Multimodal Neuroimaging Based Alzheimer's Disease Diagnosis Using Evolutionary TRVFL Classifier

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Computer Science & Engineering

by
Prajapati Raman(20223177)
Piyush Tanay(20223175)
Priyanshu Singh(20223191)
Nikhil Jarwar(20223163)

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD PRAYAGRAJ
November, 2025

UNDERTAKING

I declare that the work presented in this report titled "*Multimodal Neuroimaging Based Alzheimer's Disease Diagnosis Using Evolutionary TRVFL Classifier*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

November, 2025
Allahabad

(Prajapati Raman (20223177)
Piyush Tanay (20223175)
Priyanshu Singh (20223191)
Nikhil Jarwal (20223163)
)

CERTIFICATE

Certified that the work contained in the report titled "*Multimodal Neuroimaging Based Alzheimer's Disease Diagnosis Using Evolutionary TRVFL Classifier*", by *Prajapati Raman (20223177)*
Piyush Tanay (20223175)
Priyanshu Singh (20223191)
Nikhil Jarwal (20223163), has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Prof. Deepak Gupta)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

November, 2025

Preface

The early detection of Alzheimer's Disease (AD) is crucial for timely intervention and improving patient outcomes. This B.Tech. project focuses on developing an effective framework for diagnosing Alzheimer's using multimodal neuroimaging data along with machine learning techniques. These techniques include ConvNeXtV2,Swin Classifier Random Vector Functional Link (RVFL), TRVFL, and ensemble-based methods.

The main goal of this work is to create models that can accurately classify AD, Mild Cognitive Impairment (MCI), and cognitively normal (CN) subjects. The project highlights careful tuning of hyperparameters, the use of ensemble strategies, and a comparative evaluation of deep learning and functional link-based models to achieve the best results.

This thesis shows how theoretical knowledge connects with practical experimentation in the field of computational neuroscience. It also emphasizes the value of using multimodal data to improve diagnostic accuracy and the potential of machine learning models to help clinicians make decisions.

We hope the methods and findings in this work will lay the groundwork for more research in Alzheimer's detection and inspire future advancements in medical imaging and predictive modeling.

We sincerely thank Prof. Deepak Gupta for his guidance and support, along with our peers for their valuable feedback, which has greatly improved the quality of this work.

Acknowledgements

We would like to express our sincere gratitude to everyone who supported and guided us throughout this project.

First and foremost, we extend our heartfelt thanks to Prof. Deepak Gupta, our supervisor, for his continuous guidance, encouragement, and valuable feedback. His expertise and insightful suggestions greatly enhanced the quality of our research and helped us stay on the right path.

We are also thankful to our project mentors and the faculty members of the Department of Computer Science and Engineering for providing us with essential resources, technical support, and meaningful advice during various stages of the project.

Our sincere thanks go to our team members and peers for maintaining a co-operative and motivating environment. Their constructive feedback, discussions, and teamwork played a crucial role in refining our methodology and achieving our objectives.

Finally, we acknowledge that this project would not have been possible without the collective support, guidance, and encouragement of all these individuals.

Contents

Chapter 1

Introduction

Alzheimer's Disease (AD) is a neurodegenerative disorder characterized by progressive memory impairment, cognitive decline, and an individual's inability to execute the activities of daily living. It is important to identify AD early because effective intervention can delay the progression of the disease and increase quality of life for patients and their families. Recent advances in machine learning and deep learning provide the capability for automated analysis of neuroimaging data, which assists in the early and accurate identification of AD. This project demonstrates a multimodal neuroimaging-based AD diagnostic system based on the ADNI MRI dataset using ResNet-50 neural network feature extraction along with multiple other advanced classifiers RVFL and modified RVFLs (such as EDRVFL, TRVFL, Swift Classifier, and ConvNeXtV2).

1.1 Motivation

Alzheimer's Disease (AD) is a progressive neurodegenerative condition that steadily impacts memory, cognition, and behavioral functioning. The increasing elderly population globally increases the prevalence of Alzheimer's Disease at a rapidly growing rate, making it a significant public health issue. According to recent clinical studies, there are millions of people around the globe with AD, and this will increase dramatically in coming decades. Despite significant research efforts, AD does not have a cure, which underscores the significant importance of early and accurate diagnosis. Early diagnosis allows for timely intervention, slows the progression of the disease, and improves the quality of life for the patient.

Magnetic Resonance Imaging has become one of the best non-invasive techniques available to assess structural brain alterations associated with AD; however, the reading of MRIs relies on expert knowledge in radiology and is often based on a highly subjective and time-consuming analysis that also suffers from inter-observer variability. Traditional machine-learning approaches have offered partial solutions, but there remain challenges associated with capturing the complexity of patterns associated with neuroimaging.

Despite such advancements, the unification of deep feature extraction with fast, robust, and generalizable classifiers still remains an open challenge. Specifically, evolutionary variants of TRVFL can be further optimized to improve generalization and reduce overfitting, especially in complex neuroimaging tasks. There is, therefore, a strong motivation to develop a comprehensive multimodal framework that will unify ResNet-50-based feature extraction with advanced evolutionary TRVFL classifiers in the interest of reliable diagnosis of Alzheimer's Disease using the ADNI MRI dataset.

These findings motivate the present work, which aims at bridging existing gaps through using multimodal neuroimaging, advanced deep-learning feature extraction, and evolutionary classification techniques to be able to reach a more efficient, accurate, and automated AD diagnosis system.

1.2 Problem Statement

The difficulty in diagnosing Alzheimer's Disease from MRI scans lies in the fact that there are subtle structural brain changes, high-dimensional data, and a high reliance on subjective expert interpretation. Most of the traditional models in machine learning have poor generalization performance when applied to complex neuroimaging datasets. Though deep learning techniques like ResNet-50 extract meaningful features, choosing an efficient classifier with good accuracy is still a main issue. Thus, this project tries to overcome these shortcomings by developing a robust and reliable framework combining deep feature extraction with deep learning and machine learning models.

1.3 Objectives

The main objectives of this project are as follows:

- To create an automatic framework for diagnosing Alzheimer's Disease using MRI data from the ADNI dataset.
- To extract detailed and distinct features from MRI scans, use the ResNet-50 architecture.
- To implement and evaluate multiple classifiers, including RVFL, EDRVFL, TRVFL, Evolutionary TRVFL, Swift Classifier, and ConvNeXtV2. To perform hyperparameter tuning for all models to achieve the best performance and ensure a fair comparison.
- To compare the performance of all models, we will find the best classifier for AD detection. We will use various evaluation metrics, including accuracy, precision, recall, F1-score, specificity, sensitivity, ROC-AUC, and confusion matrix analysis.

Chapter 2

Literature Review and Related Work

2.1 Multimodal Neuroimaging with Evolutionary RVFL (Goel et al., 2023)

Goel et al. [2] proposed an Evolutionary RVFL (E-RVFL) classifier for early AD diagnosis using multimodal MRI and PET data. They combined deep and handcrafted features from both types of scans to capture structural and metabolic brain changes. The RVFL parameters were improved using an evolutionary strategy, which made the model more robust, reduced its dependence on random initialization, and improved its generalization. Experiments on ADNI showed that E-RVFL achieved higher accuracy and better overall performance compared to traditional RVFL, SVM, and deep CNN baselines.

2.2 Conv-eRVFL: CNN-Enhanced RVFL for AD Detection (Sharma et al., 2023)

Sharma et al. [12] introduced Conv-eRVFL, a hybrid model combining CNN-based feature extraction with an ensemble RVFL classifier for multimodal (MRI + PET) AD diagnosis. CNN features captured rich spatial patterns, while the ensemble RVFL improved stability and decision robustness. Using the ADNI dataset, Conv-eRVFL outperformed traditional RVFL, standalone CNNs, and other baselines across accuracy, sensitivity, and specificity, demonstrating the strength of merging deep features with ensemble RVFL learning.

2.3 Biomarker Models

Early work on Alzheimer's disease (AD) focused on understanding biomarker progression. Jack et al. [5] proposed a dynamic model describing the sequence of pathological changes in AD. The ADNI study [4] further standardized MRI acquisition and established a widely used benchmark dataset.

2.4 Classical ML Approaches

Traditional machine learning techniques relied on handcrafted MRI features. Fan et al. [1] demonstrated that feature-selected structural MRI measurements combined with SVMs can effectively classify AD and MCI.

2.5 Deep Learning Methods

Deep learning significantly improved MRI feature representation. Suk et al. [13] introduced hierarchical deep feature learning with multimodal fusion. Payan and Montana [9] used 3D CNNs to directly learn from volumetric MRI.

Liu et al. [6] proposed landmark-based deep multi-instance learning for more discriminative spatial pattern extraction.

2.6 MRI Preprocessing

Proper MRI preprocessing enhances the reliability of longitudinal studies. Reuter et al. [10] presented a robust within-subject template estimation method using FreeSurfer, improving consistency in downstream classification.

2.7 Summary

Overall, research has shifted from handcrafted structural features to end-to-end deep learning frameworks, supported by standardized datasets like ADNI and strong preprocessing pipelines.

Chapter 3

Methodology and System Architecture

3.1 Methodological Foundation

The foundation of this project combines deep feature extraction and classification techniques to create a framework for diagnosing Alzheimer's Disease (AD) using MRI data from the ADNI dataset. The methodology includes neuroimaging preprocessing, deep feature extraction, and classification with RVFL-based models, along with modern deep-learning classifiers.

First, the MRI scans go through important preprocessing steps, such as skull stripping, intensity normalization, resizing, and noise reduction. These steps ensure consistency across the dataset and maintain crucial structural features relevant to AD diagnosis. After preprocessing, deep feature extraction occurs using the ResNet-50 architecture. This architecture captures high-level spatial and structural patterns from the MRI images through its residual learning framework. The deep features extracted form the input for various classifiers.

Random Vector Functional Link (RVFL) networks are one of the main classification methods because of their fast learning ability and direct input-to-output connections. Improved variants like EDRVFL, TRVFL, and Evolutionary TRVFL are also used to boost model stability, generalization, and performance. Evolutionary TRVFL uses population-based optimization strategies to improve weights, biases, and regularization parameters. This leads to more stable decision boundaries and better classification results.

Along with RVFL-based models, the methodology includes modern deep-learning classifiers like the Swift Classifier and ConvNeXt V2. The Swift Classifier provides quick inference and a lightweight design suitable for medical imaging tasks. In contrast, ConvNeXt V2 offers better hierarchical feature learning with a next-generation convolutional design. These models work alongside the RVFL methods and allow for a thorough comparative analysis.

The performance of all models is assessed using a variety of metrics, including accuracy, precision, recall, F1-score, sensitivity, specificity, ROC-AUC, and confusion matrix analysis. These metrics help identify the most effective classifier and give a clear understanding of each model's strengths in detecting Alzheimer's Disease.

Overall, the foundation of this project merges deep neural feature extraction, evolutionary learning, classical machine-learning models, and modern deep-learning architectures. This creates a robust and scalable framework for accurately diagnosing Alzheimer's Disease using MRI data.

3.1.1 Preprocessing

The dataset in this study comes from the Alzheimer's Disease Neuroimaging Initiative (ADNI). It includes MRI scans divided into three main categories: Alzheimer's Disease (AD), Cognitively Normal (CN), and Mild Cognitive Impairment (MCI). This varied class distribution allows for a thorough evaluation of the proposed models at different stages of neurodegeneration.

1. **Feature Extraction** Feature extraction in this project uses the ResNet-50 deep convolutional neural network, known for its strong feature representation abilities. ResNet-50 has a residual learning framework with 50 layers, which helps capture both low-level and high-level structural patterns in MRI images. The residual connections reduce the vanishing gradient problem, allowing the network to learn deeper and more specific features.

   In this work, the preprocessed MRI scans go through the pretrained ResNet-50 model, excluding the final classification layers, to extract deep feature vectors. These features provide solid input representations for the next RVFL-based models, the Swift Classifier, and ConvNeXt V2. By using ResNet-50, the extracted features effectively include important spatial, textural, and anatomical information that helps differentiate between AD, MCI, and CN classes.

2. **Feature Matrix Cleaning, Imputation, and One-Hot Encoding:** After extracting features with ResNet-50, the feature matrix is cleaned by converting all feature columns into numeric format. Invalid or non-numeric values are turned into NaN. Infinite values and zeros are also treated as missing data to avoid skewing the learning process. Columns that only contain NaN values are removed. The remaining missing values are filled in using the median of each feature. Additionally, categorical attributes like class labels are processed with one-hot encoding to convert them into machine-readable binary vectors. This makes sure that all features are numeric, consistent, and properly formatted for downstream classifiers.

3.2 Model Architecture

3.2.1 RVFL Architecture:

The Random Vector Functional Link (RVFL) model used in this study has a direct input-to-output connection and a randomly generated hidden layer. The structure includes three main parts: (i) a direct linear layer that maps the input features to the output, (ii) a hidden layer with randomly assigned weights and a sigmoid activation function, and (iii) a hidden-to-output layer that transforms the activated hidden features. During the forward pass, the hidden features are combined with the original input to create a better feature representation. The final prediction comes from adding the outputs of the direct and hidden pathways. This design allows for fast training, better generalization, and effective handling of high-dimensional feature vectors.

3.2.2 EDRVFL Architecture:

The EDRVFL model has several enhancement layers. Each layer consists of randomly set weights and biases. Every layer standardizes the input, applies a nonlinear activation function (like sigmoid, sine, hardlim, tribas, radbas, ReLU, or leaky ReLU), and combines hidden features with the original input. A bias term is added, and the output weights for each layer are calculated using ridge regression. Instead of using a single output, EDRVFL combines predictions from all layers through (i) majority voting and (ii) additive softmax-based probability fusion. This group of deep random layers improves robustness, lowers overfitting, and allows for strong performance even with limited training data.

3.2.3 TRVFL Architecture:

The TRVFL model adopts a two-stage Random Vector Functional Link (RVFL) architecture enhanced with an ℓ_1-norm based quadratic programming framework. First, random weights and biases generate L nonlinear hidden features through sigmoid activation, which are concatenated with the original input to form an expanded feature representation. For each class, the model constructs a one-vs-rest binary learning setup by splitting samples into positive and negative subsets. Two quadratic programming problems with box constraints are then solved to obtain dual variables for both subsets, ensuring robust class separation. The primal output weights (β_1, β_2) are recovered using regularized inverses of $H^T H$ and $G^T G$, following the original MATLAB formulation. During inference, the enhanced feature matrix is multiplied with these learned weights to compute decision values, and the binary decision rule assigns the positive class when $|y_1| < |y_2|$. Extending this mechanism to multi-class classification, the model trains K independent one-vs-rest submodels, each producing a score based on the magnitude of $|I\beta_1|$. The final predicted label corresponds to the class yielding the minimum score, representing the closest match to its learned manifold. This architecture provides a computationally simple yet effective framework for multi-class neuroimaging classification.

3.2.4 ConvNeXtV2 Feature-Aware Classifier Architecture:

To use the high-dimensional ResNet-50 feature vectors in a convolutional architecture, a feature grid projection strategy is applied. The original feature vector is reshaped into a 2D grid by calculating an optimal height and width decomposition. If the vector length is not perfectly divisible, zero-padding is used to maintain the spatial structure. The reshaped input is treated as a single-channel image and is passed through the ConvNeXtV2 backbone.

ConvNeXtV2 has four hierarchical stages. Each stage starts with a downsampling layer, followed by several depthwise convolutional blocks. Each block includes: (1) a depthwise convolution, (2) Layer Normalization, (3) pointwise linear expansions, (4) a GELU activation, (5) a Global Response Normalization (GRN) module, and (6) a final pointwise projection. To enhance generalization, stochastic depth using the DropPath operator is applied across blocks.

The output from the final stage is globally averaged and normalized before being sent to a fully connected classification head. The classifier is optimized with the BCEWithLogits loss, using an AdamW optimizer and softmax-based probability inference. This setup allows ConvNeXtV2 to learn spatially coherent patterns from reshaped feature embeddings, achieving high performance even without raw image inputs.

3.2.5 Swin Transformer V2 Feature-Aware Classifier Architecture:

To change a high-dimensional feature vector into a structure for hierarchical attention, the input feature embedding is divided into a fixed number of tokens arranged on a 2D grid. Let the vector length be F and the number of tokens be $T = H \times W$, where T is a perfect square. Each chunk of size F/T is projected into an embedding space using a linear patch-projection layer. This creates a $(H \times W)$ token map.

This token grid goes through a two-stage Swin Transformer V2 hierarchy. Each stage has stacked Swin Transformer blocks that alternate between standard windowed multi-head self-attention (W-MSA) and shifted window attention (SW-MSA). Swin V2 improvements, like cosine attention scaling, continuous relative position bias through an MLP, logit-scale stabilization, and better normalization, are included to improve training stability on small windows.

Within each block, LayerNorm is applied before attention. This is followed by residual connections, stochastic depth (DropPath), and a GELU-activated

MLP expansion layer. Patch Merging is used between stages to decrease spatial resolution while doubling channel dimensionality, allowing a hierarchical encoder similar to CNN downsampling.

After the final stage, features are normalized and averaged globally across tokens. A fully connected classification head generates the output logits for the three target classes. Training uses the AdamW optimizer with cross-entropy loss and softmax-based inference. This setup lets the Swin V2 architecture learn structured correlations within pre-extracted features, even without using raw image inputs.

3.3 Tuning

3.3.1 RVFL Hyperparameter Tuning Setup

To find the best setup for the Random Vector Functional Link (RVFL) network, a focused search was carried out over two key parameters: the number of hidden enhancement nodes and the regularization coefficient λ. The tuning aimed to look at how model capacity, through hidden neurons, and generalization control, through Tikhonov regularization, affect performance.

The hyperparameter space explored in this study consisted of:

- **Hidden nodes:** Multiple enhancement-layer sizes were tested (e.g., 256, 512, 768, 843).
- **Regularization parameter λ:** A logarithmically spaced range was evaluated to capture both weak and strong regularization regimes.

For each hyperparameter configuration, the RVFL model was trained on the training split and evaluated on the validation split. Both training and validation accuracies were recorded to analyze how regularization and hidden-layer width affect performance. A comprehensive performance visualization was generated, including:

- **Train accuracy vs.** $\log_{10}(\lambda)$ **curves** for different hidden-node settings.
- **Validation accuracy vs.** $\log_{10}(\lambda)$ **curves** to identify overfitting trends.
- **A heatmap of validation accuracy**, with hidden-node count on the vertical axis and $\log_{10}(\lambda)$ on the horizontal axis, providing a global view of search-space performance.

The Cartesian evaluation of hidden size and regularization strength allowed for a clear identification of the best-performing configuration. We later used the optimal hyperparameter pair to train the final RVFL model on the combined training and validation set. We then reported the performance on the held-out test set.

3.3.2 EDRVFL Hyperparameter Tuning Setup

To improve the Ensemble Deep Random Vector Functional Link (EDRVFL) classifier, we conducted a focused hyperparameter search on the most important architectural and regularization parameters. The EDRVFL model builds on the traditional RVFL network by adding multiple enhancement layers and combining their predictions using both voting and addition-based ensemble strategies. Therefore, choosing the right values for the number of enhancement nodes, the depth of stacked layers, and the regularization strength is crucial for stable and effective classification.

The hyperparameter space explored consisted of the following components:

- **Number of enhancement nodes** (n_{nodes}): Determines the dimensionality of the feature expansion in each RVFL layer.
- **Regularization coefficient** (λ): Controls the magnitude of the ridge regression penalty during output-weight computation.
- **Number of stacked layers** (n_{layer}): Specifies the depth of the EDRVFL architecture, enabling multi-level feature transformation.
- **Activation function**: The nonlinear activation applied in enhancement layers (ReLU in this study).

Each possible parameter configuration was evaluated by training the EDRVFL on the training split and measuring performance on the validation split using both ensemble metrics:

- **Voting accuracy** — majority vote across layer outputs.
- **Addition accuracy** — aggregate probability across layers.

For each configuration, we recorded both training and validation accuracies. This created a detailed search log that we used later for visualization and model selection. The search looked at different regularization strengths while keeping the number of nodes, activation function, and depth the same for a controlled analysis. This setup allowed us to study how changes in the ridge penalty affect the stability and generalization of the ensemble's performance.

At the end of the search, we chose the configuration with the highest validation accuracy as the best EDRVFL setup. We then retrained the model using the combined training and validation set before reporting the performance on the test set.

3.3.3 TRVFL Hyperparameter Tuning

To optimize the performance of the TRVFL classifier, a systematic hyperparameter tuning procedure was employed. The tuning process focused primarily on two key parameters: the number of random hidden nodes (L_{hidden}) and the penalty coefficient (c_1) used in the ℓ_1-norm quadratic programming stage. For each candidate pair (L_{hidden}, c_1), a complete multi-class TRVFL model was trained using the one-vs-rest setup, followed by evaluation on the validation dataset. The validation accuracy was computed for each configuration, and the best-performing combination was selected based on the highest accuracy achieved. During this search, errors related to empty class splits or singular matrices were gracefully handled by skipping invalid parameter combinations. The tuning loop maintained records of the best accuracy, along with the corresponding hidden layer size and penalty parameter, ensuring that the optimal configuration was identified. This procedure allowed the model to strike an effective balance between representational capacity and regularization strength, ultimately improving classification performance.

3.3.4 ConvNeXtV2 Hyperparameter Tuning Setup

To optimize the ConvNeXtV2-based classifier, we designed a thorough grid search strategy that covered various architectural and training parameters. The search space included four model variants: atto, femto, pico, and nano. It also featured key parameters such as learning rate, batch size, drop-path rate, head initialization scale, weight decay, and training epochs.

The complete hyperparameter grid included these ranges:

- Model variant: atto, femto, pico, nano
- Learning rate: 1×10^{-4}, 3×10^{-4}, 1×10^{-3}
- Batch size: 16, 32, 64
- Drop-path rate: 0.0, 0.1, 0.2
- Head initialization scale: 0.5, 0.75, 1.0
- Weight decay: 1×10^{-4}, 5×10^{-4}
- Epochs: 20, 30, 40

Given the large search space, we also prepared a smaller grid for quicker experimentation. This reduced configuration centered on the femto and pico variants with more focused parameter ranges:

- Model variant: femto, pico
- Learning rate: 3×10^{-4}, 1×10^{-3}
- Batch size: 32, 64
- Drop-path rate: 0.0, 0.1
- Head initialization scale: 0.75

- Weight decay: 1×10^{-4}
- Epochs: 30

We generated each hyperparameter configuration using a Cartesian product over the selected grid. This approach allowed systematic exploration while keeping computational demands manageable. This tuning process helped us select the best ConvNeXtV2 variant and its training setup.

3.3.5 Swin Classifier Hyperparameter Tuning

To improve the performance of the Swin Transformer-based classifier, a random search strategy was used across a multi-dimensional hyperparameter space. The search considered several factors, including learning rate, weight decay, embedding dimension, hierarchical depth, number of attention heads, and drop-path regularization probability. The tuning space included:

$$lr \in \{10^{-4}, 5 \times 10^{-4}, 10^{-3}\}, \quad weightdecay \in \{10^{-3}, 10^{-2}, 5 \times 10^{-2}\},$$

$$embeddim \in \{64, 128, 256\}, \quad depths \in \{(2,2), (2,4), (4,4)\},$$

$$numheads \in \{(4,8), (8,16), (4,4)\}, \quad drop - pathrate \in \{0.05, 0.1, 0.2\}.$$

Given the large configuration space, a fixed-budget random search with five trials was performed. For each sampled configuration, the classifier was trained for 25 epochs using the AdamW optimizer and one-hot encoded labels, applying Binary Cross-Entropy Loss on the logits. The model's performance was assessed through validation accuracy, with both configuration results and the best-performing model recorded.

During each trial, the SwinClassifier was created with the sampled hyperparameters:

$$Config = \{lr, weightdecay, embeddim, depths, heads, drop - path\},$$

and optimized end-to-end on the extracted deep features. Validation accuracy acted as the fitness measure, allowing for comparison across configurations.

After all trials were completed, the random search found the best configuration as the one with the highest validation accuracy. This configuration and its trained model weights were kept for final testing. The tuning results show the usefulness of random search for high-dimensional transformer-based architectures, highlighting effective combinations of embedding size, depth, and regularization.

Chapter 4

Experimental Setup and Results Analysis

4.1 Data Set Visualization

Figure 1 illustrates the distribution of labels across the training, validation, and test subsets of the Alzheimer's disease dataset. The dataset contains three diagnostic classes: Alzheimer's Disease (AD), Mild Cognitive Impairment (MCI), and Cognitively Normal (CN). The training set has 1,081 AD, 866 MCI, and 781 CN samples, while the validation set contains 231 AD, 186 MCI, and 168 CN samples. The test set has 232 AD, 186 MCI, and 167 CN samples. The bar chart clearly shows that the AD class has the highest number of samples across all splits, whereas CN has the fewest. Maintaining a similar distribution across all subsets is important for reliable model training and evaluation.

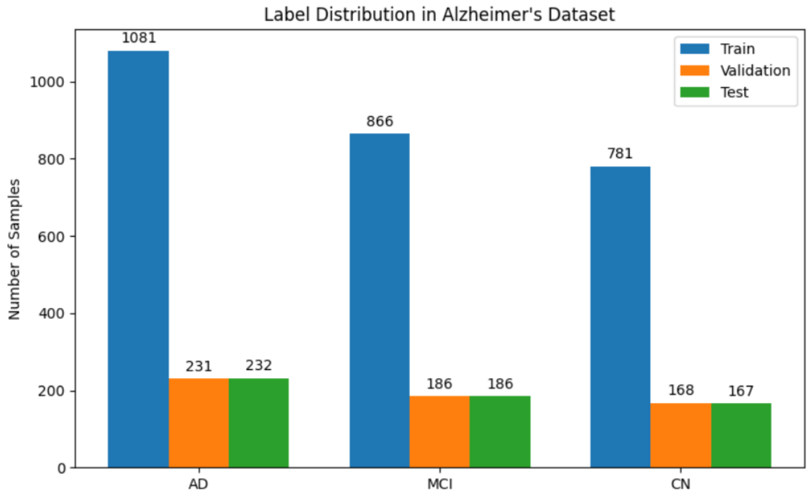


Figure 1: Label distribution in the training, validation, and test sets of the Alzheimer's dataset.

Table 1: Label distribution across the training, validation, and test sets of the Alzheimer's dataset.

Class	**Train**	**Validation**	**Test**
AD	1081	231	232
MCI	866	186	186
CN	781	168	167

4.2 Performance Evaluation

4.2.1 RVFL Model Performance Evaluation

The Random Vector Functional Link (RVFL) model was evaluated on the dataset with various hyperparameter configurations, including different numbers of hidden nodes (H) and regularization parameters (λ). The goal was to optimize validation accuracy and obtain the best performing model for the test set.

Hyperparameter Search

The hyperparameter search explored multiple configurations of hidden nodes and regularization parameters. Selected results are summarized below:

Hidden Nodes (H)	λ	**Train Acc**	**Val Acc**	**Test Acc**
10	0.1	1.0000	0.9453	0.9316
10	1	0.9996	0.9607	0.9556
20	1	1.0000	0.9624	0.9590
50	1	1.0000	0.9624	0.9590
100	1	1.0000	0.9607	0.9573
500	1	1.0000	0.9624	0.9624

Table 2: Selected results from RVFL hyperparameter search. The best configuration was Hidden Nodes=20, $\lambda = 1$.

The best configuration was found to be:

- Hidden Nodes: 20
- Regularization (λ): 1
- Train Accuracy: 1.0000
- Validation Accuracy: 0.9624

Final Test Performance

The RVFL model with the best configuration was evaluated on the test set, achieving:

Metric	**Value**
Final Test Accuracy	98.46%
Test Loss	0.6133

Table 3: Final test performance of the best RVFL model.

- *Evaluation Metrics (Test Set)*

Metric	**Value**
Accuracy	0.9846
Precision	0.9846
Recall	0.9846
F1-score	0.9846
Cohen's Kappa	0.9767
Matthews Correlation Coefficient (MCC)	0.9767

Table 4: Evaluation metrics for the RVFL model on the test set.

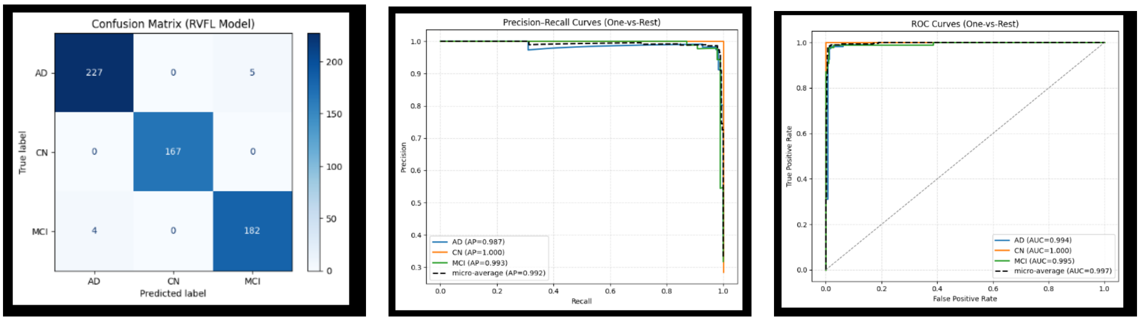
- *Classification Report (Test Set)*

Class	**Precision**	**Recall**	**F1-score**	**Support**
AD	0.9827	0.9784	0.9806	232
CN	1.0000	1.0000	1.0000	167
MCI	0.9733	0.9785	0.9759	186
Accuracy		0.9846		
Macro Avg	0.9853	0.9856	0.9855	585
Weighted Avg	0.9846	0.9846	0.9846	585

Table 5: Classification report for the RVFL model on the test set.

– Test Accuracy: 98.46%

– Test Loss: 0.6133



(a) Confusion Matrix (b) Precision-Recall Curve (c) ROC Curve

Figure 2: Performance evaluation plots for rvfl Modal.

4.2.2 Ensemble Deep RVFL (EDRVFL) Model Performance Evaluation

The Ensemble Deep RVFL (EDRVFL) model was trained and evaluated on the dataset using hyperparameter tuning to optimize performance. Two aggregation strategies were used: **Voting** and **Addition**.

- *Hyperparameter Search*

A hyperparameter search was performed to find the best combination of nodes, regularization parameter λ, layers, and activation function. The results for key configurations are summarized below:

Nodes	λ	Layers	Activation	Train Acc (vote/add)	Val Acc (vote/add)
10	1	10	relu	1.0000 / 1.0000	0.9282 / 0.9265
10	10	10	relu	1.0000 / 1.0000	0.9573 / 0.9573
10	20	10	relu	1.0000 / 1.0000	0.9590 / 0.9590
10	30	10	relu	0.9996 / 0.9996	0.9607 / 0.9607
10	40	10	relu	0.9996 / 0.9996	0.9607 / 0.9607
10	50	10	relu	0.9993 / 0.9993	0.9607 / 0.9607

Table 6: Selected results from EDRVFL hyperparameter search. The best configuration was Nodes=10, $\lambda = 30$, Layers=10, Activation=relu.

The best hyperparameter configuration achieved:

- Nodes: 10
- Lambda (λ): 30
- Layers: 10
- Activation: relu
- Train Accuracy: 0.9996 (vote/add)
- Validation Accuracy: 0.9607 (vote/add)

Final Test Performance

The best EDRVFL model was retrained on the combined train+validation data and evaluated on the test set. Both **Voting** and **Addition** aggregation methods achieved the same performance:

Metric	Voting	Addition
Test Accuracy	98.63%	98.63%

Table 7: Final test accuracy of the EDRVFL model using voting and addition strategies.

▮ *Evaluation Metrics (Test Set)*

Metric	Voting	Addition
Accuracy	0.9863	0.9863
Precision	0.9863	0.9863
Recall	0.9863	0.9863
F1-score	0.9863	0.9863
Cohen's Kappa	0.9793	0.9793
MCC	0.9793	0.9793

Table 8: Evaluation metrics for the EDRVFL model on the test set.

▮ *Classification Report (Test Set)*

Voting:

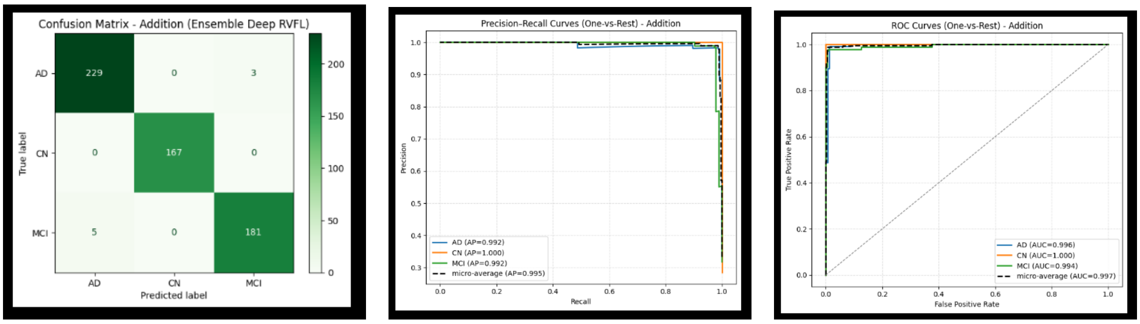
Class	Precision	Recall	F1-score	Support
AD	0.9786	0.9871	0.9828	232
CN	1.0000	1.0000	1.0000	167
MCI	0.9837	0.9731	0.9784	186
Accuracy		0.9863		
Macro Avg	0.9874	0.9867	0.9871	585
Weighted Avg	0.9863	0.9863	0.9863	585

Table 9: Classification report for the EDRVFL model using voting strategy.

Addition:

Class	Precision	Recall	F1-score	Support
AD	0.9786	0.9871	0.9828	232
CN	1.0000	1.0000	1.0000	167
MCI	0.9837	0.9731	0.9784	186
Accuracy		0.9863		
Macro Avg	0.9874	0.9867	0.9871	585
Weighted Avg	0.9863	0.9863	0.9863	585

Table 10: Classification report for the EDRVFL model using addition strategy.



(a) Confusion Matrix (b) Precision-Recall Curve (c) ROC Curve

Figure 3: Performance evaluation plots for edrvfl Modal.

4.2.3 TRVFL Model Performance Evaluation

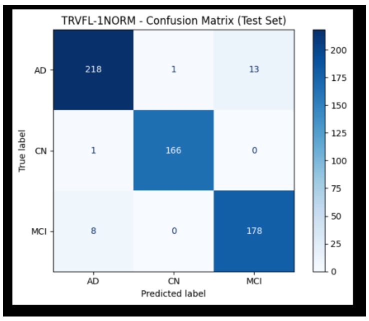
The TRVFL model was evaluated on the test set to assess its performance across multiple metrics. The model was saved and reloaded to ensure reproducibility, and identical results were obtained.

■ *Classification Report (Test Set)*

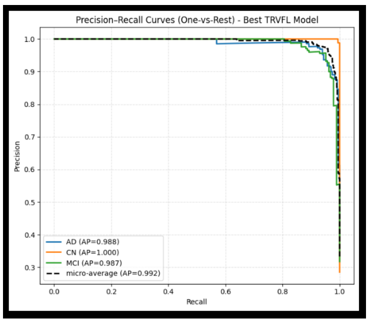
Class	Precision	Recall	F1-score	Support
AD	0.9604	0.9397	0.9499	232
CN	0.9940	0.9940	0.9940	167
MCI	0.9319	0.9570	0.9443	186
Accuracy		0.9607		
Macro Avg	0.9621	0.9636	0.9627	585
Weighted Avg	0.9609	0.9607	0.9607	585

Table 11: Classification report for the TRVFL model on the test set.

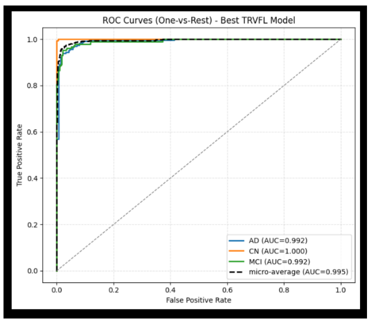
The TRVFL model demonstrates strong performance across all classes. Although slightly lower than previous results, the model still achieves high precision and recall for AD, CN, and MCI.



(a) Confusion Matrix



(b) Precision-Recall Curve



(c) ROC Curve

Figure 4: Performance evaluation plots for trvfl Modal.

4.2.4 ConvNeXt V2 Model Performance Evaluation

The ConvNeXt V2 model was evaluated on the dataset before and after hyperparameter tuning. The goal was to maximize validation accuracy and obtain a robust model for the test set.

▮ *Initial Model Performance (Before Hyperparameter Tuning)*

Before hyperparameter tuning, the model achieved the following performance on the test set:

Metric	Value
Test Accuracy	95.73%

Table 12: Performance of the ConvNeXt V2 model before hyperparameter tuning.

Classification Report (Before Tuning):

Class	Precision	Recall	F1-score	Support
AD	0.942	0.963	0.952	232
CN	0.990	0.994	0.992	167
MCI	0.960	0.930	0.945	186
Accuracy		0.9573		
Macro Avg	0.964	0.962	0.963	585
Weighted Avg	0.957	0.957	0.957	585

Table 13: Classification report for the ConvNeXt V2 model before hyperparameter tuning.

▮ *Hyperparameter Tuning*

A hyperparameter search was conducted over 16 combinations of learning rate, batch size, drop path rate, head initialization scale, weight decay, and epochs. The best configuration obtained was:

- Variant: `femto`
- Learning Rate: 0.0003
- Batch Size: 64
- Drop Path Rate: 0.1

- Head Init Scale: 0.75
- Weight Decay: 0.0001
- Epochs: 30

This configuration achieved the highest validation accuracy of 96.92%.

Final Model Performance (After Hyperparameter Tuning)

Metric	Validation Set	Test Set
Accuracy	96.92%	97.26%
Training Time	37.0 s	

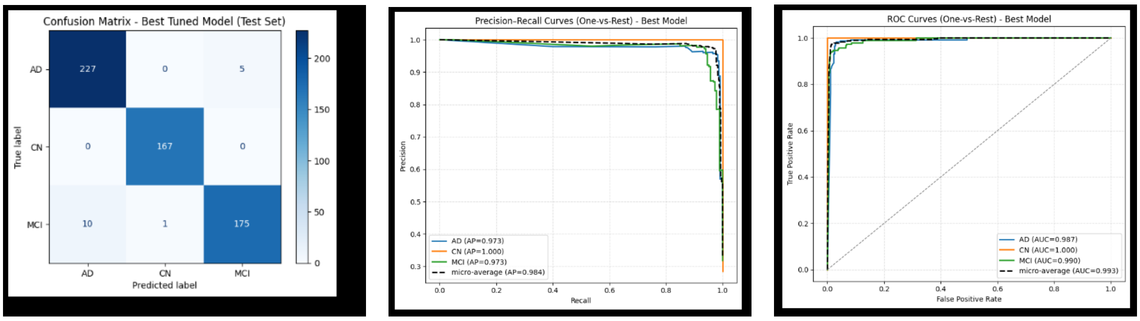
Table 14: Final evaluation metrics for the ConvNeXt V2 model after hyperparameter tuning.

Classification Report (After Tuning):

Class	Precision	Recall	F1-score	Support
AD	0.9578	0.9784	0.9680	232
CN	0.9940	1.0000	0.9970	167
MCI	0.9722	0.9409	0.9563	186
Accuracy		0.9726		
Macro Avg	0.9747	0.9731	0.9738	585
Weighted Avg	0.9727	0.9726	0.9726	585

Table 15: Classification report for the ConvNeXt V2 model after hyperparameter tuning.

This comparison clearly demonstrates that hyperparameter tuning improved the model's overall performance, increasing the test accuracy from 95.73% to 97.26% and improving class-wise precision, recall, and F1-scores.



(a) Confusion Matrix (b) Precision-Recall Curve (c) ROC Curve

Figure 5: Performance evaluation plots for ConvNeXt V2 model.

4.2.5 Swin Classifier Performance Evaluation

The Swin Classifier was evaluated on the dataset before and after hyperparameter tuning. The goal was to improve model accuracy and obtain reliable predictions on the test set.

- *Initial Model Performance (Before Hyperparameter Tuning)*

Before hyperparameter tuning, the model achieved the following performance:

Metric	Value
Test Accuracy	95.38%

Table 16: Performance of the Swin Classifier before hyperparameter tuning.

- *Final Model Performance (After Hyperparameter Tuning)*

After tuning hyperparameters, the model achieved the following performance:

Metric	Value
Final Test Accuracy	97.09%

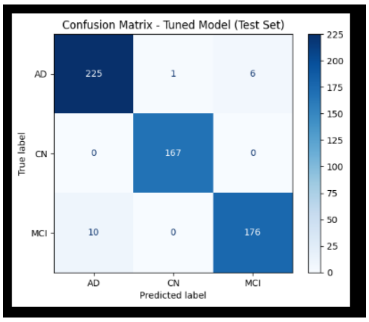
Table 17: Performance of the Swin Classifier after hyperparameter tuning.

Classification Report (Tuned Model - Test Set)

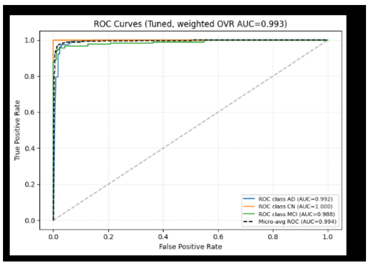
Class	**Precision**	**Recall**	**F1-score**	**Support**
AD	0.9574	0.9698	0.9636	232
CN	0.9940	1.0000	0.9970	167
MCI	0.9670	0.9462	0.9565	186
Accuracy		0.9709		
Macro Avg	0.9728	0.9720	0.9724	585
Weighted Avg	0.9709	0.9709	0.9709	585

Table 18: Classification report of the Swin Classifier on the test set after hyperparameter tuning.

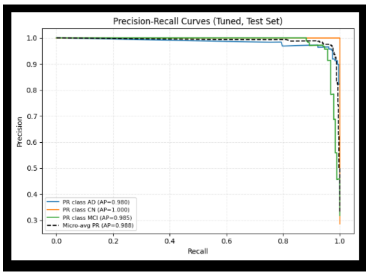
Hyperparameter tuning improved the test accuracy from 95.38% to 97.09% and enhanced class-wise precision, recall, and F1-scores, particularly for the MCI and AD classes.



(a) Confusion Matrix



(b) Precision-Recall Curve



(c) ROC Curve

Figure 6: Performance evaluation plots for Swin Classifier.

4.3 Performance Comaprision

Model	**Accuracy (%)**	**Precision**	**Recall**	**F1-score**
ConvNeXt V2[14]	97.26	0.9727	0.9726	0.9726
Swin Classifier[7]	97.09	0.9709	0.9709	0.9709
TRVFL[11]	96.07	0.9621	0.9636	0.9627
EDRVFL[3]	98.63	0.9863	0.9863	0.9863
RVFL[8]	98.46	0.9846	0.9846	0.9846

Table 19: Comparison of performance metrics for all five models on the test set.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this study, we looked at several machine learning and deep learning models to detect and classify Alzheimer's Disease (AD), Mild Cognitive Impairment (MCI), and Cognitively Normal (CN) subjects using imaging and feature data. The models we examined were ConvNeXt V2, Swin Classifier, Random Vector Functional Link (RVFL), TRVFL, and Ensemble Deep RVFL (EDRVFL).

The results show that all models performed well in classification, with significant variations in accuracy and reliability.

- ConvNeXt V2 achieved a test accuracy of 97.26
- Swin Classifier performed slightly better with a test accuracy of 97.09
- RVFL and its variants TRVFL and EDRVFL reached the highest accuracies of 98.63
- Ensemble strategies, especially in EDRVFL, improved reliability, demonstrating strength across class labels with high precision, recall, F1-scores.

Overall, the study shows that functional link neural networks, particularly when used in ensemble architectures, perform better for Alzheimer's detection.

Deep learning image-based models also produce strong results. This suggests that multi-modal approaches, which combine imaging and structured features, can further improve early detection of Alzheimer's Disease.

The models created in this project can serve as a basis for real-world clinical decision support systems. They offer reliable predictions to help neurologists and healthcare professionals with early diagnosis and intervention planning.

5.2 Future Work

Future extensions of this research may involve expanding the dataset to include more diverse subjects and imaging variations, which can further improve model stability. Additionally, exploring new deep learning architectures, transformer-based medical models, or hybrid fusion strategies could improve diagnostic accuracy. Another important direction is incorporating explainable AI techniques to provide clearer insights into model decisions, making the system more transparent and clinically reliable. Finally, real-time deployment, validation across institutions, and long-term analysis of disease progression can significantly broaden the practical application of this work.

Appendix A

Some Complex Proofs and simple Results

References

[1] Fan, Y., Batmanghelich, N., Clark, C. M., and Davatzikos, C. Structural mri-based classification of alzheimer's disease using feature selection and support vector machine. *NeuroImage 41*, 3 (2008), 352–357.

[2] Goel, T., Sharma, R., Tanveer, M., Suganthan, P., Maji, K., and Pilli, R. Multimodal neuroimaging based alzheimer's disease diagnosis using evolutionary rvfl classifier. *IEEE Journal of Biomedical and Health Informatics* (2023).

[3] Hu, M., Chion, J. H., Suganthan, P. N., and Katuwal, R. K. Ensemble deep random vector functional link neural network for regression. *IEEE Transactions on Systems, Man, and Cybernetics: Systems 53*, 5 (2022), 2604–2615.

[4] Jack, C. R., et al. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging 27*, 4 (2008), 685–691.

[5] Jack, C. R., Knopman, D. S., Jagust, W. J., Shaw, L. M., Aisen, P. S., Weiner, M. W., Petersen, R. C., and Trojanowski, J. Q. Hypothetical model of dynamic biomarkers of the alzheimer's pathological cascade. *Lancet Neurology 9*, 1 (2010), 119–128.

[6] Liu, M., Zhang, J., Adeli, E., and Shen, D. Landmark-based deep multi-instance learning for brain disease diagnosis. *Medical Image Analysis 43* (2018), 157–168.

[7] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 12009–12019.

[8] Pao, Y.-H., Park, G.-H., and Sobajic, D. J. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing 6*, 2 (1994), 163–180.

[9] Payan, A., and Montana, G. Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. In *International Workshop on Machine Learning in Medical Imaging* (2015), Springer, pp. 355–362.

[10] Reuter, M., Schmansky, N. J., Rosas, H. D., and Fischl, B. Within-subject template estimation for unbiased longitudinal image analysis. *NeuroImage 61*, 4 (2012), 1402–1418.

[11] Sarkar, C., Gupta, D., and Hazarika, B. B. 1-norm twin random vector functional link networks based on universum data for leaf disease detection. *Applied Soft Computing 148* (2023), 110850.

[12] Sharma, R., Goel, T., Tanveer, M., Suganthan, P. N., Razzak, I., and Murugan, R. Conv-ervfl: Convolutional neural network based ensemble rvfl classifier for alzheimer's disease diagnosis. *IEEE Journal of Biomedical and Health Informatics 27*, 10 (2023), 4995–5003.

[13] Suk, H.-I., Lee, S.-W., and Shen, D. Hierarchical feature representation and multimodal fusion with deep learning for ad/mci diagnosis. *NeuroImage 101* (2014), 569–582.

[14] Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S., and Xie, S. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023), pp. 16133–16142.