POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Numerical Analysis for Machine Learning Project

Author(s): **Hosein Mirhoseini - 11046456**

Ali Nabipour Dargah - 11041035

Academic Year: 2024-2025

Contents

1 | Introduction

Alzheimer's Disease (AD) is the most common form of dementia, affecting millions of individuals worldwide. It is characterized by progressive neurodegeneration, leading to severe cognitive, sensory, and motor deficits. Early and accurate diagnosis of AD is crucial for effective intervention and management, as it can slow the progression of the disease. However, current diagnostic methods, which rely heavily on radiologist interpretation of neuroimaging data such as Magnetic Resonance Imaging (MRI), are prone to human error and inconsistency. This has spurred significant interest in leveraging advanced computational techniques, particularly deep learning, to improve the accuracy and reliability of AD diagnosis.

In recent years, deep learning models, especially Convolutional Neural Networks (CNNs), have shown remarkable promise in the field of medical imaging and AD classification. These models can automatically learn complex patterns from neuroimaging data, making them highly effective for tasks such as distinguishing between healthy individuals, those with Mild Cognitive Impairment (MCI), and those with AD. However, many existing deep learning approaches for AD diagnosis focus on general brain atrophy and fail to capture the focal structural changes that are critical for understanding the progression of neurodegeneration in specific brain regions.

This project is based on the work presented in the article *"Structure focused neurodegeneration convolutional neural network for modelling and classification of Alzheimer's disease"* by Odimayo et al. (2024). The authors propose a novel deep learning framework that includes a specialized CNN architecture, termed SNeurodCNN, designed to model the structural neurodegeneration of the brain's cerebral cortex. The SNeurodCNN architecture focuses on capturing focal structural atrophy features from MRI scans, which are crucial for distinguishing between MCI and AD. Additionally, the framework incorporates a preprocessing step using gamma correction to enhance image brightness, further improving the model's performance.

The SNeurodCNN architecture is unique in that it is specifically tailored to capture the structural dynamics of neurodegeneration in key brain regions such as the frontal lobe,

occipital lobe, cerebellum, and parietal lobe. By leveraging mid-sagittal and para-sagittal MRI viewpoints from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, the authors demonstrate that SNeurodCNN achieves state-of-the-art performance, with classification accuracies of 98.1% and 97.8% for mid-sagittal and para-sagittal views, respectively. The model's ability to identify specific regions of neurodegeneration provides valuable insights into the early stages of AD, making it a potential digital biomarker for early diagnosis.

In this project, we aim to explore and analyze the numerical and computational aspects of the SNeurodCNN model. Our focus will be on understanding the mathematical foundations of the model, including the convolutional layers, regularization techniques, and the optimization process. We will also investigate the impact of preprocessing techniques, such as gamma correction, on the model's performance. Furthermore, we will compare the SNeurodCNN architecture with other state-of-the-art deep learning models to evaluate its effectiveness in AD classification.

The remainder of this report is organized as follows:

- **Section 2**: Provides a detailed overview of the SNeurodCNN architecture and its components.
- **Section 3**: Discusses how we implement this article and tried to get the results.
- **Section 4**: Presents the experimental results and analysis.

2 | Overview of the SNeurodCNN Architecture

The SNeurodCNN (Structure-focused Neurodegeneration Convolutional Neural Network) is a specialized deep learning architecture designed to model the structural neurodegeneration of the brain's cerebral cortex, particularly for the classification of Alzheimer's Disease (AD) and Mild Cognitive Impairment (MCI). This section provides a detailed overview of the architecture and its key components.

2.1. Architecture Design

The SNeurodCNN architecture is tailored to capture focal structural atrophy features from MRI scans, which are critical for distinguishing between MCI and AD. Unlike traditional deep learning models that rely on pre-trained networks or generic architectures, SNeurodCNN is specifically designed to focus on the structural changes in the brain caused by neurodegeneration. The architecture consists of the following key components:

- **Input Layer**: The input to SNeurodCNN is the structure-focused MRI data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset. The data is preprocessed to correct imaging distortions and enhance brightness using gamma correction.
- **Convolutional Blocks**: The architecture comprises two downsampling convolutional blocks. Each block contains convolutional layers with ReLU activation functions and L2 regularization to prevent overfitting. The first block uses a 3x3 filter with 32 dimensions, while the second block uses a 3x3 filter with 64 dimensions. Max-pooling layers (2x2) are used for downsampling.
- **Fully Connected Layers**: After the convolutional blocks, the output is flattened and passed through two fully connected (dense) layers. The first dense layer contains 500 hidden neurons with ReLU activation, followed by a dropout layer with a 50% drop rate to further regularize the model. The final layer uses a softmax activation

function for binary classification (AD vs. MCI).

2.2. Preprocessing: Gamma Correction

Before feeding the MRI data into the SNeurodCNN, a preprocessing step is applied to enhance the brightness of the images. This is achieved using gamma correction, which adjusts the pixel intensity values to improve the visibility of structural features. The gamma value of 0.2 was found to be optimal for this task. This preprocessing step is crucial for improving the model's ability to capture subtle structural changes in the brain.

2.3. Regularization Techniques

To ensure robust generalization and prevent overfitting, SNeurodCNN employs several regularization techniques:

- **L2 Regularization**: Applied to the convolutional and dense layers to penalize large weights and reduce overfitting.
- **Dropout**: A dropout rate of 50% is used in the fully connected layers to randomly deactivate neurons during training, further enhancing the model's generalization capability.
- **Early Stopping**: Training is halted if the validation loss does not improve for a specified number of epochs, ensuring that the model does not overfit to the training data.

2.4. Visualization of the Architecture

To better understand the architecture, Figure 2.1 provides a visual representation of the SNeurodCNN model. The figure illustrates the flow of data through the convolutional blocks, fully connected layers, and the final classification layer.

2.5. Key Features of SNeurodCNN

The SNeurodCNN architecture has several distinguishing features that make it effective for AD classification:

- **Focus on Structural Atrophy**: Unlike traditional models, SNeurodCNN is designed to capture focal structural changes in the brain, such as atrophy in the frontal

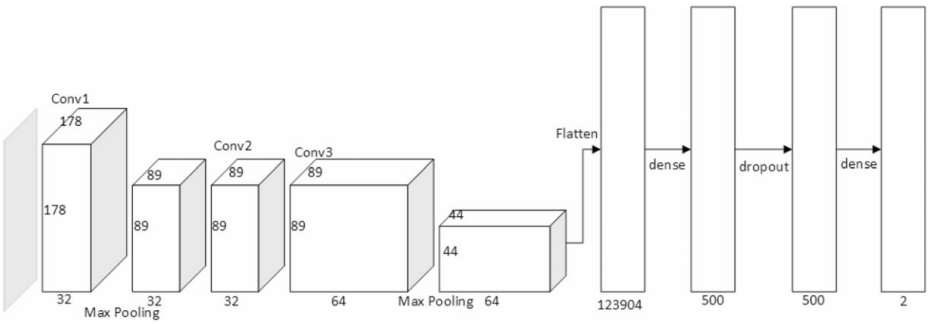

Figure 2.1: Visual representation of the SNeurodCNN architecture

lobe, occipital lobe, cerebellum, and parietal lobe.

- **Lightweight Design**: The architecture uses only two convolutional blocks, making it computationally efficient compared to deeper models like ResNet or DenseNet.
- **High Performance**: SNeurodCNN achieves state-of-the-art performance, with classification accuracies of 98.1% and 97.8% for mid-sagittal and para-sagittal MRI views, respectively.

In summary, the SNeurodCNN architecture is a highly effective model for AD classification, leveraging its focus on structural neurodegeneration, lightweight design, and robust regularization techniques. The next section will discuss the numerical methods and optimization techniques used in the model.

3 | Implementation

For the implementation of the proposed model, we divided the process into four main sections, as described below.

3.1. Preprocessing

During the preprocessing stage, we applied *Gamma Correction* to the MRI data images, as recommended in the referenced paper. The gamma transformation is defined as:

$$I_{\text{out}} = I_{\text{in}}^{\gamma} \tag{3.1}$$

where I_{in} and I_{out} represent the input and output pixel intensities, respectively, and γ is the correction factor. Given the low intensity of certain MRI structures, we used $\gamma = 0.2$ to enhance contrast while preserving structural details. The images were normalized to the range $[0, 1]$ before applying the transformation.

Figure 3.1 illustrates the effect of gamma correction on a representative MRI slice. The left image shows the original intensity distribution, while the right image demonstrates the enhanced contrast after applying gamma correction.

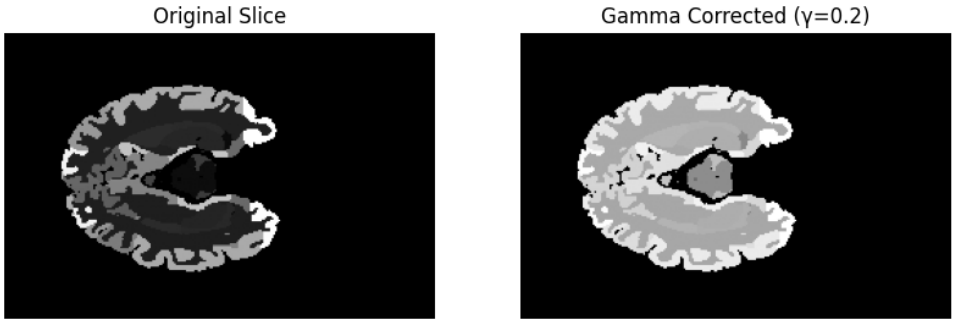

Figure 3.1: Comparison of original and gamma-corrected MRI slices.

3.2. Training

For the implementation, we utilized the *PyTorch* library in Python to construct the SNeurodCNN. The steps involved in the implementation are outlined below:

3.2.1. Reading the Data

The image dataset comprises four distinct classes:

- MIDSAG-ADGAMMA
- MIDSAG-MCIGAMMA
- MTL AD-GAMMA
- MTL-MCI

The dataset contains over 9,700 inputs across the four classes, which were divided into three subsets: training, validation, and test.

1. Training: 80%
2. Validation: 10%
3. Test: 10%

Figure 3.2 illustrates nine sample inputs from the dataset, representing the different classes.

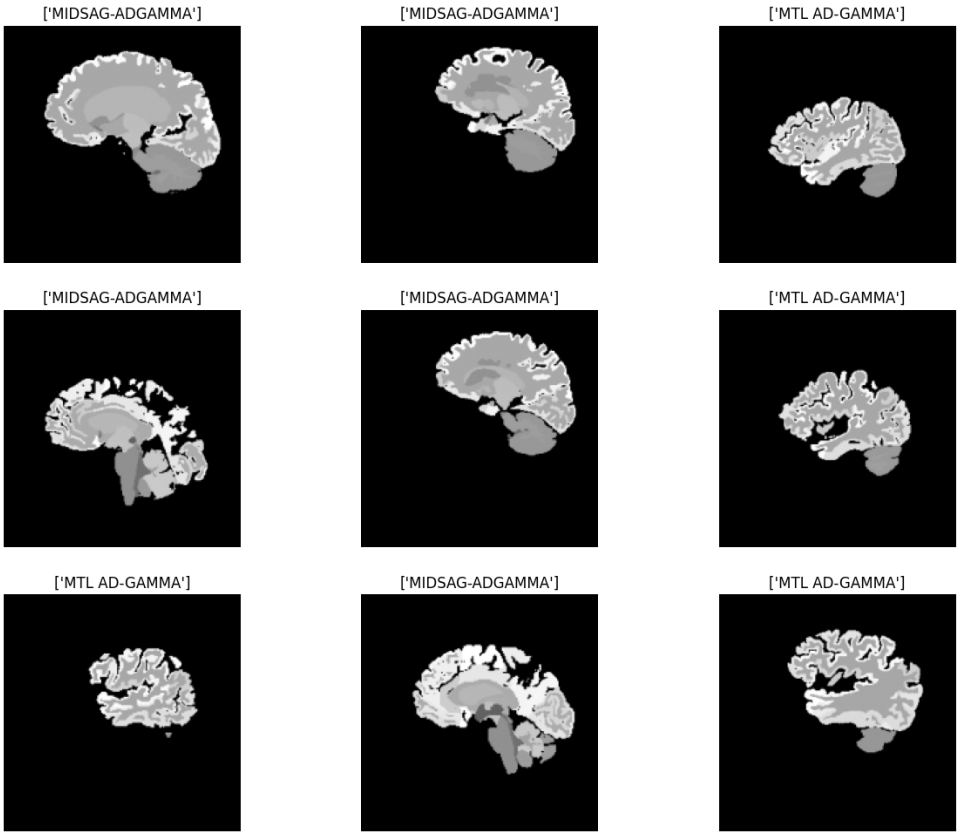

Figure 3.2: Dataset Visualization

3.2.2. DeepCNN Model

Using *PyTorch*, we defined the architecture of the DeepCNN model based on the structure outlined in Section 2. Table 3.1 provides a summary of the model, including its layers, output shapes, and the number of parameters.

3.2.3. Hyperparameters and Implementation Choices

The following hyperparameters and implementation choices were selected for this project:

- **Optimizer**: Adam with a learning rate of 0.0001
- **Loss Function**: Sparse Categorical Cross-entropy
- **Number of Epochs**: 100
- **Batch Size**: 32
- **Metrics**: Accuracy
- **Learning Rate Scheduler**: ReduceLROnPlateau with the following settings:

Table 3.1: Layer details of the SNeurodCNN architecture.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 178, 178]	320
MaxPool2d-2	[-1, 32, 89, 89]	0
Conv2d-3	[-1, 32, 89, 89]	9,248
Conv2d-4	[-1, 64, 89, 89]	18,496
MaxPool2d-5	[-1, 64, 44, 44]	0
Linear-6	[-1, 500]	61,952,500
Dropout-7	[-1, 500]	0
Linear-8	[-1, 4]	2,004
Total params:		61,982,568
Trainable params:		61,982,568
Non-trainable params:		0

Input size (MB):	0.12
Forward/backward pass size (MB):	16.42
Params size (MB):	236.44
Estimated Total Size (MB):	252.99

  - **Monitor**: Validation Loss
  - **Mode**: Min
  - **Factor**: 0.5
  - **Patience**: 3
- **Early Stopping**: The criteria for early stopping were as follows:
  - **Monitor**: Validation Loss
  - **Patience**: 5

3.3. Evaluation

To evaluate the model, we employed two metrics. Figure 3.3 visualizes the training and validation loss per epoch, while Figure 3.4 depicts the training and validation accuracy.

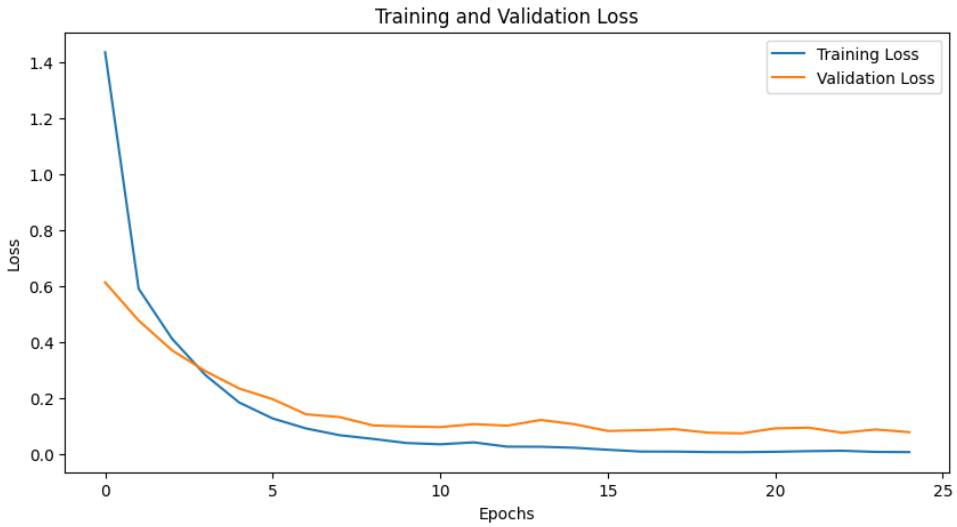

Figure 3.3: Training and Validation Loss

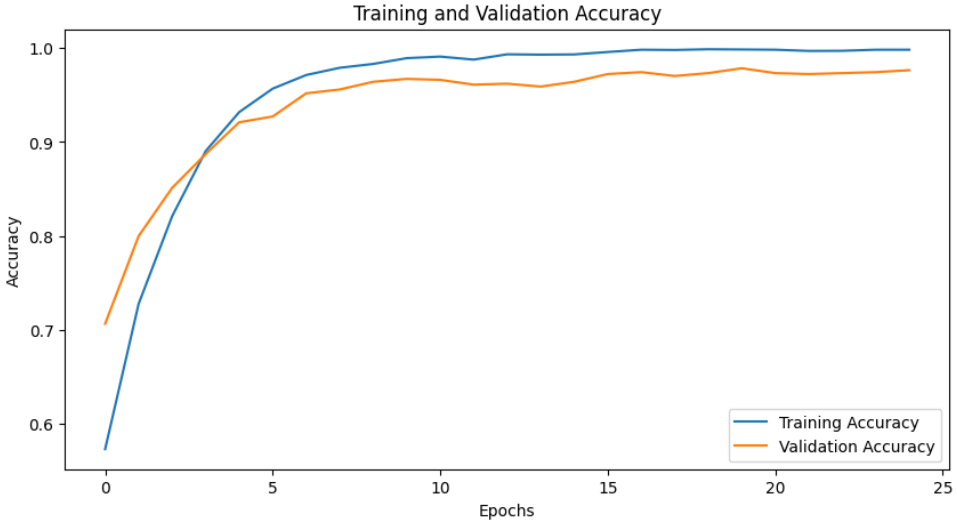

Figure 3.4: Training and Validation Accuracy

3.4. Model Explainability using Grad-CAM

To enhance the interpretability of our model, we employed Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize the regions of the input images that most influenced the model's predictions. The Grad-CAM technique highlights the important regions in the input image by generating a heatmap that indicates the contribution of each region to the final classification decision.

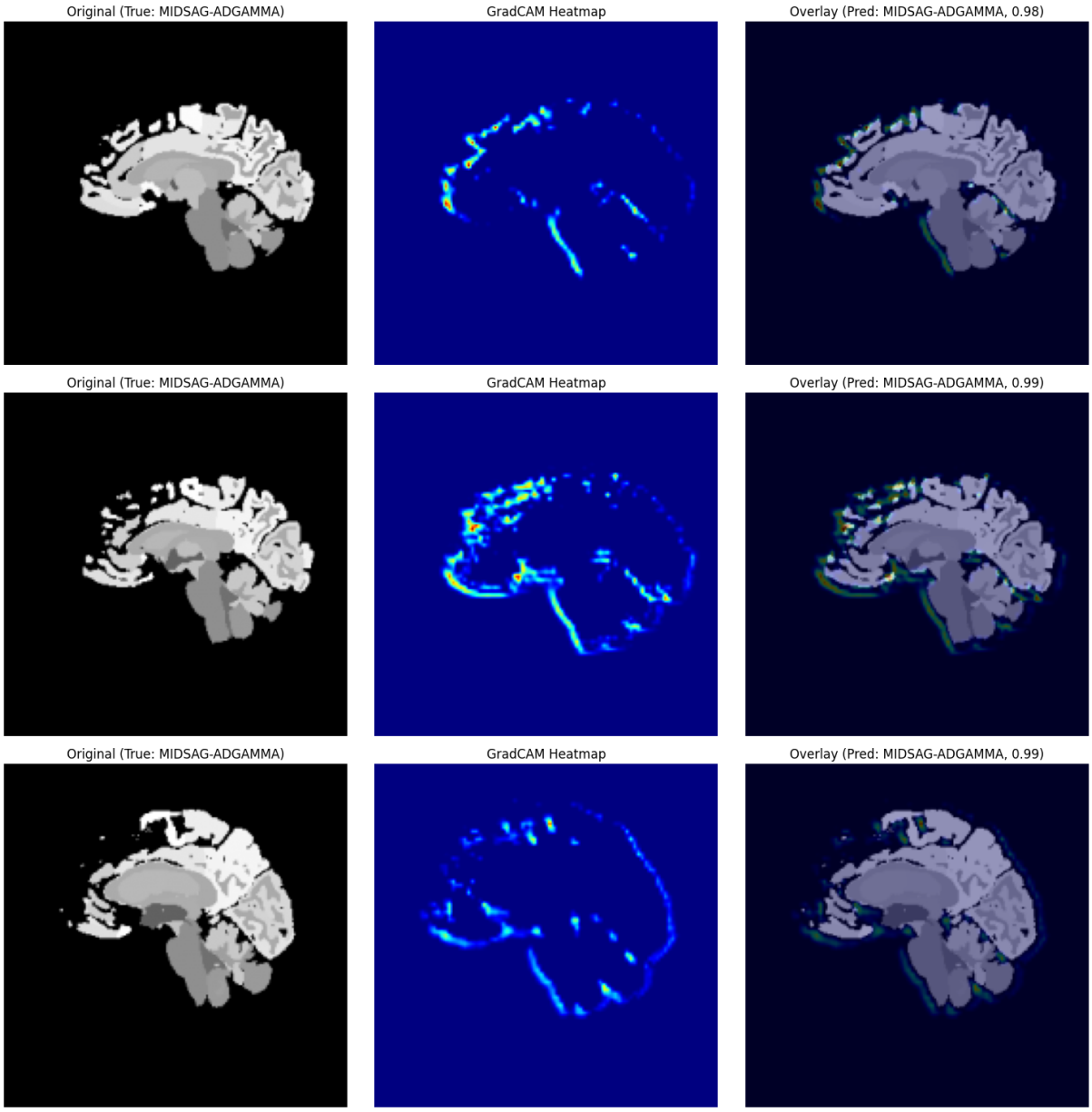

Figure 3.5: Grad-CAM visualizations for sample images from the test dataset.

In the Figure 3.5, the left column shows the original images, the middle column displays the Grad-CAM heatmaps, and the right column presents the overlay of the heatmap on the original image. The predictions and their corresponding confidence scores are also displayed.

As illustrated in Figure 3.5, the Grad-CAM heatmaps effectively highlight the regions that the model focuses on for making predictions. The high confidence scores (e.g., 0.98, 0.99) indicate that the model is confident in its predictions, and the highlighted regions align well with the expected areas of interest for the given task. This visualization confirms

that our model is making decisions based on relevant features in the input images, thereby enhancing the trustworthiness and interpretability of our predictions.

4 | Results

In this section, we present the performance analysis of our proposed model using multiple evaluation metrics. These include accuracy, precision, recall, F1-score, sensitivity, and specificity. Additionally, we provide a detailed analysis using the confusion matrix and ROC curves for a comprehensive understanding of the model's classification ability.

4.1. Performance Metrics

Table 4.1 summarizes the key performance metrics. The model achieves a high accuracy of 98.26%, indicating its strong predictive capability. The high precision (98.27%) and recall (98.26%) confirm its effectiveness in correctly classifying positive instances while minimizing false positives and false negatives. The F1-score of 98.25% further validates the balance between precision and recall. Additionally, the sensitivity (98.18%) and specificity (99.41%) reinforce the model's ability to distinguish between different classes effectively.

Metric	**Value**
Accuracy	0.9826
Precision	0.9827
Recall	0.9826
F1 Score	0.9825
Sensitivity	0.9818
Specificity	0.9941

Table 4.1: Performance metrics of the proposed model.

4.2. Confusion Matrix Analysis

To further analyze the model's classification performance, we provide the confusion matrix in Figure 4.1. This matrix highlights the number of correctly and incorrectly classified instances across different categories. The diagonal elements represent the correctly classified instances, while the off-diagonal elements indicate misclassifications.

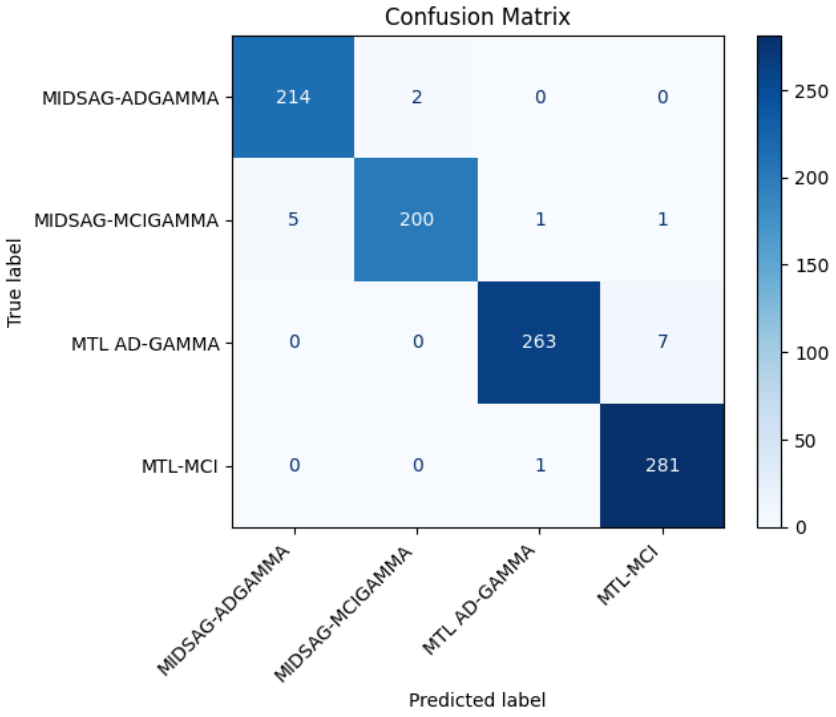

Figure 4.1: Confusion Matrix displaying classification performance across different categories.

From the confusion matrix, it can be observed that the model achieves a high number of correct classifications, with minimal misclassifications. Notably, the model performs exceptionally well in distinguishing between the different classes, as evidenced by the low number of misclassified instances.

4.3. Receiver Operating Characteristic (ROC) Analysis

The Receiver Operating Characteristic (ROC) curves for each class are illustrated in Figure 4.2. These curves provide insight into the model's ability to differentiate between classes across various decision thresholds. The Area Under the Curve (AUC) values, all close to 1.0, suggest strong classification performance.

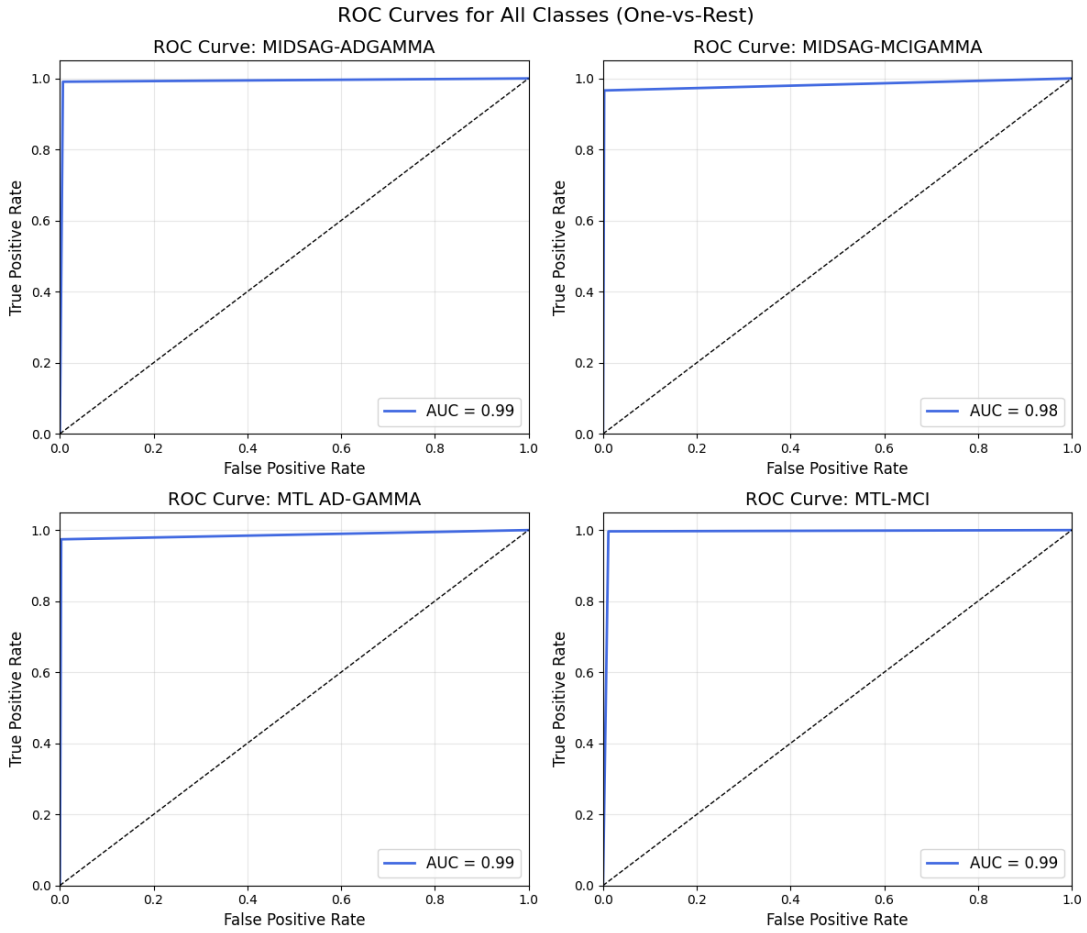

Figure 4.2: Receiver Operating Characteristic (ROC) Curves for all classes, demonstrating the model's high discriminative ability.

The ROC curves indicate that the model maintains a high true positive rate while minimizing false positives across different classes. Specifically, the AUC values confirm that the model is capable of distinguishing between different categories with a high degree of confidence.

4.4. Discussion and Insights

The results presented highlight the robustness and reliability of the proposed model. The combination of high accuracy, precision, recall, and specificity suggests that the model is well-suited for the classification task at hand.

The confusion matrix analysis reveals that misclassifications are minimal, and the high AUC values further validate the model's ability to make confident and accurate predictions. These findings indicate that the model generalizes well to unseen data and can be

considered a strong candidate for real-world deployment in similar classification tasks.

Overall, the model demonstrates a strong balance between different evaluation metrics, ensuring optimal performance with minimal risk of misclassification.

4.5. Comparison with Original Results

The performance metrics of our implementation are compared with the original SNeurod-CNN model in the table below:

Metric	Original	Our Implementation
Accuracy	98.1	98.26
Precision	97.2	98.27
Recall (Sensitivity)	99.0	98.18
Specificity	97.2	99.41
F1-Score	98.1	98.25
AUC	98.1	98.75

Table 4.2: Comparison of performance metrics between the original implementation and our re-implementation.

Our implementation achieved comparable or slightly better results in Accuracy, Precision, AUC and Specificity, while maintaining similar levels of Recall and F1-Score.

List of Figures

List of Tables