Potential use of Convolutional Neural Networks in Alzheimer's Detection

Arthur Wang

ABSTRACT

Worldwide, Alzheimer's disease (AD) affects one in nine individuals over the age of 65. AD is a progressive disease that kills neurons in parts of the brain related to mental function, causing memory loss and the irreversible decline of communication or problem solving abilities. Early diagnosis is important for those with AD, as it allows them to make informed decisions regarding treatment options and finances prior to significant mental decay. Although there doesn't exist a cure, different forms of treatment have been shown to slow disease progression, most effective at early stages (Bush 2). The cause of the disease is not yet known, but early symptoms, such as protein buildup and shrinkage in the brain, occur as much as fifteen years before signs of cognitive decline (Scheltens 5). By combining machine learning technology with brain imaging, it should be possible to quickly and effectively detect AD early in its development. The goal of this work is to train multiple convolutional neural networks (CNNs) off of MRI brain scans to detect the severity of AD, and then see if they can outcompete alternative technologies.

I. INTRODUCTION

The use of machine learning has become especially popular in recent years, as a result of advancements made in deep learning. Deep learning specifically references machine learning models with large amounts of "hidden layers", allowing them to learn from and identify complex patterns in data. CNNs are a specific type of deep learning model, and specialize in processing grid-like data. As such, they are the optimal choice when doing image recognition. Taking images as inputs, CNNs are able to break them down into arrays of pixel values. Then, by passing this array through multiple convolution and pooling layers, these models are able to

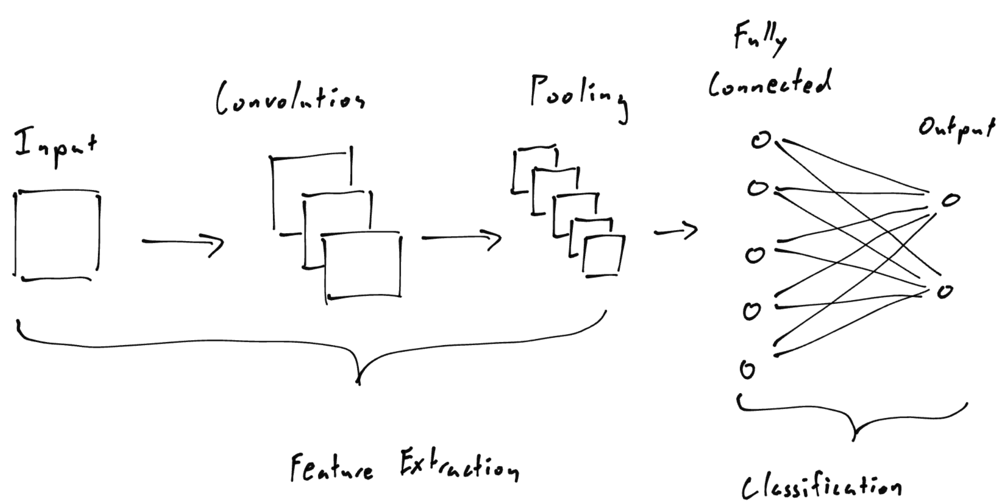


Figure 1: Simplified diagram of a CNN model

extract specific features that it can then use to make decisions, as shown in Figure 1. This ability to identify features within an image is what makes CNNs such an appealing option for disease diagnoses. So far, they have demonstrated a prominent capability in detecting AD, but there is still ongoing research to determine which specific model architectures are most efficient and consistent (Farooq 1).

Throughout this paper, multiple CNN models will be trained and have their performances compared to another common type of classifier, decision trees. Both will be trained using the same data, a set of 36,000 MRI brain scans (Uraninjo). These images have been augmented from a set of 6,000 originals, for the sake of providing sufficient amounts of training data. The

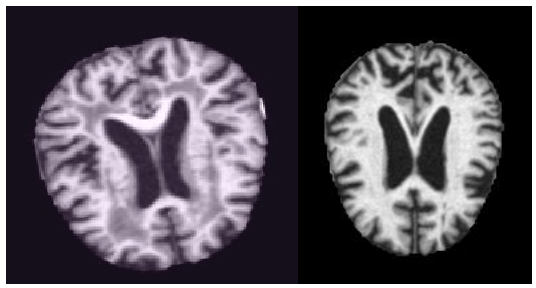


Figure 2: Augmented (left) and original (right)

original and augmented sets will be kept separate from each other; the augmented images will be used for training, and the originals will be used as a test set. It is worth noting that none of the 6,000 originals are found in the set of 36,000 augmented images, ensuring an accurate assessment of the model's abilities. The results of the test will be displayed in the form of a confusion matrix, as it provides a better insight into model performance than raw accuracy does.

II. DECISION TREES

Decision trees are another type of classifier, and are often valued for their simplicity. They take an input, known as a "root", which is connected to other nodes through "branches", as

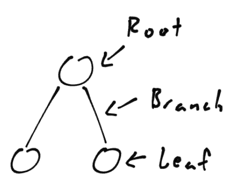


Figure 3: Simplest possible decision tree

shown in Figure 3. At each node, a binary decision is made that leads to a node further down the tree. The nodes at the bottom of the decision tree are the outputs, called "leaves". As the decision tree trains, it can grow additional nodes, branches, and leaves, improving its performance and becoming more complex. Compared to CNNs, this training happens relatively quickly and requires very little computing power. However, this comes with its own limitations. Due to their simplicity, these models are not capable of pooling image data; any complex images need to be downscaled such that they do not overwhelm the tree. This is typically done by lowering the resolution of input images, but doing so inevitably decreases data complexity which results in lower performance. The decision trees in this paper will be using images downscaled from 224 pixels x 224 pixels to 50 x 50. To counter some of this complexity loss,

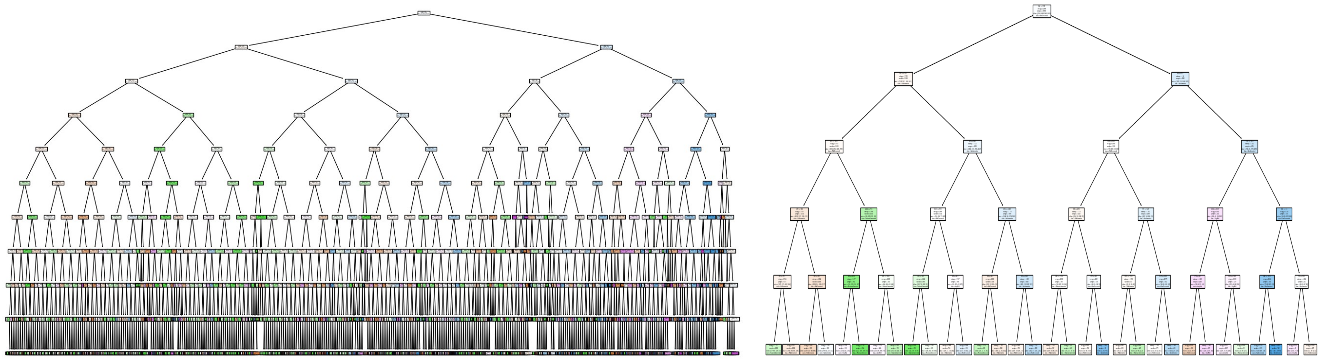


Figure 4: 10-layer tree (left) next to a 5-layer tree (right)

additional layers can be added onto the tree. As shown in Figure 4 and Figure 9, the ten layer tree is able to make exponentially more decisions than its five layer counterpart, resulting in an improved test performance of 61.48% from 42.78%. Even so, there is a limit to how many layers can be added, as decision trees have a tendency to overfit the training data. Beyond a point, the tree is able to create a leaf for every single instance within the training set. A common way to combat this is through the Random Forest algorithm, which combines a large number of individual decision trees in such a way that eliminates unnecessary leaves, but this added level of complexity greatly increases the time it takes for the model to train and produce predictions.

Given the performance of the two trees in Figure 4, it is unlikely that such simple models would be dependable in a medical setting. However, one of the most useful features of a decision tree is that it is decipherable; after training, it is possible to see the decisions made at each of the nodes, which may be useful in the identification of AD features.

III. CNNS

CNNs, similar to other machine learning models, consist of many nodes organized into layers. Each node receives a series of weighted inputs from the previous layer, applies a non-linear transformation onto them, and passes the transformed data onto the next layer, as

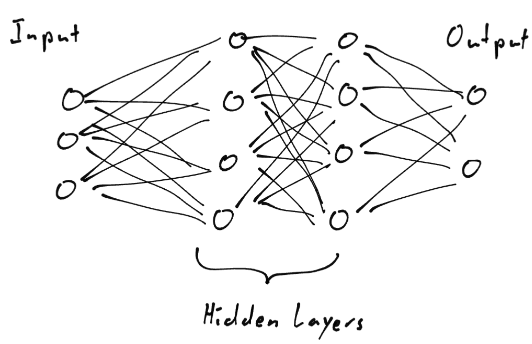


Figure 5: Simple machine learning diagram

shown in Figure 5. The method to train the model is to change the weights between layers. When the model is fed training data and an output is produced, the actual output is compared to the expected output through the use of a loss function. The higher the loss value, the more incorrect the model's output is. After the loss is calculated, calculus is used in a process known as gradient descent to determine new weights that will decrease the loss function. Once the loss value approaches a minimum, the model's accuracy should have improved.

In CNNs, the weights between layers are matrices called kernels. In both the convolution and pooling layers mentioned earlier (both are usually counted as 1 hidden layer), kernels are imperative for how the model functions. In a convolutional layer, kernels act as filters that identify patterns like edges or textures within the input data. Subsequently, a pooling layer is applied, which reduces the size of the data by blurring or removing irrelevant features. Due to a combination of these two layers, CNNs are able to process large amounts of image data in a way that decision trees cannot.

To evaluate whether these features can aid in AD detection, a simple CNN architecture consisting of seven layers (5 convolution/pooling layers & 2 dense layers) and 444,996

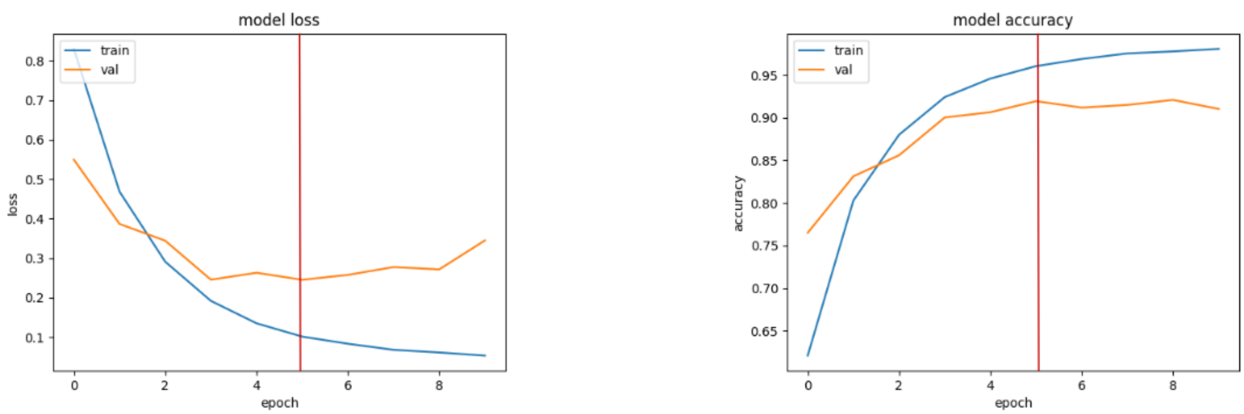


Figure 6: CNN training progress

parameters will be trained (aashidutt3). As seen in Figure 6, the validation loss does not decrease after running through the training set five times, known as an epoch, marked by the red line. As the model continues to train, the validation loss actually starts to increase. This is an indication that the model is overfitting, and has started to find patterns in the training set that should not exist. This is because the finite training set is not perfectly representative of all the MRI images. In instances like this, it is best to stop training and restore the weights of the model to their state at epoch number five, so our final model does not overfit. For future models, early stopping will automate this process. Looking at Figure 9, this model's accuracy was significantly higher than either of the decision trees. In the test set, it had an accuracy of 96.53% whereas the ten-layer decision tree only got 61.48%. This is rather strong evidence that CNNs are stronger for this specific task, so future models in this report will only focus on improving the CNN's performance.

There are two primary ways to manipulate a neural network's complexity, which is by changing the number of layers or parameters in the model architecture. To try and improve the previous model, the architecture has been revised from seven layers with 444,996 parameters to six layers (4 convolution/pooling layers & 2 dense layers) with 2,385,444 parameters. This sudden increase in parameters is largely due to the final layer having an increased number of neurons and larger convolutional filter sizes. Increased parameters usually indicates a more complex model, so an increase in model performance is expected. This model was trained in the same way as the first one, but this time the performance was significantly better. Just by increasing the amount of parameters, this second model was able to obtain a testing accuracy of 99.22%, as shown in Figure 9.

Through these simulations, it seems clear that the CNNs are performing significantly better than the decision trees. However, there are still ways to potentially improve this CNN model.

IV. TRANSFER LEARNING

Another common method in designing CNN architecture is through transfer learning. Transfer learning involves taking a previously-trained model, adding additional layers, then retraining it for a related task. This process is extremely useful when designing CNN models, as it helps with one of their greatest weaknesses - data collection. Training deep learning models requires large amounts of labeled data, which is often difficult to obtain. By importing a previously-trained model with pre-set parameters, the CNN already knows how to identify certain patterns. The hope is that these pre-set parameters place the model relatively close to a local minimum within the loss function, reducing the amount of gradient descent needed to be performed. This not only decreases the amount of training data needed, but also speeds up the training process and often results in a better model performance. (Shin 3)

To evaluate whether transfer learning can help in detecting AD, two separate transfer learning models will be evaluated. Both will use the Inception model, with two additional dense layers added. The Inception model is a 48 layer deep CNN, and has been trained to classify over 1,000,000 images ranging from household objects to animals (Du 2). One of the transfer learning models will freeze the weights of the Inception model, and the other will not. Freezing the weights prevents the pre-trained model from changing, and is typically used to fine-tune the model on a new task, but this is not always preferred. A downside to using unfrozen weights, however, is that it drastically increases training time.

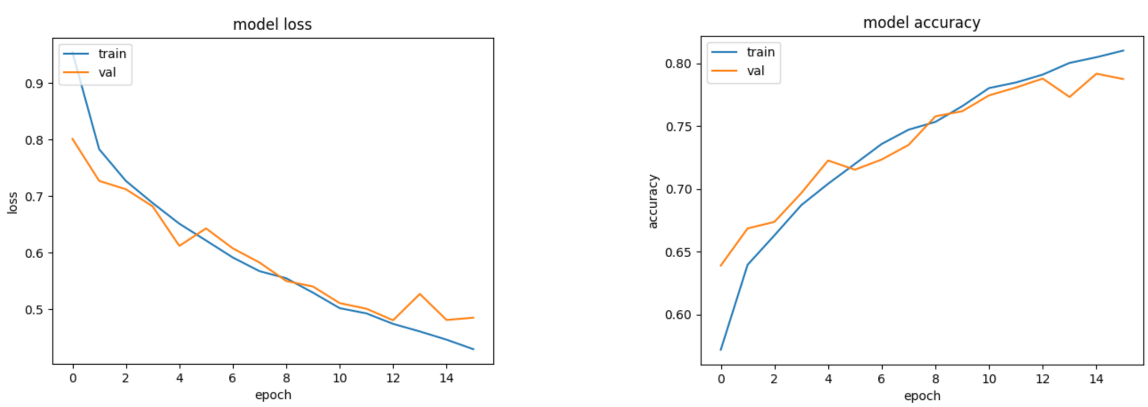


Figure 7: Transfer learning (frozen) training progress

Figure 7 demonstrates the training progress of the frozen weights model. Quite unexpectedly, both the training and validation loss decrease at a slower rate than the previous CNN models, and the training finishes at a later epoch. Looking at the model's confusion matrix in Figure 9, the testing accuracy of 74.89% is also much lower than one might expect, especially with the model's 22,853,924 parameters. A possible explanation for this is that the brain MRIs used by this model are drastically different from the images the Inception model was trained on.

As such, many of the features isolated by the Inception kernels are not applicable to the task of AD identification, and any improvements that can be gained by training are limited due to the Inception kernels being frozen.

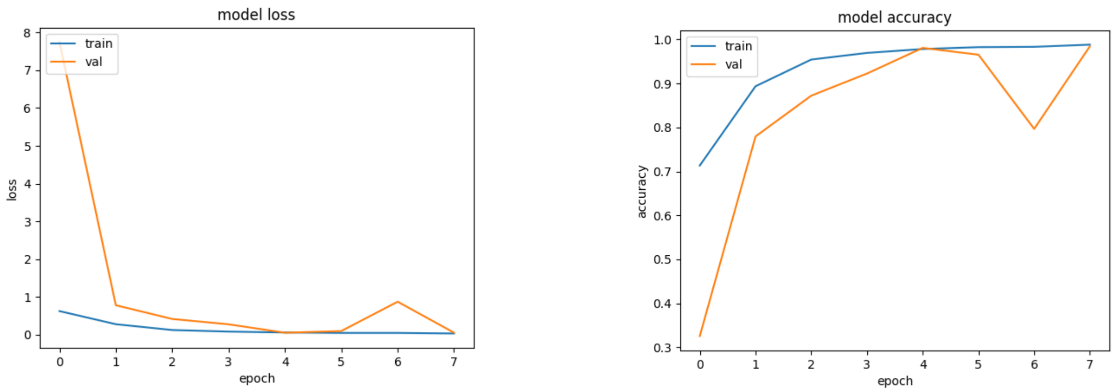


Figure 8: Transfer learning (unfrozen) training progress

Looking instead at the training progress for the unfrozen weights model, seen in Figure 8, it more closely resembles what one would expect of transfer learning. The model loss converges rather quickly, and the validation loss stops decreasing after epoch number four. However, when looking at its confusion matrix in Figure 9, this model still only has an accuracy of 98.72%, performing worse than the 99.22% model from earlier, despite having almost ten times as many parameters. This is not unreasonable, as this is simply an indicator that the task of classifying AD within this dataset has already been solved. With this many parameters, there does not exist enough complexity within the input images for all of them to be necessary. As such, it is very likely that many of the parameters within the larger model contribute almost nothing to the output. The use of L2 or L1 regularizations may provide significant increases to model performance, as these methods penalize large weights and would force many of the useless weights down to almost 0.

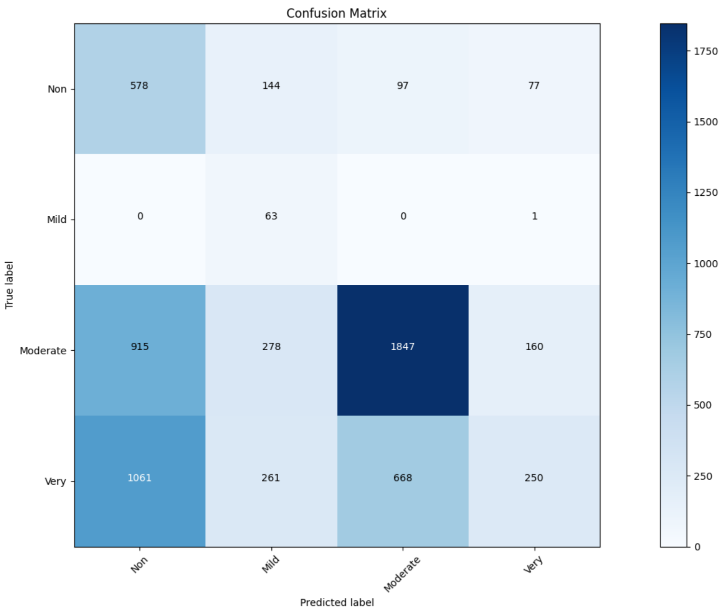
V. DISCUSSION

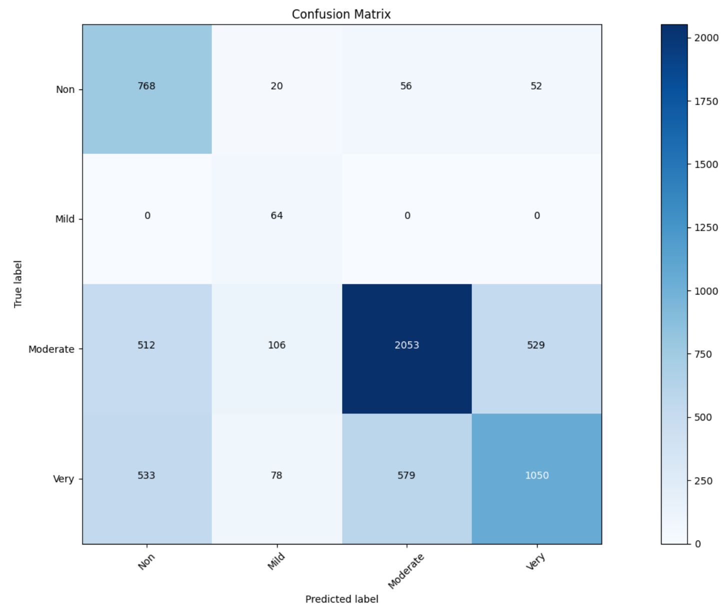
In this paper, it appears that the CNNs vastly outperformed the decision trees regarding their capabilities to categorize brain MRIs into different severities of AD. Given this, it is worth further looking into whether CNNs can reasonably benefit AD diagnosis, as there are large potential benefits in doing so. Widespread use of CNN AD detection would make diagnoses fast, accurate, and consistent. However, additional research would be required to verify the results in this paper, preferably utilizing different datasets and different methodologies. There is also no guarantee that the results presented here can be replicated in a clinical trial. Recommendations for future study would be the combined use of decision trees with CNNs, perhaps training the tree using CNN-extracted features, as well as utilizations of Random Forest and gradient boosting to increase decision tree performance.

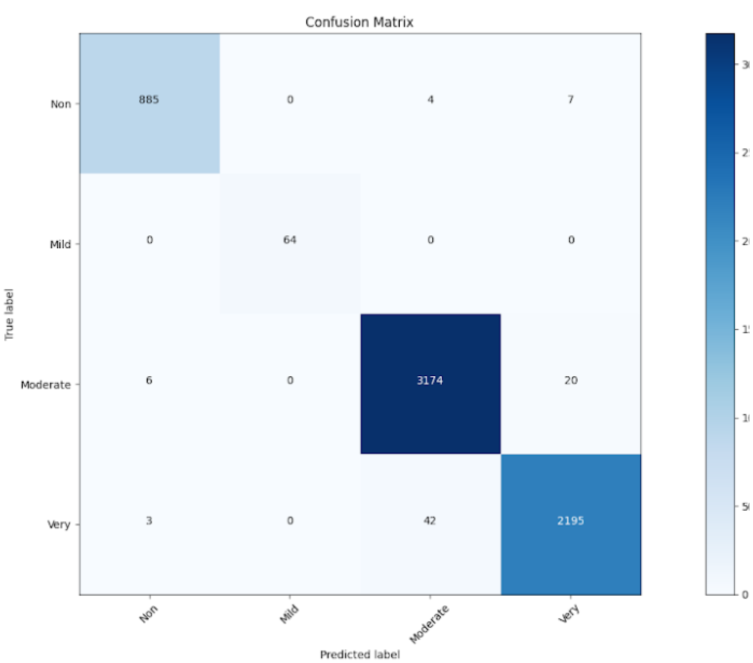
VI. CONCLUSION

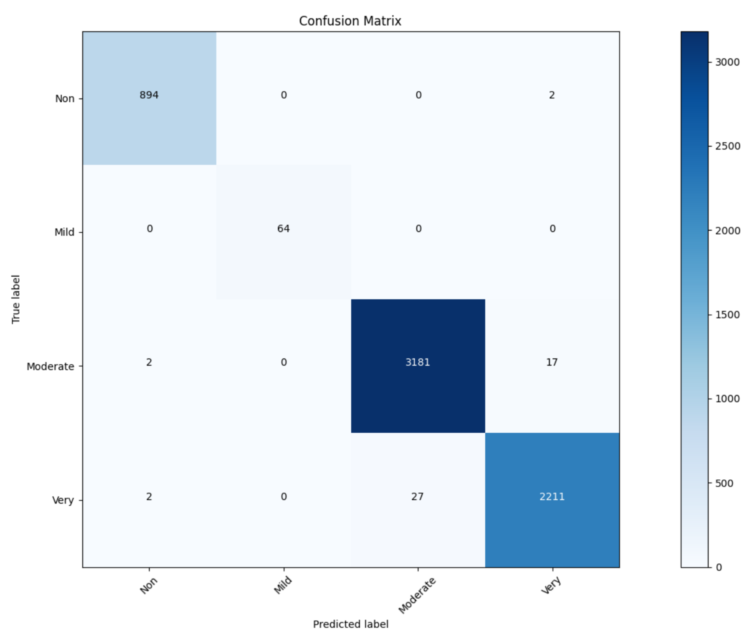
In this paper, CNN models were evaluated against decision tree models to test their capabilities in categorizing brain MRI images according to AD severity. Through multiple simulations, CNNs were consistently better at doing so than decision trees, due to their ability to handle large amounts of image data. Moving forward, clinical trials should be done of CNN AD diagnosis, in hopes that it can be widely used in the medical field.

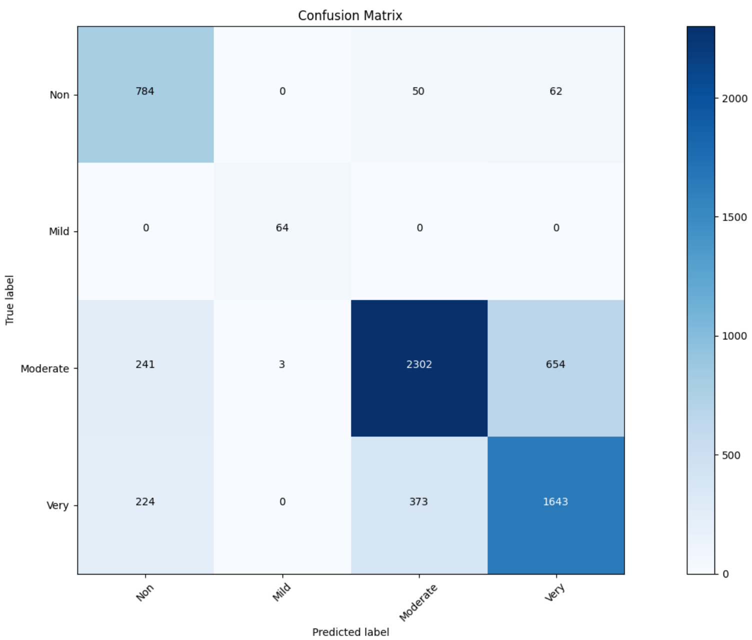
VII. RESULTS











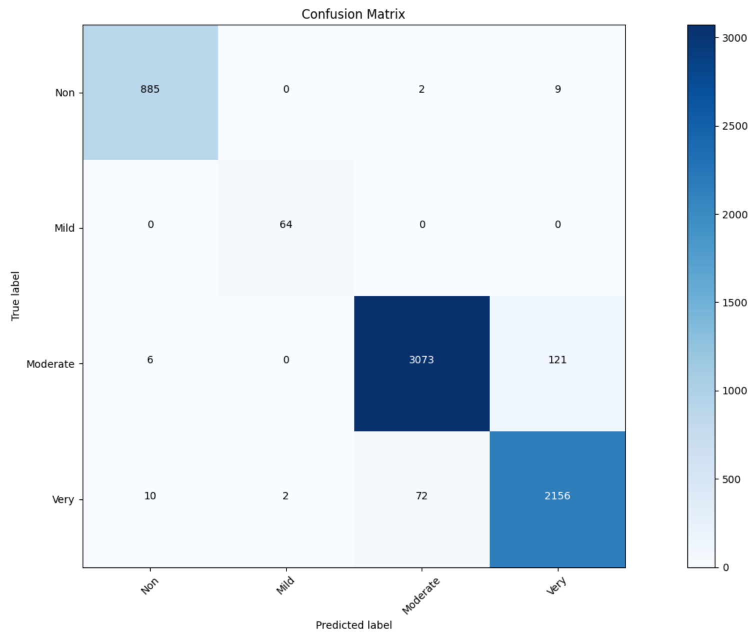


Figure 9: All confusion matrices, in the following order (left -> right) (top -> down)
5-level tree, 10-level tree, CNN #1, CNN #2, Transfer learning frozen, Transfer learning unfrozen

Works Cited

aashidutt3. “🧠alzheimer Classification with MRI Images.” Kaggle, Kaggle, 10 Oct. 2022, www.kaggle.com/code/aashidutt3/alzheimer-classification-with-mri-images.

Bush, Ashley I. "The metallobiology of Alzheimer's disease." Trends in neurosciences 26.4 (2003): 207-214.

Du, Xiao, et al. "Fault detection of aero-engine sensor based on inception-CNN." Aerospace 9.5 (2022): 236.

Farooq, Ammarah, et al. "A deep CNN based multi-class classification of Alzheimer's disease using MRI." 2017 IEEE International Conference on Imaging systems and techniques (IST). IEEE, 2017.

Iman, Mohammadreza, et al. “A Review of Deep Transfer Learning and Recent Advancements.” Technologies, vol. 11, no. 2, Mar. 2023, p. 40. Crossref, https://doi.org/10.3390/technologies11020040.

Scheltens, Philip, et al. "Alzheimer's disease." The Lancet 397.10284 (2021): 1577-1590.

Shin, Hoo-Chang, et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning." IEEE transactions on medical imaging 35.5 (2016): 1285-1298.

Uraninjo. “Augmented Alzheimer MRI Dataset, Version 1.” *Kaggle*, 20 Sept. 2022, www.kaggle.com/datasets/uraninjo/augmented-alzheimer-mri-dataset.