaches for class-wise prediction of Alzheimer's disease using MRI images, *Arch. Comput. Methods Eng.* 30 (4) (2023) 2409–2429, <http://dx.doi.org/10.1007/s11831-022-09870-0>, URL: <https://link.springer.com/10.1007/s11831-022-09870-0>.
- [10] K.M. Hasib, A. Tanzim, J. Shin, K.O. Faruk, J. Al Mahmud, M.F. Mridha, Bmnet-5: A novel approach of neural network to classify the genre of bengali music based on audio features, *IEEE Access* 10 (2022) 108545–108563.
- [11] M. Atas, C. Özdemir, İ. Ataş, B. Ak, E. Özeroglu, Biometric identification using panoramic dental radiographic images with few-shot learning, *Turk. J. Electr. Eng. Comput. Sci.* 30 (3) (2022) 1115–1126, <http://dx.doi.org/10.55730/1300-0632.3830>, URL: <https://journals.tubitak.gov.tr/elektrik/vol30/iss3/39>.
- [12] T. Illakiya, R. Karthik, A dimension centric proximate attention network and swin transformer for age-based classification of mild cognitive impairment from brain MRI, *IEEE Access* 11 (2023) 128018–128031.
- [13] Ş. Külcü, Y. Doğan, Deep learning based gender identification using ear images, *Trait. Signal* 40 (4) (2023).
- [14] K.M. Hasib, M.R. Islam, S. Sakib, M.A. Akbar, I. Razzak, M.S. Alam, Depression detection from social networks data based on machine learning and deep learning techniques: An interrogative survey, *IEEE Trans. Comput. Soc. Syst.* (2023).
- [15] C. Ozdemir, M.A. Gedik, Y. Kaya, Age estimation from left-hand radiographs with deep learning methods, *Trait. Signal* 38 (6) (2021) 1565–1574, <http://dx.doi.org/10.18280/ts.380601>, URL: <https://www.ieta.org/journals/ts/paper/10.18280/ts.380601>.
- [16] T. Illakiya, R. Karthik, Alzheimer's Disease Neuroimaging Initiative, et al., A deep feature fusion network with global context and cross-dimensional dependencies for classification of mild cognitive impairment from brain MRI, *Image Vis. Comput.* (2024) 104967.
- [17] M. Ewers, R.A. Sperling, W.E. Klunk, M.W. Weiner, H. Hampel, Neuroimaging markers for the prediction and early diagnosis of Alzheimer's disease dementia, *Trends Neurosci.* 34 (8) (2011) 430–442.
- [18] F. Falahati, E. Westman, A. Simmons, Multivariate data analysis and machine learning in Alzheimer's disease with a focus on structural magnetic resonance imaging, *J. Alzheimer's Dis.* 41 (3) (2014) 685–708.
- [19] S. Rathore, M. Habes, M.A. Ifikhah, A. Shacklett, C. Davatzikos, A review on neuroimaging-based classification studies and associated feature extraction methods for Alzheimer's disease and its prodromal stages, *NeuroImage* 155 (2017) 530–548.
- [20] M.B.T. Noor, N.Z. Zenia, M.S. Kaiser, S.A. Mamun, M. Mahmud, Application of deep learning in detecting neurological disorders from magnetic resonance images: a survey on the detection of Alzheimer's disease, Parkinson's disease and schizophrenia, *Brain Inform.* 7 (2020) 1–21.
- [21] F. Li, L. Tran, K.-H. Thung, S. Ji, D. Shen, J. Li, A robust deep model for improved classification of AD/MCI patients, *IEEE J. Biomed. Health Inform.* 19 (5) (2015) 1610–1616.
- [22] J. Wen, E. Thibeau-Sutre, M. Diaz-Melo, J. Samper-González, A. Routier, S. Bottani, D. Dormont, S. Durrleman, N. Burgos, O. Colliot, et al., Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation, *Med. Image Anal.* 63 (2020) 101694.
- [23] S. Sarraf, G. Tofighi, Classification of Alzheimer's disease using fMRI data and deep learning convolutional neural networks, 2016, [arXiv:1603.08631 \[cs\]](arXiv:1603.08631 [cs]).
- [24] M. Hon, N.M. Khan, Towards Alzheimer's disease classification through transfer learning, in: 2017 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, IEEE, Kansas City, MO, 2017, pp. 1166–1169, <http://dx.doi.org/10.1109/BIBM.2017.8217822>, URL: <http://ieeexplore.ieee.org/document/8217822/>.
- [25] A. Farooq, S. Anwar, M. Awais, S. Rehman, A deep CNN based multi-class classification of Alzheimer's disease using MRI, in: 2017 IEEE International Conference on Imaging Systems and Techniques, IST, IEEE, Beijing, 2017, pp. 1–6, <http://dx.doi.org/10.1109/IST.2017.8261460>, URL: <http://ieeexplore.ieee.org/document/8261460/>.
- [26] W. Lin, T. Tong, Q. Gao, D. Guo, X. Du, Y. Yang, G. Guo, M. Xiao, M. Du, X. Qu, The Alzheimer's Disease Neuroimaging Initiative, Convolutional neural networks-based MRI image analysis for the Alzheimer's disease prediction from mild cognitive impairment, *Front. Neurosci.* 12 (2018) 777, <http://dx.doi.org/10.3389/fnins.2018.00777>, URL: <https://www.frontiersin.org/article/10.3389/fnins.2018.00777/full>.
- [27] S. Basava, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo, M. Filippi, Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks, *NeuroImage: Clin.* 21 (2019) 101645, <http://dx.doi.org/10.1016/j.nicl.2018.101645>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S2213158218303930>.
- [28] S. Chen, J. Zhang, X. Wei, Q. Zhang, Alzheimer's disease classification using structural MRI based on convolutional neural networks, in: Proceedings of the 2020 2nd International Conference on Big-Data Service and Intelligent Computation, ACM, Xiamen China, 2020, pp. 7–13, <http://dx.doi.org/10.1145/3440054.3440056>, URL: <https://dl.acm.org/doi/10.1145/3440054.3440056>.
- [29] S. Murugan, C. Venkatesan, M.G. Sumithra, X.-Z. Gao, B. Elakkia, M. Akila, S. Manoharan, DEMNET: A deep learning model for early diagnosis of Alzheimer disease and dementia from MR images, *IEEE Access* 9 (2021) 90319–90329, <http://dx.doi.org/10.1109/ACCESS.2021.3090474>, URL: <https://ieeexplore.ieee.org/document/9459692/>.
- [30] W.H. Bangyal, N.U. Rehman, A. Nawaz, K. Nisar, A.A.A. Ibrahim, R. Shakir, D.B. Rawat, Constructing domain ontology for Alzheimer disease using deep learning based approach, *Electronics* 11 (12) (2022) 1890, <http://dx.doi.org/10.3390/electronics11121890>, URL: <https://www.mdpi.com/2079-9292/11/12/1890>.
- [31] S. Pallawi, D.K. Singh, Review and analysis of deep neural network models for Alzheimer's disease classification using brain medical resonance imaging, *Cogn. Comput. Syst.* 5 (1) (2023) 1–13, <http://dx.doi.org/10.1049/ccs2.12072>, URL: <https://onlinelibrary.wiley.com/doi/10.1049/ccs2.12072>.
- [32] D. Cheng, M. Liu, J. Fu, Y. Wang, Classification of MR brain images by combination of multi-CNNs for ad diagnosis, in: Ninth International Conference on Digital Image Processing, Vol. 10420, ICIDIP 2017, SPIE, 2017, pp. 875–879.
- [33] M. Liu, J. Zhang, E. Adeli, D. Shen, Joint classification and regression via deep multi-task multi-channel learning for Alzheimer's disease diagnosis, *IEEE Trans. Biomed. Eng.* 66 (5) (2018) 1195–1206.
- [34] F. Li, M. Liu, A.D.N. Initiative, et al., Alzheimer's disease diagnosis based on multiple cluster dense convolutional networks, *Comput. Med. Imaging Graph.* 70 (2018) 101–110.
- [35] K. Aderghal, M. Boissenin, J. Benois-Pineau, G. Catheline, K. Afdel, Classification of sMRI for AD diagnosis with convolutional neuronal networks: A pilot 2-D+ study on ADNI, in: International Conference on Multimedia Modeling, Springer, 2016, pp. 690–701.
- [36] K. Aderghal, J. Benois-Pineau, K. Afdel, C. Gwenaëlle, FuseMe: Classification of sMRI images by fusion of deep CNNs in 2D+ ε projections, in: Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, 2017, pp. 1–7.

- [37] K. Aderghal, A. Khvostikov, A. Krylov, J. Benois-Pineau, K. Afdel, G. Catheline, Classification of alzheimer disease on imaging modalities with deep CNNs using cross-modal transfer learning, in: 2018 IEEE 31st International Symposium on Computer-Based Medical Systems, CBMS, IEEE, 2018, pp. 345–350.
- [38] S. Korolev, A. Safullin, M. Belyaev, Y. Dodonova, Residual and plain convolutional neural networks for 3D brain mri classification, in: 2017 IEEE 14th International Symposium on Biomedical Imaging, ISBI 2017, IEEE, 2017, pp. 835–838.
- [39] H. Wang, Y. Shen, S. Wang, T. Xiao, L. Deng, X. Wang, X. Zhao, Ensemble of 3D densely connected convolutional network for diagnosis of mild cognitive impairment and Alzheimer's disease, Neurocomputing 333 (2019) 145–156.
- [40] S. Basaia, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo, M. Filippi, Alzheimer's Disease Neuroimaging Initiative (2019) Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks, NeuroImage: Clin 21 (2019) 101645.
- [41] E. Hosseini-Asl, G. Gimel'farb, A. El-Baz, Alzheimer's disease diagnostics by a deeply supervised adaptable 3D convolutional network, 2016, arXiv preprint arXiv:1607.00556.
- [42] T.-D. Vu, N.-H. Ho, H.-J. Yang, J. Kim, H.-C. Song, Non-white matter tissue extraction and deep convolutional neural network for Alzheimer's disease detection, Soft Comput. 22 (2018) 6825–6833.
- [43] S. Dubey, Alzheimer's Dataset ( 4 class of Images), URL: <https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images>.
- [44] K.M. Hasib, M.S. Iqbal, F.M. Shah, J.A. Mahmud, M.H. Popel, M.I.H. Showrov, S. Ahmed, O. Rahman, A survey of methods for managing the classification and solution of data imbalance problem, 2020, arXiv preprint arXiv:2012.11870.
- [45] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2011–2023, <http://dx.doi.org/10.1109/TPAMI.2019.2913372>, URL: <https://ieeexplore.ieee.org/document/8701503/>.
- [46] C. Özdemir, Avg-topk: A new pooling method for convolutional neural networks, Expert Syst. Appl. (2023) 119892.
- [47] M. Avşar, K. Polat, Classifying Alzheimer's disease based on a convolutional neural network with MRI images, J. Artif. Intell. Syst. 5 (1) (2023) 46–57.
- [48] Z. Ullah, M. Jamjoom, A deep learning for Alzheimer's stages detection using brain images, Comput. Mater. Continua 74 (1) (2022) 1457–1473, <http://dx.doi.org/10.32604/cmc.2023.032752>, URL: <https://www.techscience.com/cmc/v74n1/49878>.
- [49] T. Shahwar, J. Zafar, A. Almogren, H. Zafar, A. Rehman, M. Shafiq, H. Hamam, Automated detection of Alzheimer's via hybrid classical quantum neural networks, Electronics 11 (5) (2022) 721, <http://dx.doi.org/10.3390/electronics11050721>, URL: <https://www.mdpi.com/2079-9292/11/5/721>.
- [50] E. Kaplan, M. Baygin, P.D. Barua, S. Dogan, T. Tuncer, E. Altunisik, E.E. Palmer, U.R. Acharya, ExHiF: Alzheimer's disease detection using exemplar histogram-based features with CT and MR images, Med. Eng. Phys. 115 (2023) 103971, <http://dx.doi.org/10.1016/j.medengphy.2023.103971>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S1350453323000231>.
- [51] 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, <http://dx.doi.org/10.1016/j.combiomed.2021.104828>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0010482521006223>.
- [52] B. Nicholas, A. Jayakumar, B. Titus, T. Remya Nair, Comparative study of multiple feature descriptors for detecting the presence of Alzheimer's disease, in: P. Karuppusamy, I. Perikos, F.P. García a Márquez (Eds.), in: Ubiquitous Intelligent Systems, vol. 243, Springer Singapore, Singapore, 2022, pp. 331–339, [http://dx.doi.org/10.1007/978-981-16-3675-2\\_25](http://dx.doi.org/10.1007/978-981-16-3675-2_25), URL: [https://link.springer.com/10.1007/978-981-16-3675-2\\_25](https://link.springer.com/10.1007/978-981-16-3675-2_25).
- [53] Y.N. Fu'adah, I. Wijayanto, N.K.C. Pratiwi, F.F. Taliningsih, S. Rizal, M.A. Pramudito, Automated classification of Alzheimer's disease based on MRI image processing using convolutional neural network (CNN) with AlexNet architecture, J. Phys. Conf. Ser. 1844 (1) (2021) 012020, <http://dx.doi.org/10.1088/1742-6596/1844/1/012020>, URL: <https://iopscience.iop.org/article/10.1088/1742-6596/1844/1/012020>.
- [54] M.S. Kamal, A. Northcote, L. Chowdhury, N. Dey, R.G. Crespo, E. Herrera-Viedma, Alzheimer's patient analysis using image and gene expression data and explainable-AI to present associated genes, IEEE Trans. Instrum. Meas. 70 (2021) 1–7, <http://dx.doi.org/10.1109/TIM.2021.3107056>, URL: <https://ieeexplore.ieee.org/document/9521165>.
- [55] A. Gopinadhan, A. Prasanna G, A. S, AD-EHS: Alzheimer's disease severity detection using efficient hybrid image segmentation, Adv. Eng. Softw. 173 (2022) 103234, <http://dx.doi.org/10.1016/j.advengsoft.2022.103234>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0965997822001387>.
- [56] R. Suganthe, M. Geetha, G. Sreekanth, K. Gowtham, S. Deepakkumar, R. Elango, Multiclass classification of Alzheimer's disease using hybrid deep convolutional neural network, NVEO-Natural Volatiles & Essent. Oils J. NVEO (2021) 145–153.
- [57] A. Hashmi, O. Barukab, Dementia classification using deep reinforcement learning for early diagnosis, Appl. Sci. 13 (3) (2023) 1464, <http://dx.doi.org/10.3390/app13031464>, URL: <https://www.mdpi.com/2076-3417/13/3/1464>.
- [58] B. Rajasekhar, Early diagnosis model of Alzheimer's disease based on hybrid meta-heuristic with regression based multi feed forward neural network, Wirel. Pers. Commun. 130 (3) (2023) 1597–1615, <http://dx.doi.org/10.1007/s11277-023-10346-y>, URL: <https://link.springer.com/10.1007/s11277-023-10346-y>.