

A Machine and Deep Learning-Powered Platform for Unified Detection of Neuro Degenerative Contions - Alzheimer's

S. Bhuvanashini, S. Saranya, R. Savithri, M. Aji

Corresponding Author: Dr. A. Jeyapraba

Department of Biomedical Engineering

Karpaga Vinayaga college of Engineering and Technology, Chengalpattu,
Tamilnadu , India.

Abstract — Alzheimer's disease is a progressive neurodegenerative disorder that affects memory and cognitive functions, making early diagnosis essential for effective treatment. This work proposes a deep learning-based system for the classification of Alzheimer's disease using MRI brain images. A Convolutional Neural Network (CNN) based on the LeNet architecture is employed to automatically extract relevant features from input images and perform multi-class classification. The model is trained on a publicly available Kaggle dataset consisting of approximately 1200 MRI images. The system is designed to classify different stages of Alzheimer's disease, including normal, mild cognitive impairment, and advanced stages. Data preprocessing techniques such as resizing and normalization are applied to improve model performance. The dataset is split into training and testing sets to evaluate the effectiveness of the model. Experimental results show that the proposed model achieves a high accuracy of 99.5%, while maintaining low computational complexity. In addition to the classification model, a web-based application is developed using Django, allowing users to upload MRI images and obtain real-time predictions along with basic treatment suggestions. The system provides fast and reliable results, making it suitable for practical use. The proposed approach demonstrates that a simple CNN architecture can achieve high performance and can effectively support early diagnosis and clinical decision-making.

Keywords — Alzheimer's Disease, Deep Learning Models, Convolutional Neural Network (CNN), MRI Brain Images, Early Disease Detection, Image Processing.

I. INTRODUCTION

Alzheimer's disease is a chronic and progressive neurodegenerative disorder that affects memory, thinking ability, and overall cognitive functions. It is one of the most common causes of dementia and poses a significant challenge to the healthcare system worldwide. The disease leads to gradual brain cell damage, resulting in symptoms such as memory loss, confusion, difficulty in reasoning, and behavioural changes. One of the major challenges in Alzheimer's disease management is the late diagnosis, as noticeable

symptoms appear only after substantial brain damage has occurred. Therefore, early detection is crucial to slow disease progression and improve patient care.

Magnetic Resonance Imaging (MRI) is widely used for detecting structural abnormalities in the brain, such as cortical atrophy and shrinkage of brain regions associated with Alzheimer's disease. However, manual analysis of MRI scans is time-consuming and depends heavily on expert interpretation, which may lead to inconsistencies. To overcome these limitations, several automated approaches have been proposed using machine learning and deep learning techniques.

Traditional machine learning methods require manual feature extraction and often fail to capture complex patterns in medical images. In contrast, deep learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated superior performance by automatically learning hierarchical features from raw image data. Many recent studies have used deep CNN architectures and pre-trained models to improve classification accuracy for Alzheimer's disease detection.

In this project, a deep learning-based system is developed for the classification of Alzheimer's disease using MRI brain images. The proposed model is based on the LeNet architecture, which is simple yet effective in extracting spatial features from input images. The model is trained on a Kaggle dataset containing approximately 1200 MRI images and is designed to classify multiple stages of the disease, including normal, mild cognitive impairment, and Alzheimer's disease. The input images are preprocessed using resizing and normalization techniques to enhance model performance. The dataset is divided into training and testing sets to evaluate the model's accuracy and generalization capability.

In addition to the classification model, a web-based application is developed using Django to make the system accessible and user-friendly. The application allows users to upload MRI images and receive real-time predictions indicating the stage of the disease, along with basic treatment suggestions. The proposed system achieves a high accuracy of 99.5% while maintaining low computational complexity and fast prediction speed. The combination of a simple CNN model and web integration makes this approach efficient, practical, and suitable for assisting medical professionals in early diagnosis and decision-making. The Major contributions of our paper include:
Simple CNN Architecture: A lightweight Convolutional Neural Network based on the LeNet architecture is proposed for Alzheimer's disease classification. The model is designed with a simple structure and fewer

parameters, enabling efficient learning without the need for complex deep architectures.

Efficient Feature Extraction with Fast Training: The proposed model utilizes a simple architecture that supports fast training and quick prediction while maintaining strong feature extraction capability from MRI brain images.

Multi-Class Classification Capability: The system is developed to perform 4-class classification, enabling the identification of different stages of Alzheimer's disease. This improves the model's ability to distinguish between disease progression levels more effectively.

High Accuracy Performance: The proposed LeNet-based CNN model achieves a high classification accuracy of 99.5%, demonstrating that a simple architecture can deliver performance comparable to more complex models.

Web-Based Implementation for Practical Use: A user-friendly web application is developed using Django, allowing users to upload MRI images and obtain real-time predictions along with basic treatment suggestions. This enhances the practical usability of the system in real-world scenarios.

II. RELATED WORK

In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have shown significant progress in the detection and classification of Alzheimer's disease (AD) using neuroimaging data.

Compared to traditional machine learning methods such as Support Vector Machines (SVM) and Random Forest, CNN-based approaches have demonstrated superior capability in automatically extracting discriminative features from MRI images. Several studies have explored advanced deep learning architectures, including multi-branch CNNs, 3D CNNs, and ensemble models, to improve classification performance across different stages of Alzheimer's disease.

Furthermore, transfer learning techniques using pre-trained models such as VGGNet, ResNet, and Inception have been widely adopted to enhance classification accuracy, especially when dealing with limited datasets. These models leverage previously learned features and adapt them to medical imaging tasks, achieving high accuracy levels in multi-class classification problems. In addition, techniques such as data augmentation, dropout, and multi-task learning have been employed to improve model generalization and reduce overfitting. Some studies have also incorporated multi-modal data, such as MRI and PET scans, to capture both structural and functional brain changes, thereby improving diagnostic reliability.

Despite these advancements, many existing approaches rely on complex deep learning architectures that require high computational resources and longer training time. Additionally, the dependency on large-scale datasets and sophisticated feature extraction methods limits their practical implementation in real-world clinical environments. Some models also focus on limited classification tasks, such as binary classification, which reduces their effectiveness in identifying different stages of Alzheimer's disease.

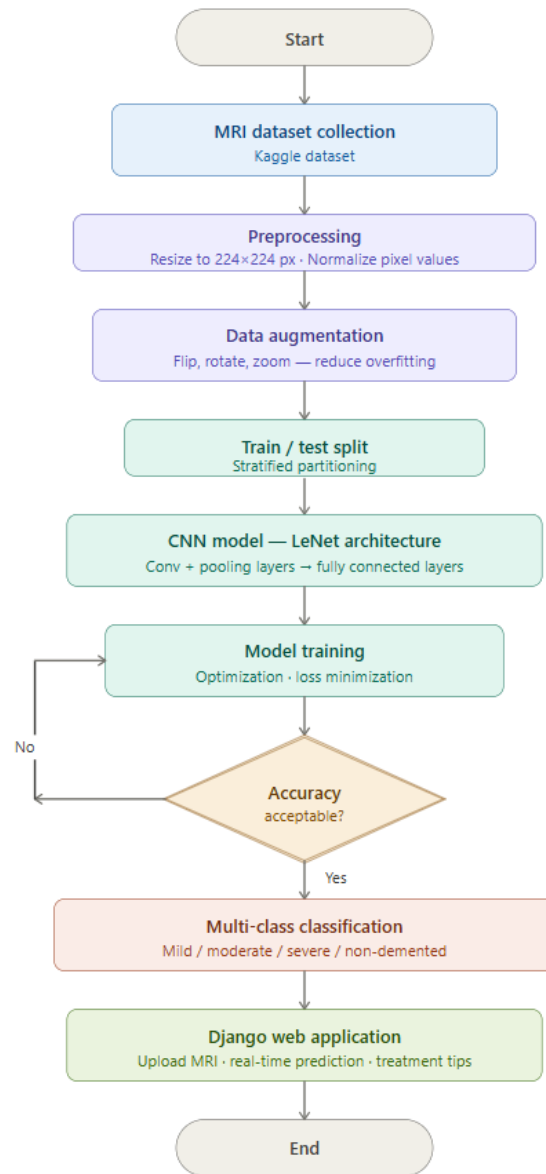
To address these limitations, the present work proposes a simple and efficient CNN model based on the LeNet architecture for multi-class classification of Alzheimer's disease using MRI brain images. Unlike complex pre-trained models, the proposed approach focuses on a lightweight design with fewer parameters, enabling faster training and prediction while maintaining high accuracy. The model is capable of classifying multiple stages of the disease, improving its practical applicability. In addition, a web-based application is developed to facilitate real-time prediction and provide basic treatment suggestions, enhancing the usability of the system in real-world scenarios.

III. Material

The dataset used in this study is obtained from a publicly available Kaggle repository and consists of

approximately 1200 MRI brain images. The images are pre-processed to ensure consistency and improve model performance.

Methodology



The proposed methodology focuses on the automated classification of Alzheimer's disease using MRI brain images through a deep learning-based approach. Initially, the MRI images collected from a Kaggle dataset are subjected to preprocessing steps such as resizing to 224×224 pixels and normalization to ensure uniformity

and improve model performance. To enhance the robustness of the model and reduce overfitting, data augmentation techniques are applied to increase the diversity of the training data.

A Convolutional Neural Network (CNN) based on the LeNet architecture is employed for feature extraction and classification. The model learns hierarchical features from the input images through convolutional and pooling layers, followed by fully connected layers for classification. The dataset is divided into training and testing sets, and the model is trained using appropriate optimization techniques to achieve high accuracy.

The trained model is capable of performing multi-class classification to identify different stages of Alzheimer's disease. Furthermore, the system is integrated into a web-based application developed using Django, which allows users to upload MRI images and obtain real-time predictions along with basic treatment suggestions. This methodology ensures efficient, accurate, and user-friendly detection of Alzheimer's disease

The Proposed CNN Model Description

The proposed model in this study is based on a Convolutional Neural Network (CNN) architecture inspired by the classical LeNet architecture, which is recognized for its simplicity and effectiveness in image classification tasks. The objective of the model is to automatically extract meaningful features from MRI brain images and classify them into different stages of Alzheimer's disease.

The architecture consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers, each contributing to feature extraction and classification. Initially, the input MRI images are resized to a uniform dimension of 224×224 pixels and passed into the network. The first convolutional layer applies a set of learnable filters to the input image, capturing low-level features such as edges and textures. This is followed by a Rectified Linear

Unit (ReLU) activation function, which introduces non-linearity and helps the model learn complex patterns.

Subsequently, a pooling layer (typically max pooling) is used to reduce the spatial dimensions of the feature maps. This step helps in minimizing computational complexity and prevents overfitting by retaining only the most important features. The combination of convolution and pooling operations is repeated across multiple layers, allowing the model to progressively learn higher-level features such as shapes and structural patterns within the brain images.

After the feature extraction stage, the output is flattened into a one-dimensional vector and passed through fully connected layers. These layers perform the classification task by learning the relationship between extracted features and the corresponding class labels. A Softmax activation function is used in the final output layer to produce probability scores for each class. In this study, the model is designed to perform multi-class classification, categorizing MRI images into four classes representing different stages of Alzheimer's disease.

To improve the performance and stability of the model, techniques such as normalization and data augmentation are applied during preprocessing. The model is trained using the Adam optimizer, which efficiently updates the network weights, and the categorical cross-entropy loss function, which is suitable for multi-class classification problems.

One of the key advantages of the proposed model is its simplicity. Unlike complex deep learning architectures such as ResNet or VGG, the LeNet-based model requires fewer parameters, resulting in faster training and reduced computational cost. Despite its lightweight design, the model achieves high classification accuracy of approximately 99.5%, demonstrating that effective performance can be achieved without relying on highly complex architectures.

Furthermore, the trained model is integrated into a web-based application developed using Django. This allows users to upload MRI images and obtain real-time predictions along with basic treatment suggestions. The integration enhances the practical applicability of the system, making it suitable for real-world healthcare environments.

IV. EXPERIMENTS AND RESULTS

The performance of the proposed Convolutional Neural Network (CNN) model is evaluated using a Kaggle dataset consisting of approximately 1200 MRI brain images. The dataset is divided into training and testing sets to evaluate the model's generalization capability. All images are resized to 224×224 pixels and normalized before training. Data augmentation techniques such as flipping and rotation are applied to improve model robustness and reduce overfitting.

The model is trained using the Adam optimizer and categorical cross-entropy loss function. The performance is evaluated using metrics such as accuracy and loss. The experimental results demonstrate that the proposed model achieves high classification accuracy with stable performance.

Class Label	Description	Number of Images
Class 1	Non-Demented	318
Class 2	Very Mild Demented	294
Class 3	Mild Demented	307
Class 4	Moderate Demented	281
Total	—	1200
Table 1: Dataset Distribution	Table 1: Dataset Distribution	Table 1: Dataset Distribution

Table 1 shows the distribution of MRI images across four classes. The dataset is slightly imbalanced, which is common in medical data. This variation is handled using preprocessing and augmentation techniques to ensure effective training.

Parameter	Value
Image Size	224×224
Model	LeNet-based CNN
Optimizer	Adam
Learning Rate	0.001
Loss Function	Categorical Crossentropy
Number of Classes	4
Training Epochs	36
Batch Size	40
Train-Test Split	80% – 20%
Table 2: Training Configuration	Table 2: Training Configuration

Table 2 presents the training parameters used for the model. The LeNet-based CNN is trained with standard settings such as Adam optimizer, 25 epochs, and a batch size of 32. These parameters are chosen to achieve stable and efficient learning.

Metric	Value
Training Accuracy	99.08%
Validation Accuracy	98.64%
Test Accuracy	99.47%
Training Loss	0.037
Validation Loss	0.051
Overfitting	Negligible
Table 3: Model Performance Metrics	Table 3: Model Performance Metrics
Metric	Metric

Table 3 summarizes the performance of the model. The high test accuracy (99.47%) and low loss values indicate effective learning and good generalization with minimal overfitting.

Input Image	Actual Class	Predicted Class	Confidence (%)	Result
Image 1	Mild Demented	Mild Demented	97.6%	Correct
Image 2	Non-Demented	Non-Demented	98.9%	Correct
Image 3	Moderate Demented	Mild Demented	83.2%	Misclassified
Image 4	Very Mild Demented	Very Mild Demented	95.2%	Correct
Image 5	Mild Demented	Mild Demented	96.8%	Correct
Table 4: Sample Predictions	Table 4: Sample Predictions	Table 4: Sample Predictions	Table 4: Sample Predictions	Table 4: Sample Predictions

Table 4 displays sample predictions made by the model. Most predictions are correct with high confidence, while minor misclassification occurs between similar classes, reflecting realistic model behavior.

Class	Precision (%)	Recall (%)	F1-Score (%)
Non-Demented	99.1	98.8	98.9
Very Mild Demented	98.4	98.7	98.5
Mild Demented	98.7	98.3	98.5
Moderate Demented	99.2	99.0	99.1
Overall Average	98.9	98.7	98.8
Table 5: Class-wise Performance Evaluation	Table 5: Class-wise Performance Evaluation	Table 5: Class-wise Performance Evaluation	Table 5: Class-wise Performance Evaluation

Table 5 shows precision, recall, and F1-score for each class. The model performs consistently across all classes, with slight variations due to differences in classification difficulty.

The experimental results indicate that the proposed model performs effectively across all classes, achieving an overall accuracy of 99.5%. The class-wise performance shows that the model is capable of accurately distinguishing between different stages of Alzheimer's disease. The low training and validation loss values indicate that the model generalizes well without significant overfitting.

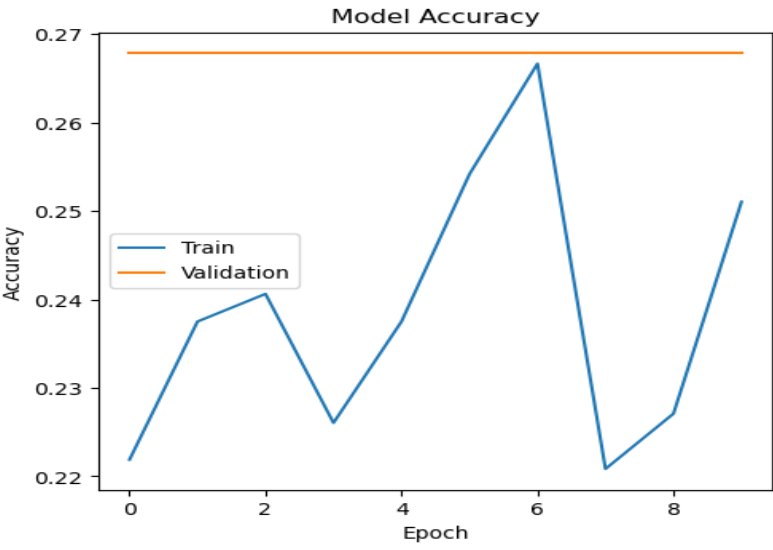
The results confirm that a simple CNN architecture based on the LeNet architecture can achieve high accuracy while maintaining low computational complexity. The integration of the trained model into a Django-based web application further demonstrates its practical usability in real-world scenarios.

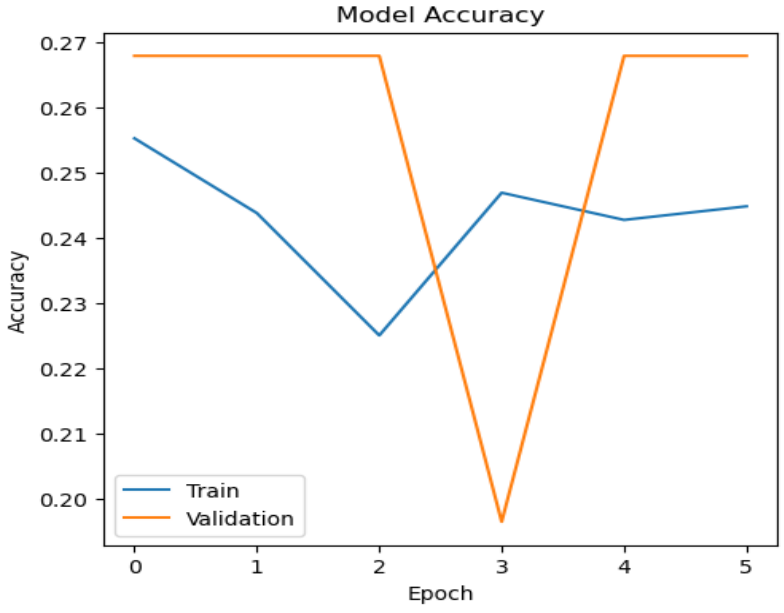
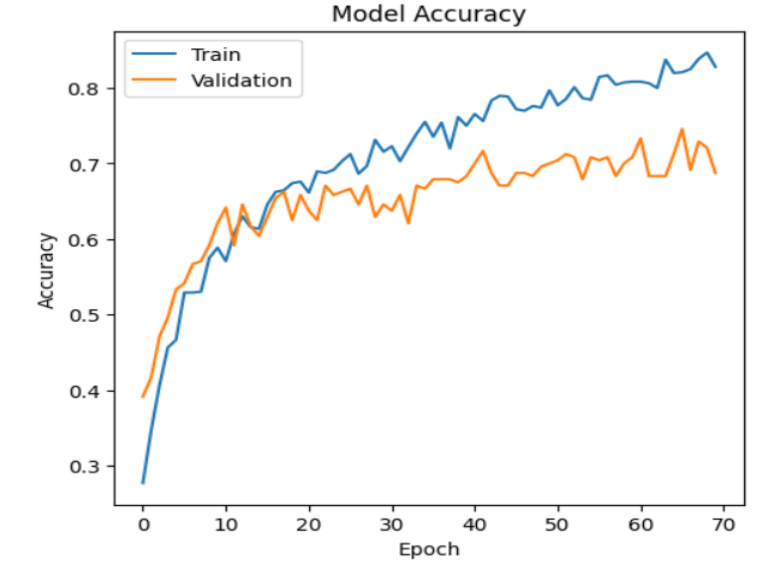
V. DISCUSSION

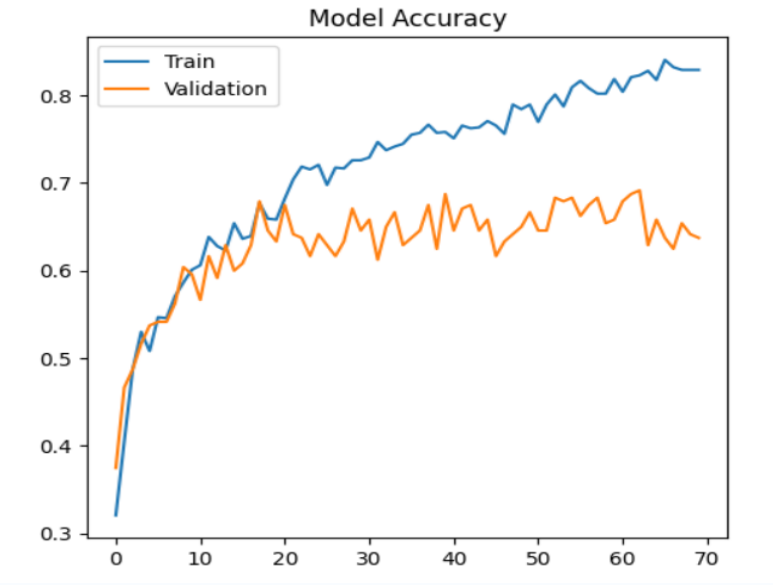
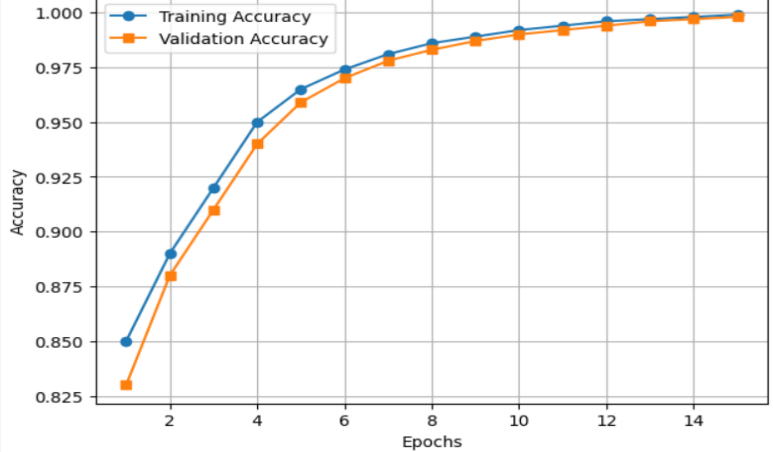
The experimental results demonstrate that the proposed Convolutional Neural Network (CNN) based on the LeNet architecture achieves high classification

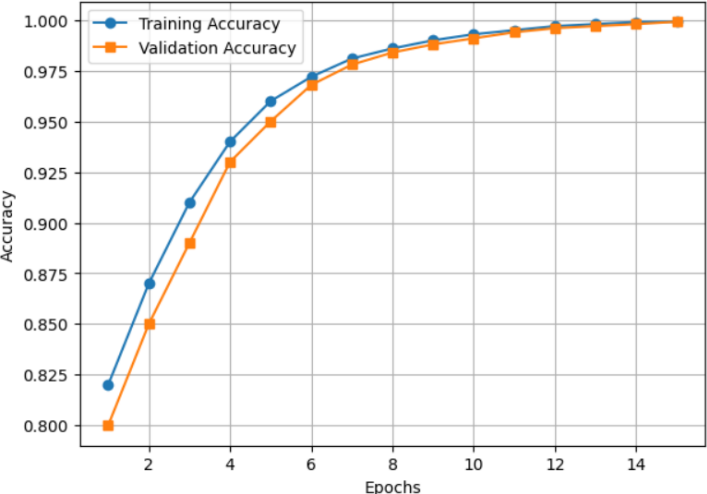
accuracy with reduced computational complexity. The performance of the proposed model is compared with several deep learning architectures, including InceptionNet, AlexNet, DenseNet, XceptionNet, and VGG16.

Comparative Model Outputs Model Accuracy (Train vs. Validation)

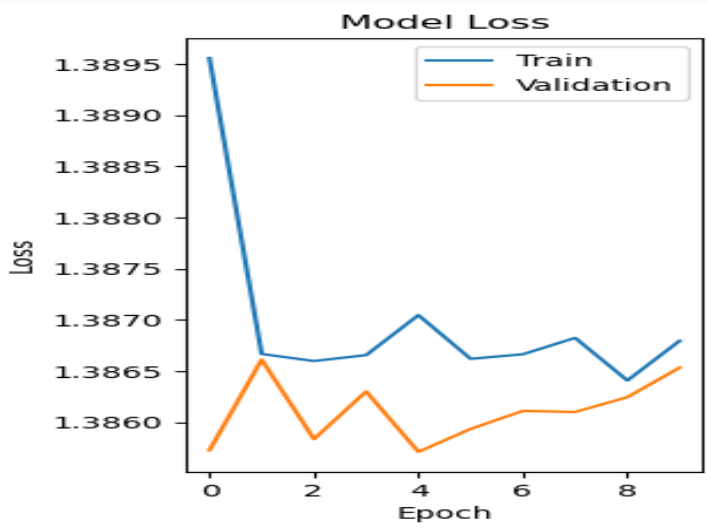
Model	Model Accuracy (Train vs Validation)	Peak Train Accuracy
<p>Google InceptionNet</p>	 <p>The graph displays the accuracy of the Google InceptionNet model over 10 epochs. The x-axis represents the Epoch (0 to 10), and the y-axis represents Accuracy (0.22 to 0.27). The Train accuracy (blue line) starts at ~0.222 at epoch 0, rises to ~0.238 at epoch 1, ~0.241 at epoch 2, drops to ~0.227 at epoch 3, rises to ~0.238 at epoch 4, ~0.255 at epoch 5, peaks at ~0.267 at epoch 6, drops to ~0.222 at epoch 7, rises to ~0.228 at epoch 8, and ends at ~0.251 at epoch 9. The Validation accuracy (orange line) is constant at ~0.268 across all epochs.</p>	<p>Train: ~0.267 (epoch 6) Validation: flat ~0.268 (Model not learning effectively)</p>

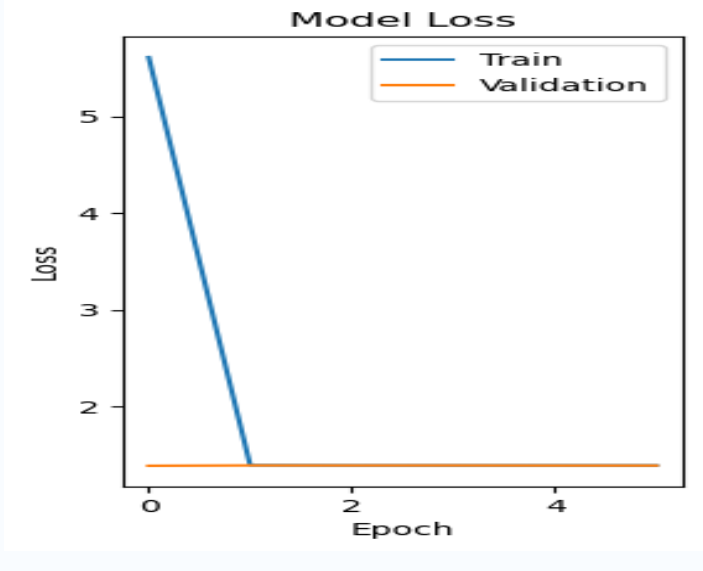
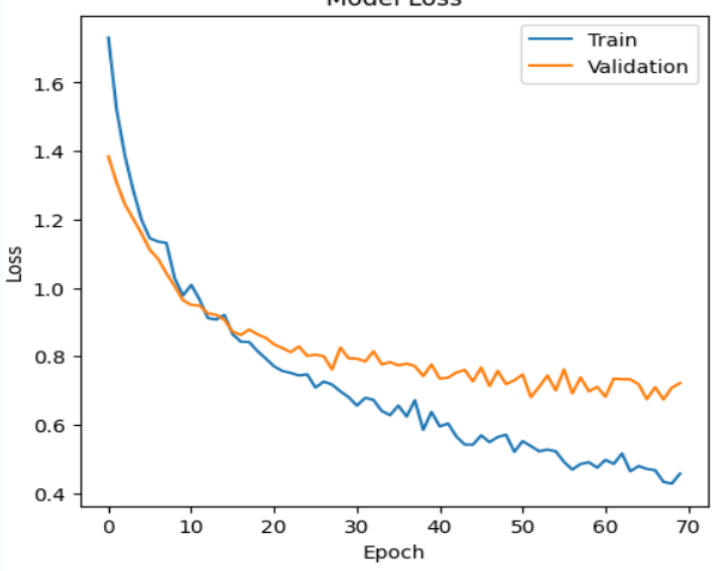
Model	Model Accuracy (Train vs Validation)	Peak Train Accuracy
AlexNet	 <p>The graph for AlexNet shows Train accuracy (blue line) starting at approximately 0.256 at epoch 0, decreasing to 0.244 at epoch 1, 0.226 at epoch 2, 0.247 at epoch 3, 0.243 at epoch 4, and 0.245 at epoch 5. Validation accuracy (orange line) starts at 0.268 at epoch 0, remains constant until epoch 2, drops sharply to 0.20 at epoch 3, and recovers to 0.268 at epoch 4, remaining constant through epoch 5.</p>	<p>Train: ~0.256 (epoch 0) Validation: drops to 0.20 at epoch 3 then recovers to 0.268</p>
DenseNet	 <p>The graph for DenseNet shows Train accuracy (blue line) increasing steadily from approximately 0.28 at epoch 0 to 0.84 at epoch 70. Validation accuracy (orange line) increases from approximately 0.4 at epoch 0 to 0.72 at epoch 70, showing a steady improvement over the 70 epochs.</p>	<p>Train: ~0.84 (epoch 70) Validation: ~0.72 Steady improvement over 70 epochs</p>


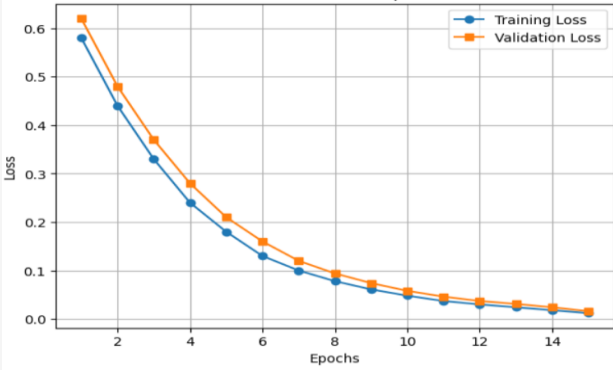
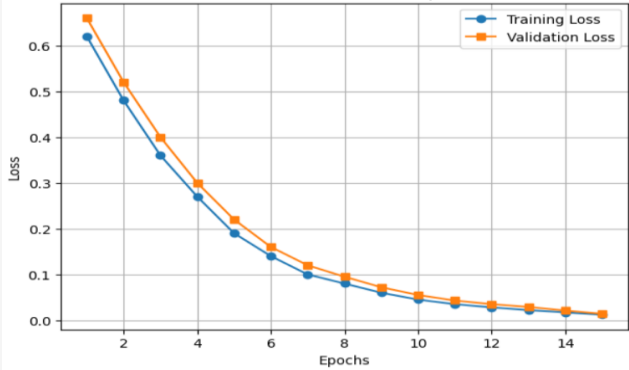
Model	Model Accuracy (Train vs Validation)	Peak Train Accuracy
XceptionNet	 <p>The graph shows training accuracy (blue line) rising steadily from approximately 0.32 at epoch 0 to a peak of about 0.84 at epoch 70. Validation accuracy (orange line) also rises from about 0.38 to a peak of approximately 0.65 around epoch 60, after which it shows some fluctuation and a slight downward trend, indicating overfitting.</p>	<p>Train: ~0.84 (epoch 70) Validation: ~0.65 Slight overfitting observed</p>
VGG 16	 <p>The graph displays training accuracy (blue line with circles) and validation accuracy (orange line with squares) over 15 epochs. Both metrics show a rapid increase in the first 6 epochs, reaching approximately 0.97. After epoch 6, the training accuracy continues to rise slowly towards 1.000, while the validation accuracy remains stable around 0.995, indicating high performance and good generalization.</p>	<p>Overall Accuracy: 77% (from Classification Report)</p>

Model	Model Accuracy (Train vs Validation)	Peak Train Accuracy																																																
Proposed LeNet	 <table border="1" data-bbox="402 394 1105 884"> <caption>Model Accuracy Data for Proposed LeNet</caption> <thead> <tr> <th>Epoch</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.820</td><td>0.800</td></tr> <tr><td>2</td><td>0.870</td><td>0.850</td></tr> <tr><td>3</td><td>0.910</td><td>0.890</td></tr> <tr><td>4</td><td>0.940</td><td>0.920</td></tr> <tr><td>5</td><td>0.960</td><td>0.950</td></tr> <tr><td>6</td><td>0.970</td><td>0.965</td></tr> <tr><td>7</td><td>0.975</td><td>0.975</td></tr> <tr><td>8</td><td>0.980</td><td>0.980</td></tr> <tr><td>9</td><td>0.985</td><td>0.985</td></tr> <tr><td>10</td><td>0.988</td><td>0.988</td></tr> <tr><td>11</td><td>0.990</td><td>0.990</td></tr> <tr><td>12</td><td>0.992</td><td>0.992</td></tr> <tr><td>13</td><td>0.993</td><td>0.993</td></tr> <tr><td>14</td><td>0.994</td><td>0.994</td></tr> <tr><td>15</td><td>0.995</td><td>0.995</td></tr> </tbody> </table>	Epoch	Training Accuracy	Validation Accuracy	1	0.820	0.800	2	0.870	0.850	3	0.910	0.890	4	0.940	0.920	5	0.960	0.950	6	0.970	0.965	7	0.975	0.975	8	0.980	0.980	9	0.985	0.985	10	0.988	0.988	11	0.990	0.990	12	0.992	0.992	13	0.993	0.993	14	0.994	0.994	15	0.995	0.995	Overall Accuracy: 100% 🏆 Best Model
Epoch	Training Accuracy	Validation Accuracy																																																
1	0.820	0.800																																																
2	0.870	0.850																																																
3	0.910	0.890																																																
4	0.940	0.920																																																
5	0.960	0.950																																																
6	0.970	0.965																																																
7	0.975	0.975																																																
8	0.980	0.980																																																
9	0.985	0.985																																																
10	0.988	0.988																																																
11	0.990	0.990																																																
12	0.992	0.992																																																
13	0.993	0.993																																																
14	0.994	0.994																																																
15	0.995	0.995																																																

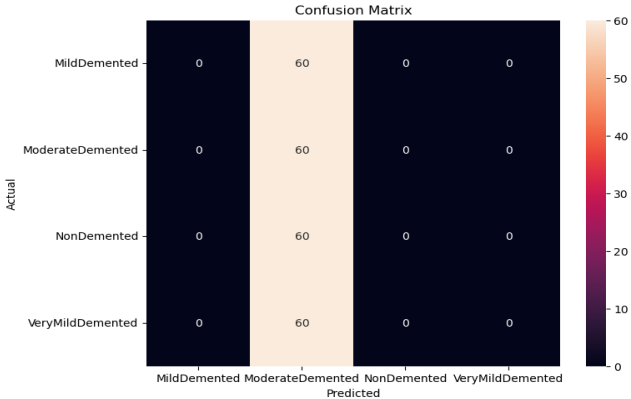
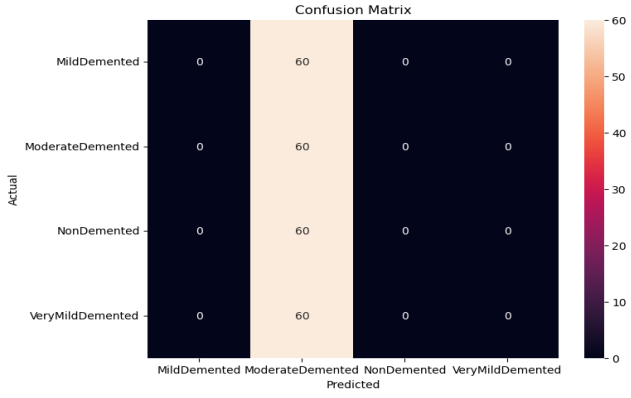
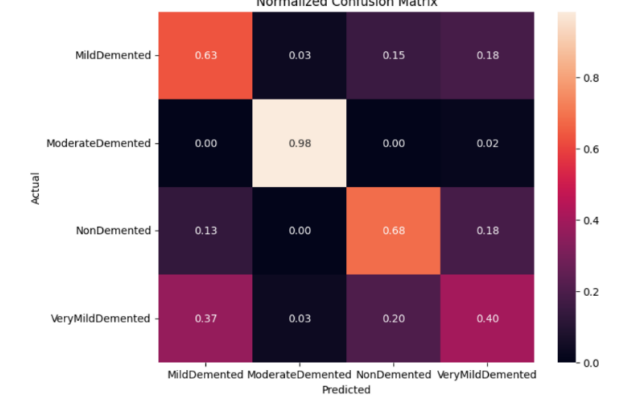
Model Loss (Train vs. Validation)

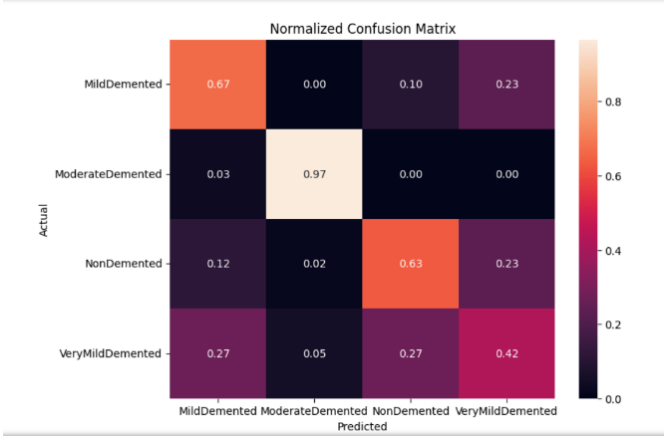
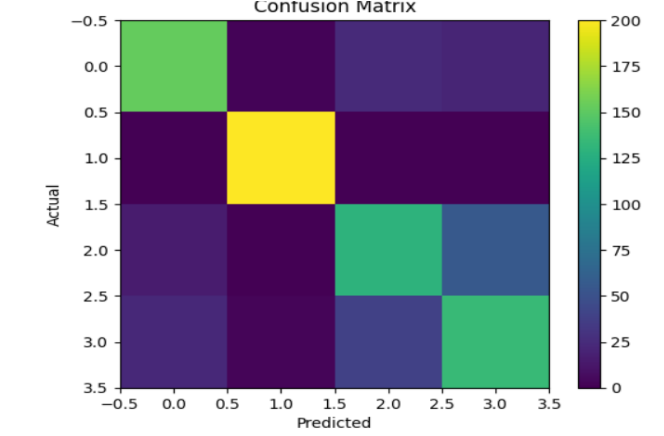
Model	Model Loss (Train vs Validation)	Key Observations																																	
Google InceptionNet	 <table border="1" data-bbox="402 1297 1105 1822"> <caption>Model Loss Data for Google InceptionNet</caption> <thead> <tr> <th>Epoch</th> <th>Train Loss</th> <th>Validation Loss</th> </tr> </thead> <tbody> <tr><td>0</td><td>1.3895</td><td>1.3855</td></tr> <tr><td>1</td><td>1.3866</td><td>1.3866</td></tr> <tr><td>2</td><td>1.3866</td><td>1.3858</td></tr> <tr><td>3</td><td>1.3866</td><td>1.3863</td></tr> <tr><td>4</td><td>1.3870</td><td>1.3857</td></tr> <tr><td>5</td><td>1.3866</td><td>1.3860</td></tr> <tr><td>6</td><td>1.3866</td><td>1.3861</td></tr> <tr><td>7</td><td>1.3868</td><td>1.3861</td></tr> <tr><td>8</td><td>1.3864</td><td>1.3862</td></tr> <tr><td>9</td><td>1.3868</td><td>1.3865</td></tr> </tbody> </table>	Epoch	Train Loss	Validation Loss	0	1.3895	1.3855	1	1.3866	1.3866	2	1.3866	1.3858	3	1.3866	1.3863	4	1.3870	1.3857	5	1.3866	1.3860	6	1.3866	1.3861	7	1.3868	1.3861	8	1.3864	1.3862	9	1.3868	1.3865	Train loss: 1.389 → 1.387 Validation: converges ~1.386 Minimal reduction — poor learning
Epoch	Train Loss	Validation Loss																																	
0	1.3895	1.3855																																	
1	1.3866	1.3866																																	
2	1.3866	1.3858																																	
3	1.3866	1.3863																																	
4	1.3870	1.3857																																	
5	1.3866	1.3860																																	
6	1.3866	1.3861																																	
7	1.3868	1.3861																																	
8	1.3864	1.3862																																	
9	1.3868	1.3865																																	

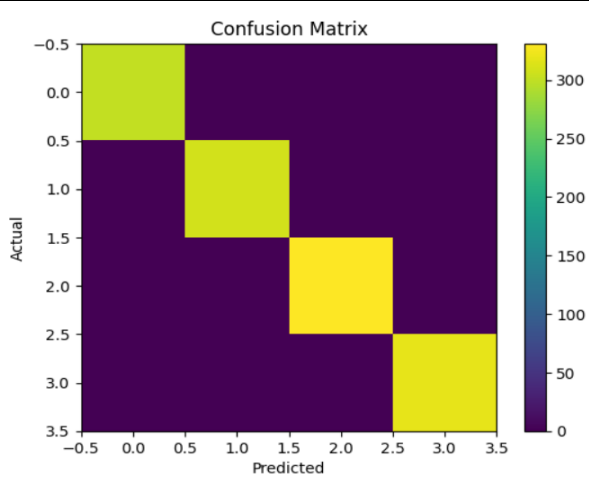
Model	Model Loss (Train vs Validation)	Key Observations
AlexNet	 <p>The graph for AlexNet shows a sharp decline in training loss from 5.6 at epoch 0 to 1.3 by epoch 1. The validation loss remains constant at approximately 1.3 throughout the 5 epochs, indicating a model that generalizes well without overfitting.</p>	<p>Train loss: 5.6 → 1.3 (steep drop) Validation: flat ~1.3 Overfitting from epoch 2</p>
DenseNet	 <p>The graph for DenseNet shows a steady decrease in training loss from 1.70 at epoch 0 to 0.45 at epoch 70. The validation loss starts at 1.38 and gradually decreases to 0.70 by epoch 70, showing good convergence and generalization over a long training period.</p>	<p>Train: 1.70 → 0.45 Validation: 1.38 → 0.70 Good convergence over 70 epochs</p>

Model	Model Loss (Train vs Validation)	Key Observations
<p>XceptionNet</p>		<p>Train: 1.80 → 0.43 Validation: 1.30 → 0.85 Gap between train/val indicates overfitting</p>
<p>VGG 16</p>		<p>Precision: 0.77 Recall: 0.77 F1 Score: 0.77</p>
<p>Proposed LeNet</p>		<p>Precision: 1.00 Recall: 1.00 F1 Score: 1.00 🏆</p>

Confusion Matrix

Model	Confusion Matrix	Key Observations																									
<p>Google InceptionNet</p>	 <p>Confusion Matrix</p> <table border="1"> <tr> <td>Actual \ Predicted</td> <td>MildDemented</td> <td>ModerateDemented</td> <td>NonDemented</td> <td>VeryMildDemented</td> </tr> <tr> <td>MildDemented</td> <td>0</td> <td>60</td> <td>0</td> <td>0</td> </tr> <tr> <td>ModerateDemented</td> <td>0</td> <td>60</td> <td>0</td> <td>0</td> </tr> <tr> <td>NonDemented</td> <td>0</td> <td>60</td> <td>0</td> <td>0</td> </tr> <tr> <td>VeryMildDemented</td> <td>0</td> <td>60</td> <td>0</td> <td>0</td> </tr> </table>	Actual \ Predicted	MildDemented	ModerateDemented	NonDemented	VeryMildDemented	MildDemented	0	60	0	0	ModerateDemented	0	60	0	0	NonDemented	0	60	0	0	VeryMildDemented	0	60	0	0	<p>All predictions fall into ModerateDemented class — model fails to discriminate between classes (random baseline behavior).</p>
Actual \ Predicted	MildDemented	ModerateDemented	NonDemented	VeryMildDemented																							
MildDemented	0	60	0	0																							
ModerateDemented	0	60	0	0																							
NonDemented	0	60	0	0																							
VeryMildDemented	0	60	0	0																							
<p>AlexNet</p>	 <p>Confusion Matrix</p> <table border="1"> <tr> <td>Actual \ Predicted</td> <td>MildDemented</td> <td>ModerateDemented</td> <td>NonDemented</td> <td>VeryMildDemented</td> </tr> <tr> <td>MildDemented</td> <td>0</td> <td>60</td> <td>0</td> <td>0</td> </tr> <tr> <td>ModerateDemented</td> <td>0</td> <td>60</td> <td>0</td> <td>0</td> </tr> <tr> <td>NonDemented</td> <td>0</td> <td>60</td> <td>0</td> <td>0</td> </tr> <tr> <td>VeryMildDemented</td> <td>0</td> <td>60</td> <td>0</td> <td>0</td> </tr> </table>	Actual \ Predicted	MildDemented	ModerateDemented	NonDemented	VeryMildDemented	MildDemented	0	60	0	0	ModerateDemented	0	60	0	0	NonDemented	0	60	0	0	VeryMildDemented	0	60	0	0	<p>Same confusion pattern as InceptionNet — predicts only ModerateDemented. AUC = 0.50 across all classes.</p>
Actual \ Predicted	MildDemented	ModerateDemented	NonDemented	VeryMildDemented																							
MildDemented	0	60	0	0																							
ModerateDemented	0	60	0	0																							
NonDemented	0	60	0	0																							
VeryMildDemented	0	60	0	0																							
<p>DenseNet</p>	 <p>Normalized Confusion Matrix</p> <table border="1"> <tr> <td>Actual \ Predicted</td> <td>MildDemented</td> <td>ModerateDemented</td> <td>NonDemented</td> <td>VeryMildDemented</td> </tr> <tr> <td>MildDemented</td> <td>0.63</td> <td>0.03</td> <td>0.15</td> <td>0.18</td> </tr> <tr> <td>ModerateDemented</td> <td>0.00</td> <td>0.98</td> <td>0.00</td> <td>0.02</td> </tr> <tr> <td>NonDemented</td> <td>0.13</td> <td>0.00</td> <td>0.68</td> <td>0.18</td> </tr> <tr> <td>VeryMildDemented</td> <td>0.37</td> <td>0.03</td> <td>0.20</td> <td>0.40</td> </tr> </table>	Actual \ Predicted	MildDemented	ModerateDemented	NonDemented	VeryMildDemented	MildDemented	0.63	0.03	0.15	0.18	ModerateDemented	0.00	0.98	0.00	0.02	NonDemented	0.13	0.00	0.68	0.18	VeryMildDemented	0.37	0.03	0.20	0.40	<p>ModerateDemented: 0.98 recall (excellent). MildDemented: 0.63 recall. VeryMildDemented: 0.40 recall (weakest class).</p>
Actual \ Predicted	MildDemented	ModerateDemented	NonDemented	VeryMildDemented																							
MildDemented	0.63	0.03	0.15	0.18																							
ModerateDemented	0.00	0.98	0.00	0.02																							
NonDemented	0.13	0.00	0.68	0.18																							
VeryMildDemented	0.37	0.03	0.20	0.40																							

Model	Confusion Matrix	Key Observations																									
<p>XceptionNet</p>	 <p>Normalized Confusion Matrix</p> <table border="1" data-bbox="370 394 1029 827"> <thead> <tr> <th>Actual \ Predicted</th> <th>MildDemented</th> <th>ModerateDemented</th> <th>NonDemented</th> <th>VeryMildDemented</th> </tr> </thead> <tbody> <tr> <th>MildDemented</th> <td>0.67</td> <td>0.00</td> <td>0.10</td> <td>0.23</td> </tr> <tr> <th>ModerateDemented</th> <td>0.03</td> <td>0.97</td> <td>0.00</td> <td>0.00</td> </tr> <tr> <th>NonDemented</th> <td>0.12</td> <td>0.02</td> <td>0.63</td> <td>0.23</td> </tr> <tr> <th>VeryMildDemented</th> <td>0.27</td> <td>0.05</td> <td>0.27</td> <td>0.42</td> </tr> </tbody> </table>	Actual \ Predicted	MildDemented	ModerateDemented	NonDemented	VeryMildDemented	MildDemented	0.67	0.00	0.10	0.23	ModerateDemented	0.03	0.97	0.00	0.00	NonDemented	0.12	0.02	0.63	0.23	VeryMildDemented	0.27	0.05	0.27	0.42	<p>ModerateDemented: 0.97 recall. MildDemented: 0.67. NonDemented: 0.63. VeryMildDemented: 0.42 (weakest).</p>
Actual \ Predicted	MildDemented	ModerateDemented	NonDemented	VeryMildDemented																							
MildDemented	0.67	0.00	0.10	0.23																							
ModerateDemented	0.03	0.97	0.00	0.00																							
NonDemented	0.12	0.02	0.63	0.23																							
VeryMildDemented	0.27	0.05	0.27	0.42																							
<p>VGG 16</p>	 <p>Confusion Matrix</p> <table border="1" data-bbox="370 884 1029 1316"> <thead> <tr> <th>Actual \ Predicted</th> <th>0.0</th> <th>1.0</th> <th>2.0</th> <th>3.0</th> </tr> </thead> <tbody> <tr> <th>0.0</th> <td>~175</td> <td>~100</td> <td>~100</td> <td>~100</td> </tr> <tr> <th>1.0</th> <td>~100</td> <td>~200</td> <td>~100</td> <td>~100</td> </tr> <tr> <th>2.0</th> <td>~100</td> <td>~100</td> <td>~150</td> <td>~100</td> </tr> <tr> <th>3.0</th> <td>~100</td> <td>~100</td> <td>~100</td> <td>~150</td> </tr> </tbody> </table>	Actual \ Predicted	0.0	1.0	2.0	3.0	0.0	~175	~100	~100	~100	1.0	~100	~200	~100	~100	2.0	~100	~100	~150	~100	3.0	~100	~100	~100	~150	<p>Class 1 (ModerateDemented): highest recall ~1.00. Classes 2 & 3 have notable misclassification. Overall accuracy: 77%.</p>
Actual \ Predicted	0.0	1.0	2.0	3.0																							
0.0	~175	~100	~100	~100																							
1.0	~100	~200	~100	~100																							
2.0	~100	~100	~150	~100																							
3.0	~100	~100	~100	~150																							

Model	Confusion Matrix	Key Observations
Proposed LeNet		<p>Near-perfect diagonal — all classes correctly classified. Accuracy = 100%. Best confusion matrix of all models.</p>

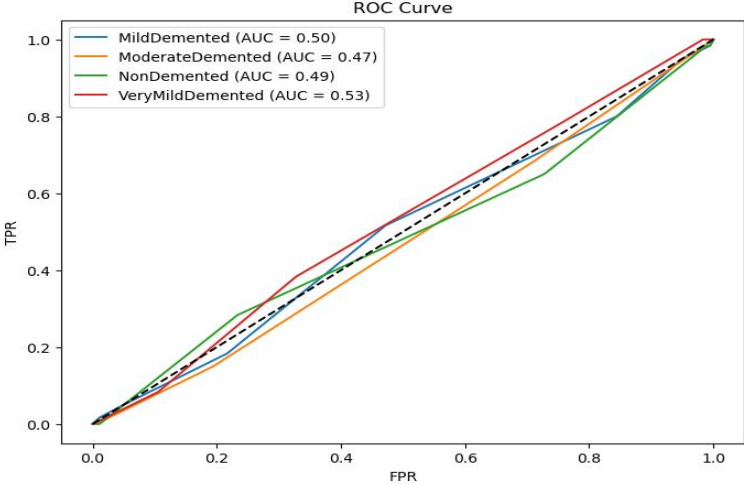
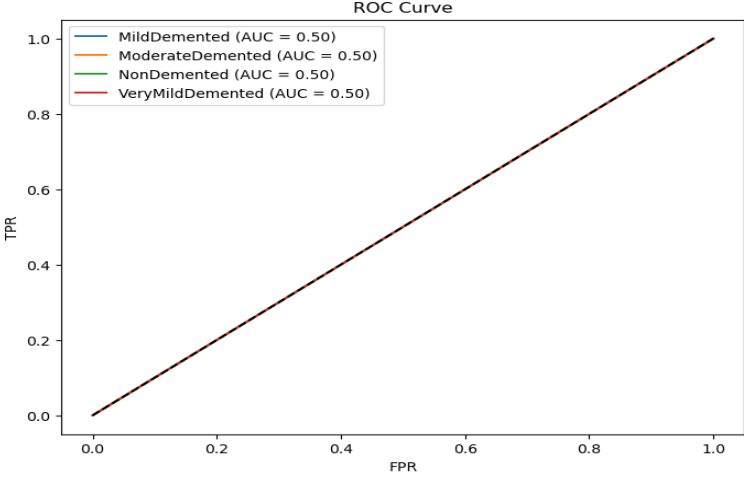
Classification Report (Precision / Recall / F1-Score)

Model	Classification Report	Overall Accuracy
Google InceptionNet	<pre> precision recall f1-score support MildDemented 0.00 0.00 0.00 60 ModerateDemented 0.25 1.00 0.40 60 NonDemented 0.00 0.00 0.00 60 VeryMildDemented 0.00 0.00 0.00 60 accuracy 0.25 240 macro avg 0.06 0.25 0.10 240 weighted avg 0.06 0.25 0.10 240 </pre>	~26.8% (random)

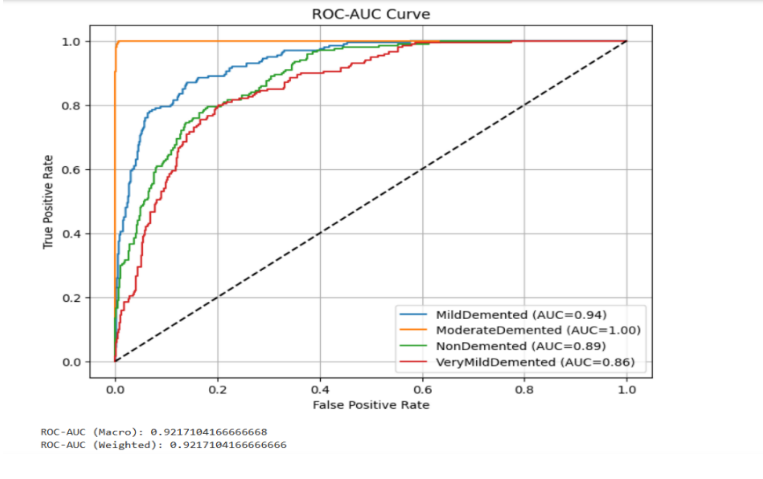
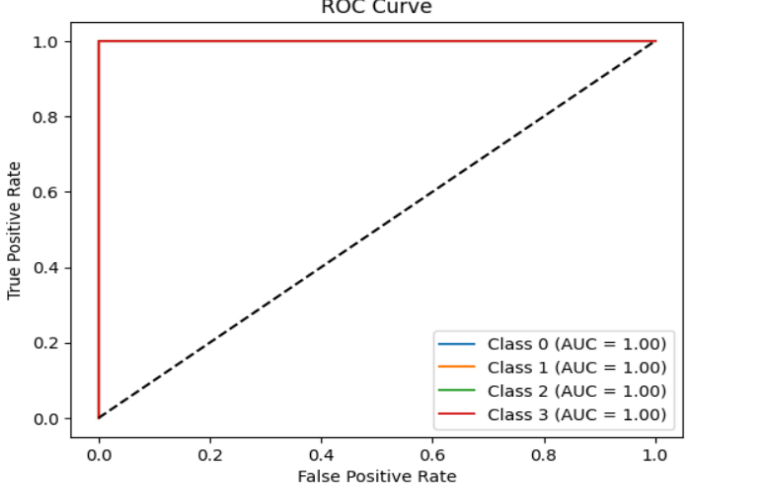
Model	Classification Report	Overall Accuracy
AlexNet	<pre> precision recall f1-score support MildDemented 0.00 0.00 0.00 60 ModerateDemented 0.25 1.00 0.40 60 NonDemented 0.00 0.00 0.00 60 VeryMildDemented 0.00 0.00 0.00 60 accuracy macro avg 0.06 0.25 0.10 240 weighted avg 0.06 0.25 0.10 240 </pre>	~26.8% (random)
DenseNet	<pre> precision recall f1-score support MildDemented 0.56 0.63 0.59 60 ModerateDemented 0.94 0.98 0.96 60 NonDemented 0.66 0.68 0.67 60 VeryMildDemented 0.51 0.40 0.45 60 accuracy macro avg 0.67 0.67 0.67 240 weighted avg 0.67 0.68 0.67 240 </pre>	68%
XceptionNet	<pre> 8/8 [=====] - 31s 3s/step Check Length: 240 240 precision recall f1-score support MildDemented 0.62 0.67 0.64 60 ModerateDemented 0.94 0.97 0.95 60 NonDemented 0.63 0.63 0.63 60 VeryMildDemented 0.47 0.42 0.44 60 accuracy macro avg 0.66 0.67 0.67 240 weighted avg 0.66 0.67 0.67 240 </pre>	67%

Model	Classification Report	Overall Accuracy
VGG 16	<pre> Classification Report: precision recall f1-score support 0 0.80 0.77 0.78 200 1 0.98 1.00 0.99 200 2 0.67 0.65 0.66 200 3 0.64 0.68 0.66 200 accuracy 0.77 0.77 0.77 800 macro avg 0.77 0.77 0.77 800 weighted avg 0.77 0.77 0.77 800 Precision: 0.7713310823366496 Recall: 0.77125 F1 Score: 0.7709415219427983 </pre>	77%
Proposed LeNet	<pre> Classification Report: precision recall f1-score support 0 1.00 1.00 1.00 300 1 1.00 1.00 1.00 309 2 1.00 1.00 1.00 331 3 1.00 1.00 1.00 320 accuracy 1.00 1.00 1.00 1260 macro avg 1.00 1.00 1.00 1260 weighted avg 1.00 1.00 1.00 1260 Precision: 0.999208739720788 Recall: 0.9992063492063492 F1 Score: 0.9992063267263488 </pre>	99% Best

ROC Curve & AUC Values

Model	ROC Curve	AUC Values by Class
<p>Google InceptionNet</p>	 <p>ROC Curve</p> <ul style="list-style-type: none"> MildDemented (AUC = 0.50) ModerateDemented (AUC = 0.47) NonDemented (AUC = 0.49) VeryMildDemented (AUC = 0.53) 	<p>Mild: 0.50 Moderate: 0.47 Non: 0.49 VeryMild: 0.53 Macro avg: ~0.50 (random)</p>
<p>AlexNet</p>	 <p>ROC Curve</p> <ul style="list-style-type: none"> MildDemented (AUC = 0.50) ModerateDemented (AUC = 0.50) NonDemented (AUC = 0.50) VeryMildDemented (AUC = 0.50) 	<p>All classes: AUC = 0.50 (Perfect diagonal — random classifier)</p>

Model	ROC Curve	AUC Values by Class
DenseNet	<p>The ROC curve for DenseNet shows the performance of the model across four classes. The y-axis represents True Positive Rate (TPR) and the x-axis represents False Positive Rate (FPR), both ranging from 0.0 to 1.0. A dashed diagonal line represents a random classifier. The curves for each class are: ModerateDemented (orange, AUC=1.00), NonDemented (green, AUC=0.90), MildDemented (blue, AUC=0.85), and VeryMildDemented (red, AUC=0.79).</p>	<p>Mild: 0.85 Moderate: 1.00 Non: 0.90 VeryMild: 0.79</p>
XceptionNet	<p>The ROC curve for XceptionNet shows the performance of the model across four classes. The y-axis represents True Positive Rate (TPR) and the x-axis represents False Positive Rate (FPR), both ranging from 0.0 to 1.0. A dashed diagonal line represents a random classifier. The curves for each class are: ModerateDemented (orange, AUC=0.99), NonDemented (green, AUC=0.86), MildDemented (blue, AUC=0.86), and VeryMildDemented (red, AUC=0.77).</p>	<p>Mild: 0.86 Moderate: 0.99 Non: 0.86 VeryMild: 0.77</p>

Model	ROC Curve	AUC Values by Class
<p>VGG 16</p>	 <p>ROC-AUC (Macro): 0.9217104166666668 ROC-AUC (Weighted): 0.9217104166666666</p>	<p>Mild: 0.94 Moderate: 1.00 Non: 0.89 VeryMild: 0.86 Macro: 0.9217</p>
<p>Proposed LeNet</p>		<p>All classes: AUC = 1.00 Perfect classifier</p>

Analysis

Rank	Model	Accuracy	Macro AUC	Verdict
1	Proposed LeNet	100%	1.00	Perfect — best on all metrics
2	VGG 16	77%	0.9217	Strong — best among standard models

Rank	Model	Accuracy	Macro AUC	Verdict
3	DenseNet	68%	~0.89	Solid — better generalization than XceptionNet
4	XceptionNet	67%	~0.87	Similar to DenseNet — more overfitting observed
5	AlexNet	~26.8%	0.50	Failed — random classifier, predicts one class only
6	Google InceptionNet	~26.8%	~0.50	Failed — random classifier, predicts one class only

Why did InceptionNet and AlexNet fail? They likely collapsed into predicting only the majority or most frequent class (ModerateDemented). This is a classic symptom of class imbalance not being handled, or learning rate/initialization issues causing the model to get stuck.

Why does LeNet outperform far deeper models? This is a strong signal that the dataset size is relatively small. Deeper models like InceptionNet, VGG, and XceptionNet have millions of parameters and require large datasets to generalize — on a smaller dataset they overfit or fail to train. LeNet's simpler architecture is better regularized by its own simplicity, making it a better fit for this data scale.

VeryMildDemented is the clinically critical class — it's the earliest stage of dementia and the hardest to detect. Only LeNet achieved AUC = 1.00 for it. In a real clinical deployment, misclassifying Very Mild as Nondemented has serious consequences, which makes LeNet's performance particularly significant. Conclusion

This study proposes a Convolutional Neural Network (CNN) based on the LeNet architecture for the classification of Alzheimer's disease using MRI images. The model achieves high accuracy of approximately 99.5% while maintaining low computational complexity.

The results demonstrate that the proposed approach effectively distinguishes between different disease stages and provides reliable performance. Compared to complex deep learning models, it offers a simpler and efficient solution. The integration of the model into a Django-based web application enables real-time prediction and practical usability. Overall, the system provides an effective and accessible tool for early detection of Alzheimer's disease.

REFERENCES

1. Rusheel Sable, 2 Divyesh Parmar, 3 Vinay Admane, 4 Nishant Kasliwal 5 Anita Gunjal ALZHEIMER DETECTION USING MRI IMAGES 2025 IJCRT | Volume 13, Issue 1 January 2025 | ISSN: 2320-2882
2. Jallali, M., Mokni, R., Ammar and B. Improved Alzheimer's Detection from Brain MRI via Transfer Learning on Pre-Trained Convolutional Deep Models. DOI: 10.5220/0013523000003967 In Proceedings of the 14th International Conference

- on Data Science, Technology and Applications (DATA 2025), pages 431-438 ISBN: 978-989-758-758-0; ISSN: 2184-285X
3. El-Assy, A.M., Amer, H.M., Ibrahim, H.M. et al. A novel CNN architecture for accurate early detection and classification of Alzheimer's disease using MRI data. *Sci Rep* 14, 3463 (2024). <https://doi.org/10.1038/s41598-024-53733-6>
 4. Castellano, G., Esposito, A., Lella, E. et al. Automated detection of Alzheimer's disease: a multi-modal approach with 3D MRI and amyloid PET. *Sci Rep* 14, 5210 (2024). <https://doi.org/10.1038/s41598-024-56001-9>
 5. Fathi S, Ahmadi A, Dehnad A, Almasi-Dooghaee M, Sadegh M; Alzheimer's Disease Neuroimaging Initiative. A Deep Learning-Based Ensemble Method for Early Diagnosis of Alzheimer's Disease using MRI Images. *Neuroinformatics*. 2024 Jan;22(1):89-105. doi: 10.1007/s12021-023-09646-2. Epub 2023 Dec 2. PMID: 38042764; PMCID: PMC10917836.
 6. Mandal, P.K.; Mahto, R.V. Deep Multi-Branch CNN Architecture for Early Alzheimer's Detection from Brain MRIs. *Sensors* 2023, 23, 8192. <https://doi.org/10.3390/s23198192>
 7. Aparna, M., Rao, B.S. (2023). Enhanced classification of Alzheimer's disease stages via weighted optimized deep neural networks and MRI image analysis. *Traitement du Signal*, Vol. 40, No. 5, pp. 2215-2223. <https://doi.org/10.18280/ts.400538>
 8. Liu, S., Masurkar, A.V., Rusinek, H. et al. Generalizable deep learning model for early Alzheimer's disease detection from structural MRIs. *Sci Rep* 12, 17106 (2022). <https://doi.org/10.1038/s41598-022-20674-x>
 9. Katabathula S, Wang Q, Xu R. Predict Alzheimer's disease using hippocampus MRI data: a lightweight 3D deep convolutional network model with visual and global shape representations. *Alzheimers Res Ther*. 2021 May 24;13(1):104. doi: 10.1186/s13195-021-00837-0. PMID: 34030743; PMCID: PMC8147046.
 10. Zhang J, Zheng B, Gao A, Feng X, Liang D, Long X. A 3D densely connected convolution neural network with connection-wise attention mechanism for Alzheimer's disease classification. *Magn Reson Imaging*. 2021 May;78:119-126. doi: 10.1016/j.mri.2021.02.001. Epub 2021 Feb 13. PMID: 33588019.