CS 584 Machine Learning

Spring 2024 – Final project report

ALZHEIMER'S DISEASE DETECTION AND CLASSIFICATION

Team Members:

- Tamilarasee Sethuraj A20553416
- Krishnavardhini Kittusamy A20502176

1 Introduction

Alzheimer's disease (AD) is the most common type of Dementia, a neurodegenerative brain disease that can significantly affect the quality of life of older individuals. This is a long-lasting problem that slowly reduces and ultimately kills one's thinking abilities and leads to a huge loss of cognitive and mental strength, behavioral problems as well as language difficulties. The commonly used method is to predict the disease based on Magnetic Resonance Imaging (MRI) for the purpose of early intervention. Although there is no cure for Alzheimer's, there are medications and interventions to manage symptoms and slow down the progression to some extent. Early treatment proves to be an effective way of preventing its deterioration. The objective of the project is to preprocess the MRI images, explore and use machine learning algorithms to classify brain MRI images into different stages of Alzheimer's disease. We will evaluate the trained models using appropriate evaluation metrics and compare their performance to identify the most effective model with highest possible accuracy and visualize the results.

1.1 Brief survey of Previous Works and Results

Many scientists have made sincere efforts to discover a variety of techniques to detect Alzheimer's using MRI data. Those techniques include the extraction of discriminative features from a large set of features, and selecting efficient classification models. The advancements in technology have had a profound impact on healthcare, revolutionizing various aspects of the industry. ML algorithms can integrate and analyze diverse data types, including neuroimaging, biomarkers, and clinical information.

Recent techniques which focus on the finding of the current stage/severity of disease, such as very mildly demented, mildly demented, moderately demented.

There are several methods explored by scholars to classify the MRI images such as Principle Component Analysis (PCA), K-means Clustering, Linear Discriminant Analysis (LDA), Support Vec- tor Machine, VGG19/XGBoost, EfficientNet, deep learning-based approaches like Convolutional Neu- ral Networks (CNNs).

Some of the difficulties that scientists have faced in implementing the models are extracting clear image features that show small variants of brain cells changes, using complex models restricts the ability to provide relevant information based on which the decision was made (classification). It is important to know the logics/root of the decision as it involves patient care and hence, we may need to rely on simple explainable models which may not have much scope for accuracy or improvement.

The methods that has been used previously with their results has been displayed in table-1below.

2 Problem description

The primary objective of this study is to discern subtle alterations in cognitive function indicative of Alzheimer's Disease (AD) severity, facilitating informed decisions in patient care and early intervention strategies. Extensive efforts have been undertaken by researchers to exploit MRI data for AD detection, encompassing the extraction of discrim- inative features from vast datasets and the imple- mentation of efficient classification models. While initial approaches predominantly focused on binary classification which determines the presence or absence of AD, recent advancements emphasize categorizing disease stages, from very mildly demented to moderately demented.

Researchers encountered several challenges during model implementation, particularly in extract-

Model	Train Results (%)	Test Results (%)
Random Forest	75.4	71
Logistic Regression	78.1	75
Support Vector Machine	99.4	99
CNN	99.06	99

Table 1: Previous works and results: Train and test results of different classification methods.

ing distinct image features reflecting minute variations in brain cell morphology. Furthermore, the utilization of intricate models often hampers the interpretability of decision-making processes, essential for patient-centric care.

Consequently, there is burgeoning need for simplistic, interpretable mod- els, albeit potentially sacrificing accuracy and scal- ability.

In this study, we propose a comprehensive investigation of various machine learning models, including Random Forest, Logistic Regression, Support Vector Machine, and Convolutional Neural Networks (CNNs). Specifically, we aim to explore different hyperparameters for each model, meticulously evaluating their performance to ascertain optimal parameter settings. Through this systematic exploration, we endeavor to identify the parameter configurations that yield superior classification accuracy and robustness.

Results from preliminary experiments showcase the following performances:

- Random Forest: Training accuracy of 75.4%, testing accuracy of 71%.
- Logistic Regression: Training accuracy of 78.1%, testing accuracy of 75%.
- Support Vector Machine: Training accuracy of 99.4%, testing accuracy of 99%.
- CNN: Training accuracy of 99.06%, testing accuracy of 99%.

Moving forward, we will delve deeper into hyperparameter tuning to discern the parameter combinations that enhance model performance, ultimately facilitating more accurate and reliable AD classification.

3 Data Description

We have used MRI images as input for our classification problem. The images are collected from the data set “Alzheimer MRI Dataset” from

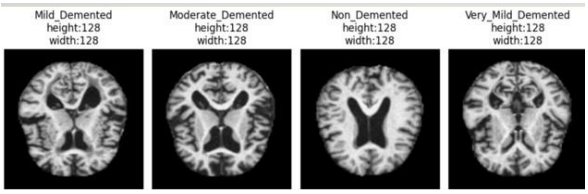

Figure 1: Image types

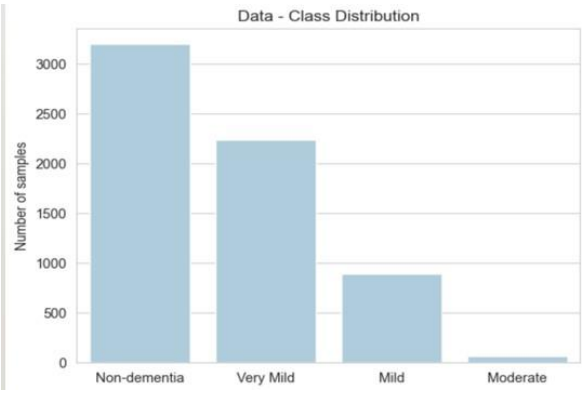

Figure 2: class distribution

Kaggle which was collected from several websites/hospitals/public repositories. The dataset comprises 6400 preprocessed MRI images, all resized to 128 x 128 pixels, depicting various stages of Alzheimer’s disease progression. There are a total of 6400 images. They are categorized into four classes based on the severity of dementia:

- Class 0: Non-Demented (3200 images)
- Class 1: Very Mild Demented (2240 images)
- Class 2: Mild Demented (896 images)
- Class 3: Moderate Demented (64 images)

4 Data Preparation

We have cleaned and processed the collected data for the implementation. We performed several preprocessing and augmentation steps on the dataset to enhance data quality and improve the model’s robustness. The data is preprocessed by scaling it

and then applying PCA to reduce its dimensionality, ensuring that the transformed data retains most of its variability.

We see a significant class imbalance in the dataset, especially for classes 2 and 3 (Mild Demented and Moderate Demented) compared to classes 0 and 1 (Non Demented and Very Mild Demented). This imbalance need to be addressed during model training to prevent bias towards the majority classes.Although we have classes with fewer samples, they are crucial for capturing the variability associated with dementia progression

Some of the key steps followed are as below:

Image Preprocessing:

- All images are resized to a standard size of 128 x 128 pixels. The images are flattened to a one-dimensional array, and the y labels are added to create a dataframe for each class.

Data Splitting:

- The dataset (`total_df`) is split into features (`X`) and the target variable (`y`).
- Features are obtained by indexing to select all columns except the last one, while the target variable is selected as the last column.

Data Normalization and Dimensionality Reduction:

- A `Pipeline` named `data_pipe` is created to sequentially apply two transformations: `StandardScaler` for feature scaling and `PCA` for dimensionality reduction.
- `StandardScaler` standardizes features by removing the mean and scaling to unit variance to ensure uniform feature scales.
- `PCA` reduces the dimensionality of the feature space while retaining a significant portion of the original information, aiming to explain 90% of the variance in the data.
- The `fit` method is invoked on `data_pipe` with training data (`X_train`) to learn parameter values (mean, standard deviation for scaling, and principal components for PCA) from the training set.

Transforming Data:

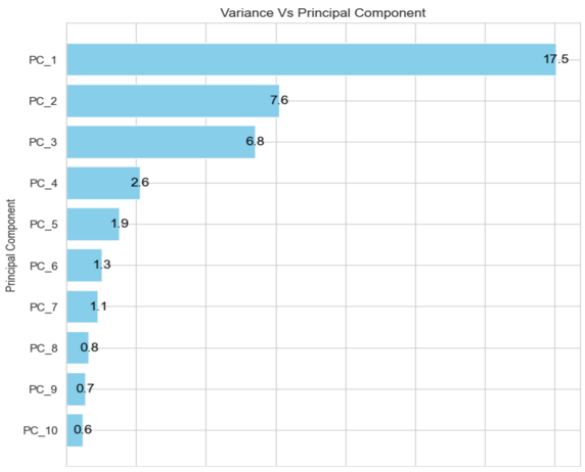

Figure 3: PCA

- After fitting the pipeline with training data, the `transform` method is used to transform both training (`X_train`) and testing (`X_test`) datasets based on learned parameters, resulting in a new space with reduced dimensionality.
- The number of features is reduced from 10801 to 622 after PCA.

Visualization:

- A visualization is generated to display the variance explained by the top 10 principal components (`PC_1` to `PC_10`).
- A horizontal bar chart illustrates the percentage of variance explained by each principal component(Figure 3)
- This visualization provides insight into the contribution of each principal component to the dataset's variance.

We have preprocess the data by standardizing features and reducing dimensionality using PCA while retaining most of the variability present in the original dataset. This prepares the data for subsequent modeling tasks and facilitates better understanding of the dataset's structure.

Below plots show the distribution of the data across classes:

.

5 Learning Models

These steps collectively prepare the dataset for training and evaluating deep learning models aimed at classifying Alzheimer's disease using MRI images of the brain

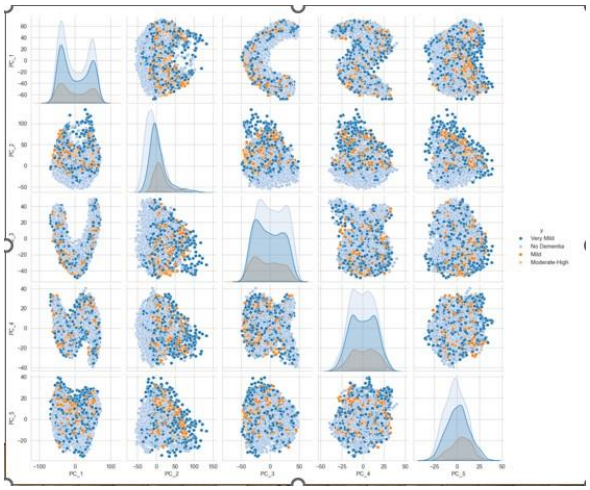

Figure 4: Peer Plotting

5.1 Logistic Regression

In our project, we have used Logistic regression as one of the classification algorithms to model the data and used multi-classification as we have more than one class of the data. Logistic regression predicts the probabilities of each possible class. The model outputs a vector of probabilities, one for each class. To ensure that the predicted probabilities sum up to 1, the Softmax function is used. It exponentiates each raw output and then normalizes the results. It transforms the raw outputs of the model into probabilities.To ensure that the predicted probabilities sum up to 1, the Softmax function is used. It exponentiates each raw output and then normalizes the results. It transforms the raw outputs of the model into probabilities.

We work with MRI images to identify patterns associated with AD. Feature selection techniques are employed to reduce dimensionality and enhance classification performance. We set up a Logistic regression model with specified parameters . It then defines a grid of hyperparameters, for hyperparameter tuning using grid search cross-validation. In training, the model is optimized using a loss function such as cross-entropy loss with maximum likelihood of the correct class label given the input features.

We have used ’l2’ (ridge) regularization terms to penalize the large coefficients and fine tuned the regularization parameter for a set of iterations to prevent overfitting. Grid search cross-validation, and various evaluation metrics tools are used to verify the performance of various combinations of the hyper parameters in the model. Regularization parameter value of 0.001 and the Newton-CG optimization algorithm are used to find the appropriate coefficients of the model to arrive at the better accuracy of 79

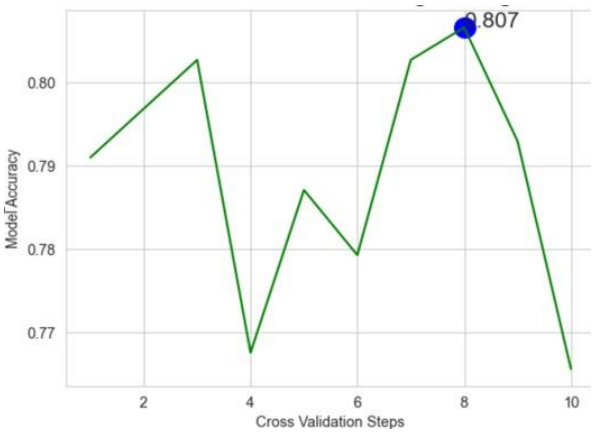

Figure 5: Cross validation accuracy for Logistic Regression

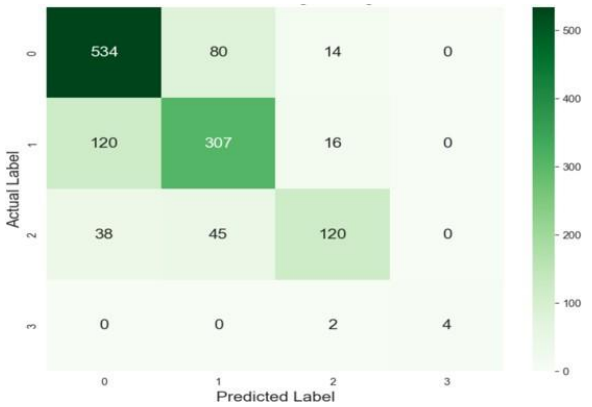

Figure 6: Confusion Matrix for Logistic Regression

From the confusion matrix and other evaluation metrics , we see that Class 0 and Class 3 have been relatively well-classified, with high precision, recall, and F1-score while Class 1 and Class 2 show moderate performance. Below are the graphs for Cross validation and Confusion Matrix for the logistic regression

.

5.2 Random Forest

The Random Forest algorithm is a popular machine learning technique used for classification tasks. Random forest is based on the concept of decision trees. A decision tree is a method in which the internal nodes represent the “question logic” and the node branches based on the decisions it can take, thereby reaching the leaf node that denotes the class label/prediction.

Random forest takes a random subset of training

	precision	recall	f1-score	support
0	0.77	0.85	0.81	628
1	0.71	0.69	0.70	443
2	0.79	0.59	0.68	203
3	1.00	0.67	0.80	6
accuracy			0.75	1280
macro avg	0.82	0.70	0.75	1280
weighted avg	0.75	0.75	0.75	1280

Figure 7: Logistic regression test results

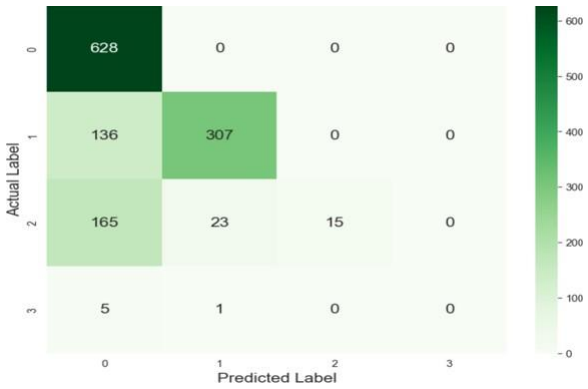

Figure 8: Confusion Matrix for Random Forest

data each time and builds multiple decision trees.At each node of the decision tree, only a random subset of features is considered for splitting. Random forest prediction is based on the number of decision trees which predict the same class.

The two main hyper parameters of the algorithm are the number of trees in the forest and the number of features that we choose to extract. We have tried a number of combinations to identify the efficient parameters for our dataset.

The square root of the number of features and 1000 decision trees finds the optimal balance between model performance and computational efficiency which is around 74.8

We plotted the confusion matrix to visualize the impact of the model on the dataset. The model performs reasonably well for class 0, with relatively high precision, recall, and F1-score. However, it performs less effectively for class 1, with lower precision, recall, and F1-score compared to class 0. Class 1 seems to have been mislabeled more compared to other classes. Classes 2 and 3 also show varying levels of performance, with relatively lower support and precision.

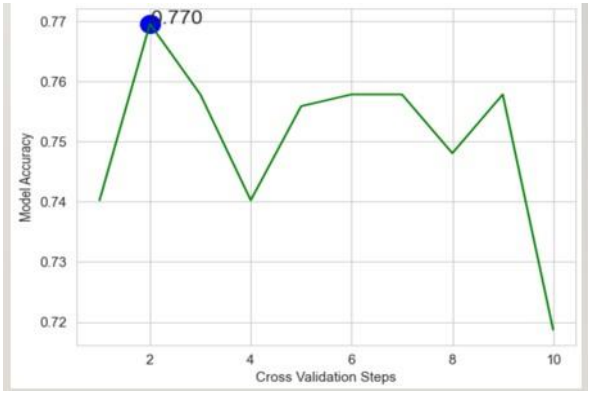

Figure 9: Cross validation accuracy for random forest

	precision	recall	f1-score	support
0	0.67	1.00	0.80	628
1	0.93	0.69	0.79	443
2	1.00	0.07	0.14	203
3	0.00	0.00	0.00	6
accuracy			0.74	1280
macro avg	0.65	0.44	0.43	1280
weighted avg	0.81	0.74	0.69	1280

Figure 10: Test Results for Random Forest

Figure 11: Cross vaidation accuracy for support vector machine

	precision	recall	f1-score	support
0	0.98	0.99	0.99	628
1	0.99	0.98	0.98	443
2	1.00	0.97	0.98	203
3	1.00	1.00	1.00	6
accuracy			0.98	1280
macro avg	0.99	0.98	0.99	1280
weighted avg	0.98	0.98	0.98	1280

Figure 12: Test results for SVM

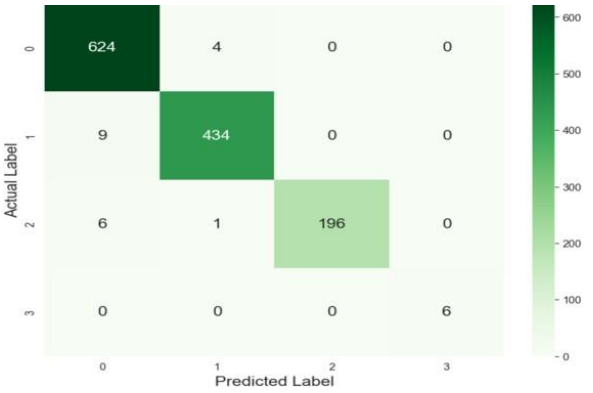

Figure 13: Confusion Matrix for SVM

5.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for classification or regression tasks. SVM works by finding the hyperplane that best separates the classes in the feature space. It aims to maximize the margin between the hyperplane and the nearest data points of any class.

After conducting a rigorous grid search on our Support Vector Machine (SVM) model, we identified the optimal hyperparameters that yielded exceptional performance on our dataset. The combination of a regularization parameter (C) set to 5 and a radial basis function (RBF) kernel emerged as the most effective configuration. This outcome was determined through an exploration of various parameter values, including different C values ranging from 0.5 to 10 and two kernel options, RBF and polynomial.

With this finely tuned SVM model, we achieved remarkable results, boasting an accuracy score of 98 percent on the test dataset. Moreover, the precision, recall, and F1-score metrics for each class exhibit impressive values, underscoring the model's robustness across multiple evaluation criteria. Notably, our model demonstrates exceptional performance in distinguishing between the classes, as evidenced by the high precision and recall scores across all categories.

Overall, the meticulous optimization process culminated in the identification of hyperparameters that not only maximize predictive accuracy but also ensure reliable performance across various metrics. By leveraging these best parameters, our SVM model stands poised to deliver reliable and accurate predictions for real-world applications, affirming its efficacy in addressing classification tasks with high-dimensional data.

5.4 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are deep learning models specifically designed to process structured grid data such as images. CNNs consist of multiple layers of convolutional and pooling operations followed by fully connected layers. They are widely used in image classification tasks due to their ability to automatically learn hierarchical features from the input data.

In other words , cnn network nodes in one layer is connected to nodes in the next layer (not neccessarily all of them). It may contain some fully connected layers and the features learnt at each layer are non-linear combination of original features (using the method of kernel/filter) and then they go through the activation function for introducing non-linearity again so that they become differentiable at the time of optimization.

This is a 13 layer Neural network with layers and parameters follows:

Input layer - It does not perform any function

Model: "sequential_1"

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 128, 128, 3)	0
conv2d (Conv2D)	(None, 128, 128, 16)	448
max_pooling2d (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_1 (Conv2D)	(None, 64, 64, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_1 (Dropout)	(None, 16, 16, 64)	0
flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 128)	2,097,280
dense_1 (Dense)	(None, 32)	4,128
dense_2 (Dense)	(None, 4)	132

Figure 14: Model Sequential-1

on the input, it only acts as an interface and just passes the input to the first hidden layer through the weighted edge.

Rescaling - We rescale images to range [0,1] by dividing by 255 for efficient computation and consistency among features

Convolution layer - In this layer, we have used 3 x 3 kernel to extract feature maps with same padding to maintaine the size of the input image and avoid edge errors. RELU - Rectifyting linear unit used for introducing non linearity (outputs 0 if z < 0 and outputs z if z > 0¿) to make it differentiable during optmization.

Max pooling layer - it is used to reduce the spatial dimensions of the features for efficient computation

Drop out layer - Introduces generalization and avoids overfitting by dropping a fraction input units randomly

Flatten layer - Flattens the input tensor into a 1D tensor. This prepares the data for input into the fully connected dense layers. It is done when we are almost ready to feed the data to output layer and stop convolution/forming new feature maps

Dense layer - The flattened input is given to a series of fully connected dense layers and then the final output layer uses the softmax function which calculates the probabilities of the classes (such that they sum up to 1) and the class with higher probability is the predicted class it belongs to.

Model Evaluation:

The figure 17 shows the decrease in the loss function as we move through the iterations and the

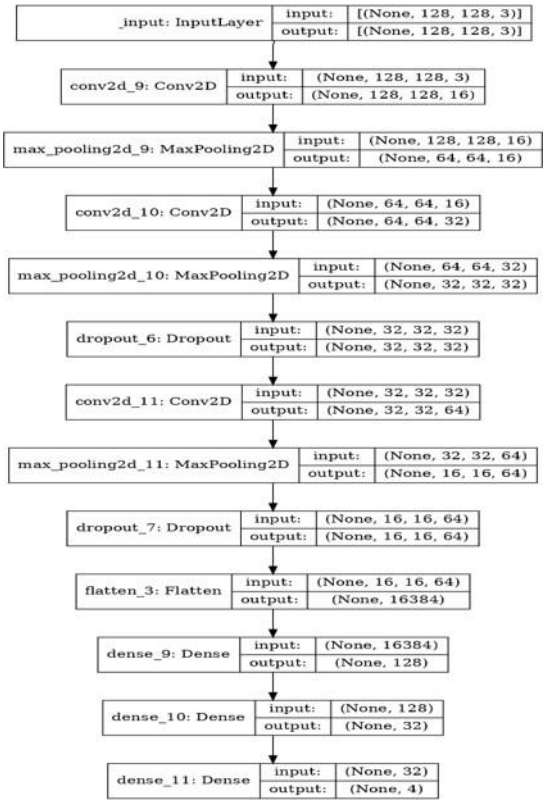

Figure 15: CNN Layers

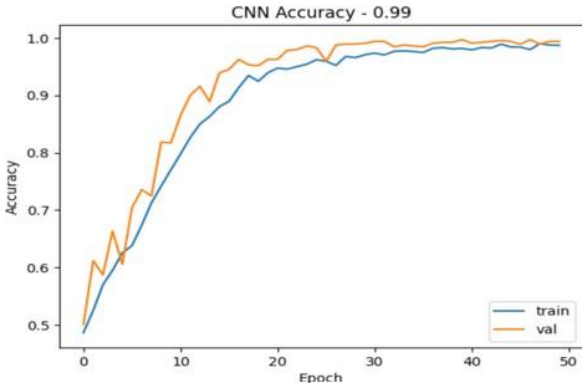

Figure 16: CNN accuracy

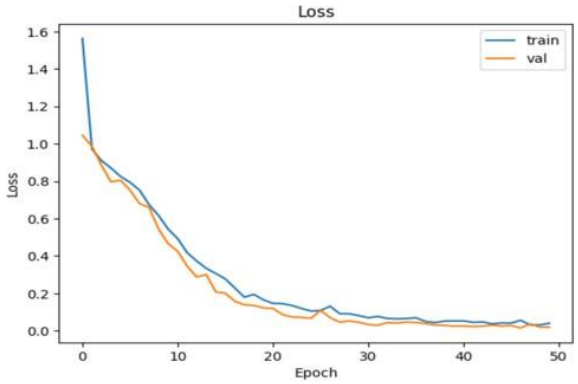

Figure 17: CNN Loss

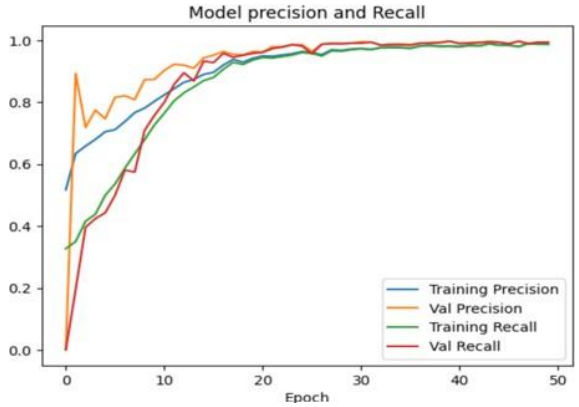

Figure 18: CNN model

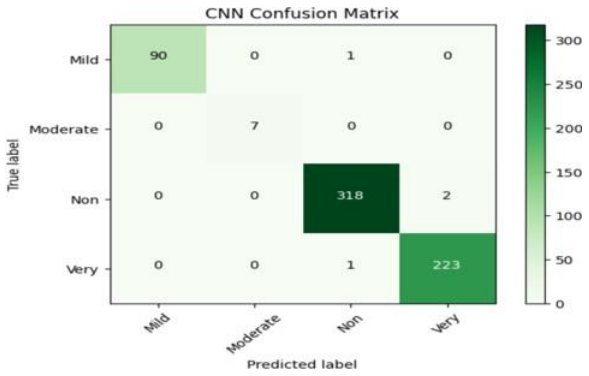

Figure 19: CNN confusion Matrix

figure 16 shows the gradual increase in the accuracy of the training and validation set with the epoch value. We have used early stopping to detect the saturation f the accuracy and stopped the iteration around 50. The figure 18 shows the high performance of the model on training and validation sets in terms of precision and recall.This shows that the model performs well in terms of detecting the positive oucome with highprecision and High recall indicates that the model can successfully capture a large proportion of positive instances in the dataset.

The confusion matrix 19 shows there is very less confusion among the classes for the model and hence the diagonals show the correct classfication and the off diagonal values are very less. Thus the model provides a reliable performance with a test accuracy of 99.22

6 Results

Figure 20 shows the comparison of the models. Logistic Regression performs moderately in terms of accuracy, with a high precision score, indicating that it tends to make fewer false positive predictions. However, its recall score suggests that it

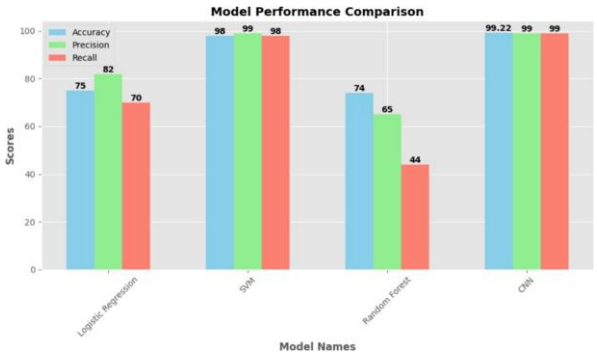

Figure 20: Model comparision

might miss some positive instances. SVM performs exceptionally well in terms of accuracy, precision, and recall. It effectively classifies both positive and negative instances with very few misclassifications.Random Forest achieves a lower accuracy compared to SVM but shows a lower precision and recall. It struggles more with correctly identifying positive instances, leading to more false positives and false negatives.

CNN achieves the highest accuracy of 99.2 among the models with high precision and recall scores.

Overall, SVM and CNN stand out as the top-performing models, with SVM performing exceptionally well across all metrics and CNN exhibitin g the highest accuracy. Logistic Regression and Ran- dom Forest perform relatively worse, with Random Forest showing particular challenges in correctly identifying positive instances.

7 Conclusions and Future Work

The primary aim of this study is to discern subtle cognitive function changes indicative of Alzheimer's Disease (AD) severity, aiding healthcare providers in informed patient care and early treatment decisions

In this study, we propose an extensive investigation of various machine learning models, including Random Forest, Logistic Regression, Support Vector Machine, and Convolutional Neural Networks (CNNs). We aim to optimize model performance by exploring different hyperparameters, identifying configurations yielding superior classification accuracy and robustness.

By meticulously fine-tuning hyperparameters, we achieved improved model performances:

- Random Forest: Utilized ”sqrt” feature and 1000 trees.
- Logistic Regression: Set regularization hyperparameter to 0.001.
- Support Vector Machine: Tuned hyperparameter to 5.
- CNN: Employed a 13-layer neural network with 3 convolution layers using Adam optimizer (Learning Rate = 0.001) with ReLU activation.

Preliminary experiments yielded the following results:

Model	**Training Accuracy (%)**	
Random Forest	75.4	71
Logistic Regression	78.1	75
Support Vector Machine	99.4	99
CNN	98.02	99

Table 2: Model Performance Before Hyperparameter Tuning

Updated model performances are as follows:

Model	**Training Accuracy (%)**	**Testing Accuracy (%)**
Random Forest	77	74
Logistic Regression	80	75
Support Vector Machine	99.8	98
CNN	98.5	99.22

Table 3: Model Performance After Hyperparameter Tuning

In future work, we plan to delve deeper into intreatability methods and explore ensemble technique’s to further enhance model performance and provide deeper insights into disease progression by Acquiring larger and more diverse datasets that can help improve the performance and generalizability. It would be useful to

integrate with clinical data by Combining MRI data with clinical information, such as cognitive test scores, medical history, genetic data, or lifestyle factors which can lead to a more comprehensive and accurate prediction model

8 References

- A deep learning model for Alzheimer’s disease diagnosis based on patient clinical records (**?**)
- Predicting Alzheimer’s Disease Progression through Machine Learning Algorithms (**?**)
- A deep CNN-based multi-class classification of Alzheimer’s disease using MRI (**?**)
- Comparative study of detection and classification of Alzheimer’s disease using Hybrid
- Early detection of Alzheimer’s disease with blood plasma proteins using support vector machines (**?**)
- Eke, Chima S., et al. "Early detection of Alzheimer's disease with blood plasma proteins using support vector machines." IEEE journal of biomedical and health informatics 25.1 (2020)