International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Multimodal Machine Learning for Early Alzheimer's Disease Detection: Leveraging Cognitive Features, and Resnet-Based Image Analysis with SVM Tuning

Rama Lakshmi Boyapati[1], Dr. Radhika Y. [2]

Submitted: 29/01/2024 **Revised**: 07/03/2024 **Accepted**: 15/03/2024

Abstract: With the growing prevalence of Alzheimer's disease globally, early and accurate diagnosis becomes imperative for effective intervention. Our study leverages a dataset comprising diverse biomarkers and cognitive features, employing advanced machine learning algorithms, particularly machine learning methods. The procedure starts with the extraction of features using ResNet, giving preference to skip connectors to minimize residual errors. ResNet50 is chosen for its exceptional capabilities in image analysis and classification. Parameters of the model are fine-tuned by adjusting them based on the discrepancy between expected and actual class scores. In the concluding layer, the SVM model undergoes tuning, specifically in the context of Alzheimer's detection for binary and multiclass assignments. Bayesian optimization with Hyperopt systematically explores the hyperparameter space, optimizing variables like kernel selection and regularization to enhance the model's effectiveness on the validation set. The proposed model demonstrates promising results in discriminating between Alzheimer's disease and normal cognitive aging, showcasing high sensitivity and specificity. The integration of multimodal data enhances the robustness of the model, providing a comprehensive and reliable tool for early detection. This research contributes to the ongoing efforts to develop precise and accessible diagnostic tools for Alzheimer's disease, with potential implications for timely clinical intervention and improved patient outcomes.

Keywords*: Hyperopt, regularization, tuning, convolution block, Activation function, bottleneck, normalization, transition block, Tree-structured Parzen Estimators, hyperplane, CNN.*

1. Introduction

MRI can help differentiate Alzheimer's disease from other types of dementia or neurological disorders. By capturing detailed images of the brain, MRI allows healthcare professionals to monitor the progression of Alzheimer's disease over time. ResNet-50 introduces residual connections, allowing information to skip one or more layers during forward and backward passes. Residual connections assist in maintaining the flow of gradients throughout the network. This addresses the challenges associated with vanishing gradients problem. The residual architecture enables the efficient learning of representations. By allowing the network to focus on learning residuals (differences between the input and the output), it becomes easier for the model to capture and learn complex patterns. Pre-trained models may not have learned disease-specific features relevant to Alzheimer's disease. If the source domain did not include a diverse range of Alzheimer's disease cases, the transferred knowledge may not capture the full spectrum of disease-related patterns.

CNN-integrated ML is needed to identify Alzheimer's early on. Early detection allows for early intervention and treatment, which may delay Alzheimer's development and improve quality of life. CNNs excel in visual data processing, making them ideal for brain scan analysis. The method uses These algorithms to extract complicated patterns and data from medical pictures, improving AD. These integrations allow automated screening techniques. These techniques can analyse neuroimaging data like MRIs and PET scans to help doctors diagnose Alzheimer's early. Automation streamlines and scales screening. Traditional AD may include subjective evaluations or medical imaging. Consistency and a more objective reduction in human error are guaranteed by CNN-integrated ML. They can develop models that identify trends and characteristics unique to each patient. With this individualised approach, physicians may optimise medicines by tailoring treatment plans to the unique symptoms and progression of Alzheimer's disease in each patient. Combining allows large datasets to yield insights. This method may be used by researchers studying Alzheimer's disease to identify underlying factors, biomarkers, and subtle patterns. These discoveries advance our understanding, treatment, and prevention of diseases.

[1] Research scholar, Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed-to-be University), Visakhapatnam-530045, AP, India
ramalakshmi.boyapati@gmail.com
[2] Professor, Department of Computer Science and Engineering, GITAM School of Technology, GITAM(Deemed-to-beUniversity), Visakhapatnam-530045,AP,India.
ryalavar@gitam.edu

1.1. Limitations of Feature Extraction using Traditional Approaches:

1.2. Some restrictions affect the efficacy of diagnostic models when using image processing techniques for the identification of Alzheimer's disease. The intrinsic diversity of illness development across people is one of

the main constraints. Alzheimer's disease can present itself in a variety of ways, and depending solely on static imaging data could not adequately reflect the condition's dynamic character over time. Furthermore, irregularities and noise can be introduced into the data by the heterogeneity of image capture devices and the absence of standardised imaging techniques, which may impair the detection algorithms' performance. An further constraint is the difficulty of attaining specificity in the diagnosis of Alzheimer's disease. Techniques for processing images may pick up characteristics shared by a number of neurodegenerative illnesses, which might result in incorrect diagnoses and decreased diagnostic specificity. Since the use of medical imaging data requires careful management and adherence to privacy legislation, ethical considerations around privacy and data security also play a role. Furthermore, it is still difficult to understand characteristics produced from images, which makes it challenging to connect patterns found to the molecular mechanisms driving Alzheimer's disease. Improving the accuracy and practical applicability of image processing techniques for Alzheimer's disease diagnosis requires addressing these drawbacks.

There are several restrictions on the use of convolutional neural networks (CNNs) for feature extraction in the identification of Alzheimer's disease. A notable obstacle is the need for extensive and varied datasets. Since CNNs are data-hungry models, it can be difficult to find a sufficiently large dataset that adequately reflects the variability of Alzheimer's disease across different stages and patient demographics. Moreover, there is still uncertainty regarding the interpretability of the characteristics that CNNs extract. Because CNN designs are hierarchical and complicated, they might produce characteristics that are challenging to understand in the context of neurobiology, which hinders our capacity to understand the underlying causes of illness. Furthermore, CNNs are susceptible to changes in picture quality, and the accuracy of feature extraction in medical images may be impacted by noise or artefacts. Finally, there may be issues with model interpretability and processing resources due to the high-dimensionality of CNN-generated features. Notwithstanding the encouraging potential of CNNs, these constraints highlight how crucial it is to take into account the unique attributes of Alzheimer's disease datasets as well as the interpretability of features that are retrieved in order to develop precise and clinically relevant detection models.

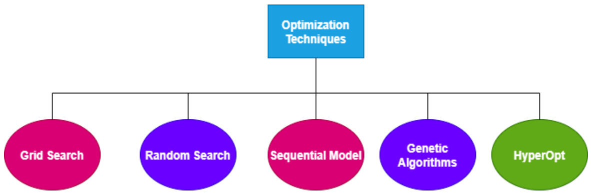


Fig 1: Optimization Categorization

Grid Search:

In machine learning, grid search is a hyperparameter optimisation method. To determine a model's ideal configuration, a predetermined set of hyperparameter combinations must be methodically evaluated. The model's performance is evaluated for every combination of the hyperparameter values, frequently through the use of cross-validation. Grid search is a thorough search technique that offers a thorough examination of the hyperparameter space. By choosing the optimal hyperparameter values, it enhances model performance despite being computationally demanding.

1.2. Optimization Techniques & their working for Classification: optimization techniques are essential for training CNNs effectively, ensuring convergence, avoiding common issues like vanishing gradients, and enabling the models to generalize well to new data. Hyperparameter tuning, especially when performed using rigorous optimization techniques, adds a level of credibility to the model-building process. It demonstrates a systematic and principled approach to finding the best model configuration. The categorization of optimization techniques is shown in figure 1.

- Random Search:

Unlike grid search, which searches exhaustively, random search is a hyperparameter tuning strategy used in machine learning that chooses hyperparameter values at random from predetermined ranges. Because it investigates configurations more haphazardly and could identify optimal settings with fewer trials, this strategy is computationally efficient. It works well for models that have large hyperparameter spaces since it makes it easier to explore and find the best possible settings. Cross-validation is frequently used in conjunction with random search to assess model performance over a range of hyperparameter combinations.

- Sequential model:

In machine learning, a sequential model is a linear stack of layers, each of which performs a particular function. The architecture for creating and refining neural networks is simple and linear. Information moves from the input layer across the hidden levels and out to the output layer in a sequential model. This methodology works well with more straightforward architectures in which the layers of a neural network structure are added one after the other.

- Genetic Algorithms:

Genetic techniques (GAs) are genetic and natural selection-inspired optimization techniques. They involve a population of chromosomes, which are possible solutions. The algorithm finds optimum or almost ideal solutions by evolving and refining the population over generations through a process of selection, crossover, and mutation. GAs are applied to optimisation issues in a variety of domains where the search space is poorly understood or convoluted.

Hyperparameter tweaking, essential to machine learning model optimisation, is done via grid search. A grid containing hyperparameter values in each dimension and various spots within each dimension to test. The method trains and assesses the ML model for each grid hyperparameter combination. A comprehensive search of the hyperparameter grid's combinations. This extensive approach explores the hyperparameter space by excluding no configuration. Model performance is recorded using the evaluation measure. Grid search records the optimal hyperparameter combination. Grid search is comprehensive but computationally costly, particularly for large hyperparameter spaces. So, practitioners combine grid granularity with computing resources and temporal restrictions to strike a balance between exploration & efficiency.

Algorithm for Grid Search:

Input: Dataset (DS); Algorithm (Algo); θ (hyper parameter Space), GS (Grid Search)

Output: Optimized Parameters

```
Result ← {};
for i to M do
θ ← select hyperparameter (GS, Algo, θ)
        Model ← train (Algo, θ, DS_train)
        Result ← eval(Model, DS_test)
end
θ ← Adjust (θ, evaluate (Result))
```

Hyperopt and grid search are two distinct optimization algorithms used in machine learning for hyperparameter tuning, each with its own approach and characteristics. Grid search is a straightforward and exhaustive method that systematically explores predefined hyperparameter values across a grid, evaluating each combination independently. Although straightforward to apply, grid search can be computationally demanding, particularly in hyperparameter spaces with high dimensions, due to its lack of adaptability throughout the optimization process. However, Hyperopt uses a more advanced method, the Tree of Parzen Estimators (TPE) algorithm, to sample hyperparameters dynamically based on performance observations and probabilistically model the search space. In contrast to grid search, Hyperopt effectively explores the hyperparameter space by concentrating on promising regions while it iteratively adjusts its search method. Because of its versatility, Hyperopt is especially useful in handling high-dimensional, complicated search spaces, since it has a tendency to converge to optimum hyperparameter configurations more quickly than grid search. In summary, Hyperopt offers a more flexible and effective method for hyperparameter optimisation than grid search, which is simple yet thorough.

2. Literature Survey

Taher M. Ghazal et al [1] established an ADDTLA approach for the early identification of Alzheimer's disease. MRI pictures of Alzheimer's disease phases from Kaggle were used for training. The ADDTLA system model detects and classifies illnesses early using MRI images. It has two layers: preparation and application. MRI data is changed into a standard file in the first one, and a tweaked AlexNet is used for transfer learning in the second one. ImageNet-trained AlexNet is medically applicable. Alzheimer's disease classification-specific convolutional, pooling, and fully connected layers are used. The updated network is trained on AD multi-class labels with output classification at the final three layers, fully linked, and softmax. TL improves target domain performance by using source domain knowledge. Cloud-stored training models are validated. The trained model classifies MRI data into Alzheimer's disease stages throughout this phase. By using a part of the original sample as test data, we can be sure that the proposed model is accurate and stable. Transfer learning using a modified AlexNet helps the ADDTLA system identify and categorise Alzheimer's disease stages. Large datasets and knowledge transfer improve AD detection.

Janani Venugopalan et al [2] has implemented an autoencoder and 3D-CNN methodologies for the identification of AD. A number of steps are taken before the MRI image data is used. These include filtering out noise, head stripping, tissue segmentation, normalisation, and co-registration to MNI space. From ADNI1, ADNI2, & ADNI GO, 1680 clinical characteristics are derived. The numeric data is normalised, and the category data is turned into binary using one-hot encoding. This is then changed into binary values of 1 or 2 so that they are all shown in the same way. Multiple filtering & feature selection stages are performed on the raw VCF file with 3 million SNPs per participant. Select 500 SNP features using mRMR while keeping SNPs on known AD-associated genes. DL generates intermediate features after modality-specific feature extraction. Auto-encoders are utilised for EHR & SNP data, 3D CNNs for imaging. There is not enough annotated training data for end-to-end training, hence DL models are used for feature representation learning. Using shared weights across modules, a 3D CNN is fed regions of interest. An integrated multi-modality DL model uses modality-generated features. EHR and SNP data are represented high-level using stacked denoising auto-encoders. After fine-tuning the auto-encoder layers, the softmax layer is removed to get intermediate features. The suggests data integration at raw, intermediate, and decision levels.

Hadeer A. Helaly et al [3] has introduced a VGG-19 technology for the identification of AD. The six phases of the framework are as follows: data collection, pre-processing, augmentation, classification, assessment, and application of medical images. To achieve the objective of early detection and categorization, each step is essential. The ADNI dataset is the source of the 2D T1w MRI data. Resampling is used to correct unbalanced classes. The dataset undergoes processing, standardisation, denoising, scaling, and format conversion. To minimise picture noise, denoising is accomplished by the use of a non-local methods technique. Conventional approaches to data augmentation, such rotation & reflection, are used to deal with the lack of medical datasets. The AD spectrum is divided into four phases, and binary classifications are done for each pair. There are two ways to do this: building simple CNN systems (2D and 3D) from scratch or utilising the VGG 19 model for transfer learning. Fully connected, pooling, and convolutional layers are components of CNN designs.

Ahmad Waleed Salehi et al [4] has proposed a CNN technology for the identification of AD. Deep learning, a form of machine learning, may improve AD detection in medical picture analysis. Deep learning algorithms may adapt to diverse architectures and hyperparameters. Includes input, hidden, & output layers. Performs regression and classification on unstructured and levelled data. Specialised in audio and picture categorization. Works with two-dimensional data using convolutional, fully connected, pooling, and normalisation layers. & voice recognition utilise it. Learns sequences using step and neuron weights. Combines CNN power with quick extreme learning machine training. Trains quickly without repetitions. Generated graphical models with hidden layer unidirectional connections. Layered input, hidden, and decoding unsupervised machine learning algorithm. Extracts features and reduces dimensions.To learn unattended and supervised. AD diagnosis using deep learning algorithms on medical imaging data is promising.

P C Muhammed et al [5] has focused to detecting AD from SVM, and DNN methodologies. Training & testing datasets, pre-processing, & extraction of features for classification and prediction comprise the common model. The new method uses deep learning techniques to minimise computing complexity and improve performance without pre-processing. AI algorithms learn categorization using pictures, text, and audio in deep learning. Uses deep neural network architecture with several layers. Deep learning is great in computer vision, facial recognition, NLP, voice recognition, & bioinformatics. Uses ADNI-labeled functional MRI. GPU with CUDA-MATLAB support. Transfer learning using pre-trained AlexNet, VGG-16, VGG-19, and GoogleNet. AD, MCI, and NC MR pictures are in the dataset. The proposed deep learning method reduces time complexity and improves accuracy over existing models.

Ahila A et al [6] has introduced a CNN technology for the identification of AD. ADNI neuroimaging data was used in the research. Traditional 18FDG-PET picture capturing includes co-registration, averaging, alignment, normalisation, and smoothing. SPM12 programme normalises images to 160 x 160 x 96 for analysis using an affine model with 12 parameters. Classification algorithms include RF, SVM, & ANN, although the paper employs a 2D CNN. 3D FDG-PET pictures are converted into numerous 2D images using a 2D CNN to distinguish AD from NC. Input, three convolutional, two subsampling/pooling (P1), dropout, fully connected, and softmax classification layers make up the CNN. Max pooling reduces feature dimensions at each layer. Non-linearity is introduced via ReLU activation. Hyperparameters are tried and corrected after several trials. A maximum iteration of 500 minimises cross entropy loss during training. A dropout layer improves generalisation and prevents overfitting. Convolutional layers extract picture low, mid, and high-level characteristics. A feature vector is concatenated and given to the fully connected layer and softmax layer for classification.

Sheng Liu et al [7] has implemented DL methodologies for the identification of AD. Imaging and diagnostic data from ADNI and NACC. Bias correction and spatial normalisation to the MNI template were done using Unified Segmentation. A 3D CNN was created to categorise instances of moderate cognitive impairment, Alzheimer's disease, and normal cognition. Convolutional layers, instance normalisation, ReLUs, and max-pooling were used. Instance normalisation, compact kernel & stride in the first layer, and larger network architecture were unique design decisions. Data enhancement includes Gaussian blurring & random cropping. A GB classifier predicted Alzheimer's disease using 138 MRI volume & thickness characteristics. Two-stage ROI quality control included Gaussian distribution fitting for outlier identification and visual inspection. t-SNE visualised high-dimensional data. Saliency maps were created to analyse the deep learning model's characteristics and reveal areas that affect its output.

Santos Bringas et al [8] has developed a CNN methodology for the identification of AD. At the AFAC creche in Santander, Spain, 35 Alzheimer's patients were followed for a week. Participants had Alzheimer's and were categorised by GDS stage. Accelerometer data sequences recorded three-axis acceleration variations. Pre-processing was necessary since the sequences were irregularly distributed throughout time and needed to be segmented into shorter, uniform-length chunks. For time-series analysis, core layers use 1-dimensional convolutions. Extracts input data properties as feature maps. The ReLU added non-linearity. Batch normalisation for each mini-batch speeds learning and generalisation. Reduced data dimensionality between layers while maintaining input mean values. Final prediction by feedforward network using convolutional layer features. Softmax activation for multi-class categorization. Convolutional layers, batch normalisation, ReLU activation, pooling, and dropout were used. Based on retrieved attributes, fully linked layer predicted. Adam optimisation and categorical cross-entropy loss function. 10-fold cross-validation with 80-20 training-testing to minimise overfitting. Goal was accelerometer-based Alzheimer's disease prediction. The seven-stage GDS was used to label the phases. Adam optimisation and categorical cross-entropy loss function. Table 1 presents different deep learning approaches and integrated approaches for disease detection.

Table 1: Analysis on Deep Learning & Integrated Approaches Analysis

Author	**Algorithm**	**Merits**	**Demerits**	**Accuracy**
Taher M. Ghazal et al	ADD TLA	The processing stages are very less.	The error rate is high.	91.7 %
Janani Venugopalan et al	Auto Encoders, 3D-CNN	The images are detected from all possible ways.	Based on the quality of the image the prediction time changes	89%
Hadeer A. Helaly et al	VGG-19	The model can classify multiple classes.	The data augmentation techniques was not appropriate.	97%
Ahmad Waleed Salehi et al	CNN	Couple of dataset are examined where the performances are great.	Time consuming was high.	97%
P C Muhammed Raees et al	SVM, DNN	This can be implemented to any application.	Based on the datasets the performance is being variated.	80%
Ahila A et al	CNN	This model can solve several	The dataset was not related to	96%

		issues at a time.	humans.	
Sheng Liu et al	DL	Images can be automatically detected.	Both analysis and performances has to be improved.	83%
Santos Bringas et al	CNN	The patterns are identified for every stage.	The dataset was little.	90%

3. Proposed Methodology

The process starts with feature extraction from ResNet. The main advantage of skip connectors in residual blocks is their ability to learn the residuals with minimum error rate. Feature extraction using ResNet50 offers several advantages in the context of image analysis and classification tasks. ResNet50 has gained acclaim for its exceptional performance in tasks related to images, thanks to its advanced deep convolutional neural network architecture. Its depth is a major benefit as it enables the network to extract complex, hierarchical characteristics from input pictures. ResNet50 solves the vanishing gradient problem by including skip connections, which facilitates the training of deeper networks without sacrificing convergence stability. The accessibility of pre-trained weights on big datasets, like ImageNet, is another noteworthy advantage. With the help of these pre-trained weights, transfer learning is made possible, enabling ResNet50 to use the information gleaned from a variety of picture data to enhance generalisation and accelerate convergence on certain tasks with a small amount of labelled data. The network's advanced design minimises the requirement for human feature engineering by allowing it to instinctively acquire and extract pertinent information. ResNet50 is a widely used and successful picture classification technique because of its depth, skip connections, and transfer learning capabilities, which together provide a potent tool for feature extraction. The operation of the ResNet blocks is covered in the section below.

1. Convolution Block: The working of Convolution block is shown in figure 2

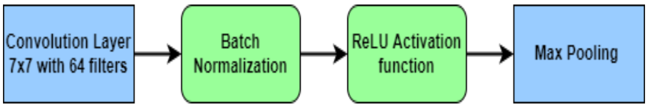


Fig 2: Layers of Convolution Block

a. 64-filter convolutional layer measuring 7 by 7: This layer performs a convolution procedure using 64 7x7 filters to extract data from the input pictures. Every filter generates 64 distinct feature maps by scanning the input and identifying patterns and geographical information. The computation of the convolution is shown in equation (1)

$$New\_Intensity(X,Y) = \sum_{i=1}^{nf}\sum_{j=1}^{nf}\sum_{k=1}^{c}(Intensity(X+i,Y+j,P) * Intensity(i,j,P)) – (1)$$

Where

nf represents number of features

c represents number of channels

Batch normalization: In medical imaging tasks, finding suitable initial weights for deep neural networks can be crucial. Batch Normalization reduces sensitivity to weight initialization, making it easier to experiment with different network architectures and training configurations. Batch normalisation is used to normalise the activations after convolution, which lessens internal covariate shift. By keeping each feature's size and mean constant throughout the mini-batch, this improves training stability and quickens convergence. The computation is shown in equation (2)

$$Normalized\_Intensity(X_i,Y_i) = \gamma * \sum_{i=1}^{K}\frac{Intensity(X_i,Y_i)-\mu}{\sqrt[2]{\sigma^2+\in}} - (2)$$

Where

γ is a learnable scale parameter

μ is mean of the corresponding kernel

k is the kernel size or filter size

σ^2 is standard deviation of the corresponding kernel

∈ is avoidance parameter to handle run exception c.

ReLU activation: ReLU induces sparsity in the network, meaning that only a subset of neurons is activated for a given input. ReLU allows ResNet-50v2 to model more complex relationships in the data, enhancing its representational power. The ability to capture non-linearities is important for the model to learn and discern the subtle patterns indicative of Alzheimer's disease in medical images. By substituting zero for every negative value in the feature maps, it encourages the discovery of subtle patterns and enhances the network's capacity to represent complex connections in the data.

d. Max pooling: In Alzheimer's disease detection, medical images, such as MRI scans, often contain hierarchical patterns and features at different scales. Max pooling allows the network to focus on the most relevant features while reducing the spatial resolution. Max pooling is used to reduce the computational complexity and improve translation invariance by down-sampling the spatial dimensions of the map of features. By choosing the greatest value, this process helps with abstraction & feature preservation by preserving the most significant data from each local area. Max pooling acts as a form of regularization by discarding less informative details

2. Residual Block: It is otherwise called to be “Bottleneck”, which contains 3 important components. The main component of the residual block is “identity”, which directly pass the gradients to the main deep layers directly ignoring the intermediate layers. Figure 3 represents the working of the residual block.

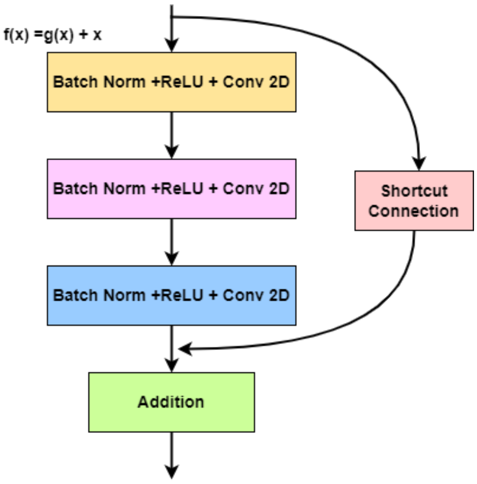

f(x) =g(x) + x
Batch Norm +ReLU + Conv 2D
Batch Norm +ReLU + Conv 2D
Batch Norm +ReLU + Conv 2D
Shortcut Connection
Addition

Fig 3: Working of Residual Block

The incorporation of a detour or skip connection, which permits the input to circumvent one or more layers, is a crucial element. The residual block models the discrepancy between the intended output and input. By using the fast link to add identity mapping, the network is pushed to learn only the residue rather than the whole change. Batch normalisation normalises activations and reduces internal covariate shift in each residual block. ResNet50v2 stacks residual blocks. The network learns hierarchical characteristics of various complexity via stacking. Skip connections guarantee smooth gradient flow during backpropagation, allowing deep network training without deterioration. Residual Blocks promote "global residual learning." This method allows direct information flow from input to output, making deep network training easier without compromising performance.

2. Transition Block: Transition Blocks usually comprise layers of pooling and convolution that minimise feature map spatial dimensions. Downsampling compresses data and highlights key characteristics, making higher-level representations easier to extract. Stage transitions may affect channel count. Transition Blocks use 1x1 convolutions to alter feature map channels, maintaining network compatibility. Transition blocks often use batch normalisation to stabilise and normalise activations, speeding training convergence. ReLU activation functions help networks learn complicated patterns by introducing non-linearity. Transition blocks facilitate network information flow. These blocks help solve the disappearing gradient problem by changing the size and shape of channels and space dimensions. This makes backpropagation and learning work well in more complex network designs. Transition blocks simplify calculation. Reduced spatial dimensions and wise channel size adjustments let these blocks maximise resource use during training and inference. Transition blocks simplify calculation. Reduced spatial dimensions and wise channel size adjustments let these blocks maximise resource use during training and inference. The block structure of transition block is shown in figure 4.

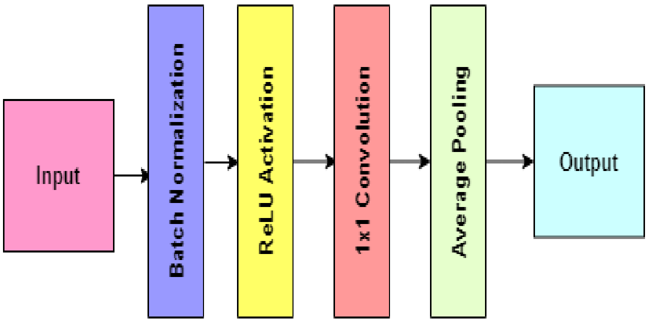
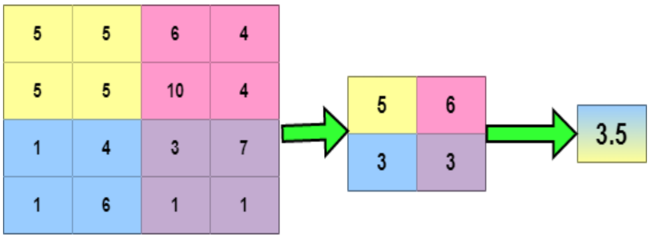


Fig 4: Transition Block Architecture

3. Global Average Pooling: Global Average Pooling averages feature map values across spatial dimensions. This reduces spatial information to one value per feature map. Global Average Pooling cuts model parameters more than completely linked layers. Reduced overfitting, computational efficiency, and model complexity are beneficial in DNN like ResNet50. Global Average Pooling is more informative by averaging instead of maxing. To preserve spatial hierarchies and critical spatial linkages in the final feature vector, this is necessary. Global Average Pooling improves generalisation. It works like regularisation by collecting the most important data from each feature map and showing it all in one place. This helps the model focus on important trends and become less sensitive to specific places. Input picture size compatibility is a major benefit of Global Average Pooling. Global Average Pooling adjusts to spatial dimensions, making it more adaptable and appropriate to a variety of input sizes than completely linked layers. The computations in GAP layer are shown in figure 5



5	5	6	4
5	5	10	4
1	4	3	7
1	6	1	1

5	6
3	3

3.5

Fig 5: Working of GAP Layer

4. Dense Block: Typically, layers of convolution and pooling produce a three-dimensional tensor. This tensor is first turned into a one-dimensional vector by the Fully Connected Layer. This procedure deconstructs the spatial structure & converts retrieved characteristics into classification-ready format. Fully linked Layers have neurons linked to all preceding layer activations. Each link has a weight, which the layer learns throughout training. The layer's neurons weight sum input values and apply biases to alter the feature vector. A ReLU or softmax activation function is usually used by the FC Layer after the weighted sum. This gives the model non-linearity to capture complicated data linkages and patterns. Final class scores or forecasts come from the Fully Connected Layer. Each neuron in this layer may represent a class in picture classification, and the output values reflect the input's class probability or confidence. During training, the model learns important Fully Connected Layer weights and biases. The model is optimised for accurate predictions by adjusting these parameters depending on the gap between anticipated class scores and real labels. Figure 6 presents the working of dense block

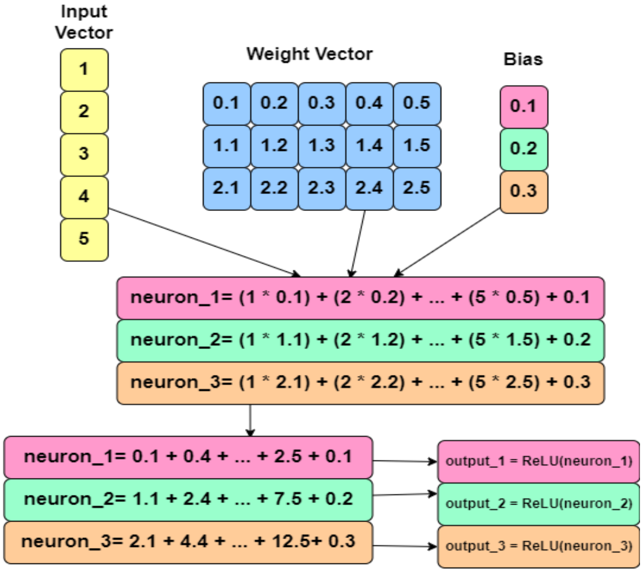

Input Vector
1
2
3
4
5
Weight Vector
0.1 0.2 0.3 0.4 0.5
1.1 1.2 1.3 1.4 1.5
2.1 2.2 2.3 2.4 2.5
Bias
0.1
0.2
0.3
neuron_1= (1 * 0.1) + (2 * 0.2) + ... + (5 * 0.5) + 0.1
neuron_2= (1 * 1.1) + (2 * 1.2) + ... + (5 * 1.5) + 0.2
neuron_3= (1 * 2.1) + (2 * 2.2) + ... + (5 * 2.5) + 0.3
neuron_1= 0.1 + 0.4 + ... + 2.5 + 0.1
neuron_2= 1.1 + 2.4 + ... + 7.5 + 0.2
neuron_3= 2.1 + 4.4 + ... + 12.5+ 0.3
output_1 = ReLU(neuron_1)
output_2 = ReLU(neuron_2)
output_3 = ReLU(neuron_3)

Fig 6: Working of Dense Block

Classification using Tuned Transfer Learning: The last layer of the model is designed by the tuning the SVM model. SVMs can be used for binary and multiclass classification tasks. In Alzheimer's detection, the problem may involve classifying subjects into categories such as normal, mild cognitive impairment (MCI), and Alzheimer's disease stages. Bayesian optimization, as implemented by Hyperopt, is particularly efficient for exploring the hyperparameter space. SVMs have hyperparameters, such as the choice of kernel, regularization parameter (C), and kernel-specific parameters (e.g., gamma for the RBF kernel). Hyperparameter tuning helps find the combination of hyperparameters that maximizes the model's performance on the validation set.

1 Working of HypOpt: Hyperopt is a Python library designed for automating the process of hyperparameter optimization, a crucial step in fine-tuning machine learning models. The library uses methods such as Tree-structured Parzen Estimators (TPE) in order to provide a Bayesian optimisation strategy. Finding the collection of hyperparameters that maximises or minimises a chosen objective function—a measure of the model's performance—is the primary goal of hyperparameter optimisation. In order to work, Hyperopt proposes hyperparameter configurations repeatedly, evaluates them employing the objective function, and modifies its search method in response to observed outcomes. Hyperopt can effectively search the hyperparameter space and converge to optimum or nearly optimal configurations thanks to this iterative approach. In Figure 7, the operation of Hyperopt is displayed.

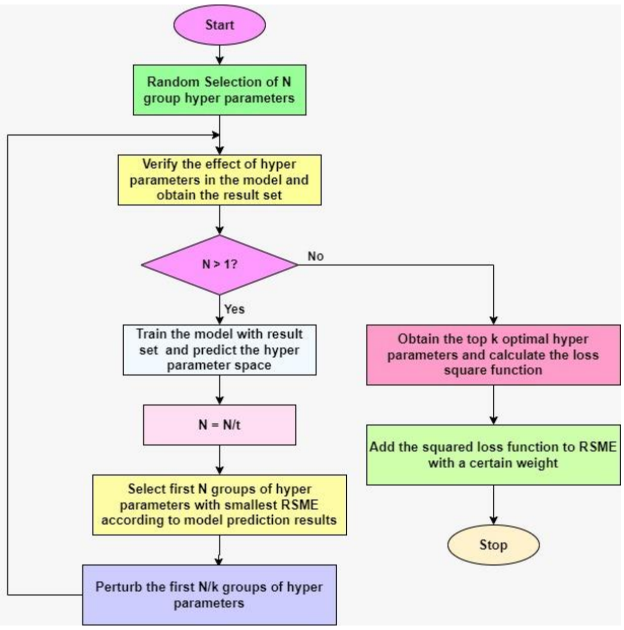

Start
Random Selection of N group hyper parameters
Verify the effect of hyper parameters in the model and obtain the result set
N > 1?
No
Yes
Train the model with result set and predict the hyper parameter space
N = N/t
Select first N groups of hyper parameters with smallest RSME according to model prediction results
Perturb the first N/k groups of hyper parameters
Obtain the top k optimal hyper parameters and calculate the loss square function
Add the squared loss function to RSME with a certain weight
Stop

Fig 7: Hyperopt Work Flow for Tuning Process

The user must define the goal function and the hyperparameter search space in order to use Hyperopt. Every hyperparameter's range or values, whether continuous, categorical, or discrete, are specified in the search space. The goal function, which takes in a set of hyperparameters and outputs a single metric that has to be optimised, must also be specified by users. TPE is one of the most widely used optimisation methods supported by Hyperopt. Based on their unique optimisation challenge and available computing power, users choose the algorithm. After setup, the optimisation procedure entails executing Hyperopt, which recommends hyperparameter settings, assessing them via the objective function, and modifying its internal model in response to the performance data obtained.

Hyperopt uses an iterative optimisation method, honing its search approach in response to the findings of earlier analyses. Because it is iterative, it can effectively explore the hyperparameter space and adjust to the peculiarities of the optimisation issue. Hyperopt can handle large and high-dimensional parameter spaces with robustness because it balances exploration with exploitation through the use of probabilistic modelling and Bayesian optimisation. Hyperopt drastically lowers the amount of human labour needed for testing by simplifying the hyperparameter tuning process. This makes it an invaluable tool for practitioners looking for an effective and efficient model setup. With its adaptability, scalability, and capacity to manage many hyperparameters, Hyperopt is a well-liked option among machine learning experts for enhancing model performance.

```
Input: Target score function H(θ), hyper-parameter space θ, max no of evaluation nmax
        Select an initial hyper-parameter configuration θ0 Є θ
Evaluate the initial score y0= H(θ0)
Set y* = H(θ0) and θ* = θ0 .
For n=2, ..., nmax do
Choose an alternative set of hyperparameter values θn Є θ using some optimization strategy
Evaluate H to obtain a new numeric score yn = H(θn)
If yn < F*
θ*= θn and y* = yn
End if
end for
Output: θ* and y*
```

Working of Tuned SVM: The data points that are nearest to the hyperplane and have an impact on its location are known as support vectors. These points are crucial in defining the margin and determining the optimal hyperplane. The main objective of SVM is to find the hyperplane that maximizes the margin between classes. The objective function can be achieved in two ways One versus One and One versus All. The proposed research uses one versus one and defines the computation of the objective function as shown in equation (3)

$$Objective\_Function\_SVM = \min (\frac{1}{2} * \sum_{i=1}^{n} W_i^2 + C.\sum_{i=1}^{n}\sum_{j=1}^{n} \in_{ij}) \text{- (3)}$$

n is the number of classes

W_i are the weights and bias terms for the ith SVM class label

C is the regularization parameter controlling the trade-off between maximizing the margin and minimizing the misclassification error

∈_ij slack variables allowing for misclassifications.

Hyper parameter	Description	Equation	Possible Values
C	The regularisation parameter is responsible for managing the trade-off between obtaining a low testing error and a low training error. Reduced regularisation is the outcome of higher values of C.	$\min (\frac{1}{2} * \sum_{i=1}^{n} W_i^2 + C.\sum_{i=1}^{n}\sum_{j=1}^{n} \in_{ij})$- (x)	Any fractional value on the scale of 10
Kernel	The kind of kernel that is used linear, polynomial, radial basis function (RBF), etc.—is specified by this hyperparameter. The particulars of the data and the issue being solved will determine which kernel is best.	$K = e^{-\gamma \sum_{i=1}^{n} Input_i - \mu_{Input}}{}^{2}$ γ is a positive parameter	[RBF, Linear, Sigmoid, Polynomial]
Degree	This hyperparameter shows the degree of the kernel that is polynomial if the kernel	$D = \gamma\mu_{Input} * \sum_{i=1}^{n} Input_i + r^d$ r is a constant term.	Any positive integer

	is polynomial.	d is the degree of the polynomial	
gamma	The kernel coefficient for polynomial and RBF kernels is the gamma hyperparameter. Decision boundaries get increasingly difficult as values increase.		
decision_function_ shape	This hyperparameter determines the shape of the decision function. 'ovo' stands for one-vs-one, and 'ovr' stands for one-vs-the-rest.		

Table 2: Parameter Description

Optimizing the performance of the Support Vector Machines (SVM) involves a critical tuning process. The *C parameter is a crucial parameter that affects how well training points are classified and how well a smooth decision boundary is achieved. A lower C number allows for a more flexible boundary that accepts certain misclassifications, whereas a higher C value places a stricter bar for precise categorization. To avoid the model being over- or underfitted to the training set, tuning C involves striking a balance. Moreover, the success of SVM depends on the **kernel function* selection; popular options include polynomial, linear, and radial basis function (RBF). RBF is often used because of its flexibility, depending on the kind of data. By adjusting kernel parameters, such as the degree in polynomial kernels and the gamma parameter in RBF, the model may more accurately represent complicated relationships in the data.

Using the *class weights argument to address class imbalances is another crucial component. Giving the minority class a larger weight in situations when one class outnumbers the other by a substantial margin aids the model in prioritising accurate forecasts for both groups. Furthermore, the tolerance for mistakes in regression tasks is influenced by optimising the **epsilon parameter* in the SVM regression model. A more sensitive model, which fits the training data accurately but may cause overfitting, is produced by a lower epsilon. On the other hand, a greater epsilon permits greater flexibility and can handle a wider variety of data patterns. To fully utilise SVMs for classification and regression tasks, careful parameter tweaking that takes into account the unique features of the dataset is necessary.

To get the most performance out of the model and modify it to fit the unique features of the dataset, Support Vector Machines (SVM) must be tuned. The behaviour of SVMs is determined by a number of parameters, including the kernel type, related kernel parameters, and the regularisation parameter (C). Adjusting is necessary because varying datasets and workloads need for customised setups to achieve the ideal ratio of generalisation to model complexity. Inadequate fine tuning might cause the SVM to either overfit or underfit the training set, producing less than ideal results. SVMs may detect intricate patterns and improve predicted accuracy when their parameters are appropriately modified to take into consideration variables such as class imbalances and data characteristics. As a result, tuning is essential to optimising SVM performance in a variety of applications and guaranteeing solid and trustworthy machine learning results. The SVM computations are displayed in Figure 8.

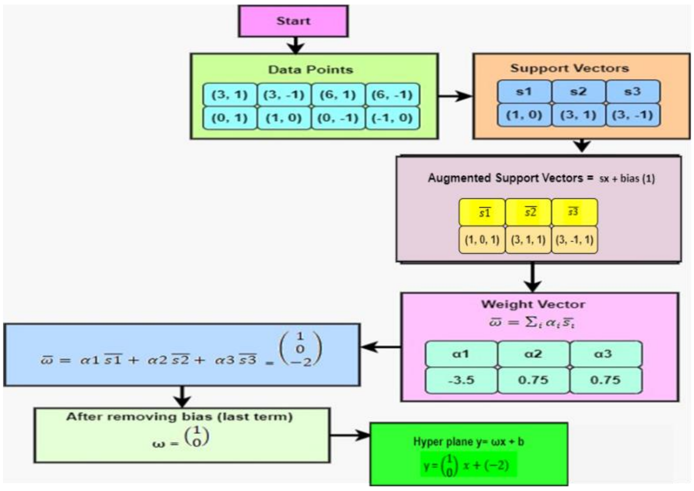
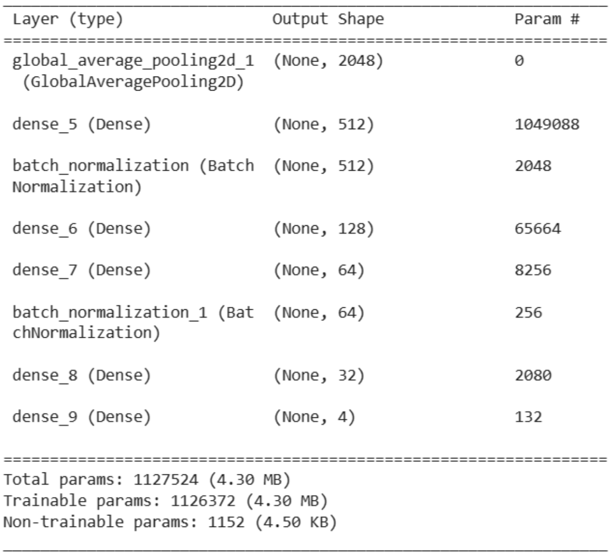

Start
Data Points
(3, 1)
(3, -1)
(6, 1)
(6, -1)
(0, 1)
(1, 0)
(0, -1)
(-1, 0)
Support Vectors
s1
s2
s3
(1, 0)
(3, 1)
(3, -1)
Augmented Support Vectors = sx + bias (1)
$\overline{s1}$
$\overline{s2}$
$\overline{s3}$
(1, 0, 1)
(3, 1, 1)
(3, -1, 1)
Weight Vector
$\overline{\omega} = \sum_i \alpha_i \overline{s}_i$
α1
α2
α3
-3.5
0.75
0.75
$\overline{\omega} = \alpha 1\, \overline{s1} + \alpha 2\, \overline{s2} + \alpha 3\, \overline{s3} = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}$
After removing bias (last term)
$\omega = \binom{1}{0}$
Hyper plane y= ωx + b
$y = \binom{1}{0} x + (-2)$

Fig 8: Kernel Computations in SVM

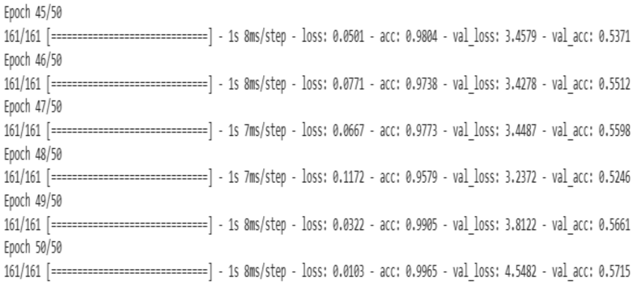
4. Results & Discussion:



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 global_average_pooling2d_1  (None, 2048)              0
  (GlobalAveragePooling2D)

 dense_5 (Dense)             (None, 512)               1049088

 batch_normalization (Batch  (None, 512)               2048
 Normalization)

 dense_6 (Dense)             (None, 128)               65664

 dense_7 (Dense)             (None, 64)                8256

 batch_normalization_1 (Bat  (None, 64)                256
 chNormalization)

 dense_8 (Dense)             (None, 32)                2080

 dense_9 (Dense)             (None, 4)                 132

=================================================================
Total params: 1127524 (4.30 MB)
Trainable params: 1126372 (4.30 MB)
Non-trainable params: 1152 (4.50 KB)
_________________________________________________________________
```

Fig 9: Summary of Neural Network

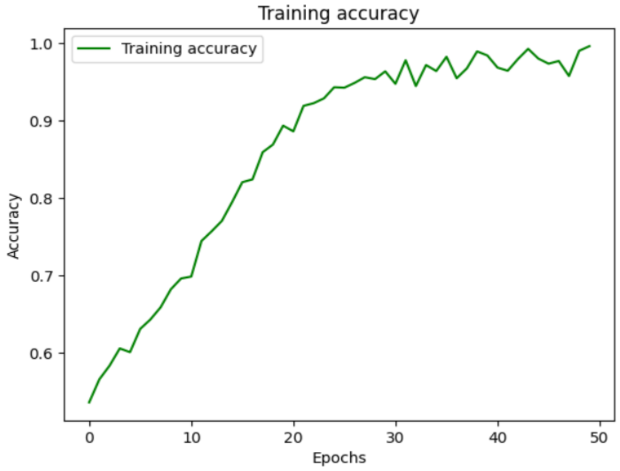
Figure 9 represents the summary of the interconnected layers in the neural network. The pooling used is the global average pooling. There are 9 dense blocks involved in each different dimension. Batch normalization has been used after every two dense layers. This sequence of layers represents the basic building blocks.



```
Epoch 45/50
161/161 [==============================] - 1s 8ms/step - loss: 0.0501 - acc: 0.9804 - val_loss: 3.4579 - val_acc: 0.5371
Epoch 46/50
161/161 [==============================] - 1s 8ms/step - loss: 0.0771 - acc: 0.9738 - val_loss: 3.4278 - val_acc: 0.5512
Epoch 47/50
161/161 [==============================] - 1s 7ms/step - loss: 0.0667 - acc: 0.9773 - val_loss: 3.4487 - val_acc: 0.5598
Epoch 48/50
161/161 [==============================] - 1s 7ms/step - loss: 0.1172 - acc: 0.9579 - val_loss: 3.2372 - val_acc: 0.5246
Epoch 49/50
161/161 [==============================] - 1s 8ms/step - loss: 0.0322 - acc: 0.9905 - val_loss: 3.8122 - val_acc: 0.5661
Epoch 50/50
161/161 [==============================] - 1s 8ms/step - loss: 0.0103 - acc: 0.9965 - val_loss: 4.5482 - val_acc: 0.5715
```

Fig 10: Epochs Report

The above figure 10 represents the training phase of the model. This shows the number steps per training epoch with the losses and accuracies at each epoch during training. It can be observed that the accuracy has been increasing by the end of the training of epochs.



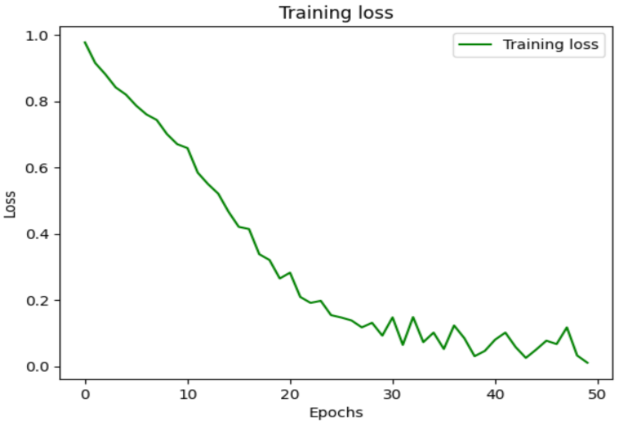


Fig 11: Accuracy & Loss Reports

Figure 11 represents the accuracy graph per epoch. It compares training data to compare the performance at every step. Training data is gradually increasing whereas loss is decreasing gradually. The accuracy and loss are on the y-axis on the respective graphs. The epoch is denoted on the X-axis in the both graphs

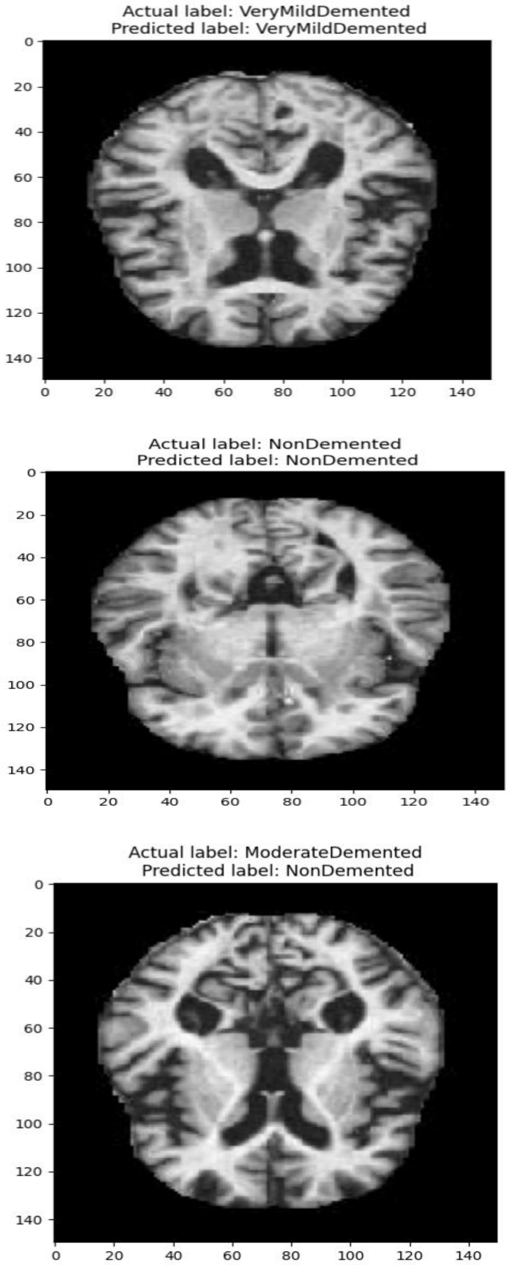

Actual label: VeryMildDemented
Predicted label: VeryMildDemented
0
20
40
60
80
100
120
140
Actual label: NonDemented
Predicted label: NonDemented
Actual label: ModerateDemented
Predicted label: NonDemented

Fig 12: Classification Reports

There are multiple classes namely, very mild demented, moderate demented, non-demented etc. The proposed model predicted the class labels as shown in the above figure 12. The comparison of actual and predicted class labels is performed. The proposed model is efficient to recognise the features and identify the disease.

5. Conclusion

To sum up, our study on the identification of Alzheimer's disease highlights the need of using machine learning techniques to provide an early and precise diagnosis. The utilization of diverse biomarkers and cognitive features, coupled with advanced algorithms, has shown promising results in effectively distinguishing individuals with Alzheimer's disease from those experiencing normal cognitive aging. Using ResNet50 for visual analysis and classification, features are extracted, parameters are adjusted based on differences between predicted and actual class scores, and the SVM model is tuned for Alzheimer's detection through the use of Hyperopt and Bayesian optimisation. The developed model, particularly employing machine learning techniques, exhibits high sensitivity and specificity, highlighting its potential as a reliable diagnostic tool. The integration of multimodal data further enhances the robustness of the approach, contributing to the ongoing pursuit of precise and accessible diagnostic solutions for Alzheimer's disease. As this continue to advance in the realm of machine learning and healthcare, the findings of this study hold promise for facilitating timely clinical interventions and improving the overall management of Alzheimer's disease, ultimately leading to enhanced patient care and outcomes.

References

[1] Ghazal, T. M., Abbas, S., Munir, S., Khan, M. A., Ahmad, M., Issa, G. F., Zahra, S. B., Khan, M. A., & Hasan, M. K. (2022). Alzheimer disease detection empowered with transfer learning. Computers, Materials and Continua, 70(3), 5005–5019. https://doi.org/10.32604/cmc.2022.020866

[2] Venugopalan, J., Tong, L., Hassanzadeh, H. R., & Wang, M. D. (2021). Multimodal deep learning models for early detection of Alzheimer’s disease stage. Scientific Reports, 11(1). https://doi.org/10.1038/s41598-020-74399-w

[3] Helaly, H. A., Badawy, M., & Haikal, A. Y. (2022). Deep Learning Approach for Early Detection of Alzheimer’s Disease. Cognitive Computation, 14(5), 1711–1727. https://doi.org/10.1007/s12559-021-09946-2

[4] Salehi, A. W., Baglat, P., & Gupta, G. (2020). Alzheimer’s Disease Diagnosis using Deep Learning Techniques. International Journal of Engineering and Advanced Technology, 9(3), 874–880. https://doi.org/10.35940/ijeat.C5345.029320

[5] Muhammed Raees, P. C., & Thomas, V. (2021). Automated detection of Alzheimer’s Disease using Deep Learning in MRI. Journal of Physics: Conference Series, 1921(1). https://doi.org/10.1088/1742-6596/1921/1/012024

[6] A, A., M, P., Hamdi, M., Bourouis, S., Rastislav, K., & Mohmed, F. (2022). Evaluation of Neuro Images for the Diagnosis of Alzheimer’s Disease Using Deep Learning Neural Network. Frontiers in Public Health, 10, 834032. https://doi.org/10.3389/fpubh.2022.834032

[7] Liu, S., Masurkar, A. v., Rusinek, H., Chen, J., Zhang, B., Zhu, W., Fernandez-Granda, C., & Razavian, N. (2022). Generalizable deep learning model for early Alzheimer’s disease detection from structural MRIs. Scientific Reports, 12(1). https://doi.org/10.1038/s41598-022-20674-x

[8] Bringas, S., Salomón, S., Duque, R., Lage, C., & Montaña, J. L. (2020). Alzheimer’s Disease stage identification using deep learning models. Journal of Biomedical Informatics, 109. https://doi.org/10.1016/j.jbi.2020.103514

[9] Raza, Noman, Asma Naseer, Maria Tamoor, and Kashif Zafar. 2023. "Alzheimer Disease Classification through Transfer Learning Approach" Diagnostics 13, no. 4: 801. https://doi.org/10.3390/diagnostics13040801

[10] S. Alatrany, W. Khan, A. J. Hussain, J. Mustafina, D. Al-Jumeily and for the Alzheimer's Disease Neuroimaging Initiative, "Transfer Learning for Classification of Alzheimer's Disease Based on Genome Wide Data," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 20, no. 5, pp. 2700-2711, 1 Sept.-Oct. 2023, doi: 10.1109/TCBB.2022.3233869.

[11] Shukla, A., Tiwari, R. & Tiwari, S. Alz-ConvNets for Classification of Alzheimer Disease Using Transfer Learning Approach. SN COMPUT. SCI. 4, 404 (2023). https://doi.org/10.1007/s42979-023-01853-7

[12] Mahmud, T. et al. (2023). Exploring Deep Transfer Learning Ensemble for Improved Diagnosis and Classification of Alzheimer’s Disease. In: Liu, F., Zhang, Y., Kuai, H., Stephen, E.P., Wang, H. (eds) Brain Informatics. BI 2023. Lecture Notes in Computer Science(), vol 13974. Springer, Cham. https://doi.org/10.1007/978-3-031-43075-6_10

[13] Mujahid, Muhammad, Amjad Rehman, Teg Alam, Faten S. Alamri, Suliman Mohamed Fati, and

Tanzila Saba. 2023. "An Efficient Ensemble Approach for Alzheimer's Disease Detection Using an Adaptive Synthetic Technique and Deep Learning" Diagnostics 13, no. 15: 2489. https://doi.org/10.3390/diagnostics13152489

[14] Nikhil J. Dhinagar, Sophia I. Thomopoulos, Priya Rajagopalan, Dimitris Stripelis, Jose Luis Ambite, Greg Ver Steeg, and Paul M. Thompson "Evaluation of transfer learning methods for detecting Alzheimer's disease with brain MRI", Proc. SPIE 12567, 18th International Symposium on Medical Information Processing and Analysis, 125671L (6 March 2023); https://doi.org/10.1117/12.2670457

[15] EL-Geneedy, M., Moustafa, H. E. D., Khalifa, F., Khater, H., & AbdElhalim, E. (2023). An MRI-based deep learning approach for accurate detection of Alzheimer's disease. Alexandria Engineering Journal, 63, 211-221. https://doi.org/10.1016/j.aej.2022.07.062

[16] Miltiadous, E. Gionanidis, K. D. Tzimourta, N. Giannakeas and A. T. Tzallas, "DICE-Net: A Novel Convolution-Transformer Architecture for Alzheimer Detection in EEG Signals," in IEEE Access, vol. 11, pp. 71840-71858, 2023, doi: 10.1109/ACCESS.2023.3294618.

[17] Gopinadhan, Archana. "AD-TL: ALZHEIMER'S DISEASE PREDICTION USING TRANSFER LEARNING." Journal of Data Acquisition and Processing 38.2 (2023): 3290. https://sjcjycl.cn/ DOI: 10.5281/zenodo.777179

[18] Ostertag, C., Visani, M., Urruty, T. et al. Long-term cognitive decline prediction based on multi-modal data using Multimodal3DSiameseNet: transfer learning from Alzheimer's disease to Parkinson's disease. Int J CARS 18, 809–818 (2023). https://doi.org/10.1007/s11548-023-02866-6