pxpx **People's Democratic Republic of Algeria** pxpx
Ministry of Higher Education and Scientific Research

University of AKLI MOHAND OULHAJ –Bouira-

Faculty of Science and Applied Sciences

Computer science department

Master's thesis

in Computer science

speciality: *Computer Systems Engineering*

Theme

Deep Learning-CNN based Alzheimer's disease diagnosis

Supervised by

- WAHIBA ALIOUAT

Submitted by

- DAHAS Abdellah
- GUEMRAOUI Nour Elhouda

2023/2024

Appreciation

First of all, we thank our God for giving us the strength and courage to carry out this work.

We extend our deepest gratitude to our thesis advisor, WAHIBA ALIOUAT, for their invaluable guidance, expertise, and unwavering support. Their mentorship and constructive feedback have played a pivotal role in shaping the direction and quality of this research work.

We express our heartfelt appreciation to the esteemed professors of the Department of Computer Science for their invaluable guidance and unwavering support throughout our academic journey. Their expertise and passion for teaching have inspired us to excel in the field of computer science. We are grateful for their dedication to our learning.

We are also grateful to the members of our thesis committee for their time, expertise, and valuable insights. Their thoughtful comments and suggestions have significantly enhanced the depth and rigor of our study.

Our appreciation extends to our colleagues and friends for their support, collaboration, and stimulating discussions throughout this endeavor. Their contributions and shared knowledge have enriched our understanding and broadened our perspectives.

Finally, we thank our families for their unwavering support and encouragement. Their love and understanding have been a source of strength and motivation throughout this journey.

Dedications

This thesis is dedicated to those who made this journey possible.

To my parents, for their unconditional love, unwavering support, and constant encouragement. Your faith in me has given me the strength to persevere.

To my family, including my grandparents and my aunt, for their moral support and wise advice throughout this journey. You have been an invaluable source of comfort and motivation.

To my partner, for their invaluable help and unwavering dedication in the realization of this thesis. Your support and collaboration have been essential to this success.

To my classmates, for their steadfast support and camaraderie. You have transformed this challenge into a more enriching and less solitary experience.

Finally, to all my friends, for their constant presence and unwavering support. Your encouragement helped me through difficult times and enabled me to achieve this goal.

Dahas Abdellah.

Dedications

This thesis is dedicated to my family, whose unwavering support and love have been my greatest source of strength.

To my parents and sister, Mom and Dad, your endless encouragement and belief in me have made this journey possible. Your sacrifices and guidance have shaped me into who I am today.

To my teacher, your wisdom and patience have been invaluable. Your dedication to my growth and success is deeply appreciated.

To my partner, i express my sincere gratitude for your dedication and teamwork. Your commitment has been instrumental in our joint successes, and I look forward to continuing our productive collaboration.

To my friends, without whom I would not have had the courage and perseverance to reach this milestone. Your support, laughter, and companionship have made all the difference.

Thank you all for being a part of this incredible journey.

Guemraoui Nour Elhouda.

Abstract

This thesis aims to utilize Convolutional Neural Networks (CNNs) for the detection of Alzheimer's disease (AD) using Magnetic Resonance Imaging (MRI). Alzheimer's disease, a prevalent neurodegenerative disorder, presents significant diagnostic challenges, necessitating advanced methodologies.

This work integrates deep learning models to enhance classification capabilities, including data preprocessing and augmentation techniques like Synthetic Minority Oversampling Technique (SMOTE). The thesis also explores the development and optimization of CNN variants tailored for AD detection.

Experimental results demonstrate the potential of CNNs in improving diagnostic accuracy, highlighting the importance of machine learning in medical diagnostics. This work underscores the critical role of technological advancements in neuroimaging and artificial intelligence.

Résumé

Ce mémoire vise à utiliser les réseaux de neurones convolutionnels (CNN) pour détecter la maladie d'Alzheimer (MA) à l'aide de l'imagerie par résonance magnétique (IRM). La maladie d'Alzheimer, un trouble neurodégénératif répandu, présente des défis significatifs en termes de diagnostic, nécessitant des méthodologies avancées.

Ce travail intègre des modèles d'apprentissage profond pour améliorer les capacités de classification, incluant des techniques de prétraitement et d'augmentation des données comme le suréchantillonnage synthétique des minorités (SMOTE). Ce mémoire explore aussi le développement et l'optimisation de variantes CNN adaptées à la détection de la MA.

Les résultats expérimentaux montrent le potentiel des CNN à améliorer la précision du diagnostic, soulignant l'importance de l'apprentissage automatique dans les diagnostics médicaux. Ce travail met en avant le rôle critique des avancées technologiques en neuroimagerie et en intelligence artificielle.

ملخص

تهدف هذه الأطروحة إلى استخدام الشبكات العصبية التلافيفية (CNNs) لاكتشاف مرض الزهايمر (AD) باستخدام التصوير بالرنين المغناطيسي (MRI). يمثل مرض الزهايمر، وهو اضطراب تنكسي عصبي شائع، تحديات تشخيصية كبيرة تتطلب منهجيات متقدمة.

يعمل هذا البحث على دمج نماذج التعلم العميق لتعزيز قدرات التصنيف، بما في ذلك تقنيات المعالجة المسبقة للبيانات وتكبيرها مثل تقنية التوزيع الأقلي الاصطناعية (SMOTE). كما تستكشف الأطروحة تطوير وتحسين متغيرات CNN المصممة خصيصًا لاكتشاف مرض الزهايمر.

تظهر النتائج التجريبية الإمكانات الواعدة للشبكات العصبية التلافيفية في تحسين دقة التشخيص، مما يبرز أهمية التعلم الآلي في التشخيص الطبي. يشدد هذا العمل على الدور الحيوي للتطورات التكنولوجية في التصوير العصبي والذكاء الاصطناعي.

Contents

Table of contents **i**

Table of figures **iv**

List of tables **vi**

Abbreviations list **vii**

General introduction **1**

1 Generalities about alzheimer's disease. **4**

1.1 Introduction 4

1.2 Definition of alzheimer's disease 4

1.3 Symptoms of alzheimer's disease 5

1.4 Classification of alzheimer's disease 6

1.4.1 Mild stage 6

1.4.2 Moderate stage 6

1.4.3 Advanced stage 6

1.5 Diagnosis of alzheimer's disease 7

1.5.1 Imaging Techniques in Alzheimer's Disease Diagnosis and Monitoring 7

1.6 Conclusion 8

2 State of the art **9**

2.1 Introduction 9

2.2 Basic principles of machine learning 9

2.3 Fundamental concepts of deep learning 10
2.3.1 Comparison between machine learning and deep learning 10
2.4 Deep learning algorithms 11
2.4.1 Recurrent neural networks 11
2.4.2 Generative adversarial networks 12
2.4.3 Convolutional Neural Networks 13
2.5 Architecture of Convolutional Neural Networks 14
2.5.1 Different types of layers used in a CNN 14
2.5.2 Choice and adjustment of hyperparameters to optimize model performance 17
2.5.3 Overfitting 18
2.6 Pre-trained convolutional neural networks 19
2.6.1 Transfert learning 19
2.6.2 The potential impact of CNNs in the health and medical field . . . 23
2.7 Advancements in Alzheimer's Disease Detection Using CNN: Related work 23
2.8 Conclusion 24

3 Our proposed approach 26
3.1 Introduction 26
3.2 Dataset 26
3.3 Our proposed framework 28
3.3.1 Data preprocessing phase 29
3.3.2 Model architecture 31
3.3.3 Our distinct CNN variants 32
3.4 Conclusion 34

4 Experiments, reults, and discussion 36
4.1 Introduction 36
4.2 Experiments configurations 36
4.3 Experimental material and platforms 37
4.3.1 Kaggle 37
4.3.2 Keras 37
4.3.3 TensorFlow 37

4.3.4 Pandas 38
4.3.5 Matplotlib 38
4.4 Performance metrics 38
4.5 Results and discussion 39
4.6 Our variants 40
4.6.1 PTW-TL-SMOTE: Pre-Trained Weights-Trainable Layer-With SMOTE 40
4.6.2 PTW-TL-SMOTE: Pre-Trained Weights-Trainable Layer-With SMOTE 43
4.6.3 PTW-NTL-SMOTE:Pre-Trained Weights-Non-Trainable Layer-With SMOTE 46
4.6.4 PTW-NTL-SMOTE:Pre-Trained Weights-Non-Trainable Layer-Without SMOTE 49
4.7 Comparison of Variants 52
4.7.1 Reasons for the Superiority of the PTW-TL-SMOTE Variant . . . 53
4.8 Comparison related work 54
4.9 Conclusion 55

General conclusion **57**

Bibliography **59**

List of Figures

1.1 Coronal section of the brain of a normal individual and a person with Alzheimer's disease[1] . . . 5
1.2 Types of AD [2] . . . 7
1.3 The principle and function of MRI [3] . . . 8

2.1 Machine Learning Deep Learning [4] . . . 10
2.2 Machine Learning vs Deep Learning [5] . . . 11
2.3 Recurrent neural networks [6] . . . 12
2.4 Generative adversarial networks [7] . . . 13
2.5 Convolutional Neural Networks Architecture[8] . . . 14
2.6 Convolution layer[9] . . . 14
2.7 Max Pooling Layer [10] . . . 15
2.8 Fully connected layers [11] . . . 16
2.9 Normalization layers [12] . . . 17
2.10 VGG-16 Archetecture [13] . . . 20
2.11 MobileNetV2 Archetecture [14] . . . 21
2.12 Xception Archetecture [14] . . . 21
2.13 MobileNet Archetecture [15] . . . 22
2.14 InceptionV3 Archetecture [16] . . . 22

3.1 Example image for Mild Demented class. . . . 27
3.2 Example image for Moderate Demented class. . . . 27
3.3 Example image for Non Demented class. . . . 28
3.4 Example image for Very Mild Demented class. . . . 28

3.5 Our proposed framework. 29

4.1 Bar graph FS-TL-SMOTE 41
4.2 curves MobileNet FS-TL-SMOTE 42
4.3 curves MobileNetV2 FS-TL-SMOTE 42
4.4 curves InceptionV3 FS-TL-SMOTE 42
4.5 curves Xception FS-TL-SMOTE 43
4.6 curves VGG16 FS-TL-SMOTE 43
4.7 Bar graph PTW-TL-SMOTE 44
4.8 Curves InceptionV3 PTW-TL-SMOTE 45
4.9 Curves MobileNet PTW-TL-SMOTE 45
4.10 Curves MobileNetV2 PTW-TL-SMOTE 45
4.11 Curves VGG16 PTW-TL-SMOTE 46
4.12 Curves Xception PTW-TL-SMOTE 46
4.13 Bar graph PTW-NTL-SMOTE 47
4.14 Curves VGG16 PTW-NTL-SMOTE 48
4.15 Curves InceptionV3 PTW-NTL-SMOTE 48
4.16 Curves MobileNet PTW-NTL-SMOTE 48
4.17 Curves MobileNetV2 PTW-NTL-SMOTE 49
4.18 Curves Xception PTW-NTL-SMOTE 49
4.19 Bar graph PTW-NTL-NS 50
4.20 Curves VGG16 PTW-NTL-NS 51
4.21 Curves InceptionV3 PTW-NTL-NS 51
4.22 Curves MobileNet PTW-NTL-NS 51
4.23 Curves MobileNetV2 PTW-NTL-NS 52
4.24 Curves Xception PTW-NTL-NS 52

List of Tables

1 Abbreviations and their meanings vii

4.1 FS-TL-SMOTE Result 40
4.2 PTW-TL-SMOTE Result 43
4.3 PTW-NTL-SMOTE Result 46
4.4 PTW-NTL-NS Result 49
4.5 Best performing models for each variant 52
4.6 Comparison related work 54

Abbreviations list

Table 1: Abbreviations and their meanings

AD	Alzheimer's disease
MRI	Magnetic Resonance Imaging
ML	Machine Learning
DL	Deep Learning
CNNs	Convolutional Neural Networks
NN	Neural Networks
AI	Artificial intelligence
RNN	Recurrent Neural Networks
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
SGD	Stochastic Gradient Descent
ReLU	Rectified Linear Unit
SCC	Subjective Cognitive Concerns
MCI	Mild Cognitive Impairment
AUC	Area Under the Curve
PET	Positron Emission Tomography
SMOTE	Synthetic Minority Over-sampling Technique

General introduction

Context and Problem Statement

Neuroscience is rapidly evolving with technological advancements, particularly in the study of Alzheimer's disease (AD). This neurodegenerative condition, affecting millions worldwide, presents significant public health challenges. Despite the lack of effective treatments, research focuses on early detection to mitigate cognitive and behavioral decline. Innovations in neuroimaging and machine learning, especially magnetic resonance imaging (MRI) and convolutional neural networks (CNNs), show promise in predicting and understanding AD before it progresses to advanced stages.

Objective

Therefore, the primary objective of this thesis is to develop methodologies to detect and predict the progression of Alzheimer's disease using MRI and CNNs.

Contribution

In this work, we employed a comprehensive approach involving advanced data preprocessing, augmentation, and the use of state-of-the-art deep learning models. The process included:

- **Data Pre-Processing and Augmentation**: Utilizing techniques such as SMOTE for oversampling to address class imbalances, and applying image augmentation strategies like brightness adjustment, zoom, and horizontal flipping to enhance the dataset.

- **Model Selection**: Implementing several pre-trained CNN models, including Xception, InceptionV3, and MobileNetV2, as base model to identify the most effective architecture for AD detection.

- **Model Customization**: All models were customized by adding additional layers for dropout, global average pooling, batch normalization, and dense layers to improve performance and reduce overfitting. These models were trained using a combination of callbacks, including early stopping based on accuracy and learning rate reduction on plateau.

- **Proposed Variants**: After customizing the models, we proposed several architecture variants to determine the optimal configuration for AD detection. These variants include:

  - **FS-TL-SMOTE**: A model where the weights are trained from scratch with trainable layers and SMOTE applied for class balancing.
  - **PTW-TL-SMOTE**: A model using pre-trained weights with trainable layers and SMOTE.
  - **PTW-NTL-SMOTE**: A model using pre-trained weights with non-trainable layers and SMOTE.
  - **PTW-NTL-NS**: A model using pre-trained weights with non-trainable layers and without SMOTE.

- **Evaluation and Metrics**: The models performance was evaluated using metrics such as categorical accuracy, AUC and F1 score. classification reports were also generated to provide detailed insights into model performance.

- **Comparison with Related Work**: To contextualize our contributions, we reviewed existing research in the field. This includes traditional methods and recent advancements in deep learning and CNN approaches for AD detection. Our methodology aligns with current trends in employing CNNs and MRI data, and we discuss the similarities and differences between our approach and those found in the related work.

Thesis Structure

The remaining sections of the thesis are organized as follows:

- **Chapter 1**: Provides a general overview of Alzheimer's disease and its relationship with MRI imaging.
- **Chapter 2**: Presents a comprehensive state-of-the-art review of current research and advancements related to AD and neuroimaging.
- **Chapter 3**: Describes our proposed methodological approach to detect and predict the progression of AD using MRI and CNNs, detailing the data preprocessing, model training, and evaluation processes.
- **Chapter 4**: Discusses the implementation of our approach, presents the development tools used, and evaluates the results obtained from our experiments.

Chapter 1

Generalities about alzheimer's disease.

1.1 Introduction

This chapter serves as a brief exploration of Alzheimer's disease, focusing on key areas including the disease progression and stages, diagnostic methodologies, symptomatic manifestations, classification frameworks, and why is better to switch from radiology to MRI. By delving into these topics, we aim to provide a basic understanding of AD, elucidating its complexities and offering insights into its diagnosis and management.

1.2 Definition of alzheimer's disease

AD is a neurodegenerative diseases [17] characterized by the gradual deterioration of:

- Cognitive function
- Memory loss
- behavioral changes.

It is the most common cause of dementia, affecting millions of people , particularly aged 65 or older [18]. The disease is marked by the accumulation of abnormal protein deposits in the brain, leading to the death of nerve cells and the disruption of neural pathways responsible for cognition, memory, and behavior. As AD advances, individuals may experience increasing difficulties with everyday tasks and social interactions such as understanding there for communication, in advanced cases AD can significantly impact language abilities. Unfortunately there is no cure for AD, and available treatments aim to manage symptoms and slow disease progression, not to fully recover. Figure 1.1 shows a comparison between a person with AD and another without it in brain regions associated with language and memory.

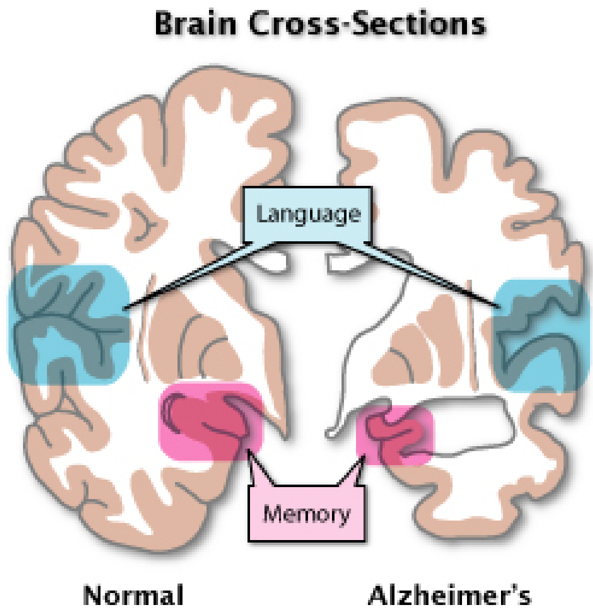

Figure 1.1: Coronal section of the brain of a normal individual and a person with Alzheimer's disease[1]

1.3 Symptoms of alzheimer's disease

AD is characterized by a progressive set of symptoms that primarily affect memory, cognitive abilities, and behavior. Some of the most common symptoms include: [19]

- **Short-term memory loss:** individuals with Alzheimer's often struggle to remember recent events or important details of their daily lives.
- **Mental confusion:** they may feel disoriented in time or space, leading to potentially dangerous situations.
- **Difficulties with reasoning and problem-solving:** Tasks requiring abstract thinking or planning become increasingly challenging.
- **Changes in personality and behavior:** people with Alzheimer's may exhibit mood swings, agitation, irritability, or social withdrawal.

- **Language problems:** they may have difficulty finding the right words to express themselves or understanding the language of others.

1.4 Classification of alzheimer's disease

AD presents three degrees of severity, including Very-Mild Dementia, Mild Dementia, and Moderate Dementia, each stage characterized by distinct symptoms.

The disease can be delineated into three stages:

1.4.1 Mild stage

In the Mild Stage, individuals may experience sporadic short-term memory lapses, struggles and have significantly be disturbed in sleep, 5% of individuals with mild stage maintain stability or revert to a normal state.[20]

1.4.2 Moderate stage

In the Moderate Stage, memory impairments become more serious, they intend to forget only recent information. Individuals may struggle with daily activities a bit more on this stage, and a noticed behavioral disturbances, including language difficulties and occasional aggression, may surface.[21]

1.4.3 Advanced stage

In the advanced stages of AD, individuals often experience a complete loss of awareness of self and surroundings. They may face personality changes. Basic functions such as eating, walking, and toileting become increasingly challenging, requiring full-time assistance. Overall, advanced Alzheimer's profoundly affects theire mental health causing paranoia necessitating comprehensive care and support to maintain their comfort and dignity.[20]

The figure 1.2 Illustrates brain images representing three stages of AD: preclinical, mild to moderate, and severe.

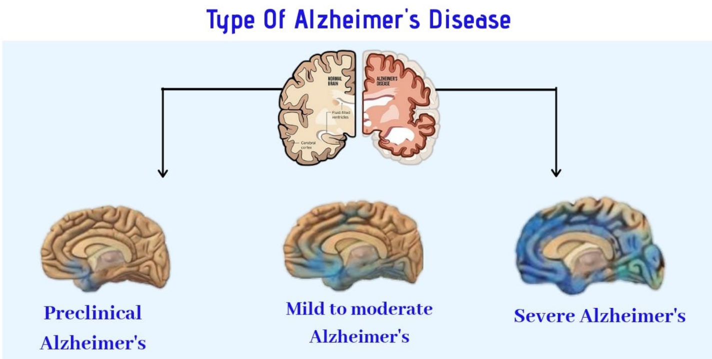

Figure 1.2: Types of AD [2]

1.5 Diagnosis of alzheimer's disease

The diagnosis of AD is a complex process that typically involves several steps: [22]

- **Clinical evaluation:** a physician conducts a comprehensive assessment of the patient's health, asking about symptoms and performing cognitive tests.
- **Exclusion of other causes:** it's essential to rule out other medical conditions that may cause similar symptoms, such as vitamin deficiencies, thyroid disorders, or brain tumors.
- **Additional tests:** blood tests, neurological evaluations, brain imaging tests (such as MRI), and sometimes neuropsychological assessments may be performed to confirm the diagnosis.

1.5.1 Imaging Techniques in Alzheimer's Disease Diagnosis and Monitoring

Conventional imaging techniques, such as X-rays and CT scans (computed tomography), may lack the sensitivity and specificity to detect the subtle changes associated with AD. MRI, however, offers significant advantages. It is a non-invasive technique that provides detailed visualization of brain structures. In the context of AD, MRI can detect structural changes in the brain, such as hippocampal atrophy, at an early stage of the disease [23]. MRI is also particularly useful for tracking the progression of AD over time, enabling doctors to compare images to assess changes in the patient's brain. It offers much better spatial resolution than other imaging modalities, allowing for the visualization of brain structures with high precision. Additionally, the contrast between soft tissues is often superior to that of CT scans, providing more detailed images. Figure 1.3 provides a representation of the principle and function of MRI

This figure (1.3) provides a representation of the principle and function of MRI

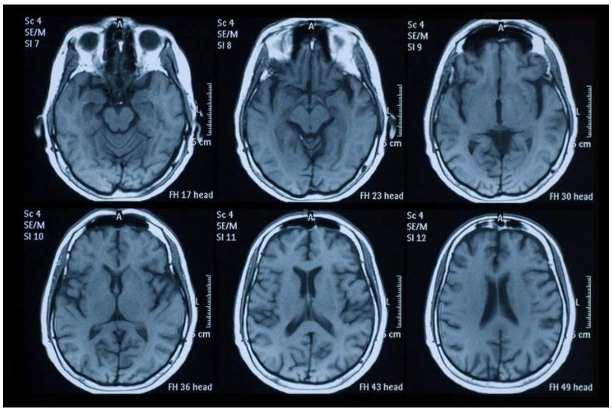

Figure 1.3: The principle and function of MRI [3]

1.6 Conclusion

In conclusion, this chapter has provided an overview of AD, covering its definition, classification, symptoms, and the crucial role of imaging techniques, particularly MRI, in its diagnosis and monitoring. MRI stands out for its ability to detect early structural changes in the brain, offer superior spatial resolution, and track disease progression. The next chapter will delve into CNNs and Deep Learning, which will be utilized in the processing of medical images associated with AD.

Chapter 2

State of the art

2.1 Introduction

In this chapter, we delve into the realm of DL, particularly focusing on CNNs and their application in the detection and classification of AD using medical imaging. We begin by exploring the foundational concepts of DL and its distinction from traditional ML approaches. Through a brief overview, we highlight the significance of CNNs in revolutionizing medical image analysis.

2.2 Basic principles of machine learning

ML constitutes a subset of AI, focused on enabling machines to evolve and fulfill tasks through learning processes, a feat unattainable through conventional algorithms alone. ML algorithms learn to solve problems automatically by analyzing data, aiming to minimize errors and enhance accuracy. ML was first introduced by Arthur Samuel[24], who defined it as an area of research and development within computer science. It's not just a single technique or method but an entire field dedicated to creating systems that are given the flexibility to learn from data without having specific instructions programmed for every situation they might encounter. Among the most adopted ML methods are supervised and unsupervised learning.[25]

2.3 Fundamental concepts of deep learning

DL, a subset of ML[26], represents a revolutionary approach to AI, enabling machines to learn autonomously from data without explicit programming. It is based on neural networks architectures inspired by the structure and function of the human brain. DL algorithms utilize multi-layered networks of artificial neurons to process information iteratively, with each layer refining and enhancing the output of the previous layer. This hierarchical learning mechanism enables the system to extract complex patterns and features from vast datasets, leading to remarkable advancements in various domains such as computer vision, natural language processing, and speech recognition. In figure 2.1, we can observe the relationship between the two concepts mentioned, ML and DL.

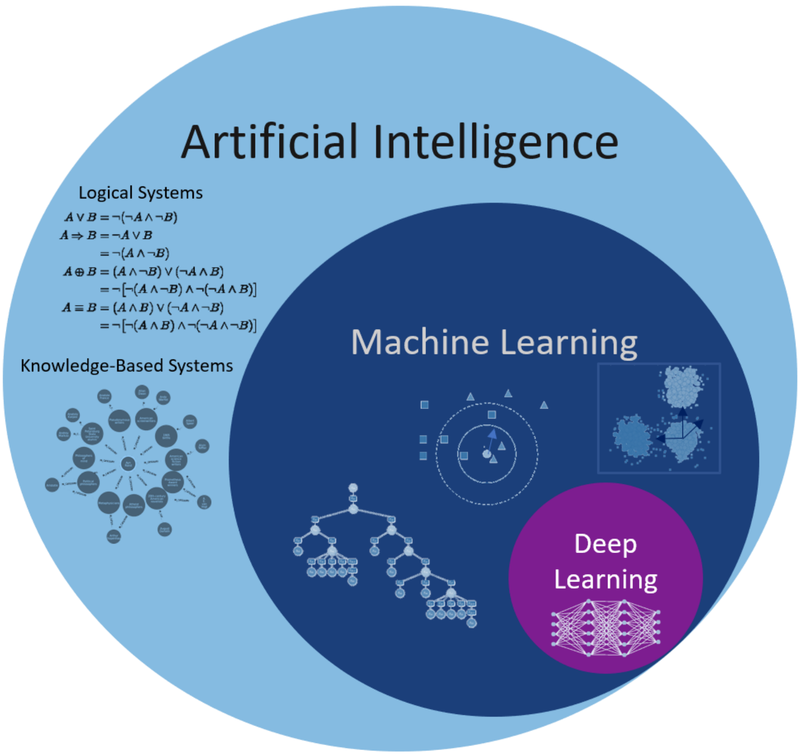

Figure 2.1: Machine Learning Deep Learning [4]

2.3.1 Comparison between machine learning and deep learning

First and foremost, it's imperative to recognize that DL operates within the broader realm of ML, as illustrated in Figure 2.1.

Another notable contrast between DL and traditional ML lies in the adaptability of DL algorithms. Unlike many ML techniques that face constraints in handling large datasets, DL models thrive with increased data, thereby exhibiting enhanced performance. DL models have been known to surpass human-level performance in domains such as image processing [27], owing to their capability to handle vast amounts of data without theoretical limitations.

Figure 2.2 underscores the reduced necessity for human intervention in DL compared to traditional ML methods. DL algorithms possess the ability for automatic feature extraction through their neural networks architecture, eliminating the need for manual feature identification by software engineers—a laborious and specialized task.

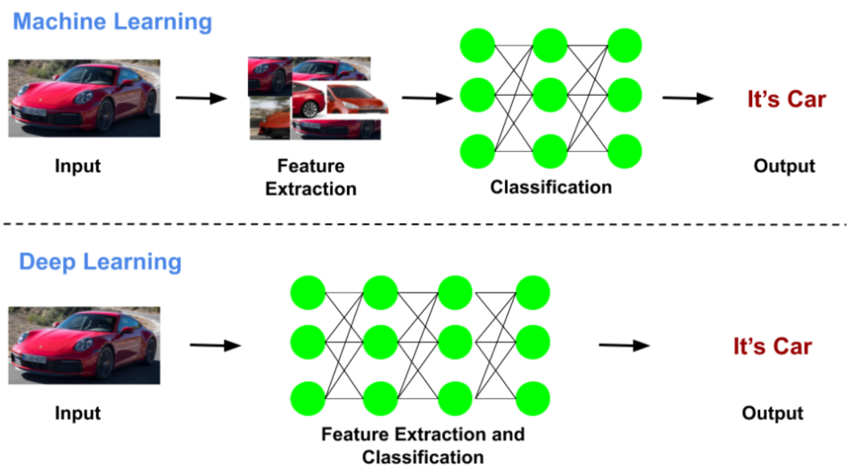

Figure 2.2: Machine Learning vs Deep Learning [5]

Moreover, the complex multi-layered structure of DL necessitates a larger volume of data for optimal functionality. This aspect is crucial for mitigating fluctuations and ensuring high-quality interpretations.

2.4 Deep learning algorithms

DL comprises a multitude of algorithms, each tailored to specific application domains. Here, we'll introduce several, with our primary focus being on detecting AD in images. This involves the realm of computer vision, utilizing CNNs for image recognition.

2.4.1 Recurrent neural networks

RNNs are pivotal in DL, especially in the realm of home automation. Their strength lies in their ability to process sequential data for tasks such as speech recognition, translation, and time-series prediction. Unlike traditional networks, RNNs retain memory of past inputs through feedback loops, enabling them to analyze data sequences and make predictions based on previous context. However, they face challenges such as the vanishing gradient problem. To address this, variants like Long Short-Term Memory (LSTM) networks have been developed, enhancing information retention over long sequences[28]. The figure 2.3 shows a recurrent neural network with three types of layers: input, hidden, and output layers.

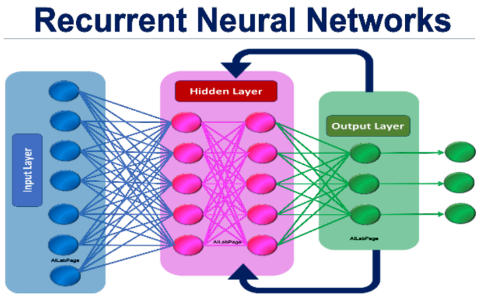

Figure 2.3: Recurrent neural networks [6]

2.4.2 Generative adversarial networks

GANs are powerful in generating lifelike images, unlike CNNs primarily used for recognition. GANs feature two neural networks, a generator and a discriminator, engaged in a competitive process. The generator produces images to fool the discriminator, which distinguishes real from fake. In medical imaging, GANs show promise for creating realistic synthetic images, improving training data quality and diversity for disease detection models, leading to enhanced performance[29].

In Figure 2.4, the GAN architecture is showcasing the flow of random input and real images through the generator and discriminator, with associated losses for both, resulting in the generation of authentic samples.

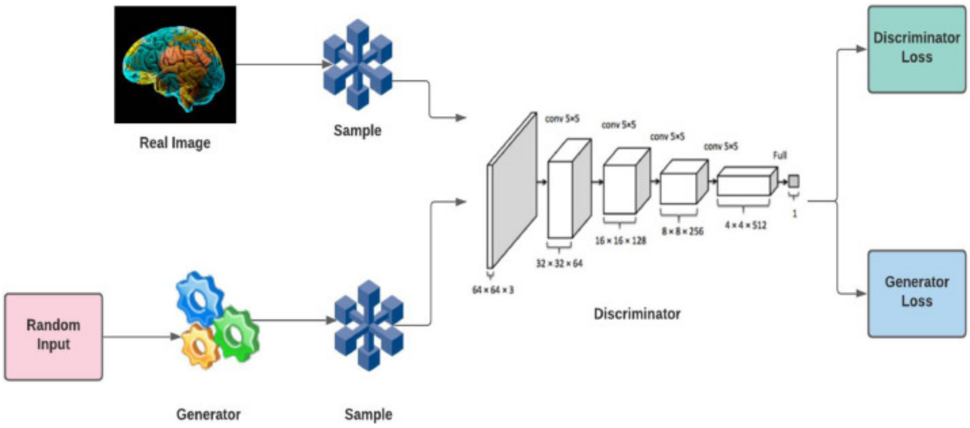

Figure 2.4: Generative adversarial networks [7]

2.4.3 Convolutional Neural Networks

CNNs have transformed AI, especially in computer vision, inspired by the human visual system. They excel in tasks like image recognition[30]. CNNs are utilized also in various other domains such as medical imaging, robot technology, predict machine failures, analyze financial transactions for fraud detection.

A typical CNN architecture includes convolutional layers for feature extraction, subsampling layers for efficiency enhancement, fully connected layers for classification, and normalization layers. Unlike traditional neural networks, CNNs operate on pixelated images, with neurons connected to specific regions, reducing training time and overfitting while improving pattern learning.[31]

In our focus on disease detection in medical images, a deep understanding of CNNs is crucial due to their expertise in computer vision tasks. By leveraging CNN capabilities, we aim to enhance disease detection methods and propel advancements in medical imaging technology.

Figure 2.5 shows the CNN archirecture.

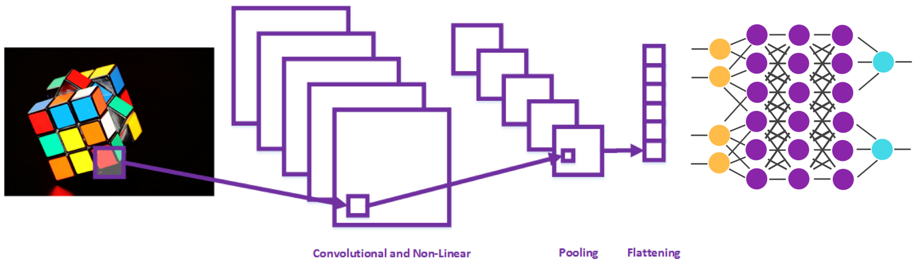

Figure 2.5: Convolutional Neural Networks Architecture[8]

2.5 Architecture of Convolutional Neural Networks

2.5.1 Different types of layers used in a CNN

In a CNN, various types of layers are employed to process and extract features from input data.

Convolutional layers

These layers apply convolution operations to the input data, extracting local features through filters or kernels. Convolutional layers play a pivotal role in feature extraction by convolving the input with learnable filters[32].
Figure 2.6 shows the pixel transformation with convolutional kernels.

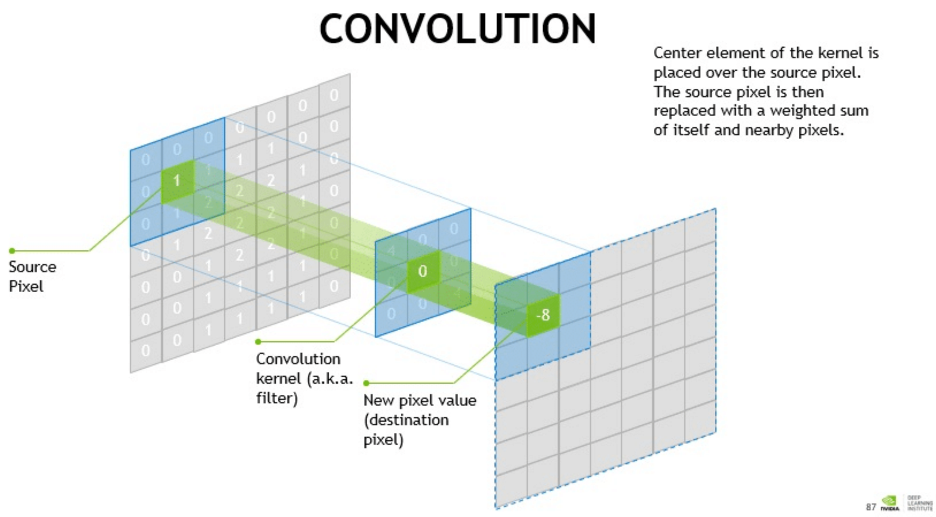

Figure 2.6: Convolution layer[9]

Pooling layers

Pooling layers reduce the spatial dimensions of the feature maps generated by convolutional layers. They help in capturing the most important features while reducing computational complexity and preventing overfitting. Common pooling operations include max pooling and average pooling[33].

The figure 2.7 illustrates the process of max pooling, where each 2x2 block in a 4x4 matrix is reduced to a single value representing the maximum value in that block.

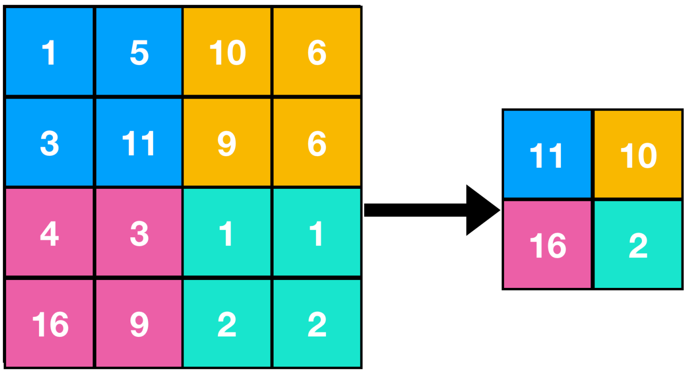

Figure 2.7: Max Pooling Layer [10]

Activation layers

play a crucial role in CNNs by introducing non-linearities between layers, impacting the network's performance [34]. Various activation functions have been explored, such as ReLU . These functions influence the network's ability to learn complex patterns and make decisions based on the input data. Additionally, the activation patterns across different layers in CNNs encode valuable information about spatial relations, temporal patterns, and features' co-occurrence, enhancing the network's performance in tasks like visual tracking. The choice and combination of activation functions in CNNs significantly impact the network's efficiency and effectiveness in various applications

Fully Connected Layers and Classification

Fully connected layers are utilized towards the end of the CNN architecture for classification or regression tasks [35]. These layers form a dense network where each neuron is connected to every neuron in the subsequent layer. These layers are critical for interpreting features extracted from medical images and classifying patients into different categories. Within these layers, the softmax activation function is often employed to convert raw scores into probabilities, thus determining the predicted class for each input. Therefore, fully connected layers with softmax act as the final classifier in the CNN, assigning a class to each input based on features learned by the network. in a figure 2.8 illustrating the fully connected layer.

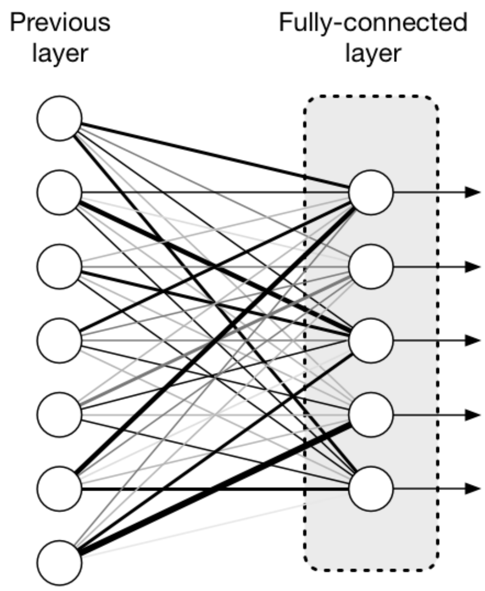

Figure 2.8: Fully connected layers [11]

Additional Layers for Enhanced CNNs

In addition to the fundamental layers discussed above, there are several additional layers that can be incorporated into CNN architectures to enhance their performance. These layers include:

- **Correction layers (Normalization layers)**: Correction layers, such as batch normalization or layer normalization, are used to improve the training stability and speed of CNNs by normalizing the input of each layer. They reduce internal covariate shift and accelerate convergence during training[36].
  The figure 2.9 demonstrates batch normalization and layer normalization, showing their application on samples and units within a neural network.

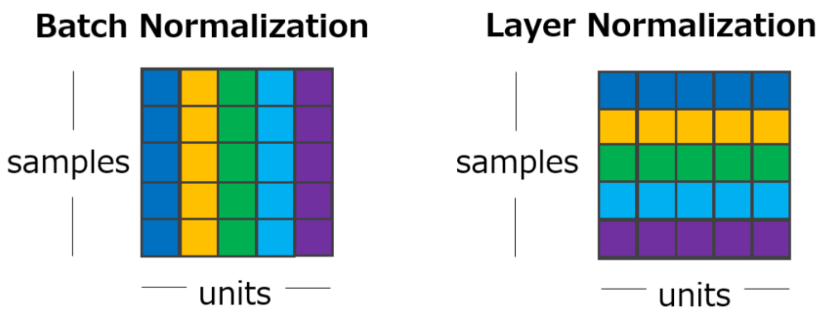

Figure 2.9: Normalization layers [12]

- **Dropout layers**: Dropout layers play a pivotal role in CNNs to prevent overfitting by randomly ignoring a certain percentage of neurons during training [37]. Dropout layers provide a mechanism to enhance model generalization and robustness. By randomly dropping out neurons, dropout layers promote the learning of more robust features, leading to improved performance on unseen data. This regularization technique is particularly beneficial in scenarios where the dataset is limited or when the model exhibits signs of overfitting. In the architecture of CNNs, dropout layers are strategically placed to mitigate overfitting and improve the network's ability to generalize to new data.

2.5.2 Choice and adjustment of hyperparameters to optimize model performance

Hyperparameters in a CNN are parameters that are set before the learning process begins. They greatly influence the network's learning process and its ability to generalize well to unseen data.[38] Here are some key hyperparameters :

- The learning rate is a crucial hyperparameter that controls the step size during optimization. Setting it too high can cause overshooting of the minima, while a value too low can significantly slow down convergence. Therefore, careful tuning is essential for achieving optimal performance [39].

- The number of filters determines the depth of the output volume in convolutional layers. Increasing the number of filters allows the network to capture more diverse features, but it also comes with a trade-off of increased computational complexity.

- The number of epochs refers to the number of times the entire dataset is passed forward and backward through the Neural Networks.
- The kernel size refers to the dimensions of the filter applied to the input data. It affects both the size of the receptive field and the granularity of detected features [40].
- Stride determines how much the filter moves after each operation. Larger strides result in a reduction of the spatial dimensions of the output volume [41].
- Activation functions introduce non-linearity into the neural network. Common choices include ReLU, Sigmoid, or Tanh [42].
- Pooling size determines the size of the window over which max or average pooling is applied. It affects the amount of downsampling and information retention [43].
- The dropout rate represents the fraction of units to drop during training. It helps prevent overfitting by introducing randomness and encouraging strong feature learning [44].
- Batch size refers to the number of samples processed before updating the model's weights. Larger batches may lead to faster convergence but require more memory [45].
- The number of layers in a neural network impacts its ability to learn hierarchical representations. Deeper networks can capture more complex patterns but are prone to issues like vanishing gradients.
- In weight initialization, setting the starting point for a neural network's learning journey is important. This process, called weight initialization, assigns initial values to the network's connections. By carefully choosing these starting points, we can help the network learn faster and avoid problems like exploding or vanishing gradients.[46]

2.5.3 Overfitting

Regularization techniques play a vital role in training CNNs effectively, improving their ability to generalize well to unseen data. Here are some key methods:

- Data Augmentation: This technique employs artificial manipulation of existing data to create new variations. By enriching the training dataset with these synthetic samples, the model encounters a broader range of features, reducing overfitting.
- Over-sampling: SMOTE is a technique in ML used to address imbalanced datasets. It creates synthetic data points for the minority class.[47]
- Dropout: In neural networks, by randomly deactivating a subset of neurons, the network is forced to develop robust feature representations that are not dependent on any single unit.
- Batch Normalization: Normalizing the activations within each layer during training, batch normalization stabilizes the learning process and facilitates faster convergence. It addresses the issue of exploding or vanishing gradients[36].
- Pre-trained Weights: Leveraging pre-trained weights from a successful model on a related task can accelerate the training process. These weights provide a strong foundation of learned features, which can be fine-tuned for the specific requirements of the new task.

By exploring these regularization techniques, model generalization can be significantly improved. The goal is to create a model that not only performs well on the training data but also demonstrates strong performance on unseen data, ensuring its effectiveness in real-world applications.

2.6 Pre-trained convolutional neural networks

2.6.1 Transfert learning

Transfer learning is a technique in DL where a pre-trained CNN model is used as a starting point for a new task. The pre-trained model is ”transferred” to the new task by using its learned features as the initial weights for the new model, and then fine-tuning the model on the new task’s dataset. This approach has several advantages, such as reducing the amount of data and computational resources required for training the new model, and often leading to improved performance compared to training a model from scratch.

Transfer learning is commonly used in computer vision tasks where there is a shortage of labeled data.

Some popular pre-trained CNNs for transfer learning include:

VGG16 : VGG16 is a CNN architecture introduced in 2014, characterized by its depth with sixteen layers of convolution and pooling, followed by three fully connected layers for classification. It was trained on the ImageNet dataset, demonstrating outstanding performance in object recognition[35].

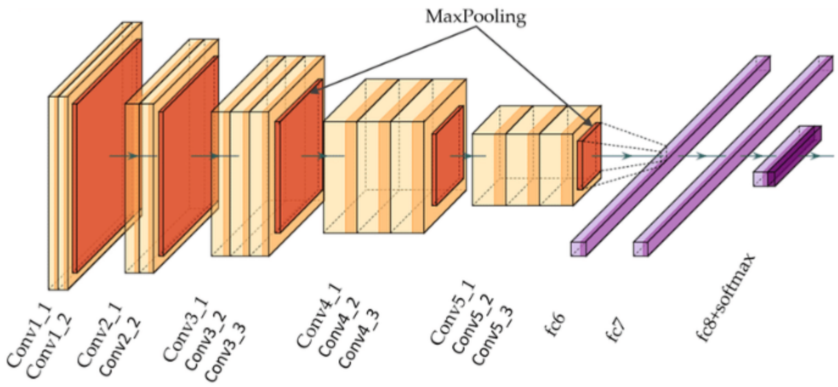

Figure 2.10: VGG-16 Archetecture [13]

MobileNetV2 : It's is a lightweight CNN architecture designed for mobile and embedded devices, offering a balance between accuracy and speed. Optimized to run on lightweight computational devices, it employs techniques such as depth-wise separable convolutions and residual connections to enhance efficiency and performance. MobileNetV2 has shown the ability to deliver accurate results while requiring fewer computations than traditional architectures, making it popular for image classification tasks on mobile platforms [48].

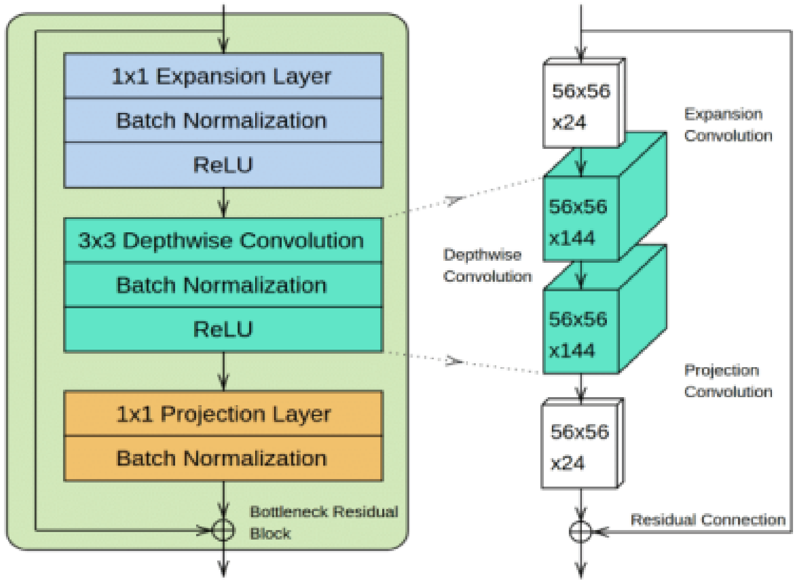

Figure 2.11: MobileNetV2 Archetecture [14]

Xception : Xception is a deep neural network design that uses depth-wise separable convolutions instead of conventional convolutional layers, requiring fewer parameters and computations. It utilizes skip and residual connections and performs well on image classification and object detection tasks, achieving state-of-the-art results [49].

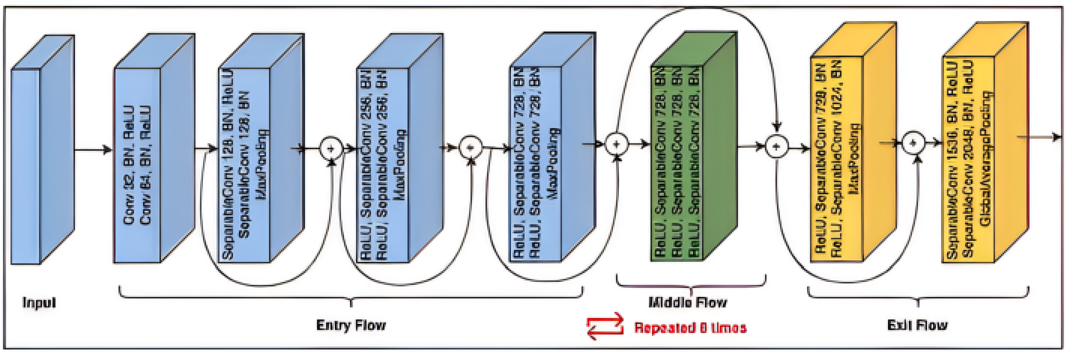

Figure 2.12: Xception Archetecture [14]

MobileNet : MobileNet is a lightweight CNN model designed for image classification on mobile platforms. It aims to effectively extend deep learning applications to peripheral devices. MobileNet is known to outperform traditional CNN models. These advances demonstrate the effectiveness of MobileNet in improving image classification tasks on devices with limited resources.[50]

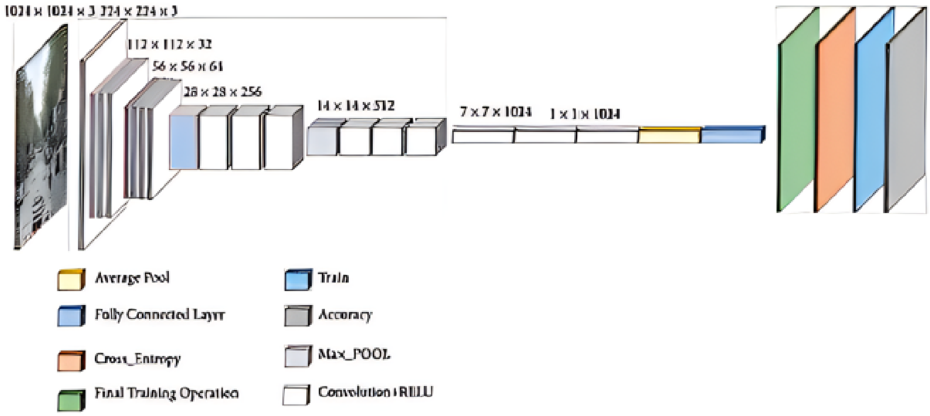

Figure 2.13: MobileNet Archetecture [15]

InceptionV3 : InceptionV3 is a deep Neural Networks that uses a combination of convolutional layers and max pooling. It has an auxiliary classifier for training and regularization, and performs well on image classification tasks. InceptionV3 has achieved state-of-the-art results and is widely used in research and industry for computer vision applications [51].

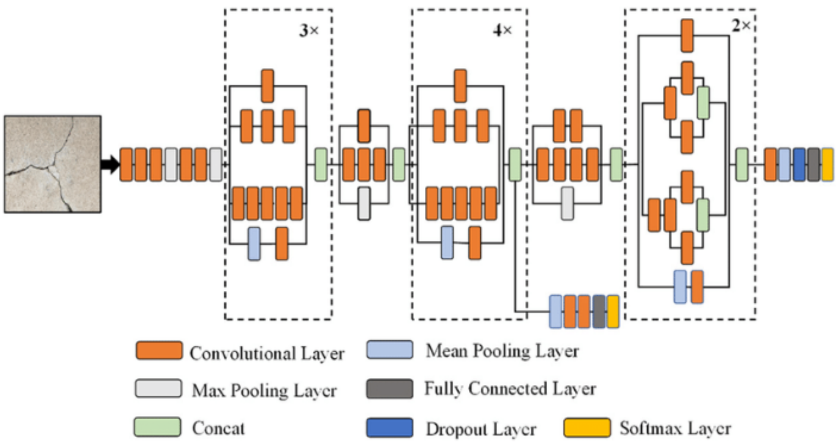

Figure 2.14: InceptionV3 Archetecture [16]

2.6.2 The potential impact of CNNs in the health and medical field

The potential impact of CNNs in the medical and health field is immense, poised to revolutionize diagnosis, treatment, and patient care. These sophisticated networks excel in analyzing complex medical imaging data, aiding in the early detection of diseases such as cancer, Alzheimer's, and cardiovascular conditions. Their ability to extract intricate patterns from images allows for more accurate diagnoses, enabling clinicians to intervene at earlier stages when treatments are often more effective. CNNs also enhance personalized medicine by analyzing genetic data to predict individual responses to medications and therapies. Additionally, they facilitate the development of smart healthcare systems, assisting in remote patient monitoring and real-time health analysis. By harnessing the power of CNNs, the medical community stands to benefit from improved diagnostic precision, optimized treatment strategies, and ultimately, better health outcomes for patients worldwide.

The utilization of CNNs in AD detection offers several advantages:

- **Facilitating early diagnosis:** CNNs assist clinicians in identifying AD at its incipient stages, enabling timely interventions for better patient outcomes.
- **Providing quantitative measures:** They furnish quantitative measures of AD progression, aiding clinicians in monitoring the disease's trajectory and evaluating treatment efficacy.
- **Enabling timely interventions:** By providing early and accurate detection, CNNs empower healthcare providers to intervene promptly, potentially slowing down disease progression and enhancing patient quality of life.

2.7 Advancements in Alzheimer's Disease Detection Using CNN: Related work

Several studies have explored the application of CNNs in AD detection using MRI data, including brain image classification for AD detection, segmentation of brain regions affected by AD, and even progression and prediction of the disease.

In rescent studies, [52] employed a hybrid deep-learning model called SegResNet, combining SegNet and ResNet-101 to improve the accuracy which was recorded for 98%.[53] Proposed to train their data on a custom-designed CNN model, the architecture consisted of four convolutional layers and two fully connected layers and it was trained using a 10-fold cross-validation technique to ensure robust evaluation. This model achieved an accuracy of 93.45%.In [54],the authors utilized the MobileNetV2 architecture as the backbone for feature extraction, input images were resized to match the input dimensions required by the model.The results indicated classification accuracy of 92.7%.

Polat et al.[55] employed a model where information flows through layers in a specific order, designed in a sequence of convolutional (conv2D) and max pooling (maxPooling2D) layers, followed by dense layers.The model's accuracy was 96.35%. Ebrahimi et all, [56] used a pre-trained 2D CNN and achieved 91.78% of accuracy.

In some research [18] [57] [58] [59] [60] [61] that used the same data set as ours[62], Adeola Ajagbe et all.[60] classified AD using DCNN and transfer learning techniques, they trained the data on three models and compared their performance: VGG-19 acheived the accuracy of 77.66%,VGG-16 acheived 77.04% and DCNN got 71.02%. Altwijri et all. [58], employed transfer learning by utilizing a pre-trained EfficientNetB0 model to classify AD severity from MRI images, the model attained an accuracy of 89.7%. [61], introduced DEMNET architecture, a deep learning model that offers an approach for automatic and potentially earlier diagnosis of AD. To address the issue of class imbalance in the dataset, the researchers employed the SMOTE method, they acheived an accuracy of 95.23% with SMOTE and 85% without it.

2.8 Conclusion

In conclusion, this chapter has delved into the realm of CNNs, discussing their potential impact in the medical field, their advantages and their application in image classification and AD detection. We have highlighted key findings and insights regarding the efficacy of CNNs in these critical areas of research.

Looking ahead, the future prospects for CNNs in the field of DL are promising. CNNs are expected to play an increasingly vital role in various domains, including healthcare, robotics, and autonomous systems. There is potential for significant advancements in CNN architectures, training techniques, and their diverse applications.

However, as with any evolving technology, there are also challenges and opportunities to consider. Future research may face obstacles in further developing CNNs, but there are ample opportunities for innovation, collaboration, and interdisciplinary exploration. By addressing these challenges and capitalizing on opportunities, we can continue to harness the power of CNNs and propel the field of DL forward.

Chapter 3

Our proposed approach

3.1 Introduction

In this chapter, we present our proposed approach for AD detection using CNNs. We focus on a comparative analysis of five distinct methods, each employing five different models. This comprehensive examination covers the entire process from data preprocessing to model architecture and variant configurations. Our goal is to identify the most effective strategies for improving model performance and generalization capability in AD detection.

3.2 Dataset

The data consists of MRI images from the Alzheimer's Dataset (4 class of images) [62].It was established in collaboration with various academic and research institutions to make MRI data accessible to researchers. The main objective of the AD dataset is to facilitate the creation of an accurate model classifying better the disease.

A total of 6400 MRI images of subjects were extracted and classified into four categories. The images were captured over several years. The initial format of the MRI data was DICOM, which was transformed into NIfTI format for processing

The initial dimensions of the MRI images were 256x256 pixels, which were reduced to 128x128 pixels for consistency and to optimize computational efficiency. Normalization applied scaled pixel values between 0 and 1.

Here are the four stages that the dataset comprises:

- **Mild Demented**: This category includes 179 files in the test data and 717 files in the training data. Figure 3.1, shows characteristic signs of mild dementia, including cortical atrophy and enlarged ventricles.

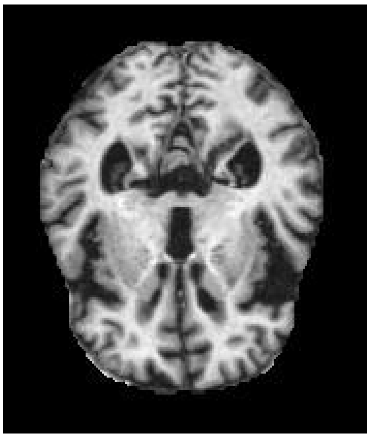

Figure 3.1: Example image for Mild Demented class.

- **Moderate Demented**: With only 12 files in the test data and 52 files in the training data, figure 3.2, exhibits severe cortical atrophy and significant ventricular enlargement, indicating advanced stages of dementia.

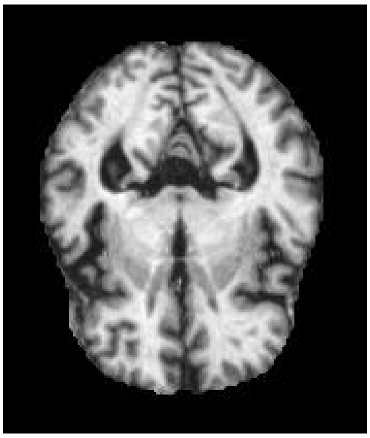

Figure 3.2: Example image for Moderate Demented class.

- **Non Demented**: This class comprises 640 files in the test data and 2560 files in the training data. Figure 3.3 shows no significant signs of cortical atrophy or ventricular enlargement, indicating absence of dementia-related changes.

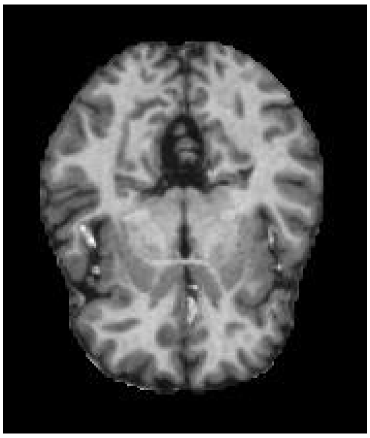

Figure 3.3: Example image for Non Demented class.

- **Very Mild Demented**: With 448 files in the test data and 1792 files in the training data, figure 3.4, shows subtle signs of cortical atrophy and mild ventricular enlargement, indicating early stages of dementia.

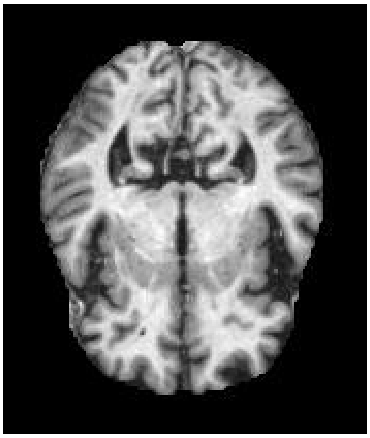

Figure 3.4: Example image for Very Mild Demented class.

The data is organized into two distinct directories: training and test data. Each directory contains a sub-folder for each class, making it easy to manage and access the data for analysis and modeling.

3.3 Our proposed framework

This section outlines our approach, which includes the comparison of four distinct methods, each utilizing five different models, within the context of AD detection. In addition to discussing the considered methods, we will also delve into the data preprocessing phase.

We start by detailing the data preprocessing steps, including normalization, data augmentation, and oversampling, which are crucial for preparing the dataset.
Figure 3.5 Presents our proposed framework:

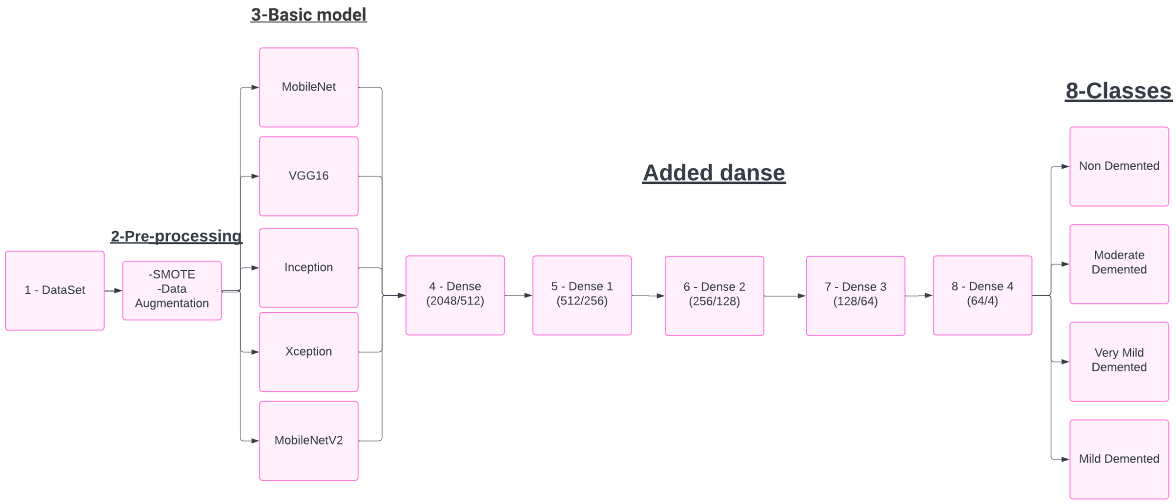

Figure 3.5: Our proposed framework.

We are going to describe our model architecture and the specific layers used to enhance feature extraction and classification. We also will explore different configurations of CNNs, assessing the impact of pre-trained weights, trainable layers, and the application of SMOTE on our Alzheimer's dataset. By comparing these configurations, we aim to provide a robust framework for AD detection that can be adapted and optimized for various clinical and research settings.

3.3.1 Data preprocessing phase

In this section, we describe the various preprocessing steps we performed on our data before using it to train our CNN model.

Image normalization

Normalization of pixel intensities was performed to make the images comparable. This ensures that the pixel values are within the same range, typically between 0 and 1.

```
work_dr = IDG(rescale = 1./255)
```

Data augmentation

The data augmentation phase of our pipeline involves the application of various transformations to the original images, aimed at generating new training data with increased diversity. These transformations include rotation, translation, scaling, flipping, and zooming, among others. By applying these techniques, we create variations in the data, which helps prevent overfitting and enhances the model's ability to generalize to new data.

Specifically, we applied the following techniques to augment our dataset:

- **Rotation range**: Images were randomly rotated within a specified range of angles, introducing variability in the orientation of the objects within the images.
- **Width and height shift range**: Random horizontal and vertical shifts were applied to the images within specified ranges, simulating changes in perspective and viewpoint.
- **Zoom range**: Random zooming in or out of the images was performed, altering the scale of objects within the images.
- **Horizontal flip**: Images were randomly flipped horizontally, further diversifying the dataset by changing the orientation of objects.

Additionally, featurewise centering and featurewise standard normalization techniques were applied to ensure that the training data has zero mean and unit variance. This normalization process helps stabilize the training process and improve model performance by reducing the effects of variations in pixel intensities across images.

Oversampling (SMOTE)

To address class imbalance, particularly with minority classes, the Synthetic Minority Over-sampling Technique was employed. This technique generates synthetic samples for the minority classes to balance the dataset.

```
sm = SMOTE(random_state=42)
```

By incorporating these preprocessing steps into our workflow, we ensure that our data is appropriately prepared for training our CNN model.

3.3.2 Model architecture

The model architecture is constructed using a Sequential API in Keras. Each layer is added sequentially to the model.

Models used

The base model is used as the starting point of the architecture. This pre-trained model, obtained from the Keras applications module, serves as an excellent feature extractor due to its deep convolutional layers. Each method was assessed using five different models to explore the variability of results. The selected models include:

- **Xception**
- **Inception**
- **MobileNetV2**
- **MobileNet**
- **VGG16**

This approach was chosen to evaluate the impact of different model configurations and preprocessing methods on AD detection performance. By comparing the results of each method-model combination, we aim to determine the most effective configurations for our detection task.

Dropout layer

A Dropout layer with a dropout rate of 0.5 is added after the base model. Dropout regularization helps prevent overfitting by randomly dropping a fraction of input units during training.

GlobalAveragePooling2D layer

A GlobalAveragePooling2D layer is added to reduce the spatial dimensions of the feature maps obtained from the base model. This operation computes the average value of each feature map across all spatial locations, resulting in a fixed-length vector.

Flatten layer

The Flatten layer is used to convert the multi-dimensional output of the previous layer into a one-dimensional array. This flattening operation is necessary to connect the output of the GlobalAveragePooling2D layer to the subsequent Dense layers.

BatchNormalization layer

A BatchNormalization layer is inserted after the Flatten layer to normalize the activations of the previous layer. This technique helps stabilize and speed up the training process by reducing internal covariate shift.

Dense layers

Several Dense layers are added to perform further feature extraction and classification. Each Dense layer consists of a specified number of neurons with the ReLU activation function, followed by another BatchNormalization layer and a Dropout layer. This sequence of operations is repeated for each Dense layer, gradually reducing the number of neurons from 512 to 4.

- **Dense(512):** This layer contains 512 neurons with ReLU activation.
- **Dense(256):** This layer contains 256 neurons with ReLU activation.
- **Dense(128):** This layer contains 128 neurons with ReLU activation.
- **Dense(64):** This layer contains 64 neurons with ReLU activation.
- **Dense(4):** The final dense layer consists of 4 neurons with softmax activation, producing probability distributions over the classes for classification.

These Dense layers, along with the preceding BatchNormalization and Dropout layers, collectively form the classification head of the model, enabling it to make predictions based on the extracted features from the base model.

3.3.3 Our distinct CNN variants

In this section, we explore different configurations of CNNs for AD and classification. Each configuration varies based on the use or non-use of pre-trained weights from ImageNet, as well as the application or non-application of the SMOTE technique to balance classes in our Alzheimer's dataset. We have chosen to explore four model variants for our project, each with different configurations. Our main objective is to understand the performance differences between models with and without the use of the SMOTE technique to address class imbalances, as well as the effects of using or not using pre-trained weights from the ImageNet dataset, and finally, the advantages of training or freezing the layers of the model.

1. **Models with pretrained weights, non-trainable layers and SMOTE**

   In this configuration, we use pre-trained weights from the ImageNet dataset to initialize our CNN model. The layers of the model are frozen to preserve the general features learned by the ImageNet layers while adapting the model to our specific data. Subsequently, the SMOTE technique is applied to balance the classes in our Alzheimer's dataset, addressing class imbalances and enhancing the model's generalization capability.

2. **Models with pretrained weights, trainable Layers and SMOTE**

   In this configuration, we use pre-trained weights from the ImageNet dataset to initialize our CNN model. Unlike the previous configuration, the layers of the model are trainable, allowing them to be fine-tuned during model training. By allowing the layers to be trainable, we aim to finely tune the model to the characteristics of our Alzheimer's dataset. Although we still use ImageNet for initialization, training the layers enables us to better capture the specific nuances of Alzheimer's in our data. The application of SMOTE remains important to ensure proper class balance.

3. **Models with random weight, trainable layers and SMOTE**

   In this setup, we do not use pre-trained weights from the ImageNet dataset to initialize our CNN model. The layers are trainable, allowing them to be trained from scratch on our Alzheimer's dataset, focusing entirely on learning Alzheimer's-specific features. This approach gives us complete control over the model's training, tailoring it precisely to our dataset. The application of SMOTE remains crucial for addressing class imbalances and improving the model's generalization capability

4. **Models with pretrained weight, non-trainable layers and without SMOTE**

In this configuration, we utilize pre-trained weights from the ImageNet dataset to initialize our CNN model. The decision to keep the layers non-trainable is based on the desire to maintain the general features learned by the ImageNet layers while preventing them from being influenced by our Alzheimer's dataset. By freezing these layers, we aim to capitalize on the knowledge transferred from ImageNet while potentially sacrificing adaptability to our specific data. The specific architecture includes convolutional layers followed by pooling layers, and possibly additional fully connected layers depending on the chosen CNN architecture. However, the absence of the SMOTE technique may affect the model's ability to generalize to test data, especially in the presence of class imbalances.

3.4 Conclusion

In this chapter, we proposed a comprehensive framework for classifying AD from brain MRI images using CNNs. Our framework integrates several advanced methodologies to address challenges in AD detection.

Firstly, for data preprocessing, we employed various techniques to prepare the dataset, including normalization and SMOTE to handle class imbalances. This ensures that the data fed into the CNNs is well-balanced and normalized, which is crucial for achieving reliable results.

Secondly, in terms of model architecture, we explored multiple CNN architectures to determine the most effective design for AD classification. This exploration included the use of pre-trained models such as VGG-16, MobileNetV2, and Xception. These models were fine-tuned to improve performance specifically for the AD classification task, leveraging their powerful feature extraction capabilities developed from training on large datasets.

Lastly, we tested different configurations of CNNs to understand the impact of transfer learning and resampling techniques on model performance. These configurations included PTW-TL-SMOTE (Pre-Trained Weights with Trainable Layers and SMOTE), PTW-NTL-SMOTE (Pre-Trained Weights with Non-Trainable Layers and SMOTE), and other combinations without SMOTE. By experimenting with these variants, we aimed to identify the optimal approach for enhancing the accuracy and robustness of AD detection.

Our proposed solutions form the basis for further experimentation and optimization in the subsequent chapters. This framework aims to advance the development of reliable diagnostic tools for Alzheimer's Disease, contributing to earlier and more accurate detection, which is crucial for patient care and treatment planning.

Chapter 4

Experiments, reults, and discussion

4.1 Introduction

In this chapter, we delve into the analysis of different variants of AD classification models, focusing on specific configurations of pre-training model, pre-processing data, and trainable layers. Our goal is to assess the performances of these variants and identify the one that offers the best performance in terms of accuracy, area under the ROC curve (AUC), and F1 score. We will also explore the impact of these different approaches on the models' ability to effectively detect AD from medical images.

4.2 Experiments configurations

Our experimental setup involved the development, training, and validation of our proposed approach on the Kaggle platform using Keras with TensorFlow, with computational resources provided by the P100 GPU. Each of the five models (VGG16, Inception, MobileNetV2, MobileNet, Xception), alongside the 4 code variants described in Chapter 3, was employed to construct diverse CNN architectures aimed at accurately classifying our dataset. To ensure comprehensive evaluation, each CNN architecture derived from the 4 code variants underwent training for a standardized number of epochs, specifically 100, utilizing the RMSprop optimizer.

4.3 Experimental material and platforms

4.3.1 Kaggle

Kaggle stands as a leading platform in the realm of data science and machine learning, granting access to potent computing resources like GPUs and TPUs. Alongside furnishing a varied collection of datasets for practice and competition, Kaggle draws a community of researchers and data scientists worldwide. This platform enables users to collaborate, learn, and compete by delving into datasets, constructing models, and sharing insights. With its GPU and TPU support, Kaggle facilitates accelerated computation, empowering users to efficiently tackle large-scale machine learning tasks.

4.3.2 Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation, allowing researchers and developers to quickly prototype and deploy deep learning models. Keras provides a user-friendly interface for building various types of neural networks, including convolutional networks, recurrent networks, and combinations of both. Its simplicity and flexibility make it an ideal choice for beginners and experts alike, facilitating the development of complex deep learning architectures with minimal code. With Keras, users can seamlessly transition from experimenting with small-scale models to training large-scale neural networks, thanks to its efficient backend implementations and integration with powerful computational resources like GPUs.

4.3.3 TensorFlow

TensorFlow is a powerful open-source machine learning framework developed by Google Brain. It facilitates the creation and deployment of machine learning models across various platforms and devices. Its computational graph abstraction enables efficient execution on CPUs and GPUs, and it supports distributed computing. TensorFlow offers high-level APIs like Keras for easy model building, as well as lower-level APIs for more control. It supports a wide range of tasks including deep learning, reinforcement learning, and probabilistic modeling. With comprehensive tools for data preprocessing, model evaluation, and visualization, TensorFlow remains a top choice for researchers, developers, and enterprises in the field of machine learning.

4.3.4 Pandas

Pandas is a Python library widely used for data manipulation and analysis. It offers intuitive data structures like DataFrames and Series, facilitating efficient handling of structured data. With functionalities for loading data from various sources and performing tasks such as cleaning, exploration, transformation, and visualization, Pandas is essential for data preprocessing. It seamlessly integrates with other Python libraries, making it a favorite among data scientists, analysts, and developers for working with tabular data.

4.3.5 Matplotlib

Matplotlib is a versatile Python library for creating static, animated, and interactive visualizations. It offers a flexible interface for generating a wide range of plots and charts, with fine-grained control over customization. Matplotlib is widely used across scientific research, engineering, finance, and data science for its ability to produce publication-quality figures. It integrates seamlessly with Jupyter notebooks and supports various output formats. Overall, Matplotlib is an essential tool for data visualization and storytelling in Python.

4.4 Performance metrics

To evaluate the performance of our overall approach, we will use several metrics such as precision, recall, F1-score, and AUC (Area Under the ROC Curve). To calculate these metrics, you need to compare the model's predictions with the ground truth labels. Here's a step-by-step guide on how to calculate these metrics:

Accuracy: It's the ratio of the correctly labeled subjects to the whole pool of subjects. Accuracy is the most intuitive one.

$$\text{Acc} = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision: Precision is the ratio of the correctly positive labeled by our program to all positive labeled.

$$\text{Pre} = \frac{TP}{TP + FP}$$

Recall (aka sensitivity): Recall is the ratio of the correctly positive labeled by our program to all who are infected in reality.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score: It is the harmonic mean (average) of the precision and recall.

$$\text{F1-Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

True Positive (TP): The prediction and the actual value are positive. Example: A sick person predicted sick.

True Negative (TN): The prediction and the actual value are negative. Example: A healthy person predicted healthy.

False Positive (FP): The prediction is positive while the actual value is negative.

False Negative (FN): The prediction is negative while the actual value is positive.

Total instances (N): The total number of instances.

4.5 Results and discussion

To facilitate the comparison between the different approaches, we have noted the following:

PTW-NTL-NS: Models with Pre-trained Weights, Non-Trainable Layers, and without SMOTE

FS-TL-SMOTE: Models from Scratch, Trainable Layers, and with SMOTE

PTW-TL-SMOTE: Models with Pre-trained Weights and Trainable Layers with SMOTE

PTW-NTL-SMOTE: Models with Pre-trained Weights and Non-Trainable Layers with SMOTE

4.6 Our variants

4.6.1 PTW-TL-SMOTE: Pre-Trained Weights-Trainable Layer-With SMOTE

Model	loss	Accuracy	AUC	F1SCORE
InceptionV3	0.4080	0.8234	0.9693	0.8231
MobileNet	0.4893	0.8305	0.9707	0.8326
MobileNetV2	0.4603	0.8035	0.9617	0.8042
VGG16	0.5662	0.7449	0.9401	0.7435
Xception	0.4249	0.8176	0.9673	0.8177

Table 4.1: FS-TL-SMOTE Result

The results obtained in table 4.1with models trained from scratch and using SMOTE to address class imbalances show significant variability. The MobileNet model stands out as the best with an accuracy of 83.05%, an AUC of 97.07%, and an F1-score of 83.26%, while maintaining a moderate loss of 0.4893. Conversely, the VGG16 model shows the worst performance with the lowest accuracy (74.49%), an AUC of 94.01%, and the lowest F1-score (74.3%), as well as a high loss of 0.5662. The InceptionV3, MobileNetV2, and Xception models show intermediate performances, with respective accuracies of 82.34%, 80.35%, and 81.76%, and AUCs of 96.93%, 96.17%, and 96.73%. The F1-scores and losses for these models follow similar trends, indicating that although all models benefit from the application of SMOTE, selecting the appropriate model architecture is crucial for optimizing performance.

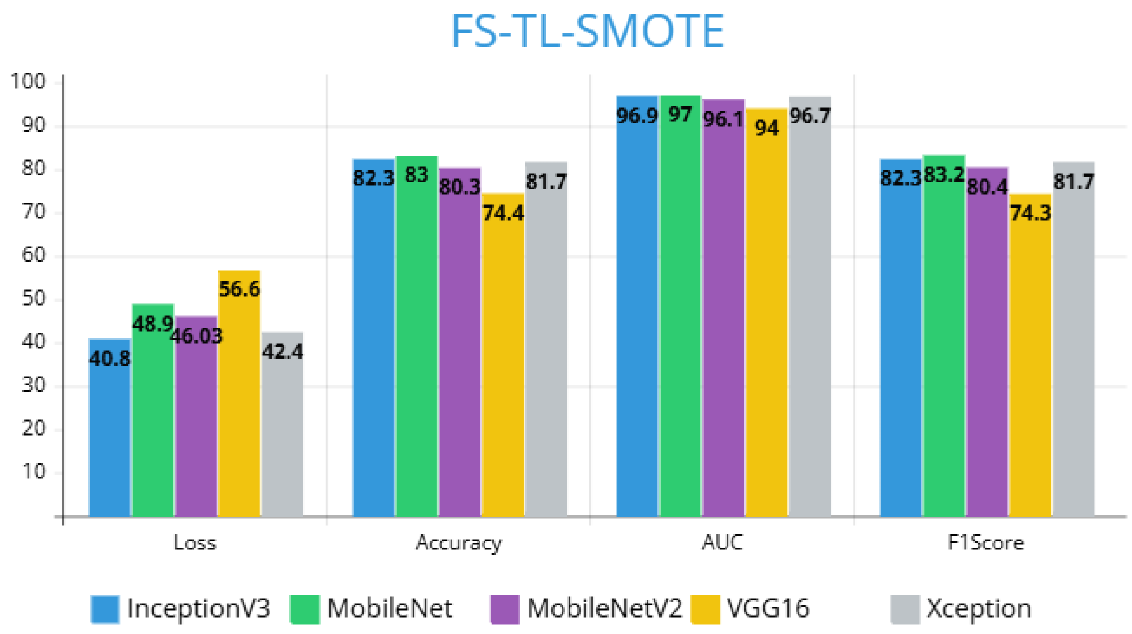

Figure 4.1: Bar graph FS-TL-SMOTE

The combined diagram (Figure 4.1) illustrates the performance of each model for the FS-TL-SMOTE variant. Each sub-figure generally shows that MobileNet stands out with its high scores in accuracy, AUC, and F1-score, indicating superior effectiveness in distinguishing classes. MobileNetV2 and InceptionV3 also show good performance, with high scores in all metrics, suggesting a good balance between precision and recall and notable training efficiency. Xception shows results similar to those of InceptionV3, with slightly lower but still high scores, reflecting robust performance. In contrast, VGG16 shows the least favorable results, with lower scores in accuracy, AUC, and F1-score, and a higher loss, indicating difficulties in balancing precision and recall and a greater tendency towards errors during training. These results highlight the importance of choosing an appropriate model architecture to optimize performance, and the effectiveness of SMOTE in addressing class imbalances.

Training models from scratch without pre-trained weights allows them to learn the specific features of Alzheimer's disease directly from the dataset. This approach appears particularly effective for the MobileNet model, which significantly outperforms the others. The application of SMOTE in this configuration helps address class imbalances, which is crucial for improving model generalization. All models benefit from SMOTE, as evidenced by their relatively high AUC scores, demonstrating effective class distinction. The choice of model architecture has a substantial impact on performance. The InceptionV3 and

MobileNet architectures seem particularly well-suited for this task, likely due to their efficient feature extraction and adaptability when trained from scratch.

The performance curves of various models trained with the FS-TL-SMOTE approach in terms of loss, AUC, and accuracy is illustrated in Figures 4.2 to 4.6. Specifically, Figure 4.2 shows the results for the MobileNet model, Figure 4.3 for MobileNetV2, Figure 4.4 for InceptionV3, Figure 4.5 for Xception, and Figure 4.6 for VGG16. These charts provide insights into each model's ability to generalize to new data and handle class imbalances using SMOTE.

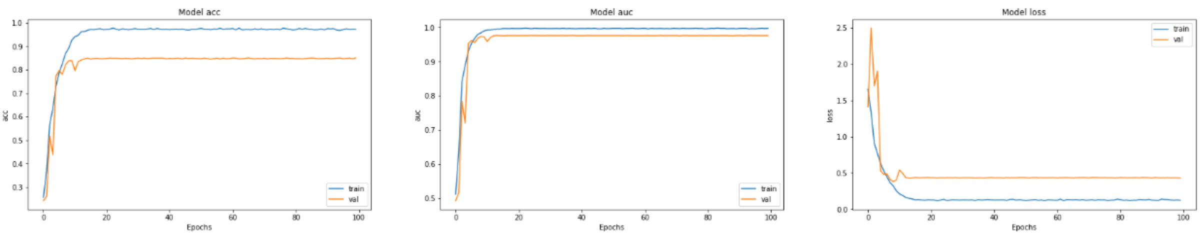

Figure 4.2: curves MobileNet FS-TL-SMOTE

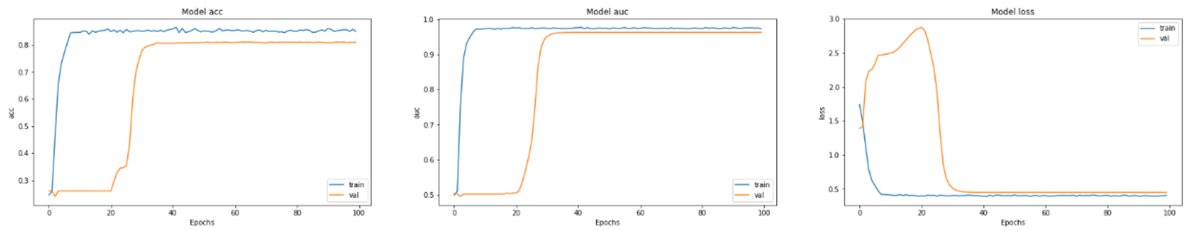

Figure 4.3: curves MobileNetV2 FS-TL-SMOTE

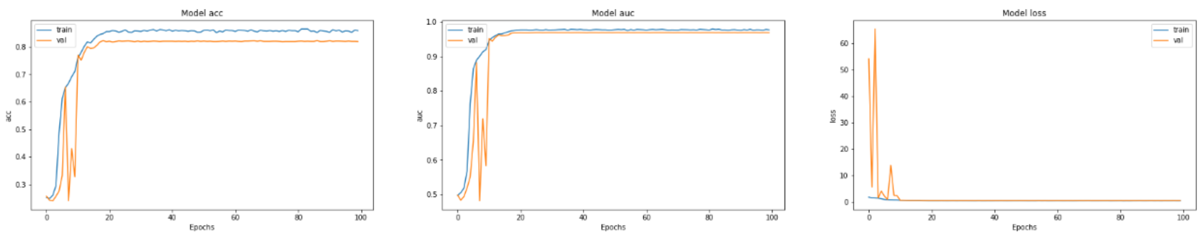

Figure 4.4: curves InceptionV3 FS-TL-SMOTE

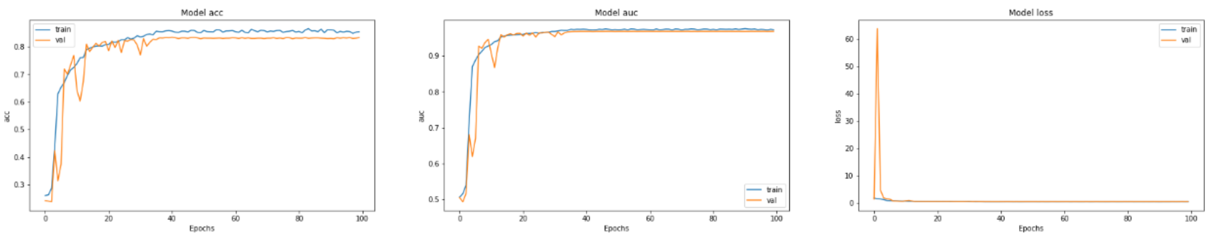

Figure 4.5: curves Xception FS-TL-SMOTE

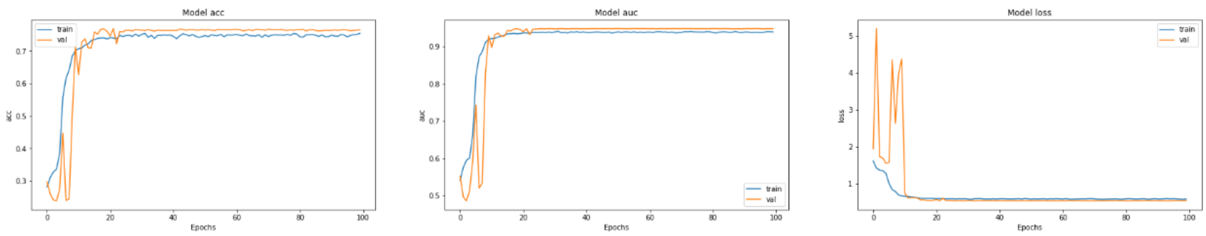

Figure 4.6: curves VGG16 FS-TL-SMOTE

4.6.2 PTW-TL-SMOTE: Pre-Trained Weights-Trainable Layer-With SMOTE

Model	**Loss**	**Accuracy**	**AUC**	**F1Score**
Inceptionv3	0.3887	0.8348	0.9727	0.8351
MobileNet	0.1577	0.9621	0.9912	0.9622
MobileNetV2	0.2202	0.9586	0.9849	0.9588
VGG16	0.4900	0.7801	0.9553	0.7787
Xception	0.1575	0.9574	0.9928	0.9576

Table 4.2: PTW-TL-SMOTE Result

The table 4.2 presents the values of each metric for each of the five models. Examining the loss column, we observe that MobileNet and xception show the lowest values (0.1577) and (0.1575), indicating the model's superior ability to minimize prediction error. Conversely, VGG16 has the highest loss (0.4900), suggesting difficulties in converging towards an optimal solution.

Regarding accuracy, MobileNet again stands out with the highest value (96.21%), demonstrating excellent capability in correctly predicting image classes. VGG16, once more, ranks last with the lowest accuracy (78.01%), highlighting its limitations in precise class distinction.

The analysis of the AUC confirms MobileNet's superiority, with a value of 99.12% indicating exceptional ability to differentiate positive from negative classes. VGG16, with an AUC of 95.53%, shows inferior performance in this area.

Finally, the F1-score, which combines precision and recall, confirms MobileNet's dominance with a value of 96.22%. VGG16, with an F1-score of 77.87%, exhibits the least satisfactory results in terms of balance between precision and recall.

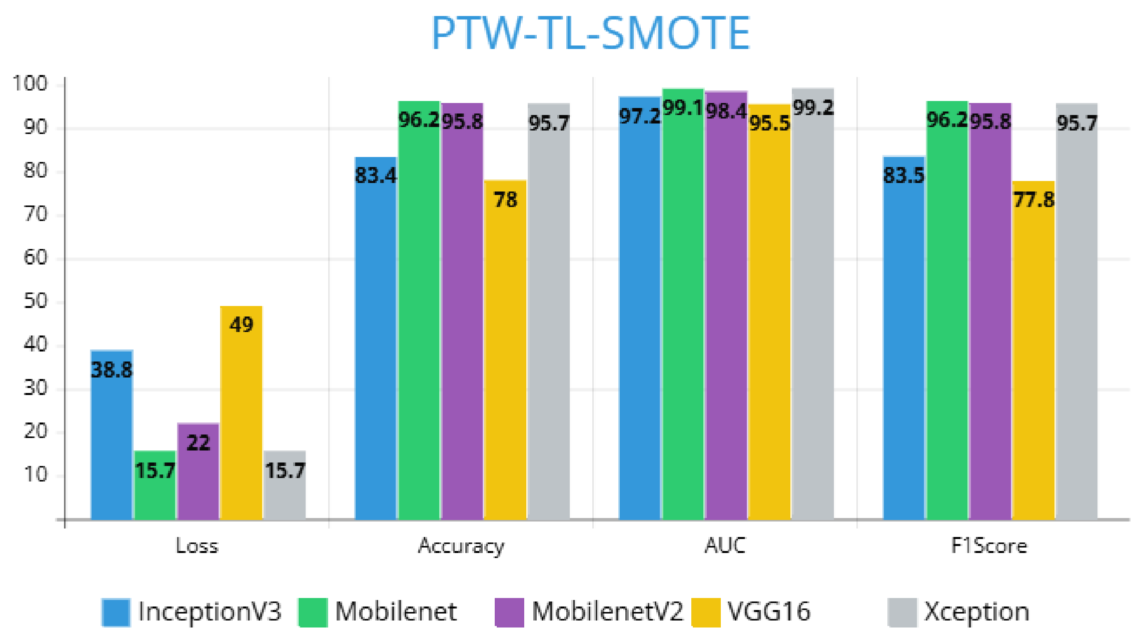

Figure 4.7: Bar graph PTW-TL-SMOTE

The figure 4.7 presents the models' performances in terms of accuracy, AUC, and F1-score as curves. The MobileNet curve clearly stands out from the others by its consistently high position in all metrics, confirming its superior overall performance. The curves of the other models (InceptionV3, MobileNetV2, and Xception) exhibit similar profiles, indicating intermediate performance. As shown by the table, VGG16 is characterized by curves located at the bottom of the graph, highlighting its limitations in terms of accuracy, AUC, and F1-score.

Figures 4.8 to 4.12 show a plot of accuracy ,loss and AUC models using the PTW-TL-SMOTE approach. Specifically, Figure 4.8 shows the performance of InceptionV3, Figure 4.9 of MobileNet, Figure 4.10 of MobileNetV2, Figure 4.11 of VGG16, and Figure 4.12 of Xception. These graphs illustrate loss, AUC, and accuracy, offering a clear view of each model's ability to generalize to new data while addressing class imbalances with SMOTE.

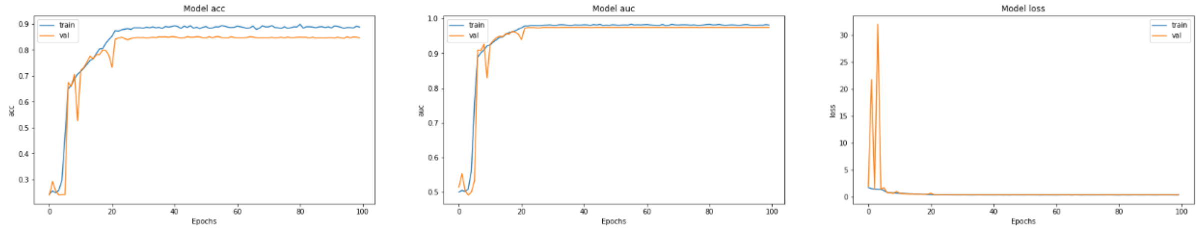

Figure 4.8: Curves InceptionV3 PTW-TL-SMOTE

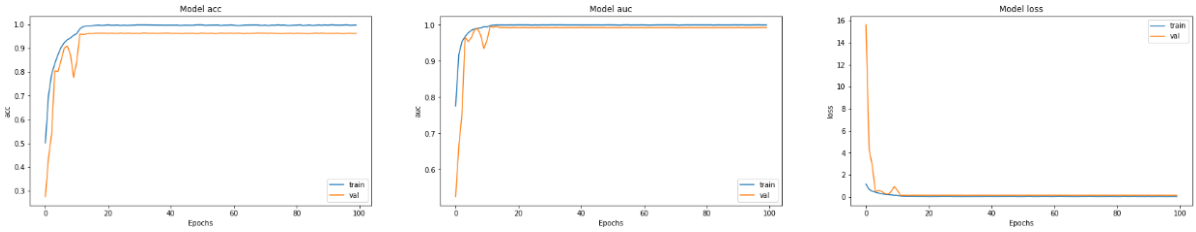

Figure 4.9: Curves MobileNet PTW-TL-SMOTE

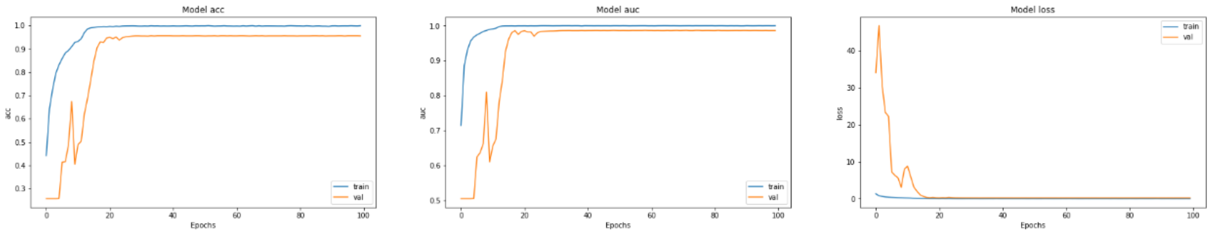

Figure 4.10: Curves MobileNetV2 PTW-TL-SMOTE

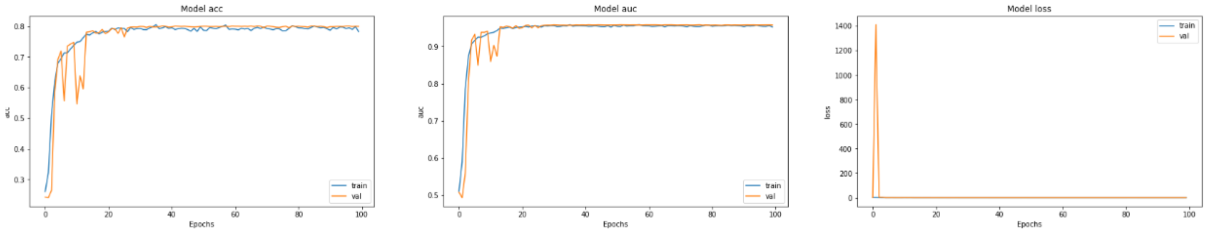

Figure 4.11: Curves VGG16 PTW-TL-SMOTE

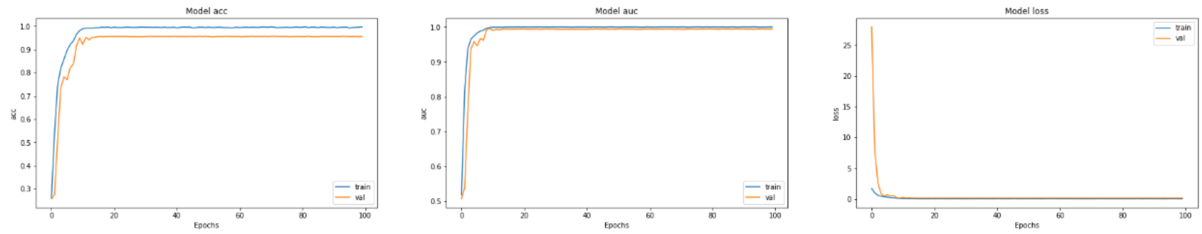

Figure 4.12: Curves Xception PTW-TL-SMOTE

4.6.3 PTW-NTL-SMOTE:Pre-Trained Weights-Non-Trainable Layer-With SMOTE

Model	**Loss**	**Accuracy**	**AUC**	**F1Score**
InceptionV3	0.2999	0.8891	0.9843	0.8887
MobileNet	0.1793	0.9383	0.9939	0.9384
MobileNetV2	0.2711	0.8977	0.9865	0.8978
VGG16	0.2364	0.9164	0.9907	0.9160
Xception	0.3366	0.8770	0.9810	0.8762

Table 4.3: PTW-NTL-SMOTE Result

The results obtained with models using pre-trained weights and non-trainable layers combined with SMOTE show competitive performance. MobileNet stands out with the best performance, achieving an accuracy of 93.83%, an AUC of 99.39%, and an F1-score of 93.84%, while maintaining a very low loss of 0.1793. VGG16 closely follows with an accuracy of 91.64%, an AUC of 99.07%, and an F1-score of 91.60%, while maintaining a moderate loss of 0.2364. The InceptionV3, MobileNetV2, and Xception models show intermediate performances, with respective accuracies of 88.91%, 89.77%, and 87.70%, and AUCs of 98.43%, 98.65%, and 98.10%. The F1-scores and losses for these models follow similar trends, indicating that the use of pre-trained weights and non-trainable layers combined with SMOTE can be very effective in improving the performance of Alzheimer's disease detection models.

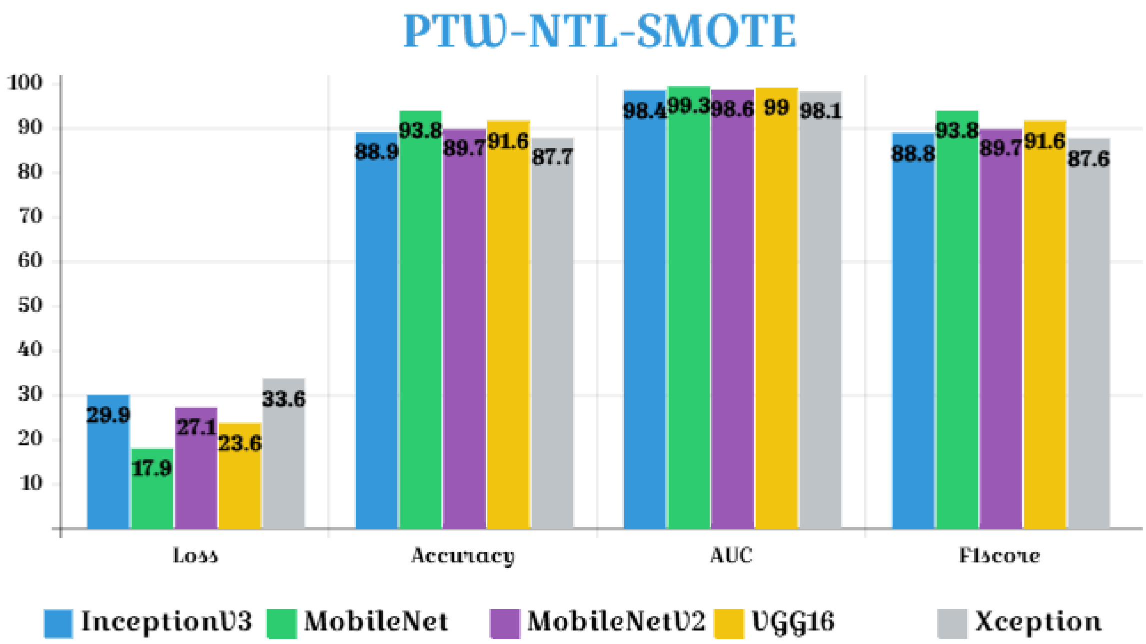

Figure 4.13: Bar graph PTW-NTL-SMOTE

The combined diagram (Figure 4.13) illustrates the performance of each model for the PTW-NTL-SMOTE variant. Each sub-figure shows the performance curves of the individual models, allowing for a clear visual comparison. MobileNet stands out with higher curves in accuracy and AUC, as well as high F1-scores, confirming its effectiveness. VGG16, on the other hand, shows similar curves, reflecting its competitive performance. The other models fall between these extremes, each showing specific strengths and weaknesses.

Models using pre-trained weights and non-trainable layers with SMOTE demonstrate competitive performance, indicating the effectiveness of leveraging pre-trained knowledge while freezing certain layers to avoid overfitting. MobileNet presents the highest performance across all metrics, closely followed by VGG16. This suggests that architectures capable of effectively leveraging pre-trained weights, even with non-trainable layers, can achieve strong performance in Alzheimer's disease detection tasks.

Figures 4.14 to 4.18 display the results of models using the PTW-NTL-SMOTE approach. Specifically, Figure 4.14 shows the performance of VGG16, Figure 4.15 of InceptionV3, Figure 4.16 of MobileNet, Figure 4.17 of MobileNetV2, and Figure 4.18 of

Xception. These graphs depict loss, AUC, and accuracy, providing a clear visualization of each model's capacity to generalize to new data while managing class imbalances with SMOTE.

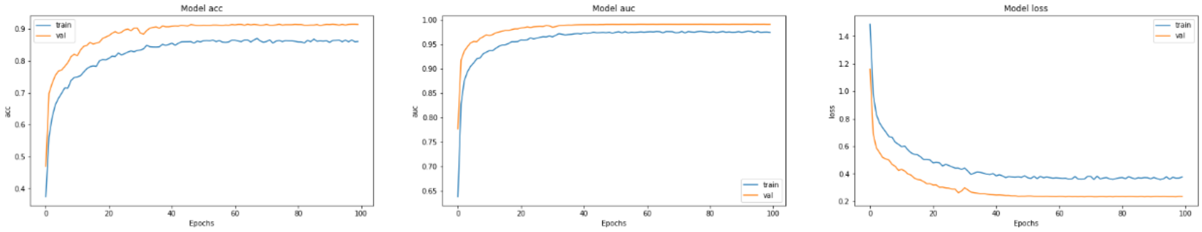

Figure 4.14: Curves VGG16 PTW-NTL-SMOTE

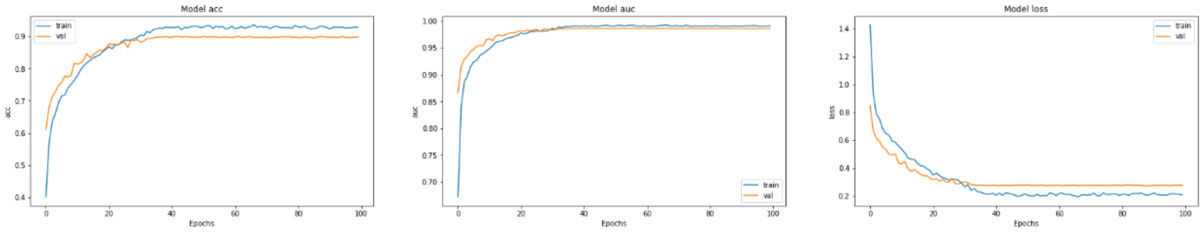

Figure 4.15: Curves InceptionV3 PTW-NTL-SMOTE

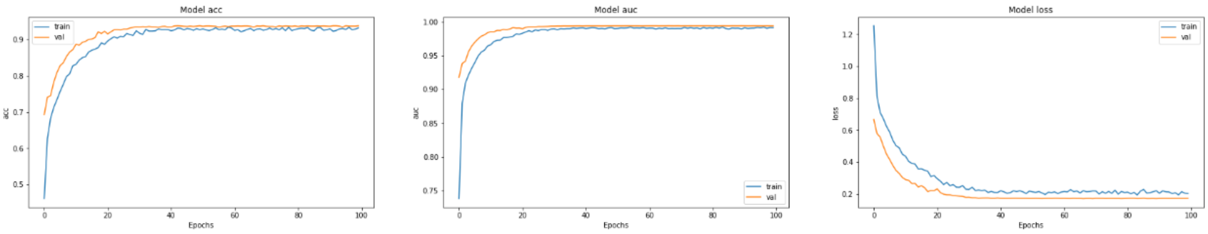

Figure 4.16: Curves MobileNet PTW-NTL-SMOTE

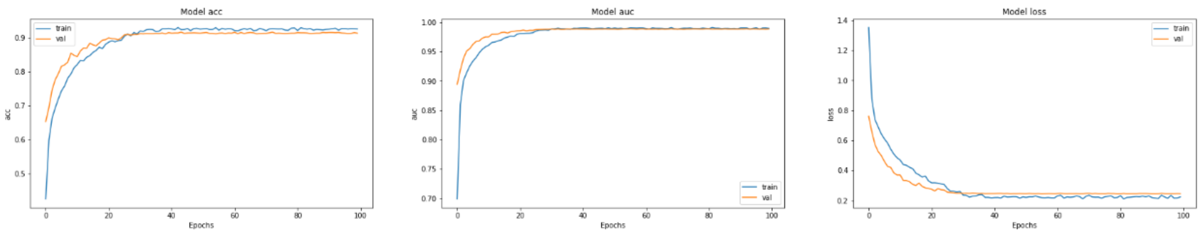

Figure 4.17: Curves MobileNetV2 PTW-NTL-SMOTE

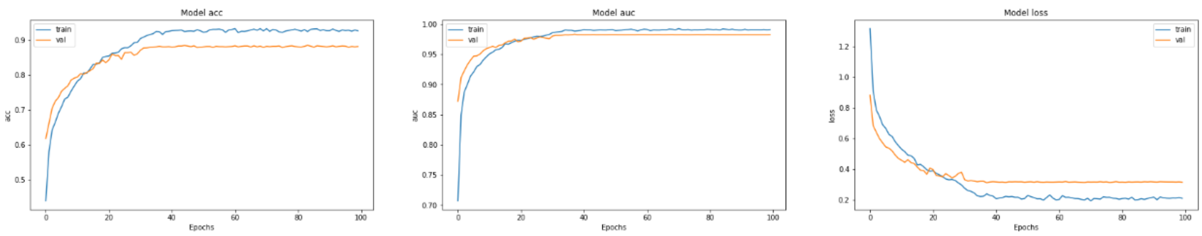

Figure 4.18: Curves Xception PTW-NTL-SMOTE

4.6.4 PTW-NTL-SMOTE:Pre-Trained Weights-Non-Trainable Layer-Without SMOTE

Model	loss	Accurency	AUC	F1SCORE
InceptionV3	0.6477	0.7445	0.9317	0.5800
MobileNet	0.4079	0.8609	0.9699	0.8420
MobileNetV2	0.6169	0.7805	0.9404	0.7666
VGG16	0.4361	0.8328	0.9666	0.8541
Xception	0.7454	0.6984	0.9077	0.4834

Table 4.4: PTW-NTL-NS Result

The results obtained with models using pre-trained weights and non-trainable layers without the application of SMOTE show significant variability. MobileNet stands out with the best performance, displaying an accuracy of 86.09%, an AUC of 96.99%, and an F1-score of 84.20%, while maintaining a relatively low loss of 0.4079. VGG16 closely follows with an accuracy of 83.28%, an AUC of 96.66%, and an F1-score of 85.41%, although its loss is slightly higher at 0.4361. The InceptionV3, MobileNetV2, and Xception models show less impressive performances, with respective accuracies of 74.45%, 78.05%, and 69.84%, and AUCs of 93.17%, 94.04%, and 90.77%. The F1-scores and losses for these models follow similar trends, highlighting the importance of SMOTE for improving model performance by balancing data classes. These models demonstrate competitive performance, although less optimal compared to using SMOTE. MobileNet presents the highest performance across all metrics, closely followed by VGG16. This suggests that architectures capable of effectively leveraging pre-trained weights, even with non-trainable layers, can achieve solid performance in Alzheimer's disease detection tasks, although the absence of SMOTE may reduce overall effectiveness due to untreated class imbalances.

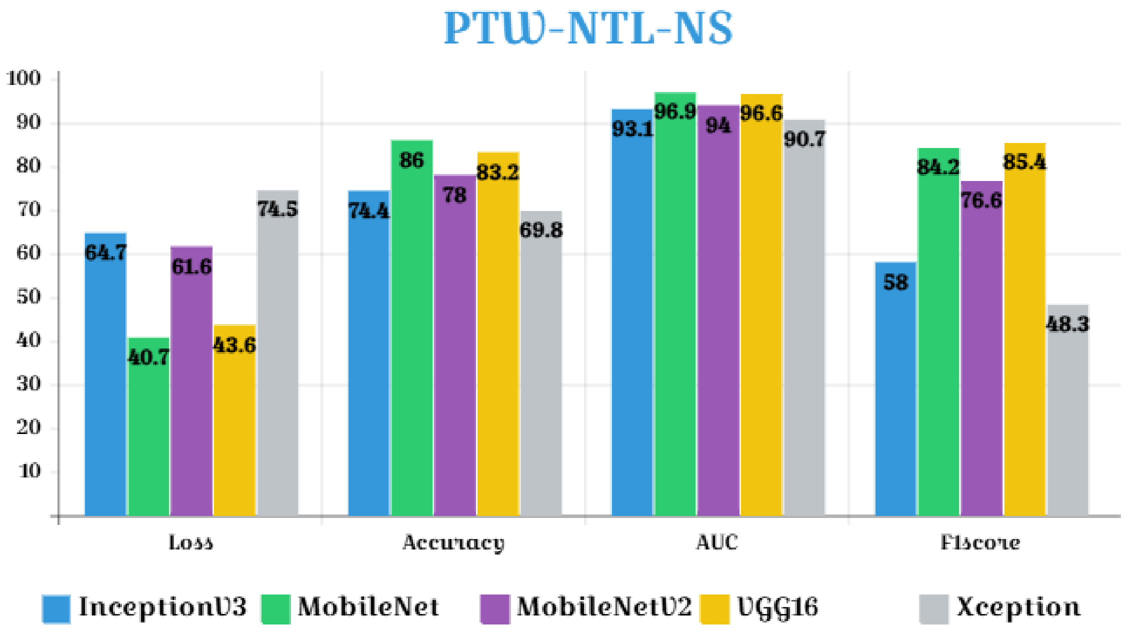

Figure 4.19: Bar graph PTW-NTL-NS

The combined diagram (Figure 4.19) illustrates the performance of each model for the PTW-NTL-NS variant. Each sub-figure shows the performance curves of the individual models, allowing for a clear visual comparison. MobileNet stands out with higher curves in accuracy and AUC, as well as high F1-scores, confirming its effectiveness even without the application of SMOTE. VGG16 shows similar curves, reflecting its competitive performance. The other models show lower curves, particularly Xception, indicating that the absence of SMOTE negatively impacts the performance of these models.

Figures 4.20 to 4.24 show a plot of accuracy ,loss and AUC of models using the PTW-NTL-NS approach. Specifically, Figure 4.20 illustrates the performance of VGG16, Figure 4.21 of InceptionV3, Figure 4.22 of MobileNet, Figure 4.23 of MobileNetV2, and Figure 4.24 of Xception. These charts highlight loss, AUC, and accuracy, offering a detailed view of each model's ability to generalize to new data while handling class imbalances without using SMOTE.

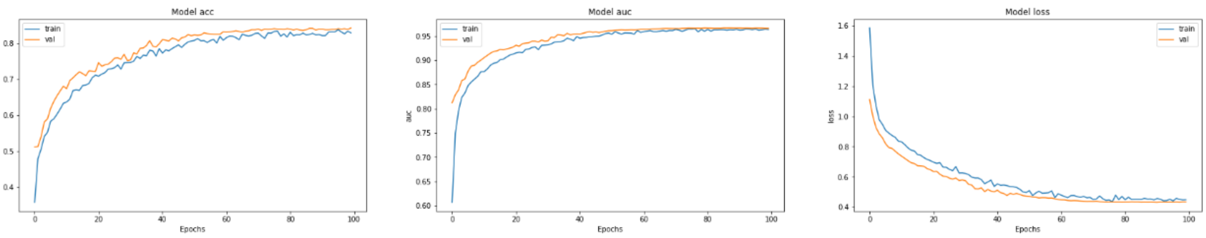

Figure 4.20: Curves VGG16 PTW-NTL-NS

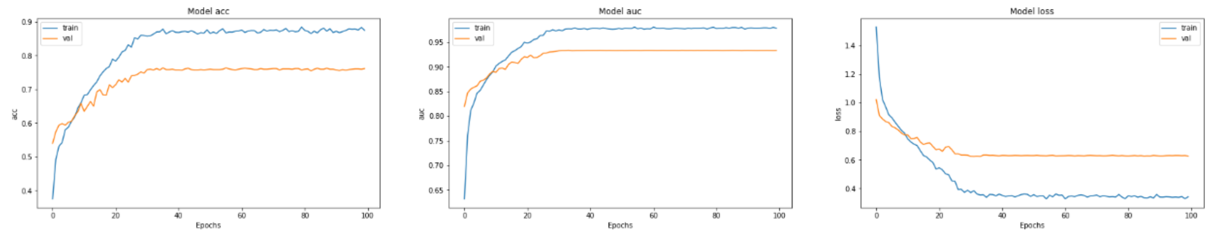

Figure 4.21: Curves InceptionV3 PTW-NTL-NS

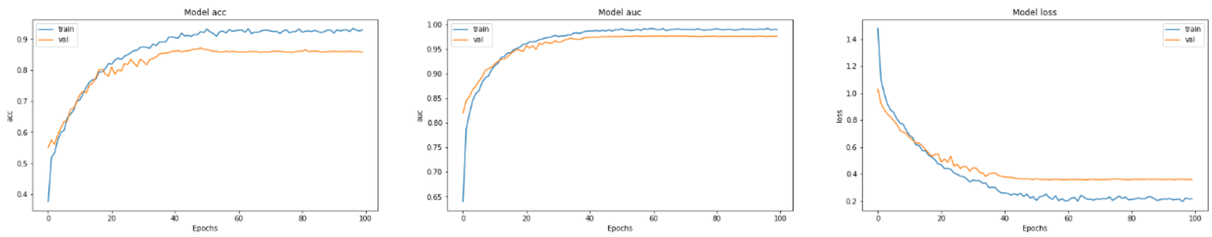

Figure 4.22: Curves MobileNet PTW-NTL-NS

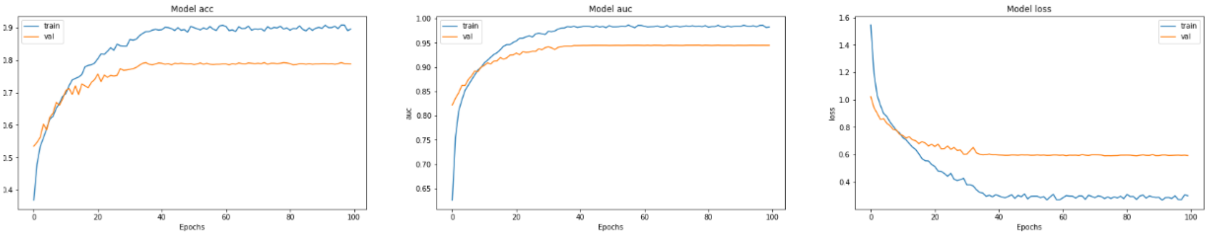

Figure 4.23: Curves MobileNetV2 PTW-NTL-NS

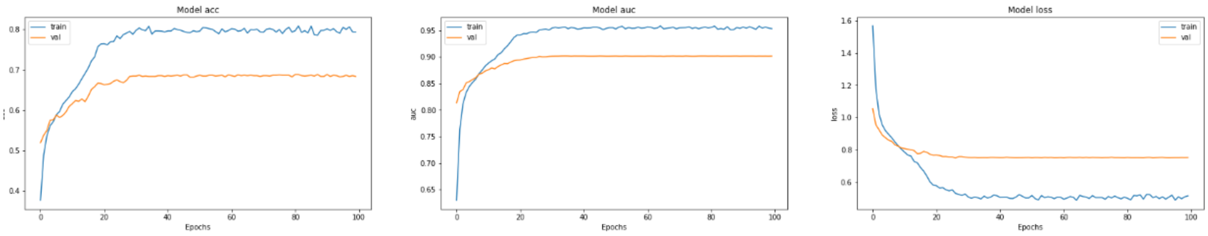

Figure 4.24: Curves Xception PTW-NTL-NS

4.7 Comparison of Variants

Variant	**Model**	**Accuracy (%)**	**AUC**	**F1 Score**
FS-TL-SMOTE	MobileNet	83.05	97.07	83.26
PTW-TL-SMOTE	MobileNet	96.21	99.12	96.22
PTW-NTL-SMOTE	MobileNet	93.83	99.39	93.84
PTW-NTL-NS	MobileNet	86.09	96.99	84.20

Table 4.5: Best performing models for each variant

Table 4.5 presents the best performances achieved for each variant using the MobileNet model. The performance metrics include accuracy, AUC (Area Under the ROC Curve), and F1 score, which are key indicators of classification model performance. Notably, MobileNet provided the best results across all variants.

Based on the table, the **PTW-TL-SMOTE** variant stands out as the top performer, surpassing other variants across all performance metrics. The MobileNet model under PTW-TL-SMOTE achieves an accuracy of 96.21%, an AUC of 99.12%, and an F1 score of 96.22%. These results indicate that this model has an exceptional ability to correctly classify Alzheimer's Disease (AD) and non-AD cases, maintaining a good balance between precision and recall, as evidenced by the F1 score.

PTW-NTL-SMOTE ranks second, with strong but slightly lower metrics compared to PTW-TL-SMOTE. MobileNet achieves an accuracy of 93.83%, an AUC of 99.39%, and an F1 score of 93.84%. Although this variant performs well, it does not reach the levels of PTW-TL-SMOTE, mainly due to the limitation of non-trainable layers reducing model adaptability.

FS-TL-SMOTE, while performing well with an accuracy of 83.05%, an AUC of 97.07%, and an F1 score of 83.26%, lags behind the variants that utilize pre-trained weights. This can be attributed to the fact that training models from scratch requires much more data and computational resources to achieve optimal performance, which is not always practical or feasible.

PTW-NTL-NS exhibits the lowest performance across all metrics. MobileNet under this variant achieves an accuracy of 86.09%, an AUC of 96.99%, and an F1 score of 84.20%. The absence of SMOTE in this variant severely limits its ability to handle class imbalance, resulting in poorer performance.

4.7.1 Reasons for the Superiority of the PTW-TL-SMOTE Variant

1. **Utilization of Pre-trained Weights**: Models under PTW-TL-SMOTE benefit from general features learned from large datasets, providing a solid starting point for training on specific task data.

2. **Adaptability through Trainable Layers**: The ability to fine-tune layers allows models to adapt specifically to the AD dataset, enhancing overall performance.

3. **SMOTE for Class Balance**: The application of SMOTE helps address class imbalance issues, ensuring the model is not biased towards the majority class, thus improving precision and recall.

4. **Combination of Prior Knowledge and Specific Adaptation**: This variant effectively balances the use of prior knowledge (via pre-trained weights) and specific adaptation (via trainable layers), which is crucial for complex tasks like AD classification.

In conclusion, the PTW-TL-SMOTE variant emerges as the most robust and effective approach for AD classification, offering an optimal balance between leveraging prior knowledge and adapting to specific data characteristics.

4.8 Comparison related work

Table 4.6 compares the best-performing model from our study with models from related studies using the same dataset as ours . Our PTW-TL-SMOTE MobileNet model achieves the highest accuracy among the compared models.

Reference	**Model**	**Accuracy**
[60]	VGG16	77.04%
	VGG19	77.66%
[58]	EfficientNetB0	89.7%
[61]	From Scratch with Smote	95.23%
	From Scratch without Smote	85%
[18]	DenseNet-169	83.82%
	ResNet-50	81.92%
[57]	From Scratch	94.39%
FS-TL-SMOTE	MobileNet	83.05%
PTW-TL-SMOTE	MobileNet	96.21%
PTW-NTL-SMOTE	MobileNet	93.83%
PTW-NTL-NS	MobileNet	86.09%

Table 4.6: Comparison related work

Our proposed models, including VGG16, MobileNet, MobileNetV2, InceptionV3, and Xception, demonstrated superior accuracy in classifying AD compared to previously reported methods. The detailed comparison presented in Table 4.5 highlights the performance of our models against several prior works, underscoring the effectiveness of our approaches. For instance, Ajagbe et al. [61] reported accuracies of 77.04% and 77.66% using VGG16 and VGG19, respectively. In contrast, our FS-TL-SMOTE with MobileNet achieved an accuracy of 83.05%, demonstrating a notable improvement over these baseline models.

Altwijri et al. [59] achieved an accuracy of 89.7% with EfficientNetB0, which is surpassed by our PTW-TL-SMOTE variant of MobileNet, reaching 96.21%. This is the highest accuracy reported in our study and significantly exceeds the previously reported highest accuracy of 95.23% by Murugan et al. [62] using a model trained from scratch with SMOTE. Additionally, Murugan et al. [62] reported an accuracy of 85% for a model trained without SMOTE, while our PTW-NTL-NS variant with MobileNet achieved an accuracy of 86.09%, still higher than their non-SMOTE model. Furthermore, Ibrahim et al. [18] reported an accuracy of 94.39% using a model trained from scratch, which is slightly lower than our PTW-NTL-SMOTE variant with MobileNet, which achieved 93.83%. The comparison clearly indicates that our models, particularly when leveraging pre-trained weights and SMOTE, consistently outperform previous methodologies. These results emphasize the robustness and effectiveness of our proposed models in the classification of AD, offering significant improvements over existing approaches.

4.9 Conclusion

In conclusion, this chapter has allowed us to compare and evaluate various variants of AD classification models. We found that the PTW-TL-SMOTE variant, which utilizes pre-trained weights with trainable layers and applies SMOTE to balance classes, offers the best performance in terms of accuracy, AUC, and F1 score. These results provide valuable insights for the development of more effective computer-aided diagnostic systems in the early detection of AD. Our extensive evaluation of models such as VGG16, MobileNet, MobileNetV2, InceptionV3, and Xception demonstrated their superior performance, with the PTW-TL-SMOTE variant achieving an accuracy of 96.21%. The careful preprocessing of MRI images and the integration of advanced sampling techniques were crucial in achieving these results. This study underscores the potential of deep learning in advancing medical diagnostics, emphasizing the importance of innovative approaches in medical imaging and diagnostics. The insights gained from this research can significantly contribute to the development of more accurate and reliable diagnostic tools, ultimately improving patient outcomes. By leveraging the power of deep learning, we can enhance the precision and efficiency of Alzheimer's Disease detection, fostering advancements in healthcare technology and personalized medicine.

General conclusion

In conclusion, this thesis explored the use of pre-trained Convolutional Neural Networks (CNNs) for the detection of Alzheimer's Disease (AD) from MRI images. The objective was to enhance diagnostic accuracy and enable early intervention through advanced deep learning techniques.

The results obtained from the experiments demonstrate the effectiveness of the proposed approaches. Pre-trained models combined with data rebalancing techniques such as SMOTE achieved remarkable accuracy levels. For instance, the MobileNet model with SMOTE achieved an accuracy of 96.21%. These results highlight the potential of leveraging pre-trained models to optimize CNN architectures and capitalize on the knowledge acquired to improve early AD detection.

Similarly, other models such as VGG16 and InceptionV3, when combined with SMOTE, also showed impressive performance, underscoring the efficacy of these models for medical image classification. The results demonstrate that this approach can be a valuable tool for automating the detection of Alzheimer's Disease in a clinical context.

The integration of pre-trained models and data rebalancing techniques further enhanced the performance of the generated models. With accuracies reaching up to 96.21%, the proposed approaches showcase significant potential to improve the efficiency and accuracy of Alzheimer's Disease diagnosis.

This thesis provides valuable insights into the use of pre-trained CNN architectures for the detection of Alzheimer's Disease. The results demonstrate the potential of data rebalancing techniques and transfer learning to significantly enhance the efficiency and accuracy of classification tasks in the medical field. Overall, the results underscore that approaches based on pre-trained models are capable of generating better CNN architectures for Alzheimer's Disease detection compared to models trained from scratch. It is important to note that while the achieved accuracies are impressive, further research and evaluation are necessary to assess their robustness and generalizability across diverse datasets and environmental conditions. Additionally, exploring new optimization techniques and architectural modifications may yield even higher accuracy in future studies.

Bibliography

[1] cip. Health problems caused by alzheimer's disease, July 25, 2013.

[2] Dr. A.K. Makkar. Alzheimer's dementia: Homeopathic treatment, Year of publication.

[3] BRENT LINDEQUE. Fasting: The new trend in wellness could do more harm than good, APRIL 20, 2016.

[4] Coding Compiler. Machine learning vs. deep learning.

[5] Smlease. Machine learning vs. deep learning: What is the difference between ml and dl?, Year of publication.

[6] Adnan Mohsin Abdulazeez. Recurrent neural networks, Jan 2021.

[7] Farzan Rahman and Neerav Patel. Deep learning in production with pytorch. *Patterns*, 1(1):100004, 2020.

[8] Vrogue. A structure of the convolutional neural network from the classification, Year of publication.

[9] NVIDIA. What's the difference between a cnn and an rnn?, Year of publication.

[10] OpenGenus IQ. Pooling layers, Year of publication.

[11] tecokids.monastir. Fully connected layer with dynamic input shape, Year of publication.

[12] Hussein G. Deeb. Understanding layer normalization, 2020.

[13] D. Mueller et al. Multiple-oriented and small object detection with convolutional neural networks for aerial image. *ResearchGate*, 2019.

[14] Abid Mehmood, Yonis Gulzar, Qazi Mudassar Ilyas, Abdoh Jabbari, Muneer Ahmad, and Sajid Iqbal. Sbxception: A shallower and broader xception architecture for efficient classification of skin lesions. *Cancers*, 15(14), 2023.

[15] FirstName1 Author1, FirstName2 Author2, et al. The architecture of mobilenet v1. *ResearchGate*, 2021.

[16] Anil Kumar Biswal, Bighnaraj Naik, and Pravas Ranjan Tripathy. Performance evaluation of deep cnn-based crack detection and localization techniques for concrete structures. *ResearchGate*, 2021.

[17] Guillermo Moya-Alvarado, Noga Gershoni-Emek, Eran Perlson, and Francisca C Bronfman. Neurodegeneration and alzheimer's disease (ad). what can proteomics tell us about the alzheimer's brain? *Molecular & Cellular Proteomics*, 15(2):409–425, 2016.

[18] Waleed Al Shehri. Alzheimer's disease diagnosis and classification using deep learning techniques. *PeerJ Computer Science*, 8:e1177, 2022.

[19] Mayo Clinic. Alzheimer's disease: Symptoms & causes, February 2024. Accessed on February 17, 2024.

[20] Alzheimer's Association. Stages of alzheimer's, 16 avril 2024.

[21] Alois Alzheimer, Jurgen Gaupp, Hans-Jochen Möller, and Robert Katzman. *Alzheimer's Disease and Other Dementias*. Springer Science+Business Media, 2011.

[22] National Institute on Aging. How alzheimer's disease is diagnosed, 2022. Accessed on February 17, 2024.

[23] Clifford R Jack Jr, David S Knopman, William J Jagust, Ronald C Petersen, Michael W Weiner, Paul S Aisen, Leslie M Shaw, Prashanthi Vemuri, Heather J Wiste, Stephen D Weigand, et al. Update on hypothetical model of alzheimer's disease biomarkers. *Lancet neurology*, 12(2):207, 2013.

[24] Arthur L Samuel. Machine learning. *The Technology Review*, 62(1):42–45, 1959.

[25] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3(19-48):5–1, 2010.

[26] Pramila P Shinde and Seema Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.

[27] Pariwat Ongsulee. Artificial intelligence, machine learning and deep learning. In *2017 15th international conference on ICT and knowledge engineering (ICT&KE)*, pages 1–6. IEEE, 2017.

[28] Jiyeon Yu, Angelica de Antonio, and Elena Villalba-Mora. Deep learning (cnn, rnn) applications for smart homes: a systematic review. *Computers*, 11(2):26, 2022.

[29] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[30] Ajay Shrestha and Ausif Mahmood. Review of deep learning algorithms and architectures. *IEEE Access*, 7:53040–53065, 2019.

[31] Anamika Dhillon and Gyanendra K Verma. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, 9(2):85–112, 2020.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[34] Comparison of different convolutional neural network activation functions and methods for building ensembles, 2021.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[37] Integrating dropout regularization technique at different layers to improve the performance of neural networks. *International Journal of Advanced Computer Science and Applications*, 2023.

[38] Tzu-Yen Hong and Chin-Chien Chen. Hyperparameter optimization for convolutional neural network by opposite-based particle swarm optimization and an empirical study of photomask defect classification. *Applied Soft Computing*, 148:110904, 2023.

[39] Anil Johny and KN Madhusoodanan. Dynamic learning rate in deep cnn model for metastasis detection and classification of histopathology images. *Computational and Mathematical Methods in Medicine*, 2021:1–13, 2021.

[40] Khaled Koutini, Hamid Eghbal-zadeh, and Gerhard Widmer. Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1987–2000, 2021.

[41] Ngoc-Son Pham, Sangwon Shin, Lei Xu, Weidong Shi, and Taeweon Suh. Starspa: Stride-aware sparsity compression for efficient cnn acceleration. *IEEE Access*, 2024.

[42] S Yegneshwar Yadhav, T Senthilkumar, S Jayanthy, and J Judeson Antony Kovilpillai. Plant disease detection and classification using cnn model with optimized activation function. In *2020 international conference on electronics and sustainable communication systems (ICESC)*, pages 564–569. IEEE, 2020.

[43] Rajendran Nirthika, Siyamalan Manivannan, Amirthalingam Ramanan, and Ruixuan Wang. Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study. *Neural Computing and Applications*, 34(7):5321–5347, 2022.

[44] Christian Garbin, Xingquan Zhu, and Oge Marques. Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia tools and applications*, 79(19):12777–12815, 2020.

[45] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

[46] Meenal V Narkhede, Prashant P Bartakke, and Mukul S Sutaone. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1):291–322, 2022.

[47] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[48] Parvathaneni Naga Srinivasu, Jalluri Gnana SivaSai, Muhammad Fazal Ijaz, Akash Kumar Bhoi, Wonjoon Kim, and James Jin Kang. Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm. *Sensors*, 21(8):2852, 2021.

[49] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[50] Analysis of densenet -mobilenet-cnn models on image classification using bird species data. 2023.

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[52] Emmanuel Mathews and J Jabez. Classifying alzheimer's disease from smri data using a hybrid deep learning approaches. *Salud, Ciencia y Tecnología-Serie de Conferencias*, 3:651–651, 2024.

[53] Nicodemus Songose Awarayi, Frimpong Twum, James Ben Hayfron-Acquah, and Kwabena Owusu-Agyemang. A bilateral filtering-based image enhancement for alzheimer disease classification using cnn. *Plos one*, 19(4):e0302358, 2024.

[54] Seung Kyu Kim, Quan Anh Duong, and Jin Kyu Gahm. Multimodal 3d deep learning for early diagnosis of alzheimer's disease. *IEEE Access*, 2024.

[55] Murat Avşar and Kemal Polat. Classifying alzheimer's disease based on a convolutional neural network with mri images. *Journal of Artificial Intelligence and Systems*, 5(1):46–57, 2023.

[56] Amir Ebrahimi, Suhuai Luo, Raymond Chiong, Alzheimer's Disease Neuroimaging Initiative, et al. Deep sequence modelling for alzheimer's disease detection using mri. *Computers in Biology and Medicine*, 134:104537, 2021.

[57] Rahmeh Ibrahim, Rawan Ghnemat, and Qasem Abu Al-Haija. Improving alzheimer's disease and brain tumor detection using deep learning with particle swarm optimization. *AI*, 4(3):551–573, 2023.

[58] Omar Altwijri, Reem Alanazi, Adham Aleid, Khalid Alhussaini, Ziyad Aloqalaa, Mohammed Almijalli, and Ali Saad. Novel deep-learning approach for automatic diagnosis of alzheimer's disease from mri. *Applied Sciences*, 13(24):13051, 2023.

[59] Zachary Burns. Group 52: Early detection of alzheimer's disease through machine learning in mri scans. *La Jolla, CA*, pages 92093–0238, 2019.

[60] Sunday Adeola Ajagbe, Kamorudeen A Amuda, Matthew A Oladipupo, F AFE Oluwaseyi, and Kikelomo I Okesola. Multi-classification of alzheimer disease on magnetic resonance images (mri) using deep convolutional neural network (dcnn) approaches. *International Journal of Advanced Computer Research*, 11(53):51, 2021.

[61] Suriya Murugan, Chandran Venkatesan, MG Sumithra, Xiao-Zhi Gao, B Elakkiya, Muthuramalingam Akila, and S Manoharan. Demnet: a deep learning model for early diagnosis of alzheimer diseases and dementia from mr images. *Ieee Access*, 9:90319–90329, 2021.

[62] tourist55. Alzheimer's dataset - 4 class of images. https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images/data, year.

[63] Nancy J Donovan, Rebecca E Amariglio, Amy S Zoller, Rebecca K Rudel, Teresa Gomez-Isla, Deborah Blacker, Bradley T Hyman, Joseph J Locascio, Keith A Johnson, Reisa A Sperling, et al. Subjective cognitive concerns and neuropsychiatric predictors of progression to the early clinical stages of alzheimer disease. *The American Journal of Geriatric Psychiatry*, 22(12):1642–1651, 2014.

[64] Shams Forruque Ahmed, Md Sakib Bin Alam, Maruf Hassan, Mahtabin Rodela Rozbu, Taoseef Ishtiak, Nazifa Rafa, M Mofijur, ABM Shawkat Ali, and Amir H Gandomi. Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, pages 1–97, 2023.

[65] Li Deng. Deep learning: from speech recognition to language and multimodal processing. *APSIPA Transactions on Signal and Information Processing*, 5:e1, 2016.

[66] Kristin P Bennett and JA Blue. A support vector machine approach to decision trees. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 3, pages 2396–2401. IEEE, 1998.

[67] Amey Thakur and Archit Konde. Fundamentals of neural networks. *International Journal for Research in Applied Science and Engineering Technology*, 9:407–26, 2021.

[68] Abhishek Gupta, Alagan Anpalagan, Ling Guan, and Ahmed Shaharyar Khwaja. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, 10:100057, 2021.

[69] Hugo Larochelle and QC Sherbrooke. Historique de travail. *Études*, 3:2Y5.

[70] Md Anwar Hossain and Md Shahriar Alam Sajib. Classification of image using convolutional neural network (cnn). *Global Journal of Computer Science and Technology*, 19(D2):13–18, 2019.

[71] BH Pansambal and AB Nandgaokar. Integrating dropout regularization technique at different layers to improve the performance of neural networks. *International Journal of Advanced Computer Science and Applications*, 14(4), 2023.

[72] Belhassen Bayar and Matthew C Stamm. Design principles of convolutional neural networks for multimedia forensics. *Electronic Imaging*, 29:77–86, 2017.

[73] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[74] Richard Elderman. *Exploring Improvements for Gradient Descent Optimization Algorithms in Deep Learning.* PhD thesis, 2019.

[75] Saad Hikmat Haji and Adnan Mohsin Abdulazeez. Comparison of optimization techniques based on gradient descent algorithm: A review. *PalArch's Journal of Archaeology of Egypt/Egyptology*, 18(4):2715–2743, 2021.

[76] NOOR AMIRAH BINTI AJMAL KHAN and SITI MAHANI MARJUGI. Prediction analysis of covid-19 in selangor by using backpropagation algorithm with conjugate gradient method. *Journal of Quality Measurement and Analysis JQMA*, 20(1):187–204, 2024.

[77] Reza Moradi, Reza Berangi, and Behrouz Minaei. A survey of regularization strategies for deep models. *Artificial Intelligence Review*, 53(6):3947–3986, 2020.

[78] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018.

[79] Murugan Subramanian, Anne Wojtusciszyn, Lucie Favre, Sabri Boughorbel, Jingxuan Shan, Khaled B Letaief, Nelly Pitteloud, and Lotfi Chouchane. Precision medicine in the era of artificial intelligence: implications in chronic disease management. *Journal of translational medicine*, 18:1–12, 2020.

[80] Maren Mahsereci, Lukas Balles, Christoph Lassner, and Philipp Hennig. Early stopping without a validation set. *arXiv preprint arXiv:1703.09580*, 2017.

[81] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[82] R Ibrahim, R Ghnemat, and Al-Haija Abu. Q. improving alzheimer's disease and brain tumor detection using deep learning with particle swarm optimization. ai 2023, 4, 551–573, 2023.

[83] Yasmina M Elgammal, MA Zahran, and Mohamed M Abdelsalam. A new strategy for the early detection of alzheimer disease stages using multifractal geometry analysis based on k-nearest neighbor algorithm. *Scientific reports*, 12(1):22381, 2022.

[84] Himanshu XD. Activation functions: Sigmoid, relu, leaky relu, and softmax — basics for neural networks and deep learning. *Medium*, 2019.

[85] Jason Brownlee. Rectified linear activation function for deep learning neural networks. *Machine Learning Mastery*, 2020.

[86] Google Cloud. Using vertex ai for peer group benchmarking in capital markets, Year of publication.