Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Alzheimer's Disease Prediction Using Deeplearning Algorithms

M. Rupa, Dr. K. Rajkumar

[1]*M.Tech, MISTE, Lecturer/ECE in Government Institute of Electronics, Secunderabad*
[2]*Associate Professor, SR University*

Alzheimer's disease is a degenerative brain illness that cannot be reversed. Alzheimer's disease is now being discovered in the globe every four seconds. As a consequence, the outcome is lethal, as it results in death. As a consequence, detecting the illness at an early stage is critical. It is estimated that Alzheimer's disease is the biggest cause of dementia in the world. A decrease in cognitive abilities and interpersonal coping skills is caused by dementia, which reduces a person's capacity to operate on their own. In the early phases, the patient will forget previous occurrences. Their memories will begin to fade as their disease worsens. The sooner the condition is diagnosed, the better. Brain MRI pictures may be used to assess whether someone has mild, moderate or no Alzheimer's disease as a result of the model proposed in this research. A comparative comparison of the ResNet50,Inception V3,Inception Resnet V2,Densenet, and MobileNet designs are used for this categorization.

Keywords: Alzheimer's, Deeplearning Algorithms.

1. Introduction

The system's behaviour and ideas are depicted in its architecture. It's a view that shows how our project modules' models are trained using the database. The following graphic shows how training dataset data is fed to models. After that, it's put to the test against a real dataset to see whether it works as expected. Sick images from the dataset are categorised as mild, moderate, non-demented, and very mild demented after comparing their accuracy. The architectural diagram shows how the project's components are connected and integrated in order to get the desired results.

Biomarkers based on Machine Learning are used in their study. They tested a customised disease classifier based on local weights and biomarkers. In this method, biomarkers are first categorised and then ordered. Those who acquired AD were compared to those who did not. .(Subramani, P., et al., 2021.)

In this work, the MRI Alzheimer's detection accuracy of two cutting-edge deep learning algorithms is compared to see which is better. Deep learning library Keras is used to implement VGG19. Image Data Generator was used to enrich the model's data. 128 batches and 50 earlystopping epochs make up the training. Data is loaded into the Image Data Generator function by the Keras module, which uses it to create the Dense net Model. 128 images are used in the densenet model. MRI images from four classes are used to train both models, which are then tested on a total of 2067. Everything was done in Google colab.

2. LITERATURE REVIEW

Alzheimer is a kind of dementia that causes memory loss. To cut down on the death toll from Alzheimer's disease, scientists are focusing on early detection of the disease. Computer-aided diagnosis in medicine employs deep learning techniques. Deep Neural Network (DNN) with the corrected Adam optimizer is used in this research to automatically identify Alzheimer's disease (AD) from MRI data. HOG is a tool for extracting feature values from NIMHANS and ADNI databases' images of the brain. A corrected Adam optimizer was used to input the retrieved features into DNN .(Varatharajan., et al., 2018) which was then used to identify healthy individuals, those with Alzheimer's disease, and people suffering from mild cognitive impairment . A 16 percent improvement in classification accuracy was seen in the HOG-DNN with the corrected Adam optimizer compared to the landmark-based features with the SVM classifier (Suresha, H. S., & Parthasarathy, 2020).

Degeneration of brain cells occurs as a result of Alzheimer's disease (AD). Many senior people worldwide suffer from memory loss and cognitive impairment as a result of Alzheimer's disease. Healthcare's primary goals include early detection and the simplification of diagnostic procedures. Machine learning (ML) and multivariate data exploration approaches are used in the research of AD. Automatic categorization is presented in this study in order to discover information patterns

(Pavitra, B et al., 2020). Our five-stage machine learning pipeline comprises sub-levels at each stage. MRI (Magnetic Resonance Imaging) brain images from the OASIS database were analysed in this study.

A total of 150 MRI patients. MMSE, CDR, and ASF scores were used into the analysis for this project. Data transformation, feature selection, and a classifier system make up the ML pipeline. The best accuracy values are seen in the RF classifier. (Khan & Zubair S, 2020)

Evaluation of Random Forest (RF) ensemble classifiers in imputation and non-implementation, and their effect on Alzheimer's disease (AD) detection using longitudinal MRI data. Over the course of 373 MRI scans, researchers looked at 150 people with Alzheimer's disease (AD) ranging in age from 60 to 90 years old. To get T1-weighted MRI images, a 1.5-T Vision scanner was used. The OASIS database provided the MRI data. We used missing data imputation using RF ensemble to categorise people as demented or non-demented based on MRI characteristics. In AD diagnosis, RF model-based imputation analysis is superior than RF non-imputation (Asim, Y et al., 2018).

Images of Alzheimer's disease and moderate cognitive impairment (MCI) may be improved by using biomarkers (MCI). Brain parcellation is used to predict AD and MCI in computer-aided techniques for detection. The anatomical atlas templates used for brain parcellation differ in their boundaries and number of regions. Rather than relying on a single atlas to represent the whole brain, this research looks at how different atlases might be combined to provide a more complete picture. A combination of the ADNI database, the LONI Probabilistic Brain Atlas (LPBA40), and Automated Anatomical Labelling (AAL) was utilised to segment the brains (AAL). Gray matter density and glucose metabolism were measured using MRI and FDG-PET baselines in each region. We then used individual and combined atlas features to classify people with AD, MCI, and CN. SVMs and PCA both reduced the dimensionality of individual and combined attributes, but PCA (Singh, D. et al., 2020) reduced the dimensionality by a greater margin. Classification characteristics were assessed according to their usefulness in accurately classifying individuals. The findings showed that multiple atlas features outperformed single atlas features (Nawaz, A et al., 2020).

An Experimental Section:

In Deep Learning, low, middle, and high-level qualities are represented in a hierarchical manner. Complex data may be handled by deep neural networks. New data may be generalized more easily with the use of many layers. Deep Learning's algorithms are trained and evaluated using a variety of datasets as shown in Figure 1. Using layers, like neurons, deep learning is able to understand and comprehend data more effectively (Roshanzamir, A et al., 2021). There are many levels that process incoming data as it moves through them. The deep learning model's output is expected to be applied after the last layer, which is an activation function A. If the trained deep learning model is accurate, it allows us to predict or detect anything else This is how deep learning works (Fiscon, G., et al., 2014).

Deep neural networks can adapt to more complex data sets. It's better in generalizing previously unseen data because of its multiple layers (Doan, D. et al., 2021). We can predict or detect disease from the learned deep learning model (Nawaz, A et al., 2020)

(i) System Architecture:

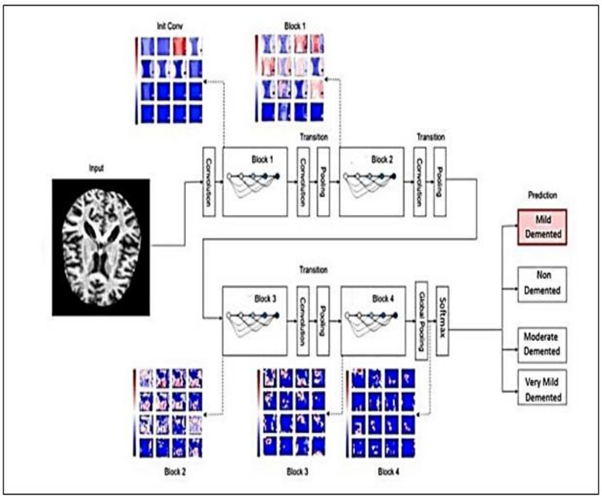

Figure 1. Architecture of the System

(ii) Data Flow Diagram: As shown in Figure 2

Step 1: A DFD is a bubble chart. Input, processing, and output data are all represented graphically in this way.

Step 2: A crucial modelling tool is the data flow diagram. System components are modelled in this way. An external entity, the system's data, and the way in which information travels through the system are all included in this list.

Step 3: A data flow diagram (DFD) shows how data moves through a system and how it is changed. It visually depicts the flow of data and the modifications that occur between the input and the output.

Step 4: A DFD chart is a bubble chart. It is possible to use DFDs to model any level of abstraction. DFD levels boost the flow of information and functional specificity

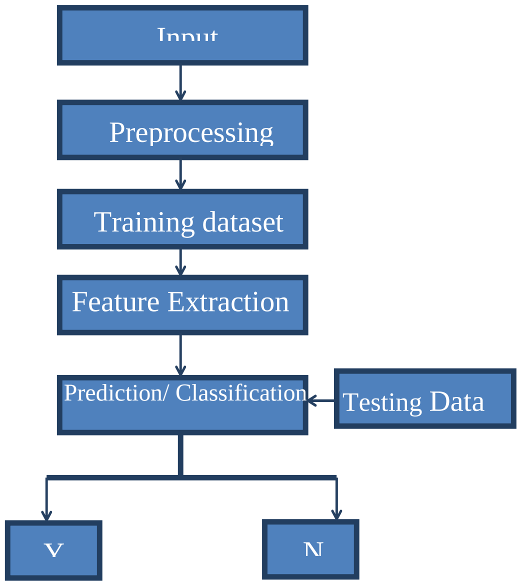

Figure 2. Data Flow Diagram

(iii) UML Diagrams

With UML, you may create models that are directed toward objects. The standard was developed by the Object Management Group.

As a software modelling language, UML intends to become the de facto standard. A Meta-model and a notation are included in UML. Methods and processes may one day be included in UML.

To describe, visualise, develop, and record both software and non-software objects using the Unified Modelling Language is a typical practise.

Engineering approaches such as UML are ideal for modelling large and complicated systems.

In order to design object-oriented software and to create software, UML is a must. The graphical notations of UML are used to represent software project designs.

(iv) Goals:

The following are the primary goals of UML:

- Assist users in creating and trading models by offering a ready-to-use visual modelling language to them. Specify and expand on the most important concepts. Be agnostic to language and procedure.

- Modelling language understanding should be formalised. Initiate and foster the expansion of OO tools.Collaborations, frameworks, patterns, and components may all benefit from this support.

- Make use of the finest practises that you can find.

(v) Use case diagram:

Use-case diagrams as shown in Figure 3 UML are representations of behaviour produced from use-case studies. A graphical depiction of system participants and their aims (represented as use cases) is shown. Use case diagrams demonstrate who is in charge of what components of the system. Actors may be seen in Character

Figure 3. Use case diagram

(vi) Class diagram:

Class diagrams may include object diagrams as shown in Figure 4. An item of a certain kind. One may think of an object as an abstraction of an instance of a class. The object diagram shows the many classes and relationships that exist inside a given system.

(vii) Object diagram:

Class diagrams take on a new twist with the addition of object diagrams. An object is a specific instance of a certain kind of class. An object reflects the current state of a class throughout the course of the system's operation (Varatharajan., et al., 2018). The object diagram depicts the current state of the system's many classes, as well as the connections or associations that exist between them at any one moment.

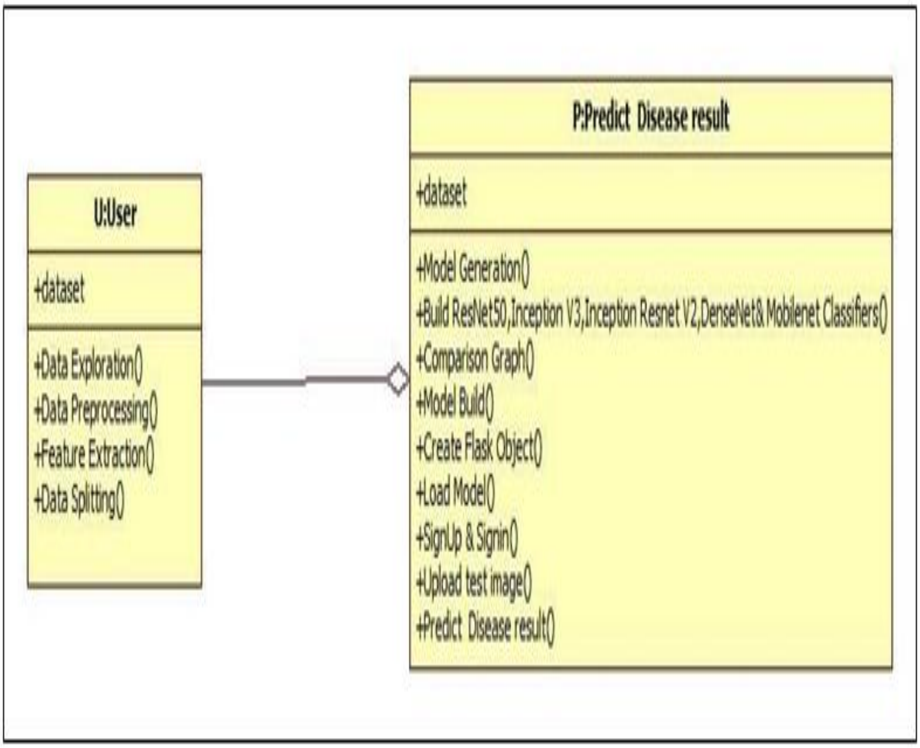

Figure4.Objectdiagram

(viii) State diagram: It is a representation of the numerous life cycle states that system objects go through, as its name suggests. Whenever something occurs within the system, the statuses of the objects undergo a change. An additional use of a state diagram is to document the progression of an object's state from its initial state to its final state as events have an effect on the system.

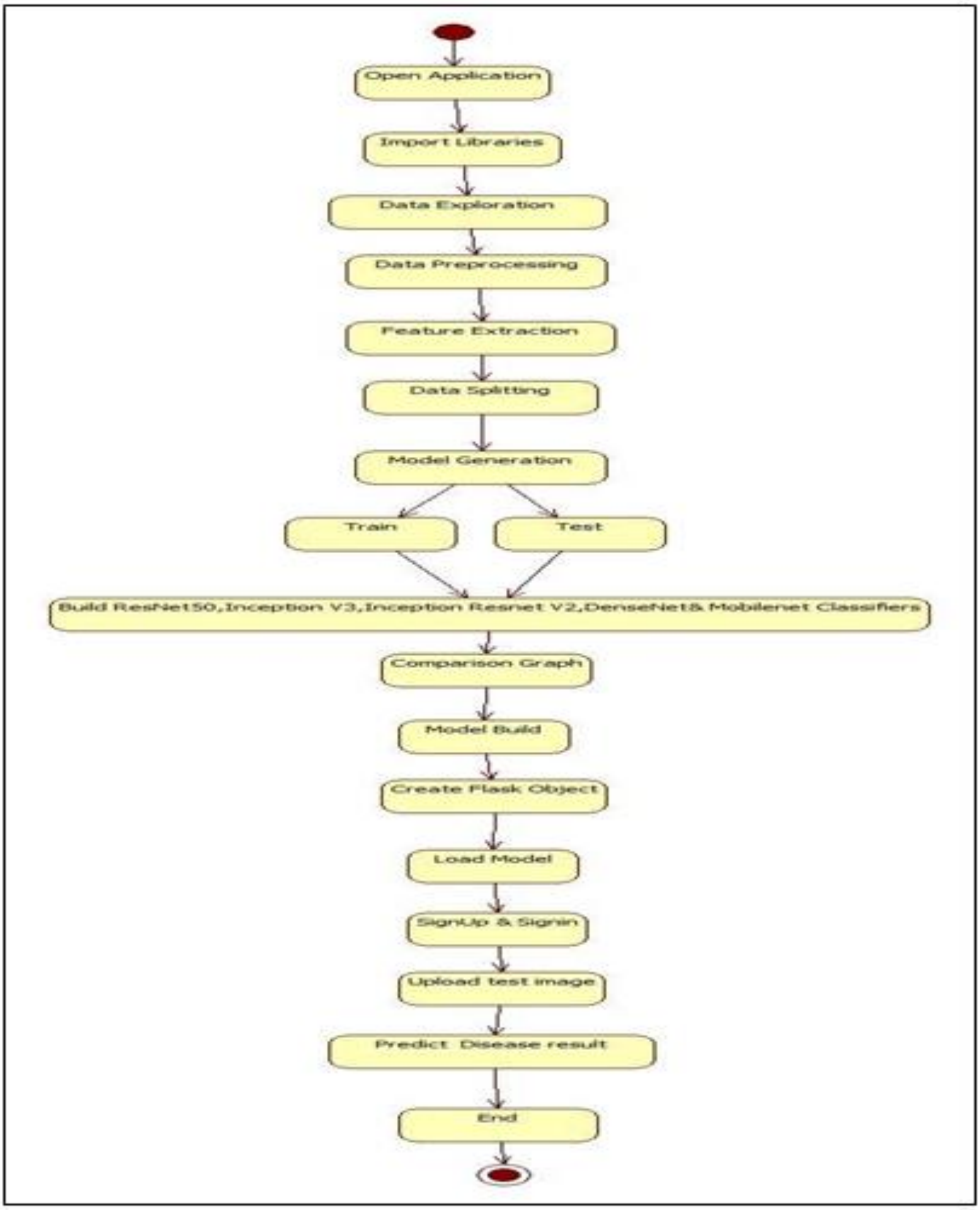

Figure 5. State diagram

(ix) Activity diagram:

The activity diagram, as seen in Figure 6, captures the system's process flows. The components of an activity diagram are quite similar to those of a state diagram: tasks, actions, transitions, starting and ending points, and safety criteria

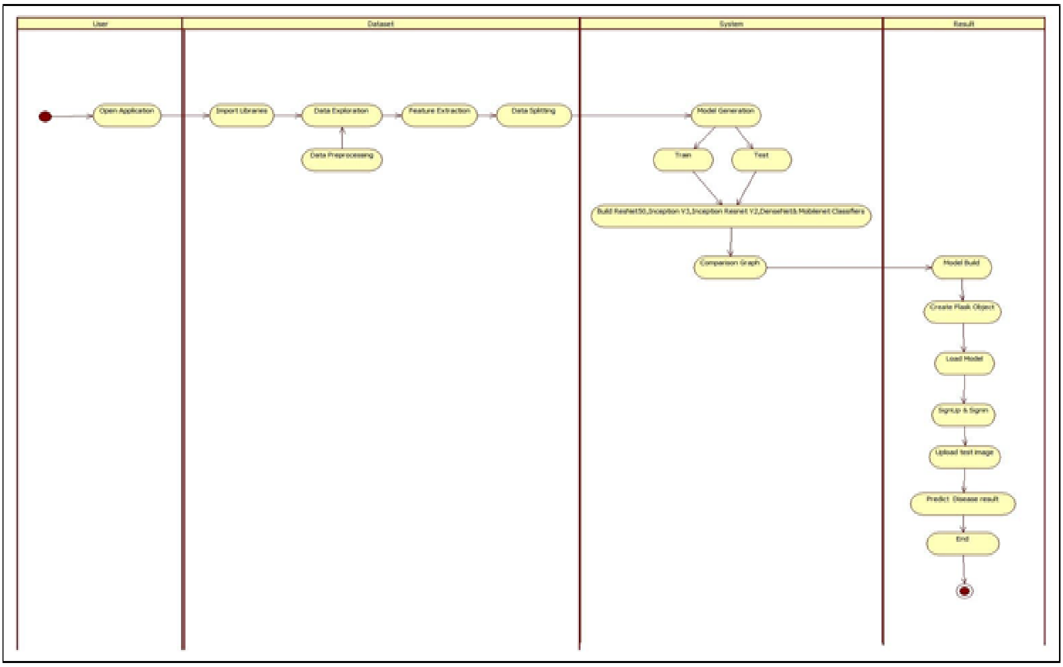

Figure 6. Activity diagram

(x) Sequence diagram:

Interactions between system objects are shown in sequence diagrams as shown in Figure 7. There are time-ordered sequence diagrams Detailed depiction of how objects interact with each other. Objects in a flowchart communicate by exchanging.

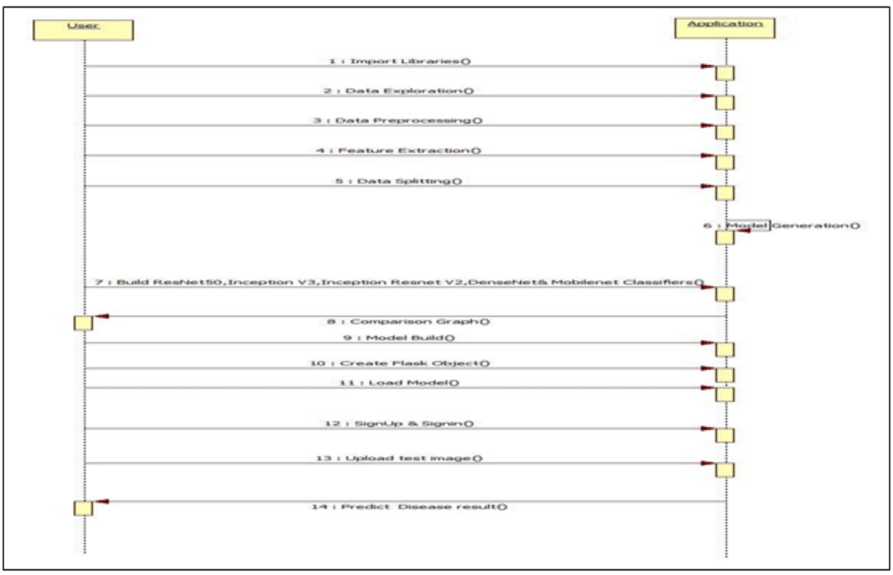

Figure 7. Sequence diagram

(xi) Collaboration diagram:

As shown in Figure 8 of item interactions are shown in collaboration diagrams. To keep track of their sequence, interactions are given numbers. The cooperation diagram depicts all interactions between the various objects in the diagram.

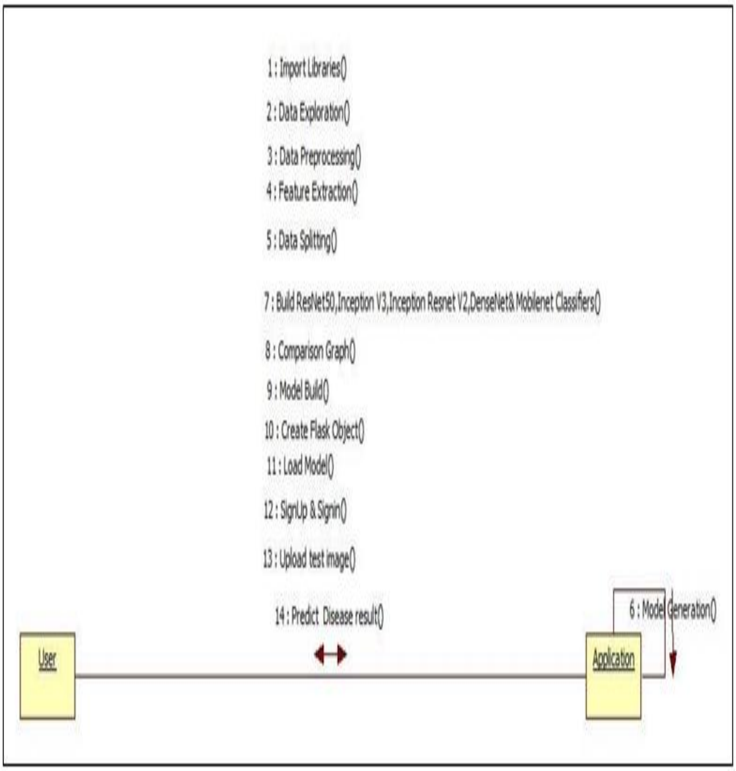

Figure 8. Collaboration diagram

- A Results section :

(i) Module Description

Data-gathering

The dataset's structure and links to other features. There is a two-thirds training and three-thirds testing ratio in our system. Training and testing datasets should be equivalent to a random sample in terms of their representation. Kinds of data that were used to train and test the paper

Pre-processing

Lack of data may lead to inconsistencies. Preprocessing data improves the algorithm's efficiency. Outliers should be removed from variables before they are converted. The map function addresses these concerns.

Modelling

Machine learning is all about predicting and recognising patterns. In order to train ML algorithms, data patterns are needed. Every attempt at training a machine learning model yields better results. It is necessary to separate the data into training and testing sets in order to assess the performance of a model. Training set (70 percent) and test set (30 percent) were segregated prior to training our models (30 percent). After that, we supplemented our model's predictions with actual performance data.

Make predictions

Loan from a financial institution Characteristics of the Defaulters dataset. The ability of a consumer to pay back a loan is not affected by all circumstances. The performance of the created system is tested. Behavior changes through time may be described and modelled using an evolutionary perspective. Misunderstandings abound when it comes to the terms precision and accuracy. DenseNet models must have the following features in order to be accurate predictors.

(ii) Algorithms:

DenseNet

DenseNet (Dense Convolutional Network) Shorter layer connections make deep learning networks deeper and cheaper to train. DenseNet. DenseNet is a convolutional neural network with every layer connected to the previous one.. As a result, data travels more quickly between the various levels of the network. Because of this, each layer takes input from all preceding layers and offers its own feature maps for use by all future layers. Resnets concatenates qualities rather than summarising them. Each convolutional block in the 'ith' layer has I inputs and feature maps. All 'I-i' layers have access to its feature maps. As opposed to normal deep learning architectures, this adds (I(I+1))/2 network links. Unlike ordinary convolutional neural networks, this one doesn't need to learn feature mappings that aren't essential. There are two critical components of DenseNet besides the convolutional and pooling layers. Layers of Transition and Dense Blocks as shown in Figure 9

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112		7 × 7 conv, stride 2		
Pooling	56 × 56		3 × 3 max pool, stride 2		
Dense Block (1)	56 × 56	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6
Transition Layer (1)	56 × 56		1 × 1 conv		
	28 × 28		2 × 2 average pool, stride 2		
Dense Block (2)	28 × 28	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12
Transition Layer (2)	28 × 28		1 × 1 conv		
	14 × 14		2 × 2 average pool, stride 2		
Dense Block (3)	14 × 14	[1 × 1 conv, 3 × 3 conv] × 24	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48	[1 × 1 conv, 3 × 3 conv] × 64
Transition Layer (3)	14 × 14		1 × 1 conv		
	7 × 7		2 × 2 average pool, stride 2		
Dense Block (4)	7 × 7	[1 × 1 conv, 3 × 3 conv] × 16	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48
Classification Layer	1 × 1		7 × 7 global average pool		
			1000D fully-connected, softmax		

Figure 9. Architectures of DenseNet for ImageNet

ResNet:

The pyramidal cells of the cerebral cortex serve as the basis for ResNet, an ANN. Skip connections are used in residual neural networks to reduce the number of layers. In typical ResNet models, nonlinearities (ReLU) and batch normalisation are sandwiched between layers of double- or triple-layer skips. as shown in Figure 10 The skip weights may be learned using a model known as HighwayNets, which utilises an extra weight matrix. (Deng, L., & Wang, Y,2021). DenseNets are models with several parallel skips (Khan, A., & Zubair, S,2020) A simple network is a good way to characterise a non-residual network. Reconstruction of a pyramidal cell.

The axon arbor is blue, and the soma and dendrites are red. This class includes the soma, basal, apical, axon, and collateral axons. Both to prevent vanishing gradients and the Degradation (accuracy saturation) issue are the major reasons to add skip connections. Adding additional layers in an appropriately deep model results in increasing training error. As you go through your training, the weights adjust to magnify the previously skipped layer while muting the upstream layer.

It is possible to adjust the weights of the next layer's connection only in the simplest case A single nonlinear layer or linear intermediary layers may benefit from this technique. If not, utilise a HighwayNet to find the specific weight matrix of the missing connection (a HighwayNet should be used). Initial training may be simplified by skipping network layers. This reduces the influence of fading gradients, which speeds up learning. Over time, the network's ability to catch up on lost levels improves as it picks up more feature space. It learns faster since it is closer to the manifold when all layers are increased. More features may be explored by unconstrained neural networks. Recovery from perturbation needs a large amount of training data.

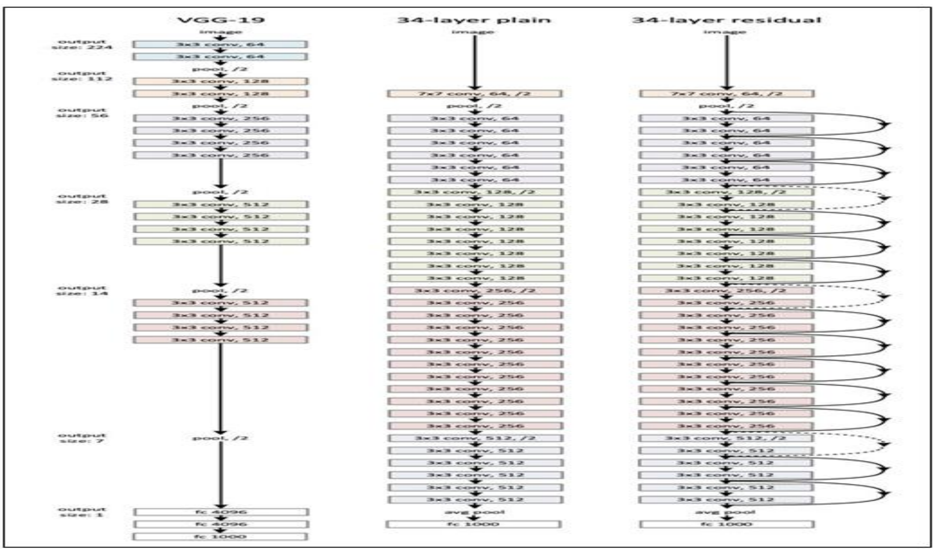

Figure 10. Example network architectures for ImageNet

MobileNet

MobileNet convolutions are depth-wise. exception introduces as shown in Figure 11 convolution block. A depth-wise separable convolution has point-wise and depth-wise portions. Standard convolution uses feature map spatial dimensions and input/output channels. Df is the input feature map dimension, M and N are input and output channels, and Dk is kernel size. Deep convolution maps one input channel per convolution. Input and output channels are equal. It costs Df2*M*Dk2.

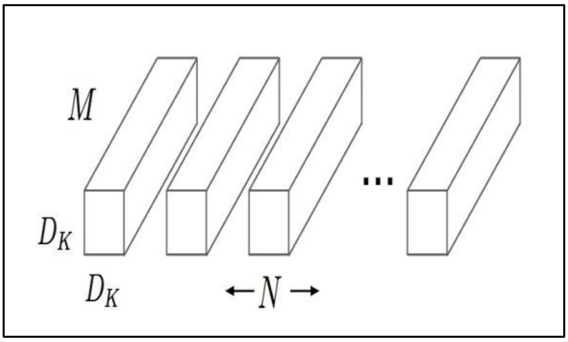

Figure 11. Standard Convolution Filters

- A Discussion section

Testing identifies flaws. It is only via thorough testing that a product's flaws and vulnerabilities are discovered. Parts, assemblies, and/or final products are all examined to see how effectively they perform their intended roles. Software testing is the process of putting a piece of software through its paces to ensure that it meets the expectations of its users and does not malfunction unexpectedly. Possibly, the results will vary. Each kind of test serves a certain purpose.

(i) Types of evaluation tests

Unit tests

Unit testing is the process of creating test cases to evaluate the program's inputs and outputs. The logic of the decision branches and the flow of the code should be examined. 'Individual components of an application are tested before they are merged, and that is what is meant here. This invasive procedure needs a thorough understanding of the structure in order to be successful. A unit test is a test of a business process, application, or system configuration at the component level. Every stage in a company's process must be tested to make sure it works as planned, and unit tests are a great way to do this.

Integration testing

To guarantee that the merged software can be used as a single application, integration tests are performed. The outcomes of a screen or field are the focus of event-driven testing. In order to ensure that a system's general functioning is proper and consistent, it is necessary to conduct integration tests. Integration testing is primarily concerned with identifying issues with specific components.

Black Box Testing

Black box testing is the practice of doing software examines the code and structure without understanding it. Similar to other types of testing, black box tests necessitate a document outlining criteria or specifications prior to their creation. As though the programme being tested were a black box, it is used for testing purposes. There are no considerations for programme functionality when evaluating input and output.

Unit Testing Unit testing is often carried out in conjunction with coding and unit testing throughout e software development lifecycle. Coding and unit testing may, and often do, take place at different points in the software development process.

Test strategy and approach

An extensive functional test set is being built for manual testing in the field.

The objectives of the test.

There must be no errors with any of the data entered into any of the fields.Go to the URL that has been supplied in order to activate pages.The input screen, messages, and responses should all be instantaneous.Criteria to be used for judging Verify that the data you've input is in the correct format before saving it.

You cannot have more than one entry in a row. Users should always arrive at their intended location when they click on any of the links provided. Inquiry into the integration process

The purpose of software integration testing is to create failures due to interface difficulties via the gradual integration testing of two or more integrated software components on a single platform.

Software system components or, to take it one step further, company-level software application components must communicate without causing any problems for the integration test to fulfil its task. There were no failures in any of the aforementioned tests. We discovered nothing wrong with the system.

Preliminary Testing

An important part of every project is the User Acceptance Testing phase, in which the end user plays an important role. In addition, it ensures that the system meets all of the functional criteria that were defined.All of the above-mentioned test scenarios were found to be successful, as shown by the test results. We discovered nothing wrong with the system. A few examples of the many kinds of tests you can encounter as mentioned in table 1.

Table 1. Parameters of model

Parameters	Description
Use case ID	Alziheimers Disease Detection Using MRI Images Based DeepLearing
Use case Name	Home button
Description	Display home page of application
Primary actor	User
Precondition	User must open application
Post condition	Display the Home Page of an application
Frequency of Use case	Many times
Alternative use case	N/A

Use case Diagrams	
Attachments	N/A

(ii) RESULTS

Figure 12,13,14 and 15 are the MRI scan results of the Alzmeirs patients of Normal and abnormal with mild,moderate and very mild cases and figure 16 and 17 are the graph of training loss and validation loss and graph training accuracy.

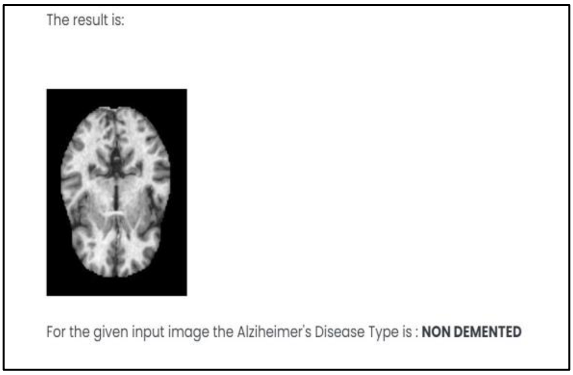

Figure 12. Non-Demented

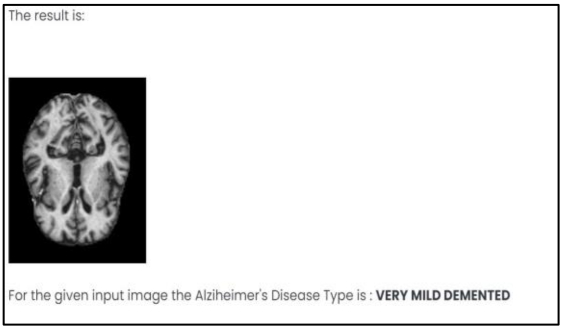

Figure 13. Very Mild Demented

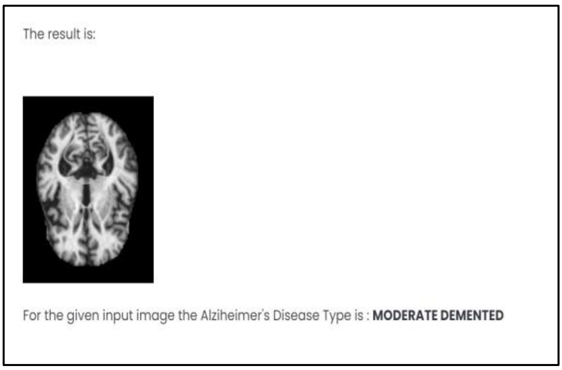

Figure 14. Moderate Demented

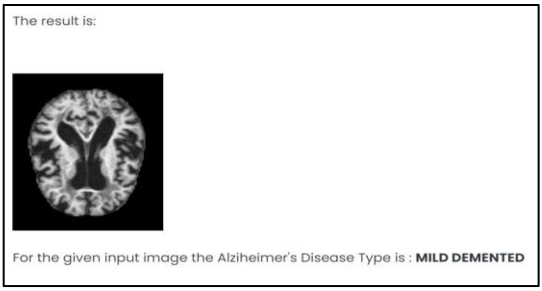

Figure 15. Mild Demented

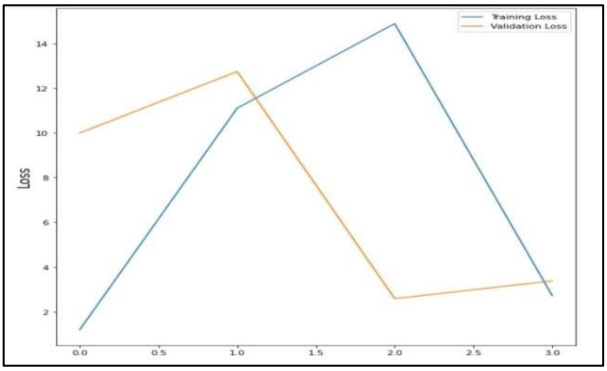

Figure 16. Graph of training loss and validation loss

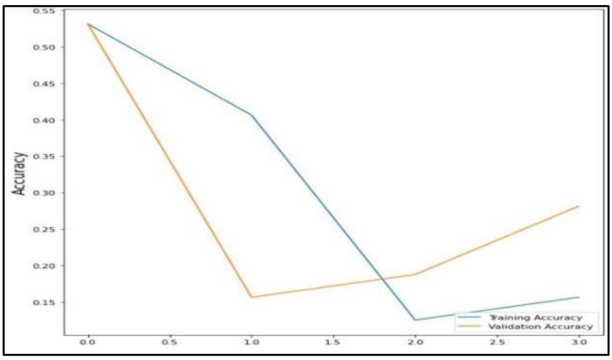

Figure 17. Graph of training accuracy and validation accuracy

3. CONCLUSION

Alzheimer's causes most dementia. This research proposes an early illness detection method. The models in this article correctly categorised pictures into four classes, yielding promising results. DenseNet outperforms ResNet50, Inception V3, Inception Resnet V2, and Mobilenet. More research is needed to deploy this approach in clinical settings and improve care for this condition. People should learn about this condition and be checked out. We're putting this model online for greater use.

Future enhancements

Future tests may be done on a bigger dataset. The 'Moderate Demented' dataset only comprised 52 training and 12 test photos. The suggested approach may help physicians better detect Alzheimer's and other neurodegenerative disorders in the future.

References

1. Suresha, H. S., & Parthasarathy, S. S. (2020, December). Alzheimer Disease Detection Based on Deep Neural Network with Rectified Adam Optimization Technique using MRI Analysis. In 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC) (pp. 1-6). IEEE.
2. Deng, L., & Wang, Y. (2021). Hybrid diffusion tensor imaging feature-based AD classification. Journal of X-Ray Science and Technology, 29(1), 151-169.
3. Khan, A., & Zubair, S. (2020). An improved multi-modal based machine learning approach for the prognosis of Alzheimer's disease. Journal of King Saud University-Computer and Information Sciences.
4. Asim, Y., Raza, B., Malik, A. K., Rathore, S., Hussain, L., & Iftikhar, M. A. (2018). A multi-modal, multi-atlas-based approach for Alzheimer detection via machine learning. International Journal of Imaging Systems and Technology, 28(2), 113-123.
5. Pavitra, B., Singh, D. N., & Sharma, S. K. (2020). Predictive and Interactive IOT Diagnosis System with AI and ML Tools. In International Conference on Advances in Computational Intelligence and Informatics (pp. 9-16). Springer, Singapore.
6. Singh, D. N., Hashmi, M. F., & Sharma, S. K. (2020). Predictive analytics & modeling for modern health care system for cerebral palsy patients. Multimedia Tools and Applications, 79(15), 10285-10307.
7. Roshanzamir, A., Aghajan, H., & Soleymani Baghshah, M. (2021). Transformer-based deep neural network language models for Alzheimer's disease risk assessment from targeted speech. BMC Medical Informatics and Decision Making, 21(1), 1-14.
8. Doan, D. N. T., Ku, B., Choi, J., Oh, M., Kim, K., Cha, W., & Kim, J. U. (2021). Predicting dementia with prefrontal electroencephalography and event-related potential. Frontiers in Aging Neuroscience, 13, 180.
9. Nawaz, A., Anwar, S. M., Liaqat, R., Iqbal, J., Bagci, U., & Majid, M. (2020, November). Deep Convolutional Neural Network based Classification of Alzheimer's Disease using MRI Data. In 2020 IEEE 23rd International Multitopic Conference (INMIC) (pp. 1-6). IEEE.
10. Fiscon, G., Weitschek, E., Felici, G., Bertolazzi, P., De Salvo, S., Bramanti, P., & De Cola, M. C. (2014, December). Alzheimer's disease patients classification through EEG signals processing. In 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM) (pp. 105-112). IEEE.
11. Varatharajan, R., Manogaran, G., Priyan, M. K., & Sundarasekar, R. (2018). Wearable sensor devices for early detection of Alzheimer disease using dynamic time warping algorithm. Cluster Computing, 21(1), 681-690.
12. Subramani, P., & BD, P. (2021). Prediction of muscular paralysis disease based on hybrid feature extraction with machine learning technique for COVID-19 and post-COVID-19 patients. Personal and ubiquitous computing, 1-14.