[1]Shashi Kant Mourya
[2]Dr. Ajit Kr. Singh Yadav

A Versatile Deep Learning Model for Alzheimer's Disease Detection by Using Structural MRIs

Journal of Electrical Systems

Abstract: - Alzheimer's disease (AD), a global neurodegenerative condition, primarily affects the elderly, necessitating precise and timely diagnosis for effective intervention, despite the potential for errors and time-consuming methods. Despite various techniques used for diagnosing and categorizing this disease, there is a growing requirement for improved precision in early detection. This article proposes a deep learning technique for detecting and categorizing this ailment into different categories: non-dementia, very mild, mild, and moderate dementia. It does this by employing convolution neural network(CNN) topologies. The suggested approach can be used to analyze and categorize Alzheimer's patients in real time.

Keywords:Alzheimer's disease, CNN, MRI, Deep Learning, neurodegenerative.

I. INTRODUCTION

The neurodegenerative disorder known as AD is a worldwide condition that is typified by the build-up of β-amyloid, which leads to the creation of intracellular tau tangles and extracellular plaques[1]. This disease is a cognitive impairment that primarily affects the elderly; ten percent of cases have an untimely inception and afflict people under the age of 65. This disease also affects memory loss, ability to think, talking, mind, and comprehension. Individuals with symptoms of this ailment can receive care from professionals. Two-thirds of instances of dementia, mainly AD, are related to aging. Dementia is a cognitive deterioration that interferes with day-to-day functioning.

In 2020, AD ranked as the 6th most widespread source of casualty in the US. While there are therapies for AD, a full cure is yet unattainable [2]. In contrast to the DSM-5 categorization, AD is divided into four categories: non-dementia, very-mild dementia, mild dementia, and moderate dementia[3]. Short-term memory loss and language impairments are two signs of AD that frequently manifest in the early stages, making prompt and efficient treatment difficult.

As at present there is no proven treatment for AD, timely discovery of the condition may result in viable therapies. Although manual analysis can be tedious and time-consuming, early identification of AD frequently necessitates a neuropsychological test, with clinicians having a critical role in assessing patients [3,4]. Timely diagnosis of this disease is possible thanks to medical specialists, however manual analyses are impractical due to large data sets and patient counts. This raises the possibility of erroneous assessments and further consequences. Medical technology improvements depend heavily on automated patient imaging data analysis expertise, such as IoMT, machine and deep learning [5]. The MICCAI BRATS competitions analyze MRI or CT images using recent learning techniques [6]. Figure 1 illustrates how the suggested approach, which produced an overall classification accuracy of 83%, greatly enhanced the diagnostic performance of late-onset AD in patients 65 years of age and older.

1.1 Main contribution

This research is a focus on the analysis and categorization of AD utilizing an image dataset. Given the subtle nature of AD symptoms, there remains a need to tackle the identified research issues. The following points highlight the key contributions of this study to AD research:

- The study offers a method for accurately and promptly diagnosing AD in its early stages, taking into account the disease's mild symptoms.
- With a few familial cases connected to genetic abnormalities, the research concentrates on the diagnosis rather than exploring the illness's unexplained origin.

[1]Department of Computer Science & Engineering, NERIST, Itanagar, India. shashimourya@gmail.com

[2]Department of Computer Science & Engineering, NERIST, Itanagar, India. aky@nerist.ac.in

- Using convolution neural network (CNN) architectures, the suggested techniques for classifying normal or abnormal images across the 4 phases of AD are validated.
- The research technique, addressing stated issues is discussed in the succeeding section.

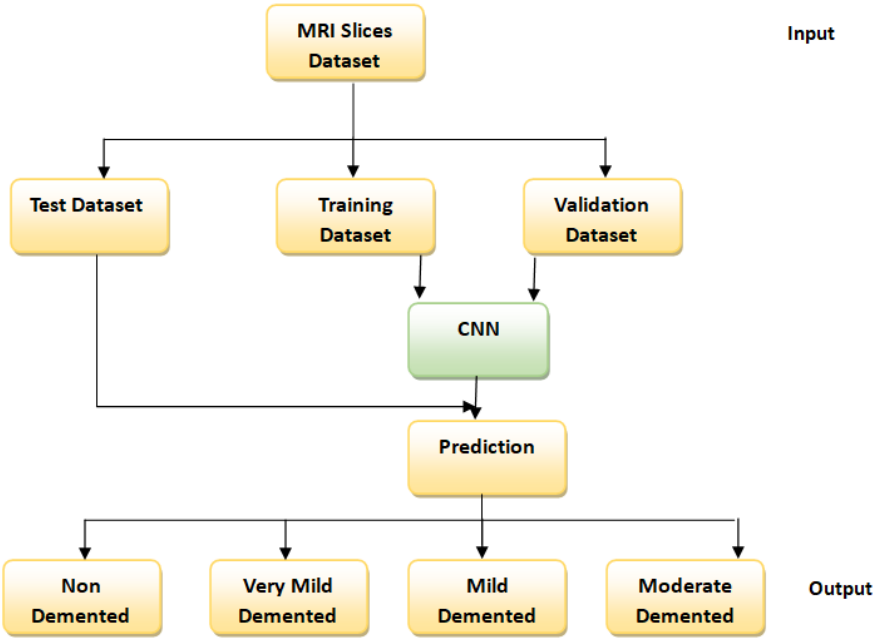

Figure 1. Flowchart for Alzheimer's disease prediction Model

II. Materials And Method

This research investigates diverse techniques to tackle the identified research issues in order to combat this widespread disease. Several points requiring attention are discussed below, with some currently under examination.

- Early diagnosis of AD is crucial for timely treatment, and automated techniques are needed to manage extensive medical image data from patients [7].
- The key reason of the disease remains unknown, except in rare familial cases attributed to genetic mutations **[8].**
- Presently, none specific treatment exists for the disease, highlighting the need for a solution capable of managing large image data to address numerous patients**[9]**.

This study proposes a deep learning model using CNN designs for early identification and categorization of AD.

III. Methodology

Figure 2 illustrates the research methodology employed in this study to develop a solution for the precise and early identification of AD. The offered research technique seeks to address the issues raised in the introduction. Earlier debates focused on deep learning-based algorithms, however these approaches fall short of detecting AD early, whether symptoms are present or absent.

This work focused on CNN-based deep learning models, specifically CNN, used to diagnose and classify AD.

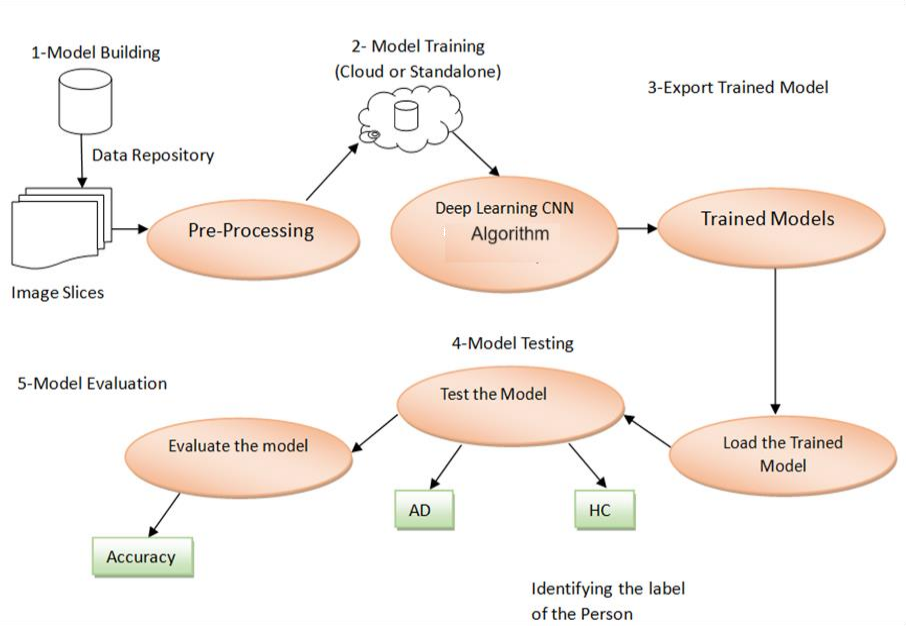

Figure 2. Proposed Methodology for Model

3.1 Pre-processing:

The image in the dataset is loaded, modified, and verified during the model development phase in order to produce enough data for deep learning models. To improve the performance of the model and increase the size of the original dataset, data augmentation techniques are applied. Four categories—non-dementia, very mild dementia, mild dementia, and moderate dementia—were applied to the dataset. To make analysis easier, each image was given a unique label. The dataset was prepared for further processing after completing these procedures.

3.2 Model building

CNN, used for feature categorization, has several layers. Looking into the model design, our CNN has 22 layers in total. 9 layers of convolution, 4 layers with maximum pooling, 6 dropout layers, and 3 thick layers make up this configuration. Throughout the model, ReLU activation functions are used; the output layer employs a softmax activation function. The following describes many typical layers seen in CNN architectures.

3.3 The input layer

The dataset's image size is 150 * 150 * 3 (width * height * channels), and the primary layer of CNN is used as the input layer. Since automatic image shuffling is implemented during each training period, it is deemed superfluous.

3.4 The convolutional layer

This serves as the primary master layer for feature maps and crucial parameters, forming the central component of the CNN architecture. The choice of kernel affects how well these layers perform. Padding is used in the feature maps and convolutional layer to align the dimensions of the input and output layers. The default padding assignment is determined by presets, which are normally set to one.

3.5 Batch normalization layer

Batch normalization is a time-efficient optimization technique for network training, facilitating gradient normalization and network propagation, a technique that combines non-linearity and convolutional layers.

3.6 ReLU layer

This is the activation function that neural network use most commonly. The batch normalizing layer is applied first, and then the ReLU layer.

3.7 Max-pooling layer

Spatial characteristics are often extensive. This layer is crucial for shrinking the feature map and eliminating unnecessary geographical data. Downsampling keeps crucial information in the feature maps while reducing computing costs.

3.8 Fully connected layer

Layers are appropriately denoted by the names. At this juncture, all the preceding layers of learning are interconnected. The final layer amalgamates the knowledge acquired in the previous stages and is responsible for model classification. The no of O/P matches the no of classes in the data set. There are four distinct classes in the dataset used in this study.

3.9 Softmax layer

The output of the fully connected layer lacks normalization. To address this, normalization is performed using an output Softmax layer. An integer result that is positive and suitable for categorization is produced by this layer.

3.10 Classification layer

This is the end layer in the CNN design and is used for categorization by the Softmax layer. In addition to computing the relevant loss values, as illustrated in figure 3, it generates probabilities for every input image in order to verify the manually chosen classes.

The CNN's comprehensive and generalized architecture is depicted in the accompanying figure 3. CNN architectures, which are further subdivided into several dense blocks within the architecture, are utilized in this study.

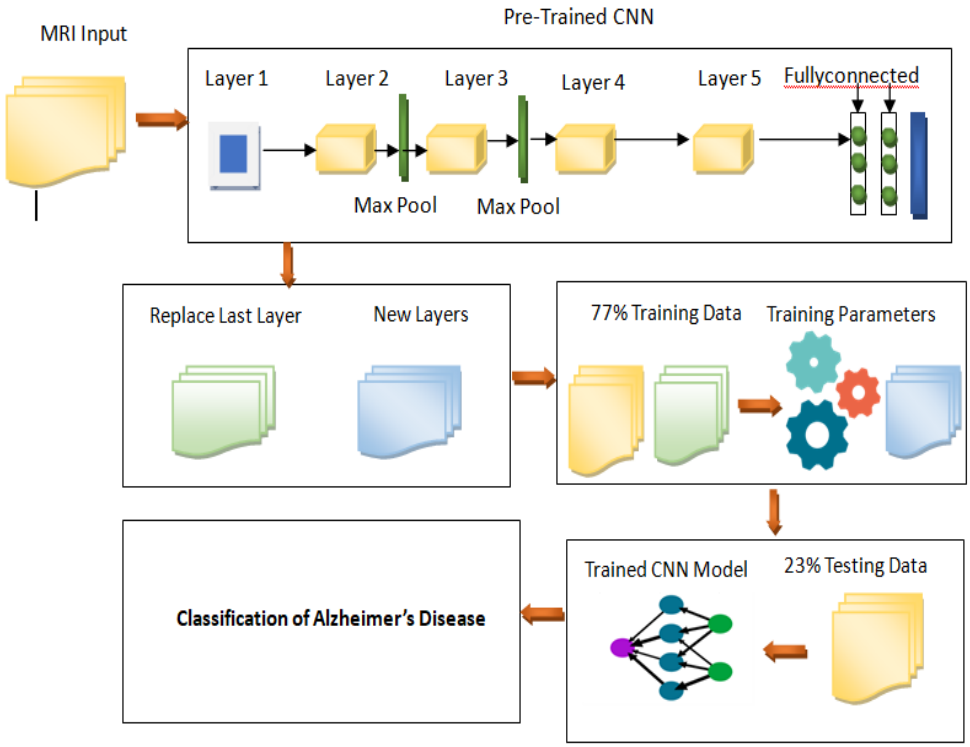

Figure 3. CNN Algorithm Working Architecture

3.11 Model training

Training comes next, after the deep learning model architecture has been developed successfully. Each training epoch involves rearranging the entire dataset, and this process is repeated 100 times. In this study, twenty three percent of the data are reserved for testing, while the remaining seventy percent are used for model training. The results of the experiment are shown in figure 6(A)(B).

3.12 Export-trained model

Figure 8 illustrates model summary how the models were exported for testing and usage after training was finished. After all of the dataset has been used for training, the training process does not need to be repeated. The models that have been trained are used for further processing.

3.13 Testing model

Using a variety of photos, the trained model is evaluated during the testing step. The model's job is to identify the four different categories and diagnose AD. Subsequently, an analysis is conducted to compare the model's output values with the real values that match the input photographs. The model's performance is assessed using this comparison, which was carried out on the testing dataset as shown in figure 4.

```
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model

# Load the trained model
model = load_model("/content/cnn_model.h5")

# Define class labels
class_labels = {0: "Mild_demented", 1: "Moderate", 2: "Non_demented", 3: "Very_mild_demented"}

# Define paths to test images
test_image_paths = [
    "/content/drive/MyDrive/80 percent dataset/6400_1/Dataset_6400/Non_Demented/non_1650.jpg",
    "/content/drive/MyDrive/80 percent dataset/6400_1/Dataset_6400/MildDemented/26 (19).jpg",
    "/content/drive/MyDrive/80 percent dataset/6400_1/Dataset_6400/Moderate Demented/Moderate new 1 (1).jpg",
    "/content/drive/MyDrive/80 percent dataset/6400_1/Dataset_6400/Very_Mild_Demented/verymild.jpg"
]
    if img_array.shape[1:] == input_shape:
        # Get the predicted class probabilities
        predictions = model.predict(img_array)

        # Get the predicted class label
        predicted_class_index = np.argmax(predictions)
        predicted_class = class_labels[predicted_class_index]

        # Print the prediction result
        print("Predicted Class:", predicted_class)
    else:
        print(f"Error: Expected input shape {input_shape} but got {img_array.shape[1:]}")
```

```
1/1 [==============================] - 0s 62ms/step
Predicted Class: Non_demented
1/1 [==============================] - 0s 19ms/step
Predicted Class: Mild_demented
1/1 [==============================] - 0s 20ms/step
Predicted Class: Moderate
1/1 [==============================] - 0s 21ms/step
Predicted Class: Very_mild_demented
```

Figure 4. Disease Detection using CNN Algorithm

3.14 Model evaluation

The model is evaluated using both the training model and the testing dataset. As can be seen in figure 7, for each image in the testing dataset, the evaluation metric is determined by comparing the estimated values obtained from the model with the confirmed true values. These evaluation indicators demonstrate the effectiveness and efficiency of the model's operation. Accuracy is the evaluation statistic that's utilized to gauge how important the model is. It is applied to the performance analysis of the proposed model.

3.15 Measurement of Model Performance

Measurements of recall, accuracy, precision, and F1-score were taken to evaluate the model's performance. The confusion matrix, which was created by comparing the actual conditions with the model's test results, was used to compute these measures. The confusion matrix yielded values for true positive (TP), true negative (TN), false positive (FP), and false negative (FN), as shown in figure 5.

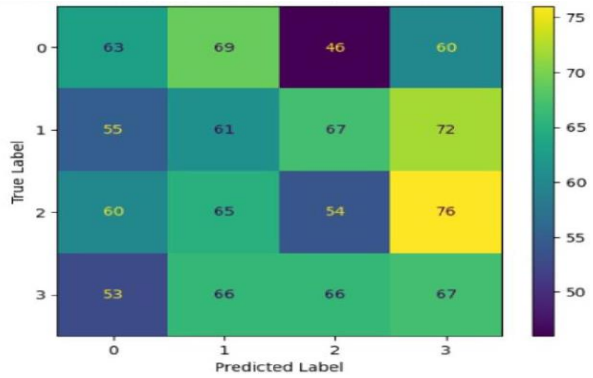

Figure 5. Alzheimer's disease (AD) Prediction Model (Confusion Matrix)

'False' represents instances of incorrect predictions, and 'True' indicates test samples that were correctly predicted. TP denotes the accurate prediction of a positive test sample as positive, TN denotes the accurate prediction of a negative test sample as positive, FP denotes the incorrect prediction of a negative test sample as positive, and FN denotes the inaccurate prediction of a positive test sample as negative. Accuracy indicates the percentage of correct predictions out of all test outcomes. Recall is the percentage of actual positive samples that were precisely predicted, whereas precision is the ratio of actual positives to those projected as positives. The F1-score functions as an adjusted statistic to mitigate any misjudgments resulting from sample bias, as it is a harmonic mean of precision and recall as shown in Table 1.

Table 1: Performance of Proposed Model

Model	Accuracy	Precision	Recall	f1-Score
M5	**83**	**0.666**	**0.666**	**0.906**

IV. Experimental Results And Discussion

The dataset that was utilized to train and test the model is described in this section, along with the findings of the experimental evaluation that was carried out for the given study.

4.1 Dataset

The proposed model was tested using an open-source dataset from the Kaggle platform [10]. With the assistance of a radiology specialist, the dataset was manually annotated after being compiled from multiple sources.

This dataset consists of magnetic resonance imaging (MRI) scans that have been classified into four distinct classes: Very Mild-Dementia, Moderate Dementia, Non-Dementia, and Mild Dementia. Both training and evaluation is possible with these image classifications. Two main folders are used to store the data: one for training and the other for testing. For validation, 1,023 out of 4,098 photos in the training folder were used.1,279 photos were in the test folder at the same time. The complete dataset was used in a 77–23% split for training and testing.

4.2 Results using CNN Deep learning Algorithm

```
32/32 [==============================] - 44s 1s/step - loss: 0.8524 - accuracy: 0.5988 - val_loss: 0.6082 - val_accur
Epoch 2/100
32/32 [==============================] - 38s 1s/step - loss: 0.6167 - accuracy: 0.7082 - val_loss: 0.6057 - val_accuracy: 0.7104
Epoch 3/100
32/32 [==============================] - 38s 1s/step - loss: 0.6093 - accuracy: 0.7104 - val_loss: 0.6113 - val_accuracy: 0.7104
Epoch 4/100
32/32 [==============================] - 40s 1s/step - loss: 0.6103 - accuracy: 0.7104 - val_loss: 0.6080 - val_accuracy: 0.7104
Epoch 5/100
32/32 [==============================] - 38s 1s/step - loss: 0.6089 - accuracy: 0.7104 - val_loss: 0.6039 - val_accuracy: 0.7104
Epoch 6/100
32/32 [==============================] - 38s 1s/step - loss: 0.6066 - accuracy: 0.7104 - val_loss: 0.6021 - val_accuracy: 0.7104
Epoch 7/100
32/32 [==============================] - 39s 1s/step - loss: 0.6035 - accuracy: 0.7104 - val_loss: 0.6034 - val_accuracy: 0.7104
Epoch 8/100
32/32 [==============================] - 39s 1s/step - loss: 0.5853 - accuracy: 0.7044 - val_loss: 0.5417 - val_accuracy: 0.7104
Epoch 9/100
32/32 [==============================] - 38s 1s/step - loss: 0.5606 - accuracy: 0.7142 - val_loss: 0.5358 - val_accuracy: 0.7190
Epoch 10/100
32/32 [==============================] - 39s 1s/step - loss: 0.5622 - accuracy: 0.7206 - val_loss: 0.5313 - val_accuracy: 0.7092
Epoch 11/100
32/32 [==============================] - 38s 1s/step - loss: 0.5474 - accuracy: 0.7212 - val_loss: 0.5731 - val_accuracy: 0.7396
Epoch 12/100
32/32 [==============================] - 38s 1s/step - loss: 0.5357 - accuracy: 0.7307 - val_loss: 0.5213 - val_accuracy: 0.7533
Epoch 13/100
32/32 [==============================] - 40s 1s/step - loss: 0.5344 - accuracy: 0.7402 - val_loss: 0.5006 - val_accuracy: 0.7602
Epoch 14/100
```

Figure 6(A). Training Accuracy using CNN Deep learning Algorithm

```
Epoch 86/100
32/32 [==============================] - 39s 1s/step - loss: 0.3844 - accuracy: 0.8164 - val_loss: 0.3419 - val_accuracy: 0.8363
Epoch 87/100
32/32 [==============================] - 37s 1s/step - loss: 0.3825 - accuracy: 0.8157 - val_loss: 0.3235 - val_accuracy: 0.8357
Epoch 88/100
32/32 [==============================] - 37s 1s/step - loss: 0.3751 - accuracy: 0.8268 - val_loss: 0.3267 - val_accuracy: 0.8392
Epoch 89/100
32/32 [==============================] - 39s 1s/step - loss: 0.3893 - accuracy: 0.8107 - val_loss: 0.3315 - val_accuracy: 0.8474
Epoch 90/100
32/32 [==============================] - 37s 1s/step - loss: 0.3761 - accuracy: 0.8246 - val_loss: 0.3738 - val_accuracy: 0.8141
Epoch 91/100
32/32 [==============================] - 37s 1s/step - loss: 0.3861 - accuracy: 0.8135 - val_loss: 0.3670 - val_accuracy: 0.8278
Epoch 92/100
32/32 [==============================] - 38s 1s/step - loss: 0.3694 - accuracy: 0.8249 - val_loss: 0.3522 - val_accuracy: 0.8344
Epoch 93/100
32/32 [==============================] - 37s 1s/step - loss: 0.3561 - accuracy: 0.8243 - val_loss: 0.3409 - val_accuracy: 0.8351
Epoch 94/100
32/32 [==============================] - 37s 1s/step - loss: 0.3654 - accuracy: 0.8214 - val_loss: 0.3290 - val_accuracy: 0.8490
Epoch 95/100
32/32 [==============================] - 37s 1s/step - loss: 0.3805 - accuracy: 0.8170 - val_loss: 0.3533 - val_accuracy: 0.8319
Epoch 96/100
32/32 [==============================] - 38s 1s/step - loss: 0.3678 - accuracy: 0.8284 - val_loss: 0.3345 - val_accuracy: 0.8513
Epoch 97/100
32/32 [==============================] - 37s 1s/step - loss: 0.3771 - accuracy: 0.8281 - val_loss: 0.3448 - val_accuracy: 0.8379
Epoch 98/100
32/32 [==============================] - 37s 1s/step - loss: 0.3727 - accuracy: 0.8211 - val_loss: 0.3147 - val_accuracy: 0.8541
Epoch 99/100
32/32 [==============================] - 38s 1s/step - loss: 0.3715 - accuracy: 0.8319 - val_loss: 0.3365 - val_accuracy: 0.8481
Epoch 100/100
```

Figure 6(B). Training Accuracy using CNN Deep learning AlgorithmResult accuracy 83%

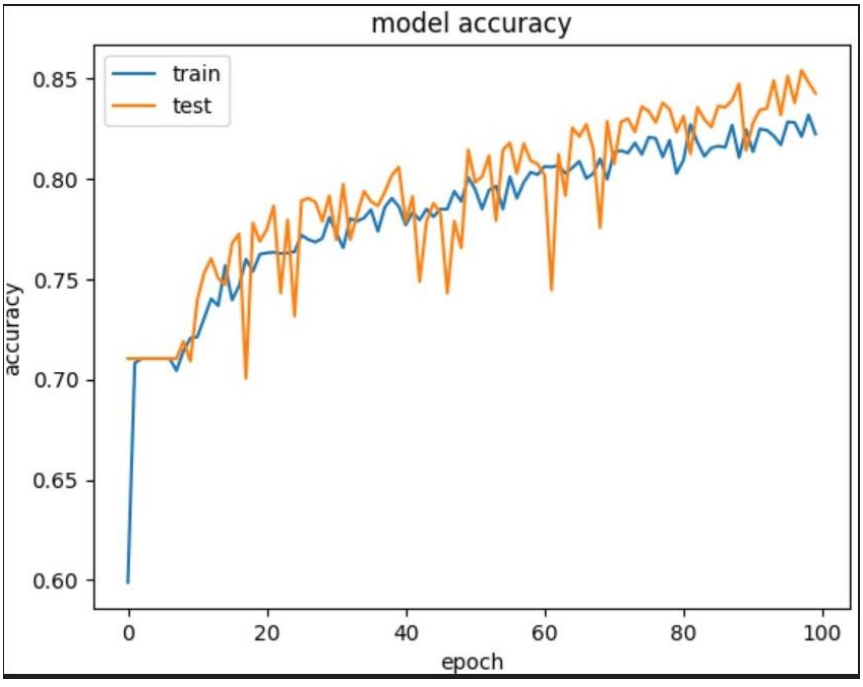

Figure 7. Model Accuracy

```
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
conv2d_1 (Conv2D)	(None, 146, 146, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 73, 73, 64)	0
dropout (Dropout)	(None, 73, 73, 64)	0
conv2d_2 (Conv2D)	(None, 71, 71, 64)	36928
conv2d_3 (Conv2D)	(None, 69, 69, 64)	36928
dropout_1 (Dropout)	(None, 69, 69, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 34, 34, 64)	0
dropout_2 (Dropout)	(None, 34, 34, 64)	0
conv2d_4 (Conv2D)	(None, 32, 32, 128)	73856
conv2d_5 (Conv2D)	(None, 30, 30, 128)	147584
conv2d_6 (Conv2D)	(None, 28, 28, 128)	147584
conv2d_5 (Conv2D)	(None, 30, 30, 128)	147584
conv2d_6 (Conv2D)	(None, 28, 28, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 128)	0
dropout_3 (Dropout)	(None, 14, 14, 128)	0
conv2d_7 (Conv2D)	(None, 12, 12, 128)	147584
conv2d_8 (Conv2D)	(None, 10, 10, 256)	295168
max_pooling2d_3 (MaxPooling2D)	(None, 5, 5, 256)	0
dropout_4 (Dropout)	(None, 5, 5, 256)	0
flatten (Flatten)	(None, 6400)	0
dense (Dense)	(None, 512)	3277312
dense_1 (Dense)	(None, 512)	262656
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 4)	2052

```
Total params: 4447044 (16.96 MB)
Trainable params: 4447044 (16.96 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 8. Model Summary

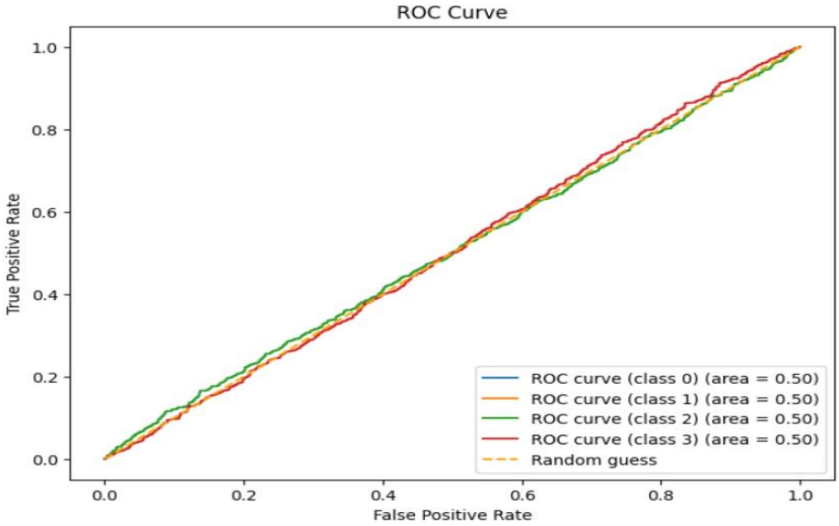

Figure 9. ROC Curve

4.3 Analysis and comparison

The suggested model performed better than the different techniques conversed in the literature section, according to the statistical and graphical data. The proposed model provides an accurate early diagnosis and correct categorization. The approach can be applied in real-time scenarios and is helpful for correctly categorizing AD. A larger Maximizing ROC curve indicates a model's Capacity to transform between +ve and -ve categories (AUC) as shown in figure 9.

V. CONCLUSION AND FUTURE WORK

AD is known to be an irreversible neurological illness that affects memory, especially in the elderly. Manual diagnosis is impractical due to the sheer amount of patients, and errors may occur since health specialists have limited time and must navigate a complex process. Although there are several methods for identifying and diagnosing this disease, a precise and quick diagnostic answer is needed.

The given model supports a deep learning-based approach to AD diagnosis and classification using CNN architectures. The model classifies this condition into 4 categories: very mild dementia, mild dementia, moderate dementia, and non-dementia. In the phases of testing and training, the CNN method performed better. Real-time categorization and analysis are possible with this suggested methodology. Future goals entail expanding the datasets used in the sickness identification process and assessing the system's accuracy using a range of indicators.

REFERENCES

[1] P. G. Altbach, L. Reisberg, and Knopman DS, Amieva H, Petersen RC, Chételat G, Holtzman DM, Hyman BT, Nixon RA,Jones DT. 2021. Alzheimer disease. Nature Reviews Disease Primers 7(1):1–21DOI 10.1038/s41572-021-00269-y.

[2] Kumar A, Sidhu J, Goyal A, Tsao JW. 2018. Alzheimer disease. StatPearls. 1–27. Available athttp://europepmc.org/books/
NBK499922 (accessed 4 December 2022).

[3] Salehi AW, Baglat P, Gupta G. 2020. Alzheimer's disease diagnosis using deep learningtechniques. International Journal of Engineering and Advanced Technology 9(3):874–880DOI 10.35940/ijeat.C5345.029320.

[4] Butt AUR, Ahmad W, Ashraf R, Asif M, Cheema SA. 2019. Computer aided diagnosis (CAD) forsegmentation and classification of burnt human skin. In: 2019 International Conference onElectrical, Communication, and Computer Engineering (ICECCE). Piscataway: IEEE, 1–5.

[5] Kundaram SS, Pathak KC. 2021. Deep learning-based alzheimer disease detection. In: Proceedingsof the Fourth International Conference on Microelectronics, Computing and CommunicationSystems. Singapore: Springer, 587–597.

[6] Frozza RL, Lourenco MV, De Felice FG. 2018. Challenges for Alzheimer's disease therapy:insights from novel mechanisms beyond memory defects. Frontiers in Neuroscience 12:37DOI 10.3389/fnins.2018.00037.

[7] Bilal M, Barani M, Sabir F, Rahdar A, Kyzas GZ. 2020. Nanomaterials for the treatment anddiagnosis of Alzheimer's disease: an overview. NanoImpact 20:100251DOI 10.1016/j.impact.2020.100251.

[8] DeTure MA, Dickson DW. 2019. The neuropathological diagnosis of Alzheimer's disease.Molecular Neurodegeneration 14(1):1–18 DOI 10.1186/s13024-019-0333-5.

[9] Ulep MG, Saraon SK, McLea S. 2018. Alzheimer disease. The Journal for Nurse Practitioners14(3):129–135 DOI 10.1016/j.nurpra.2017.10.014.

[10] Kaggle. 2019. Alzheimer's dataset (4 class of Images). Available at https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images (accessed 20 January 2022).