Hippocampus Segmentation-Based Alzheimer's Disease Diagnosis and Classification of MRI Images

A. Balasundaram[1] · Sruthi Srinivasan[2] · A. Prasad[2] · Jahan Malik[2] · Ayush Kumar[2]

Abstract

Alzheimer's disease represents a neurological condition characterized by steady cognitive decline and eventual memory loss due to the death of brain cells. It is one of the most prominent dementia types observed in patients and which hence underlines the imminent need for potential methods to diagnose the disease early on. This work considers a novel approach by utilizing a reduced version of one of the datasets used in this work to achieve a considerably accurate prediction while also enabling quicker training. It leverages image segmentation to isolate the hippocampus region from brain MRI images and then strikes a comparison between models trained on the segmented portions and models trained on complete images. This research uses two datasets—4 classes of images from Kaggle and a popular OASIS 2 MRI and demographic dataset. A deep learning-based approach was adopted to train the Kaggle dataset to perform severity classification, and the hippocampus region segmented from a reduced version of the OASIS dataset was trained on supervised and ensemble learning algorithms to detect Alzheimer's disease. The metric used for the assessment of model performance is classification accuracy. A comparative analysis between the proposed approach and existing work was also performed, and it was observed that the proposed approach is effective in the early diagnosis of Alzheimer's disease.

Keywords Hippocampus segmentation · Alzheimer's · Deep learning · Computer vision · Medical image processing

1 Introduction

Alzheimer's disease has been identified as the most prevalent variety of dementia. Starting with a seemingly mild memory loss, this disease gradually escalates to a loss of the ability to hold conversations and respond to the environment. Alzheimer's disease affects parts of the brain that control language, thought, and memory. It can adversely impact a person's ability to run daily errands. Statistics show that 1 in 9 people above the age of 65 has Alzheimer's disease. This comprises 11.4% of the world's population. Research shows that Alzheimer's disease cases have increased by 16% due to the Covid-19 pandemic. Another research shows that 1 in 3 Alzheimer patients die, which results in a higher number of

deaths than other chronic diseases such as breast cancer and prostate cancer combined [1]. Hence, early diagnosis of the disease is of vital importance.

Alzheimer's disease is classified into different classes concerning the rate of affection in the brain, namely mild, moderate, and severe. Patients with mild Alzheimer's experience greater memory loss. Patients are often found wandering and getting lost, having trouble handling money, and struggling to do everyday tasks. Patients with moderate Alzheimer's disease have trouble reasoning, have poor hearing, and lose the ability to smell. At this stage patients also have delusions, hallucinations, and paranoia. Patients with severe Alzheimer's are completely dependent on others for day-to-day tasks, the brain tissue shrinks significantly, and the patient ends up in bed for the rest of their life [2].

The objective of this research is to use computer vision to analyze brain MRI images of 2 different datasets, namely, the four classes of images from Kaggle and the OASIS dataset to classify and diagnose Alzheimer's disease. Subsequently, deep learning techniques are applied to the Kaggle dataset after necessary pre-processing, to determine the most suitable model that yields the highest accuracy. On the other hand,

✉ A. Balasundaram
   balasundaram.a@vit.ac.in

1 School of Computer Science and Engineering, Center for Cyber Physical Systems, Vellore Institute of Technology, Chennai, Tamil Nadu, India

2 School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Chennai, Tamil Nadu, India

the brain MRI images in the OASIS dataset are segmented to extract the hippocampus region in the brain, the shrinkage of which is known to aggravate Alzheimer's disease and dementia. These segmented images are trained alongside complete images, and the best model is once again determined based on the accuracy score. Moreover, to reduce the training time, the dataset is reduced to half its original size and the training results of this reduced dataset are analyzed alongside the complete dataset training results.

The main highlights of this paper are as follows:

- This work considers a novel approach by utilizing a reduced version of the OASIS-2 dataset to speed up the training process without compromising on accuracy.
- Image segmentation was used to isolate the hippocampus region from brain MRI images and then strike a comparison between models trained on the segmented portions and models trained on complete images.
- Deep learning models such as the CNN model, multilayer model, Resnet50, and more were utilized for the classification of severity.
- Machine learning and ensemble learning algorithms were used for the detection of Alzheimer's disease by segmenting the hippocampus region of the brain from MRI images.

This work mainly introduces an approach to obtain significantly accurate results using a reduced version of the OASIS-2 dataset to train models. In place of using all the images from each.nifti file in the dataset, this work uses only one image per.nifti file in the dataset. Furthermore, only one slice of each brain image was used to perform training, compared to conventional approaches that use the entire.nifti file for training. Additionally, this work performs a striking comparison using only the cropped portion of the brain for training. Followed by that it also compares training with only half the dataset to training with the entire dataset. This approach is especially useful in training computationally heavy deep learning models to achieve faster training times while also obtaining fairly accurate predictions.

The remainder of this paper is organized as follows—the next section is a literature survey of existing approaches for Alzheimer's disease detection and classification. This is followed by a proposed approach section detailing the methodology used in this work. The next section is the experimental setup which describes the software used for implementation. Following this, the results obtained in this work are discussed and explained and the paper is wrapped up with a conclusion section.

2 Literature Survey

Several studies pertaining to the early detection of Alzheimer's disease have been carried out. While some of the works used comma-separated values as input features,

some other works used brain MRI images and extracted the features from these images. Table 1 lists the contemporary works related to Alzheimer's disease detection.

Conventional approaches that experiment with the OASIS dataset use a CSV file format to diagnose Alzheimer's disease. This brings with it challenges such as manual inconsistencies, susceptibility to errors, and the tedious work that comes with populating the file. The proposed system eliminates these challenges by adopting an image-based approach where classification is performed solely by analyzing the brain MRI scans. Other contemporary approaches that use this image-based paradigm, analyze the entire area of the brain MRI image. The proposed approach, however, concentrates on the hippocampus region which is known to be the underlying factor behind Alzheimer's and cognitive decline. The proposed approach also demonstrates the benefits of using only 50% of the dataset for training over 100% of the dataset, thereby saving memory and processing time.

3 Proposed Approach

The proposed approach has been developed by considering two different datasets, namely—the Kaggle dataset consisting of four classes of brain images and the OASIS 2 dataset. The flow diagrams about how the data are processed for these two datasets are shown in Figs. 1 and 2, respectively.

As seen in Fig. 1, the images from the Kaggle dataset are read by specifying the path pertaining to the image location. Once all the images are extracted, they are subjected to pre-processing where images are resized to 45×45 pixels following which images are thresholded to highlight dark and light pixels more accurately. This is followed by converting the image list into a NumPy array and then separating and defining the labels for every image. The string labels are then converted to integer labels ranging from 0 to 3. The image array and the labels are used to create the training and testing sets. The pixel values ranging from 0 to 255 of both training and testing images are normalized to 0–1. The labels are categorically hot encoded before model training. The training dataset is oversampled to increase the number of images with which the model can predict.

$$\text{dst}(x, y) = \{\text{maxvalif} \text{src}(x, y) > \text{thresh0otherwise} \quad (1)$$

As shown in Fig. 2, while processing the images in OASIS 2 dataset, images read from the dataset are rotated to maintain uniformity following which a threshold is applied over the images to highlight dark and light areas. The hippocampus region in the brain is then cropped out and stored in a list called segmented features, whereas the full image is appended to a list known as full features. This process is termed hippocampus segmentation. For ease of training, the

Table 1 Comparison of the contemporary works on Alzheimer's disease

Reference number	Dataset used	Model used	Accuracy reported
[3]	OASIS	M-CAP net, LGBM, RF, Adaboost, Ensemble, ExtraTrees, GradientBoost, XGB, Decision tree, SVC, and KPCA	92.39%
[4]	Curated Dataset	Naive Bayes, KNN, SVM, ANN, RF, XGB, Ensemble Classification	83%
[5]	OASIS	Ensemble of Densenet 121, Densenet 161 and Densenet 169	93.18%
[6]	OASIS-3	Convolutional Neural Network	83.3%
[7]	OASIS	Convolutional Architecture with a Convolutional Layer, followed by a Pool Layer and Dense Layer with Activation function as ReLU	80.25% accuracy obtained on the OASIS dataset
[8]	OASIS	SVM	
[9]	OASIS	Convolutional Neural Network with six layers including convolutional layer, Pool layer, and Dense layer	
[10]	OASIS	The CNN-based network called AlzNet, Enhanced AlzNet with Delta Optimization	99.3%
[11]	OASIS	Ensemble of deep CNN models with Voxel-based, ROI-based, and patch-based feature extraction	93.18%
[12]	Kaggle	RESNET50 model followed by 6 layers that include Convolution, BatchNormalization, Dropout, Maxpooling, and Dense	90%
[13]	OASIS	Convolutional Neural Network with 18 layers with ReLU as activation function	98.51%
[14]	OASIS	Logistic Regression, SVM, Decision Tree, and Random Forests	Highest accuracy of 86.84% obtained with Random Forest

Table 1 (continued)

Reference number	Dataset used	Model used	Accuracy reported
[15]	OASIS	InceptionV3, Xception, MobileNetV2, VGG19, and a custom 12-layer CNN model	Highest Accuracy of 97.75% obtained with a custom CNN model
[16]	ADNI, OASIS	18-layer 3D Convolutional Neural Network model	73.4% accuracy was obtained on the ADNI dataset and 69.09% was obtained on the OASIS dataset
[17]	OASIS	Hybrid DenseNet architecture that trains on all 3 orientations such as Coronal, Sagittal, and Axial	95.23%
[18]	ADNI, OASIS	Transfer learning approach with the first 47 layers of ResNet ANN	78.64% accuracy was obtained on the ADNI dataset and 86.81% was obtained on the OASIS dataset
[19]	ADNI	YOLOv3, SSD, and Faster R-CNN on sagittal images	Accuracy of 99.8% for YOLOv3, 98.2% for SSD, and 98.8% for Faster R-CNN
[20]	ADNI	DNN uses a fusion of different modalities such as MRI, SNPs, and Clinical test data	The model achieved a maximum accuracy of 84%
[21]	ADNI fMRI and PET databases	Feature extraction using VGG-16 followed by training on SVM, KNN, Linear Discriminant Analysis, and Decision Tree	99.95% accuracy was obtained on ADNI fMRI images and 73.46% was obtained on ADNI PET images
[22]	ADNI and Kaggle Dataset	DEMNet consists of convolutional, pooling, dropout and dense layers with a DEMNet block	95.27% highest accuracy obtained on the Kaggle Dataset and 84.83% accuracy obtained on the ADNI dataset
[23]	ADNI, OASIS	Preprocessing techniques such as dual-tree complex wavelet transform, PCA, and LDA was applied to the dataset followed by training on the extra learning machine algorithm	90.26% accuracy was obtained on the ADNI dataset and 95.72% accuracy was obtained on the OASIS dataset

two lists comprising the full stack of features and segmented features are converted into NumPy arrays. Labels required for training are extracted from a CSV file provided in the OASIS dataset, and they are subsequently encoded as integers ranging from 0 to 2. The training sets and testing sets for both the full features and the segmented features are then created. Following this, the pixel values in these training and testing datasets are scaled down from 0–255 to 0–1 and converted to one-dimensional arrays. Once the training dataset is oversampled, each machine learning algorithm is applied with conditions such as default parameters, best parameters, default parameters trained on the oversampled dataset and best parameters on the oversampled dataset for an entire 2D

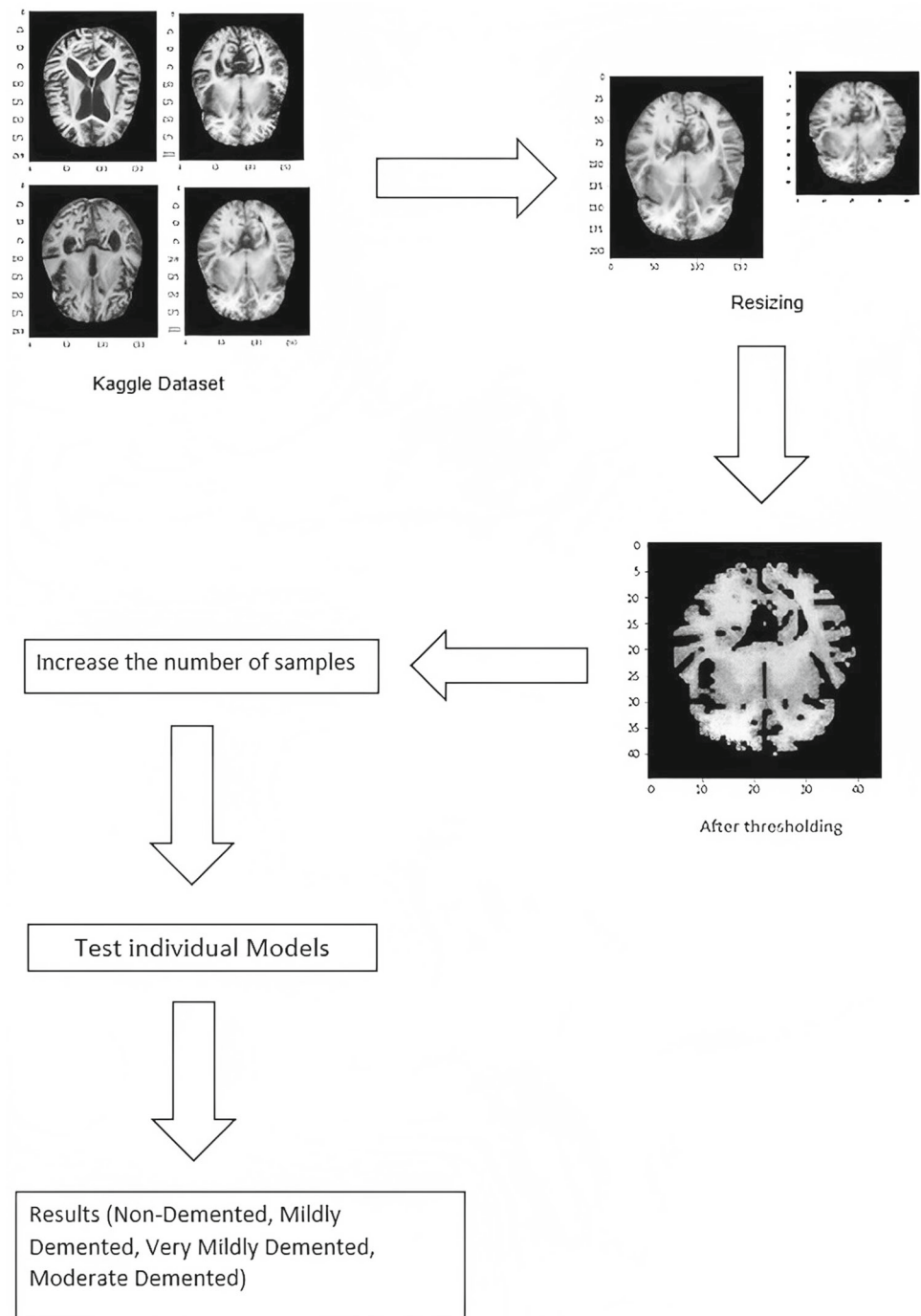**Fig. 1** Block diagram for processing Kaggle brain MRI data

image (4 models) as well as a segmented 2D image (4 models).

3.1 Architecture

Three models each using a different architecture—namely a simple multi-layered architecture, convolution neural network (CNN), and ResNet50—were developed and tested

over the two datasets. The subsequent section discusses in detail the individual architecture specifications.

3.1.1 Simple Multilayer Model

This model contains a total of 15 layers of which 8 are dense layers and 7 are dropout layers. All the dropout layers are placed between two dense layers to avoid overfitting. All the dropout layers are given a value of 0.05. Each dense layer
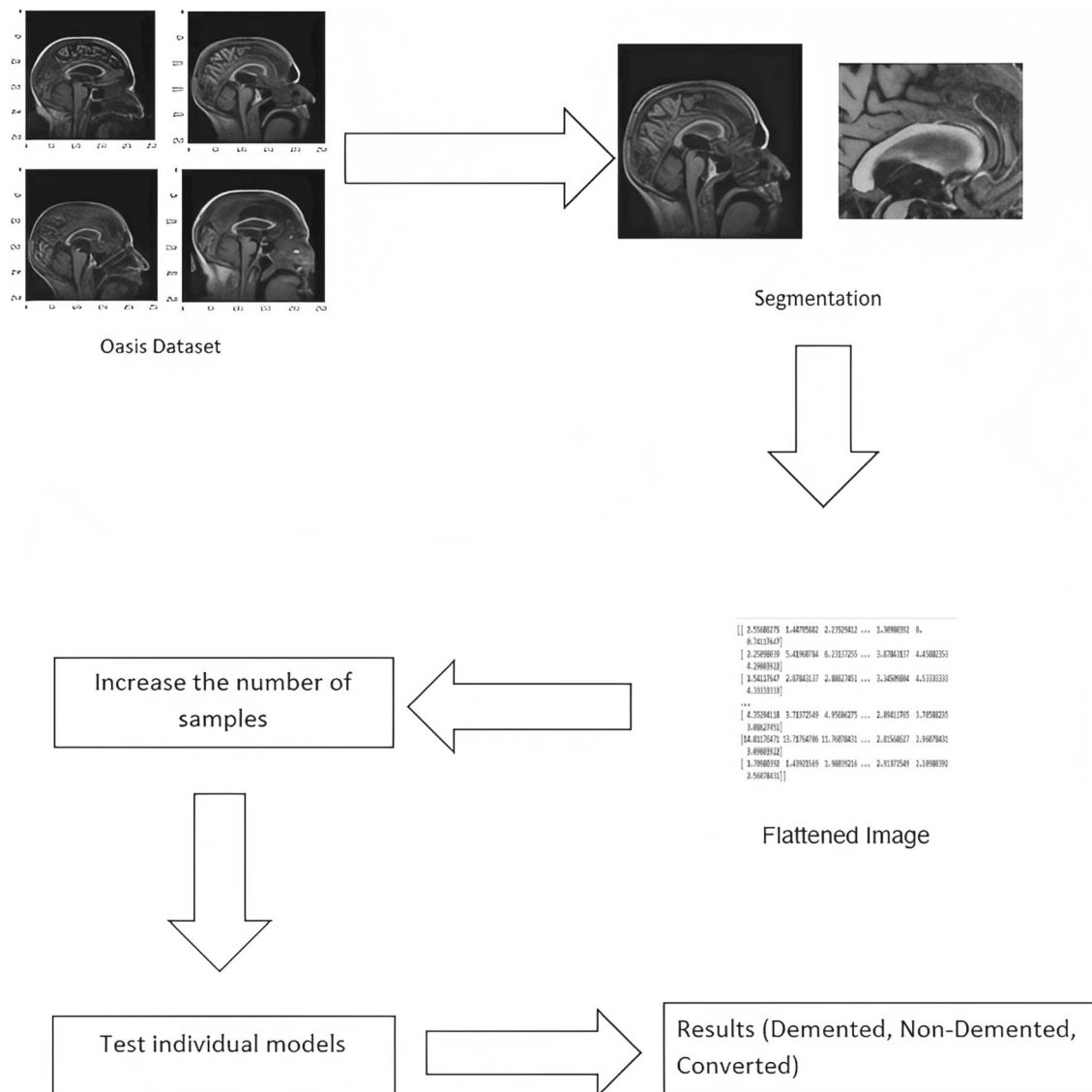
Fig. 2 Block diagram for processing OASIS 2 MRI data

contains a certain number of neurons along with an activation function. Table 2 lists the specifications pertaining to each of the layers in the fully connected neural network. The model is compiled by setting the loss metric to categorical cross-entropy and the optimizer to RMSprop. The batch size is set to 32.

3.1.2 Convolutional Neural Network (CNN) Model

This model has a total of 24 layers including 6 conv2d layers, 3 Maxpool 2d layers, 4 Separableconv2d layers, 4 Dropout layers, 4 BatchNormalization layers, and 3 dense layers. Table 3 lists the specifications pertaining to each of the layers in the CNN model. The model is compiled by setting the

loss metric to categorical cross-entropy and using the Adam optimizer.

3.1.3 ResNet50 Model

This model is a ResNet50 model with an additional 18 layers including 6 dense layers, 5 Dropout layers, 1 flatten layer, and 6 BatchNormalization layers. After loading the 50 pre-trained layers in the ResNet 50 model the layers are added in the following sequence as specified in Table 4. ResNet is adept at handling the vanishing gradient problem which can result in inaccurate predictions.

Table 2 Simple multilayer model—layer specifications

Layers	Role	Specifications
Input Layer	Dense	Activation Function—Relu, Neurons—256
Hidden—1	Dropout	Value—0.05
Hidden—2	Dense	Activation Function—Relu, Neurons—128
Hidden—3	Dropout	Value—0.05
Hidden—4	Dense	Activation Function—Relu, Neurons—128
Hidden—5	Dropout	Value—0.05
Hidden—6	Dense	Activation Function—Relu, Neurons—128
Hidden—7	Dropout	Value—0.05
Hidden—8	Dense	Activation Function—Relu, Neurons—128
Hidden—9	Dropout	Value—0.05
Hidden—10	Dense	Activation Function—Relu, Neurons—64
Hidden—11	Dropout	Value—0.05
Hidden—12	Dense	Activation Function—Relu, Neurons—64
Hidden—13	Dropout	Value—0.05
Output Layer	Dense	Activation Function—SoftMax, Neurons—4

Table 3 Convolutional Neural Network model—Layer Specifications

Layers	Role	Specifications
Input Layer	Conv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—1	Conv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—2	Conv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—3	Conv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—4	Conv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—5	Conv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—6	MaxPool2D	Pool size—2,2
Hidden—7	SeparableConv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—8	Dropout	Value—0.2
Hidden—9	SeparableConv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—10	Batch Normalization	No specifications
Hidden—11	MaxPool2D	Pool size—2,2
Hidden—12	SeparableConv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—13	Dropout	Value—0.2

3.2 Pseudo-Logic When Experimented with Kaggle Dataset

The following pseudocode could be categorized into the following steps: Image reading, Image preprocessing, Train test split, and Oversampling. The first step involves loading all the files in the dataset using the glob function. The second step comprises resizing and applying thresholding to the images. The images are resized from the dimensions 208×176 to 45×45 to reduce the overall processing time. The threshold value is set to 120, and the thresholding technique used here is THRESH_TOZERO, where the pixels above the given threshold will retain their value and the rest will be made 0. The processed images are stored in a list, which is later converted to a NumPy array. The string labels are encoded as integers to facilitate model training. The dataset has been split into training and testing using the train_test_split function, where the training set comprises 80% of the dataset and the testing set comprises 20% of the dataset. The dimensions of the images are reshaped from $45 \times 45\times3$ to 6075 to give the deep-learning models a total of 6075 parameters per image. The final step involves oversampling the training sets using the SMOTE function to make images of both classes in the dataset comparable.

Table 3 (continued)

Layers	Role	Specifications
Hidden—14	SeparableConv2D	Neurons—64, Kernel size—3, Padding—'Same', Activation Function—Relu
Hidden—15	Batch Normalization	No specifications
Hidden—16	MaxPool2D	Pool size—2,2
Hidden—17	Dense	Neurons—128, Activation Function—Relu
Hidden—18	Batch Normalization	No specifications
Hidden—19	Dropout	Value—0.3
Hidden—20	Dense	Neurons—64, Activation Function—Relu
Hidden—21	Batch Normalization	No specifications
Hidden—22	Dropout	Value—0.3
Output Layer	Dense	Neurons—4, Activation Function—SoftMax

Table 4 ResNet50 model—Layer Specifications

Layers	Role	Specifications
1	Dropout	Value—0.5
2	Flatten	No specifications
3	BatchNormalization	No specifications
4	Dense	Neurons—64, Kernel Initializer—he_uniform, Activation Function—Relu
5	BatchNormalization	No specifications
6	Dropout	Value—0.5
7	Dense	Neurons—64, Kernel Initializer—he_uniform, Activation Function—Relu
8	BatchNormalization	No specifications
9	Dropout	Value—0.5
10	Dense	Neurons—64, Kernel Initializer—he_uniform, Activation Function—Relu
11	Dropout	Value—0.5
12	Dense	Neurons—32, Kernel Initializer—he_uniform, Activation Function—Relu
13	BatchNormalization	No specifications
14	Dropout	Value—0.5
15	Dense	Neurons—32, Kernel Initializer—he_uniform, Activation Function—Relu
16	BatchNormalization	No specifications
17	Dropout	Value—0.5
18	Dense	Neurons—4, Activation Function—SoftMax

PSEUDO CODE FOR IMPLEMENTATION USING KAGGLE DATASET:

initialize images:=NULL

initialize labels:=NULL

set datapath := ['Alzheimer_s Dataset/train','Alzheimer_s Dataset/test']

set label := ['MildDemented', 'VeryMildDemented', 'ModerateDemented', 'NonDemented']

__for__ every element __Path__ in datapath do

    __for__ every element __X__ in label do

        __for__ every element __dir_path__ in glob->__Path__ do

            __for__ every element __image_path__ in path.join->__dir_path__, 'jpg'

                set __image:__=imread->__image_path__

                resize __image__ from (208,176) -> (45,45)

                __image__ := threshold->__image__

                    threshold value :=120

                    maxValue :=255

                    thresholding technique := THRESH_TOZERO

                    images.append -> __image__

                    labels.append -> __X__

            End __For__

        End __For__

    End __For__

End __For__

images := array->images

labels := array->labels

initialize label_to_id_dict := NULL

__for__ every element __I__ in unique->labels do

    __i,v__ := enumerate->__I__

    set label_to_id_dict[__v__] := __i__

End __For__

initialize label_ids :=NULL

__for__ every element __J__ in labels do

    label_ids.append->label_to_id_dict[__J__]

End __For__

initialize X_train,y_train,X_test,y_test := NULL

set X_train,y_train,X_test,y_test := train_test_split->images,labels_ids

X_train := X_train/255

X_test := X_test/255

reshape X_train from (5120,45,45,3) to (5120,6075)

reshape X_test from (1280,45,45,3) to (1280,6075)

set y_train:=to_categorical->y_train,4

set y_test:=to_categorical->y_test,4

initialize sm :=NULL

set sm := SMOTE->random_state=42

initialize train_data, train_labels := NULL

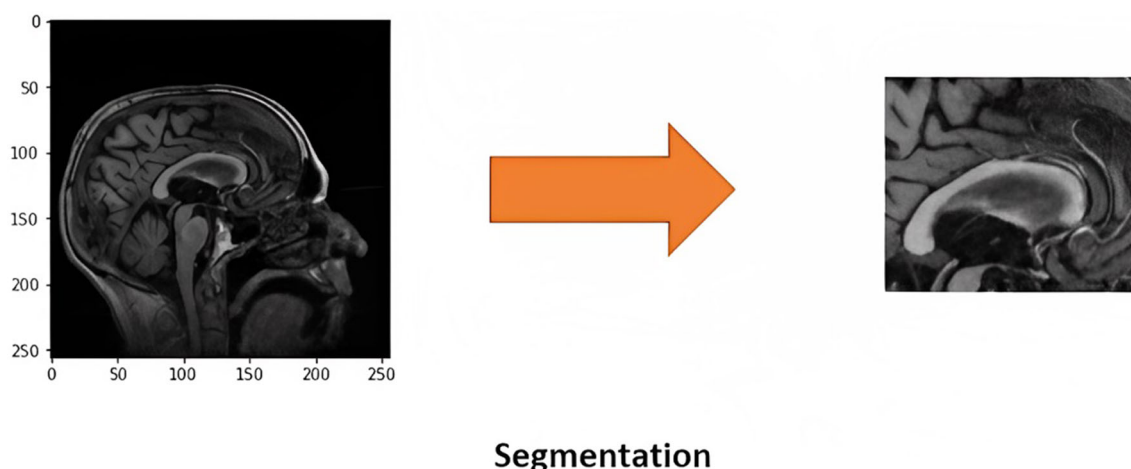*set train_data, train_labels = sm.fit_resample->X_train, y_train*
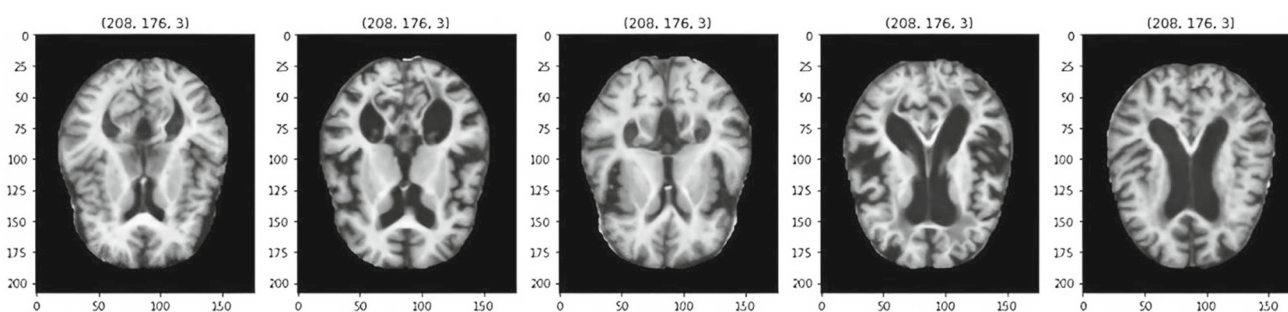
Fig. 3 Segmented hippocampus region

Fig. 4 Sample images from the Kaggle dataset

3.3 Hippocampus Segmentation

Shrinkage observed in the hippocampus region is a major factor leading to Alzheimer's disease. Hence, segmenting the area surrounding the hippocampus to achieve better accuracy has been viewed as a prospective approach. To achieve this, a brute force approach was followed to determine the coordinates. After numerous attempts, the coordinates were obtained as (70, 70), (150, 70), (70, 170), and (150, 170). This area was extracted by slicing the image array as an image [70:150, 70:170]. The implementation of the proposed approach while using the Kaggle dataset and the OASIS dataset is discussed in the subsequent sections. Figure 3 is a sample depiction of the segmented hippocampus region.

3.4 Pseudo-Logic When Experimented with OASIS 2 Dataset

The following pseudocode can be categorized as the following: Dataset loading, Image processing, Hippocampus segmentation, Train test split, and Oversampling. The first step involves loading the nifti images and their corresponding labels. The labels are stored in a CSV file, which is loaded using read_csv. The images are loaded using the glob function, and the images inside the nifti files are extracted using the get_fdata function. The second step involves choosing one 2D image from each nifti file and applying thresholding to it. The middlemost image has been chosen for predicting the presence of Alzheimer's because the shape of the hippocampus is more apparent in this image. The threshold value is set to 120 and the thresholding technique used here is THRESH_TOZERO, where the pixels above the given threshold will retain their value and the rest will be made 0. The processed images are stored in a list which is later converted to a NumPy array. The third step involves segmenting the hippocampus region, which is achieved by cropping the hippocampus region from the 2D image and using it for prediction. Two sets of datasets are created: one with the entire 2D images and one with the hippocampus segmented images; these images are stored in individual lists and later converted to NumPy arrays. The fourth step involves creating separate training and testing sets to compare the results across various models using train_test_split. The images in the arrays are flattened into a one-dimensional array using the reshape function to facilitate the training of the machine learning models. The final step involves oversampling the training sets using the SMOTE function to increase the accuracy of predicting each class in the dataset.

PSEUDO CODE FOR IMPLEMENTATION USING OASIS 2 DATASET:

initialize df := NULL
set df := read_csv->"oasis_longitudinal_demographics.csv"
initialize labels := NULL
set labels := df['Group']
initialize label_to_id_dict := NULL

for *every element **I** in unique->labels do*
   i,v := *enumerate->**I***
   *set label_to_id_dict[**v**] := **i***
*End **For***
initialize label_ids :=NULL
for *every element **J** in labels do*
   *label_ids.append->label_to_id_dict[**J**]*
*End **For***
initialize full_features , segmented_features := NULL
initialize dataset_path := ["OAS2_RAW_PART1/", "OAS2_RAW_PART2/"]

for *every element **PATH** in dataset_path do*
 *initialize files := glob->**PATH***
  for *every element **file_path** in files do*
    *initialize filename := join->**file_path** , "RAW/mpr-1.nifti.img"*
    initialize img := load->filename
    initialize image_data := img.get_fdata
    initialize image := rotate->image_data[: , : , 64 , 0] , 90
    image** := threshold->**image
        threshold value :=120
        maxValue :=255
       thresholding technique := THRESH_TOZERO
    full_features.append->image
    segmented_features.append->image[70:150,70:170]
  *End **For***
*End **For***
full_features := array->full_features
segmented_features := array->segmented_features
initialize X_full_train,y_train,X_full_test,y_test := NULL
set X_full_train ,y_train ,X_full_test ,y_test := train_test_split->full_features,label_ids
initialize X_seg_train,y_train,X_seg_test,y_test := NULL
set X_seg_train,y_train,X_seg_test,y_test := train_test_split->segmented_features,label_ids

4 Experimental Setup

This research work called for a high RAM and memory. The maximum RAM utilized was 35 GB. A high-end TPU was leveraged especially to carry out the experiments pertaining to deep learning. Jupyter notebook and Google Colaboratory were used to carry out most of the experimental works. The dataset was mounted on Google Drive. As part of this work, several packages such as TensorFlow, imblearn, OpenCV, and nibabel were also installed for performing specific tasks. Tensorflow was used to analyze the Kaggle dataset. Imblearn was used for oversampling existing images in the dataset to augment the number of images involved in training. OpenCV was used for vision and image processing tasks such as reading, displaying, thresholding, and resizing the images. Nibabel was used in the reading and extraction of files with the.nifti extension in the OASIS dataset. In addition to these, several other python libraries such as NumPy, pandas, and sklearn were imported and used for each model.

As discussed earlier, two datasets, namely Kaggle 4 classes dataset and OASIS 2 dataset, were considered for this work. The Kaggle dataset is a collection of MRI images of the back portion of the brain in jpg format. There are a total of 4 classes present in the dataset, namely non-demented, very mildly demented, mildly demented, and moderately demented. In total, the dataset comprises 6400 images. Among these images, 5120 images are taken for training the model and the remaining 1280 images are used for testing. In total, 3200 images are of type non-demented, 2240 images are of type very mildly demented, 896 images are of type mildly demented, and 64 images are of type moderately demented. A few sample images pertaining to the Kaggle dataset are shown in Fig. 4.

The Oasis dataset consists of 150 different subjects with a total of 373 MRI sessions. The age group of the subjects ranges from 60 to 96 and includes both men and women. Throughout their study of the subjects, 72 people were termed non-demented, 64 were demented, and among these 64 subjects, 51 were individuals possessing mild to moderate Alzheimer's disease. Another 14 were initially termed non-demented, but over time they were termed demented. The Oasis dataset has 3 classes demented, non-demented, and converted. The MRI scans are in the form of nifti.hdr files. These images are the MRI scans of the side portions of the brain. A CSV file is also attached along with the dataset, and it contains information such as the EDUC, SES, MMSE, CDR, eTIV, nWBV, ASF, and MR delay. A few sample images pertaining to the Kaggle dataset are shown in Fig. 5.

5 Results and Discussion

5.1 Oasis Dataset

Various models were tested over the OASIS 2 dataset. A reduced dataset approach was also tried along with the complete set of OASIS 2 images.

5.1.1 Full Features Vs Segmented Features

Figure 6 performs a comparison of the full feature and segmented feature-based approaches for the input Oasis dataset images. The X-axis corresponds to each model, and the Y-axis corresponds to the accuracy. This hypothesis is postulated based on the fact that the presence of dementia is dependent on the size of the hippocampus in the brain. According to neurology, the smaller the Hippocampus, the more severe the dementia. Thus, the region comprising the hippocampus is segmented and tested.

To improve the above results, discrete wavelet transforms were used as a pre-processing technique. From the following trends observed in the graphs, we could infer that on average, training using the segmented hippocampus part of the brain MRI image provided a higher accuracy as compared to using the entire image for training.

5.1.2 Complete Dataset VS Reduced Dataset

This hypothesis was tested as the Oasis dataset was available in 2 parts—Part A and Part B. Part A had nearly 209 MRI images, and hence, this portion of the dataset was utilized for training. Based on the trends reflected in Fig. 7, it is evident that, on average, the reduced dataset yields a higher accuracy as compared to the accuracy obtained by using the complete dataset. The reduced dataset gave the highest overall accuracy of 94% when trained on the XGBoost model. The reason for this might be because of the concept of overfitting.

5.2 Kaggle Dataset

Many multi-layered models aside from the multi-layered model and CNN model were tested, and the results of those models are mentioned in Table 5:

After these model parameters were tested, the dataset was oversampled, and the final accuracy was 94.45% for the same model with 28 epochs. The epoch vs. accuracy plot and epoch vs. loss plot are shown in Fig. 8. Table 6 shows the classification report for the multi-layered model.

Upon training the CNN model, an accuracy of 94.1% was achieved, and its accuracy and loss for increasing epochs are depicted in Fig. 8. When trained with ResNet50, the model gave an accuracy of 75.64%. Also, the proposed approach
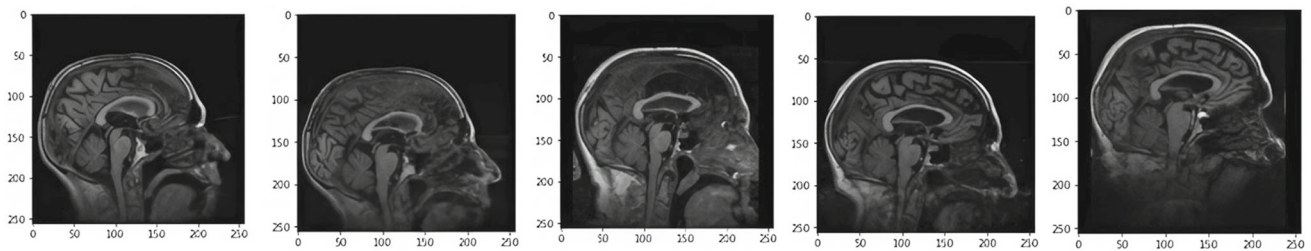
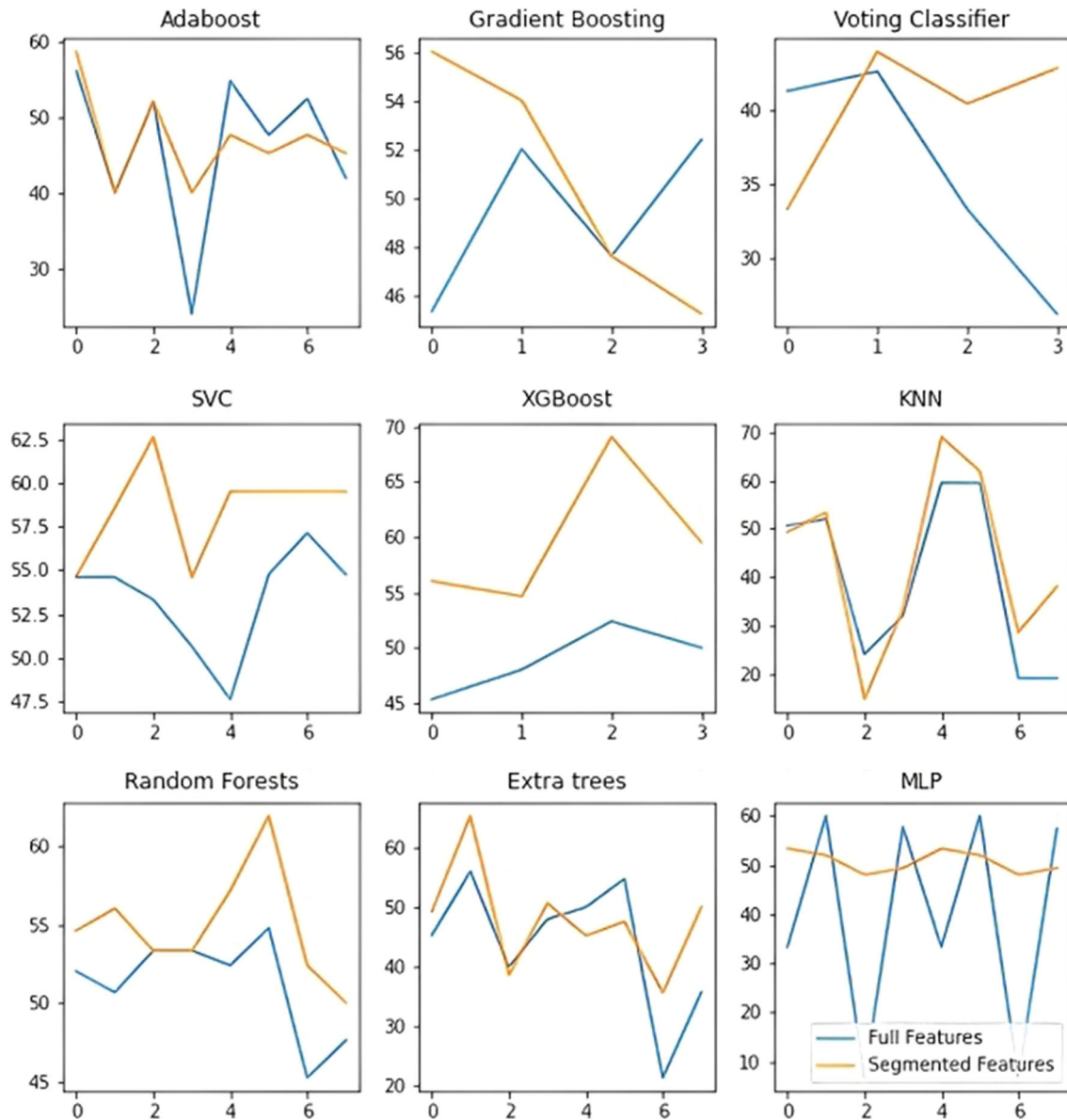**Fig. 5** Sample images from the OASIS dataset

Fig. 6 Comparison between full features and segmented features using a pairwise vector graph

was compared with an existing work [12] in terms of precision, recall, F1 score, and accuracy as both of these works use the same Kaggle dataset for training and testing. From Table 7, it can be seen that the proposed system has higher

precision, recall, and F1 score compared to the existing work [12].

The proposed approach to the OASIS dataset has been compared with the existing approaches. From Table 8, it
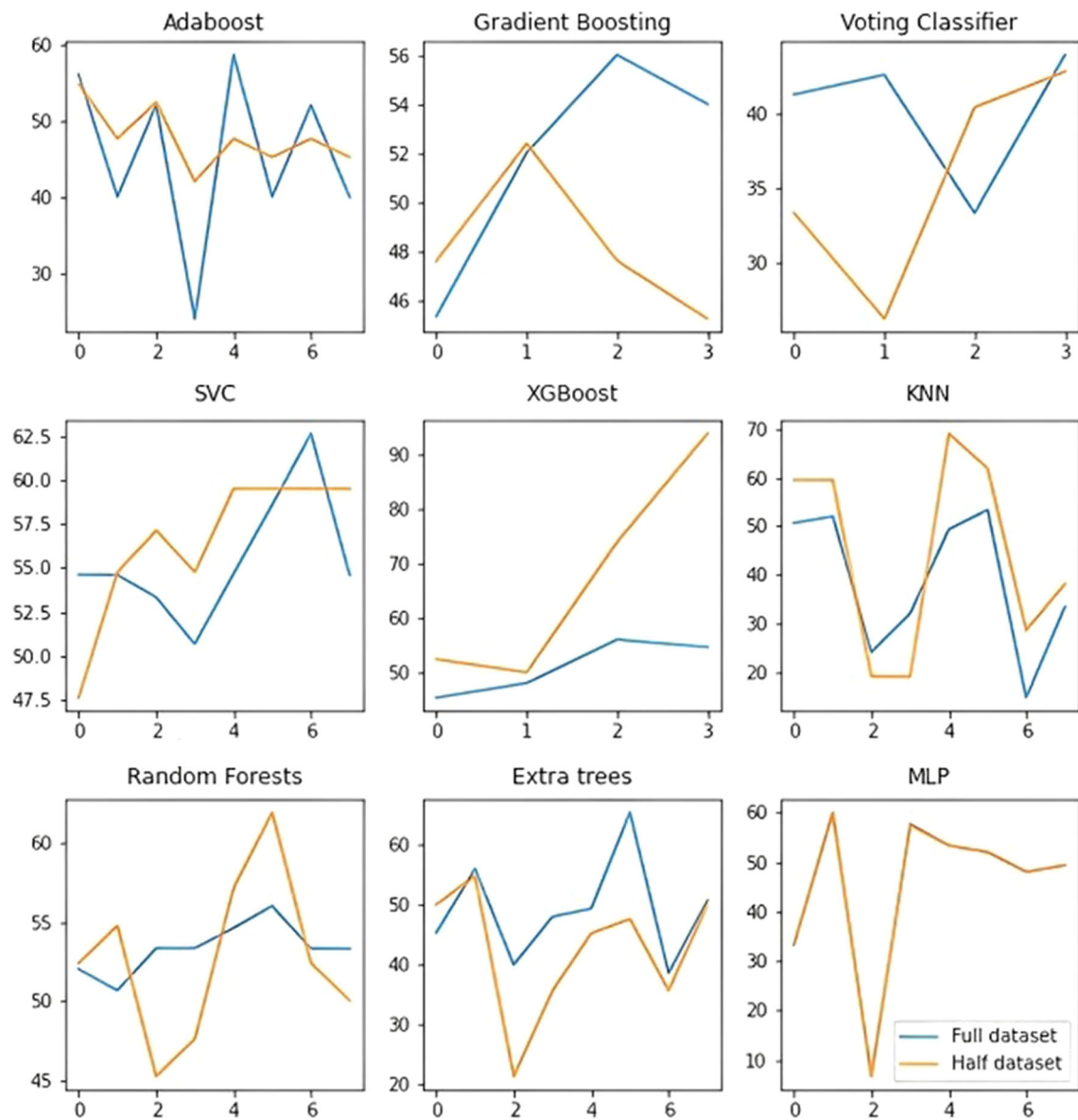
Fig. 7 Comparison between the complete dataset and reduced dataset using a pairwise vector graph

Table 5 Classification report for the multi-layered model

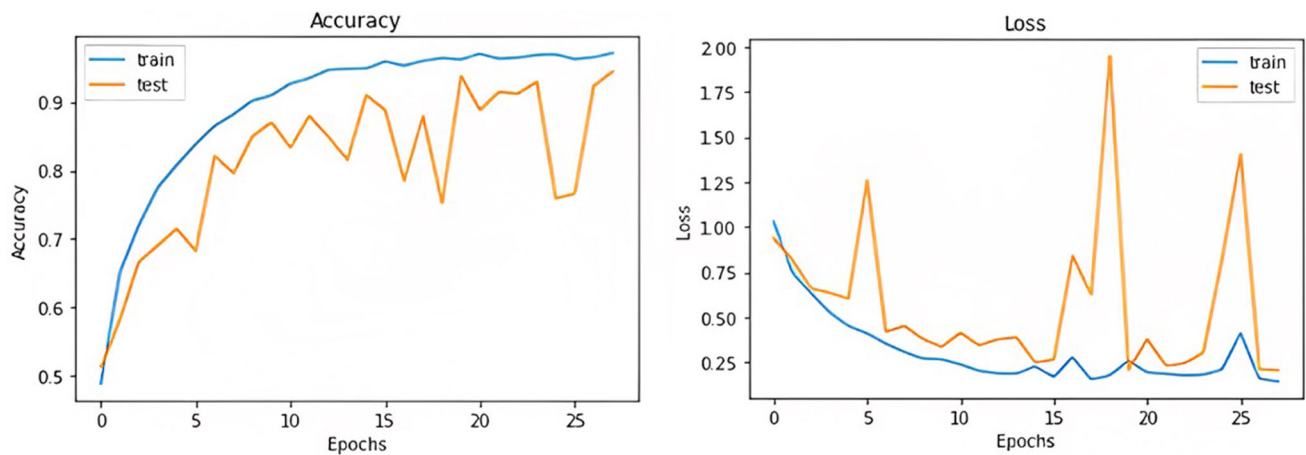| Total no. of Layers	No. of Dense layers	No. of Dropout Layers	No. of Epochs	Accuracy
13	7	6	10	61.01%
5	3	2	10	64.29%
11	6	5	10	67.18%
17	9	8	10	70.23%
17	9	8	15	72.42%
17	9	8	29	80.46%
17	9	8	28	89.84%

Fig. 8 Comparison of accuracies and losses with increasing epochs

Table 6 Classification report for the multi-layered model

	Precision	Recall	F1 score	Support
Mild demented	0.98	0.91	0.94	185
Moderate demented	1.00	1.00	1.00	16
Non-demented	0.93	0.98	0.96	653
Very mildly demented	0.95	0.90	0.92	426
Microaverage	0.95	0.94	0.94	1280
Macroaverage	0.97	0.95	0.94	1280
Weighted average	0.95	0.94	0.94	1280
Samples average	0.94	0.94	0.94	1280

Table 7 Comparison of proposed approach with existing approach [12] that uses same Kaggle dataset

	Precision		Recall		F1 score		Overall accuracy	
	Proposed approach	Existing approach [12]	Proposed approach	Existing approach [12]	Proposed approach	Existing approach [12]	Proposed approach	Existing approach [12]
Mild demented	98%	96.6%	91%	90.62%	94%	93.51%	94.1%	90%
Moderate demented	100%	70%	100%	70%	100%	70%		
Non-demented	93%	90%	98%	96.42%	96%	93.09%		
Very mildly demented	95%	90%	90%	90%	92%	90%		

is apparent that the proposed approach has higher accuracy compared to most other approaches. There are some entries marked as not available (NA) where in the metric details were not provided in the existing works. Despite the observation that existing approaches [10, 13, 15], and [17] produced a marginally better accuracy than the proposed approach, the most striking aspect of the proposed approach is that an accuracy of 94% was achieved with half the size of the dataset that was used for training. This underlines the fact that the

training time will be minimized without compromising much on the accuracy.

6 Conclusion

As a part of this research, two datasets were analyzed, namely four classes of images from Kaggle and the OASIS 2 MRI and demographic dataset. The Kaggle dataset classifies brain MRI images into four classes, namely, Non-demented,

Table 8 Comparison of proposed approach with other existing approaches that use OASIS dataset

Approach	Precision	Recall	F1 score	Accuracy
[3]	94.77	82.3	88.1	92.39%
[5]	93	92	92	93.18%
[6]	NA	NA	NA	83.3%
[7]	NA	NA	NA	80.25%
[9]	67.25	67.5	67.37	NA
[10]	98.92	99.53	98.92	99.3%
[11]	94	93	92	93.18%
[13]	NA	NA	NA	98.51%
[14]	NA	80	NA	86.84%
[15]	96.5	96	97.5	97.75%
[16]	NA	NA	NA	69.09%
[17]	NA	NA	NA	95.23%
[18]	32.08	37.81	33.47	86.81%
Proposed approach [Reduced Dataset Approach]	95.65	91.66	93.61	94%

Very Mildly Demented, Mildly Demented, and Moderately demented. The area of focus for this dataset was deep learning, and several multi-layered neural networks with a varying number of epochs and batch sizes were tested. The images were also oversampled to expand the training dataset. Additionally, an experimental study on the OASIS 2 dataset was carried out, and this involved utilizing the images for training and analysis and the CSV demographic data as labels. The images were first extracted from the existing nifti file extension. This was followed by pre-processing, where the images were flattened from a 3-dimensional orientation to a 2-dimensional and then resized to make them suitable for further training. Further training was conducted using the complete dataset as well as the reduced dataset. The central idea of this study tied into segmenting a portion of the brain image known as the hippocampus, the shrinkage of which is responsible for Alzheimer's disease. The images were oversampled to increase the volume of training data. Several machine learning models such as SVM, KNN, XGBoost, Random Forest, Extra Trees, Gradient Boosting, AdaBoost, and Multilayer Perceptron were then applied to the segmented images as well as full images of both halves of the dataset and the complete dataset. GridSearchCV was also used for fine-tuning of models wherever appropriate.

The accuracy achieved in case of the Kaggle dataset was 94.45% using a CNN model with batch size set to 10 and number of epochs set to 10. In the case of OASIS dataset, the proposed method achieved 94% accuracy using only 50% of the available data for training. It was also observed that segmentation of the hippocampus region yields higher classification accuracy than using entire images. The highest accuracy was obtained when the XGBoost algorithm was applied to the reduced dataset and then segmented to focus on the hippocampus region, to yield better results.

References

1. https://www.alz.org/alzheimers-dementia/facts-figures
2. https://www.nia.nih.gov/health/alzheimers-disease-fact-sheet
3. Basheer, S.; Bhatia, S.; Sakri, S.B.: Computational modeling of dementia prediction using deep neural network: analysis on OASIS dataset. IEEE Access **9**, 42449–42462 (2021)
4. Vinayak, S.S.; Shahina, E.A.; Nayeemulla Khan, A.: Dementia prediction on OASIS dataset using supervised and ensemble learning techniques. Int. J. Eng. Adv. Technol. **10**(1):244-254 (2020)
5. Islam, J.; Zhang, Y.: An ensemble of deep convolutional neural networks for Alzheimer's disease detection and classification. arXiv preprint: arXiv:1712.01675 (2017)
6. Battineni, G.; Chintalapudi, N.; Amenta, F.; Traini, E.: Deep learning type convolution neural network architecture for multiclass classification of alzheimer's disease. In: BIOIMAGING (pp. 209–215) (2021)
7. Ullah, H.T.; Onik, Z.; Islam, R.; Nandi, D.: Alzheimer's disease and dementia detection from 3D brain MRI data using deep convolutional neural networks. In: 2018 3rd International Conference for Convergence in Technology (I2CT) (pp. 1–3). IEEE (2018)
8. Herrera, L.J.; Rojas, I.; Pomares, H.; Guillén, A.; Valenzuela, O.; Baños, O.: Classification of MRI images for Alzheimer's disease detection. In 2013 International Conference on Social Computing (pp. 846–851). IEEE (2013)
9. Islam, J.; Zhang, Y.: Early diagnosis of Alzheimer's disease: A neuroimaging study with deep learning architectures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 1881–1883) (2018)
10. Al-Khuzaie, F.E.; Bayat, O.; Duru, A.D.: Diagnosis of Alzheimer's disease using 2D MRI slices by convolutional neural network. Appl. Bionics Biomech. (2021)
11. Islam, J.; Zhang, Y.: Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. Brain informatics **5**(2), 1–14 (2018)
12. Yildirim, M.; Cinar, A.C.: Classification of Alzheimer's Disease MRI images with CNN based hybrid method. Ingénierie des Systèmes d Inf. **25**(4), 413–418 (2020)
13. Khagi, B.; Lee, B.; Pyun, J.Y.; Kwon, G.R.: CNN Models Performance Analysis on MRI images of OASIS dataset for distinction between Healthy and Alzheimer's patient. In: 2019 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1–4). IEEE (2019)
14. Baglat, P.; Salehi, A.W.; Gupta, A.; Gupta, G.: Multiple machine learning models for detection of Alzheimer's disease using OASIS dataset. In: International Working Conference on Transfer and Diffusion of IT (pp. 614–622). Springer, Cham (2020)
15. Hussain, E.; Hasan, M.; Hassan, S.Z.; Azmi, T.H.; Rahman, M.A.; Parvez, M.Z.: Deep learning based binary classification for alzheimer's disease detection using brain mri images. In: 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA) (pp. 1115–1120). IEEE (2020)
16. Yagis, E.; Citi, L.; Diciotti, S.; Marzi, C.; Atnafu, S.W.; De Herrera, A.G.S.: 3d Convolutional neural networks for diagnosis of

alzheimer's disease via structural mri. In: 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS) (pp. 65–70). IEEE (2020)

17. Jabason, E.; Ahmad, M.O., Swamy, M.N.S.: Classification of Alzheimer's disease from MRI data using an ensemble of hybrid deep convolutional neural networks. In: 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS) (pp. 481–484). IEEE (2019)

18. Puente-Castro, A.; Fernandez-Blanco, E.; Pazos, A.; Munteanu, C.R.: Automatic assessment of Alzheimer's disease diagnosis based on deep learning techniques. Comput. Biol. Med. **120**, 103764 (2020)

19. Fong, J.X.; Shapiai, M.I.; Tiew, Y.Y.; Batool, U.; Fauzi, H.: Bypassing MRI Pre-processing in Alzheimer's Disease Diagnosis using Deep Learning Detection Network. In: 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA) (pp. 219–224). IEEE (2020)

20. Venugopalan, J.; Tong, L.; Hassanzadeh, H.R.; Wang, M.D.: Multimodal deep learning models for early detection of Alzheimer's disease stage. Sci. Rep. **11**(1), 1–13 (2021)

21. Janghel, R.R.; Rathore, Y.K.: Deep convolution neural network based system for early diagnosis of Alzheimer's disease. Irbm **42**(4), 258–267 (2021)

22. Murugan, S.; Venkatesan, C.; Sumithra, M.G.; Gao, X.Z.; Elakkiya, B.; Akila, M.; Manoharan, S.: DEMNET: a deep learning model for early diagnosis of Alzheimer diseases and dementia from MR images. IEEE Access **9**, 90319–90329 (2021)

23. Jha, D.; Alam, S.; Pyun, J.Y.; Lee, K.H.; Kwon, G.R.: Alzheimer's disease detection using extreme learning machine, complex dual tree wavelet principal coefficients and linear discriminant analysis. J. Med. Imag. Health Inf. **8**(5), 881–890 (2018)