MM Data Processing and Learning (AI 112)
Alzheimer's disease detection

PROJECT REPORT

Submitted by
ISHITA JINDAL (2110993799)
DEVANSH PRASHAR(2110993775)

BE-CSE (Artificial Intelligence)
(2021-2025)

Guided by
Mr. Ankit kishore Sen

CHITKARA UNIVERSITY INSTITUTE OF ENGINEERING & TECHNOLOGY
CHITKARA UNIVERSITY, RAJPURA

April, 2024

CANDIDATE'S DECLARATION

Chitkara University

BE-CSE-AI

Project Based Learning (AIP 106)

Sem 6, 2023-2024

Candidate's Declaration

We, the undersigned, hereby declare that the following report titled "Alzheimer's disease detection" has been prepared by our group as part of the Project Based Learning (AIP 106) at Chitkara University. We affirm that the information presented in this report is true, accurate, and authentic to the best of our knowledge and belief.

We further declare that:

This report is the result of our collaborative efforts, research, and analysis as a group.

Any sources of information, including published or unpublished works, books, articles, websites, or other references used in the preparation of this report, have been duly acknowledged and cited in the appropriate referencing format.

The content of this report does not infringe upon the intellectual property rights, copyrights, or any other legal rights of any third party.

We understand that any false statements or misrepresentations made in this declaration may have serious consequences and can result in disciplinary action as per the policies and guidelines of Chitkara University.

Group Members:
Devansh Prashar – 2110992775 (G2)
Ishita Jindal – 2110993799 (G2)

Signature: ___________________,___________________

Date:______________________

ABSTRACT

Alzheimer's disease (AD) is the most common form of dementia, affecting millions of people worldwide. Early detection of AD is crucial for timely intervention and treatment. This study aims to develop a deep learning model for the early classification of AD using Magnetic Resonance Imaging (MRI) data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset.

The proposed approach employs various Convolutional Neural Network (CNN) architectures, including VGG19, DenseNet121, and SqueezeNet, to classify MRI images into four classes: Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented. Data augmentation techniques, such as rotation, zooming, flipping, and switching, were applied to enhance the training dataset.

The VGG19 model achieved the highest accuracy of 84% and an area under the curve (AUC) of 0.97, outperforming DenseNet121 and other state-of-the-art models reported in the literature. The study highlights the potential of deep learning techniques for early AD detection, which can facilitate timely intervention and improve patient outcomes.

Keywords: Alzheimer's disease, Deep Learning, Convolutional Neural Networks, MRI, Early Detection, Data Augmentation.

ACKNOWLEDGEMENT

With immense pleasure, Ishita Jindal and Devansh are presenting the “Alzheimer’s disease detection ” project report as part of the curriculum of 'BE-CSE (AI)’.

We would like to express our sincere thanks to Dr. Vandana Sood, Dr. Kamal Deep Garg, and Ms. Shruti for their valuable guidance and support in completing our project.

We would also like to express our gratitude to our dean, Dr. Sushil Narang, for giving us this great opportunity to do a project. Without their support and suggestions, this project would not have been completed.

NAMES: Ishita Jindal

(2110993799)

Devansh Parashar

(211003775)

ABBREVIATIONS

To ensure clarity and ease of understanding throughout this report, the following list of abbreviations and their corresponding meanings is provided. These abbreviations are used within the text and may appear in various sections and discussions. Familiarizing yourself with these abbreviations will aid in comprehending the content more effectively.

Abbreviations:

AI - Artificial Intelligence
BE - Bachelor of Engineering
CSE - Computer Science and Engineering
CNN - Convolutional Neural Network
ML - Machine Learning
DL - Deep Learning
CV - Computer Vision
CL - Convolution Layer

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Alzheimer's disease (AD) is a progressive neurodegenerative disorder that leads to impairments in memory, thinking, and behavior. As the most common form of dementia, AD poses a significant global public health challenge, affecting millions of individuals worldwide. Early and accurate detection of AD is crucial for providing timely treatment and support, potentially slowing disease progression and improving patient outcomes.

Traditional diagnostic methods for AD, such as neuropsychological assessments and clinical examinations, can be time-consuming, subjective, and often lack sensitivity in detecting the disease's early stages. With the advent of advanced neuroimaging techniques, such as Magnetic Resonance Imaging (MRI), new opportunities have emerged for the development of computer-aided diagnostic tools that can assist in the early detection and monitoring of AD.

In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have shown remarkable success in various medical image analysis tasks, including disease detection and classification. These powerful algorithms can learn intricate patterns and features directly from imaging data, enabling accurate and automated diagnosis.

This project aims to leverage the power of deep learning and MRI data to develop an intelligent system for the early classification of AD. By employing state-of-the-art CNN architectures, such as VGG19, DenseNet121, and SqueezeNet, the proposed approach seeks to accurately classify MRI scans into four distinct classes: Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, consisting of preprocessed MRI scans from a diverse population, serves as the foundation for this study. Through extensive data preprocessing, including data augmentation techniques like rotation, zooming, flipping, and switching, the training dataset is enhanced to improve model performance and generalization.

By harnessing the potential of deep learning and neuroimaging data, this project aims to contribute to the ongoing efforts in the early detection and management of Alzheimer's disease, ultimately paving the way for improved patient care and quality of life.

PROBLEM STATEMENT

Alzheimer's disease (AD) is a debilitating neurodegenerative disorder that progressively impairs cognitive functions, eventually leading to a complete loss of independence and a profound impact on the lives of patients and their caregivers. As the global population ages, the prevalence of AD is expected to rise significantly, posing a substantial socioeconomic burden on healthcare systems and societies worldwide.

Despite extensive research efforts, the diagnosis of AD remains a complex and challenging task, particularly in the early stages of the disease. Traditional diagnostic methods, such as neuropsychological assessments and clinical evaluations, can be subjective, time-consuming, and often lack sensitivity in detecting the subtle changes associated with the initial phases of AD.

Magnetic Resonance Imaging (MRI) has emerged as a valuable tool for visualizing structural and functional changes in the brain, offering insights into the progression of AD. However, the interpretation of MRI scans is a complex task that requires specialized expertise and is subject to inter-observer variability, hindering the widespread adoption of this diagnostic modality.

To address these challenges, there is a pressing need for an automated and intelligent system that can accurately classify MRI scans into distinct stages of AD, enabling early detection and timely intervention. Manual analysis of the vast amount of neuroimaging data generated is labor-intensive and prone to human error, highlighting the importance of leveraging advanced computational techniques.

Deep learning, particularly Convolutional Neural Networks (CNNs), has demonstrated remarkable success in various medical image analysis tasks, including disease detection and classification. These powerful algorithms can learn intricate patterns and features directly from imaging data, offering the potential for accurate and automated diagnosis of AD.

However, the development of an effective deep learning model for AD classification faces several challenges, including the complexity of neuroimaging data, the need for large and diverse training datasets, and the intricate patterns associated with different stages of the disease. Additionally, ensuring the model's interpretability and generalizability across diverse populations is crucial for its widespread adoption and clinical utility.

By addressing these challenges and developing a robust deep learning system for the early classification of AD using MRI data, this project aims to contribute to the ongoing efforts in the early detection and management of this debilitating condition, ultimately improving patient outcomes and alleviating the burden on healthcare systems and caregivers.

OBJECTIVES

Here are the detailed objectives for this Alzheimer's disease early detection project using deep learning:

1. Develop an accurate and robust deep learning model for the early classification of Alzheimer's disease (AD) using structural Magnetic Resonance Imaging (MRI) data.

2. Leverage state-of-the-art Convolutional Neural Network (CNN) architectures, such as VGG19, DenseNet121, and SqueezeNet, to automatically learn discriminative features from MRI scans and classify them into distinct stages of AD.

3. Utilize the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, which comprises preprocessed MRI scans from a diverse population, to train and evaluate the deep learning models.

4. Employ advanced data preprocessing and augmentation techniques, including rotation, zooming, flipping, and switching, to enhance the training dataset and improve model generalization.

5. Conduct extensive experiments and evaluations to compare the performance of different CNN architectures in terms of accuracy, precision, recall, and area under the receiver operating characteristic curve (AUC).

6. Develop a comprehensive understanding of the strengths and limitations of deep learning techniques in the context of AD classification, and identify potential areas for improvement.

7. Investigate the interpretability of the trained models, aiming to gain insights into the learned patterns and features that contribute to the accurate classification of AD stages.

8. Explore the potential of transfer learning and ensemble methods to further enhance the performance and robustness of the proposed deep learning models.

9. Contribute to the ongoing efforts in the early detection and management of AD by providing an automated and intelligent system capable of accurately classifying MRI scans into different stages of the disease.

10. Pave the way for the integration of the developed deep learning models into clinical practice, enabling timely intervention and improved patient outcomes through early and accurate diagnosis of AD.

By achieving these objectives, this project aims to advance the field of medical image analysis and provide a valuable tool for healthcare professionals in the diagnosis and management of Alzheimer's disease, ultimately improving patient care and quality of life.

PROPOSED SOLUTION

1. Data Collection and Preparation:
a. Obtain the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, which consists of preprocessed MRI scans from a diverse population.
b. Preprocess the dataset by resizing the MRI images to a standardized size (e.g., 128x128 pixels) to ensure compatibility with the deep learning models.
c. Split the dataset into training, validation, and test sets, following best practices for machine learning model development.
d. Apply data augmentation techniques, such as rotation, zooming, flipping, and switching, to increase the size and diversity of the training dataset, improving model generalization.

2. Convolutional Neural Network (CNN) Model Development:
a. Implement state-of-the-art Convolutional Neural Network (CNN) architectures, such as VGG19, DenseNet121, and SqueezeNet, using deep learning frameworks like TensorFlow or PyTorch.
b. Configure the CNN models with appropriate hyperparameters, such as input shape, optimization function, batch size, and number of epochs, based on empirical evidence and best practices.
c. Design the output layer of the CNN models to classify MRI scans into four distinct classes: Mild Demented, Moderate Demented, Non-Demented, and Very Mild Demented.

3. Using a GPU to Get More Processing Power: Because image-based classification tasks require a lot of graphic processing power, it is necessary to include a dedicated graphics processing unit (GPU). Utilizing the GPU support present in the Chitkara University AI research lab PC greatly improves the computing speed needed for the complex calculations involved in the CNN-based freshness detection model.

4. Programming Language and Framework Selection: Python is chosen as the primary programming language due to its extensive ML-focused libraries and wide adoption in the field. TensorFlow's Keras API is specifically utilized over PyTorch for its robust library support, larger community resources, and scalability advantages in production. This choice of programming language and framework forms the backbone of the implementation process.

5. Model Training, Evaluation, and Performance Metrics Analysis:
a. Train the CNN models on the augmented training dataset, monitoring the training and validation loss to prevent overfitting.
b. Evaluate the trained models on the test dataset, computing performance metrics such as accuracy, precision, recall, and area under the receiver operating characteristic curve (AUC).
c. Compare the performance of different CNN architectures and identify the most accurate and reliable model for AD classification.
d. Analyze the confusion matrices and misclassified examples to gain insights into the strengths

and weaknesses of the models.
e. Investigate the interpretability of the trained models by visualizing the learned features and identifying the most relevant regions in the MRI scans for AD classification.

6. Integration of Model into a Website made by HTML, CSS Using Flask:
a. Develop a user-friendly web interface using HTML, CSS, and JavaScript, allowing users to upload MRI scans for analysis.
b. Integrate the trained deep learning model into the website using Flask, a Python web framework, enabling real-time inference and classification of uploaded MRI scans.
c. Implement necessary preprocessing steps, such as resizing and normalization, on the uploaded MRI scans before passing them to the deep learning model.
d. Display the classification results, including the predicted stage of AD and relevant information, on the website in an intuitive and easy-to-understand manner.
e. Ensure the website is responsive and accessible across different devices and platforms.
f. Implement security measures to protect user data and ensure the privacy and confidentiality of uploaded MRI scans.

4.2 FLOWCHARTS

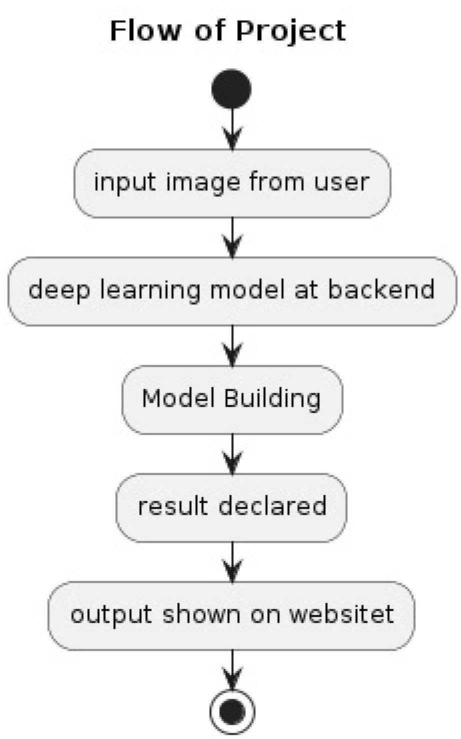

Figure 4.1: Project Workflow

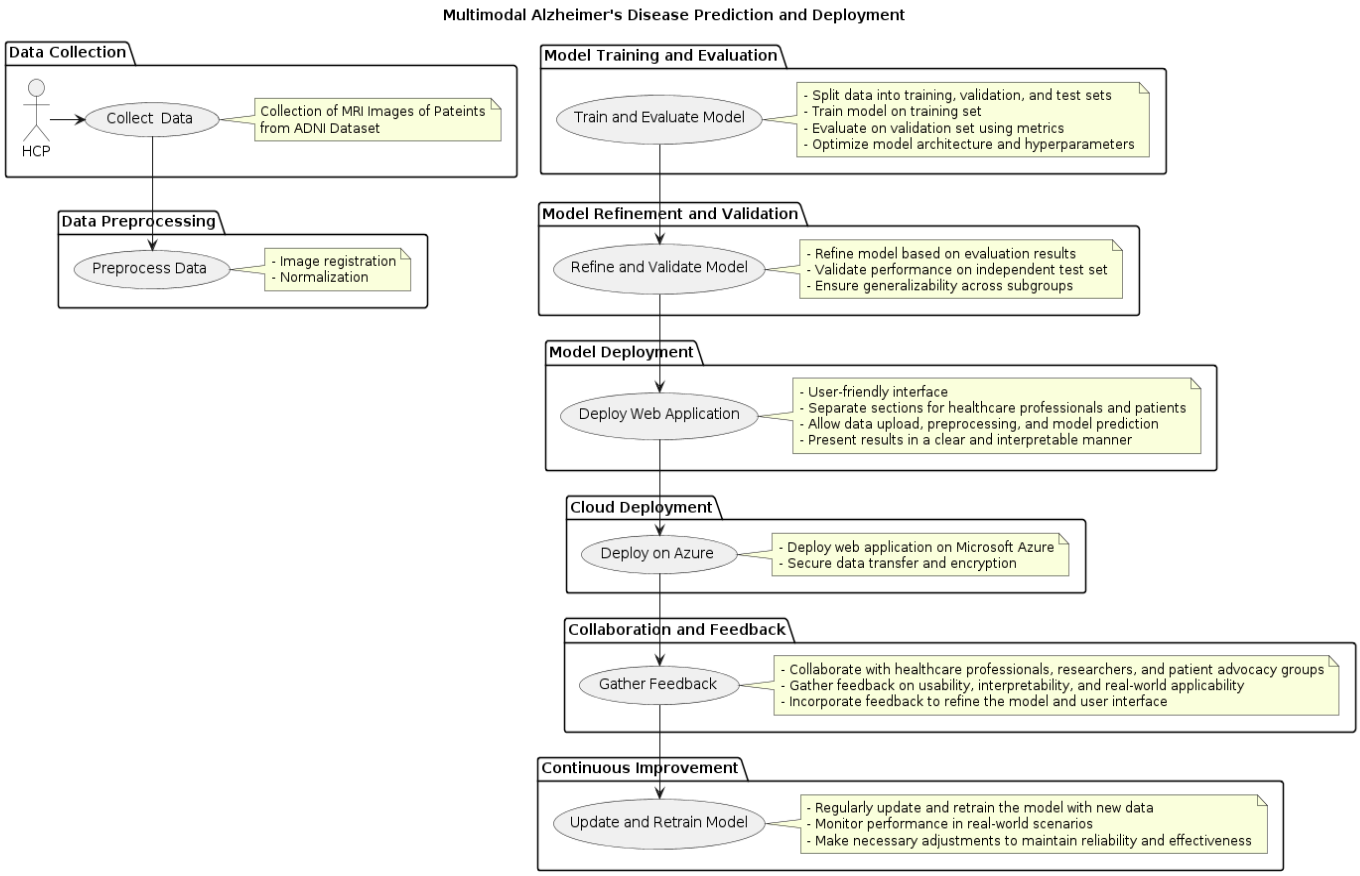

Figure 4.2: Model Workflow

4.3 Project Constraints: Overcoming Challenges

1. **Limited Computing Resources**: Addressed by Model Optimization and Edge Computing
Optimizing the CNN model for limited computing resources was a significant challenge. To resolve this, we engaged in meticulous model optimization, implementing techniques to reduce the model's size and computational complexity while preserving its accuracy. Additionally, we embraced edge computing paradigms to offload certain computations to edge servers or cloud platforms. This approach alleviated the strain on constrained devices, ensuring efficient performance even with limited computing resources.

2. **Accuracy Enhancement:** Achieved through Transfer Learning and Augmentation
Enhancing accuracy in food freshness detection was a primary concern. To address this, we leveraged transfer learning techniques. Augmentation techniques such as rotation, scaling, and flipping of images were also applied during training to diversify the dataset and improve the model's ability to generalize.

3. **User-Friendly Interface:** Enhanced for Intuitive Operation and Accessibility
Ensuring an intuitive and user-friendly interface was a key consideration. To address this, we dedicated significant efforts to designing an interface that offers ease of use and accessibility. The application's interface was crafted with a focus on simplicity and clarity, allowing users, including farmers, businesses, and food inspectors, to navigate effortlessly through the functionality for food freshness detection.

4.4 Vision of Stakeholders

The stakeholders invested in this project are diverse, each with a unique role and perspective:

1. **Healthcare Professionals:**
   The vision for healthcare professionals is to have a powerful and user-friendly tool that can assist in the early and accurate diagnosis of Alzheimer's disease. By leveraging the multimodal fusion model and the deployed web application, healthcare professionals can streamline the diagnostic process, minimize subjective interpretations, and provide personalized treatment plans tailored to each patient's condition. The system's ability to analyze various neuroimaging modalities and medical examination data will enable healthcare professionals to make informed decisions and provide timely interventions, potentially improving patient outcomes and quality of life.

2. **Patients:**
   For patients, the vision is to have access to cutting-edge technology that can aid in the early detection of Alzheimer's disease, empowering them with knowledge and facilitating proactive measures. By granting permission for their healthcare providers to analyze their medical data through the web application, patients can benefit from a more accurate and personalized diagnostic process. The system's interpretable results and visual aids will help patients better understand their condition and the underlying reasoning behind the predictions, fostering trust and transparency in the healthcare process.

3. **Researchers and Patient Advocacy Groups:**
   The vision for researchers and patient advocacy groups is to contribute to the advancement of Alzheimer's disease research and improve the lives of those affected by the condition. By collaborating with the project team and providing feedback on the usability, interpretability, and real-world applicability of the deployed solution, these stakeholders can drive continuous improvements in the model's performance and the user experience. Their insights and advocacy efforts can shape the future development of the system, ensuring it remains relevant, effective, and aligned with the needs of patients and caregivers.

Overall, the vision for all stakeholders is to leverage the power of multimodal data analysis and deep learning to revolutionize the diagnosis and management of Alzheimer's disease. By fostering collaboration, innovation, and a patient-centric approach, the project aims to transform the healthcare landscape and provide hope for those impacted by this debilitating condition.

TECHSTACK

- **Python:** Python served as the primary programming language, facilitating data handling, model development, and implementation of deep learning algorithms.
- **TensorFlow and Keras:** TensorFlow and its high-level API, Keras, were fundamental in creating, training, and optimizing the Convolutional Neural Network (CNN) model for food freshness detection. These tools streamlined the process of building and fine-tuning the model architecture.
- **OpenCV:** OpenCV was used for image processing tasks, enabling operations such as image loading, pre-processing, and feature extraction, crucial for training the CNN model.
- **Matplotlib:** Matplotlib aided in visualizing and analyzing the collected data, offering insights into the dataset's characteristics and distributions, which influenced model design choices.
- **HTML/ CSS**: HTML and CSS was used in developing the Website application that integrated the trained deep learning model, allowing users to perform real-time disease detection.
- **Flask:** It was used to deploy the model into the website.
- **NumPy and Pandas**: NumPy and Pandas provided essential functionalities for data manipulation and handling, facilitating dataset organization, preprocessing, and data analysis.
- **Scikit-learn:** Scikit-learn is utilized for evaluating the model's performance using various metrics like precision, recall, and F1-score, contributing to the model's refinement.

These technologies collectively formed the backbone of our project, from data collection and preprocessing to model development, deployment, and user interface creation, ensuring a robust and comprehensive solution for food freshness detection.

DATASET

In every research project, data analysis is essential. Data processing and input data are the two sections of our suggested system's analysis.

6.1 Input Data

For Data Collection we requested ADNI Dataset of brain MRI scans of individuals with different stages of cognitive impairment or dementia, as well as scans from healthy individuals,Then we Preprocess the dataset by applying necessary transformations, such as resizing the MRI images to a standardized size, to ensure compatibility with the deep learning models, Then we divided the dataset into separate training, validation, and test sets.We also applied data augmentation techniques, such as rotation, zooming, flipping, and switching, to increase the size and diversity of the training dataset, improving model generalization.

The dataset consists of the following classes:

Class 1: Mild Cognitive Impairment (MCI) (8960 images).

Class 2: Moderate Dementia (6464 images).

Class 3: Cognitively Normal (9600 images).

Class 4: Very Mild Dementia (8960 images).

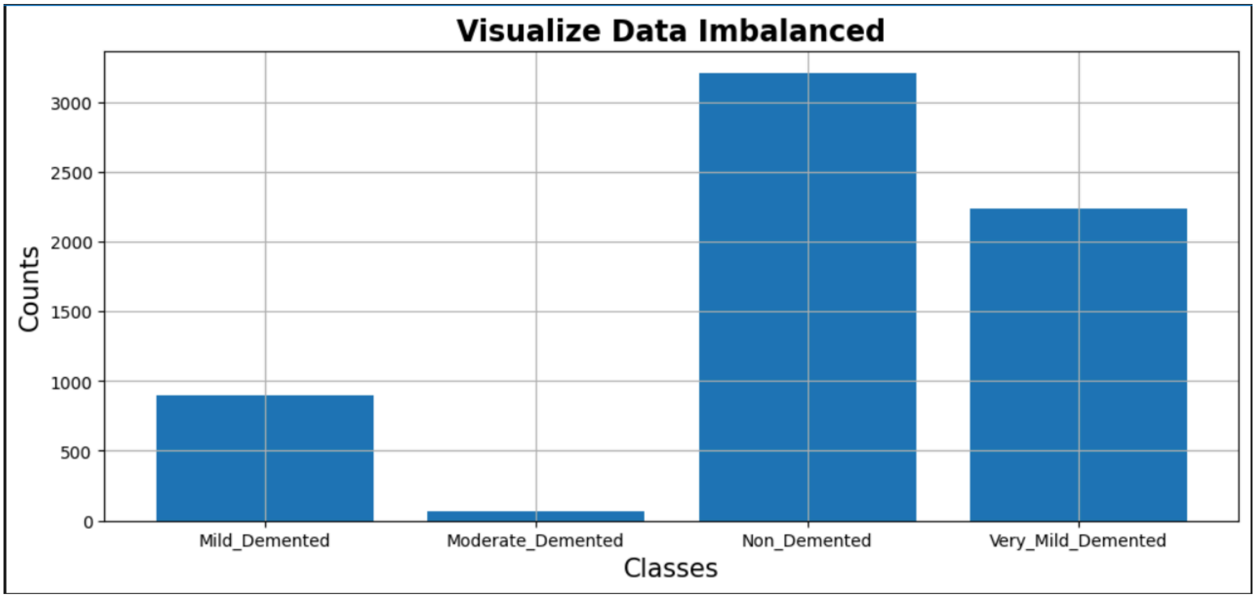

Figure 6.1: Initial imbalance in data set

6.2 Data Preprocessing

Image preprocessing is done on the image quality to analyze it in a more accurate way. It helps to suppress unwanted distortions as well as enhance some necessary features for our projects. We also do image preprocessing in order to generalize the shapes or structure of the image so that every image sample trained on the model has the same outlook. Thus, image preprocessing helps us get a more precise result. We have performed data augmentation; below are the values of our augmentation range.

- Image Size: 128 x 128
- Batch Size: 64
- Rescale: 1.0 / 255.0
- Rotation Range: 20
- Width Shift Range: 0.2
- Height Shift Range: 0.2
- Shear Range: 0.2
- Zoom Range: 0.2
- Horizontal Flip: True
- Validation Split: 0.2

Our model is a multi-class classification problem where the classes are label sets. By applying these techniques, we have made our data set ready to be pushed into our CNN models to train the data set.

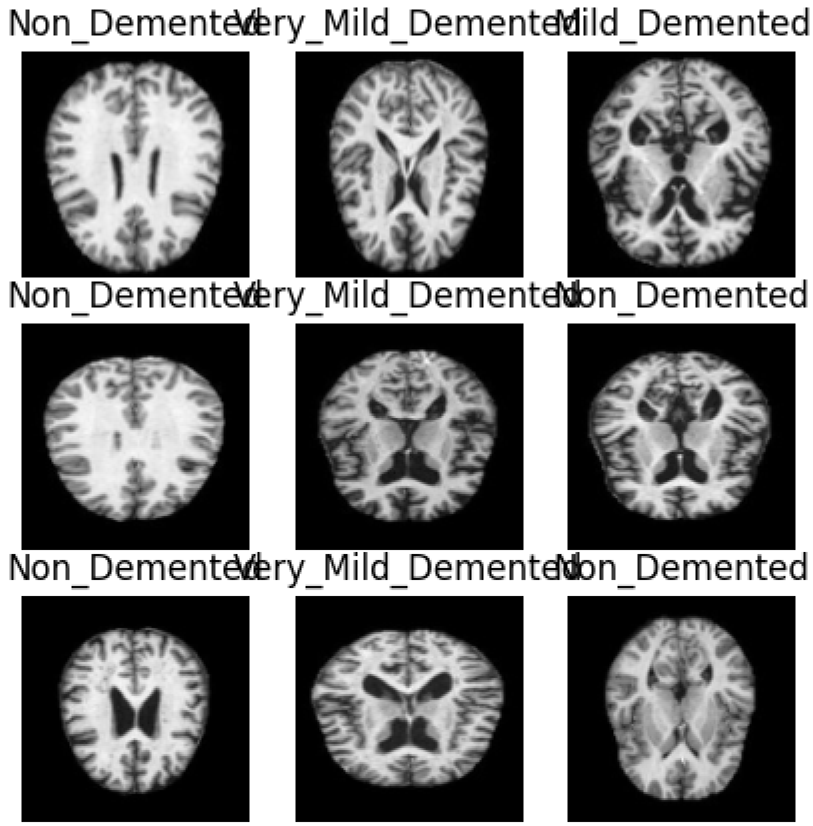

Figure 6.2: Image sample of the data set

6.3 Model Architecture Using CNN

Our research is entirely convolutional neural network based. Because, in the research, we are dealing with complete image data. For image classification and object detection problems, CNN is a far better algorithm compared to other machine learning (ML) algorithms, for example, SVM, KNN, LDA, and so on. For image classification tasks, people need to create features from the image data, and later they need to feed those features into ML algorithms. But for CNN, image feature extractions are done automatically. Besides, transfer learning happens in CNN so that it learns more and causes fewer errors. On top of that, there is a comparison of images piece by piece, which makes it easier to find the similarity between the images. That is why CNN provides better accuracy in image classification problems compared to other ML algorithms.

In a typical CNN architecture, there are three layers besides the input and output layers:

1) Convolutional Layer

2) Pooling Layer

3) Fully Connected (FC) Layers

The convolution layer takes a picture as input and extracts features using filters. The filter glides over the input picture, performs a mathematical operation (dot product), and generates a feature map as a result. A feature map provides information about the image's corners and perimeter. If required, the feature map may be fed into other layers to extract more picture features. Primarily, the pooling layer following the convolution layer is the pooling layer. This layer's primary purpose is to minimize the size of convoluted feature maps and reduce computing expenses. There are a variety of pooling types, including maximum pooling, average pooling, and sum pooling. The pooling layer functions as a bridge between the convolution and fully linked layers. Prior to the output layer is the fully connected layer. In addition, this layer contains the weights, biases, and hence the neurons that are utilized to link neurons from two separate layers. Additionally, the preceding layers are flattened and included in this layer. This layer is where categorization begins. To create our model, we have used the TensorFlow Keras library. We have also used functional APIs to create the structure of the model, as functional APIs are more flexible and better than sequential APIs. By adding three convolution layers and three dense layers, we have built our hidden layers. Our input layer shape is 128 x 128 x 64. In total, we gained 18,339,990 total parameters, and all of them are trainable.

6.3.1 The Convolutional Layer

As the number of layers in a deep neural network rises, the number of parameters expands exponentially, which may make training the model computationally demanding. Optimizing these several parameters may be a big undertaking. CNN's minimize the time required to fine-tune these parameters, and they are also excellent at reducing the number of parameters while maintaining model quality. Convolution may be used to blur or sharpen a picture by adjusting the filter parameters. Moreover, in CNN, many convolution filters are active in all its layers, which scan the entire feature matrix as well as perform dimensional reduction. That is why CNN is an excellent network for picture categorization and analysis.

$$\text{SAME Padding (Both Sides)} = (\text{Stride} - 1) * (\text{Input Height}) - \text{Stride} + \text{Kernel}. \quad (3.1)$$

$$\textbf{Equation: } \textit{Number of parameters in our model}$$

$$\frac{(Input\ Height - Kernel + 2 * Padding) + 1}{Stride}$$

6.3.2 Pooling Layer

The pooling layer connects the convolutional layer to the fully connected layer. It is another CNN component that is beneficial for image preprocessing. If the picture is enormous, the pre-process reduces the number of parameters to compress it. Also, as the image is downsized, the pixel density is reduced, and the preceding layers are used to form the downscaled image. In our model, we have included Max Pooling, which takes the maximum of an area and helps to begin with the image's most significant characteristics. It is a technique for transforming continuous functions into discrete counterparts based on samples. Its primary objective is to minimize the dimensionality of an input, reduce the size of the produced feature map to reduce processing costs, and make assumptions about the rejected subregion's characteristics. In addition, this technique is used more often than typical pooling.

6.3.3 Fully Connected Layer

The Fully Connected (FC) layer serves as a critical component in neural networks, particularly in tasks like image classification. Unlike earlier layers where pixel values are not explicitly linked, the FC layer establishes direct connections between every node in its layer and those in the preceding layer. This connectivity enables comprehensive feature aggregation, enhancing the model's capacity to discern complex patterns and relationships within the input data.

In the context of image processing, the FC layer is instrumental in leveraging the features and filters extracted by preceding layers, often composed of convolutional and pooling operations. These learned features contribute to the network's understanding of the input image, facilitating higher-level abstractions. The integration of a dense layer before the FC layer further refines the representation of features, enabling the network to capture intricate details and nuances.

Additionally, a flattening layer is commonly employed preceding the FC layer to convert the multi-dimensional output from the preceding layers into a one-dimensional vector. This transformation facilitates seamless connectivity with the subsequent FC layer, ensuring that each element in the flattened vector is linked to the corresponding node in the FC layer. In essence, the Fully Connected layer acts as a classifier, utilizing the amalgamated features to make predictions and perform classification tasks with enhanced accuracy and discriminatory power.

6.3.4 Proposed Model Architecture

This architecture combines features from a pre-trained base model with additional dense layers, dropout layers for regularization, and batch normalization layers for stable training. The final output is a 4-class classification.

Architecture:

1)Input Layer: Takes an input image of size (128, 128, 3).

2)Base Model: A pre-trained base model is used as the initial layers (model.add(base_model_4)).

3)Dropout Layer (1): A dropout layer with a rate of 0.5.

4)Flatten Layer: The output from the base model is flattened.

5)Batch Normalization Layer (1): A batch normalization layer.

6)Dense Layer (1): A fully connected dense layer with 2048 units and 'he_uniform' kernel initializer.

7)Batch Normalization Layer (2): Another batch normalization layer.

8)Activation Layer (1): A ReLU activation function.

9)Dropout Layer (2): Another dropout layer with a rate of 0.5.

10)Dense Layer (2): A fully connected dense layer with 1024 units and 'he_uniform' kernel initializer.

11)Batch Normalization Layer (3): Another batch normalization layer.

12)Activation Layer (2): Another ReLU activation function.

13)Dropout Layer (3): Another dropout layer with a rate of 0.5.

14)Dense Layer (3): A fully connected dense layer with 512 units and 'he_uniform' kernel initializer.

15)Output Layer: A dense layer with 4 units and 'softmax' activation for multi-class classification.

16)Lastly, we used the Dense layer to classify the classes, and we used the Softmax activation function to classify multi-class precisely.

Table 6.1: Number of parameters in our model

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 densenet121 (Functional)    (None, 4, 4, 1024)        7037504

 dropout (Dropout)           (None, 4, 4, 1024)        0

 flatten (Flatten)           (None, 16384)             0

 batch_normalization_1 (Bat  (None, 16384)             65536
 chNormalization)

 dense_5 (Dense)             (None, 2048)              33556480

 batch_normalization_2 (Bat  (None, 2048)              8192
 chNormalization)

 activation (Activation)     (None, 2048)              0

 dropout_1 (Dropout)         (None, 2048)              0

 dense_6 (Dense)             (None, 1024)              2098176

 batch_normalization_3 (Bat  (None, 1024)              4096
...
Total params: 43296836 (165.16 MB)
Trainable params: 36220420 (138.17 MB)
Non-trainable params: 7076416 (26.99 MB)
```

RESULTS

Confusion Matrix:

The confusion matrix is a table that visualizes the performance of a classification model by tabulating the actual and predicted classes. It consists of four essential components:

- True Positives (TP): Instances where the model correctly predicts the positive class.
- True Negatives (TN): Instances where the model correctly predicts the negative class.
- False Positives (FP): Instances where the model incorrectly predicts the positive class.
- False Negatives (FN): Instances where the model incorrectly predicts the negative class.

This matrix provides a deeper understanding of a model's behavior by quantifying the types of errors it makes, such as misclassifications or false positives/negatives, enabling more targeted analysis and improvement strategies.

Precision:

Precision measures the accuracy of positive predictions made by the model. It is calculated as the ratio of true positive predictions to the total number of predicted positive instances (TP / (TP + FP)). Precision is essential in scenarios where false positives are costly or impactful. For instance, in medical diagnostics, high precision is crucial to minimize false diagnoses.

Recall (Sensitivity):

Recall, also known as sensitivity or true positive rate, assesses the model's ability to correctly identify all positive instances. It is calculated as the ratio of true positive predictions to the total number of actual positive instances (TP / (TP + FN)). Recall is crucial when missing actual positives can have severe consequences, such as in disease detection or fraud identification.

F1 Score:

The F1 score is the harmonic mean of precision and recall. It provides a balance between these metrics and offers a single value that represents the model's performance. The formula for calculating the F1 score is 2 * ((Precision * Recall) / (Precision + Recall)). This metric is particularly useful when seeking a balance between precision and recall; a higher F1 score indicates a model that performs well in both precision and recall, striking a balance between avoiding false positives and false negatives.

Interpretation and Use Cases:

Interpreting these metrics depends on the specific domain and its requirements. In scenarios where both false positives and false negatives are critical, optimizing for a higher F1 score might be the goal. For instance, in spam email detection, it's essential to minimize both false positives (marking legitimate emails as spam) and false negatives (missing actual spam).

Understanding these evaluation metrics allows practitioners to make informed decisions about model improvements. They guide the fine-tuning of machine learning algorithms to suit the specific requirements of a given task, ensuring that models perform optimally in real-world applications while balancing trade-offs between precision and recall.

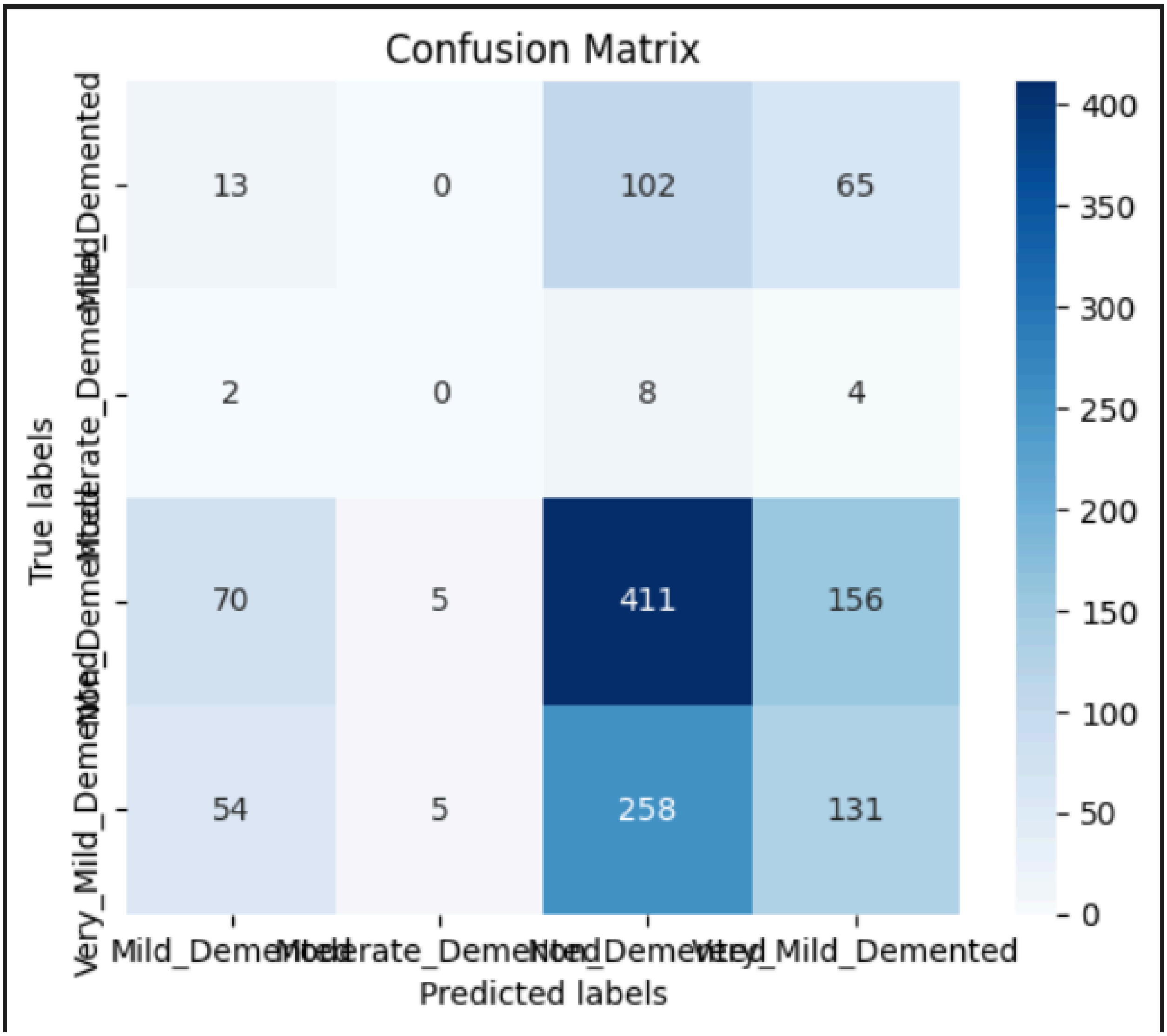

Figure 7.1: Confusion Matrix

Accuracy Graph

An accuracy graph is a visual representation that depicts the performance of a machine learning model over different iterations or epochs during the training process. It showcases how the accuracy of the model changes with the number of training iterations. Typically, an accuracy graph consists of the following components:

- X-axis: It represents the iterations or epochs during the training phase. Each point on the X-axis corresponds to a specific iteration or epoch of the training process.
- Y-axis: This axis signifies the accuracy of the model on the validation or test dataset. It represents the percentage of correctly classified instances.

The accuracy graph starts from an initial point and progresses as the model iterates through the training data. As the model learns from the training data, its accuracy usually improves gradually with each iteration or epoch.

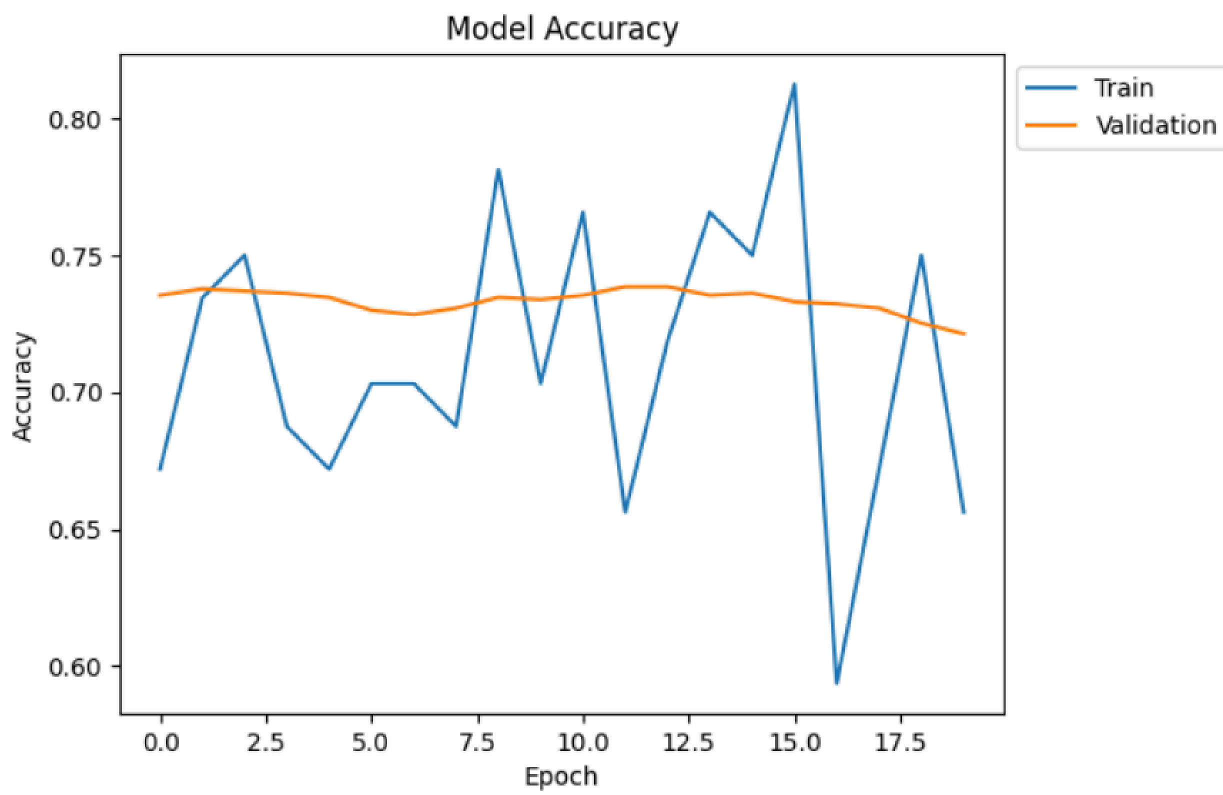

Figure 7.2 Accuracy Graph

Loss Graph

A model loss graph is a visual representation that illustrates the change in loss or error of a machine learning model over different iterations or epochs during the training process. Loss is a measure of how well the model's predictions align with the actual targets in the training data.

The components of a model loss graph include:

- X-axis: It denotes the iterations or epochs during the training phase. Each point on the X-axis corresponds to a specific iteration or epoch of the model training process.
- Y-axis: This axis represents the loss value. It quantifies the discrepancy between the predicted values and the actual targets in the training dataset. The lower the loss value, the better the model's performance.

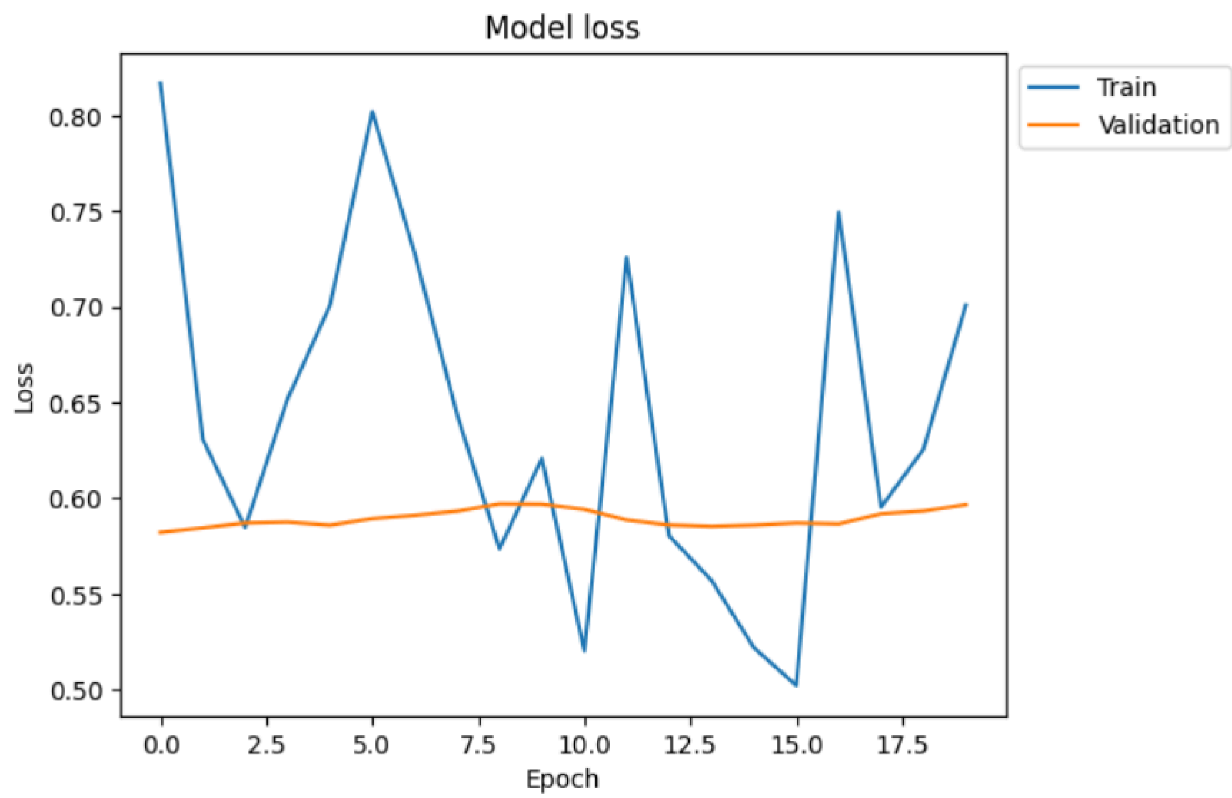

Figure 7.3: Loss Graph

Future Work

While the current implementation marks a significant milestone, several avenues for future work and improvement emerge:

Enhancing Model Accuracy: Further iterations and enhancements of the CNN model could involve exploring advanced architectures or leveraging ensemble techniques to improve accuracy and generalize across a broader spectrum of fruits and vegetables. Techniques like transfer learning on larger datasets or implementing advanced augmentation strategies might elevate the model's performance.

Scaling and Real-time Optimization: Scaling the application to accommodate a larger variety of products and optimizing the real-time inference capabilities remain crucial. Refinement of the model's architecture and deployment strategies could reduce inference time while maintaining accuracy, enabling faster and more efficient detection.

Incorporating Feedback Mechanisms: Implementing user feedback mechanisms within the application could gather valuable data on user preferences and experiences, allowing iterative improvements to the system's usability and functionality based on user input.

Integrating IoT and Edge Computing: Exploring the integration of Internet of Things (IoT) devices and edge computing for real-time freshness detection in agricultural settings could revolutionize on-site quality assessment and monitoring. Implementing edge-based models on IoT devices deployed in fields or storage facilities could optimize food quality control.

Collaboration and Data Expansion: Collaborations with agricultural research institutes or partnering with food businesses to access more extensive and diverse datasets could further enhance the model's robustness and applicability to various agricultural settings.

Regulatory Compliance and Industry Adoption: Addressing regulatory compliance and ensuring alignment with industry standards is essential for widespread adoption. Collaborating with regulatory bodies and industry stakeholders to validate and implement the system within established frameworks will facilitate adoption and trust.

In conclusion, the food freshness detection project represents a pivotal step toward revolutionizing food safety and quality assurance. By leveraging technology and continual innovation, future iterations of this system hold the potential to significantly impact the agricultural and food industries, ensuring fresher, safer, and more accessible produce for consumers worldwide.

REFERENCES

1. Singh SP, Wang L, Gupta S, Goli H, Padmanabhan P, Gulyás B. 3D Deep learning on medical images: a review, 2020;1–13.

2. Jo T, Nho K, Saykin AJ. Deep learning in Alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data, Frontiers in Aging Neuroscience. 2019;11.

3. Wen J, et al. Convolutional neural networks for classification of Alzheimer's disease: overview and reproducible evaluation, Medical Image Analysis. 2020;63:101694.

4. Altinkaya E, Polat K, Barakli B. Detection of Alzheimer ' s disease and dementia states based on deep learning from MRI images: a comprehensive review. 2020;39–53.

5. Physicians PC. 2020 Alzheimer's disease facts and fgures. Alzheimer's Dement. 2020;16(3):391–460.

6. Yang Y, Li X, Wang P, Xia Y, Ye Q. Multi-Source transfer learning via ensemble approach for initial diagnosis of Alzheimer's disease, IEEE J Transl Eng Heal Med. 2020;1–10.

7. Adelina C. The costs of dementia: advocacy, media, and stigma. Alzheimer's Dis Int World Alzheimer Rep. 2019;2019:100–1.

8. Pulido MLB, Hernández JBA, Ballester MAF, González CMT, Mekyska J, Smékal Z. Alzheimer's disease and automatic speech analysis: a review, Expert Syst Appl.2020;150.

9. Irankhah E. Evaluation of early detection methods for Alzheimer's disease. 2020;4(1):17–22.

10. He Y, et al. Regional coherence changes in the early stages of Alzheimer's disease: a combined structural and resting-state functional MRI study. Neuroimage. 2007;35(2):488–500.

11. Vemuri P, Jones DT, Jack CR. Resting-state functional MRI in Alzheimer's disease. Alzheimer's Res Ther. 2012;4(1):1–9.

12. Bron EE, et al. Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: the CADDementia challenge. Neuroimage. 2015;111:562–79.

13. Klöppel S, et al. Accuracy of dementia diagnosis — a direct comparison between radiologists

and a computerized method. Brain. 2008;131(11):2969–74.

14. Segato A, Marzullo A, Calimeri F, De Momi E. Artificial intelligence for brain diseases: a systematic review, APL Bioeng. 2020;4;4.

15. Yamanakkanavar N, Choi JY, Lee B. MRI segmentation and classifcation of the human brain using deep learning for diagnosis of Alzheimer's disease: a survey. Sensors (Switzerland). 2020;20(11):1–31.

16. Noor MBT, Zenia NZ, Kaiser MS, Al Mamun S, Mahmud M. Application of deep learning in detecting neurological disorders from magnetic resonance images: a survey on the detection of Alzheimer's disease, Parkinson's disease, and schizophrenia, Brain Informatics, 2020;7(1):11.

17. Li F, Tran L, Thung K-H, Ji S, Shen D, Li J. A robust deep model for improved classifcation of AD/MCI patients. IEEE J Biomed Health informatics. 2015;19(5):1610–6.

18. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E. Deep learning for computer vision: a brief review. Comput Intell Neurosci. 2018;2018:1–13.

19. Payan A, Montana G. Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. 2015;1–9.

20. Jarrett K, Kavukcuoglu K, Ranzato MA, LeCun Y. What is the best multi-stage architecture for object recognition?, Proc IEEE Int Conf Comput Vis pp. 2146–2153, 2009.

21. Sarraf S, Tofighi G. Classification of Alzheimer's disease structural MRI data by deep learning convolutional neural networks. 2016;8–12.

22. Hosseini-asl E, Keynton R, El-baz A. Alzheimer's disease diagnostics by adaptation of 3d convolutional network Electrical and Computer Engineering Department, University of Louisville, Louisville, KY, USA, Proc. - Int Conf Image Process ICIP. 2016;(502).

23. Korolev S, Safullin A, Belyaev M, Dodonova Y. Residual and plain convolutional neural networks for 3d brain MRI classifcation Sergey Korolev Amir Safullin Mikhail Belyaev Skolkovo Institute of Science and Technology Institute for Information Transmission Problems, 2017 IEEE 14th Int. Symp. Biomed. Imaging (ISBI 2017). 2017;835–838.