الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria

وزارة التعليم الـعـالي والبحـث العـلمـي
Ministry of Higher Education and Scientific Research

جـامعة سعد دحلب البليدة
SAAD DAHLAB University of BLIDA

كلية التكنولوجيا
Faculty of Technology

قسم الإلكترونيك
Electronics Department

Final year project thesis

Presented by

SNOUSSI Bouchera
&
HICHEUR Chahla

For the Master's degree in Telecommunication
'Network and Telecommunications' option

Topic

Diagnosis of Parkinson's Disease using Deep Learning

Proposed by: Mrs. BOULKRINAT Nour El Houda
&
Pr. BENBLIDIA Nadjia

Academic year 2023-2024

Acknowledgements

First and foremost, we would like to thank Almighty God for granting us the health, willpower, and patience to embark on and complete our journey and studies.

We extend our heartfelt appreciation to our supervisor, **Mrs. Nour El Houda BOULKRINAT**, for her invaluable guidance, unwavering support, and insightful feedback, which have been instrumental in shaping this work. We are equally grateful to my co-supervisor, **Mrs. Nadjia BENBLIDIA**, for her encouragement and assistance.

We are indebted to the esteemed members of the jury for their time, expertise, and constructive criticism, which have contributed to enhancing the quality of this research.

We would also like to extend our deepest thanks to our beloved parents, siblings, and friends for their unwavering encouragement, endless support, and motivation throughout this endeavor. Their belief in us has been our driving force.

We are grateful to all those who, directly or indirectly, have contributed to the completion of this thesis. Your support has been invaluable.

Lastly, A big thank you also to all the teachers of the Electronics Department at Saad Dahleb University in Blida for all the knowledge they have imparted to us throughout our university studies.

Dedication

The journey was neither short nor easy, but I did it. Praise be to Allah who facilitated the beginnings and brought us to the endings.

First, I want to dedicate this success to myself first, to my resilient self that endured setbacks, persevered, and completed the journey despite the difficulties.

Dedicated to my beloved mother, Hassina, whose love and sacrifices have been my guiding light.

To my father, Noureddine, whose wisdom and encouragement have shaped my path.

To my sisters, Chaimaa and Aya, and to my brother, Fares, whose unwavering support has been my strength, making every challenge easier to bear.

To my dear nephew, Djaouad, who fills my heart with joy.

To my cherished little nieces, Safa and Meriem, who bring endless smiles into my life.

To my grandmother Azouzou, who prays for my kindness and success, prayers that are close to my heart.

To all my family members for their constant love, understanding, and support; and to my dear friends, whose presence and encouragement have made this journey truly memorable.

Bouchera.

Dedication

With gratitude to the Almighty, I am pleased to present this thesis, a culmination of perseverance and dedication.

First and foremost, I extend appreciation to myself for the relentless pursuit of this achievement.

This work is dedicated to my father, Zouhir, whose unwavering encouragement and support have been instrumental in shaping my academic journey.

To my mother, Zouhra, whose boundless love and sacrifices have been a constant source of inspiration, I owe a debt of gratitude.

To my cherished sister, Wissam, and my brothers, Walid and Anes, whose steadfast support has been invaluable throughout this endeavor.

I extend heartfelt thanks to my nephews, Iyad, Djawad, and Youcef, whose presence fills my heart with joy and serves as a reminder of what truly matters.

To my dear friends, whose unwavering presence and encouragement have been a source of strength, I am profoundly grateful.

May this work stand as a testament to the collective support and guidance I have received along this journey.

Chahla.

ملخص :لا يزال تطور مرض باركنسون يمثل تحديًا كبيرًا، مع تزايد عدد الحالات وصعوبة التشخيص الدقيق على الرغم من التقدم الطبي وتقنيات التصوير. يمكن لتقنيات الذكاء الاصطناعي تحسين دقة الكشف. هدفنا هو تطوير نظام لتصنيف صور الرنين المغناطيسي لتشخيص مرض باركنسون باستخدام قاعدة البيانات NTUA مع نماذج الشبكات العصبية التلافيفية المدربة مسبقًا مثل VGG16 وResNet50 وDenseNet121 وMobileNetV2 وVGG19 بالإضافة الى نموذج CNN المقترح.

كلمات المفاتيح: الذكاء الاصطناعي، صور الرنين المغناطيسي، نماذج مدربة مسبقًا، الشبكات العصبية الالتفافية، NTUA.

Résumé : La progression de la maladie de Parkinson reste un défi majeur, avec un nombre de cas en augmentation et un diagnostic exact souvent difficile malgré les avancées médicales et les techniques d'imagerie. Les techniques d’Intelligence Artificielle peuvent améliorer la précision de la détection. Notre objectif est de développer un système de classification d'images IRM pour le diagnostic de la maladie de Parkinson, en utilisant la base de données NTUA avec des modèles pré-entraînés de réseaux de neurones convolutifs tels que VGG16, ResNet50, DenseNet121, MobileNetV2 et VGG19, ainsi que le modèle CNN proposé.

Mots clés : Intelligence artificielle, Images IRM, Modèle pré-entrainés, CNN, NTUA.

Abstract: The progression of Parkinson's disease remains a major challenge, with an increasing number of cases and often difficult exact diagnosis despite medical advancements and imaging techniques. AI techniques can enhance detection accuracy. Our objective is to develop an MRI image classification system for Parkinson's disease diagnosis using NTUA dataset with pre-trained convolutional neural network models such as VGG16, ResNet50, DenseNet121, MobileNetV2, and VGG19, with a proposed CNN model.

Keywords: Artificial intelligence, MRI images, Pre-trained model, CNN, NTUA.

Lists of acronyms and abbreviation

AAAI	American Association for Artificial Intelligence
AD	Alzheimer Disease
Adam	Adaptive Moment Estimation
AE	Autoencoders
AI	Artificial Intelligence
ANN	Artificial Neural Network
BET	Brain Extraction Technique
CNN	Convolutional Neural Network
CT	Computed Tomography
DNN	Deep Neural Network
FFNN	Feedforward Neural Networks
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
HC	Healthy Controls
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
KNN	K Nearest Neighbor
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-layer Perceptron
MRI	Magnetic Resonance Imaging
NTUA	National Technical University of Athens
PD	Parkinson Disease
PPMI	Parkinson's Progression Markers Initiative
RBM	Restricted Boltzmann Machines
RBM	Restricted Boltzmann Machine
RLS	Restless Legs Syndrome
RNN	Recurrent Neural Network
SPM	Statistical Parametric Mapping

SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UPDRS	Unified Parkinson’s Disease Rating Scale

Table of contents

Figures list

Table list

General introduction

Parkinson's disease (PD) is a progressive neurodegenerative disorder that affects millions of people worldwide, primarily impacting older adults. It severely limits individuals' physical and functional abilities, diminishing their mobility and capacity to perform daily activities. Early and accurate diagnosis of PD is crucial for implementing timely interventions that can enhance patients' quality of life and slow disease progression. However, despite advancements in medical imaging and diagnostic techniques, diagnosing PD, especially in its early stages, remains a significant challenge.

In recent years, technology has made significant strides, particularly with the spread of digitalization, which has enhanced various fields, including medical diagnostics. The advent of artificial intelligence (AI) and machine learning has introduced new possibilities for analyzing medical data. Machine learning, a subset of AI, allows computers to learn from data and make predictions without explicit programming. Deep learning, a more specialized subfield of machine learning, utilizes artificial neural networks to perform complex tasks, traditionally done by humans, such as facial and speech recognition and decision-making.

This study explores the application of AI, specifically convolutional neural networks (CNNs), in detecting Parkinson's disease from MRI images. CNNs, a type of deep learning architecture, have shown remarkable effectiveness in image analysis tasks. We investigate both custom pre-trained and newly developed CNN models to assess their effectiveness in accurately diagnosing PD. Pre-trained models such as VGG16, VGG19, ResNet50, MobileNetV2, and DenseNet121 are utilized to compare their performance against our proposed CNN model.

A comprehensive examination of machine learning principles, techniques, and various algorithms is provided to establish the foundational knowledge required for deep learning. The study delves into the stages of Parkinson's disease, its motor and non-motor symptoms, and methods for detecting PD through handwriting analysis, voice assessment, and MRI imaging. Furthermore, the powerful role of transfer learning, enhanced by the ImageNet dataset, is highlighted by exploring advanced neural network designs like VGG, ResNet, DenseNet, and

MobileNet. These pre-trained models are fine-tuned for the specific task of diagnosing PD, with adjustments to parameters such as batch size, epochs, and learning rate.

We also discuss the platforms and tools used to implement our networks, including Google Colab, Google Drive, and associated libraries. The study covers the database used, data augmentation techniques, and preprocessing methods employed. Finally, the performance of the models is evaluated using metrics such as accuracy, precision, recall, F1 score, and confusion matrices, and the results are summarized to highlight the effectiveness of the deep learning models in diagnosing Parkinson's disease.

The structure of this paper is organized as follows:

Chapter 1 provides an overview of fundamental concepts in AI, machine learning, and deep learning, with a focus on CNNs and their application in medical image analysis.

Chapter 2 delves into Parkinson's disease, detailing its stages, symptoms, and current diagnostic methods.

Chapter 3 introduces pre-trained models and describes our custom CNN model tailored for PD detection. It discusses the architectural details and functionalities of each model used in this study.

Finally, chapter 4 presents the results of our experiments, comparing the performance of each model in terms of accuracy, confusion matrix, and F1-score. This analysis aims to identify the most effective approach for detecting Parkinson's disease from MRI images, contributing to advancements in early diagnosis and patient care.

1.1 Introduction

In recent years, technology has made great strides in many areas, mainly due to the spread of digitalization. The data can be of many different types, such as photos, videos, text, voice recordings or other forms of information. Nowadays, people are familiar with artificial intelligence (AI) and machine learning. Machine learning is a subset of AI that empowers computers to acquire knowledge without the need for explicit programming. Deep learning is a subfield of machine learning; AI uses predictions to optimize and solve complex tasks that humans have historically done, such as facial and speech recognition, and decision making.

In this chapter, we present the principles of machine learning, its capabilities, the various techniques used, the different types and its algorithms. Next, we present the artificial neural network model that forms the basis of deep learning.

1.2 Artificial Intelligence (AI)

As the AAAI (American Association for Artificial Intelligence) give a definition provided in the foreword to [1]; artificial learning refers to the ability of a system to autonomously acquire and integrate knowledge. Artificial intelligence is the process of making an artificial being intelligent through various theories and techniques, aimed at enabling machines to think, process, and even learn like humans. The figure 1.1 shows the subfield of AI.

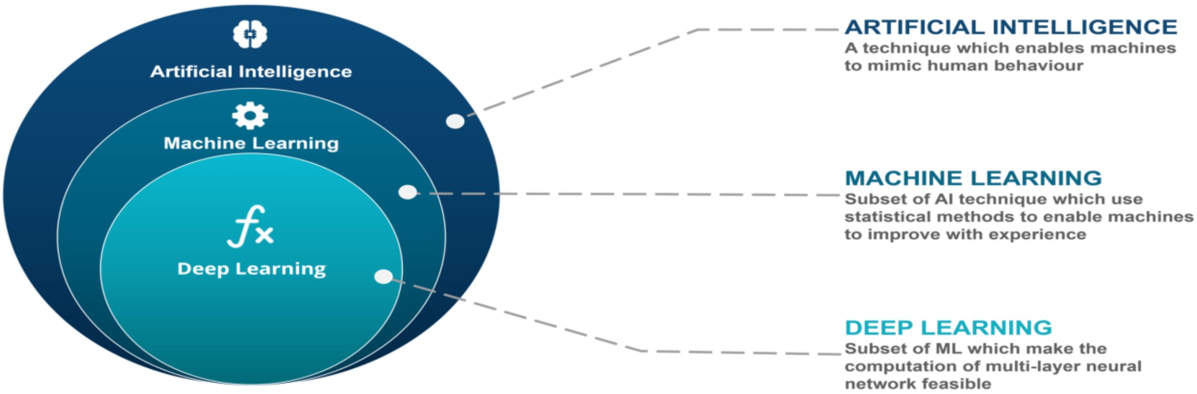

Artificial Intelligence
Machine Learning
f_x
Deep Learning
ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour
MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

Figure 1.1: *The AI domains.*

1.3 Machine Learning (ML)

Artificial or machine learning is defined as, "A subfield of artificial intelligence concerned with giving machines the ability to improve at performing a task, by interacting with their environment". It refers to the development, analysis, and implementation of methods that enable a machine, in the broadest sense, to evolve and perform tasks related to artificial intelligence using a learning process; also, it is a set of methods for extracting knowledge from available observations and using it to search for new information, or to describe observations differently. This learning process results in a system that optimizes itself according to the environment, experience and results observed [1].

Machine learning is the field of study that enables computers to learn without being explicitly programmed. This data can be of many different types, such as photos, videos, text, voice recordings or other forms of information.

1.3.1 Types of Machine Learning

There are many types of Machine Learning such as supervised learning, unsupervised learning, semi-supervised leaning, and reinforcement learning (figure 1.2).

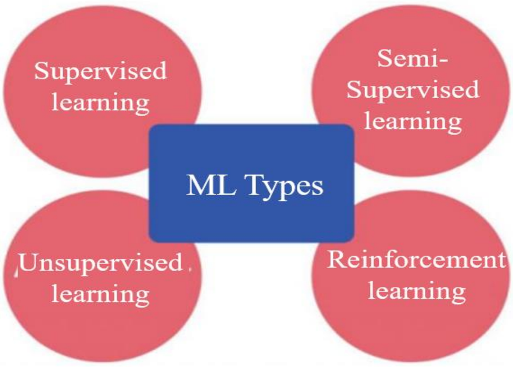


Figure 1.2: *The machine learning types.*

1.3.1.1 Supervised learning

Supervised learning focuses on the creation of learning models that establish links between variables and previously known results, using labeled datasets. In this process, the machine is fed with example data, represented by different features (X), and the correct output values associated with these data (Y) [2]. In supervised learning, a dataset of labeled data points is used for building predictive models. This process involves instructing a machine based on input data that are linked to corresponding labels. This enables the algorithm to make predictions about outputs based on the input data. Supervised learning is usually used for classification and regression problems.

The figure 1.3 explains the functionality of a supervised learning algorithm by analyzing labeled inputs and comparing them with new test inputs to generate prediction.

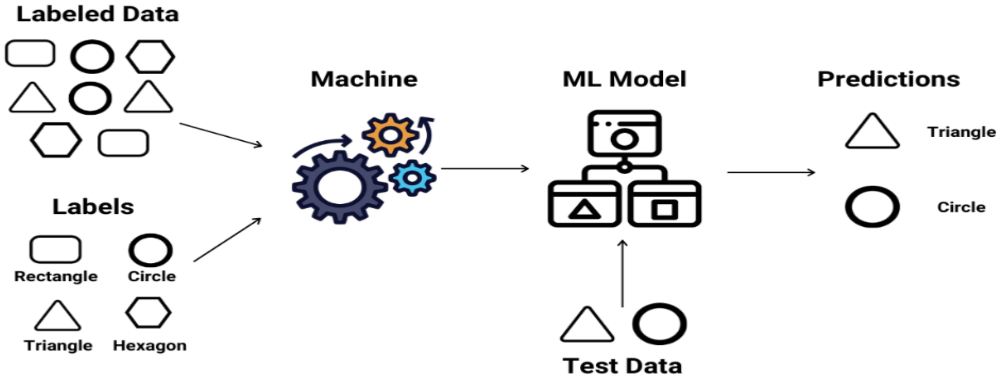


Figure 1.3: *Supervised learning algorithm.*

1.3.1.1.1 Classification

Classification is the task in which ideas and objects are determined, differentiated and understood. Classification defines that objects are collected into categories, usually for some specific purpose [3].

1.3.1.1.2 Regression

Regression is a supervised learning method that seeks to identify relationships between variables and anticipate continuous values as a function of these variables. When the expected result is a continuous value, the task is referred to as a regression problem [4].

1.3.1.2 Unsupervised learning

The model is learned only from the input data. Unsupervised learning algorithms learn the distribution and detect the similarity among input samples. It then separates them into different groups based on the distribution and similarity index. Unsupervised learning algorithms are frequently applied for clustering and dimensionality reduction [5].

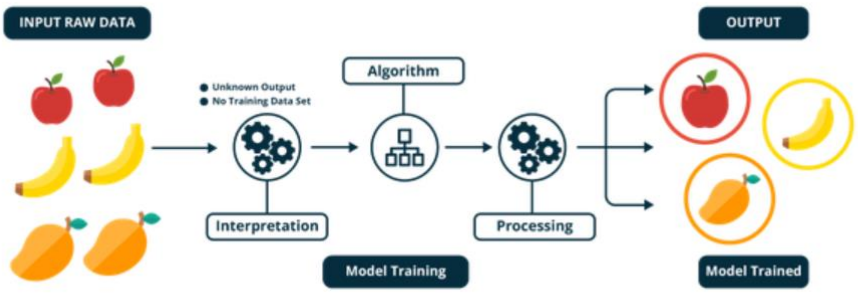


Figure 1.4: *Unsupervised learning.*

The figure 1.4 explains the functionality of an unsupervised learning algorithm by analyzing unlabeled inputs and regrouping the output according to their similarities.

1.3.1.2.1 Clustering

According to Jiand et al, clustering implies partitioning a particular dataset into groups, whose components are similar to each other [6]. It mainly deals with finding a structure or pattern in a collection of uncategorized data. It can be considered as the most important unsupervised learning problem; it deals with finding a structure in a collection of unlabeled data.

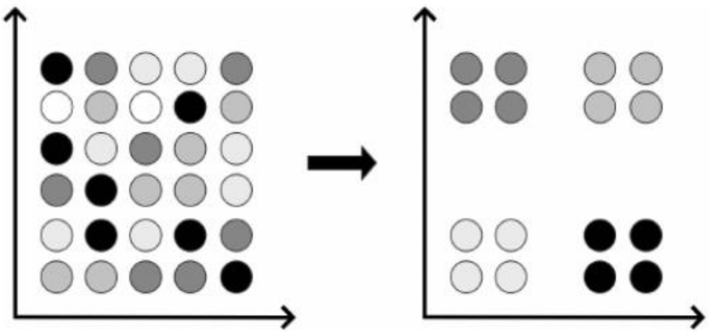


Figure 1.5: *The clustering.*

1.3.1.3 Semi-supervised learning

Semi-supervised learning is a technique in machine learning that uses a combination of unsupervised learning and supervised learning. It can be fruit-full in those areas of machine learning and data mining where the unlabeled data is already present and getting the labeled data is a tedious process. With more common supervised machine learning methods, you train a machine learning algorithm on a “labeled” dataset in which each record includes the outcome information [7]. The figure 1.6 demonstrates the labeled and unlabeled data inputs within the framework of a semi-supervised learning algorithm, resulting an output prediction.

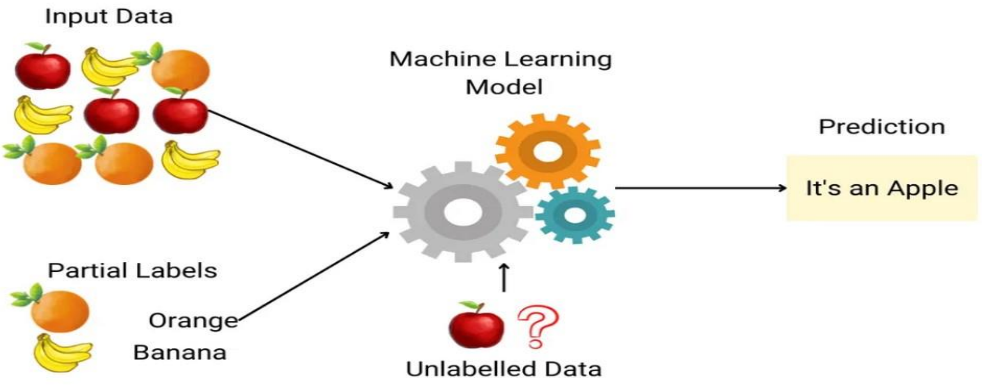


Figure 1.6: *Semi-supervised learning.*

1.3.1.4 Reinforcement learning

Reinforcement learning involves learning by trying different actions. An agent observes the situation and takes actions. Following each action, a numerical reward is provided. The goal for the agent is to increase the total reward it gets over time. In standard reinforcement learning, an agent interacts with its environment. At each step, the agent gets information about the current state of the environment, selects an action, and acts accordingly. This action affects the environment, generating feedback in the form of a reward or reinforcement signal. The agent's goal is learning to take actions that maximize the rewards received in the long term [8].

1.3.2 ML algorithms

The field of machine learning features four notable algorithms: decision trees, support vector machines (SVM), K-nearest neighbors (KNN), and K-means.

1.3.2.1 Decision Tree

Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represent an event or choice and the edges of the graph represent the decision rules or conditions. Each tree consists of nodes and branches. Each node represents attributes in a group that is to be classified and each branch represents a value that the node can take [7].
The figure 1.7 illustrates an instance of decision-making regarding the acceptance of a new job offer based on selecting optimal conditions.

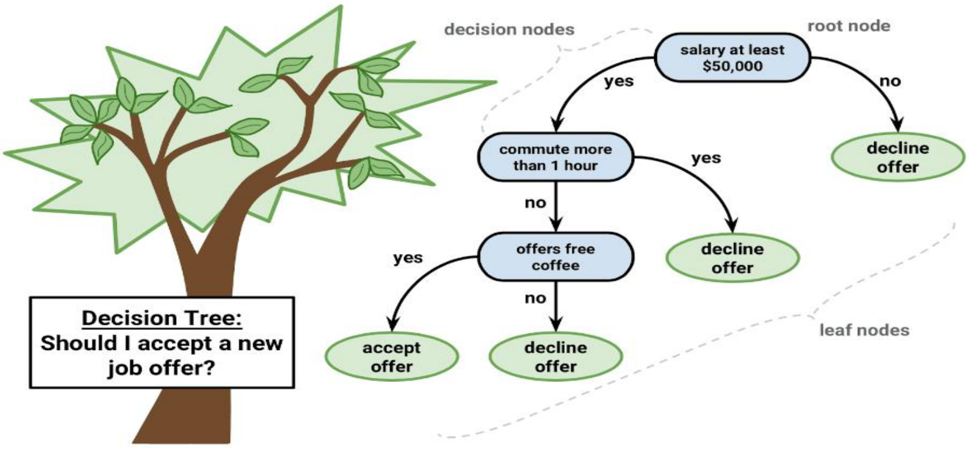


Figure 1.7: *The decision tree.*

1.3.2.2 Support Vector Machine (SVM)

The support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. It basically, draw margins between the classes. The margins are drawn in such a fashion that the distance between the margin and the classes is maximum, and hence, minimizing the classification error [7].

The figure 1.8 [4] shows the SVM algorithm; which divides into two classes.

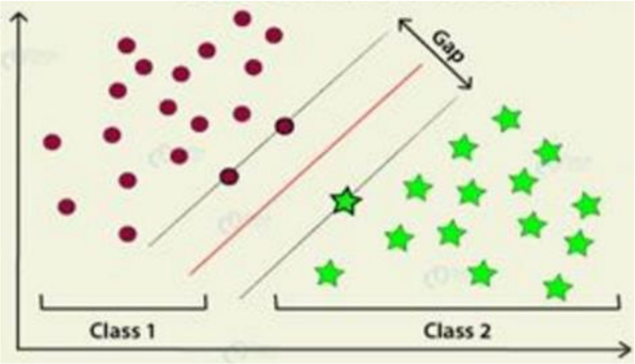


Figure 1.8: *The SVM.*

1.3.2.3 K nearest neighbors (KNN)

The K-nearest neighbor (KNN) algorithm is a supervised learning method that operates based on the principle of "tell me who your friends are, and I will tell you who you are". This means it classifies a data point based on the class of its nearest neighbors. It classifies a data point by considering the classifications of its nearest neighbors, it ranks the new instances using the information provided by the k nearest neighbors, so that the class assigned is the most common among them (majority vote). It stores all training data using the distance function [9].

Figure 1.9 illustrates how the KNN algorithm operates. A new glass has been added to a group consisting of red and white glasses of juice. The objective is to determine whether the new glass is red or white. The k is chosen equal to 5. As a result, the new point would be classified as a red juice since four out of five neighbors are red.

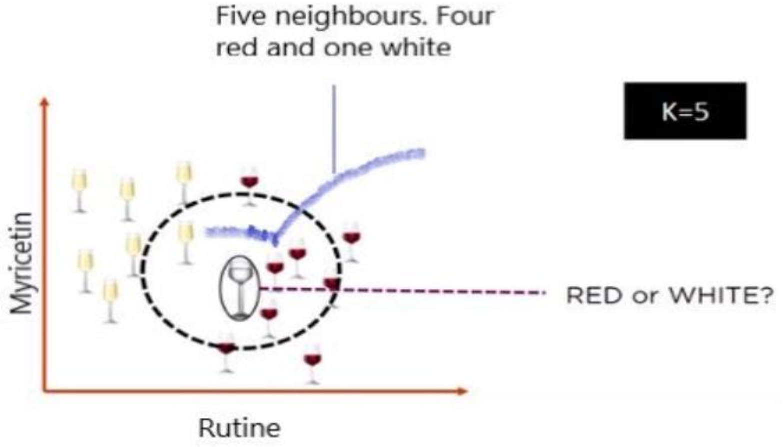


Figure 1.9: *The KNN example.*

1.3.2.4 K means

Simple unsupervised learning algorithm consisting on grouping data into a set of k clusters. Each cluster has a centroid μ, the center of the cluster. Starting from randomly selected k centroids, in the first step, distances of each of the samples in the dataset to each of the centroids are calculated. Then, samples are assigned to the nearest centroid. Finally, each centroid is updated to the average value of the samples assigned to this corresponding cluster. The process is repeated until the centroids are stable [10].

1.4 Neural Networks

1.4.1 Definition of Neural Networks

Neural networks have their roots in attempts to create a mathematical model of the human brain. The earliest investigations date back to 1943 and were carried out by W.M. Culloch and W. Pitts [11]. They proposed that a neural impulse arises from a simple calculation performed by each neuron and that complex thinking emerges from the collective interactions of interconnected neurons. It is a collection of basic components known as "neurons," a nod to their biological counterparts, which are interconnected. Each neuron independently performs a simple task, and their collaborative operation results in the emergence of complex global properties. This system functions in a highly parallel manner. Information is stored in a distributed fashion within the network, using synaptic weights or activation functions. There is no distinct separation between memory and computation they are closely intertwined.

Neural networks are not programmed but rather trained through a learning process. Tasks that suit neural networks well include association, classification, discrimination, prediction, estimation, and the control of intricate processes. Artificial neural networks are constructed with varying degrees of inspiration from the human brain's operation, primarily based on the concept of neurons.

1.4.2 Biological Neurons

In biology, a neuron is a functional cell of the nervous system responsible for transmitting electrical signals linked to our metabolism. It is made up of dendrites connected to the cell body and an axon that ends in synapses. The electrical signal passes from the cell body to the synapses via the axon, then to the cell body of the next neuron where it undergoes modifications facilitated by the dendrites. This biological model has served as the basis for the development of artificial neurons, which have the ability to receive, process and relay data to other neurons [12], as shown in figure 1.10 [11].

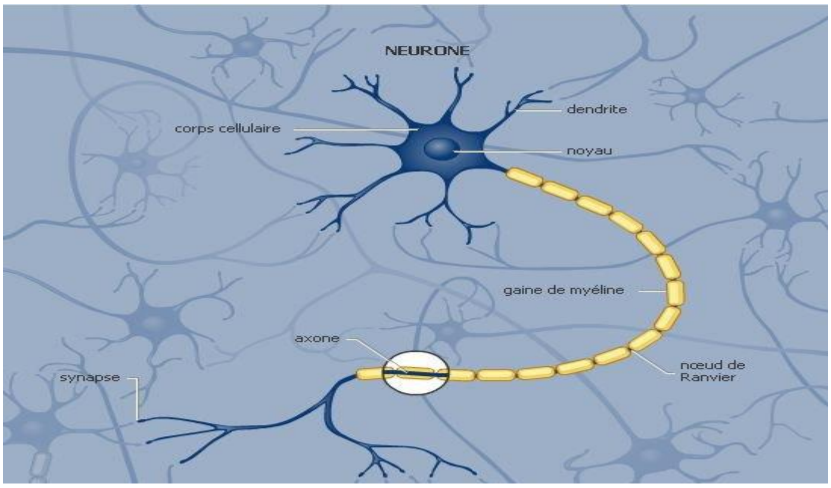


Figure 1.10: *Model of a biological neuron.*

1.4.3 Formal neurons

The ‘formal neuron’ is an algebraic function that is both non-linear and constrained. Its result is influenced by parameters called coefficients or weights. The variables used in this function are generally called the ‘neuron inputs’, while the calculated result is called the ‘output’. In essence, a neuron functions primarily as a mathematical operator, and its numerical value can be determined using a concise set of software instructions. It has become common practice to represent a neuron in graphical form, as shown in figure 1.11 [13].

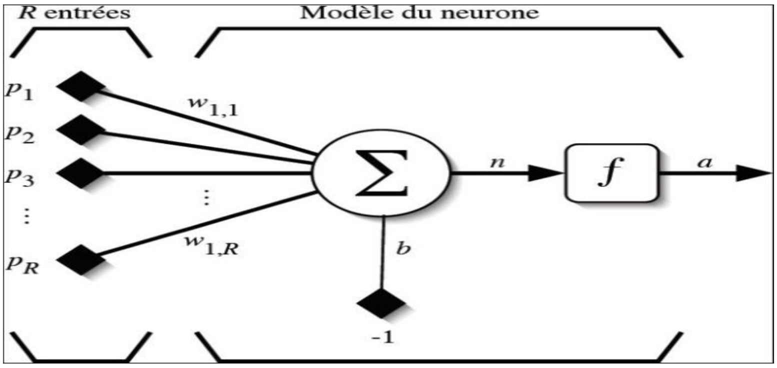


Figure 1.11: *Model of a formal neuron.*

The variables P_R represent input vectors, which may come from the outputs of other neurons or from sensory stimuli (various external signals or inputs detected by the sensory organs and transmitted to the nervous system).

The values $W_{1,R}$ are the synaptic weights of neuron R. They correspond to the synaptic efficacy in biological neurons ($W_{1,R}$> 0: excitatory synapse; $W_{1,R}$< 0: inhibitory synapse). These weights are responsible for weighting the inputs and can be adjusted during the learning process.

Bias: the input often takes on values of -1 or +1, which adds flexibility to the network by allowing the activation threshold of neurons to be adjusted by modifying weights and bias (b) during the learning process.

Kernel: This combines all the inputs and the bias, and calculates the neuron's output using a generally non-linear activation function, which improves the network's learning flexibility [11].

1.4.4 Modelling a formal neuron

The modelling process involves the creation of a neural network system from an artificial, as opposed to a biological, point of view. This approach assumes that, based on biological principles, there is an equivalence for each constituent element of the biological neuron, resulting in a separate model for each of these elements. We can summarize this modelling process using Table 1.1, which provides a lucid representation of the transition from the biological neuron to the formal neuron.

Neurone biologique	Neurone artificiel
Synapses	Weights of connections
Axones	Output signal
Dendrites	Input signal
Noyau cellulaire	Activation function

Table 1.1: *Analogy between the biological neuron and formal neuron [11].*

1.5 The architecture of an artificial neural network

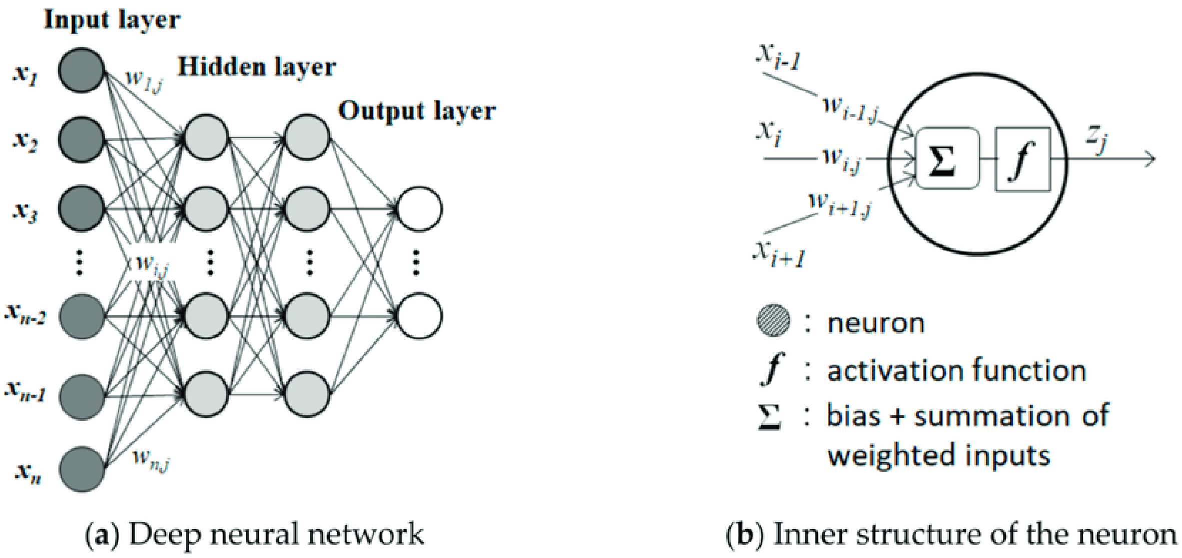
The structure of a neural network is determined by the way in which the multiple neurons are organized, with most networks using the same type of neuron. However, the specific architecture is adapted to the task in hand. Depending on the learning task, a neural network generally consists of the following elements.

A) Input layer: This layer comprises the neurons responsible for transmitting input signals into the network. Each neuron in this layer is connected to the next layer.

B) Hidden layers: These layers can vary in number and complexity and play a crucial role in revealing relationships between variables. The choice of the number of hidden layers and neurons is guided by the practitioner's intuition and expertise.

C) Output layer: The output layer represents the final result of the network, often called the prediction layers.

Figure 1.12 below provides a visual representation of the architecture of an artificial neural network [13].



(**a**) Deep neural network

(**b**) Inner structure of the neuron

Figure 1.12: *Structure of a surface neural network and its neuron [13].*

1.5.1 Activation function

There are several activation functions, generally non-linear, such as ReLU, Softmax and sigmoid (as shown in Table 1.2). The purpose of these functions is to manipulate the output within a specific range ‘they are able to spatially modify the data representation, allowing it to change from a linear to a non-linear form’. The choice of activation function depends on the nature of the model you wish to represent [14].

ReLU (Rectified linear unit) function: Known for its simplicity, which makes it the most widely used of the other activation functions, the rectified linear unit or ReLU function [15] is a function that is used to determine the maximum between x and 0.

It uses the following formula:

$$\mathbf{ReLU(x) = max\ (x, 0)} \quad \begin{cases} \mathbf{x\ if\ x\ >\ 0} \\ \mathbf{0\ if\ x\ <\ 0} \end{cases} \qquad \mathbf{(1.1)}$$

Softmax: This is a function that is often used in classification models for multi-class problems. It treats each vector independently of the others and transforms a real vector into a probability vector. The input axis to which Softmax is applied is determined by the axis argument [16].

It uses the following formula:

$$\mathbf{Softmax(x)} = \frac{e^{zj}}{\sum_{k=1}^{k} e^{z\,k}} \ \mathbf{Pour\ j=1, \ldots, K} \qquad \mathbf{(1.2)}$$

Sigmoid: This is a function used in binary classification where the model has to determine only two labels since the results are always between 0 and 1[16].

It uses the following formula:

$$\boldsymbol{Sigmoid}(\mathrm{x}) = \frac{1}{1+ e^{-x}} \qquad \mathbf{(1.3)}$$

This table resume the activation function [14].

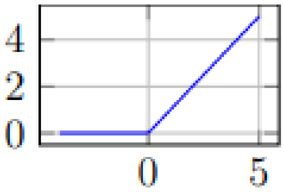
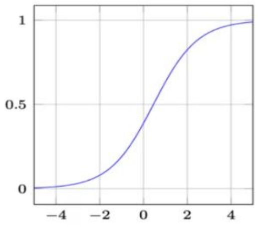
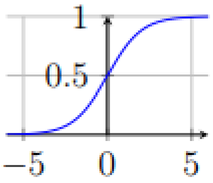
Function	Equation	Graphe
ReLU	$f(x) = max(0, x)$	
Softmax	$f(x) = \dfrac{e^{zj}}{\sum_{k=1}^{k} e^{z\,k}}$	
Sigmoïde	$f(x) = \dfrac{1}{1 + e^{-x}}$	

***Table 1.2:** Activation functions.*

1.6 Different neural network architectures

Network architectures can be divided into two main categories based on the connections between neurons: '**feedforward neural networks'** and '**recurrent neural networks'**. In direct progression neural networks, signals flow in a single direction, from input to output, without any feedback loop, which means that the outputs do not influence the inputs. In contrast, recurrent neural networks have feedback loops where outputs can be reintroduced into the network as inputs, either to the same neurons or to other neurons, creating cycles [17].

1.6.1 Feedforward neural networks (FFNN)

FFNN Also known as non-loop neural networks, have a distinctive feature: information flows in one direction only, from inputs to outputs, with no backtracking. These networks use supervised learning by error correction, where the error is used to adjust the weights of the neurons by feeding it back to the inputs. The simplest form of neural network follows this feed-forward structure. In such a structure, the neurons are connected in a single direction: the signals start at the input layer, pass through the hidden layers and end up at the output layer (figure 1.13). Each neuron in the hidden and output layers is connected to each neuron in the previous layer, allowing information to propagate through the network. This type of structure has stable behavior. Non-looped (feed-forward) neural networks are highly effective for addressing specific problems, making them the most commonly used type of neural network [17].

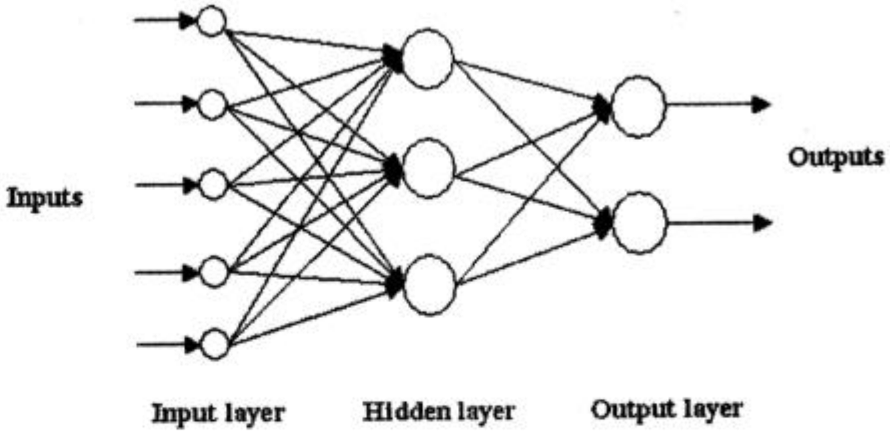


***Figure 1.13**: A multilayer Feedforward neural network.*

1.6.2 Recurrent neural networks

Unlike a Feed-Forward Neural Network, a Recurrent Neural Network (RNN) is a neural network whose connection graph contains at least one cycle (figure 1.14). The connectivity of units in recurrent neural networks is not limited, as in the case of feedforward networks, to architectures in which information propagates from input to output layer by layer. Any type of connection is allowed, from one neuron to any other, including itself. This gives rise to dynamic behaviors which can be highly complex. A discrete-time loop neural network is therefore governed by one or more non-linear difference equations, resulting from the composition of the functions performed by each of the neurons and the delays associated with each of the connections [18].

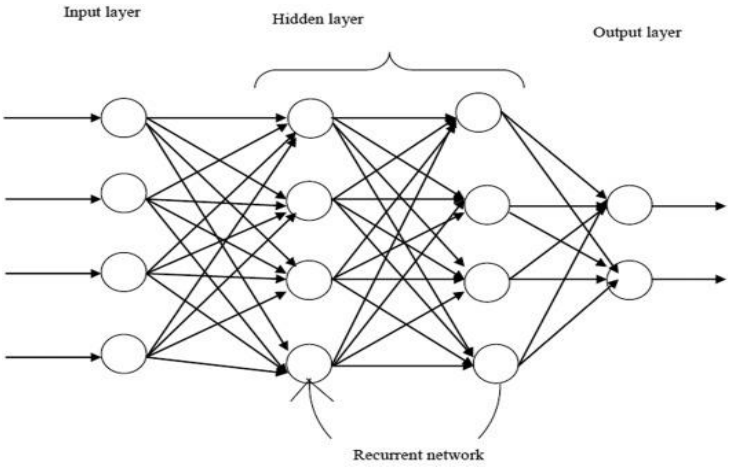


Figure 1.14: *Recurrent Neural Network [19].*

1.7 Deep Learning (DL)

Deep learning represents a branch of machine learning that has evolved from neural networks and has benefited from the availability of abundant training data. It has achieved substantial success across numerous applications compared to shallow machine learning models. This success can be attributed to the increased computational capabilities of computers, enabling the creation of complex models that excel in processing and learning. The key advantages of deep learning in contrast to shallow machine learning models include its superior performance with large datasets and its incorporation of both feature learning and model training within a unified architecture [20]. On the other hand, deep learning algorithms extract knowledge from training data in a hierarchical manner through multiple layers of non-linear processing, rendering them highly adaptable for modeling intricate relationships [21].
This approach also minimizes the need for human intervention and reduces wasted time.

1.7.1 Definition of Deep Learning

Deep learning is a part of machine learning that uses complex models and algorithms to simulate the way the human brain works. These models are known as artificial neural networks (ANNs). When the brain receives new information, it compares it with what it already knows to understand it better. It labels and categorizes this information to give it meaning. Deep learning acts in the same way, using the same methods to process and understand new data [22].

1.7.2 Difference between Deep Learning and Machine Learning

The difference between deep learning and machine learning is summarized in the table 1.3 below.

	Deep Learning (DL)	**Machine Learning (ML)**
Data	Hundreds or thousands of data.	Performs well with a small to a medium dataset.
Hardware requirements	Requires machines with GPU.	Works with low-end machines (using CPU is often sufficient).
Feature extraction	Automatically learns features.	Manual feature engineering often needed.
Training time	Longer due to complexity.	Usually faster.

Table 1.3: *Difference between DL and ML.*

1.7.3 Neural Network Model

Neural networks characterized by a significant number of parameters and layers, an illustrative example of this is the multi-layer perceptron (MLP).

1.7.4 A multilayer perceptron (MLP)

A multilayer perceptron (MLP) is a type of neural network characterized by the organization of neurons into several consecutive layers, unlike single-layer or simple artificial neural networks (ANNs). In an MLP, the connections between neurons are strictly unidirectional, running from lower to higher layers, and neurons in the same layer have no connections with each other, meaning that a neuron can only transmit its information to neurons in subsequent layers (figure 1.15). Designing the architecture of an MLP involves decisions about the number of hidden layers, the number of neurons in each layer, the nature of the connections between neurons and the properties of the neurons in each layer [23].

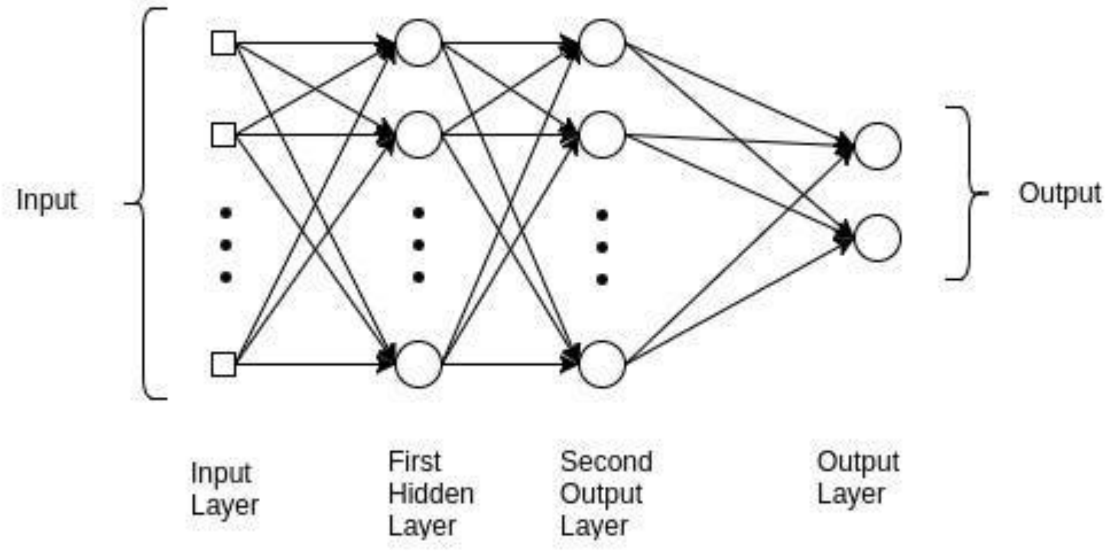


Figure 1.15: *Multilayer perceptron (MLP) [24].*

1.7.5 Convolutional Neural Network (CNN)

The term 'Convolutional Neural Network' signifies that this network makes use of a mathematical operation known as convolution. CNNs represent a specialized category of neural networks where convolution is employed in at least one of their layers instead of the more general matrix multiplication. These networks are highly regarded as effective learning algorithms, particularly in handling the convolution operation, which aids in extracting valuable features from locally related data points. The outcomes of the convolutional kernels are then directed to a non-linear processing unit. This unit not only assists in learning abstract features but also introduces non-linearity into the feature space's functionalities. This non-linearity gives rise to distinct activation patterns for various responses, thereby facilitating the learning of semantic differences in images [25]. The topology of a CNN is characterized by several learning stages, encompassing convolutional layers, non-linear processing units, and subsampling layers (figure 1.16).

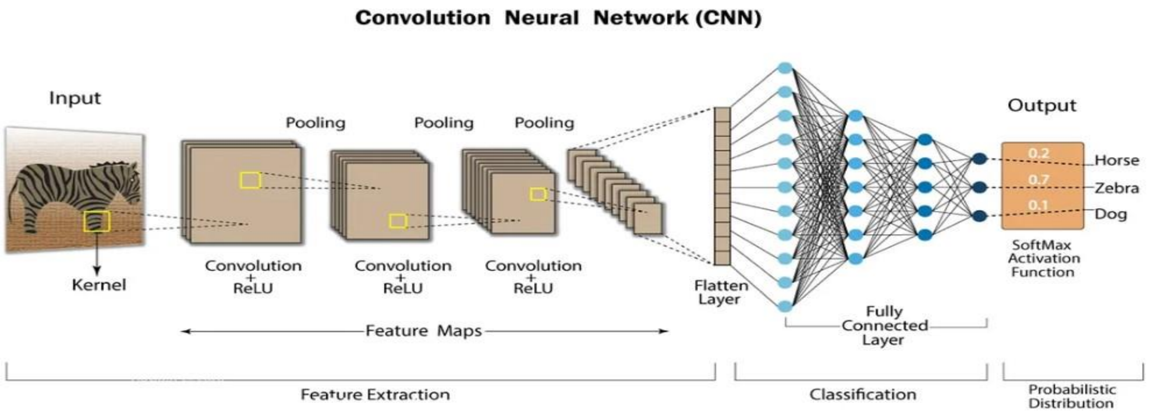
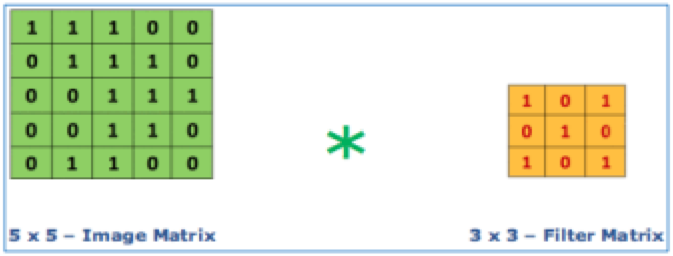


Figure 1.16: The general structure of a CNN network.

1.7.5.1 Convolution layer

The convolution layer serves as the initial step in identifying the features of an input image. It does this by preserving pixel relationships and acquiring image characteristics by analyzing small portions of the input data. This process is essentially a mathematical operation involving two inputs: an image matrix and a filter or kernel [26]. Therefore, the role of this layer is to analyze the input images and detect the presence of a set of features. The output of this layer is a set of feature maps. A basic example of this filter operation for a convolution step is shown in Figure 1.17 below [27].



$$\begin{matrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{matrix} \quad * \quad \begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}$$

5 x 5 – Image Matrix

3 x 3 – Filter Matrix

***Figure 1.117:** Example explaining the convolution operation.*

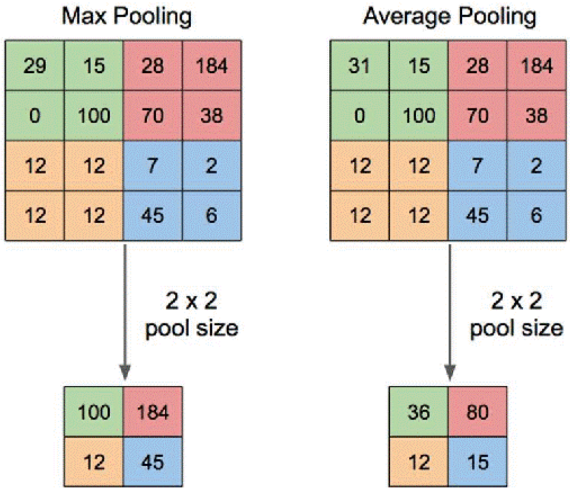
1.7.5.2 Pooling Layer

Typically placed between two convolution layers, the pooling layer facilitates a sampling-based discretization process. Its main purpose is to create a reduced representation, such as a hidden-layer image or production matrix, by reducing its dimensionality while considering underlying assumptions about the features present in the pooled sub-regions.

Various pooling techniques are used, including:

Average pooling, which calculates the average of all pixels in the selected region, as shown in Figure 1.18 (b).

Maximum pooling, which identifies the pixel with the highest value among all the pixels in the selected region, as shown in Figure 1.18 (a).



29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

100	184
12	45

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

36	80
12	15

Figure .18: *(a) Maximum pooling, (b) Average pooling.*

The pooling layer reduces the number of parameters and calculations in the network and avoids overfitting. Among the most commonly used are the Max-Pooling mentioned above and Average Pooling (Figure 1.18) [27].

In our work, we used Max-Pooling, which is more efficient than Average Pooling, as it maximizes the weight of strong activations.

1.7.5.3 Fully-Connected Layer

The fully connected layer is similar to the fully connected networks found in traditional models. It takes the output of the initial phase, which includes convolution and repeated pooling, and processes it through the fully connected layer. This process involves calculating the dot product between the weight vector and the input vector to produce the final output. As it shown in the Figure 1.19 [3].

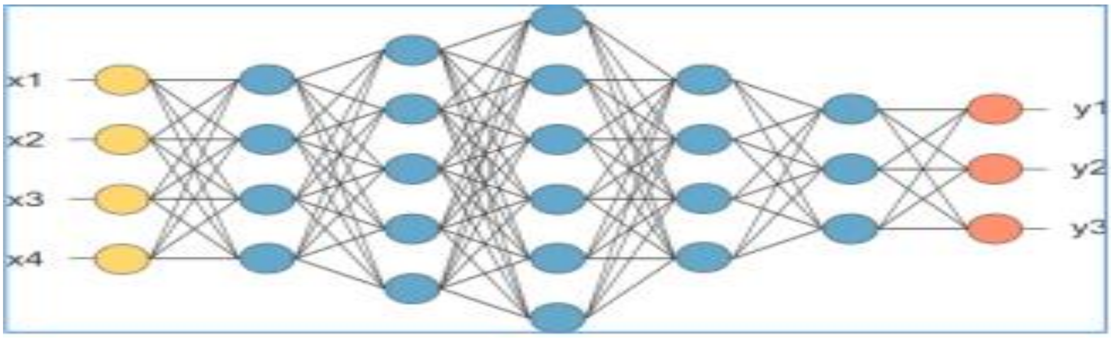


Figure 1.19: *Fully-connected layer.*

1.7.6 Long-Short Term Memory (LSTM) neural networks

LSTMs were initially introduced by Hochreiter and Schmid Huber in 1997. They emerged as a response to the quest for a novel model capable of addressing the limitations inherent in traditional recurrent neural networks (RNNs) [28]. As a result, LSTMs are considered a distinctive subset of conventional neural networks (RNNs) due to their exceptional capacity to retain and make sense of sequential data, essentially endowing the model with a temporal understanding.

The fundamental divergence between an LSTM and a basic RNN is evident in the constituents of the network's hidden layers. In an LSTM, every node consists of an assembly of cells entrusted with the task of preserving prior data streams. The uppermost line within each cell serves as a conduit for data transfer, conveying information from the past to the present. The cell's autonomy furnishes the model with a mechanism to sift and incorporate values from one cell to another. Concludingly, the gates that make up the sigmoidal neural network layer direct the cell towards an optimal state by either permitting or obstructing data flow. The overarching objective here is to govern the status of each cell, and this is achieved through gate control, as delineated as follows [28].

The forget gate is responsible for deciding what information is to be purged from the cell's state, rendering a value between 0 and 1, with 1 denoting "retain this entirely" and 0 signifying "discard this entirely."

The input gate selects which new data should be incorporated into the cell, sieving data incoming from previous cells.

The output gate determines the nature of the output produced by each cell. The output value hinges on the cell's current state, in conjunction with the most recent filtered and added data. The closer the value is to 0, the less information it conveys to subsequent cells.

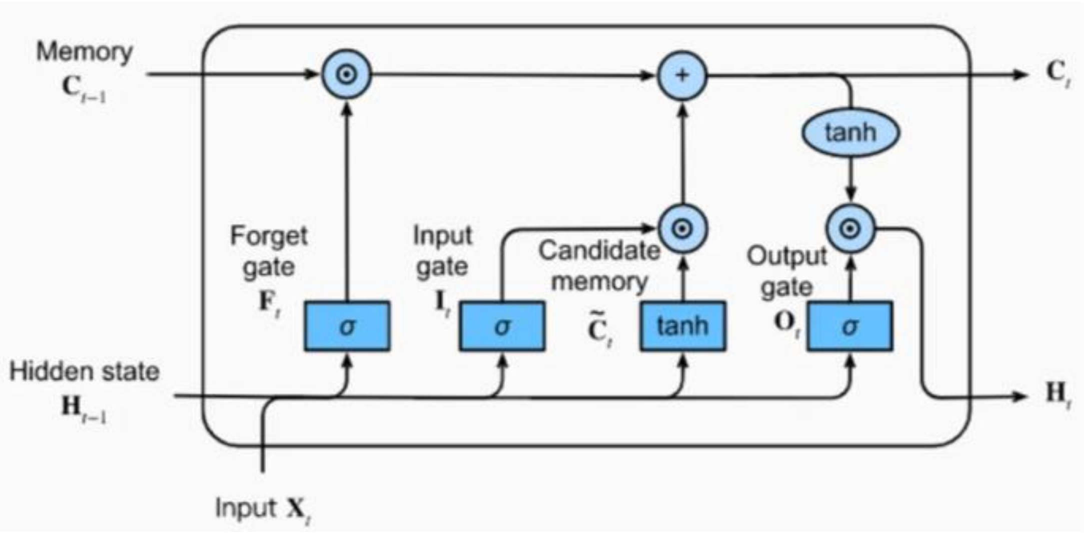


***Figure 1.20**: Architecture of a LSTM Unit.*

1.8 Conclusion

In conclusion, this chapter has introduced various aspects of machine learning and deep learning. At the beginning, we have discussed the artificial intelligence and its subnets which are machine learning and deep learning. In the machine learning we have explored the ML types, tasks and its algorithms.

In addition, we have presented deep learning, a subset of machine learning, and delved into its fundamental structure. We have explored various deep learning models, such as MLP, RNN, CNN, and LSTM architectures, each tailored for specific applications based on specific requirements.

In the next chapter, we will provide the Parkinson's disease, which is a neurodegenerative disease characterized by the destruction of a specific population of neurons, the diagnosis of the disease, and the medical detection of this last using AI (deep learning).

Chapter 2 Parkinson Disease

2.1 Introduction

Parkinson's disease (PD) is a neurodegenerative disorder that primarily affects older adults. Parkinson's disease impacts individuals' physical and functional abilities, limiting their mobility and capacity to perform daily activities. The diagnosis of PD remains essentially a clinical one, and it is important to recognize the early features together with symptoms and signs suggesting other causes of parkinsonism. There has also been a rapid expansion in the treatment options both in the early and the later stages of the illness together with a greater awareness of non-motor complications.

This chapter includes: the definition of Parkinson's disease (PD), its motor and non-motor symptoms, the stages of the disease, and methods for detecting PD through handwriting analysis, voice assessment, and MRI imaging.

2.2 Neurodegenerative diseases

Neurodegenerative diseases are a group of disorders characterized by the progressive degeneration of neurons in the central nervous system. Among the most well-known diseases are Alzheimer disease (AD), which is characterized by memory loss and cognitive decline, and Parkinson's disease.

Parkinson's disease (PD) is one of the age-related neurodegenerative conditions that leads to significant limitations in physical, cognitive, and functional abilities. PD is a persistent and advancing ailment resulting from the gradual loss of neurons in the substantia nigra, a region involved in producing dopamine neurotransmitters. These neurotransmitters have a vital role in regulating motor functions [29]. To date, there is still no curative treatment available for this disease. The figure 2.1 illustrates the contrast between a typical neuron and a Parkinson's disease affected neuron, which exhibits reduced dopamine levels.

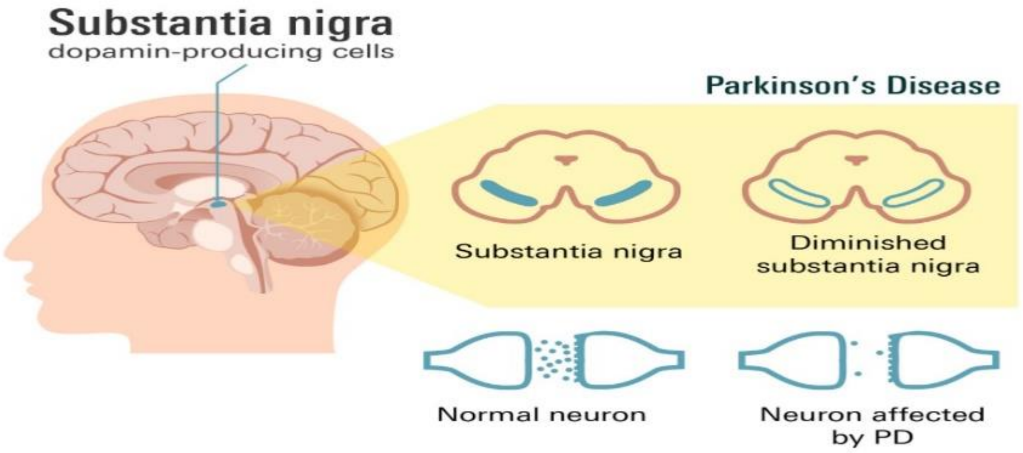


Figure 2.1: *Parkinson disease.*

2.3 Parkinson disease

Parkinson's disease is a neurodegenerative disorder first described by English physician James Parkinson in 1817, a member of the Royal College of Surgeons. He originally termed it "Shaking Palsy" and provided a detailed characterization involving involuntary tremors, reduced muscle strength in resting body parts, a tendency to bend forward, and shifting from walking to running pace while maintaining intact senses and intellect [30]; Parkinson's Disease is a chronic condition marked by the gradual loss of neurons in the substantia nigra, impacting dopamine production. PD is an extrapyramidal syndrome with evolving symptoms that notably affect motor functions and also impact autonomic, cognitive, and psycho-behavioral functions [31].

2.4 Clinical manifestations

The symptoms of Parkinson's disease are both motor and non-motor. Both types of symptoms have a major impact on the daily lives of Parkinson's sufferers.

2.4.1 Motor symptoms

- **The resting tremor:** The first perceptible sign of Parkinson's disease is resting tremor, which often begins as a sensation of internal vibration before becoming visible and obvious. Depending on the case, it may appear or be accentuated by stress, fatigue or intense mental concentration. In the early stages of the disease, the tremor only affects more often than just one side of the body, then extends to the opposite half of the body over time [32].

- **Akinesia:** Is prevalent in Parkinson's disease, refers to struggles in starting movements. It's closely linked to bradykinesia (slow movements) and hypokinesia (reduced movement size). Parkinsonian akinesia manifests as difficulty initiating walking, with symptoms like short steps, limited arm movement, stooped posture, and stiff neck. Other signs include reduced gestures during conversation, facial immobility, decreased blinking, and diminished emotional expression [33].
- **Rigidity:** Characterized by increased resistance to passive movement of the limbs, often accompanied by the "cogwheel" phenomenon, rigidity may occur in the neck, shoulders, hips, wrists, and ankles. These limbs maintain the posture at the end of movement, referred to as a "lead pipe" rigidity [32].

2.4.2 Non-motor symptoms

a) **Sleep disorders:** Is common nonmotor issues in PD, encompassing difficulties in falling asleep, frequent awakenings, nocturnal cramping, painful dystonia, motor symptoms at night, restlessness, Restless Legs Syndrome (RLS), nocturnal incontinence, confusion, hallucinations, and daytime sleepiness. Recognition of their clinical significance has grown in recent years, spurring new research efforts [32].

b) **Cognitive disorders and dementia:** Can develop in the early stages of the disease without worsening dementia. The main characteristic of this early cognitive syndrome is an impairment of executive function including disorders of memory, perception, slowed thinking, difficulty planning, organizing, and problem-solving behavior [34].

c) **Neurovegetative disorders or dysautonomia:** Neurovegetative problems impact over 50% of Parkinson's patients, stemming from lesions in the central and peripheral nervous systems. Functions such as cardiovascular, digestive, respiratory, and vesicourethral are most affected, leading to disability and reduced quality of life for both patients and caregivers. Additionally, these issues are often worsened by antiparkinsonian treatments [35].

d) **Psychiatric disorders:** Around 30 to 40% of Parkinson's patients experience depression, with factors like cognitive issues, sleep problems, fatigue, and functional limitations contributing to this risk. Anxiety, the most prevalent neuropsychiatric symptom after depression, affects 20 to 46% of individuals with Parkinson's, presenting as various forms including generalized anxiety, specific phobias, social anxiety, or sudden panic attacks [32].

2.5 Stages of Parkinson's disease

The speed at which the disease progresses varies greatly from patient to patient. This evolution can be divided into five phases: a presymptomatic phase, a diagnostic phase, a "honeymoon" phase, during which symptoms are significantly reduced by chemical treatment, a phase of motor complications, with the appearance of dyskinesias, a phase of decline, with worsening of axial and cognitive signs, and a major handicap in daily life.

According to [36], Parkinson's disease advances over time and spreads through different areas of the body in the following stages:

- **Stage 1:** At this initial stage of Parkinson's disease, symptoms are relatively mild and don't significantly disrupt daily life. While there may be subtle changes in posture, walking, or facial expressions, they're often overlooked or attributed to other factors. Typically, symptoms are confined to one side of the body, and individuals can still perform routine tasks without much difficulty.
- **Stage 2:** As Parkinson's progresses to stage 2, symptoms become more pronounced and noticeable. Stiffness, tremors, and changes in facial expressions become more evident. Tasks may take longer to complete due to muscle stiffness, but balance remains relatively unaffected at this point. However, individuals may start to notice alterations in their posture, and there's a gradual increase in difficulty with movement.
- **Stage 3:** Progressing to stage 3 signifies an escalation in the severity of Parkinson's symptoms. While many of the symptoms from stage 2 persist, individuals now experience a decline in balance and reflexes. Movements become slower and more deliberate, and falls become more frequent due to impaired coordination. Despite these challenges, individuals may still maintain some level of independence but may require assistance with certain tasks.
- **Stage 4:** It represents a further deterioration in motor function and independence. While individuals can still stand and walk with assistance, mobility becomes increasingly impaired, often necessitating the use of a walker or other assistive devices. Daily tasks become more challenging, and living alone may become impractical or unsafe due to the heightened risk of falls and difficulties with movement.
- **Stage 5:** In the advanced stage of Parkinson's disease, individuals experience severe motor impairments and a profound loss of independence. Stiffness in the legs may lead to freezing upon standing, making it nearly impossible to walk without assistance. Wheelchair dependence becomes necessary, and patients are highly susceptible to falls without constant supervision and assistance. Daily activities become exceedingly difficult, requiring around-

the-clock care to ensure safety and well-being.

The figure 2.2 summarizes the various stages of Parkinson's disease.

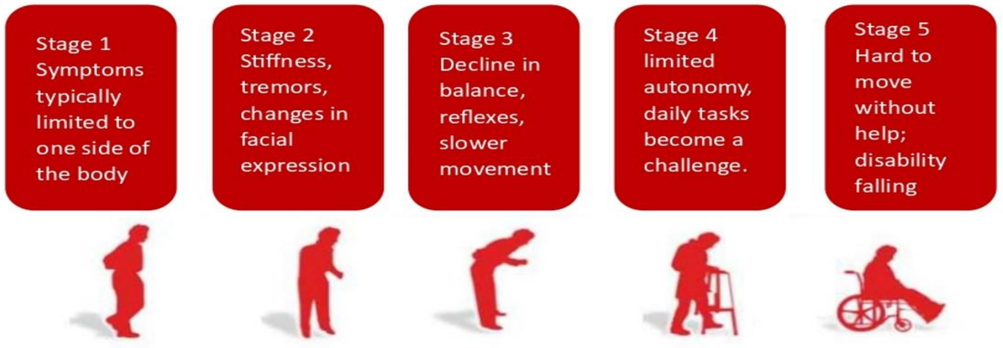

Stage 1
Symptoms typically limited to one side of the body
Stage 2
Stiffness, tremors, changes in facial expression
Stage 3
Decline in balance, reflexes, slower movement
Stage 4
limited autonomy, daily tasks become a challenge.
Stage 5
Hard to move without help; disability falling

Figure 2.2: *Parkinson disease stages.*

2.6 Parkinson detection methods

Early detection of Parkinson's disease plays an important role in effective management and treatment. Various screening methods and diagnostic tools are employed to identify PD in its initial stages.

2.6.1 Voice recognition

Speech requires the integrity and integration of numerous neurocognitive activities. These activities can be summarized as follows:

- ✓ When thoughts, feelings, and emotions generate intent to communicate verbally, they must be organized and converted into a code that abides by the rules of language. These combined activities are referred to as cognitive linguistic processes.
- ✓ Getting ready to speak involves arranging the words so that the muscles used for speech can say them correctly. This means picking the right words, putting them in the right order, and making sure the speech muscles move at the right times and with the right strength. All of these tasks together are called motor speech planning, programming, and control.

The speech disorders associated with PD are termed hypokinetic dysarthria and lead to reduced speech intelligibility. Dysarthria reflects neuromuscular disturbances of strength, speed, tone, steadiness, or accuracy of the movements that underlie the execution of speech. They also reflect disturbances at any or a combination of the major components of the speech mechanism, including respiration, phonation, and resonance [37].

In this paper [38], Loh et al, developed a 2D-CNN model for automated Parkinson's disease diagnosis, the model achieved remarkable accuracy, reaching 99.46% in multi-categorization through tenfold cross-validation. The dataset, obtained from OpenNeuro dataset, included EEG recordings from 16 healthy controls and 15 PD patients recorded both on and off dopaminergic medications.

2.6.2 Handwriting

PD leads to a disruption in the execution of practiced skills such as handwriting. Handwriting in Parkinson's disease often exhibits several types of dysfluencies, including a lack of control, sudden changes in direction, tremors, slowness, hesitations, rigidity, and variability in baseline. A specific component of the writing movements is generated by the fingers, wrist, and arm [39].

In their research [40], Mahima et al, focused on the early detection of Parkinson's disease (PD). through a logical analysis of time-series data obtained from a spiral drawing assessment test conducted on both Parkinson's patients and individuals without the condition, utilizing digital tablets. Initially, a machine learning approach is applied separately to static and dynamic drawing tests using logistic regression and Support Vector Machine classifier to assess accuracies. Their study introduces an innovative strategy involving a Restricted Boltzmann Machine (RBM)[1] coupled with a multi-layer perceptron model classifier. This combined approach achieves a noteworthy accuracy of 95.32% when considering both static and dynamic spiral drawing assessments.

The handwriting of individuals with Parkinson's disease is often small and cramped, also known as micrographia as it is shown it figure 2.3 [41].

[1] RBM is a Restricted Boltzmann Machine is a type of neural network used for unsupervised learning tasks, such as feature learning and dimensionality reduction.

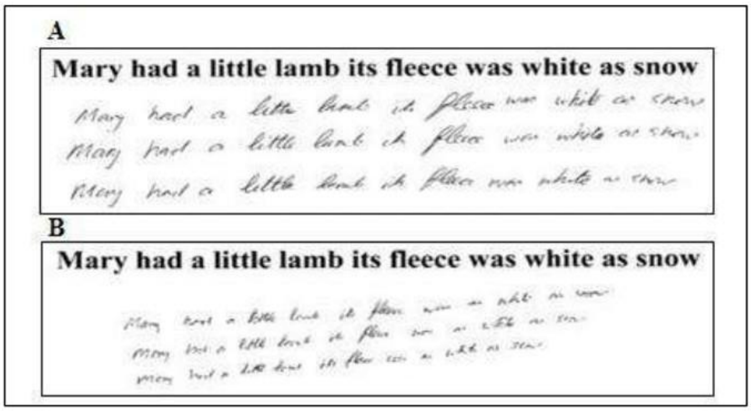

A
Mary had a little lamb its fleece was white as snow
B
Mary had a little lamb its fleece was white as snow

Figure 2.3: *Micrograph of a 58-year-old right-handed Parkinson's patient.*

A: under treatment, B: without treatment.

2.6.3 Radiological images

Medical imaging plays a pivotal role in the diagnosis and treatment of various illnesses by providing non-invasive visual representations of internal organs and tissues. These images aid in disease detection, treatment planning, and monitoring effectiveness. In the case of Parkinson's disease, radiological imaging techniques like Computed Tomography (CT) scans and Magnetic Resonance Imaging (MRI) are instrumental in swiftly and accurately diagnosing the condition. CT scans offer detailed images, enabling the identification of specific disease indicators, while MRIs provide a comprehensive view of the brain structures, facilitating the detection of subtle abnormalities associated with Parkinson's, such as changes in dopamine density [42].

2.7 The physical means of the various processes used in medical imaging

Medical imaging refers to the methods and technologies used to generate visual depictions of the interior of the body. These visual representations, termed images, are utilized for diagnosing, monitoring, or treating a variety of medical conditions. Medical imaging enables the examination and visualization of different bodily structures.

2.7.1 Scanner: CT stands as one of the primary medical imaging techniques. It involves generating a three-dimensional reconstruction of tissues from a tomographic analysis obtained by subjecting the patient to X-ray beam scanning. The principle of computed tomography revolves around measuring the spatial distribution of X-ray attenuation, examined from various angles during rotation around the object (figure 2.4), and reconstructing images from these angular projections [43].

Figure 2.4: *CT Scan.*

2.7.2 MRI: Is a noninvasive medical imaging technique that uses a magnetic field and radio waves to capture the internal structure of the organs in the body. The subject is placed in a magnetic field that polarizes the atoms in his body. A radio wave pulse sequence is then sent to the areas of interest, exciting the protons and breaking them from the alignment of the field. Once the pulse stops the protons realign by emitting energy, in the form of a radio signal, which is captured by sensors. Because of the varying speeds that protons realign in different tissues types, a distinct signal is sent from each. These signals are then captured by a detector and meticulously processed by a computer, culminating in the creation of detailed representations of the body's internal anatomy (figure 2.5) [44].

Figure 2.5: *MRI Machine.*

2.8 Diagnosis of Parkinson’s Disease using medical imaging

Many people are diagnosed with Parkinson's disease at later stages when it's harder to treat because of the lack of medical labs. When people lose a lot of dopamine, doctors start looking into a PD diagnosis. It's really tough to accurately detect PD, and if someone with PD is mistakenly thought to be healthy, it can cause big problems, even leading to coma. Using image processing techniques, PD can be detected earlier. There are a couple of common imaging methods. However, MRI scans can also spot signs of PD. Scientists are using machine learning methods like decision trees, support vector machines, and artificial neural networks to analyze medical images and detect PD early.

In this section, we review some of the existing machine learning and deep learning techniques that have already been used to diagnose Parkinson's disease:

- In the study of Pir Masoom [45], utilized a dataset sourced from the Parkinson's Progression Markers Initiative (PPMI) for their research. This dataset comprised 250 MRI scans of individuals diagnosed with Parkinson's disease (PD) alongside 250 scans of healthy controls (HC). To analyze these MRI scans, the researchers implemented a neural network architecture featuring three convolutional layers (CNN). Their proposed network achieved an accuracy rate of 96%, indicating its effectiveness in accurately distinguishing between PD patients and healthy individuals based on MRI data.

- In their research Soheil Esmaeilzadeh et al [46], proposed a Parkinson's disease detection method using 3D Convolutional Neural Network 3D-CNN and deep learning techniques on MR images. The dataset, sourced from the PPMI dataset, included 452 PD-affected patients and 204 healthy individuals. Data processing involved skull-stripping using Brain Extraction Technique (BET)[2] with Statistical Parametric Mapping (SPM)[3] during pre-processing, enhancing accuracy and speed. Data augmentation included flipping MR images, doubling the dataset size. The 3D CNN model, with padding and max-pooling layers, achieved a validation accuracy of 60% on the split datasets.

- In the study of Ortiz et al [47], proposed a method for identifying Parkinson's disease detection, their approach involved employing CNN architectures such as LeNet-5 and

[2] BET Brain Extraction Tool is a software tool used as a preprocessing step in neuroimaging to extract brain tissue from MRI or CT scans.

[3] SPM Statistical Parametric Mapping is a software package in neuroimaging used for the analysis of brain imaging data.

AlexNet (existing CNN models). The study utilized the PPMI dataset. The dataset comprised 158 subjects suffering from PD and 111 normal controls (NC), enabling a thorough investigation into the effectiveness of their proposed method in distinguishing between PD patients and healthy individuals.

- In their study, Wenzel et al [48] utilized the PPMI dataset, consisting of 207 healthy controls and 438 patients diagnosed with PD. Their research focused on leveraging CNNs in conjunction with an ImageNet-based transfer learning model. This approach enabled the extraction of intricate features from medical imaging data, facilitating the identification and classification of neurological conditions. They achieved an accuracy of 97%.

- The authors El Maachi et al in [49] introduced a novel approach using a 1D convolutional neural network (1D-Convnet) to assess gait data for Parkinson's disease. The framework processes 18 parallel 1D data streams from foot sensors measuring vertical ground response force. Their model achieved an impressive 98.7% accuracy in detecting Parkinson's disease, showcasing the effectiveness of deep learning in gait analysis. The extended work proposed an intelligent Parkinson detection system, not only identifying the disease but also predicting severity with 85.3% accuracy based on the Unified Parkinson's Disease Rating Scale (UPDRS), demonstrating state-of-the-art performance in gait recognition for Parkinson's.

- The paper conducted by Sabyasachi Chakraborty [50], the PPMI dataset was utilized. This dataset consisted of 203 individuals classified as healthy controls and an equal number suffering from Parkinson's Disease. Employing a three-dimensional convolutional neural network (3D), their research achieved an accuracy rate of 95.29%.

- The paper of Balaji E et al [51], presents a new deep learning architecture using an LSTM network to assess Parkinson's disease severity based on gait patterns, eliminating the need for manual feature crafting. The LSTM captures long-term dependencies in the gait cycle, addressing the vanishing gradient problem by incorporating memory blocks. The Adam-optimized LSTM achieves 98.6% accuracy.

- In the study of Vyas et al [52], they explored the identification of potential biomarkers for Parkinson's disease progression using magnetic resonance imaging (MRI) brain images. They employed 2D and 3D Convolutional Neural Networks (CNNs) trained on axial plane MRI scans from the PPMI dataset. Pre-processing techniques, including bias field correction and histogram matching, were crucial for accurate model training. The 3D CNN model outperformed the 2D model, exhibiting an 88.9% accuracy and a 0.86 area under the curve (AUC), highlighting its superior performance in classifying test data compared to the 2D model with a less favorable 72.22% accuracy and 0.50 AUC.

- In their research of Kollia et al [53], the authors present a diagnostic method utilizing trained Deep Convolutional or Convolutional-Recurrent Neural Networks (DNNs). This approach integrates transfer learning, k-means clustering, and k-Nearest Neighbor classification of DNN representations to improve disease prediction from MRI and/or DaT Scan data in the NTUA dataset, their research achieved an accuracy of 94%.

- In the study of BASNIN et al [54], a DenseNet combined with Long Short-Term Memory (LSTM) is applied to MRI data samples from the National Technical University of Athens (NTUA) dataset. DenseNet improves feature selection by accounting for the temporal proximity of images in each layer, and the LSTM layer processes the output to uncover important dependencies in temporal features, their research achieved an accuracy of 94%.

- The authors Sukanya Pechetti et al [55], they utilized the optimized MobileNet V3, coupled with the Improved Dwarf Mongoose Optimization algorithm (IDMO), to analyze MRI data for the identification of Parkinson’s disease classes. To enhance feature extraction from preprocessed images, they introduced a novel Pyramid Channel-based Feature Attention Network (PCFAN) employing a multi-stage design with attention blocks at each stage. their experimental work involved the utilization of the PPMI and NTUA datasets. The suggested methodology outperformed existing systems, demonstrating superior accuracy achieving 95%.

- In 2023 the study of Aditi Govindu et al [56], they focused on addressing challenges related to Parkinson's disease (PD) in an aging population, highlighting the importance of early and remote detection. Utilizing machine learning in telemedicine, the study analyzed audio data from 30 PD patients and healthy individuals, focusing on attributes like jitter, shimmer, and MDVP of vowel sounds. Four machine learning (Random Forest, KNN, Logistic Regression Model, Yield Random Forest Classifier) models were trained on 75% of the PPMI dataset, with the Random Forest classifier proving most effective, achieving a high detection accuracy of 91.83% and a sensitivity of 0.95. The methodology involved comprehensive data preprocessing, analysis, and visualization for enhanced attribute comprehension, and the findings underscore the potential of machine learning in telemedicine for early PD detection, offering the promise of improved quality of life for those affected.

Reference	Year	Dataset	Algorithm	Accuracy
SHAH et al. **[45]**	2018	PPMI (250 PD, 250 HC)	3D CNN	96%
ESMAEILZADEH et al. **[46]**	2018	PPMI (452 PD, 204 HC)	3D CNN	60%
ORTIZ et al. **[47]**	2019	PPMI (158 PD, 111 NC)	Lenet 5 and AlexNet	95.1%
Kollia et al. **[53]**	2019	NTUA (55 PD, 23 HC)	CNN CNN-RNN	94% 98%
WENZEL et al. **[48]**	2019	PPMI (438 PD, 207 HC)	CNN	97%
EL MAACHI et al. **[49]**	2020	PD dataset (95 PD, 73 HC)	DNN (1D-Convnet)	98.7%
CHAKRABORTY et al. **[50]**	2020	PPMI (203 PD, 203 HC)	3D CNN	95.29%
Balaji E et al. **[51]**	2021	-	LSTM	98.6%
BASNIN et al. **[54]**	2021	NTUA	DenseNet-LSTM	90%
VYAS et al. **[52]**	2022	PPMI (250 PD, 250 HC)	2D and 3D CNN	88.9%
PECHETTI et al. **[55]**	2023	PPMI and NTUA	Optimized MobileNetV3	95%
GOVINDU et al. **[56]**	2023	PPMI (23 PD ,31 HC)	Random Forest Classifier	91.83%

Table 2.1: *Synthesis of existing approach.*

This table (Table 2.1) synthesizes prior research from 2018 to 2023 on diagnosing Parkinson's Disease using various machine learning algorithms. It includes details on the datasets, algorithms, and achieved accuracies. Notably, the Parkinson's Progression Markers Initiative (PPMI) and National Technical University of Athens (NTUA) datasets are frequently used, indicating their significance in this field. The algorithms range from 3D Convolutional Neural Networks (CNNs) to Long Short-Term Memory (LSTM) networks and DenseNet-LSTM hybrids, with accuracies spanning from 60% to 98.7%. The variation in accuracy underscores the ongoing challenges and advancements in developing effective diagnostic models for Parkinson's Disease.

2.9 Conclusion

In conclusion, this chapter has provided an exploration of Parkinson's disease, encompassing both its motor and non-motor symptoms. We have examined the various stages in the progression of PD and explored various methods used for its detection. In addition, we discussed in detail previous research efforts aimed at diagnosing PD, including the database, algorithm and accuracy. The next chapter will be focused on the pretrained model architectures.

Chapter 3 Transfer Learning

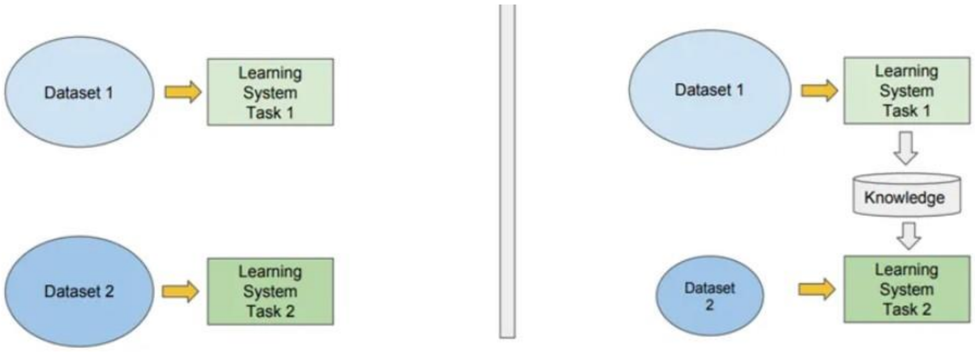
3.1 Introduction

In this chapter, we will look at how transfer learning in machine learning is strengthened by the ImageNet dataset. We will explore different advanced neural network designs like VGG, ResNet, DenseNet, and MobileNet, which were trained using ImageNet. We will also talk about how to tweak these pre-trained models for specific tasks by adjusting parameters like batch size, epochs, and learning rate. Finally, we will discuss how to measure the performance of these models using metrics like accuracy, precision, recall, F1 score, and confusion matrices.

3.2 Transfer learning

The Transfer Learning method is a widely used technique in Machine Learning, particularly adept at handling large and demanding datasets for training Deep Learning models. Transfer learning involves training a model on a task in one domain and then applying it to a related domain and task. The original training domain is referred to as the source domain, while the new domain is called the target domain. This deep learning technique allows for rapid and accurate training of a Convolutional Neural Network (CNN) by initializing its weights from another CNN that was previously trained on a larger dataset, such as the popular ImageNet dataset [57],

Transfer learning is defined as "enhancing learning in a new task by leveraging knowledge from a related task that has already been mastered". Humans possess a unique capability to transfer knowledge between different individuals and various tasks. They can reuse the knowledge acquired from solving a previous problem to address a new problem more effectively. The closer the new problem is to previous learning experiences, the more efficiently the transfer can be carried out [58]; figure 3.1 can explain the difference between using a) the traditional learning and b) the transfer learning.



a) Traditional learning. *b) Transfer learning.*

Figure 3.1: *Transfer Learning.*

3.3 ImageNet Dataset

ImageNet is a dataset containing over 15 million labeled high-resolution images across approximately 22,000 categories. These images were sourced from the web and annotated by human labelers using Amazon’s Mechanical Turk crowd-sourcing tool. Since 2010, the annual ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held as part of the Pascal Visual Object Challenge. ILSVRC utilizes a subset of ImageNet, featuring around 1,000 images for each of the 1,000 categories. Overall, this subset comprises roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images [59].

It is used to train a system for a specific task or assignment. Also, ImageNet helps train the initial layers of the system to identify generalizable features from larger datasets [60].

3.4 Pretrained models

In this work, five pre-trained CNN models were utilized: VGG16, VGG19, ResNet50, DenseNet121, and MobileNet. These models were chosen because they are designed for images with a size of 224x224 pixels. The selected models are pre-trained on a large-scale dataset called ImageNet, which primarily includes classes of animals and everyday objects.

3.4.1 MobileNet

MobileNet is a type of convolutional neural network designed for mobile and embedded vision applications. As the first mobile computer vision model from TensorFlow, MobileNet features a simplified architecture that uses depthwise separable convolutions to create lightweight deep neural networks. These networks are optimized for low latency on mobile and embedded devices. MobileNet employs an approach where one layer performs filtering and another creates the output, significantly reducing computation time. MobileNet's efficiency is further enhanced by introducing adjustable parameters such as the width and resolution multipliers. The base configuration of MobileNetV2 typically encompasses 54 layers, although variations in layer count may occur based on specific hyperparameters like width and resolution factors [61]; the figure 3.2 below illustrates the MobileNet architecture.

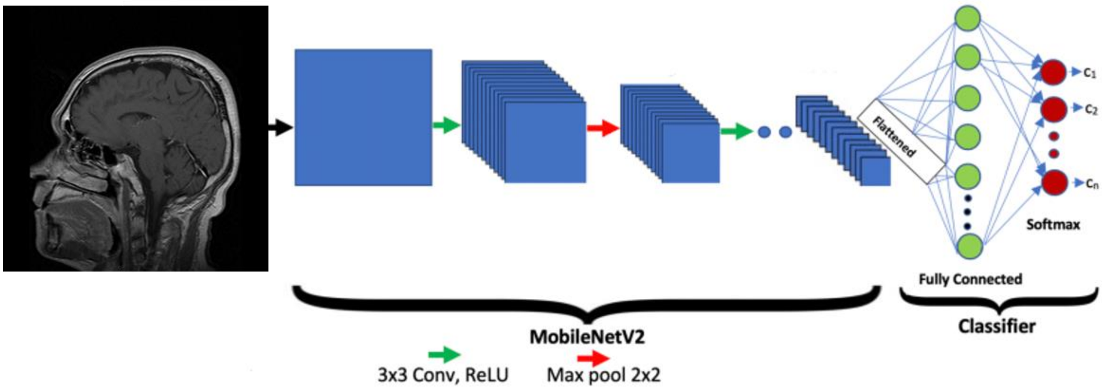


Figure 3.2: *MobileNetV2 architecture.*

3.4.2 VGG

The VGG network is a deep neural network introduced by Karen Simonyan and Andrew Zisserman in their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". They aimed to explore the effect of network depth on accuracy for large-scale image datasets and discovered that increasing the network's depth substantially improves accuracy. This was achieved by using more, but very small (3x3) convolution filters. The VGG architecture comes in multiple sizes, with the number of layers ranging from 13 to 19 [62].

3.4.2.1 VGG 16

The VGG16 model, developed by Simonyan and Zisserman, is a convolutional neural network known for its depth and simplicity. It consists of 16 layers that directly contribute to learning, including 13 convolutional layers and 3 fully connected layers. The architecture employs small (3x3) convolution filters throughout, which significantly contribute to its success in various image classification tasks. VGG16 takes a 224 × 224 × 3 image as input [62]. VGG16 consists of five blocks of convolutional and pooling layers. The first two blocks each have two convolutional layers followed by a max pooling layer, while the next three blocks each have three convolutional layers followed by a max pooling layer. This results in a total of 13 convolutional layers and 16 hidden layers.

3.4.2.2 VGG 19

The VGG19 model is very similar to the VGG16, except that VGG19 includes three additional convolutional layers. Specifically, VGG19 consists of five blocks of convolutional and pooling layers. The first two blocks each have two convolutional layers followed by a max pooling layer, while the next three blocks each have four convolutional layers followed by a max pooling layer. This results in a total of 16 convolutional layers and 19 layers in total, including the fully connected layers [63].

The figure 3.3 shows the architecture of the two previous models VGG16 and VGG19.

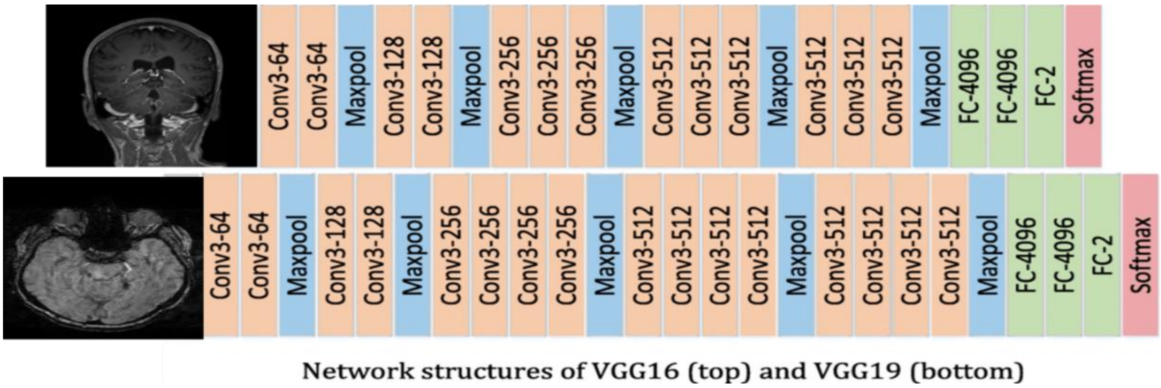


Figure 3.3: *VGG16 and VGG19 architecture.*

3.4.3 ResNet

ResNet, short for Residual Network, is a well-known deep learning model first introduced in 2015 by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang in the study titled "Deep Residual Learning for Image Recognition." Presented by Microsoft Research Asia, this model addresses the issue of performance degradation that occurs as the depth of neural networks increases. While increasing the number of layers initially improves accuracy, it eventually leads to a decline. To tackle this problem, the researchers developed a residual learning framework that incorporates "skip connections" and extensive batch normalization which is illustrated in figure 3.4, enabling the training of hundreds of layers without sacrificing speed. The network has an input image size of 224x224 [64].

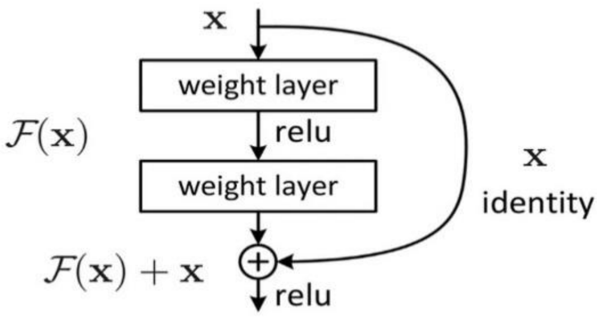

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 3.4: *ResNet model.*

The first thing we notice in the diagram above is the presence of a direct link that skips several levels of the model. This "skip connection," as it is called, is at the core of the residual blocks. Because of this skip connection, the output is different.

3.4.3.1 ResNet50

ResNet50 is a specific variant of this model, consisting of 50 layers, including 48 convolutional layers, one MaxPool layer, and one Average Pool layer (figure 3.5), with a total of 26 million parameters. This architecture has significantly contributed to the use of very deep neural networks by reducing gradient loss in the deepest layers through the addition of residual connections between each convolutional layer [65].

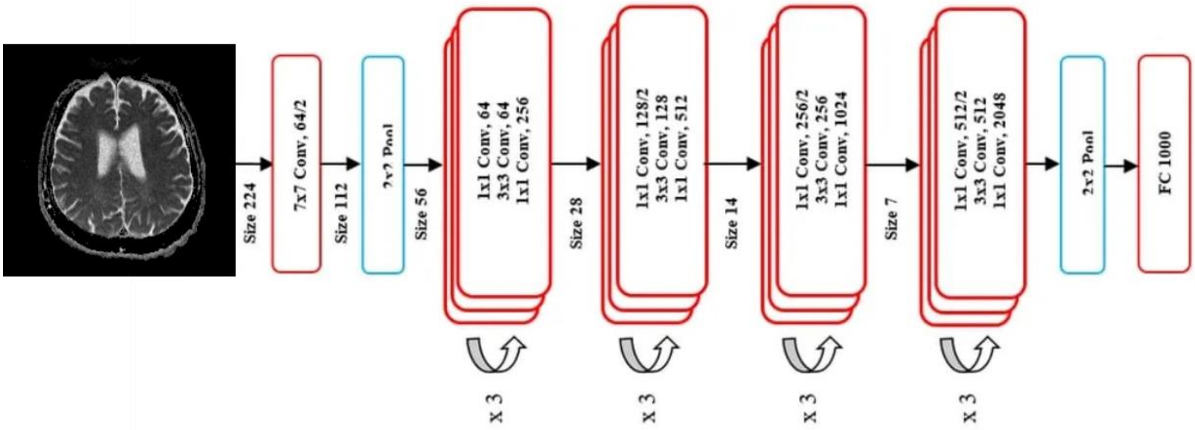


Figure 3.5: *ResNet50 architecture.*

3.4.4 DenseNet 121

DenseNet, introduced by Gao Huang and team in 2017, features densely connected CNN layers where each layer's outputs are linked to all subsequent layers in a dense block. This dense connectivity reduces network parameters, boosting feature reuse efficiency. DenseNet includes dense blocks and transition blocks between adjacent dense blocks. Each convolutional layer within a dense block receives input from the concatenation of the block's global input and the feature maps of preceding layers within the block. DenseNet leverages residual layers, with each layer receiving input from previous layers and passing feature maps to subsequent layers. Figure 3.6 explain the architecture of the DenseNet121 model, that have varying numbers of layers, such as 117 convolution layers, 3 transition layers, and 1 classification layer.

DenseNet has achieved excellent performance, while also utilizing less memory and processing power than other state-of-the-art techniques [66].

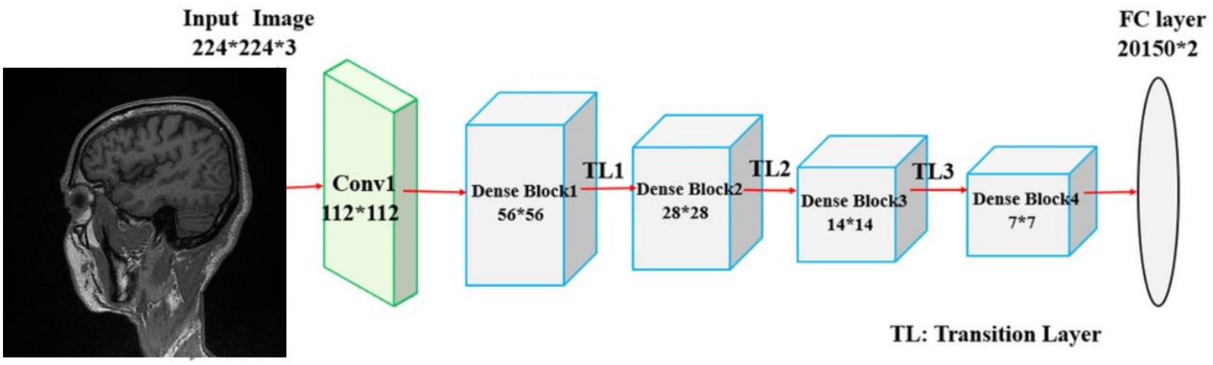


Figure 3.6: *DenseNet121 architecture.*

3.5 Train parameters

In the training process of learning models, defining the training parameters plays an important role in the model's performance and convergence. The key parameters in this regard are the "batch size", the number of iterations, commonly referred to as "epochs" and the learning rate; according to [67] these last are defined.

3.5.1 Batch-size

Training batch size is a hyperparameter that controls the number of training samples used in each iteration before updating the model's internal parameters. We chose a batch size of 32, meaning each batch processes 32 images at a time. The batch size represents the number of samples fed into the neural network inputs in a single step.

3.5.2 Epochs

An epoch is a hyperparameter that defines the number of times the entire dataset is passed through the neural network during the training process. Essentially, an epoch consists of one forward pass and one backward pass of all the training examples. In our case, we used a constant value of 50 epochs. This means the learning algorithm is applied to the whole dataset 50 times, allowing the model to learn and adjust its parameters iteratively. Each epoch represents a complete cycle of passing all the samples in the dataset through the neural network.

3.5.3 Learning rate

The learning rate, a crucial parameter in neural network optimization, determines the size of adjustments made at each iteration during model training. It represents the magnitude of the change applied to the network weights in response to the error estimation.

3.6 Model evaluation metrics

When assessing the performance of machine learning models, various evaluation metrics are utilized to determine their ability to accurately capture patterns. The selection of these metrics depends on the specific task and characteristics of the data.

3.6.1 Loss function

The loss function, also called the cost function, measures the probabilities or uncertainty of a prediction based on the difference between the prediction and the actual value. The loss is not expressed as a percentage but as a sum of the errors made for each sample in the training or validation sets. During training, the goal is to minimize this value. In neural networks, we aim to reduce the error. Loss functions, which are differentiable functions, guide the learning process of a neural network. To train a CNN for an object detection task, the networks are generally optimized for both classification and regression components. One of the most commonly used loss functions is the Softmax loss function, typically employed in the last layer of the neural network [68].

3.6.2 ADAM optimizer

Adam (Adaptive Moment Estimation) is an optimizer that incorporates information about the learning rate, L2 regularization factor, and mini-batch sizes. This parameter is based on the estimation of adaptive moments for optimizing the learning rate.

Additionally, Adam includes bias correction terms to account for the initializations of the first and second moments, making it particularly effective for optimizing deep learning models with large datasets [69].

3.6.3 The Confusion Matrix

A confusion matrix, or error matrix, is a table that displays the number of correct and incorrect predictions made by a model compared to the actual classifications in a dataset. This matrix provides insight into the confusions (errors) made by the model. It summarizes the performance of a classification model on test data where the true values are known. The confusion matrix is a key tool in understanding the errors made by the classifier and, more importantly, the types of errors that occur. It breaks down the prediction results by each class, highlighting not only the accuracy of the model but also the specific areas where the model's predictions are incorrect [70].

True Positive (TP): Number of normal data correctly classified as normal.

False Positive (FP): Number of abnormal data incorrectly classified as normal.

True Negative (TN): Number of abnormal data correctly classified as abnormal.

False Negative (FN): Number of normal data incorrectly classified as abnormal.

3.6.4 Accuracy

Accuracy is a fundamental metric used to evaluate the performance of a model. It is calculated by dividing the number of correct predictions by the total number of predictions made. This metric is easy to compute and intuitive to understand, making it a popular choice for assessing classification models. It provides a straightforward measure to determine if a model or algorithm is properly trained and performing its classifications correctly [71]; it is measured using this formula:

$$Accuracy = \frac{(\mathrm{TP + TN})}{(\mathrm{TP + TN + FP + FN})} \tag{3.1}$$

3.6.5 Precision

Precision is the number of correct positive results divided by the number of positive results predicted by the classifier. This metric should be as high as possible, as it indicates the accuracy of the positive class predictions, showing how likely a positive class prediction is to be correct. It defines the ratio of true positives to the sum of true positives and false positives [72]; such as:

$$Precision = \frac{\mathrm{TP}}{(\mathrm{TP + FP})} \tag{3.2}$$

3.6.6 Recall

Recall represents the ratio between the number of true positive examples correctly classified and the total number of positive examples. Sensitivity, or recall, is the percentage of positive instances that are correctly identified. This metric estimates a model's ability to accurately classify all positive cases [72].

$$Recal = \frac{\mathrm{TP}}{(\mathrm{TP + FN})} \tag{3.3}$$

3.6.7 F1 score

The F1 score is an essential parameter for measuring the accuracy of a classification system. It evaluates the effectiveness of two key metrics: precision and recall. The F1 score ranges between 0 and 1 and provides an overall estimate of a model's performance. It is defined as the harmonic mean of precision and recall, offering a balanced measure that accounts for both false positives and false negatives [70]. The F1 score can be expressed as follows:

$$F1 - score = \frac{2*(\mathrm{Recall}*\ \mathrm{Precision})}{(\mathrm{Recall}+\ \mathrm{Precision})} \tag{3.4}$$

3.7 Conclusion

In conclusion, this chapter highlighted how useful transfer learning is in machine learning, different powerful neural networks were explored like VGG, ResNet, DenseNet, and MobileNet, all trained on ImageNet dataset.

In the next chapter will delve into the implementation phase, where we apply these pre-trained models and the proposed CNN model to practical problems.

Chapter 4 Implementation and Results

4.1 Introduction

In this chapter, we explore the different platforms and tools utilized for implementing our networks, including Google Colab, Google Drive, and associated libraries. We also discuss the database, data augmentation techniques, and preprocessing methods employed. Furthermore, we outline the models utilized and summarize the results obtained.

4.2 Working environment

The working environment is based on two parts: hardware and software.

4.2.1 The hardware parts

The hardware that has been used is an HP computer with the characteristics below:

Processor	**I7-8550U CPU @1.80GHz**
RAM	8Go
Hard disk	1To HDD
Operating system	Windows 10 x64
Graphics card	Intel® UHD Graphics 620/ NVIDIA GeForce MX150

4.2.2 The software parts

The software that has been used is Google Colab and Google Drive.

Google Colab, short for Google Collaboratory, is a free platform provided by Google that enables users to write and execute Python code.

Google Drive is a cloud-based file storage service provided by Google, allowing users to save their files online, access them from anywhere, and share them with others.

4.2.3 Libraries

The libraries that have been used in the code are:

- **TensorFlow**: is a toolkit for solving complex mathematical problems. In other words, it is a Python library used for building, developing, training, and deploying machine learning and deep learning models.
- **Keras**: is a high-level library for machine learning, enabling quick creation of neural

network models in a few lines of code.

- **Pandas**: is a widely used Python library that excels in organizing data, making it easy to use and understand. It's primarily employed for data manipulation and analysis tasks. Pandas provides features for reading and writing data from CSV and Excel formats.
- **NumPy**: is a library dedicated to scientific and mathematical computing, handling data in multidimensional arrays or matrices.
- **Matplotlib**: is a library in Python used for generating visual representations such as images and graphs, often with interactive features. It's commonly employed for visually presenting results and for comparing images.
- **Gradio**: is a powerful Python library that allows for the creation of interactive graphical user interfaces easily.

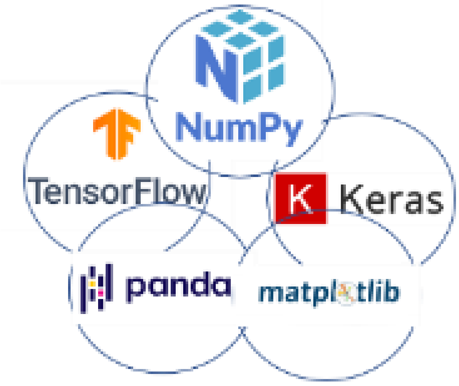


Figure 4.1: *Library's Logo.*

4.3 Dataset description

In this study, the NTUA[4] (National Technical University of Athens) database has been utilized, a publicly accessible dataset containing MRI images in PNG format with varying dimensions. The dataset consists of 43,087 images, categorized into 32,706 PD and 10,381 HC instances, representing two classes: PD Patients and Non-PD Patients.

Specifically, the T1 MRI images have been used, with 410 entries for PD patients and 390 entries for Non-PD individuals [54]. According to [73], T1 MRI images are more accurate because they highlight anatomical details and provide excellent contrast between gray and white matter. This enhanced visual clarity improves the demonstration of brain atrophy patterns, which is crucial for diagnosing and understanding the disease. The following figures illustrate the images of the database between PD and Non-PD patients (figure 4.2), and the figure 4.3

[4] NTUA: https://github.com/ails-lab/ntua-parkinson-dataset

illustrates the dataset distribution.

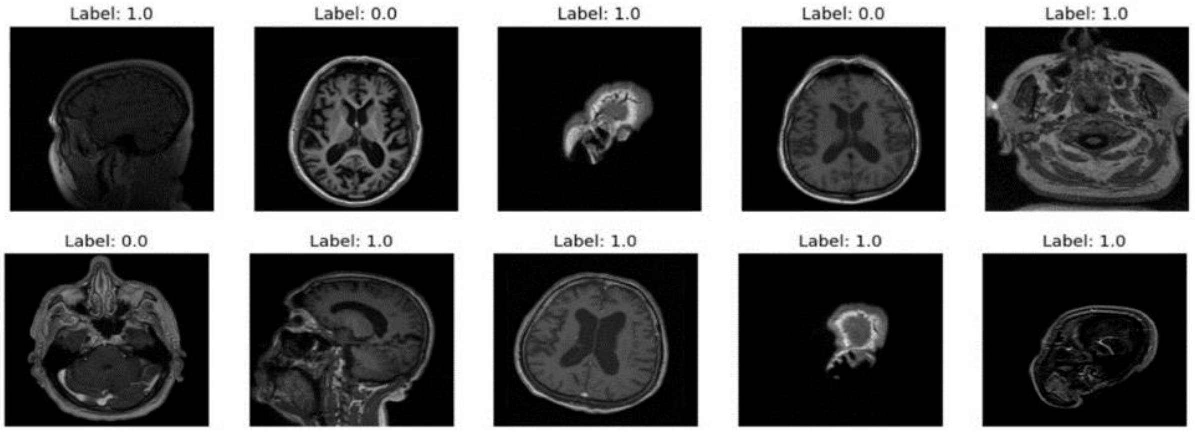


Figure 4.2: *NTUA images.*

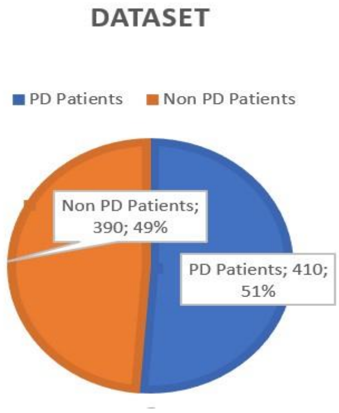


Figure 4.3: *Dataset distribution.*

4.4 Architecture of our system

In this section, we will outline the different steps used in our study. Before transmitting the dataset to our models, we must initially undertake several data loading, preprocessing, and data augmentation steps to streamline the training process. The following diagram (figure 4.4) outlines the steps followed before obtaining our results.

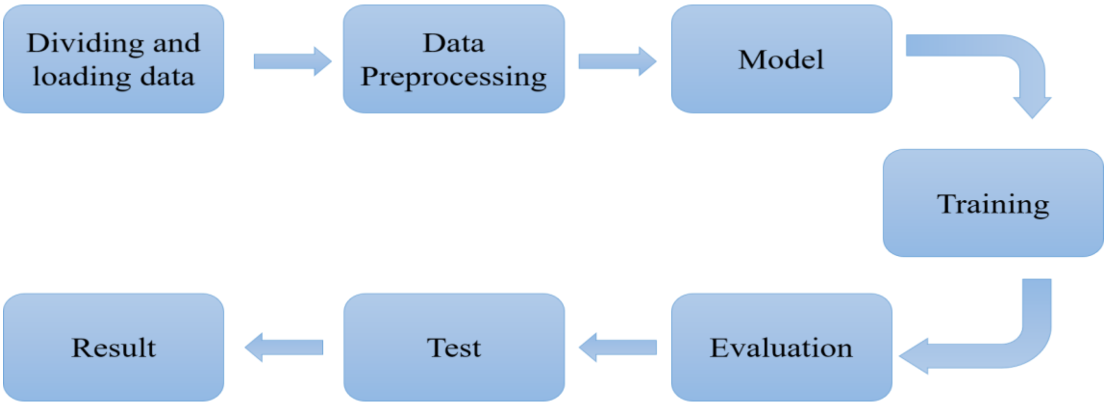
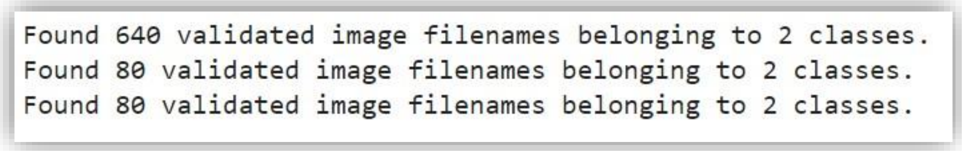


Figure 4.4: *Architecture model.*

4.5Dividing and loading data

In this database, the data has been partitioned such that 80% was designated for the training set, 10% for validation, and the remaining 10% for testing. Afterward, this data was organized into CSV files to facilitate exploration and analysis. These files contain the file paths of images along with their corresponding labels. Specifically, the training set consists of 640 entries, the validation set comprises 80 entries, and the test set encompasses 80 entries.



```
Found 640 validated image filenames belonging to 2 classes.
Found 80 validated image filenames belonging to 2 classes.
Found 80 validated image filenames belonging to 2 classes.
```

Figure 4.5: *Number of images for each dataset.*

In terms of label distribution, each set exhibits a consistent pattern. For the training data, there are 312 entries labeled as PD Patients and 328 entries labeled as Non-PD Patients. Similarly, in the validation set, there are 41 PD patients and 39 Non-PD patients. Lastly, the test set consists of 41 PD patients and 39 Non-PD patients; the following graphs (figure 4.6) show the distribution of each dataframe.

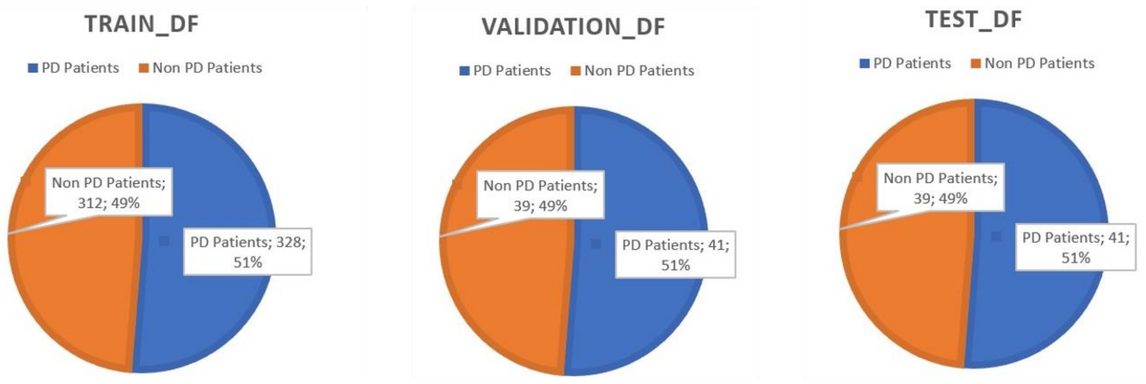


Figure 4.6: *Train, validation and test df.*

4.6Data Preprocessing

The preprocessing applied in this study involves resizing and data normalization:

4.6.1 Resizing: Since the sizes of images in the dataset vary, we addressed this issue by resizing them to a standard dimension of 224x224.

4.6.2 Normalization: The pixel values are normalized by dividing them by 255, ensuring that the pixel intensities ranged from 0 to 1.

These preprocessing steps was crucial to make the images compatible with the models used in the study; the following figure illustrates the different images for each dataframe.

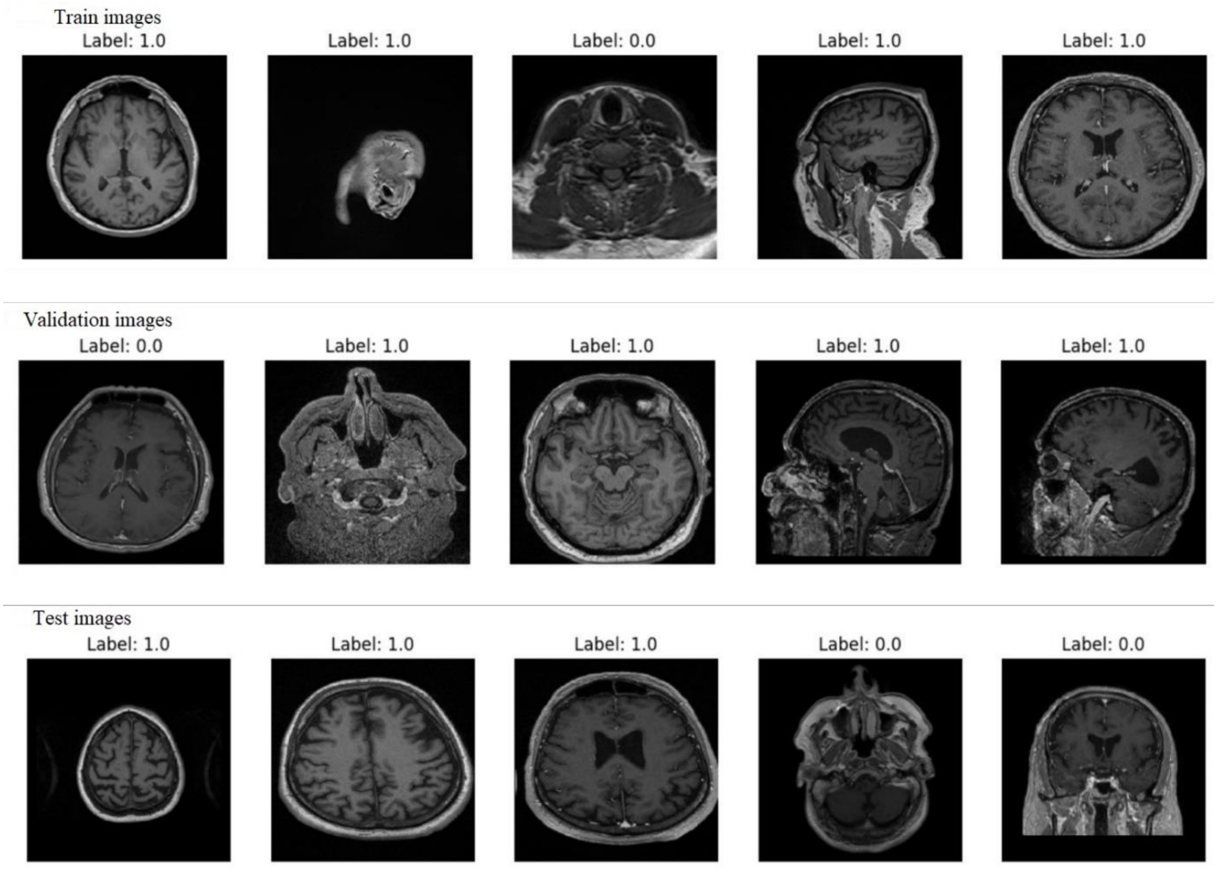


Figure 4.7: *the dataset after preprocessing.*

4.6.3 Data augmentation

The data augmentation technique performed in Keras using the DataGenerator class allows for adding new photos to the dataset based on existing ones after undergoing transformations without losing the random characteristics of each image, such as rotation_range, zoom_range, width_shift_range, height_shift_range, horizontal_flip.

4.6.3.1 Rotation_range: This parameter controls the range of rotation for images during data augmentation, allowing them to be rotated slightly left or right to create different viewing angles.

4.6.3.2 Zoom_range: This parameter specifies that images can be slightly enlarged or reduced, enabling the model to learn to recognize objects at different sizes.

4.6.3.3 Width_shift_range and **height_shift_range**: These parameters indicate that images can be slightly shifted horizontally or vertically, mimicking variations in the position of objects within the image.

4.6.3.4 Horizontal_flip: This parameter allows images to be randomly flipped horizontally, adding additional variety to the training data.

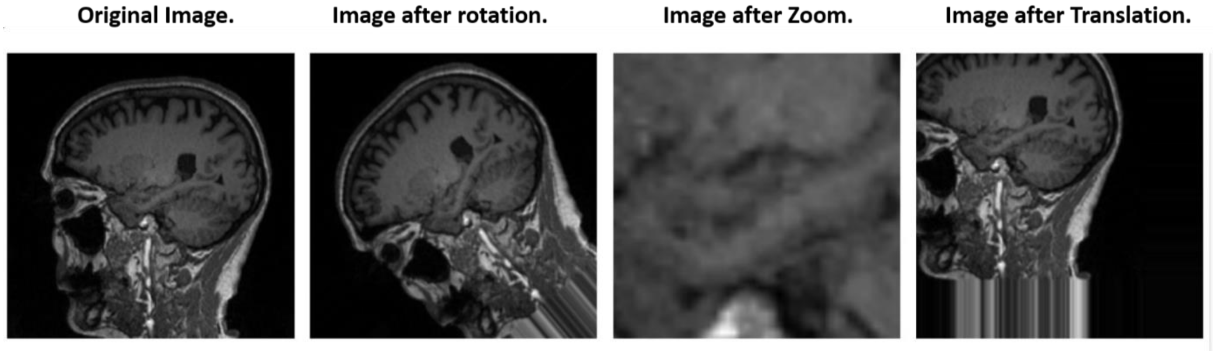


Figure 4.8: *The application of data augmentation.*

4.7Proposed CNN model

The use of pre-trained CNN architectures such as DenseNet121, VGG16, VGG19, ResNet50, and MobileNetV2 requires significant hardware resources and computational power due to the large number of trainable parameters each architecture contains: more than 143 million for VGG19, over 138 million for VGG16, more than 25 million for ResNet50, about 3.5 million for MobileNetV2, and over 8 million for DenseNet121. Additionally, the deeper the network, the longer the training time, and the inference time also increases as the data needs to pass through all the layers. Therefore, in this work, we propose to develop our own CNN architecture from scratch for diagnosing Parkinson's disease, which is much less demanding than pre-trained architectures.

This architecture takes as input an image of size 224x224 pixels, which passes through an initial convolutional layer with 32 filters of size 3x3 pixels. This is followed by several convolutional layers with an increasing number of filters: 64, 128, and finally 256 filters of size 3x3 pixels. Each convolutional layer is followed by a ReLU activation function and a max-pooling layer of size 2x2 pixels to reduce the spatial dimensions. A dropout layer is added to prevent overfitting. The final feature maps are then flattened into a vector, which is passed to a dense layer with 64 neurons and a ReLU activation. The last dense layer with a sigmoid activation provides a probability of the presence or absence of Parkinson's disease. This architecture is optimized to be both computationally efficient and accurate for the specific task of detecting Parkinson's disease. This architecture has 4383649 parameters and is summarized in table 4.1.

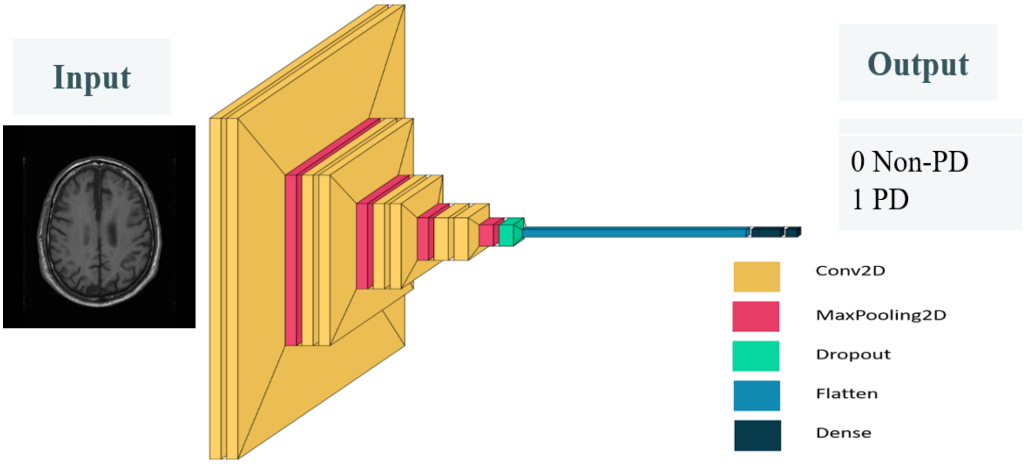


Figure 4.9: *Proposed CNN model architecture.*

Layer (Type)	Output Shape	Parameters
Conv2D	(None, 224, 224, 32)	896
Conv2D	(None, 224, 224, 32)	9248
MaxPooling2D	(None, 112, 112, 32)	0
Conv2D	(None, 112, 112, 64)	18496
Conv2D	(None, 112, 112, 64)	36928
MaxPooling2D	(None, 56, 56, 64)	0
Conv2D	(None, 56, 56, 128)	73856
Conv2D	(None, 56, 56, 128)	147584
MaxPooling2D	(None, 28, 28, 128)	0
Conv2D	(None, 28, 28, 256)	295168
Conv2D	(None, 28, 28, 256)	590080
MaxPooling2D	(None, 14, 14, 256)	0
Dropout	(None, 14, 14, 256)	0
Flatten	(None, 50176)	0
Dense	(None, 512)	3211328
Dense	(None, 1)	65
Number of parameters		4383649

Table 4.1: *Architecture of the proposed model.*

4.8 Implementation and results

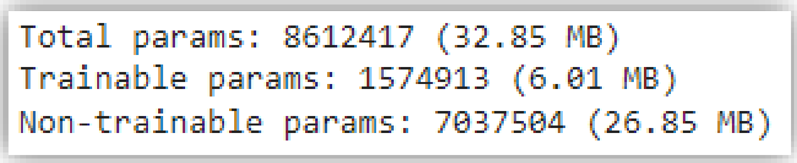
We will describe the pre-trained models used and summarise the results obtained.

4.8.1 DenseNet121

For the DenseNet121 model (Figure 4.10), all the base layers were used without including the top layers. The pre-trained weights on ImageNet were utilized to initialize the network. The base layers were frozen to prevent the weights from being updated during training. To adapt the model to our specific task of Parkinson's disease detection, we added custom layers.

Here are the details of the layers used and added:

- **GlobalAveragePooling2D**: This layer is used to reduce the dimensionality of the feature space and obtain a fixed representation of the output from the last convolutional layer of DenseNet121. It performs a global average on each feature map.
- **Dense (1024 units, activation ReLU)**: A dense layer with 1024 units and a ReLU (Rectified Linear Unit) activation. This layer is used to learn non-linear combinations of features from the global representations.
- **Dense (512 units, activation ReLU)**: A second dense layer with 512 units and a ReLU activation. It further refines the representations learned by the previous layer.
- **Dense (1-unit, activation sigmoid)**: The output layer with a single unit and a sigmoid activation, used for binary classification (presence or absence of Parkinson's disease).

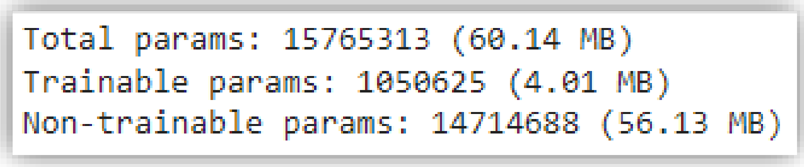


```
Total params: 8612417 (32.85 MB)
Trainable params: 1574913 (6.01 MB)
Non-trainable params: 7037504 (26.85 MB)
```

Figure 4.10*: DenseNet121 parameters.*

4.8.2 VGG16

For the VGG16 model (Figure 4.11), a similar approach was followed to tailor the model to our Parkinson's disease detection task. By using pre-trained weights on ImageNet, we initialized the network and froze the base layers to retain the knowledge gained from training on ImageNet. The added custom layers include a GlobalAveragePooling2D to reduce feature dimensionality, followed by two Dense layers with ReLU activations to learn non-linear combinations of features and an output layer with a sigmoid activation for binary classification. By freezing the base layers and adding these custom layers, we effectively adapt the VGG16 model to our specific task of Parkinson's disease detection.

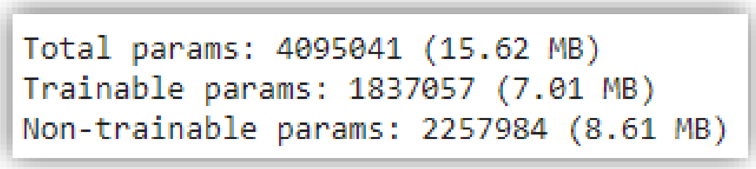


```
Total params: 15765313 (60.14 MB)
Trainable params: 1050625 (4.01 MB)
Non-trainable params: 14714688 (56.13 MB)
```

Figure 4.11: *VGG16 parameters.*

4.8.3 MobileNetV2

For the MobileNetV2 model, a similar approach was followed to tailor the model for our task. We utilized pre-trained weights from ImageNet to initialize the network and excluded the top layers. The added custom layers include GlobalAveragePooling2D to reduce feature dimensionality, followed by two Dense layers with ReLU activations for learning non-linear combinations of features, and an output layer with a sigmoid activation for binary classification. By freezing the base layers and adding these custom layers, we effectively adapt the MobileNetV2 model to our specific task. This allows us to leverage the learned representations from ImageNet while fine-tuning the model for our particular application.

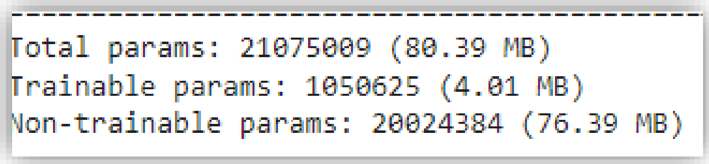


```
Total params: 4095041 (15.62 MB)
Trainable params: 1837057 (7.01 MB)
Non-trainable params: 2257984 (8.61 MB)
```

Figure 4.12: *MobileNetV2 parameters.*

4.8.4 VGG19

For the VGG19 model, a similar approach was employed as with other pre-trained models. Utilizing pre-trained weights from ImageNet, we initialized the network while excluding the top layers. Custom layers were then added, including GlobalAveragePooling2D to reduce feature dimensionality, followed by two Dense layers with ReLU activations for learning non-linear combinations of features, and an output layer with a sigmoid activation for binary classification. By freezing the base layers and incorporating these custom layers, we effectively adapted the VGG19 model to our specific task. This strategy enables us to capitalize on the learned representations from ImageNet while fine-tuning the model for our particular application.

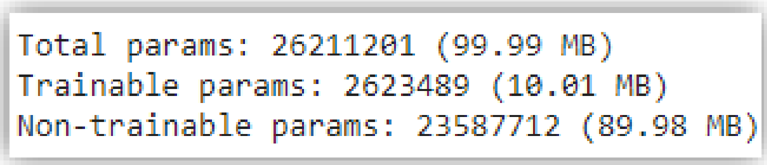


```
------------------------------------------
Total params: 21075009 (80.39 MB)
Trainable params: 1050625 (4.01 MB)
Non-trainable params: 20024384 (76.39 MB)
```

Figure 4.13: *VGG19 parameters.*

4.8.5 ResNet50

For the ResNet50 model, a similar procedure was followed as with other pre-trained models. We initialized the network using weights pretrained on ImageNet, excluding the top layers. Then, custom layers were added to tailor the model for our specific task. These new layers included GlobalAveragePooling2D to condense the feature space, followed by two Dense layers with ReLU activations to capture nonlinear feature combinations, and an output layer with a sigmoid activation for binary classification. By freezing the base layers and integrating these custom layers, we effectively fine-tuned the ResNet50 model for our task. This approach allowed us to benefit from the pretrained representations learned on ImageNet while adapting the model to our particular application.



```
Total params: 26211201 (99.99 MB)
Trainable params: 2623489 (10.01 MB)
Non-trainable params: 23587712 (89.98 MB)
```

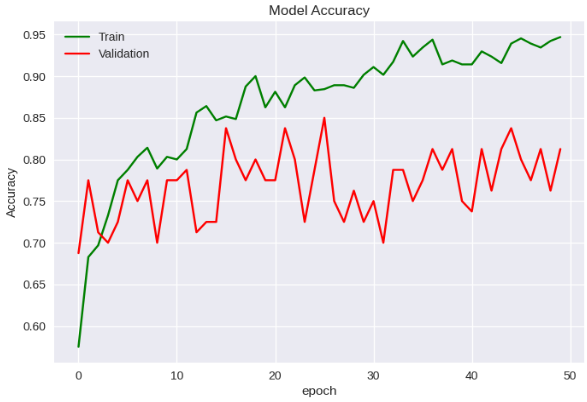
Figure 4.14: *ResNet50 parameters.*

4.9 Results and discussion

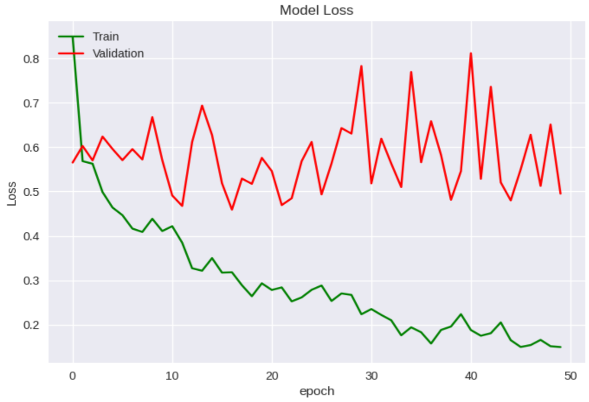
We have generated the results obtained during the realization and execution of the different pre-trained models below.

4.9.1 Accuracy and Loss result

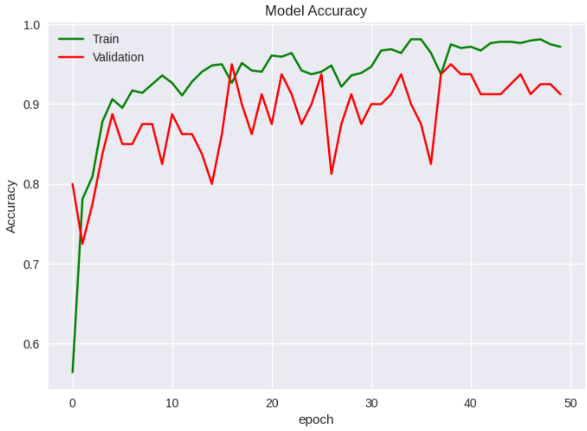
Here is a recap of the accuracy and the loss of the pre-trained model.



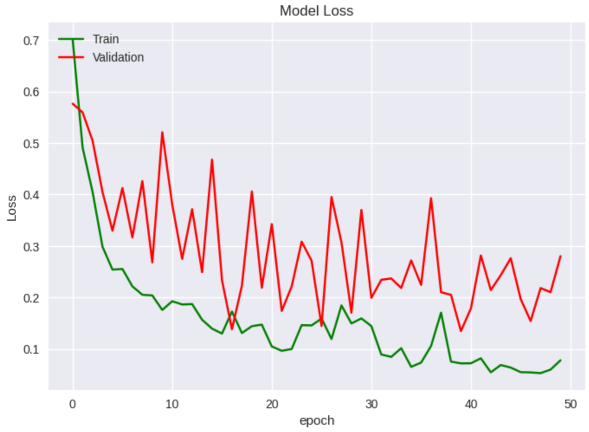
DenseNet121 Accuracy.



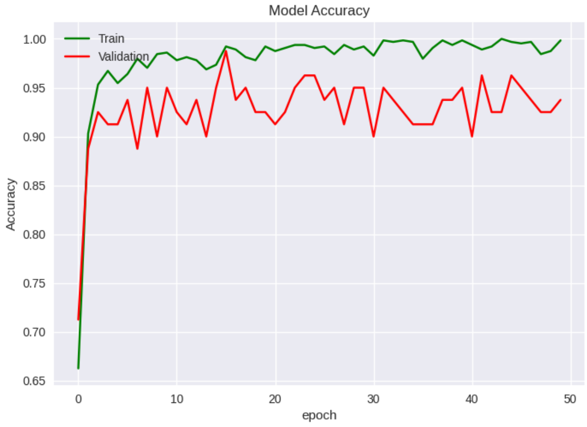
DenseNet121 Loss.



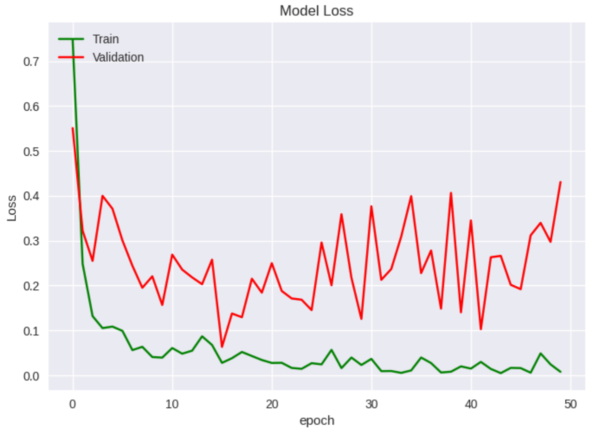
VGG16 Accuracy.



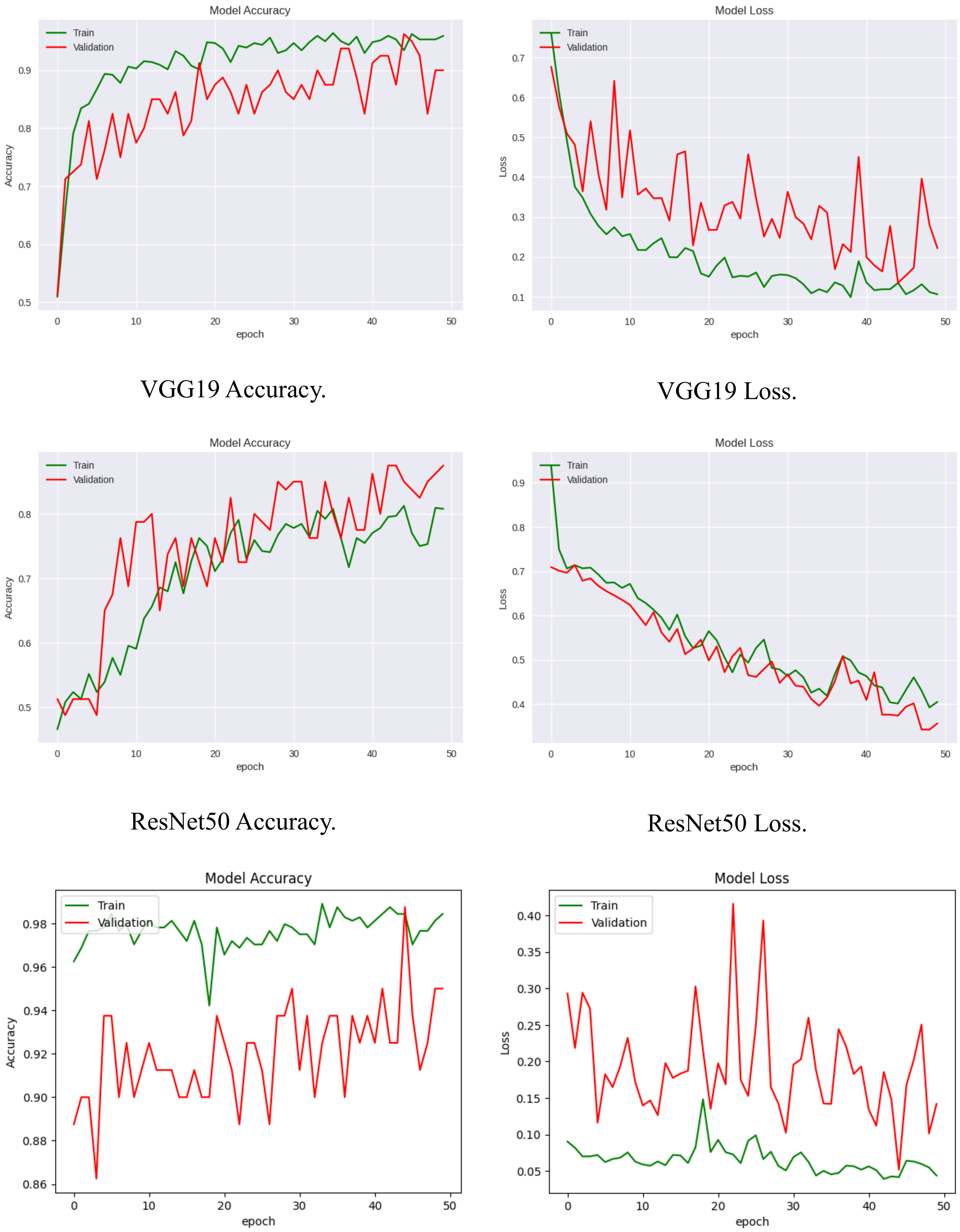
VGG16 Loss.



MobileNetV2 Accuracy.



MobileNetV2 Loss.



VGG19 Accuracy.

VGG19 Loss.

ResNet50 Accuracy.

ResNet50 Loss.

CNN Accuracy.

CNN Loss.

Table 4.2: *Accuracy and loss of pre-trained model.*

4.9.1.1 Discussion the pre-trained model accuracy graph

The results from various pre-trained models over 50 epochs reveal distinct patterns of learning and generalization. The MobileNetV2 model shows rapid initial improvement in both training and validation accuracy, reaching 98.91% and 91.25% respectively, with some validation fluctuations indicating slight overfitting. The VGG16 model demonstrates steady improvement, with training and validation accuracies converging around 95.63% and 93.75% respectively, with minimal overfitting, showcasing robust performance. DenseNet121 achieves a training accuracy of 93.91%, but exhibits erratic validation accuracy stabilizing at 81.25%, despite significant fluctuations. The VGG19 model displays steady training accuracy growth to 96.41%, with validation accuracy also improving to slightly above 88.75%, showing good generalizability with minor fluctuations. Finally, the ResNet50 model improves training accuracy of 93.28%, while its validation accuracy, despite an overall increase to 87.50%, shows sporadic drops, indicating challenges in generalization.

As we can illustrate, our CNN model performs very well in terms of training accuracy and validation, with 98.28% and 91.25% respectively.

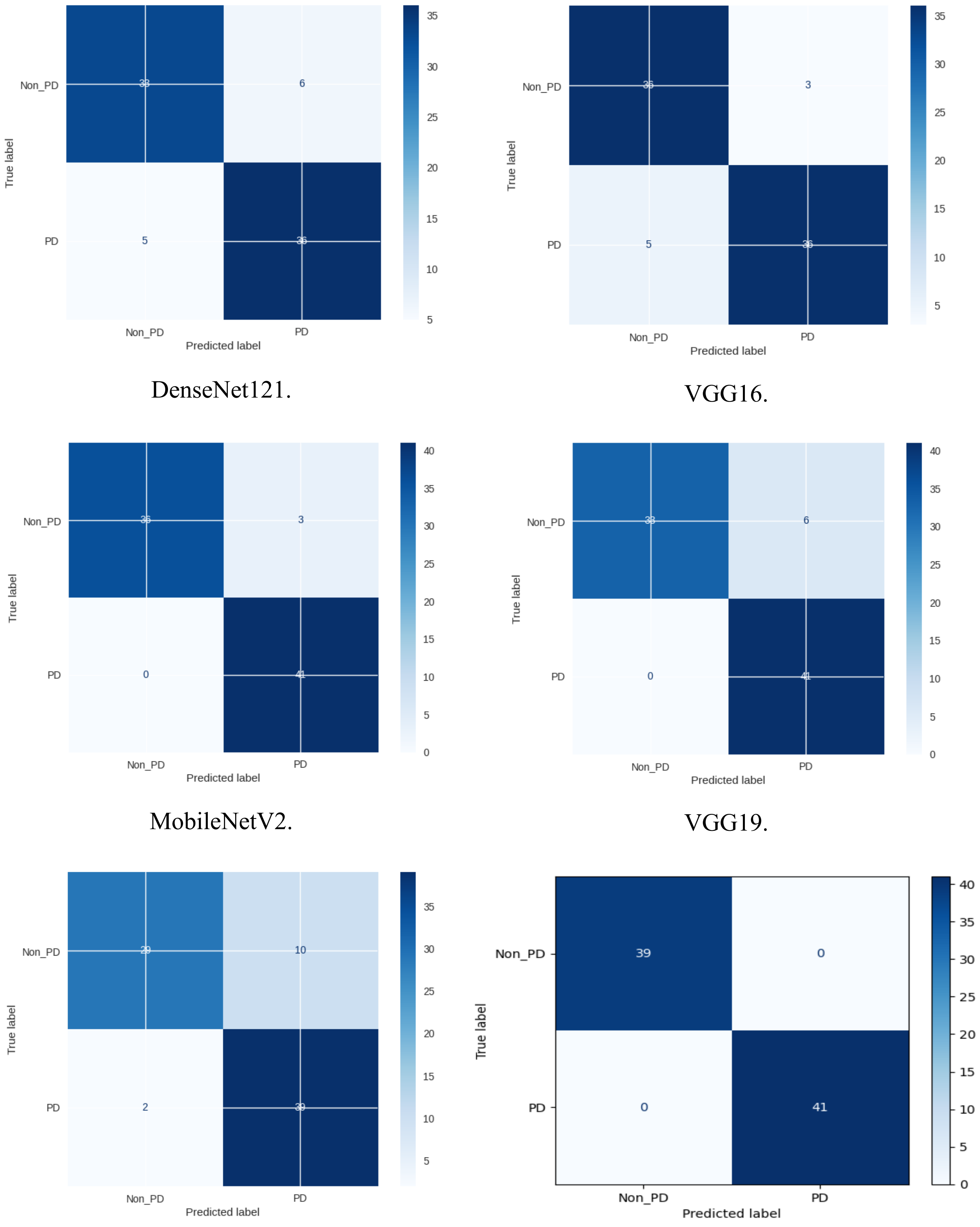
4.9.1.2 Discussion on the pre-trained model loss graph

The loss graphs for various pre-trained models over 50 epochs reveal distinct learning patterns and generalization abilities. The MobileNetV2 model shows rapid initial learning with a steady decline in training loss to 0.01 and a more fluctuating validation loss ending at 0.5, indicating slight overfitting. The VGG16 model exhibits initial loss reduction with a divergence between epochs 10 and 40, suggesting overfitting, but eventual convergence of losses indicates good generalization with a 0.1 of train loss and 0.12 of validation loss. DenseNet121 shows training with a loss of 0.14, but an erratic validation loss of 0.4, indicating some overfitting. The VGG19 model shows rapid initial learning with a steady decline in training loss to 0.09 and a more fluctuating validation loss ending at 0.3, indicating slight overfitting. The ResNet50 model has a learning loss of 0.2, and despite occasional fluctuations in the validation loss, it ends with a validation loss of 0.3. Collectively, these results highlight varying overfitting and generalization levels across models, each showing unique strengths and weaknesses in handling training and validation data.

After these pre-trained models results we illustrate, that our CNN model performs very well in terms of training loss and validation, with 0.04 and 0.1 respectively.

4.9.2 Confusion Metric result

Here is a recap of the confusion metric of the pre-trained model.



DenseNet121.

VGG16.

MobileNetV2.

VGG19.

ResNet50.

CNN Model.

***Table 4.3:** Recap of Confusion Metric.*

4.9.2.1 Discussion on the pre-trained model confusion metric

The confusion metrics for various models highlight their classification performance for PD and Non-PD patients. The MobileNetV2 model shows strong performance with 41 true positives, 36 true negatives, 3 false positives, and 0 false negatives, indicating high reliability. The VGG16 model correctly identified 36 PD patients and 36 Non-PD patients, with 3 false positives and 5 false negatives, demonstrating effective classification. DenseNet121 achieved 36 true positives, 33 true negatives, 6 false positives, and 5 false negatives, indicating high accuracy with minimal errors. The VGG19 model classified 41 PD patients and 33 Non-PD patients correctly, with 6 false positives and 0 false negatives, still showing notable effectiveness. Lastly, the ResNet50 model had 39 true positives, 29 true negatives, 10 false positives, and 2 false negatives, indicating effective but less precise classification compared to the other models. Overall, all models exhibited strong performance, particularly with low false negative rates, which is crucial in medical diagnostics.

After these results of the different pre-trained models, we note that our CNN model shows very extremely result with 0 false positives and 0 false negatives.

4.10 Comparison of the Results

Model	**DenseNet121**	**VGG16**	**MobileNetV2**	**VGG19**	**ResNet50**	**Our CNN**
Train Accuracy	0.9391	0.9563	**0.9891**	0.9641	0.9328	0.9828
Validation Accuracy	0.8125	**0.9375**	0.9125	0.8875	0.8750	0.9125
Test Accuracy	0.8625	0.9250	0.9625	0.9000	0.8500	**0.9875**
F1-score	0.862	0.900	0.962	0.924	0.847	**1**
Recall	0.878	0.878	1	1	0.951	**1**
Precision	0.857	0.923	0.931	0.872	0.795	**1**

Table 4.4: Results' comparison.

In comparing the results of various deep learning models for classification tasks, our custom CNN model exhibits outstanding performance across multiple evaluation metrics, making it the best-performing model in the set. The CNN achieves the highest test accuracy of 98.75%, significantly surpassing DenseNet121 (86.25%), VGG16 (92.50%), MobileNetV2 (96.25%), VGG19 (90.00%), and ResNet50 (85.00%). This indicates that our CNN model is the most accurate in predicting the correct class labels for the test dataset.

The F1-score of the CNN model is perfect at 1, showcasing its exceptional balance between precision and recall. This means the CNN not only identifies all relevant instances (high recall) but also ensures that all identified instances are correct (high precision). In contrast, other models like DenseNet121 and VGG16 have lower F1-scores, with DenseNet121 scoring 0.862, and VGG16 scoring 0.9.

Furthermore, the CNN model's recall and precision are both perfect at 1, indicating that the model accurately detects all true positive cases (recall) and that all predicted positive cases are indeed true positives (precision). This perfect balance underscores the CNN model's superior capability in handling the classification task with minimal false positives and false negatives. Other models, such as DenseNet121 and VGG16, exhibit lower recall and precision, further highlighting the effectiveness of the CNN model.

The CNN's performance on training and validation datasets is impressive, with a training accuracy of 98.28% and a validation accuracy of 91.25%.

4.11 Comparison of our CNN model with related work

To position our approach with regard to the literature, we compare it with the works in [54], [55], and [56] since we are using the same dataset. This table presents the comparison results in terms of accuracy metrics.

Reference	**Year**	**Model**	**Imaging Type**	**Testing Accuracy**
Kollia et al **(53)**	2019	CNN	MRI	94%
BASNIN et al **(54)**	2021	DenseNet-LSTM	T1, T2, Flair MRI	90%
PECHETTI et al **(55)**	2023	Optimized MobileNet V3	T1, T2, Flair MRI	95%
Our Model		CNN Model	T1 MRI	**98.75%**

Table 4.5: *Theoric comparison.*

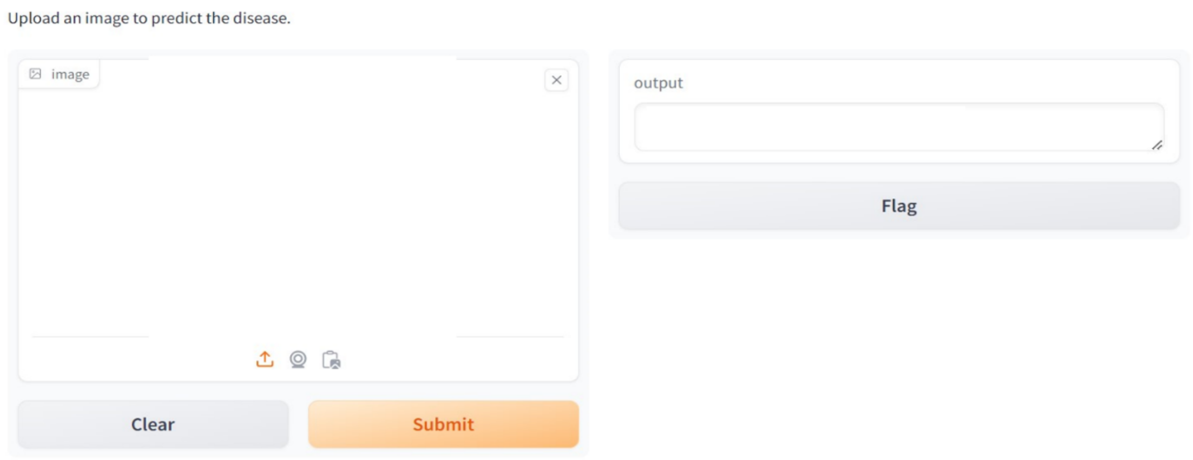
This study demonstrated the effectiveness of deep learning models, particularly our proposed CNN model, in diagnosing Parkinson's disease using MRI data. Our experiments showed that this model outperformed existing studies, as evidenced by a testing accuracy of 98.75%, higher than those achieved by another model such as Optimized MobileNet V3 (95%). This indicates that our CNN model can significantly improve the accuracy of Parkinson's disease diagnosis.

4.12 Model Diagnosis

As an extension of our work, we aimed to develop an application or website where users could upload MRI images to predict whether they indicate Parkinson's disease or not. However, due to time constraints, we were unable to do this task. Instead of halting our efforts, we have found an alternative solution which is the Gradio library.

Gradio is a powerful Python library that allows for the creation of interactive graphical user interfaces with ease. Utilizing Gradio enabled us to implement a user-friendly interface for our prediction model, facilitating the process of uploading and analyzing MRI images for PD detection.

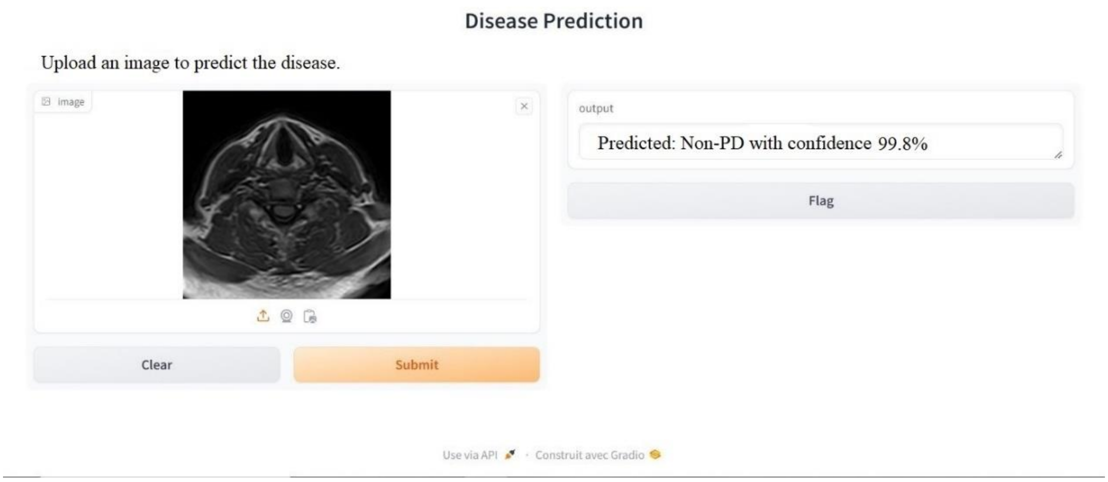
The figure 4.15 shows the Gradio interface.



Upload an image to predict the disease.

Figure 4.15: *Gradio interface.*

Figure 4.16 shows the Gradio interface predicting that the input MRI image does not indicate Parkinson's disease, with a perfect confidence score of 99.8%. This result demonstrates the model's accuracy and reliability in distinguishing Non-PD cases from PD cases.



Disease Prediction

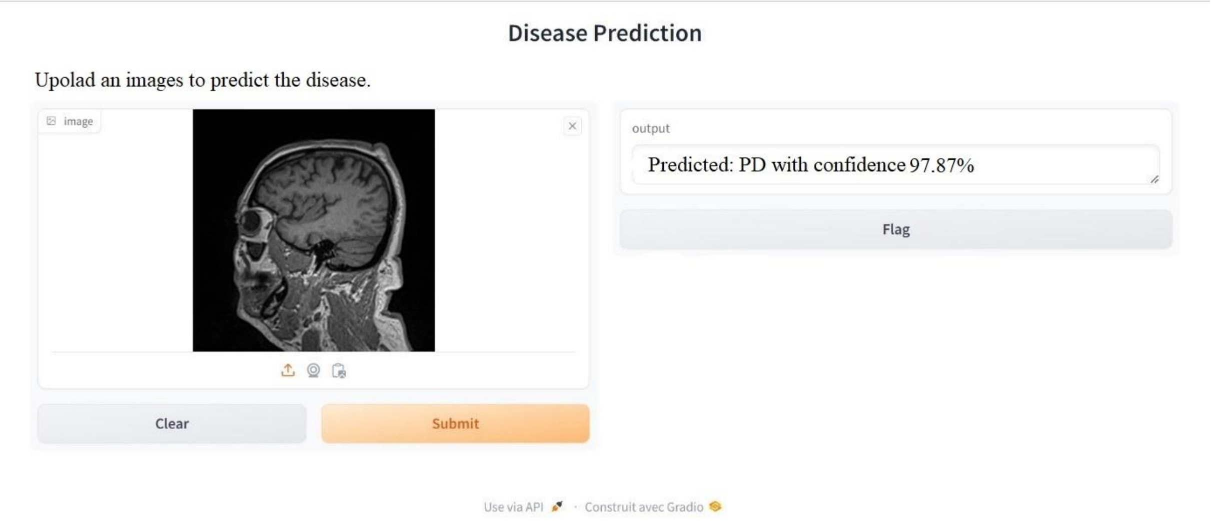
Upload an image to predict the disease.

output

Predicted: Non-PD with confidence 99.8%

__Figure 4.16__: Non_PD prediction.

In Figure 4.17, the Gradio interface is used to predict that the input MRI image indicates the presence of Parkinson's disease with a confidence level of 97.87%. This high confidence score reflects the model's strong ability to identify features associated with PD in MRI images.



Disease Prediction

Upolad an images to predict the disease.

output

Predicted: PD with confidence 97.87%

***Figure 4.17:** PD prediction.*

4.13 Conclusion

In this chapter, first, we have discussed the libraries utilized for the implementation of our models. The dataset was defined, detailing the number of images in each dataframe. then we have explored the results from several pre-trained models as well as our proposed CNN model. Our proposed CNN model demonstrated exceptional performance, achieving an accuracy of 98.75% and a loss of 5.23%. This significant accuracy proves the effectiveness of our model in accurately classifying images, and the low loss value indicates the model's reliability in prediction.

General conclusion

This work explored the use of deep learning for the detection of Parkinson's disease, the second most common neurodegenerative disease after Alzheimer's disease. For this purpose, we used the NTUA database, which contains MRI (Magnetic Resonance Imaging) images essential for our analysis.

In the first chapter, we laid the foundation of our study by defining the key concepts of Artificial Intelligence (AI), Machine Learning, and Deep Learning. We also explored the different architectures underlying these technologies, providing a solid theoretical framework for our research.

The second chapter focused on understanding neurodegenerative diseases, with a particular emphasis on Parkinson's disease. We described the different stages of the disease, its main symptoms, and included an overview of previous work in this field. This section highlighted the importance and urgency of developing early and accurate detection tools.

In the third chapter, we delved into the technical details by presenting the architectures of the pre-trained models we used, as well as our own proposed Convolutional Neural Network (CNN) model. We explained why and how each model was selected and adapted for our specific task.

Finally, the last chapter was dedicated to the implementation and analysis of the results obtained. We tested our models with a batch size of 32 over 50 epochs, using the ADAM optimizer, and achieved the following accuracies:

- VGG16 : 92.50%
- VGG19 : 90%
- MobileNetV2 : 96.25%
- DenseNet121 : 86.25%
- ResNet50 : 85%

Our own CNN model outperformed all these models with a remarkable accuracy of 98.75%. This result demonstrates the potential and effectiveness of our customized approach for detecting Parkinson's disease.

In conclusion, our study shows that deep learning, and especially CNN architectures, can play an important role in the early and accurate diagnosis of Parkinson's disease. The outstanding performance of our proposed model opens promising prospects for Future research will involve testing additional MRI scans from the PPMI dataset to enhance diagnostic performance and exploring the identification of Parkinson's disease stages to aid in personalized treatment planning and monitoring disease progression. Integration of other imaging modalities, such as DaT Scans, with MRI could further improve diagnostic accuracy. Implementing real-time processing capabilities will facilitate immediate diagnosis during clinical visits, and validating the model across multiple centers and diverse populations will ensure its generalizability and robustness. Developing a user-friendly interface for clinicians to easily interpret the results and integrate them into their workflow will enhance the practical utility of the proposed system.

Bibliography

[1] CORNUÉJOLS, Antoine et MICLET, Laurent. Apprentissage artificiel : concepts et algorithmes. . s.l. : Editions Eyrolles., 2011.

[2] THEOBALD, Oliver.. Machine learning for absolute beginners: a plain English introduction. . London, UK : Scatterplot press,, 2017.

[3] KADHIM, Ammar Ismael. Survey on supervised machine learning techniques for automatic text classification. . Artificial Intelligence Review, 2019,. 2019. Vol. vol. 52, , no 1, p. 273-292.

[4] MUHAMMAD, Iqbal et YAN, Zhu. SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. ICTACT Journal on Soft Computing, 2015, vol. 5, no 3.

[5] NGUYEN, Duc-Phong. Enhanced facial behavior recognition and rehabilitation using 3D biomechanical features and deep learning approaches. 2022. Thèse de doctorat. Compiègne.

[6] JIANG, Jung Yi, CHENG, Wen Hao, et LEE, Shie Jue. A dissimilarity measure for document clustering. ICIC Express Letters, 2012, vol. 6, no 1, p. 15-21.

[7] MAHESH, Batta. Machine learning algorithms-a review. International Journal of Science and Research (IJSR). [Internet], 2020, vol. 9, no 1, p. 381-386.

[8] KAELBLING, Leslie Pack, LITTMAN, Michael L., et MOORE, Andrew W. Reinforcement learning: A survey. Journal of artificial intelligence research, 1996, vol. 4, p. 237-285.

[9] SOUCY, Pascal et MINEAU, Guy W. A simple KNN algorithm for text categorization. In: Proceedings 2001 IEEE international conference on data mining. San Jose, California. IEEE, 2001. p. 647-648.

[10] SCHRITTWIESER, Julian, ANTONOGLOU, Ioannis, HUBERT, Thomas, et al. Mastering Atari, go, chess and shogi by planning with a learned model. Nature, 2020, vol. 588, no 7839, p. 604-609.

[11] Youcef Djeriri, Université de Sidi-Bel-Abbès, Les réseaux de neurones Artificiels 09/2017.

[12] AOUEDI, Ons, PIAMRAT, Kandaraj, HAMMA, Salima, et al. Network traffic analysis using machine learning: an unsupervised approach to understand and slice your network. Annals of Telecommunications, 2022, vol. 77, no 5, p. 297-309.

[13] KIMURA, Nobuaki, YOSHINAGA, Ikuo, SEKIJIMA, Kenji, et al. Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. Water, 2019, vol. 12, no 1, p. 96.

[14] KRIZHEVSKY, Alex, SUTSKEVER, Ilya, et HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, vol. 25.

[15] L. Deng et al., “Recent advances in deep learning for speech research at Microsoft,” ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, pp. 8604–8608, 2013, doi: 10.1109/ICASSP.2013.6639345.

[16] BENRADI, Hicham, CHATER, Ahmed, et LASFAR, Abdelali. A hybrid approach for face recognition using a convolutional neural network combined with feature extraction techniques. IAES Int J Artif Intell, 2023, vol. 12, no 2, p. 627.

[17] SAZLI, Murat H. A brief review of feed-forward neural networks. Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, 2006, vol. 50, no 01.

[18] BULLINARIA, John A. Recurrent neural networks. Neural Computation: Lecture, 2013, vol. 12, no 1.

[19] KUMARASWAMY, Balachandra. Neural networks for data classification. In : Artificial intelligence in data mining. Academic Press, 2021. p. 109-131.

[20] ZHANG, Qingchen, YANG, Laurence T., CHEN, Zhikui, et al. A survey on deep learning for big data. Information Fusion, 2018, vol. 42, p. 146-157.

[21] ZHANG, Guoqiang Peter. Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2000, vol. 30, no 4, p. 451-462.

[22] Jakhar, D., & Kaur, I. (2019). Artificial intelligence, machine learning & deep learning: Definitions and differences. Clinical and Experimental Dermatology. doi:10.1111/ced.14029.

[23] BADRAN, Fouad et THIRIA, Sylvie. Les perceptrons multicouches : de la régression non-linéaire aux problèmes inverses. In: Journal de Physique IV (Proceedings). EDP sciences, 2002. p. 157-188.

[24] PEIXOTO, Francke. A simple overview of multilayer perceptron (MLP). Data Science Blogathon, December, 2020, vol. 13.

[25] KHAN, Asifullah, SOHAIL, Anabia, ZAHOORA, Umme, et al. A survey of the recent architectures of deep convolutional neural networks. Artificial intelligence review, 2020, vol. 53, p. 5455-5516.

[26] INDOLIA, Sakshi, GOSWAMI, Anil Kumar, MISHRA, Surya Prakesh, et al. Conceptual understanding of convolutional neural network-a deep learning approach. Procedia computer science, 2018, vol. 132, p. 679-688.

[27] ABINAYA, R., et al. Acoustic based scene event identification using deep learning cnn. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2021, vol. 12, no 5, p. 1398-1405.

[28] MOGHAR, Adil et HAMICHE, Mhamed. Stock market prediction using LSTM recurrent neural network. Procedia Computer Science, 2020, vol. 170, p. 1168-1173.

[29] MORO-VELAZQUEZ, Laureano, GOMEZ-GARCIA, Jorge A., ARIAS-LONDOÑO, Julian D., et al. Advances in Parkinson's disease detection and assessment using voice and

speech: A review of the articulatory and phonatory aspects. Biomedical Signal Processing and Contro.

[30] PARKINSON, James. An essay on the shaking palsy. The Journal of neuropsychiatry and clinical neurosciences, 2002, vol. 14, no 2, p. 223-236.

[31] JANKOVIC, Joseph. Parkinson’s disease: clinical features and diagnosis. Journal of neurology, neurosurgery & psychiatry, 2008, vol. 79, no 4, p. 368-376.

[32] ROMDHAN, Sawssan Ben. Étude génétique et corrélation génotype-phénotype de la maladie de Parkinson dans la population tunisienne. Thèse de doctorat. Université Paris sciences et lettres et 2019., Université de Sfax (Tunisie).

[33] DEMAAGD, George et PHILIP, Ashok. Parkinson’s disease and its management: part 1: disease entity, risk factors, pathophysiology, clinical presentation, and diagnosis. Pharmacy and therapeutics, 2015, vol. 40, no 8, p. 504.

[34] PFEIFFER, Helle Cecilie Viekilde, LØKKEGAARD, A., ZOETMULDER, Marielle, et al. Cognitive impairment in early-stage non-demented Parkinson's disease patients. Acta Neurologica Scandinavica, 2014, vol. 129, no 5, p. 307-318.

[35] JOST, Wolfgang H. Autonomic dysfunctions in idiopathic Parkinson's disease. Journal of neurology, 2003, vol. 250, p. i28-i30.

[36] PAHUJA, Gunjan et NAGABHUSHAN, T. N. A comparative study of existing machine learning approaches for Parkinson's disease detection. IETE Journal of Research, 2021, vol. 67, no 1, p. 4-14.

[37] Duffy, J. “Motor speech disorders: Substrates, differential diagnosis, and management.” St. Louis, MO: Elsevier (2013).

[38] LOH, Hui Wen, OOI, Chui Ping, PALMER, Elizabeth, et al. GaborPDNet: Gabor transformation and deep neural network for Parkinson’s disease detection using EEG signals. Electronics, 2021, vol. 10, no 14, p. 1740.

[39] TEULINGS, Hans-Leo, CONTRERAS-VIDAL, José L., STELMACH, George E., et al. Parkinsonism reduces coordination of fingers, wrist, and arm in fine motor control. Experimental neurology, 1997, vol. 146, no 1, p. 159-170.

[40] Automated restrictes Boltzmann machine transfert for early diagnosis pf Parkinson's disease using digitized spiral drawings. THAKUR, Mahima, DHANALAKSHMI, Samiappan, KURESAN,Harisudha, et al. no 1, s.l. : Journal of Ambient Intelligence and Humanized Computing., 2023, Vol. col. 14. p. 175-189.

[41] Helen Ling, Luke A. Massey, Andrew J. Lees, Peter Brown, Brian L. Day, Hypokinesia without decrement distinguishes progressive supranuclear palsy from Parkinson's disease, Brain, Volume 135, Issue 4, April 2012, Pages 1141–1153.

[42] ISLAM, SK M. Shadekul, NASIM, Md Abdullah Al, HOSSAIN, Ismail, et al. Introduction of Medical Imaging Modalities. In: Data Driven Approaches on Medical Imaging. Cham: Springer Nature Switzerland, 2023. p. 1-25.

[43] Kak AC, Slaney M. Principles of Computerized Tomographic Imaging: IEEE Press, 1987. Kalender WA. Computed Tomography. Fundamentals, System Technology, Image Quality, Applications. Erlangen: Publicis.

[44] TAGARIS, Athanasios, KOLLIAS, Dimitrios, STAFYLOPATIS, Andreas, et al. Machine learning for neurodegenerative disorder diagnosis—survey of practices and launch of benchmark dataset. International Journal on Artificial Intelligence Tools. 2018. Vol. vol. 27, no 03. p.1850011.

[45] SHAH, Pir Masoom, ZEB, Adnan, SHAFI, Uferah, et al. Detection of Parkinson disease in brain MRI using convolutional neural network. In: 2018 24th international conference on automation and computing (ICAC). IEEE, 2018. p. 1-6.

[46] ESMAEILZADEH, Soheil, YANG, Yao, et ADELI, Ehsan. End-to-end parkinson disease diagnosis using brain mr-images by 3d-cnn. arXiv preprint arXiv:1806.05233, 2018.

[47] ORTIZ, Andrés, MUNILLA, Jorge, MARTÍNEZ-IBAÑEZ, Manuel, et al. Parkinson's disease detection using isosurfaces-based features and convolutional neural networks. Frontiers in neuroinformatics, 2019, vol. 13, p. 48.

[48] WENZEL, Markus, MILLETARI, Fausto, KRÜGER, Julia, et al. Automatic classification of dopamine transporter SPECT: deep convolutional neural networks can be trained to be robust with respect to variable image characteristics. s.l. : European journal of nuclear medicine and molecular imaging, 2019. Vol. vol. 143.

[49] EL MAACHI, Imanne, BILODEAU, Guillaume-Alexandre, et BOUACHIR, Wassim. Deep 1D-Convnet for accurate Parkinson disease detection and severity prediction from gait. Expert Systems with Applications, 2020, vol. 143, p. 113075.

[50] CHAKRABORTY, Sabyasachi, AICH, Satyabrata, et KIM, Hee-Cheol. Detection of Parkinson's disease from 3T T1 weighted MRI scans using 3D convolutional neural network. Diagnostics, 2020, vol. 10, no 6, p. 402.

[51]. Balaji E, Brindha D, Elumalai VK, Vikrama R. Automatic and non-invasive Parkinson's disease diagnosis and severity rating using LSTM network. Appl Soft Comput. 2021 et 108:107463.

[52] VYAS, Tarjni, YADAV, Raj, SOLANKI, Chitra, et al. Deep learning-based scheme to diagnose Parkinson's disease. Expert Systems, 2022, vol. 39, no 3, p. e12739.

[53] Kollia, I., Stafylopatis, A.G., Kollias, S.: Predicting parkinson's disease using latent information extracted from deep neural networks. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2019).

[54] BASNIN, Nanziba, NAHAR, Nazmun, ANIKA, Fahmida Ahmed, et al. Deep learning approach to classify Parkinson's disease from MRI samples. In: International conference on brain informatics. Cham: Springer International Publishing, 2021. p. 536-547.

[55] PECHETTI, Sukanya et RAO, Battula Srinivasa. Optimized MobileNetV3: a deep learning-based Parkinson's disease classification using fused images. PeerJ Computer Science, 2023, vol. 9, p. e1702.

[56] GOVINDU, Aditi et PALWE, Sushila. Early detection of Parkinson's disease using machine learning. Procedia Computer Science, 2023, vol. 218, p. 249-261.

[57] Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248–255). https://doi.org/10.1109/CVPR.2009.5206848.

[58] Jude Shavlik Lisa Torrey. Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques. In IGI Global, chapter 11. Transfer Learning. IGI Global, 2009. ISBN 9781605667669. doi: 10.4018/978-1-60566-766-9.

[59] KRIZHEVSKY, Alex, SUTSKEVER, Ilya, et HINTON, Geoffrey E. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2017, vol. 60, no 6, p. 84-90.

[60] A. A. Abbasi, L. Hussain, I. A. Awan, I. Abbasi, A. Majid, M. S. A. Nadeem, and Q.-A. Chaudhary, "Detecting prostate cancer using deep learning convolution neural network with transfer learning approach," Cognitive Neurodynamics,. 2020. Vol. vol. 14,, no. 4, . pp. 523–533, .

[61] A. G. Howard, M. Zhu. B. Chen, D. Kalenichenko. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arxiv: 170404861v1, 2017.

[62] Simonyan, K. et Zisserman, A. (2015): Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556v6.

[63] Mascarenhas, Sheldon et Mukul Agarwal. (2021). A comparison between the VGG16, VGG19 and ResNet50 architecture frameworks for image classification. 2021 International Conference on Disruptive Technologies for Multidisciplinary Research and Applications.

[64] He, K., et al. He, K.; Zhang, X.; Ren, S.; Sun, J. (2015): Deep Residual Learning for Image Recognition. arXiv:1512.03385v1.

[65] A. Victor Ikechukwu, S. MuraliR. DeepuR.C. Shivamurthy. (2021). ResNet-50 vs VGG-19 vs training from scratch: a comparative analysis of pneumonia segmentation and classification from chest X-ray images. Proceedings on Global Transitions. Vol. Vol, no2. pp375-381.

[66] Huang, Gao et al. (2016). "Densely Connected Convolutional Networks". In: arXiv: 1608. 06993 [cs.CV].

[67] W. Liu, H. Li, C. Hua, L. Zhao. Classification of Breast Cancer Images by Deep Learning, medrxiv, pages 1-13, https://doi.org/10.1101/2020.06.13.20130633, 2020.

[68] Venali Sonone. (2019). Notes on Deep Learning Softmax Classifier. [Online]. Available:https://medium.com/datadriveninvestor/. notes-on-deep-learning-softmax-classifier-971b3df27466.

[69]. Introduction douce à l'algorithme d'optimisation Adam pour l'apprentissage en profondeur . J Brownlee - Maîtrise de l'apprentissage automatique, 2017.

[70] S. Deepak, P.M. Ameer. (2019). Classification des tumeurs cérébrales à l'aide de fonctionnalités CNN approfondies via l'apprentissage par transfert. Informatics in Biology and Medicine. Vol 11.

[71] FATOUMATA Y & AMOR A. (2021), « Machine learning pour la maintenance prédictive », Mémoire de fin d'études, Université Larbi Ben Mhidi d'Oum El-Bouaghi, juillet, 42p.

[72] MIFDAL R, « Application des techniques d'apprentissage automatique pour la prédiction de la tendance des titres financiers », L'obtention De La Maitrise, Sous la direction de M. Edmond Miresco, École De Technologie Supérieure Université Du Québec, 2019,. p 176.

[73] Heim, B., Krismer, F., De Marzi, R., Seppi, K. Magnetic resonance imaging for the diagnosis of parkinson's disease. 124, 915–964. s.l. : Journal of neural transmission, 2017. 124. 915–964.