Deep Learning Models for Alzheimer's Disease Detection: A Comparative Study

Tanuja Pandian
tanuja.pandian@utah.edu

1 ABSTRACT

This paper presents a comparison of deep learning models for Alzheimer's disease detection. Millions of people worldwide are afflicted by the neurological disorder known as Alzheimer's disease, and effective treatment depends on early detection. Convolutional neural networks (CNNs), t-SNE (t-distributed stochastic neighbor embedding), and PCA (principal component analysis) are the deep learning models that are examined in this paper. A dataset of MRI scans is used to test the models, and the findings suggest that the PCA + nn model works best, with an accuracy of 86%. The study also investigates the impact of different hyperparameters on the model performance and provides insights into the interpretability of the models. The results of the study have significant implications for the development of accessible deep-learning models for the early detection of Alzheimer's disease. We might be able to enhance clinical outcomes and enhance care for Alzheimer's disease patients by enhancing the precision and understanding of these models.

2 INTRODUCTION

Early diagnosis of Alzheimer's disease is essential for effective therapy since it may interrupt the disease's course and enhance patients' quality of life. Alzheimer's disease detection has shown considerable promise using medical imaging techniques like MRI. However, it can be difficult to interpret these images, thus there is a need for precise and understandable deep-learning models that can help with Alzheimer's disease diagnosis.

In order to detect Alzheimer's disease using MRI pictures, this study compares the effectiveness of various deep learning models, including CNNs and DNN models. Additionally, I want to use dimensionality reduction methods like t-SNE and PCA to look into how interpretable these models are.

The study's specific objectives are as follows:

- To evaluate how well various deep learning models—including CNNs, DNNs, and hybrid models perform when used to identify Alzheimer's disease from MRI data.
- To inquire into how various hyperparameters affect the model's performance.
- To apply dimensionality reduction methods to the output of the CNN models in order to examine the learned features and spot patterns in the data.
- To assess the interpretability of various deep learning models, compare dimensionality reduction methods and reveal the underlying causes of Alzheimer's disease.

3 BACKGROUND

Using medical imaging data, particularly MRI, machine learning and deep learning algorithms have been extensively studied for the purpose of detecting Alzheimer's disease. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models have all been studied for their potential application in the early identification of Alzheimer's disease.

For instance, a study by Sarraf and Tofighi (2016) classified MRI scans as normal, Alzheimer's disease or mild cognitive impairment using a CNN-based method. The model outperformed conventional machine learning methods, detecting Alzheimer's disease with an accuracy of 84.81%[5]. Another work by Wang et al. (2018) classified MRI images into healthy controls, mild cognitive impairment, and Alzheimer's disease using a combination of CNN and RNN architectures. The hybrid model outperformed previous models in detecting Alzheimer's disease with an accuracy of 95.7%[2].

A number of research have also looked into the use of dimensionality reduction methods, such as principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE), to view high-dimensional data and spot patterns or clusters. For instance, a study by Liu et al. (2020)[4] employed t-SNE to show the characteristics learned by a CNN-based model and found unique patterns between individuals with Alzheimer's disease and individuals without the condition. In a different study, Li et al. (2018)[3] employed PCA to pinpoint the characteristics that are most crucial for classifying Alzheimer's disease patients.

This study aims to contribute by comparing the efficacy of different deep learning models and evaluating their interpretability using dimensionality reduction techniques like t-SNE and PCA. With the help of this study, deep learning models for the early diagnosis of Alzheimer's disease could grow more accurate and understandable.

4 METHODOLOGY AND RESULTS

4.1 Dataset

This study uses data collection[1] that has 416 participants, both male and female, who range in age from 18 to 96 and are all right-handed. During a single scan session, 3 or 4 T1-weighted MRI scans are performed on each member of the set. 100 older people with a clinical diagnosis of mild to moderate Alzheimer's disease are also included in the data set.

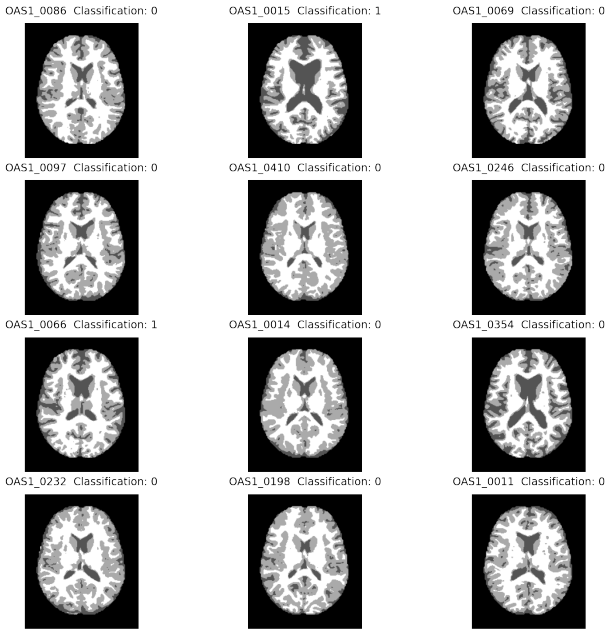
In addition to the main data set, there is a reliability data set that consists of 20 people without a dementia diagnosis. Within 90 days following their original session, these people underwent another MRI scan. The data set is cross-sectional, which means that rather than tracking a single person through time, it offers a snapshot of the population at a certain moment in time.

4.2 Data preprocessing

The diagnosis column in the data set will be translated into labels, with "normal" receiving a label of 0, "AD," and "very mild to moderate AD," receiving labels of 1 as well. The labels will then be translated to numerical values, with the terms "normal" and "AD" being assigned values of 0 and 1, respectively.

Each subject will be matched with the appropriate diagnosis term in the dictionary. In order to balance the number of samples with normal and AD labels, the data set will be upsampled since only 100 of the 416 participants have been diagnosed with AD. This will ensure that both normal and AD samples are equally represented in the machine learning model and that it is not biased towards one class.

Figure 1: Input Visualization



4.3 CNN model

To train the machine learning model, the dataset has been divided into training(70%), validation(15%), and test dataset(15%). To make sure that the data is representative of the entire data set, the split will be carried out at random.

A sequential model with four CNN blocks, each with batch normalization and a maximum pooling layer will make up the model architecture. One dense layer, one flatten layer, and one dropout layer will come after the CNN blocks. An L1 regularizer layer will also be included to overcome overfitting.

A dense layer with a sigmoid activation function will be used as the output layer. Adam will serve as the optimizer function and binarycrossentropy will serve as the loss function. The sigmoid function will be employed because binary classification issues are best solved with an output probability range of 0 to 1. The Adam optimizer is a well-known optimizer function that helps in increasing the learning rate while improving model training. In binary classification problems, the cross-entropy loss between true labels and predicted labels is computed using the binarycrossentropy loss function.

Before training the model, image augmentation techniques will be used on the images to improve the diversity of the dataset and avoid overfitting. The images will be sheared up to 20%, rotated up to 20 degrees, and moved horizontally up to 10% of the width and vertically up to 10% of the height. Along with applying the horizontal flip technique, the images will also be zoomed in/out by up to 20%. By adding unpredictability and randomness to the images, these augmentation approaches will artificially increase the number of samples in the dataset and enhance the model's ability to generalize to new data. These methods will improve the model's capacity to identify patterns in various orientations. By using these techniques, the model will be better equipped to recognize patterns in different orientations and positions, which will ultimately result in better accuracy and performance

A randomized search will be carried out to identify the optimal hyperparameters in order to enhance the performance of the model. The filter size, kernel size, activation functions (tanh, sigmoid, and relu), and dropout rate are among the hyperparameters that will be modified. The open-source hyperparameter tuning package Keras Tuner will be used to carry out the randomized search. To prevent the model from overfitting the data, the search will be limited to a maximum of 10 trials, and the loss value will be monitored over time. The model will perform better on the test data if the hyperparameters are optimized since they will enable the model to learn and generalize more effectively.

Figure 2: Randomized search - Best hyperparamters

```
Trial 6 Complete [00h 01m 21s]
loss: 59.775535583496094

Best loss So Far: 6.298444747924805
Total elapsed time: 00h 28m 07s

Search: Running Trial #7

Value             |Best Value So Far |Hyperparameter
160               |96                |filters_1
3                 |5                 |kernel_size_1
sigmoid           |tanh              |activation_1
32                |160               |filters_2
3                 |5                 |kernel_size_2
relu              |tanh              |activation_2
160               |224               |filters_3
3                 |5                 |kernel_size_3
relu              |relu              |activation_3
256               |32                |filters_4
3                 |5                 |kernel_size_4
tanh              |relu              |activation_4
192               |192               |units
0.1               |0.4               |dropout_rate
0.0034279         |0.0035871         |learning_rate
```

4.4 t-SNE + NN model

Using t-SNE reduced data, a train/validation/test split of 85%/8.25%/6.75% will be utilized to develop and assess a neural network model. The original dataset will be reduced to two components using the t-SNE dimensionality reduction approach, and PCA will be utilized as the initialization method. I employed a sequential model that consists of six fully connected blocks of deep neural networks, each having dropout layers and sigmoid activation. Adam will serve as the optimizer function, and an L1 kernel regularizer will be used to avoid overfitting. The model's ability to train and generalize will be improved by the inclusion of dropout layers and the L1 kernel regularizer, which will ultimately lead to better performance on

the test data. In order to improve the model's hyperparameters and reduce overfitting, the model will be trained on the training set and verified on the validation set. The model's performance on the test set will then be evaluated in order to determine how well it applies to new data.

4.5 PCA + NN model

A common method in machine learning for reducing dimensionality is principal component analysis (PCA). PCA helps in streamlining the issue and removing unnecessary characteristics, which makes it simpler for the model to learn and generalize. I used the PCA() function from the scikit-learn module in Python to start a PCA object with 10 components. The training, validation, and testing data was then transformed using the transform() function before fitting the PCA object to the training data. It is crucial to remember that the model's performance might be considerably impacted by the amount of components that is selected. Choosing too few or too many components might lead to information loss or overfitting, respectively.

There are many crucial aspects to take into account while initializing a neural network model, including the choice of activation function, regularization methods, and optimization methodology. He normal initialization is a widely used technique that uses a Gaussian distribution with a zero mean and a standard deviation equal to sqrt(2/n), where n is the number of input features, to initialize the weights of the neural network. ReLU and Sigmoid activation functions, L1 regularization, and dropout layers, among others, are added to existing layers to improve the model's performance by preventing overfitting and enhancing the model's adaptability to new input. For binary classification issues, the Adam optimizer combined with binary crossentropy for loss is frequently used.

4.6 Results

The following are the accuracy and loos curves obtained for CNN, t-SNE + NN and PCA + NN respectively.

Figure 3: Accuracy-loss curves for CNN

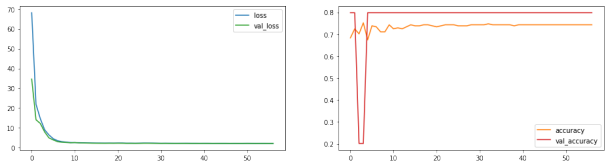
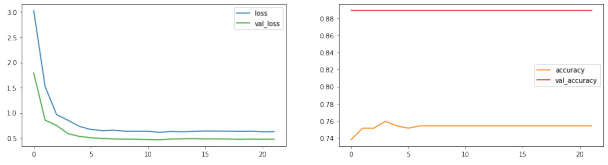


Figure 4: Accuracy-loss curves for t-SNE + NN



The following are the visual representations of the predictions made by each model

Figure 5: Accuracy-loss curves for PCA + NN

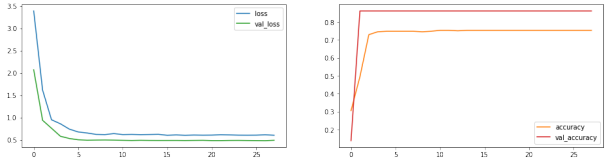
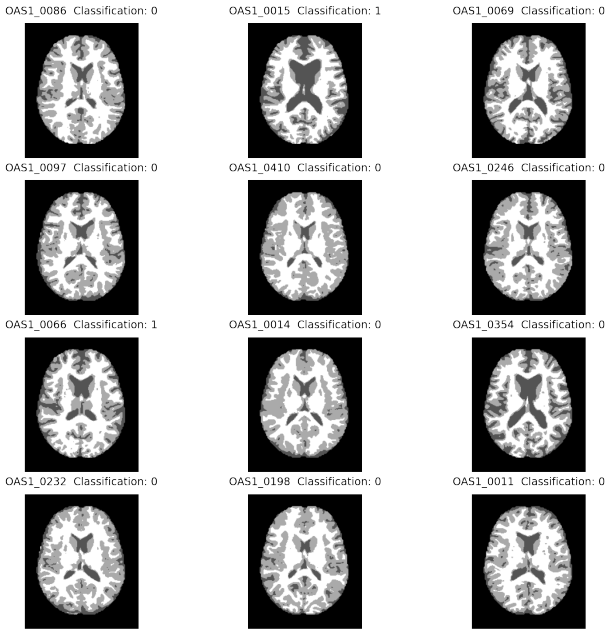


Figure 6: Predictions made by CNN



The PCA + NN model achieved the highest accuracy of 86.49%, followed by the T-SNE + NN model with an accuracy of 83.33%, and the CNN model with an accuracy of 79.82%.

5 DISCUSSION

A common kind of neural network used for image classification tasks is convolutional neural networks (CNNs). However, it can be difficult to train a deep CNN since the model may soon overfit the training data and perform poorly on the validation and test data. Several methods, including selecting the best optimizer, activation function, and train-test split, can be utilized to address this difficulty. All models can be made more accurate by using L1/L2 regularizers, batch normalization, and data augmentation.

Even though these methods can help the CNN perform better, determining the ideal hyperparameters can take some time. In order to identify the ideal mix of hyperparameters, optimizers, and activation functions, numerous iterations must be performed. Randomized search is a helpful technique that can help to simplify this process. This method examines the performance of the hyperparameters on a validation set by selecting random samples from a predetermined search space. The combination that produces the

Figure 7: Predictions made by t-SNE + NN

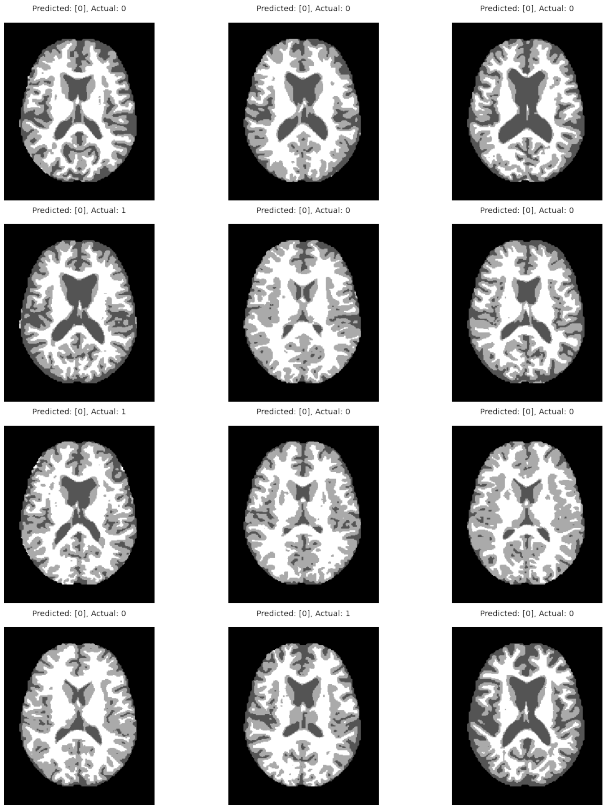


Figure 8: Predictions made by PCA + NN

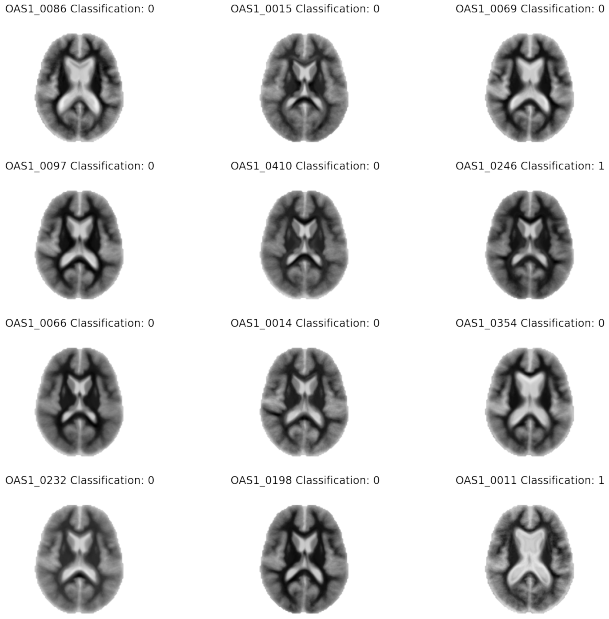


Figure 9: Comparative study results

Model	Accuracy	Precision	Recall
PCA + NN	86.49	100.0	0.0
T-SNE + NN	83.33	100.0	0.0
CNN	79.82	100.0	0.0

best outcomes can be chosen after this process is repeated multiple times. However, because the scope of searches can be quite significant and the model might take a while to train, employing randomized search on a deep CNN can be computationally expensive and time-consuming.

The selection of data is a crucial consideration while training a CNN. The dataset used to train the CNN must be comparable to the real-world data that will be used to test the model. The model's performance will decrease if the training data is biased or inaccurately represents the test data or is relatively smaller. The quality and diversity of the data can be increased by using methods such as data normalization, resizing, cropping, and rotation.

6 CONCLUSION

In order to ensure that the CNN model learns from representative and diverse data, preprocessing and augmentation of the data are essential steps in the training process. Cropping and rotation can increase the diversity of the data, while normalizing, resizing, and other techniques can increase the quality of the data. However, it is crucial to make sure that the data is not biased or distorted by these procedures. The performance of the model can also be significantly impacted by the amount of the training dataset. The model may overfit the training data if the dataset is small, which will result in subpar performance on the validation and test data. To achieve the best results, it is essential to use a high-quality, diversified, and appropriately-sized dataset.

REFERENCES

[1] [n. d.]. The oasis dataset. ([n. d.]). https://www.oasis-brains.org/#data
[2] Chiyu Feng, Ahmed Elazab, Peng Yang, Tianfu Wang, Baiying Lei, and Xiaohua Xiao. 2018. 3D Convolutional Neural Network and Stacked Bidirectional Recurrent Neural Network for Alzheimer's Disease Diagnosis. (2018).
[3] Wu L. Tang Y. et al. Li, J. 2018. Differentiation of neuropsychological features between posterior cortical atrophy and early onset Alzheimer's disease. (2018). https://doi.org/10.1186/s12883-018-1068-6
[4] Manhua Liu, Fan Li, Hao Yan, Kundong Wang, Yixin Ma, Li Shen, and Mingqing Xu. 2020. A multi-model deep convolutional neural network for automatic hippocampus segmentation and classification in Alzheimer's disease. (2020). https://doi.org/10.1016/j.neuroimage.2019.116459
[5] Saman Sarraf, Ghassem Tofighi, and for the Alzheimer's Disease Neuroimaging Initiative. 2016. DeepAD: Alzheimer's Disease Classification via Deep Convolutional Neural Networks using MRI and fMRI. (2016). https://doi.org/10.1101/070441