Predicting the Severity of Alzheimer's/Dementia from Magnetic Resonance Imaging Scans Using a Deep Learning Approach

Akshaj V. Satyawada

Author Note

I would like to thank my mentor Kasra Koushan for all the time and effort he spent helping me go through the process of writing this research paper. His knowledge and insight was invaluable to the success of this project. I would also like to thank my family for their support throughout this process.

Correspondence concerning this article should be addressed to Akshaj Satyawada. Email: 25aswada@gmail.com

Abstract

Dementia and Alzheimer's disease are growing health challenges, as an estimated 10 million new diagnoses happen annually, highlighting the need for more efficient diagnostic processes. The lengthy process involves scheduling appointments, pre-screening assessments, and arranging screening appointments, consuming valuable time for healthcare professionals. A streamlined process leveraging computer-based screening could optimize efficiency. In fact, artificial intelligence-based systems also have potential to decrease overall healthcare costs by upwards of 10%, or over $360 billion annually (The Financial Cost of AI in Healthcare, 2023). Deep learning models, with their precision and technical capabilities, offer a promising solution. By automating screening tasks using deep learning, they can reduce administrative burdens, enhance accuracy, and expedite diagnoses, ultimately improving patient care and outcomes. The overall approach used in this research was utilizing a deep learning model, specifically a convolutional neural network. A convolutional neural network is a type of neural network architecture that specializes in identifying and making sense of patterns in image data. For this research, this type of neural network was mostly used for classifying images of dementia MRI scans based on their severities. It used certain classifying methods such as multi-class classification and a high volume of data (roughly 6400 images) to accurately distinguish between different levels of dementia severity. The model's performance metrics are robust, achieving an accuracy of 0.989, a precision of 0.99, a recall of 0.989, and an f1-score of 0.99 after 20 training epochs. These results suggest a high potential for CNN-based models to be integrated into clinical workflows, providing rapid and reliable assessments of dementia severity.

Keywords: dementia, Alzheimer's, deep learning, convolutional neural network, images, accuracy

Predicting the Severity of Alzheimer's/Dementia from Magnetic Resonance Imaging Scans Using a Deep Learning Approach

What is the effectiveness of deep learning models in predicting dementia diagnoses using brain scan images, considering varying degrees of dementia severity (no dementia, mild dementia, moderate dementia, and very mild dementia)? Dementia is a general term that refers to a decline in cognitive function. Alzheimer's disease is the most common cause of dementia. It is categorized by unusual protein deposits in the brain that lead to the gradual destruction of brain cells. Alzheimer's disease leads to a significant alteration of the structure of the brain, which can be recognized by deep learning algorithms. Deep learning models show promising effectiveness in predicting dementia diagnosis using brain scan images across varying degrees of dementia severity. This capability is crucial as it enables computers to perform this task with remarkable accuracy and efficiency, surpassing human capabilities. The process of diagnosing dementia using older methods is much more costly and time consuming for both the patient and the healthcare professional. By leveraging deep learning algorithms, healthcare professionals can rely on advanced technology to quickly and accurately assess dementia severity from brain scans. This not only streamlines the diagnostic process but also frees up valuable time for doctors and healthcare professionals to dedicate to more critical and human-centric tasks, ultimately enhancing patient care and treatment outcomes.

Using MRI brain scans to identify dementia severity is a multi-class classification supervised learning problem. While conducting this research, over 6,000 MRI brain scan images of varying severities of dementia were used to create a convolutional neural network that returns the predicted label of the image that was inputted. For example, if an inputted image was of

"Very-Mild" severity, the output of the model would be a label predicting the classification of this image, for example "Very-Mild".

Background

There have been many approaches to solving the problem of accurate dementia severity analysis. One such approach was to analyze the accuracy of different types of input data in predicting the correct dementia diagnosis. For example, in that study, the authors analyzed different types of data modalities such as image data, clinical variables, and voice data. According to their research, machine learning based on image data seemed to yield the most accurate results when compared to other forms of data (Zhang et al., 2024). The aim was to build off of this knowledge, which is why over 6000 images were used as data to accurately train the model. Images were likely the most accurate input because of their versatile uses. Image data contains a lot of information for a model to learn from and also allows for feature extraction from the raw pixel data. Additionally, a simple grayscale image (for example, a brain MRI scan) contains a lot of data about pixel intensity values and contrast, which helps highlight key features and patterns for the model to recognize. Another study that was conducted aimed to find the performance of machine learning algorithms for predicting progression to dementia in memory clinic patients. After using this approach, the researchers found out that using machine learning models was superior to 2 other pre-existing predictive functions. In that particular study, the researchers used a Support Vector Machine, which is a type of classification algorithm that aims to find the optimal hyperplane to adequately separate different classifications of data (Battineni et al., 2019). This again reinforced the idea that using machine learning would be an effective way to analyze images for the highest accuracy prediction. Machine learning offers a wide variety of advantages compared to other predictive methods because machine learning

algorithms can learn patterns and relationships from data without relying on predefined rules or assumptions. To add to this, machine learning models can automatically select relevant features that regular predictive functions cannot, because they have to be hand selected, leading to a possible source of error. All in all, examining other approaches to a similar research question provided valuable insight and helped me set up my model as effectively as possible. However, there has not been much research on the use of convolutional neural networks to identify dementia/Alzheimer's disease severity, which was the aim of this research.

Methodology/Models

Dataset

The dataset that was used for this research was a dementia/Alzheimer's MRI scans dataset found at https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset/data. The images that were in the dataset were 128 pixels by 128 pixels grayscale images of MRI brain scans. The data was already preprocessed by the author of the dataset. The images were collected from several hospitals, websites, and public repositories, and were separated into four categories: No Dementia, Moderate Dementia, Mild Dementia, and Very-Mild Dementia. The No Dementia class contained the most images with 3200, followed by Very-Mild Dementia which contained 2240 images, then Mild Dementia with 896 images, and finally Moderate Dementia with 64 images. A useful feature of the images in the dataset is that their individual pixel values can be extracted, which helps in identifying patterns and relationships within the individual pixels themselves. Figure 1 shows a sample image from the dataset.

Figure 1. Very-Mild image from the dataset.

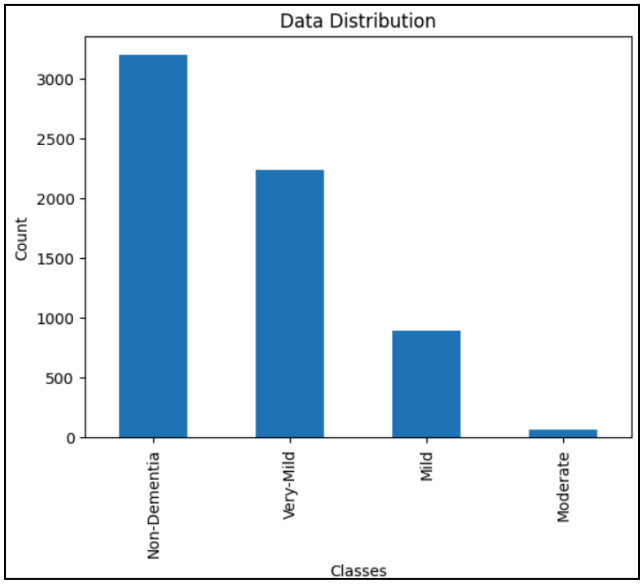

Figure 2. Histogram of the amount of images in each category.

Another helpful feature that can be extracted from images is the mean intensity value. The mean intensity value is the average intensity of all the pixels in an image. It is most useful when

normalizing images, which is when you subtract the mean intensity value from each pixel and dividing by the standard deviation. This allows the input data to be standardized, making it easier for the model to learn effectively. Figure 3 shows the mean intensity values separated by classification of the images.

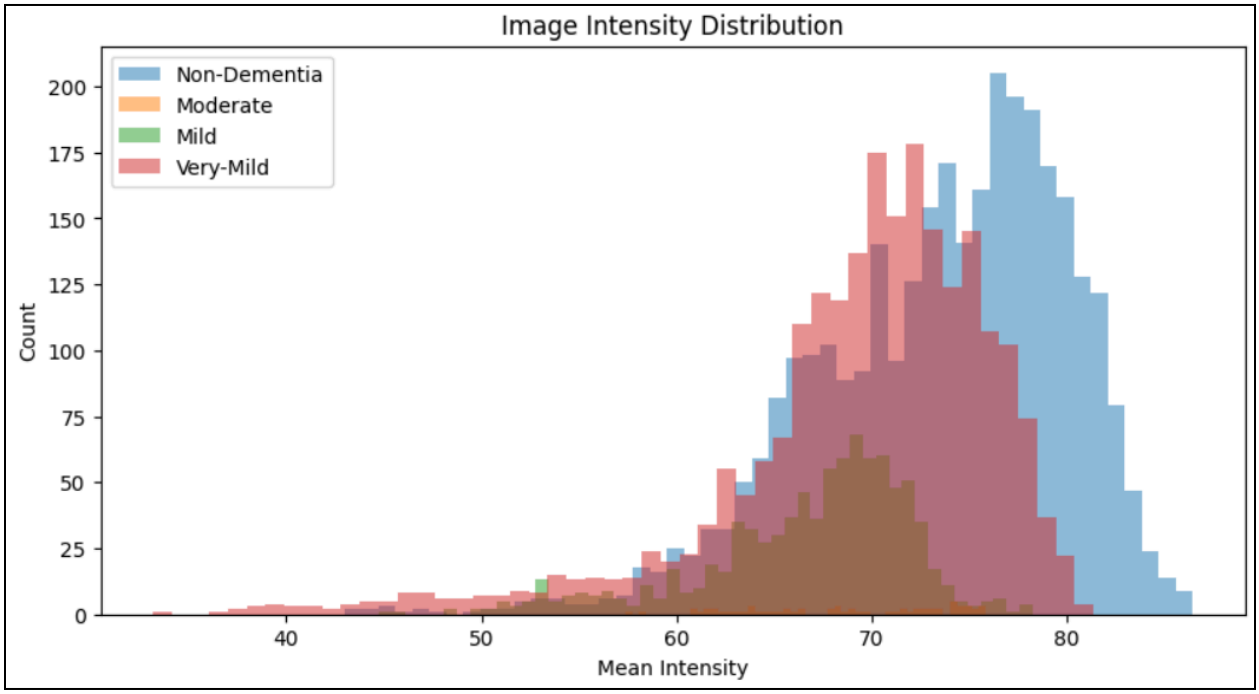

Figure 3. Image intensity distribution.

In Figure 3, the median of the mean intensities is approximately around 70-80 and the overall distributions all seem to be skewed to the left. This could indicate that the image is predominantly dark or has large areas of low intensity values. This usually occurs in images with dark backgrounds, shadows, or areas of low illumination, which is true in this case. Additionally, image equalization is a great way to enhance subtle parts of the image more. Equalization improves image contrast by redistributing pixel intensity values, resulting in a more balanced distribution across the intensity range. This enhances visual quality, making details more

distinguishable, which is particularly useful in tasks like medical imaging and object detection. In the context of my research, this means highlighting the ridges of the brain tissue more clearly. Figure 4 compares the original image to the equalized version of the same image.

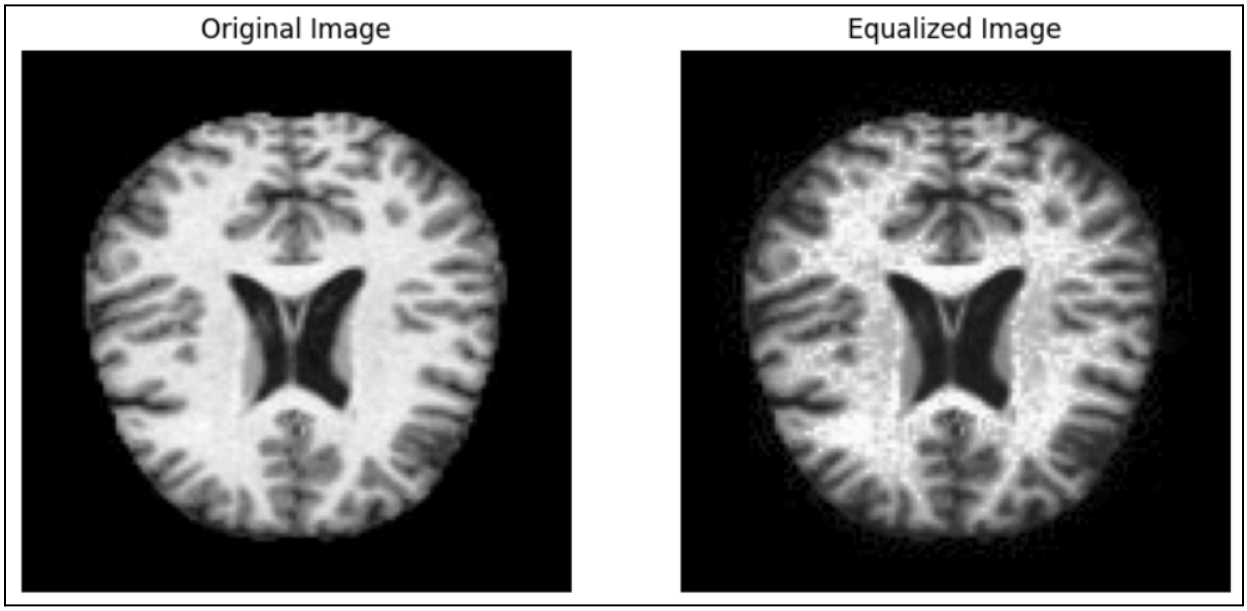

Figure 4. Equalizing the MRI scan images.

The pixel arrays for each image were all loaded into a Pandas DataFrame as this helps analyze each data point effectively. In terms of setting up the training and testing data, the train_test_split function from the sklearn library was utilized, and a test size of 0.2, which means that 80% of the dataset will be training data and 20% of the dataset will be testing/validation data, was used.

Model

To utilize a convolutional neural network, the TensorFlow-Keras framework was used for this specific research. As stated previously, CNNs are powerful machine learning algorithms that specialize in image classification tasks. A simple overview of the CNN model that was used is as follows. The first few layers of the algorithm extract features from input images, and then an

activation function (in this case ReLU - Rectified Linear Unit) is applied to make the model capable of learning complex patterns in the image data. After that, pooling layers are added to retain the most relevant information while also predicting the input's spatial size. Finally, flattening layers are added to flatten the feature map (a multi-dimensional representation of features of an input image) into a one-dimensional array. The result is then outputted.

When starting to build the model, the following was imported from tensorflow.keras: .models, .layers, and .optimizers, and from sklearn: .preprocessing and .model_selection. The input data was first split using the train_test_split function from sklearn with a test size of 0.2, as stated previously. The training data was then split, where X_train became the images without the labels and y_train became the labels that corresponded to the images. The same was done for X_val and y_val (for validation/testing data). An image preprocessing step was taken to normalize the pixel values of the images of the training and validation sets within the range [0, 1]. This is important because it prevents certain features from dominating the others due to having a larger magnitude. After that, a class from sklearn called "LabelEncoder" was used to assign a unique numerical value to each of the 4 categories. The input shape was defined following the label encoding process, with the images being 128 pixels by 128 pixels by 1 (because they are in grayscale). Then the learning rate hyperparameter is set to 0.001. The learning rate (or step-size) is a hyperparameter that defines how fast or slow a model adjusts the weights of its neurons towards the optimal weight. Finding an optimal learning rate is important to the success of a model because it is one of the most crucial hyperparameters that determines the efficiency and effectiveness of a model. In this case, a learning rate of 0.001 paired with 20 epochs produced the most accurate result. After defining the learning rate, the Sequential model

was defined. A Sequential model is a type of deep learning model that takes input data and goes through layer by layer sequentially until it reaches an output.

```
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(4, activation='softmax'))  # Output layer

optimizer = Adam(learning_rate=learning_rate)
model.compile(optimizer=optimizer,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Display model summary
model.summary()

# Display model summary
model.summary()

X_train = X_train.reshape(-1, 128, 128, 1)
X_val = X_val.reshape(-1, 128, 128, 1)

# Train the model
history = model.fit(X_train, y_train_encoded,
                    epochs=20,
                    validation_data=(X_val, y_val_encoded))
```

Figure 5. Model layers and training.

Figure 5 shows the layers that were added to the model. Lastly, the optimizer that was used for this multi-class classification problem was Adam with the learning rate as a parameter. An optimizer in a convolutional neural network is an algorithm that updates the weights and biases of the network during the training process. Adam stands for “Adaptive Moment Estimation” and is especially useful in image classification tasks. It updates the weights of the network while also

using very little memory in the process, which is why it is used for large datasets and deep networks.

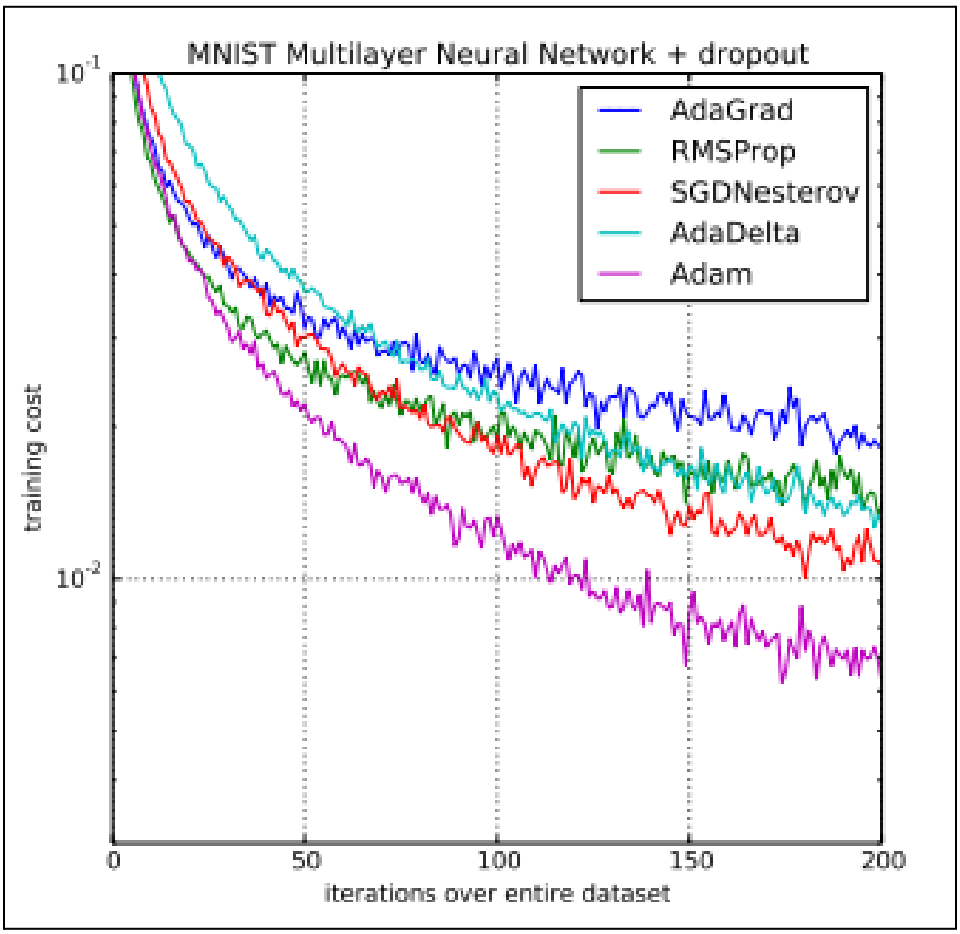

Figure 6. Comparison of neural network optimizers.

Figure 6 shows the Adam optimizer compared to other optimization algorithms. A lower training cost is ideal and over the same number of iterations over a dataset, Adam performed the best overall. For the loss function, sparse_categorical_crossentropy was used as it is good for multi-class classification. It calculates the cross-entropy loss between the true labels and the

predicted probabilities, taking into account the sparsity of the target distribution. The formula for categorical cross-entropy is shown in Figure 7.

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i$$

Figure 7. Categorical cross-entropy formula.

For metrics, accuracy was used as it is most common for classification tasks. It calculates the proportion of correctly classified samples out of all samples. The last step before training the model is to reshape the training and validation data. Typically, CNNs take 4-dimensional data as input, so our current (128, 128, 1) images need to be reshaped. To accomplish this, numpy's .reshape function was used to reshape the original input into (-1, 128, 128, 1) where -1 tells NumPy to automatically calculate the first dimension based on the total number of elements and the other dimensions. The model was then trained for 20 epochs.

Results and Discussion

Model Performance

The main goal of this research was to maximize the accuracy of the model for the validation data. In terms of accuracy, the model performed well, yielding an accuracy score of around 0.989 on the validation data. Accuracy is calculated by dividing the total number of correct predictions by the total number of predictions. In the case of this model, out of 1280

predictions, it predicted 1267 correctly and 13 incorrectly. To see which images were incorrectly classified, a confusion matrix can be utilized.

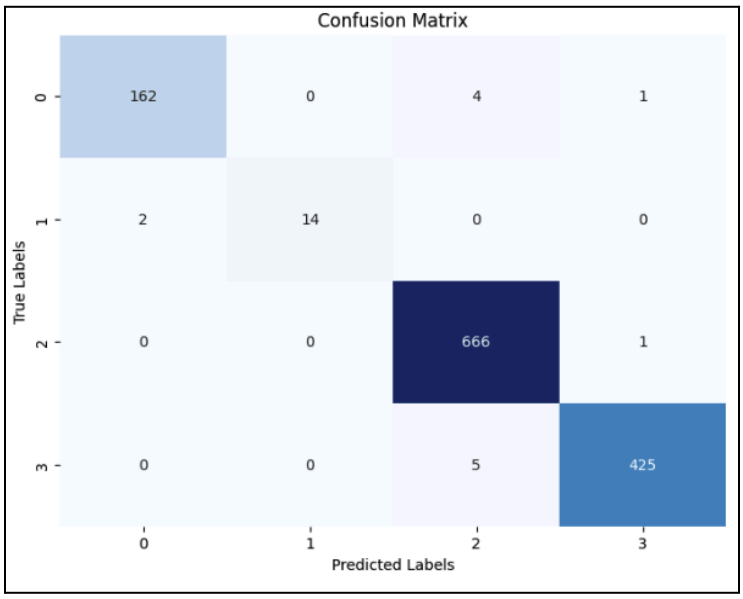

Figure 8. Confusion Matrix of the model's predictions vs. the true label.

	Predicted Mild	Predicted Moderate	Predicted Non	Predicted Very-Mild
Actually Mild	162	0	4	1
Actually Moderate	2	14	0	0
Actually Non	0	0	666	1
Actually Very-Mild	0	0	5	425

Figure 9. Confusion Matrix table.

Figure 8 and Figure 9 both show the categories that were correctly and incorrectly predicted, along with the frequencies. Very-Mild and Mild were both incorrectly predicted the most with 5

incorrect predictions followed by Moderate with 2 incorrect predictions and Non with only 1. An interesting observation is that even though the Moderate dataset had only 64 images, it still produced less inaccuracies compared to Mild and Very-Mild which had 896 and 2240 images each respectively, although proportionally, the error is greater. The difference between the Mild images and the Very-Mild images could be very slight compared to the difference between Moderate and No Dementia, thus increasing the likelihood of error. Also, the proportion of inaccuracies for the Moderate category was 216 or 12.50%, whereas the proportion of inaccuracies for the Mild and Very-Mild categories was 5167 or 2.99% and 5430 or 1.16% respectively. Another measure of effectiveness for the model is precision. Precision is calculated by dividing the number of correct positive predictions by the total number of positive predictions. In terms of Alzheimer's/dementia prediction, precision is arguably the most important statistic. The cost of false positives is high in this situation, as a patient could be spending thousands of dollars on treatment for a condition they do not even have. The precision of my model was 0.9898875232645038. Additionally, recall is another important metric used to determine the proportion of true positive predictions out of the total number of actual positive instances, including both true positives and false negatives. A high recall score is crucial in situations where the cost of a false negative is high, such as in Alzheimer's/dementia diagnoses. If a patient receives a false negative report, they could miss a potentially life-threatening condition which is very dangerous.

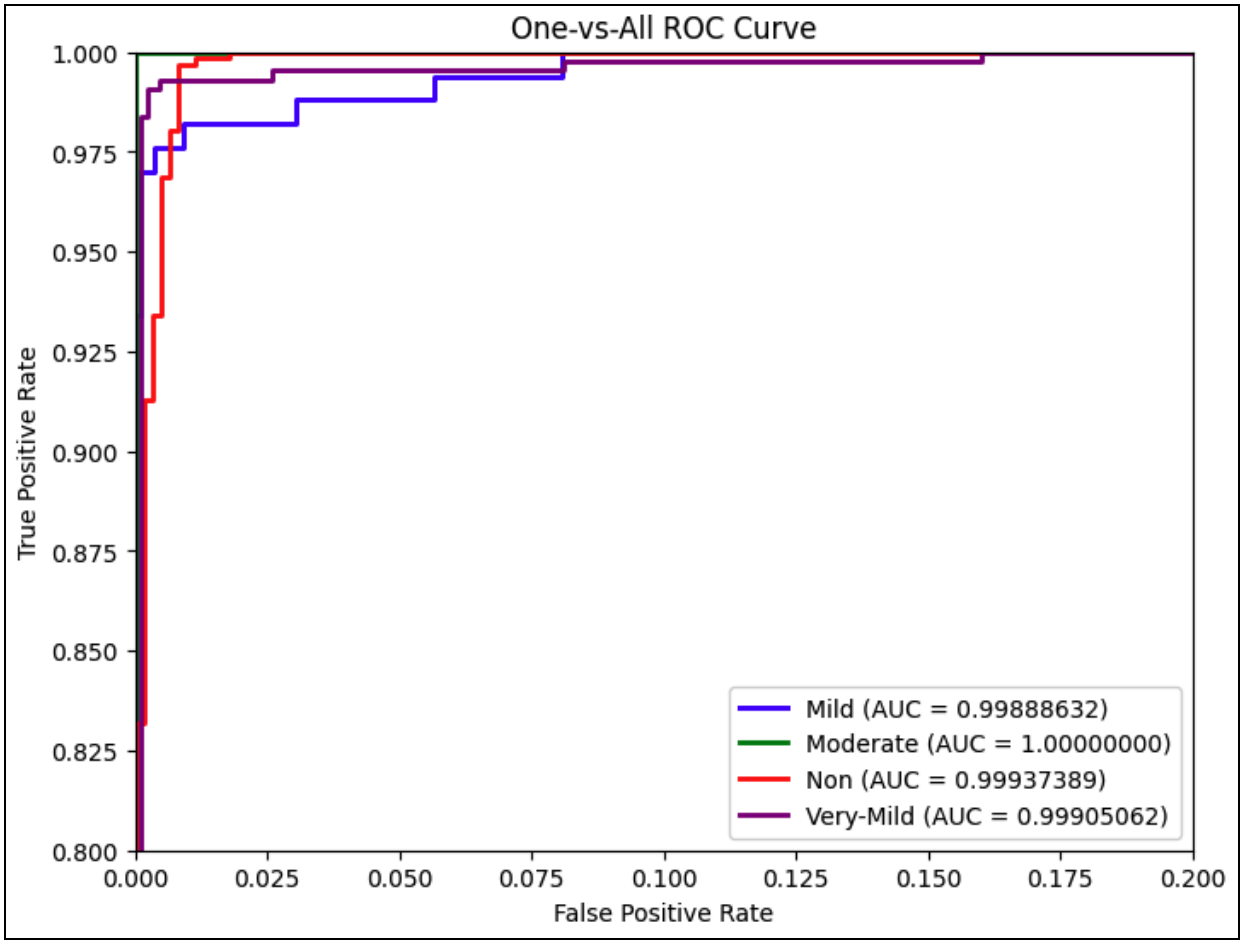

Figure 10. One-vs-All receiver operating characteristic curve.

Displayed in Figure 10 is the One-vs-All ROC (Receiver operating characteristic) curve. This curve is an important part of verifying the accuracy of a machine learning model. It plots the false positive rate on the x-axis against the true positive rate on the y-axis. In terms of our model, a higher AUC (area under the curve) for an ROC curve is considered better. This is reflected in our training as none of the AUCs are below 0.99, indicating high performance accuracy. A model with perfect accuracy would have an AUC of 1, as seen with the moderate data, but this is not exactly the case. A model that is random would have an approximate AUC of 0.5, indicating low performance. Because the Moderate training dataset was limited, the ROC curve has a hard time

rounding to adjust to this lack of quantity which causes it to round up to 1.0. The actual AUC would be slightly less.

Hyperparameters

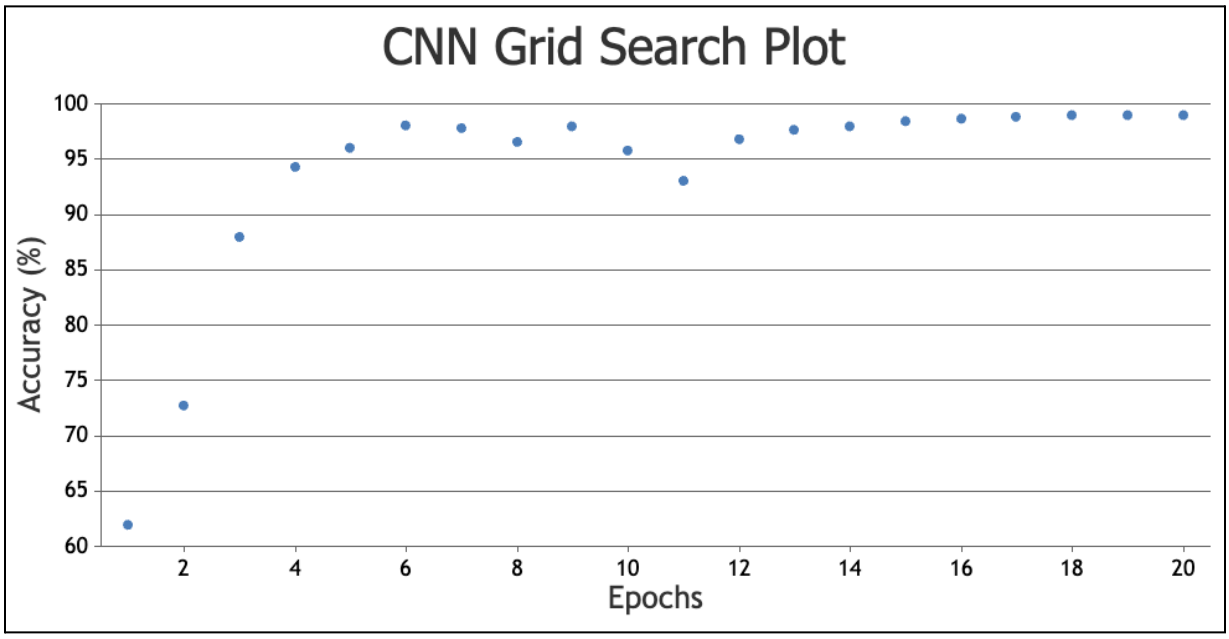

Figure 11. CNN grid search plot.

Another way to fine tune this model for the best results is to create a grid search plot. This grid search plot in Figure 11 plots the epochs against the accuracy. As the number of epochs approaches 11, the accuracy starts to dip, but after that it recovers and approaches 100% as the number of epochs approaches a larger and larger number. Grid search plots make it easier to find the right parameters for peak model performance.

Hyperparameter	Value 1	Result 1	Value 2	Result 2	Value 3	Result 3
Layers	6	98.20%	8	95.78%	14	86.48%
Epochs	10	95.78%	15	98.44%	20	98.98%
Learning Rate	0.1	52.11%	0.01	69.45%	0.001	98.28%

Figure 12. Hyperparameters and adjustments.

Figure 12 shows 3 different hyperparameters and their effects on the overall performance of the model.

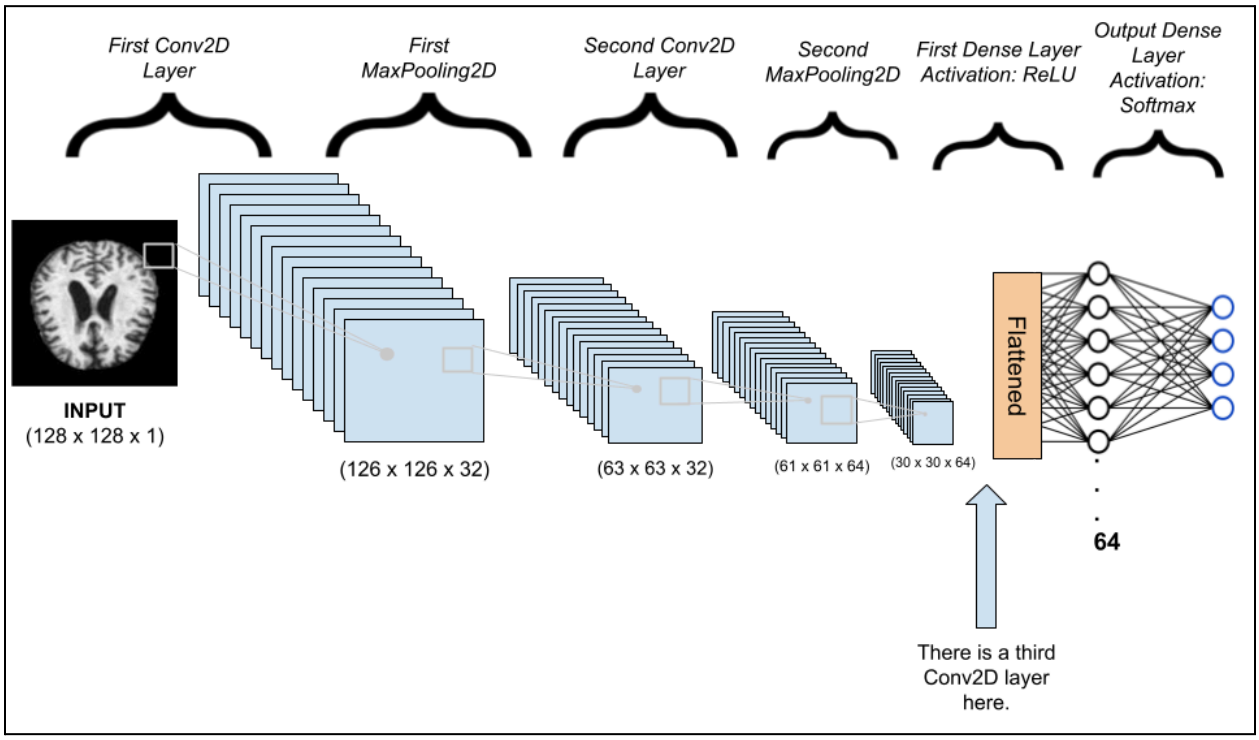

Figure 13. Simple model illustration.

The first of three hyperparameters that were tuned is the number of layers of the CNN. In the case of our data, having more layers didn't necessarily make the model better, as adding more

layers caused the model to overfit and thus reduced the overall accuracy. Figure 13 shows a rough outline of the model and its layers. The number of epochs of training the model had the opposite effect of the number of layers, where training with more epochs contributed to the model performing better. A higher performance could have been achieved by training even more than 20 epochs, and this is something that will be looked at in the future. Lastly, a slower learning rate seemed to be more beneficial for the model. The learning rate is a hyperparameter that chooses how fast or slow the model's algorithm updates its parameters during training, which influences the speed and convergence of the learning process. A slower learning rate turned out to be the best for this particular situation, as the faster learning rates like 0.1 and 0.01 caused the model to converge too fast to a suboptimal solution.

Transfer Learning

Transfer learning could be used with this model to see how well it performs using different validation data. Transfer learning offers many advantages. For example, if the model is successful with transfer learning, it allows us to generalize our original model and increase its scope. Also, it allows the training process to use less training data, which increases computational efficiency. For this particular model, MRI scans for brain tumors could be inputted to see how accurate the model is with this new input data. To do this, 3 new layers were added to the existing dementia analysis model for brain tumor classification. Then, the number of classes was reduced from 4 to 2, as the brain tumor model was a simple binary classification problem. Everything else was kept the same. The results are as follows. The overall validation accuracy was 0.8286. Specifically for the "No Tumor" dataset, the precision, recall, and f1-score were 0.89, 0.80, and 0.84 respectively. For the "Yes Tumor" dataset, the precision, recall, and f1-score were 0.76, 0.87, and 0.81 respectively.

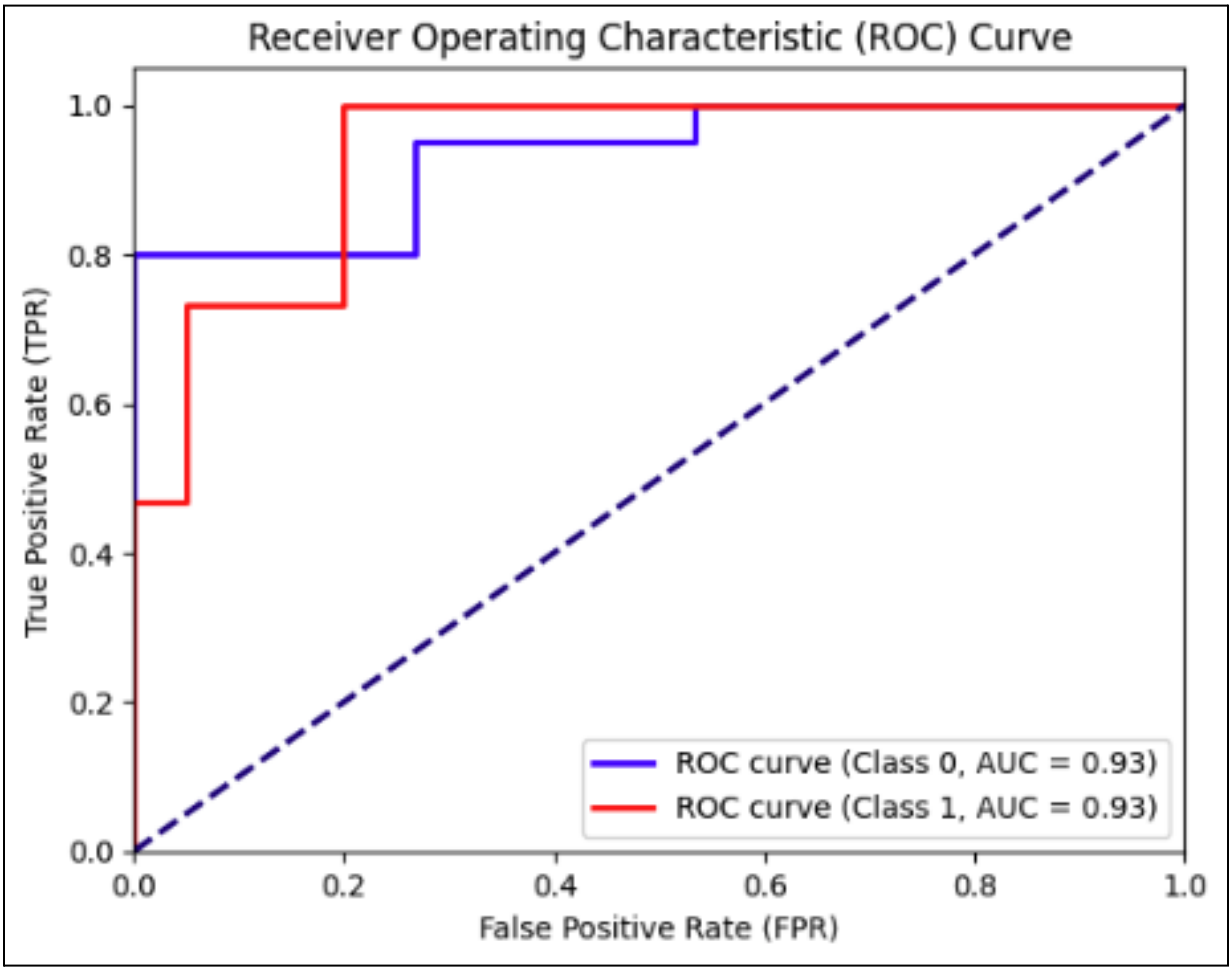

Figure 14. ROC curve for the brain tumor model.

Shown in Figure 14 is the ROC curve for this model. The area under the curve is 0.93 which is relatively high considering the model was not specifically designed for brain tumor classification. Overall, the results of this model are promising, as they show that CNNs, such as the dementia classification model, can be generalized to more data than initially thought.

Conclusion

All in all, using convolutional neural networks proved to be significantly successful. A maximum validation accuracy of 0.989 proved that the network was able to accurately assess

factors in the intricacies and nuances of the image data to come up with the final prediction. There were many hyperparameters that could have been adjusted, but more epochs and a slower learning rate proved to be the most important when trying to maximize accuracy. These results are very significant in a practical way because they show that machine learning is a very open and doable solution for the efficiency problem of diagnosing Alzheimer's/dementia. The reason the model might have performed well could be because of the high volume of training data. Thousands of images per category definitely helped the model in locating the fine details of each individual image. However, there are a few things that could have been improved. First, the Moderate Dementia dataset only had 64 images. More data in this category could have helped in increasing the accuracy and true positive rate seen in the confusion matrix. Also, using more nuanced data could have been helpful, as not all MRI scans will look exactly how the dataset images looked. One thing that could definitely be done with this project is to utilize it for more transfer learning. Using this model on other similar but different data would be interesting to see how well the model assesses different types of image data, like for brain cancer or other cognitive diseases. All in all, the CNN model works accurately when categorizing MRI brain scans for different dementia severities, and the hope is to broaden the scope for its usage in the real world and improve upon it in the future.

References

Kumar, S. (2022, March 27). Alzheimer MRI preprocessed dataset. Kaggle. https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset/data

Zhang, J., Song, L., Miller, Z., Chan, K. C. G., & Huang, K. (2024). Machine learning models identify predictive features of patient mortality across dementia types. Communications Medicine, 4(1), 1–13. https://doi.org/10.1038/s43856-024-00437-7

1.17. Neural network models (supervised) — scikit-learn 0.23.1 documentation. (n.d.). Scikit-Learn.org. https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Battineni, G., Chintalapudi, N., & Amenta, F. (2019). Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). *Informatics in Medicine Unlocked*, *16*, 100200. https://doi.org/10.1016/j.imu.2019.100200

Javeed, A., Dallora, A. L., Berglund, J. S., Ali, A., Ali, L., & Anderberg, P. (2023). Machine Learning for Dementia Prediction: A Systematic Review and Future Research Directions. Journal of Medical Systems, 47(1). https://doi.org/10.1007/s10916-023-01906-7

tf.keras.layers.Conv2D | TensorFlow Core v2.4.1. (n.d.). TensorFlow. https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D

Alzheimer's Association. (2024). What Is Dementia? Alzheimer’s Disease and Dementia; Alzheimer’s Association. https://www.alz.org/alzheimers-dementia/what-is-dementia

Dementia symptoms and areas of the brain | Alzheimer’s Society. (n.d.). Www.alzheimers.org.uk. https://www.alzheimers.org.uk/about-dementia/symptoms-and-diagnosis/how-dementia-rogresses/symptoms-brain#:~:text=As%20Alzheimer

https://www.facebook.com/folio3software. (2023, July 26). The Financial Cost of AI in Healthcare - Best Guide for 2023. Folio3. https://digitalhealth.folio3.com/blog/cost-of-ai-in-healthcare/