Early Diagnosis of Alzheimer's Disease using MRI

Kasthuri A S Nair, Hridya Nair S, Riya Ratnakaran, Niharika Padmanabhan
Department of Computer Science
Amrita Vishwa vidyapeetham,
Amritapuri, India

***Abstract*—Deep Neural Networks (DNNs) have revolutionised medical field by lending a variety of computer models that aid in the accurate detection of diseases. The goal of the study is to evaluate how effective the different deep neural network networks are for detecting Alzheimer's disease in humans.**

Alzheimer's disease (AD) is a chronic, ongoing brain disorder that slowly destroys memory and thinking skills and, eventually, affects the ability to carry out simple tasks. Early diagnosis of the disease will help in achieving treatment with maximum efficiency and a good chance of recovery. AD is detected using magnetic resonance imaging (MRI), a medical imaging technique. Deep neural networks play an important role in the diagnosis of Alzheimer's disease by interpreting MRI images. We used two different deep neural network classifiers for classifying a dataset of brain MRIs into four classes: Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented. We attained an accuracy of about 50-60% by implementing a feed-forward neural network model with two hidden layers and an accuracy of 90-100% with a convolutional Neural network model.

Keywords: MRI, Alzheimer's disease, FNN, CNN, SqueezeNet

I. INTRODUCTION

Early detection is essential for the mighty control of Alzheimer's disease. It is a neurologic ailment that causes the brain to shrink (atrophy), which in turn leads to the destruction of brain cells. As memory and cognitive skills deteriorate, the symptoms of dementia become more severe. According to studies, Alzheimer's patients are more likely than the general population to have heart and blood circulation abnormalities, strokes, and diabetes.

If you are diagnosed with this disease early on, you will have a greater chance of receiving treatment and will have a lower risk of developing other linked conditions. You can improve your quality of life and preserve cognitive function by adopting some lifestyle adjustments, such as controlling blood pressure, quitting smoking, exercising, and being mentally and socially active.

Brain abnormalities linked to moderate cognitive impairment can be detected using magnetic resonance imaging (MRI) . It can be used to predict which patients with MCI might also develop Alzheimer's disease. Among numerous deep-learning strategies that have been carried out to assess structural brain changes on magnetic resonance imaging (MRI), the convolutional neural network (CNN) has received a reputation because of its incredible performance in computerised function studying by using several multilayer perceptrons.

This project aims to implement an optimal model that uses brain images as input and classifies those images into 4 different classes. We have increased the accuracy of our model by incorporating different hyperparameter tunings. We began with a simple model and eventually increased its complexity.

II. RELATED WORK

Alzheimer's disease is a brain disorder with no existing treatment for disease cure. So many efforts has been done to come up with strategies for early detection, at the pre-symptomatic stages of this disease. In [1] study, the performance of different transfer learning approaches using convolutional neural networks such as VGG-16, VGG-19, ResNet-18, ResNet-34, ResNet-50 and ResNet-101 in Alzheimer's disease classification. ResNet-101 gave the highest accuracy for each of the classes. By evaluating the dataset using different CNN like GoogleNet, AlexNet, ResNet-18, and VGG-6. GoogleNet had the highest accuracy in [2]. VGG-16 architecture gave an accuracy of 99% for test data in [3].

The performance of the Feed forward network and the SqueezeNet model for the classification of AD is discussed in our paper. We got an accuracy of 52.81% for feedforward and 95.55% for SqueezeNet model.

III. METHODOLOGY

A. FeedForward Neural Network

We commenced with a simple neural network with 1 hidden layer. Then attempted out different activation functions to see which one was the best. Furthermore, the number of hidden layers was suitably increased in order to produce the best feasible feed forward model.

1) Activation function: An Activation Function determines whether a neuron should be activated or not. In our model, we used non linear activation functions such as ReLU, Sigmoid, and Tanh to compute the accuracy for each of the functions. Tanh activation function provided the highest accuracy and hence Tanh activation function was used to implement the final model.

2) Number of Neurons/ Hidden dimension: We started with 100 neurons in the hidden layer. To increase the accuracy with the help of the below equation from the article [4], we got the number of neurons in hidden layers as 382.
No of hidden neurons = $\sqrt[2]{a * b}$,
where a=input neurons and b=output neurons

3) Input Dimension: After applying some random pixel sizes such as 28, 32, and 54, which all provided less accuracy, the average image size of the whole training dataset was utilised

4) Number of Hidden layers: Increasing the number of hidden layers may or may not increase the accuracy. It all depends on the complexity of the problem. We began by experimenting with a single hidden layer. However it resulted in a reduced accuracy. Then we attempted with 3 layers. Finally, when compared to other options, 2 layer model came up to be the best one.

5) Output Dimension: Four classes were there in total. Hence, we chose 4 as our output dimension.

6) Learning rate: The learning rate of a model reveals how rapidly it learns.It can be compared to the step size. Larger weight changes will occur from a high learning rate and hence the performance of the model oscillate over epochs. Simultaneously, a smaller learning rate will take a longer time to converge. As a result an optimal learning rate is a critical component impacting the model's performance. Following our reading of the article, [5] we found a learning rate equation for computing the learning rate, and was used in this model.

$$lr = (neurons\_per\_category * batch\_size)^{-}1$$

7) Dropout: It's a regularisation technique. In this scenario, we randomly destroyed neurons, and their contribution was halted. We started by applying dropout to all of the layers, but it wasn't accurate, so we started removing dropout from each layer. When we applied dropout to the first hidden layer and we got the most accurate model.

B. SqueezeNet Model

A pre-trained model is one that has been trained on a big benchmark dataset to tackle a problem. We have used the Model squeezenet1_0 from the [6] which is a a CNN architecture with fewer parameters as compared to the AlexNet architecture.

1) Architectural Design:

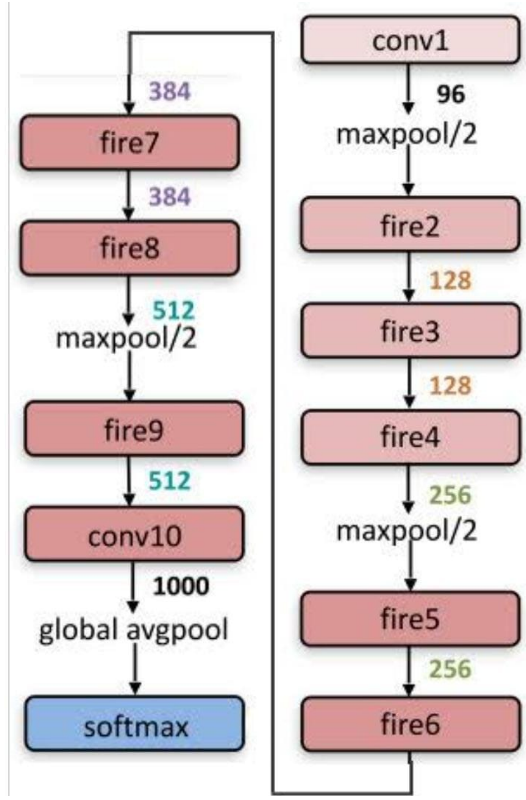

Fig. 1. SqueezeNet Architecture

SqueezeNet has a design that is more compact than the AlexNet model. The model is 3 times faster than AlexNet because it has 50 times fewer parameters..
The three important ideas behind SqueezeNet are:

1.The 3x3 filters were replaced with 1x1 filters to reduce computation by 1/9.

2.The number of inputs for the remaining 3x3 filters is reduced.

3.Down-sample late so that the convolution layers have large activation maps

The SqueezeNet architecture comprises fire modules, which are the building blocks of the model. There are two layers in the fire module: a squeeze layer and an expand layer. The squeeze layer has 1x1 filters and these are fed into the expand layers. The expand layer has a mix of 1x1 and 3x3 convolution filters.

The input image is fed to the first convolutional layer, which is followed by eight fire modules named fire 2-9, and then to a convolutional layer. Max-pooling with stride two is performed after the first convolutional layer, fire4, fire8, and the last convolutional layer.

The other parameters used in this network are Relu as the activation function which is applied between the layers, Dropout with a probability of 0.5, which is used before the last convolutional layer, 0.0003 as the learning rate, and a batch size of 8 for both train and test dataset.

In SqueezeNet, the fully connected layers are absent, and instead uses a 1x1 convolutional layer.

The first two fire module paratmeters in the network were freezed and the rest were used for the learning.

2) Data Augmentation: Data augmentation is the process of artificially increasing the size of a dataset by generating different versions of the existing data. Initially our dataset had 6000 brain images. The accuracy of our model was excessively low when evaluated. When particularly evaluated the moderate demented class and the mild demented class was found to be underrepresented. So the dataset was augmented using different data augmentation techniques. For both train and test data augmentation, the work was performed separately. The augmentation techniques used to enhance the dataset are horizontal flip, vertical flip, brightness, gray-scale and saturation.

a) Flipping: Flipping denotes the rotation of an image around a horizontal or vertical axis. The object's category is usually not impacted by flipping the image left and right. An image can be both horizontally and vertically flipped. We have employed both horizontal and vertical flipping.

b) Colour Change: Changing colours is also another form of image augmentation. We can alter the colour of an image in four ways: brightness, contrast, saturation, and hue. In our dataset, we applied the brightness parameter to the original image using the colourjitter instance to change the brightness.

c) Gray-scale: Image processing techniques that rely on gray-level transformations work directly on pixels. A grayscale image is a simple one in which the only colors are shades of gray. So the images are converted into gray-scale using this augmentation technique.

3) Learning rate: To get the best performing model, we attempted several learning rates. When comparing the different values tried, our model achieved the highest accuracy when initiated with a learning rate of 0.0003.

4) Optimizer: Optimizers are tools for altering parameters like weights in order to minimise losses. The optimizer we used in the network is Adam. It is the best adaptive optimizer. It is a combination of 2 gradient descent techniques which are momentum based and Root Mean Square Propagation (RMSP). It gives much higher performance when compared to other methods.

IV. Experimental Results and Performance Evaluation

A. DataSet

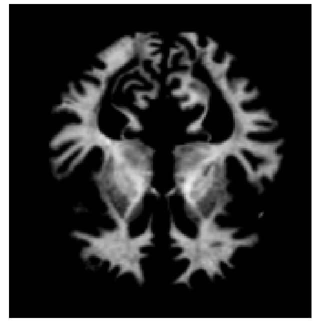

Fig. 2. Mild-Demented

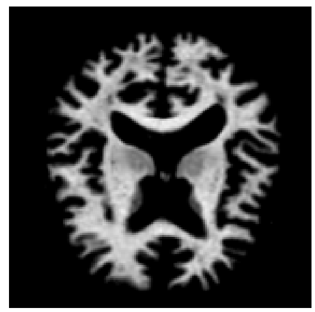

Fig. 3. Moderate-Demented

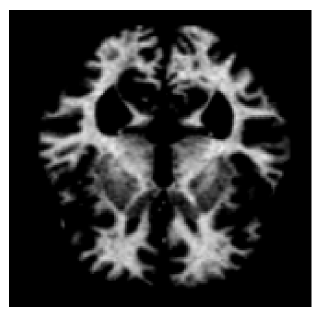

Fig. 4. Very mild-Demented

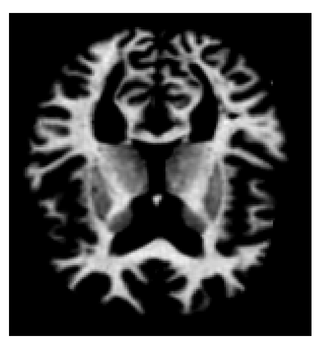

Fig. 5. Non-Demented

The Alzheimer's Dataset (4 classes of photos) from Kaggle was used to train the model. The dataset consists of MRI images of the brain. There are 4 classes in the dataset: Mild Demented, Moderate Demented, Non-Demented, Very Mild Demented.
In order to avoid any sort of imbalance, we concatenated the train and test dataset and randomly shuffled them to produce a new train and test set. Ultimately The training set and test sets were finalized with 8294 images and 2074 images respectively

Experimental Set up

The model building process has several steps as mentioned below:

1. Data loading from Kaggle
2. Performing the required data augmentations.
3. Training the augmented data using a feedforward neural network and with a pretrained CNN architecture(Squeezenet) through transfer learning.
4. Using the test dataset to apply the final model.
5. Evaluating the predicted data using different evaluation metrics

For coming up with the best architecture with good accuracy we performed several hyper parameter tuning.

Regarding the FNN model, the number of neurons, number of hidden layers, learning rate were tuned. The number of neurons is increased to 382 in the hidden layer, the number of hidden layers is also increased to 2 and the learning rate is fixed as 0.0002. We calculated the image size by averaging the sizes of all the images in the dataset.

The graphs of train and test loss were plotted against the number of epochs as shown in Fig. 6. Also, the Train and test accuracy versus epoch graph was plotted as shown in Fig. 7.

For the SqueezeNet model, the learning rate was fixed to be 0.0003 after comparing the model accuracies for different values. We attempted using different optimizers for the model and finalized Adam as it provided better performance. The validation accuracy versus epoch graph was plotted as shown in Fig. 8.

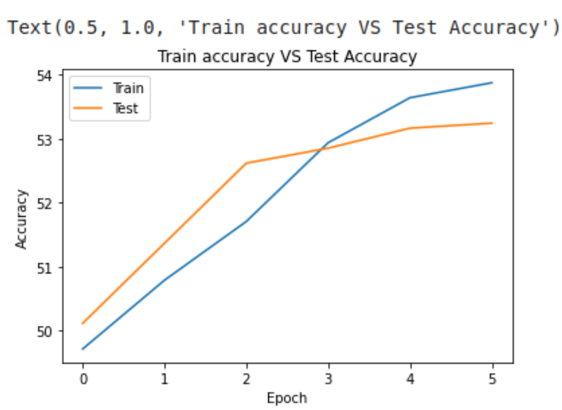

Fig. 6. Train Accuracy and Test Accuracy Vs Epochs

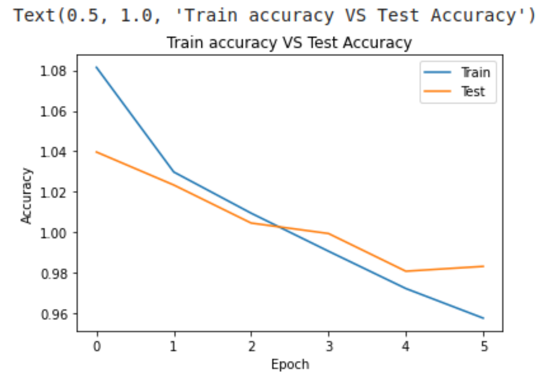

Fig. 7. Train Loss and Test Loss Vs Epochs

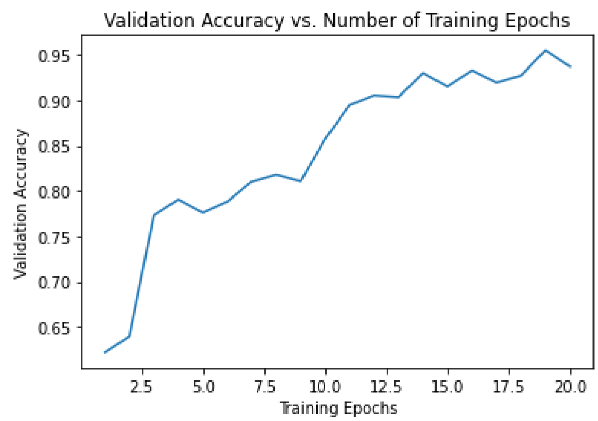

Fig. 8. Validation Accuracy vs. Epochs

B. Results

For the experiment, we used the Kaggle AD dataset with MRI images and classified it into 4 classes : MildDemented, VeryMildDemented, NonDemented, ModerateDemeneted based on the severity of Alzheimer's. In this work, 2 DNN models: FeedForward Model and SqueezeNet Model were implemented. Their performance was evaluated with the parameters accuracy, precision, recall, and F1-score. The number of classifications a model successfully predicts divided by the total number of predictions made is known as model accuracy.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

Precision refers to the number of true positives divided by the total number of positive predictions.

$$Precision = (TP)/(TP + FP)$$

The fraction of all positive samples correctly predicted as positive by the classifier is observed as recall.

$$Recall = (TP)/(TP + FN)$$

F1 Score is the weighted average of Precision and Recall.

$$F1Score = (2TP)/(2TP + FP + FN)$$

TP(True Positive) : The model predicts the positive class correctly as positive

TN(True Negative) : The model predicts the negative class correctly as negative.

FP(False Positive) : The model predicts the positive class incorrectly as positive.

FN(False Negative) : The model predicts the negative class incorrectly as negative.

FeedForward Model and SqueezeNet Model provided an accuracy of 52.81% and 95.55% respectively. The confusion matrix for both the models was plotted and are shown in fig.9 and fig.10. Also, the precision, recall, f1-score and accuracy for each class were calculated for the squeezenet model which is shown in fig.11

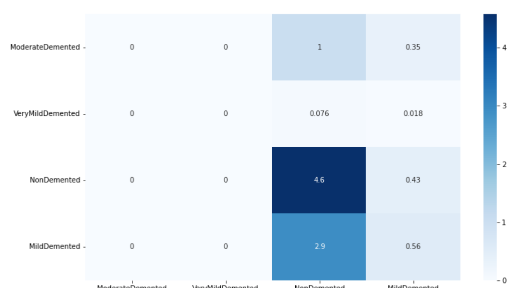

Fig. 9. Confusion Matrix for FNN

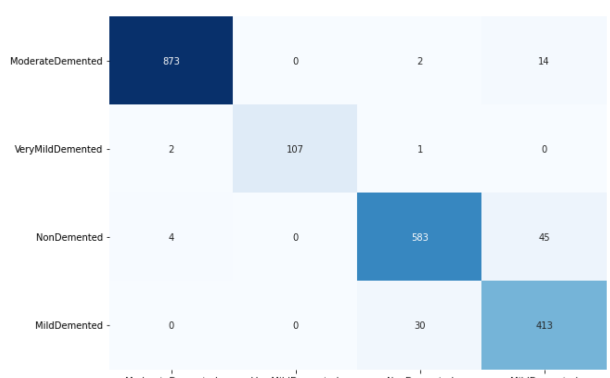

Fig. 10. Confusion Matrix for CNN

Architecture	Classes	Accuracy	Precision	Recall	F1-Score
SqueezeNet	Moderate Demented	99	0.99	0.98	0.99
	Very Mild Demented	100	1	0.97	0.99
	Non Demented	96	0.95	0.92	0.93
	Mild Demented	96	0.88	0.93	0.90

Fig. 11. Evaluation Metrics for CNN

V. Conclusion

Alzheimer's disease detection is difficult, but still necessary for early detection and treatment. In today's scenario, there are different deep learning algorithms which has the capability of providing optimal results. But getting enough data is a major problem. Transfer learning overcomes these issues as it requires only minimal data and time to classify.

In this report, we used Squeezenet architecture for detecting Alzheimer's from MRI scans. Squeezenet was originally trained on the Imagenet dataset, it was repurposed for recognition of Alzheimer's disease through transfer learning the network. To solve the problem of a limited number of MRI scans, augmentation techniques were done on the dataset such as flipping and brightness variation. After augmenting, training was performed on the dataset and was able to reach an accuracy of 95.55% with very few misclassifications.

For future works, more data of patients can help to know whether the accuracy would stay consistently high. The Team would also like to explore more augmentation techniques in future to address the issue of limited data and would try to implement more CNN architectures and perform a comparative study.

References

[1] A deep learning based prediction of alzheimer's disease from magnetic resonance images. [Online]. Available: https://www.researchgate.net/publication/348486602_Deep_learning_based_prediction_of_Alzheimer's_disease_from_magnetic_resonance_images

[2] A comparative study of alzheimer's disease classification using multiple transfer learning models. [Online]. Available: http://www.jmis.org/archive/view_article?pid=jmis-6-4-209

[3] Early detection of alzheimer's disease through machine learning in mri scans. [Online]. Available: http://noiselab.ucsd.edu/ECE228-2020/projects/Report/52Report.pdf

[4] Choosing number of hidden layers and number of hidden neurons in neural networks. [Online]. Available: https://www.linkedin.com/pulse/choosing-number-hidden-layers-neurons-neural-networks-sachdev/

[5] Improving classification accuracy of feedforward neural networks for spiking neuromorphic chips. [Online]. Available: https://www.ijcai.org/proceedings/2017/0274.pdf

[6] Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size. [Online]. Available: https:https://arxiv.org/pdf/1602.07360.pdf