A Comparative Study of Deep Neural Networks for Alzheimer's Disease Stage Classification Using MRI

Onkar Dangi

1 Introduction

Alzheimer's disease (AD) is a neuro degenerative, progressive illness with a largely senile group of elderly individuals with subsequent impairment of memory, intellect, and ultimately loss of ability to carry out daily functions. The aging population in the world leads to an increasing number of cases of Alzheimer's disease and therefore early and accurate diagnosis is an issue of significance in clinical practice.

Early identification is worthwhile as it can theoretically allow families to plan in advance, financially, legally and medicinally. Early identification can allow interventions to become available that have better control of symptoms and better quality of life overall, although there is still no cure for Alzheimer's disease. I am using MRI images for early detection as it can reveal brain abnormalities that are the potential markers of the disease. These abnormalities can help us predict who might be at risk of developing Alzheimer's before symptoms occur. Now, machine learning combined with this MRI data can help us achieve high accuracy in predicting Alzheimer's disease as I will further explore in this report.

In recent years, convolution neural network (CNNs), transfer learning and pretrained models like ResNet, DenseNet and many other similar models have become the state of the art approaches for medical image classification. These methods have proven to be very efficient in tasks such as disease detection, classification and segmentation.

In this report, I aim to explore the effectiveness of deep learning models for the detection and classification of Alzheimer's disease using the MRI data. I am focus on the OASIS dataset which provides us with labeled brain images across four stages of it, from non demented to moderate dementia. By incrementally comparing multiple neural network models, including baseline architecture and advanced models like ResNet and DenseNet, I intend to explore and improve the accuracy and diagnostic capabilities as Itransition to more sophisticated models. Our goal is to show how deep learning can significantly enhance early detection of Alzheimer's disease diagnoses and treatment.

2 Methods/Case Study

For this project, I used the publicly available **OASIS MRI dataset** (https://sites.wustl.edu/oasisbrains), which contains around **80,000 structural brain MRI slices** from **461 patients**. Each patient is labeled using the **Clinical Dementia Rating (CDR)**, allowing us to categorize them into

four classes: non-demented, very mild demented, mild demented, and demented. These labels help track the progression of Alzheimer's disease in different stages.

The dataset originally came in `.img` and `.hdr` file formats. These were already converted into **NIfTI format (.nii)** using **FSL**, and then further processed into **JPEG files** for compatibility with deep learning pipelines. Magnetic resonance volumes were cut along the **z-axis** into 256 slices per patient. For the neural network training, 2D images were used as input. The brain images were sliced along the z-axis into 256 pieces, and slices ranging from 100 to 160 were selected from each patient. This approach resulted in a comprehensive dataset for analysis.

Each JPEG slice was resized to a consistent resolution, normalized to [0, 1] range, and used as input to our models. The final data set was divided into training (70%), validation (15%) and test (15%) splits, maintaining class balance as much as possible. The total size of the data set after pre-processing was approximately **1.3 GB**.

This well-curated and preprocessed data set allowed us to train and compare various deep learning models effectively for the task of identifying Alzheimer's stages.

3 Training

To evaluate the effectiveness of various models in detecting stages of Alzheimer's disease from MRI images, I designed a series of experiments using progressively more powerful neural architectures. The models include a simple Multi-Layer Perceptron (MLP), a custom Convolutional Neural Network (CNN), and pretrained models like ResNet18 and DenseNet121.

3.1 Training Setup

Training was performed locally first, then on Google Colab, due to the need for more control over resources. Given the computational overhead, I limited the number of training epochs to 10 for all models. The dataset was split into 70% for training, 15% for validation, and 15% for testing.

- **Input Format**: 2D MRI slices resized to 224x224 pixels
- **Batch Size**: 32
- **Epochs**: 10
- **Optimizer**: Adam (learning rate = 0.001)
- **Loss Function**: CrossEntropyLoss
- **Evaluation Metrics**: Accuracy, Precision, Recall, and F1-Score

Due to computational constraints, the models were trained for **10 epochs**.

3.2 Model 1: Simple MLP (Baseline)

I began with a baseline MLP model that takes flattened input images and passes them through two fully connected hidden layers with ReLU activations and dropout regularization. This model provides a reference point to measure improvements from more complex architectures.

- Training Accuracy: Improved from 42.94% to 50.00%
- Validation Accuracy: Constant at 50.00%
- F1-Score: Remained around 0.16 across epochs

```
--------------------------------------------------------------------
Layer (type:depth-idx)                   Output Shape
====================================================================
SimpleMLP                                [1, 4]
├─Sequential: 1-1                        [1, 4]
│    └─Flatten: 2-1                      [1, 150528]
│    └─Linear: 2-2                       [1, 512]
│    └─ReLU: 2-3                         [1, 512]
│    └─Dropout: 2-4                      [1, 512]
│    └─Linear: 2-5                       [1, 128]
│    └─ReLU: 2-6                         [1, 128]
│    └─Dropout: 2-7                      [1, 128]
│    └─Linear: 2-8                       [1, 4]
--------------------------------------------------------------------
```

Figure 1: SimpleMLP Architecture

3.3 Model 2: Custom CNN

To leverage spatial features in brain scans, I implemented a custom CNN model composed of convolutional and pooling layers, followed by dense layers. This model outperformed the MLP by learning local patterns within the images.

- Training Accuracy: Improved from 54.29% to 95.97%
- Validation Accuracy: Increased from 64.19% to 94.38%
- F1-Score: Grew from 0.53 to 0.96

```
Layer (type:depth-idx)                   Output Shape
====================================================================
AlzheimerCNN                             [1, 4]
├─Sequential: 1-1                        [1, 128, 28, 28]
│    └─Conv2d: 2-1                       [1, 32, 224, 224]
│    └─ReLU: 2-2                         [1, 32, 224, 224]
│    └─MaxPool2d: 2-3                    [1, 32, 112, 112]
│    └─Conv2d: 2-4                       [1, 64, 112, 112]
│    └─ReLU: 2-5                         [1, 64, 112, 112]
│    └─MaxPool2d: 2-6                    [1, 64, 56, 56]
│    └─Conv2d: 2-7                       [1, 128, 56, 56]
│    └─ReLU: 2-8                         [1, 128, 56, 56]
│    └─MaxPool2d: 2-9                    [1, 128, 28, 28]
├─Sequential: 1-2                        [1, 4]
│    └─Flatten: 2-10                     [1, 100352]
│    └─Linear: 2-11                      [1, 128]
│    └─ReLU: 2-12                        [1, 128]
│    └─Dropout: 2-13                     [1, 128]
│    └─Linear: 2-14                      [1, 4]
```

Figure 2: CustomCNN

3.4 Model 3: ResNet18 (Transfer Learning)

I used a pretrained ResNet18 architecture with its final layers modified to suit the four-class classification task. Transfer learning improved training stability and generalization despite limited training time.

- Training Accuracy: Reached 98.18%
- Validation Accuracy: Peaked at 94.57%
- F1-Score: Achieved up to 0.95

```
Layer (type:depth-idx)                             Output Shape
====================================================================
AlzheimerResNet18                                  [1, 4]
├─ResNet: 1-1                                      [1, 4]
│    └─Conv2d: 2-1                                 [1, 64, 112, 112]
│    └─BatchNorm2d: 2-2                            [1, 64, 112, 112]
│    └─ReLU: 2-3                                   [1, 64, 112, 112]
│    └─MaxPool2d: 2-4                              [1, 64, 56, 56]
│    └─Sequential: 2-5                             [1, 64, 56, 56]
│    │    └─BasicBlock: 3-1                        [1, 64, 56, 56]
│    │    └─BasicBlock: 3-2                        [1, 64, 56, 56]
│    └─Sequential: 2-6                             [1, 128, 28, 28]
│    │    └─BasicBlock: 3-3                        [1, 128, 28, 28]
│    │    └─BasicBlock: 3-4                        [1, 128, 28, 28]
│    └─Sequential: 2-7                             [1, 256, 14, 14]
│    │    └─BasicBlock: 3-5                        [1, 256, 14, 14]
│    │    └─BasicBlock: 3-6                        [1, 256, 14, 14]
│    └─Sequential: 2-8                             [1, 512, 7, 7]
│    │    └─BasicBlock: 3-7                        [1, 512, 7, 7]
│    │    └─BasicBlock: 3-8                        [1, 512, 7, 7]
│    └─AdaptiveAvgPool2d: 2-9                      [1, 512, 1, 1]
│    └─Linear: 2-10                                [1, 4]
```

Figure 3: ResNet18

3.5 Model 4: DenseNet121 (Transfer Learning)

DenseNet121 was also evaluated for its feature reuse and improved gradient flow. Its dense connections are known to perform well on medical image classification tasks.

- Training Accuracy: Exceeded 95% by epoch 6
- Validation Accuracy: Fluctuated early, then reached up to 92.19% by epoch 5
- F1-Score: Peaked at 0.93

```
Layer (type:depth-idx)                        Output Shape
===================================================================
AlzheimerDenseNet121                          [1, 4]
├─DenseNet: 1-1                               [1, 4]
│    └─Sequential: 2-1                        [1, 1024, 7, 7]
│    │    └─Conv2d: 3-1                       [1, 64, 112, 112]
│    │    └─BatchNorm2d: 3-2                  [1, 64, 112, 112]
│    │    └─ReLU: 3-3                         [1, 64, 112, 112]
│    │    └─MaxPool2d: 3-4                    [1, 64, 56, 56]
│    │    └─_DenseBlock: 3-5                  [1, 256, 56, 56]
│    │    └─_Transition: 3-6                  [1, 128, 28, 28]
│    │    └─_DenseBlock: 3-7                  [1, 512, 28, 28]
│    │    └─_Transition: 3-8                  [1, 256, 14, 14]
│    │    └─_DenseBlock: 3-9                  [1, 1024, 14, 14]
│    │    └─_Transition: 3-10                 [1, 512, 7, 7]
│    │    └─_DenseBlock: 3-11                 [1, 1024, 7, 7]
│    │    └─BatchNorm2d: 3-12                 [1, 1024, 7, 7]
│    └─Linear: 2-2                            [1, 4]
```

Figure 4: DenseNet121

4 Results and Discussion

I evaluated the models on validation metrics including precision, recall, and F1-score.

SimpleMLP

The SimpleMLP model failed to outperform a random classifier, with all key validation metrics stuck at low values:

- Validation Accuracy: 50.00%
- F1-Score: ∼0.17

AlzheimerCNN

AlzheimerCNN achieved robust results with consistently improving validation metrics:

- Validation Accuracy (Epoch 10): 94.38%
- Test Accuracy: 95.97%
- Test F1-Score: 0.97

ResNet18

ResNet18 delivered competitive performance, slightly exceeding AlzheimerCNN in some epochs:

- Validation Accuracy (Epoch 10): 94.57%
- Test Accuracy: 94.43%
- Test F1-Score: 0.95

DenseNet121

Despite an unstable start, DenseNet121 eventually delivered strong results:

- Peak Validation Accuracy: 92.19% (epoch 5)
- Peak F1-Score: 0.9317

Model Comparison

The table/figure below summarizes and compares the performance of all the models evaluated in this study. As expected, pretrained models significantly outperform the baseline MLP and the custom CNN.

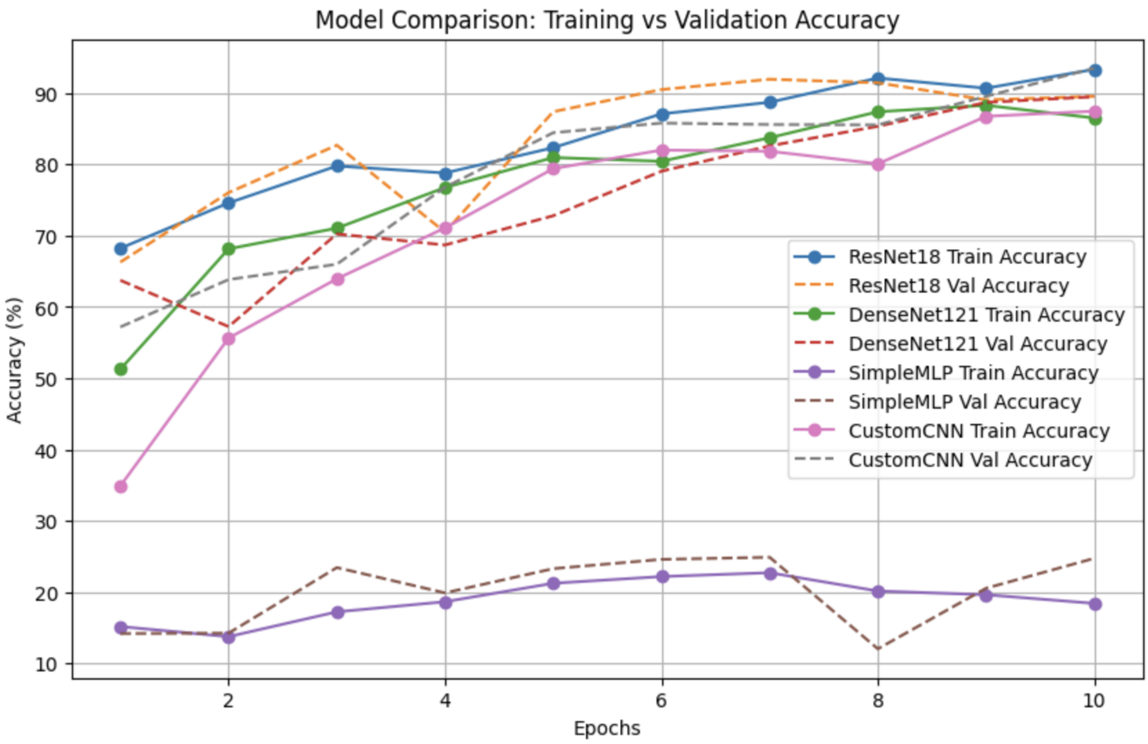

Figure 5: Model Comparison

Model	Accuracy	Precision	Recall	F1-Score
Simple MLP	50.00%	0.12	0.25	0.17
Custom CNN	95.96%	0.97	0.96	0.96
ResNet18	94.43%	0.93	0.95	0.94
DenseNet121	87.73%	0.91	0.86	0.88

Table 1: Performance Comparison Across Models

Summary: Among the models, AlzheimerCNN and ResNet18 showed the best trade-off between stability and performance, while DenseNet121 also achieved high scores but required more careful tuning. SimpleMLP served as a baseline and showed the need for deep architectures in this task.

5 Conclusion

In this research, I compared the performance of four models — Simple MLP, a self-designed CNN, ResNet18, and DenseNet121—for detecting Alzheimer's disease from brain MRI scans. The results demonstrate a clear improvement in accuracy and overall performance as I progressed from simpler architectures to deeper and more complex networks.

Simple MLP was utilized as a baseline and worked poorly with weak capability, most likely due to its ineffectiveness in maintaining spatial features inherent in image data. Custom CNN provided a significant performance boost with the outcome of demonstrating that a moderately deep network carefully tailored to the data can be highly effective. ResNet18 worked less than the custom CNN. Strangely, DenseNet121, though a more complex architecture, worked slightly inferior to ResNet18 in our model, presumably due to overfitting or flaws in the specific dataset used.

CustomCNN overall worked the best as compared to others in our trials, achieving a golden balance between depth, efficiency during training, and generalizability. Our results suggest that for comparable small-data medical image challenges, customCNN is an adequate starting point for robust and robust prediction.