SELINUS UNIVERSITY
OF SCIENCES AND LITERATURE

ARTIFICIAL INTELLIGENCE-BASED MEDICAL PREDICTIONS

By Abdoulaye Diallo

A DISSERTATION

Presented to the Department of
Artificial Intelligence
program at Selinus University

Faculty of Computer Science
in fulfillment of the requirements
for the degree of Doctor of Philosophy
in Artificial Intelligence
2022

Table of Contents

CHAPTER I Biomedical and machine learning

CHAPTER II Proposed system

CHAPTER III Experimental results

GENERAL INTRODUCTION

The objective of this study will be to update the state of the application of artificial intelligence techniques in the medical predictions of patients as well as its consequences.

In medicine, one of the main challenges is to take a sample or a vital signal to a patient via a specific sensor or by physico-chemical analysis to study and process it and extract relevant information to identify related diseases. Due to the very high cost of these treatments and the number of patients that are increasing day by day, the need to automate this treatment or behavior has become a necessity and a solution to this problem, and with the development of intelligence artificial in recent years when one can replace the health professional by a machine where one can.The latter make critical decisions and a very low error rate compared to the facilities provided by the inclusion of artificial intelligence,

We can assume that the problem of disease identification is seen as the problem of pattern recognition, where the patterns represent a group of observations or samples of patients. Currently, pattern recognition has been widely studied in recent years in the scientific literature. However, despite all the extensive research, the recognition techniques available in this field are still inferior to human visual abilities, because in the latter it is mathematical operations that contain an error rate, even if it is low.

To enable pattern recognition, machine learning (ML) can learn to separate shapes from any of the other shapes where the data needs to be linear. This is why the so-called deep learning has emerged which is even capable of processing non-linear data, as many techniques have been developed to improve the performance of biomedical systems such as tumor recognition, segmentation of medical images. , cell classification and disease identification. Unfortunately, in the systems of the literature, we find efficient techniques but their implementation can be long or rapid but do not give sufficient satisfactory results.

In this thesis, we present a disease recognition system using a set of biomedical data for several different diseases (diabetes, Alzheimer's, plant disease ...) including deep learning techniques where the difference between these techniques is highlighted according to the ways of connecting the layers and methods of "neurons", there is also uncensored, supervised and enhanced learning in deep learning algorithms, as they indicate how the algorithm is fed with data by the researchers. Beyond that, convolutional neural networks (CNNs) are typical for recognizing image, video, and audio data, due to their ability to work with dense data.

In order to complete this work, our thesis is organized in three chapters as follows:

- The first chapter provides a general overview of biomedicine and its relation to machine learning.
- In Chapter Two, we'll highlight the most important differences between machine learning and deep learning, and then explain how deep learning techniques work.
- The third chapter presents and discusses the results obtained with four different real databases and a comparison of the results to determine which technique performs better than others.

ABREVIATIONS LIST

EMG: Electromyogram

ECG: Electrocardiogram

NHLBI: National Heart, Lung, and Blood Institute

GPS: genaral problem solver

IA: Artificial intelligence

FAIR: Facebook Artificial Intelligence Research

SVM: Vector Machines Support

RNA: Artificial Neural Networks

MLP: Multi Layer Perceptron

MMG: Gaussian Mixture Model

MMC: Hidden Markov Models

ML: Machine learning

DL: Deep Learning

DNN: deep neural network

DFFnn: deep feed forward neural network

CNN: convlutional neural network

RNN: Recurrent neural network

LSTM: Long Short-Term Memory

AI

I.1.INTRODUCTION

The terms “biomedicine” and “biomedical” have different origins and trajectories in English, in German “Biomedizin” and in French “biomedicine”, but their meaning is very similar today. "Biomedical" first appeared in the writings of American and British authors in the 1920s, followed ten years later by "biomedicine". The American Medical Dictionary (1923) defines it as "clinical medicine based on the principles of physiology and biochemistry". Thus, from the beginning, biomedicine and biomedical research were understood as a kind of medicine closely associated with experimentation and the laboratory rather than with the knowledge of the doctor and the clinic.

I.2 WHAT IS SCREENING AND WHY?

Screening is a public health action, that is to say a list of operations to be undertaken of a collective nature with a view to improving the health of a population.

Screening for a disease has positive outcomes in subjects correctly identified as positive or negative.

For the true positives, the benefits may be greater effectiveness of the intervention (treatment or preventive measure) initiated earlier, better chances of patient survival, or improvement in the quality of life of patients by reducing treatments, and a saving of resources generated by a reduction in the total cost of the management of the disease (less radical initial treatments and reduction in morbidity and mortality).

For the real negatives, the benefits may be a sense of tranquility experienced by patients and a possible easing of surveillance measures. In addition, screening can contribute to equity by providing access to care for part of the screened population [1].

Because of the importance of medical screening on human health, we can use new, more developed and precise methods to improve the efficiency and the result of our screening such as digital screening and this is what we will see in our memory

I.3. BIOMEDICAL AND COMPUTING

Nowadays, the study of biomedical signals and the extraction of useful information in an automatic way are very important and help physicians to fully understand the behavior of different cell tissues in order to make a correct diagnosis and to construct the correct ones. patient databases. In the following, we list the biomedical signals most used in the automatic treatment of patients using machine learning:

I.3.1 Electromyogram

The electromyogram (EMG) is recorded using an electromyography machine, which measures the electrical potential of the muscle. The central nervous system (the brain, spinal cord and peripheral nerves) controls the action of muscle fibers which usually results in movement. Muscle is made up of special cells that are able to expand and relax and are controlled by neurons.

An EMG can be recorded by two methods, surface EMG (by putting electrodes on the surface of the skin) which is more popular than the second called intramuscular EMG (where a needle electrode is inserted into the skin) because it is non-invasive.

A surface EMG measures one or more motor units, it is called the motor unit action potential (PAUM). The real potential is around 100 mV but because of the connective tissue layers and skin, EMG is a complex signal with a much smaller amplitude (around 5 mV) (Figure I.1).

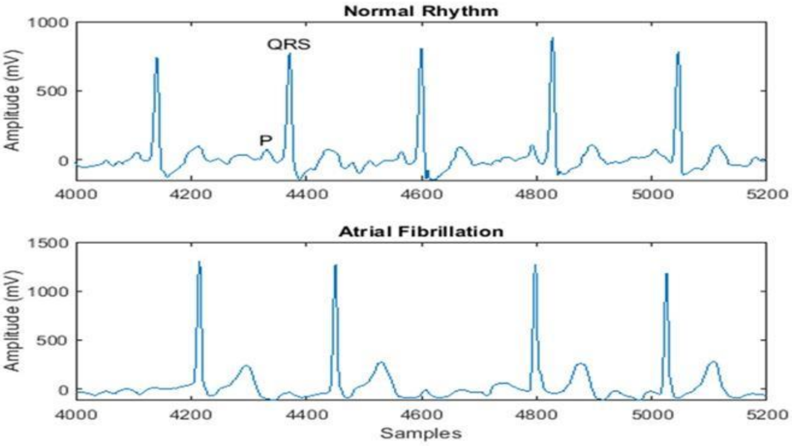


Fig. I.1. the shape of an EMG signal.

I.3.2 Electrocardiogram

The electrocardiogram (ECG) is the representation of the electrical activity of the heart; this heartbeat is controlled by the stimulator cell known as the sinus node (SA).

The PQRST and sometimes U waves constitute an ECG cycle (Figure I.2): the P wave is seen when the SA node triggers the impulse which propagates in the atria and creates the atrial contraction; the PQ interval (where isometric segment) is the duration of the propagation of the pulse from the atrium to the ventricles, allowing blood flow to flow in the same direction; QRS complex occurs when the impulse arrives at the ventricles and reflects ventricular contraction.

The ST segment is the period of depolarization of the ventricles, in the normal case it should be isoelectric, otherwise, the level of amplitude and slope of this segment are indicators of the ischemic state of the myocardium and finally the wave T, which expresses ventricular repolarization, is the state of rest of the ventricles (relaxation). Electrical impulses are recorded through electrodes placed on particular surfaces of the body, using a 12-lead or 12-channel ECG (in hospitals) or 3-channel ECG.

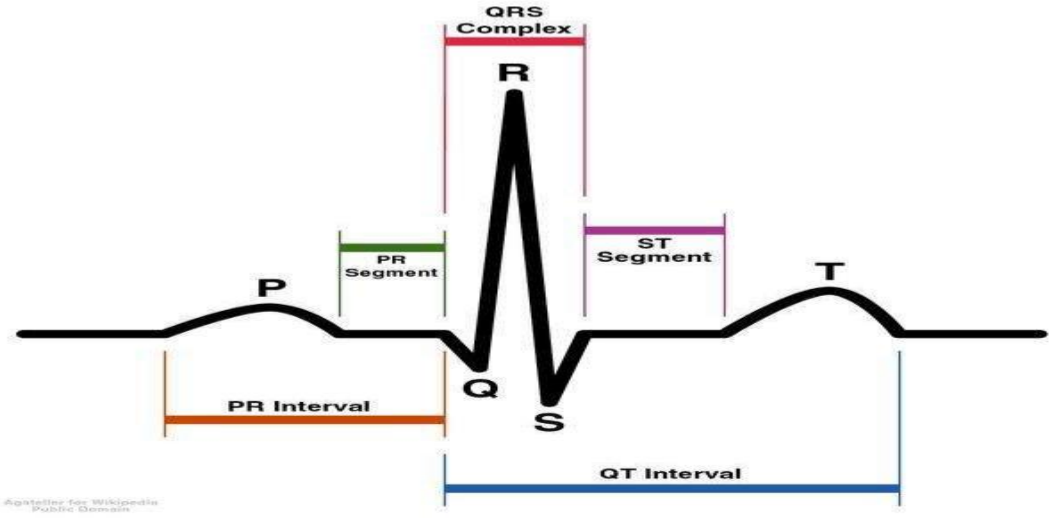


Fig. I.2. ECG signal.

I.3.3 Heart Disease

The number of people suffering from heart disease is increasing. Accurate diagnosis at an early stage followed by appropriate subsequent treatment can save a considerable number of lives. New data released by the National Heart, Lungs and Blood Institute (NHLBI) shows that older women, in particular, are at higher risk for heart disease.

A recent field study of heart disease can be effectively controlled if diagnosed at an early stage. But it is not easy to make an accurate diagnosis due to the many complex factors of heart disease. For example, many clinical symptoms are associated with many human organs other than the heart, and very often heart disease can present with a variety of syndromes.

Due to this complexity, there is a need to automate the medical diagnostic process which can help the doctors in the diagnostic process.

I.3.4. Alzheimer's disease

900,000 people are affected by Alzheimer's disease in France and each year

[23] . memory impairment, difficulty in locating oneself in time and space, progressive loss of language and autonomy… are some of the characteristic symptoms of neuro generative diseases such as Alzheimer's disease.

In our brain, neurons are organized into a network. these cells transmit nerve impulses and information. In Alzheimer's disease, this brain tissue is damaged

In the brain of patients, there are three characteristics of the disease:

- Loss of neurons
- The formation of so-called amiloid plaques, due to the accumulation of certain proteins (proteins Tau) that make up the internal structure of neurons.

All of these lesions progress through the brain year after year. They begin by affecting the most internal structures of the brain, including the hippocampus which is dedicated to memory. This is the reason why the disease usually manifests itself initially as memory impairment.

The lesions then reach the posterior and outer areas of the brain which treat the more global information, linked to gestures and language. Hence language disorders (called aphasia). Patients no longer manage to name objects (this is agnosia), they no longer recognize things, have difficulty coordinating their gestures and carrying out concrete gestures (what is called apraxia)… Ultimately, the brain of patients ends up completely atrophying.

When the first signs of Alzheimer's disease appear, the process has already been at work for several years.

These signs often include disturbances in memory, thinking, judgment, unreasoning, language and behavioral disturbances (agitation, aggression, insomnia, depression, anxiety, paranoia, visual hallucinations) .At a more advanced stage, the person becomes dependent, no longer recognizes his relatives and runs the risk of falls and accidents.

Diagnosis is primarily based on observing the signs and interviewing the family.

A neurological examination and tests evaluating the memory as well as the capacities to cope with daily activities are also carried out. Blood test, electroencephalogram, lumbar puncture, scan of the skull, Doppler of the vessels of the neck are sometimes necessary to rule out a curable cause of dementia.

Advancement to screen for Alzheimer's disease, we use neuropsychological tests, medical imaging to visualize the area responsible for memory in the brain, or lumbar puncture to detect biological markers of the disease [2].

I.3.5 Diabetes disease

Diabetes is an abnormal rise in blood sugar. Defined by the level of sugar in the blood. This increase in blood sugar can cause damage to various organs in the longer or shorter term. Like the eyes, kidneys, nerves and blood vessels. Almost 90% of diabetics live for years with this disease without knowing it because diabetes (or sugar diabetes even if the term is a little longer) is mentioned when the fasting blood sugar level is greater than or equal to 1.20 g / l. it is advisable to check this figure a second time to have two blood sugar tests [3].

I.3.6 Cancer screening

Screening consists of a series of medical examinations, repeated at regular intervals, which discover the possible presence of a disease well before it begins.

à cause noticeable abnormalities (symptoms). Only certain cancers are currently affected by screening. Screening can detect certain cancers (or certain anomalies precancerous) long time before the appearance from first symptoms. Its objective is clear:

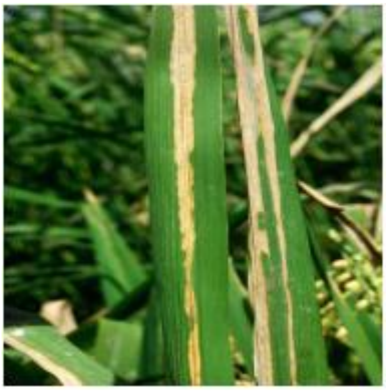
- prevent cancer by treating precancerous lesions.
- increase the chances of recovery (or allow a much less aggressive treatment) thanks to an early management of the disease.

Screening is currently only possible for a limited list of cancers. Other screenings are not the subject of a scientific consensus, even if some patients resort to them after discussion with their attending physician.

I.3.7 Rice plant disease: causes, symptoms, preventive measures.

Rice production requires the adoption of certain cultural practices that prevent a multitude of diseases, most of which are capable of decimating an entire crop. The pathologies affecting rice are indeed very numerous and can be of a fungal, bacterial or viral nature, caused respectively by parasitic fungi, bacteria, viruses. For our work, we will focus on three types of diseases.

>> Bacterial leaf blight (BLB): Bacterial leaf blight



Cause: Xanthomonas cam-pestris pv. Oryzae = X, oryzae pv, oryae

Symptom:

• on seedlings, the infected leaves first turn straw yellow, then wilt and
die;

• in mature plants, light green to greyish green, water-soaked streaks appear on the leaves;

• formation of larger yellowish lesions with irregular edges

• at the final stage of infection, a milky bacterial ooze flows from the leaves

• the drops of milk dry up later and leave a white coulter

Preventive measures :

• Plant resistant varieties of rice available in your area.

• control the irrigation water so as not to flood the fields

• Destroy and remove weeds that could serve as alternative hosts

• Burn crop residues after harvest

• Adjust nitrogen fertilizer applications. Use only healthy or certified seed

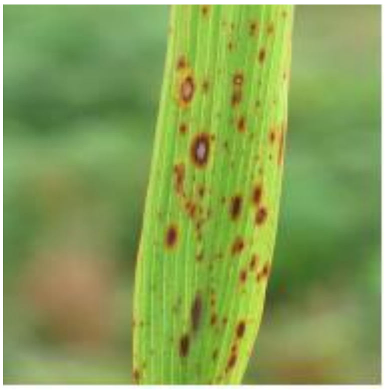
• Handle seedlings with care during transplanting

• Ensure adequate drainage of fields and nursery to avoid cross contamination.

• Adjust nitrogen fertilizer applications to avoid excess and distribute nitrogen applications when weather conditions are favorable.

• Allow fields to dry out between seasons to control pathogens in the soil and plant residues.

>> Brown Spot: Brown spots



Cause: Cochliobolus miyabeanus

Symptoms

- Presence of circular or oval brown spots with a yellow halo hanging

tillering stage;

- a gray center develops in the center of the spots and a reddish-brown border becomes visible;
- discoloration of the stems
- In susceptible varieties, lesions can be as long as 5–14mm and cause leaf wilting.
- on resistant varieties, the lesions are yellow-brown and the size of a head

pin

Preventive measures

- For soils poor in silicon, apply supplements containing silicate of

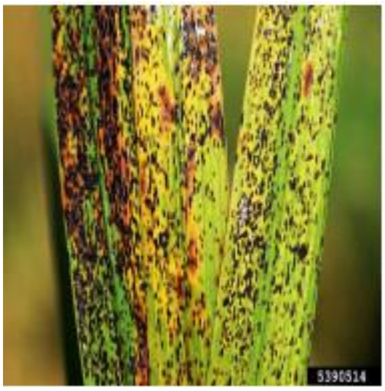
calcium before planting. If possible use seeds from certified sources;

- Ensure a balanced intake of

nutrients and regularly monitor the nutrients in the solution;

- Monitor fields for disease symptoms early in the tillering stage.

>> Leaf Smut: Leaf smut



Cause: Entyloma oryzae: minor fungal disease

Symptoms :

- presence of small spots on the leaves
- they are slightly raised and angular and give the leaves the appearance of

been sprinkled with ground pepper

- the tips of some of the most infected leaves may die
- the blackened spots are approximately 0.5 to 5.0 mm long and 0.5 to 1.5 mm wide
- heavily infected leaves turn yellow and the leaf tips die and turn gray

Preventive measures :

- Plant resistant varieties of rice available in your area.
- good nutritional balance
- control the irrigation water so as not to flood the fields
- Adjust nitrogen fertilizer applications to avoid excess and distribute nitrogen applications
- Destroy and remove weeds that could serve as alternative hosts
- Burn crop residues after harvest

I.4 ARTIFICIAL INTELLIGENCE

It is a scientific discipline relating to the processing of knowledge and reasoning with the aim of enabling a machine to perform functions normally associated with human beings. Artificial intelligence attempts to reproduce human cognitive processes in order to perform "intelligent" actions. It is like "the construction of computer programs which engage in tasks which are so far more satisfactorily performed by human beings because they require high level mental processes such as:

- Perceptual learning.
- The organization of memory and critical reasoning.

The ISO 2382-28 standard defines artificial intelligence as “the ability of a functional unit to perform functions generally associated with human intelligence, such as reasoning and learning”. Qualified as the next computer revolution, artificial intelligence is at the heart of all current affairs, it seems essential to define this disruptive technology and to clarify its legal regime, but also to identify the applications in progress or in development in businesses and the benefits they derive from them.

I.4.1.History

It is interesting to review the origins and history of artificial intelligence in order to fully understand its first directions and its prospects for the future:

- 1950: Alan M. Turing, mathematician and pioneering computer science theorist, launches the concept of artificial intelligence.
- 1955-1956: Launch of the first artificial intelligence program by Allen Newell, John C. Shaw and Herbert A. Simon, Logic Theorist.
- 1957: Modeling of chess games.

- 1958: John McCarthy invents Lisp (list processing), an interactive programming language (development at MIT).
- 1958: Construction of the first neural network, the Perceptron, by Frank Rosenbaltt, a so-called connectionist machine.
- 1959: Development of the first GPS (general problem solver) - end of the first period of artificial intelligence.
- 1970: Neoconnectionism.
- 1989: DeepThought, IBM supercomputer, two million strokes per second.
- 1990 - 1997: Development of Deep Blue renamed DeeperBlue: design of a system of 256 processors operating in parallel, each processor can calculate approximately three million strokes per second.
- 2009: MIT launched a project to rethink artificial intelligence research.
- 2011: Watson, IBM's supercomputer wins two of the three rounds of the Jeopardy game show! The performance consisted for this artificial intelligence in answering questions of general culture.
- 2013: Human Brain Project. Google opens a research laboratory on the premises of NASA.
- 2014: DeepKnowlegdge Ventures: appoints VITAL to its board of directors, an algorithm capable of making decisions by analyzing the balance sheets of potentially interesting companies, clinical trials, intellectual property and previous investments.
- 2015: Facebook Artificial Intelligence Research (FAIR). Google is making its TensorFlow artificial intelligence technology accessible to everyone. Development of a fear that artificial intelligence will ultimately exceed the performance of human intelligence.
- 2016: Amelia from IPSoft a virtual agent. Also, AlphaGo defeats the champion of the game of Go, Lee Se-Dol three times in a row in five innings.

I.4.2 Multiple applications of artificial intelligence

Artificial intelligence has already found multiple uses within society:

- **Search Engines:**All search engines (including travel agencies) are based on intelligent systems of extraction, analysis, and classification of data to produce as quickly as possible a result relevant to the user's request. This is how Google implemented a system using machine learning techniques for its search engine in October 2015, called RankBrain9. This system converts large amounts of text into mathematical vectors to help the system guess the meaning of unfamiliar words or phrases, and thus process the 15% never-before-done requests it receives each day.

- **The recommendation engines:**By relying on data from a user's browsing and purchases, sites like Amazon or Netflix are able to offer them other similar products that might be of interest to them. These predictive technologies are also used for online advertising platforms (Google, Criteo) to offer visitors content from advertisers related to the pages they have visited.

- **Automatic translation:**It is based on statistical modeling algorithms of natural language. They incorporate the construction rules for each language.

- **Personal assistants (Siri, Cortana, Google Now ...):** They are deployed on

  Smartphones that rely on several technological building blocks: voice recognition to convert sound into text, natural language to understand the meaning of words, a search engine to find the answer to the question and speech synthesis to communicate the answer to the question. user, scheduling for event management, etc.

- **Conversational agents:** They are used in the areas of customer support and telemarketing and consist of chat windows that open by themselves on a website, or voice server that answers questions 24 hours a day. They use natural language and their access to large databases allows them to answer the simplest questions.

- **Autonomous vehicles:**While some prototypes are already rolling on the roads in contact with other vehicles, cars that park on their own or brake in anticipation are already a reality. The automatic piloting of airplanes or the management of trajectory of space vehicles, or drones are also based on artificial intelligence.

- **GPS navigation systems:** Developed in 1968 by the Stanford Research Institute, this algorithm makes it possible to optimize the path between several points in a network based on the cost of the path or the distance traveled.

- **Finances :**They are managed by intelligent systems to organize their operations, invest in the stock market and manage their assets, but also to identify transactions that are out of the ordinary. Banks also have expert systems for assessing risks linked to the granting of loans (credit-scoring).

- **Cyber security:**Cyber security players have adopted machine learning techniques in order to detect abnormal behavior in information systems, and to detect persistent threats to avoid espionage or data extraction operations10. Nearly 300 parameters (hours and IP of connections and machines, downloads, etc.) are taken into account to establish the behavioral analysis model, the first learning phase of which lasts about a week. Let us mention the young Lyon-based Sentryo, which integrates machine learning algorithms to secure critical industrial sites.

- **Video games :**They use artificial intelligence techniques to bring non-player characters to life or to create entire universes from algorithms. In 1997, IBM's supercomputer DeepBlue defeated reigning world chess champion Garry Kasparov. In 2016, it was DeepMind, Google's artificial intelligence program, that announced the victory of its AlphaGo program against the reigning European go champion, Fan Hui13. This result is based on the technology of neural networks, which we described previously. Deepmind had already developed an artificial intelligence system capable of determining the most judicious action to beat the man in twenty arcade games.

- **Medicine :**IBM's Watson computer aims to compile the largest healthcare database in the world, covering 300 million patients. IBM's health arm has acquired four medical companies since its inception: Phytel (health & population), Explorys (clinic health files), Merge Healthcare (medical imaging) and Truven (medical analytics). This supercomputer has notably proven its worth in an oncology department for personalized services:

analyzing DNA, patient records and other publications and clinical trials. The computer offers a protocol adapted to the patient. The American company Enlitic offers artificial intelligence technologies to analyze medical imaging, which would make it possible to detect fractures more efficiently than radiologists according to the company.

I.4.3 Future of artificial intelligence

We have just seen that artificial intelligence is already a reality for multiple applications. But in view of the research being carried out around the world, it still has a bright future ahead of it. The applications of artificial intelligence for the future can go even further, particularly in the field of: autonomous cars, robotics, connected buildings and medicine.

I.5 WHAT IS AUTOMATIC LEARNING OR MACHINE LEARNING?

Machine learning is an AI discipline that offers computers the possibility of learning from a set of observations called a learning set [4].

Each observation, such as for example "I ate such and such foods at such and such time of day during such period which caused such disease" is described by means of two types of variables:

- The first are called the predictor variables (or attributes or characteristics), in our example my age, my medical history, my medical history. These are the variables from which we hope to be able to make predictions. The n predictive variables associated with an observation will be noted as a vector x = (x1 ... xn) with n components. A set of M observations will be made up of M such vectors x (1) ... x (M) [4].
- A target variable whose value we want to predict for events not yet observed. In our example, this would be the contracted disease. we will denote by y this target variable [4].

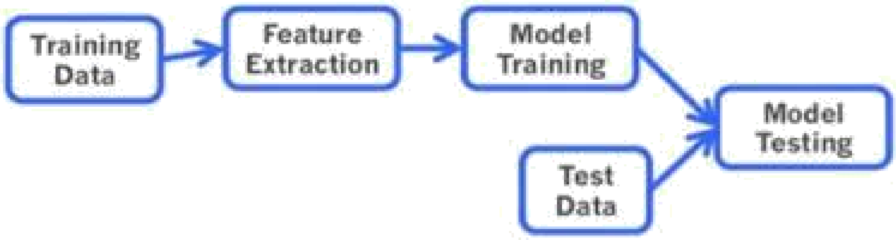


Fig. I.3. the typical machine learning process

In summary, the value of the variable y depends on:

- A function F (x) determined by the predictive variables.
- A noise ε (x) which is the result of a number of parameters which cannot be taken into account.

Both F and ε will never be known but the objective of a model of ML is to obtain the best possible approximation of F from the available observations. This approximation will be denoted f, we call it the prediction function [4].

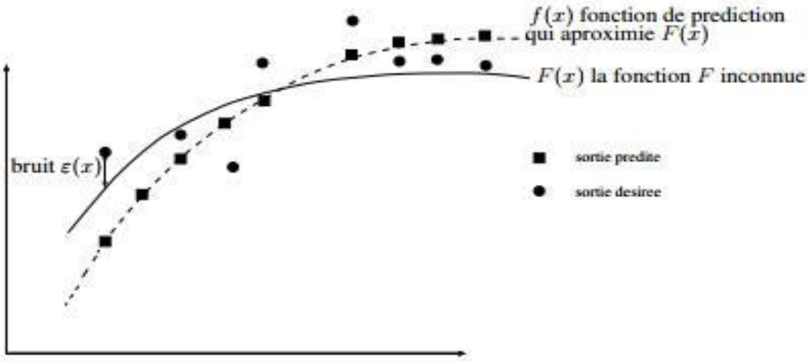


Fig I.4. a model of ML which tries to obtain the best possible approximation of F

Here are some examples of the use of ML [4]:

- Computer vision
- Fraud detection
- Classification (image, text, video, sound ...)
- Targeted advertising
- Medical diagnosis

I.5.1 Learning

Learning is the mechanism by which the nervous system adapts to the outside world. The precise mechanisms of learning by the brain remain very largely misunderstood and involve both genomic modifications partly under the control of OTX2 (methylations of citosines, retrotransposons), neuronal modifications (increase in the exchange surface of synapses, the number of post-synaptic receptors or the number of pre-synapatic neurotransmitters) but also the microenvironment where the astrocytic feet also play a key role.

Machine learning is most likely fundamentally different from learning in the brain. Indeed, today the main learning techniques are said to be supervised, namely that we provide the algorithm with ground truths. Our brain seems to learn in a largely unsupervised way. One of Tom Michels' historical definitions of machine learning said that a system was intelligent if for a task T measured by a performance P, P improves at the task T with the experiment E. Recent advances in parallel computing have made it possible in many visual applications to match or even exceed human capacities for the task assigned to them. However, the current stake is the transferability of this knowledge and to move from so-called discriminative networks to generative networks by integrating them into representations larger than simple tasks. Machine learning is done essentially by successive iterations, seeking to minimize the overall error of a model measured by its cost function. This type of problem refers to all optimization techniques and more particularly to minimization. One of the minimization techniques used in machine learning in our work is CNN Convolutional Neural Networks.

I.5.2. Machine learning and its biomedical applications

Artificial intelligence researchers always aim to program machines capable of performing tasks that require intelligence. However, programming machines capable of adapting to all situations and possibly evolving according to new constraints is difficult. The challenge is to get around this difficulty by providing the machine with learning capabilities allowing it to benefit from its experience. This is why, in parallel with research on automatic reasoning, research on machine learning in English "machine learning" has developed. The main objective of his research is the automatic resolution of complex problems by making decisions based on observations of these problems.

The use of machine learning for biomedical applications is experiencing a dramatic increase. This renewed interest has several causes. On the one hand, the successful application of machine learning techniques in different fields On the other hand, the most recent development is the advent of electronic medical records.

I.6. CATEGORIES OF LEARNING MACHINES I.6.1. Separation machines

I.6.1.1. Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are very powerful nonlinear binary classification algorithms.

The principle of SVM is to build a nonlinear separator band of maximum width that separates two sets of observations and to use it to make predictions. the trick of SVMs to achieve this is to use a nonlinear transformation ϕ which sends the points x (1), ..., x (M) of the original space to n dimensions (n is the number of predicative variables) to new points ϕ (x (1)), ..., ϕ (x (M)) in a space of dimension greater than n or they will be easier to separate

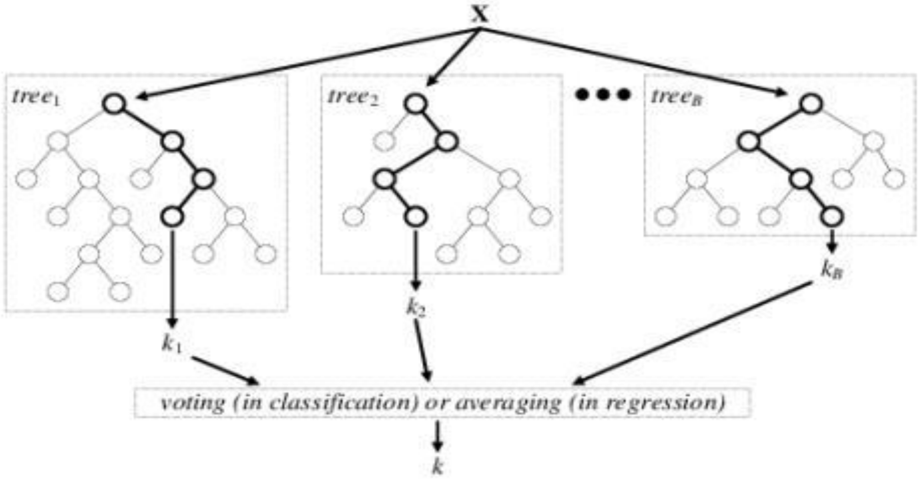


Fig I.5. Creation of B boostrap from the learning examples, each one will be used to train a decision tree. If it is a case of classification, the final class is assigned by majority vote. If it is a case of regression, we average the predictions

SVMs are classifiers based on two key ideas:

The first idea consists in finding a linear separator of maximum width, it is the notion of maximum margin. The margin is the distance between the separation boundary and the closest samples. These are called support vectors. The problem is to find this optimal dividing border.

If the problem is linearly separable, the choice of the separating hyperplane is not obvious. There is in fact an infinite number of separator hyperplanes, the performance of which in the learning phase is identical, but the performance of which in the test phase can be very different.

To solve this problem, it has been shown that there is a single optimal hyperplane, defined as the hyperplane which maximizes the margin between the samples and the separating hyperplane.

There are theoretical reasons for this choice. Vapnik has shown that the capacity of the separating hyperplane classes decreases as their margin increases.

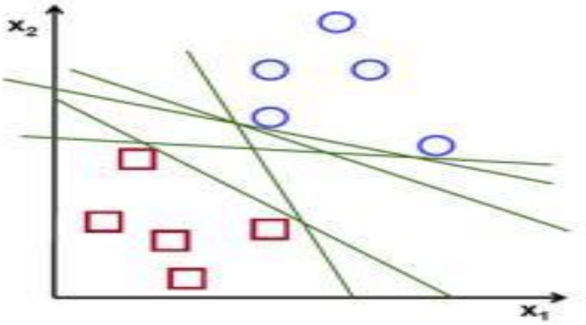


Fig I.6. We are looking for a hyperplane which divides the observations into two categories. Let us consider an example x that we want to classify, if f (x)> 0, it belongs to the class of circles, otherwise it belongs to the class of squares. In this figure we can see that there is an infinity of possible separating hyperplanes.

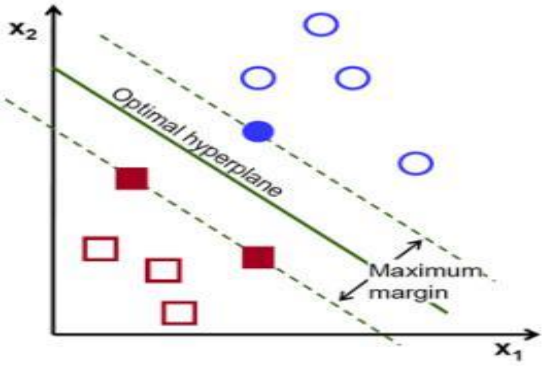


Fig I.7. The optimal hyperplane (in green) with the maximum margin. Filled samples are support vectors

In order to be able to deal with cases where the data is not linearly separable, the second key idea of SVMs is to transform the representation space of the input data into a space of greater dimension, in which it is likely that it there is a linear separation. This is achieved thanks to a kernel function, which must respect the conditions of Mercer's theorem, and which has the advantage of not requiring the explicit knowledge of the transformation to be applied for the change of space.

Kernel functions make it possible to transform a dot product in a large-dimensional space, which is expensive, into a simple one-off evaluation of a function. This technique is known as the kernel trick.

The two most used kernel functions are the polynomial kernel and the Gaussian kernel.

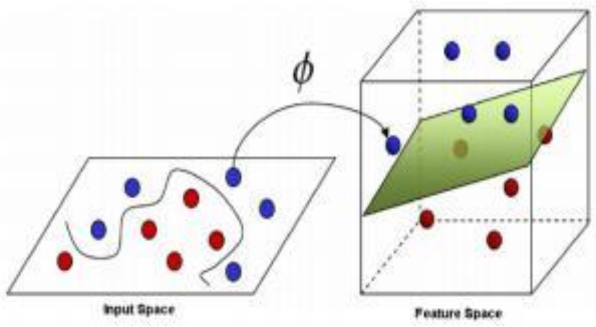


Fig I.8. Example of a nonlinearly separable problem. The curve becomes a linear band after applying the nonlinear transformation ϕ

SVM can deal with complex nonlinear classification problems and also they constitute an alternative to neural networks because they are easier to train but SVMs are often less efficient than random drills [4].

I.6.1.2 Neural Networks:

Neural networks provide a simulation of the functioning of the nerve cell to

Using an automaton.

Neural networks are made up of a set of neurons (nodes) connected to each other by links that make it possible to propagate signals from neuron to neuron.

Thanks to their learning capacity, neural networks make it possible to discover complex non-linear relationships between a large number of variables, without external intervention.

As a result, they are widely used in many classification problems (marketing targeting, pattern recognition, signal processing,…) estimation (modeling of complex phenomena,…) and forecasting (stock market, sales,…).

There is a trade-off between model clarity and predictive power.

The simpler a model, the easier it will be to understand, but the less it will be able to take into account too varied dependencies.

I.6.1.2.1. Presentation of some types of neural networks:

There are many types of neural networks, each type being developed for a particular purpose [5].

- **Formal Neuron:**

A formal neuron is a mathematical and computational representation of a biological neuron. The formal neuron generally has several inputs and one output which correspond respectively to the dendrites and to the emergence cone of the biological neuron (starting point of the axon) [5].

The excitatory and inhibitory actions of synapses are represented, most of the time, by numerical coefficients (synaptic weights) associated with the inputs [5].

The digital values of these coefficients are adjusted in a learning phase. In its simplest version, a formal neuron calculates the weighted sum of the inputs received, then applies to this value an activation function, generally nonlinear [5].

The final value obtained is the output of the neuron [5].

The formal neuron is the elementary unit of artificial neural networks in which it is associated with its fellows to calculate arbitrarily complex functions, used for various applications in artificial intelligence [5].

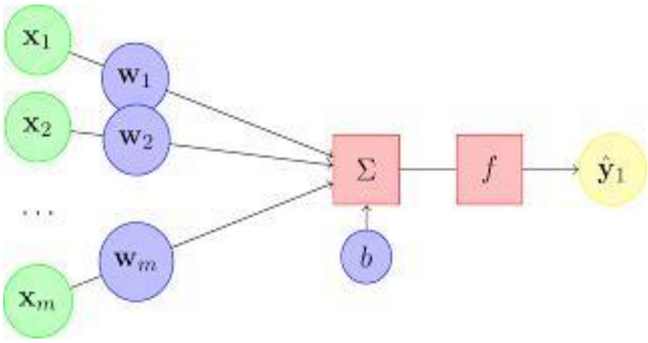


Fig I.9. The figure represents the functioning of a formal neuron.

- **Multilayer neurons:**

The multilayer perceptron (MLP) is a linear classifier of the formal neural network type organized in several layers (Figure I.9) within the

which information flows from the input layer to the output layer only; it is therefore a feedforward type network. Each layer consists of a variable number of neurons, the neurons of the output layer always corresponding to the outputs of the system [5].

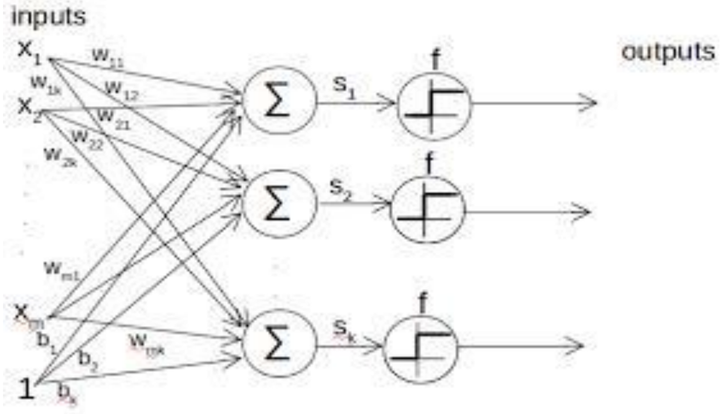


Fig I.10. Multilayer Perceptron

- **Recurrent neurons:**

Recurrent Neural Networks (RNNs) make it possible to analyze vector sequences as well as hidden Markov models. Time is taken into account here because the outputs (of the exit check mark and / or the hidden check mark) are calculated at time t are re-injected at the input of the network and / or at the input of the hidden layer [5].

- **Hopfield Networks:**

The Hopfield neural network is a discrete-time recurrent neural network model whose matrix of connections is symmetrical and zero on the diagonal and where the dynamics are asynchronous (a single neuron is updated at each unit of time) . It was discovered by the physicist John Hopfield in 1982. Its discovery made it possible to revive the interest in neural networks which had run out of steam during the 1970s following an article by Marvin Minsky and Seymour Papert [5].

A Hopfield network is a memory addressable by its content: a memorized form

is

regained by a stabilization of the network, if it has been stimulated by an adequate part of this form [5].

- **Convolutional Neural Networks:**

In machine learning, a convolutional neural network (or neural network

à convolution, or CNN or ConvNet) is a type of acyclic artificial neural network wherein the pattern of connection between neurons is inspired by the visual cortex of animals. Neurons in this region of the brain are arranged so that they correspond to overlapping regions when tiling the visual field. Their operation is inspired by biological processes, they consist of a multilayer stack of perceptrons, the purpose of which is to preprocess3 small amounts of information [5].

Convolutional neural networks have wide applications in the recognition of an image and video vector (uni, doble and trible dimension), recommendation systems and natural language processing. These characteristics ensure that it is one of the techniques whose application can be very suitable for the classification of biomedical data, as we will see in chapter 2 where we will explain the CNNs in detail [5].

I.6.1.2.2.Types of problems that can be solved by neural networks

Neural networks can solve two main types of problems:

- The problems of classifications, where we try to predict a class, a category. We classically distinguish binary classifications (two classes to be predicted), "multi-class" classifications where several classes are to be predicted [6].

TEXT
class A
class A
Classification

Fig I.11. Example classification.

- Regression problems, where we try to predict a continuous function [6].

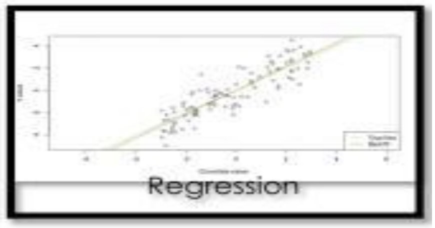


Fig I.12. Regression example.

This distinction is important because for each type of problem, the performance evaluation of this network will be done in a different way. This performance measure, also called the "cost-function" function, reflects how much the neural network is in error for the task assigned to it. For example, in the context of classification problems, log-loss 1.3 or cross-entropy is conventionally used.

(1-3)

J is the cost function, M represents the number of classes to predict, yo, c, a binary indicator equal to 0 or 1 depending on whether the class c is correctly predicted for the observation o etpo, c the predicted probability for each observation classes o. This kind of cost function thus penalizes particularly the results which are wrong and are sure of themselves [6].

I.6.2. Modeling machines

I.6.2.1. Gaussian mixture model

A Gaussian mixture model (MMG) is a category of probabilistic model that states that all generated data points are derived from a mixture of finite Gaussian distributions with no known parameters. The parameters of the Gaussian mixing models are derived from a maximum a posteriori estimate or from an iterative anticipation-maximization algorithm from a well-trained prior model. Gaussian mixture models are very useful for modeling data, especially data from multiple groups.

Gaussian mixture models are used in biometric systems, where the parametric model helps to understand the characteristics or measurements associated with characteristics such as the spectral characteristics of the vocal tract.

Gaussian mixing models are also used for density estimation and are considered to be the most statistically mature techniques for classification.

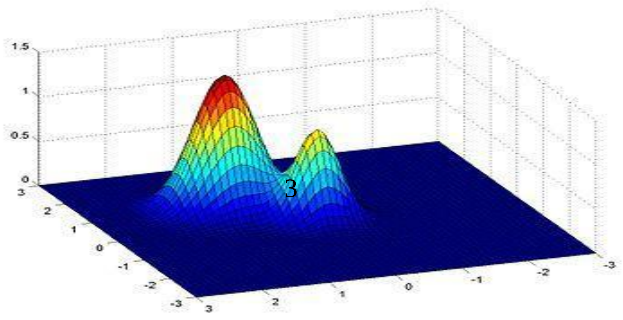


Fig. I.13. The principle of MMG.

I.6.2.2. Hidden Markov chains

Hidden Markov models MMCs are statistical models, rich and widely used in signal processing. They are developed by Andrew Markov (student of Chebyshev), and they are firstly, oriented towards linguistic objectives in works of Russian literature. These are effective tools for modeling sequential data or 'time-series Data'. Later used in speech recognition problems by Baker, their basic theory was introduced by Baum and his colleagues in the late sixties.

Currently, these models are increasingly adopted in automatic speech recognition, for the analysis of DNA sequences, and in problems related to writing and word processing. Also, their use in vision is vast. They are implemented in image segmentation, face recognition, gesture interpretation, as well as background modeling and recently in video processing.

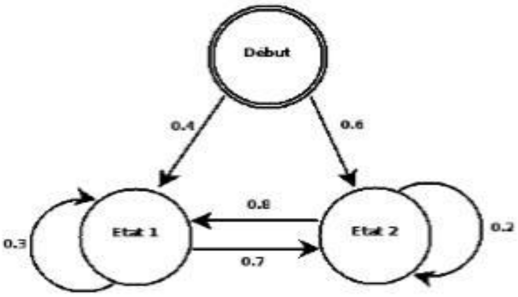


Fig. I.14. The principle of MMC.

I.7 THE DIFFERENT METHODS OF CLASSIFICATION AND LEARNING:

Supervised methods:

In the case of supervised learning, we have a set of labeled data, or examples that have been associated with a class by a teacher or an expert. This set of examples constitutes the learning base [5].

The supervised learning methods then set themselves the general objective of building from the learning base, or classification functions. Such a function makes it possible, from the description of an object, to recognize a particular attribute, the class (Figure I.15.) [5].

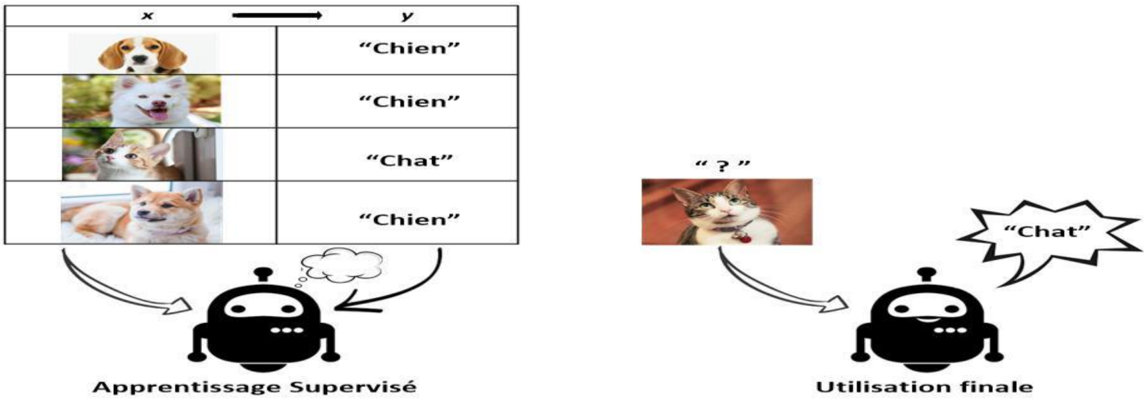


Fig I.15. supervised learning

In [(3), inductive inference is defined as a process which, starting from a specific knowledge observed on certain objects and from an initial inductive hypothesis, makes it possible to obtain an inductive assertion involving or accounting for strongly or weakly observations. In the case of supervised inductive learning, which is a subdomain of inductive inference, specific knowledge consists of a set of objects belonging to known classes. The inductive assertion is expressed by a classification rule which assigns a class to each object. The strong implication is satisfied if the rule correctly classifies all known objects [5].

Unsupervised methods:

Unsupervised learning, also called learning from observations or discovery, consists in determining a "sensible" classification from a set of objects or given situations (unlabeled examples) [5].

We have a mass of undifferentiated data, and we want to know if they have any group structure. This involves identifying a possible tendency for the data to be grouped into classes. This type of learning, also called ING Cluster or Cluster Analyzes, is found in automatic classification and digital taxonomy. This form of classification has existed since time immemorial. It concerns in particular natural sciences (FigI.15), the classifications of documents and books but also the classification of sciences developed over the centuries by philosophers [4].

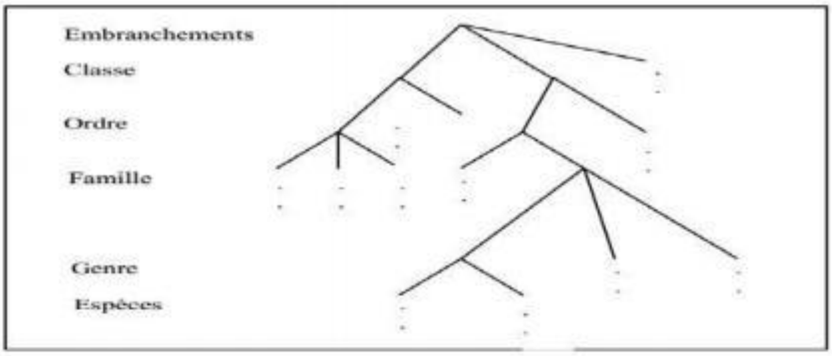


Fig I.16. Extract from the taxonomic classification of Linnaeus.

The automation of the construction of classification constitutes today a real field of research. The key notion used to create object classes is a measure of the similarity between objects. Classes or concepts are constructed in such a way as to maximize intra-class similarity and to minimize inter-class similarity.

Unsupervised learning also corresponds to conceptual classification, where a collection of objects forms a class if this class can be described by a concept, taking into account a set of predefined concepts [5].

Hierarchical methods:

In ascending hierarchical classification. The procedure consists of grouping the individual observations into classes by a part of the same class.

The methods are distinguished by the choice of the distance between the observations and the definition of the aggregation strategy.

In the basic algorithm, the calculation of the distance (it is more exactly a criterion quantity which one calls distance by abuse of language) made by recurrence starting from the matrix of the distances between observations.

Non-hierarchical methods:

The non-hierarchical classification or partitioning, resulting in the decomposition of the set of all individuals into m disjoint sets or equivalence classes, the number m of classes is fixed. The result obtained is then a partition of the set of individuals, a set of parts, or classes of the set I of individuals such as:

- Any class is not empty
- Two distinct classes are disjoint.

· Every individual belongs to a class. This algorithm is called "aggregation around variable centers". A slightly different version, known as "dynamic clouds", is to represent each group not by its center, but by a set of points (nucleus) chosen randomly within each group.

We then calculate an "average" distance between each observation and these nuclei and we proceed to the assignment.

Nonparametric methods:

A classified is said to be nonparametric if no parametric statistical distribution is used, only the spectral distance will then be taken into account. This category includes in particular the methods based on distance minimization (hyper box or parallelepiped, the minimum distance and the Mahalanobis distance, K nearest neighbors, K-means, ISODATA, etc.), new methods have recently appeared. to this category like neural networks and Supported Vector Machines (SVM) [5].

Parametric methods:

A classifier is said to be parametric if it associates with the spectral signature (or profile) a known statistical distribution, most frequently for image processing, the normal or multi-normal law. This association offers the possibility of assigning to each pixel a probability of belonging to a given class [5].

Structural methods: This type of method exploits structural and contextual information of an object, they analyze the object in terms of its components (primitives) and their properties, we find for example the syntactic analysis of a form or an object from of a grammar, the distance of trees, the distance of graphs (isomorphisms of graphs, of subgraphs, with error correction, etc.). In the structural method the class is presented mainly in the form of small round regions [5].

Reinforcement methods : The input data is the same as for supervised learning, however the learning is guided by the environment in the form of rewards or penalties given according to the error made during the learning [5].

Semi-Supervised Method: The input data consists of labeled and unlabeled examples. This can be very useful when you have two types of data, because it allows not to leave any aside and to use all the information [5].

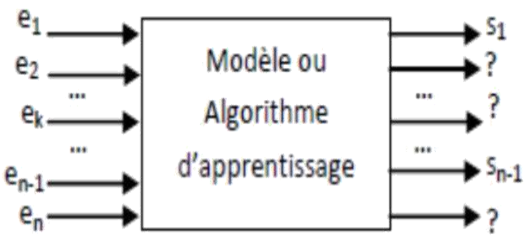


Fig. I.17. Diagram of a semi-supervised model.

I.8 CONCLUSION

In this chapter, we have presented the relationship of our work with medical screening and its importance on human health, we have also seen notions on biomedical and its relationship with learning machines. We have also seen what a learning machine (ML) is and we have shown its relationship with the improvement of treatments and medical decisions (digital screening) by health professionals.

We have thus presented the different categories of learning machines and also the different types of learning. In the next chapter, we will see the principle of Deep learning and Convolutional Neural Networks (CNNs)

Chapter II

System offers

II.1. INTRODUCTION:

In the context of our work, the proposed solution is to recognize the pathology through pattern recognition techniques using deep learning techniques.In this chapter, we will first introduce concepts about learning in depth and then we will start to clarify the differences between deep learning and machine learning. Then we will see what Deep Learning (DL) is and what is its use and how it works, we will also cover the programming part and its general essence of the application of classification.Finally, we will mention in detail most of the DL types it is DNN, CNN, DFFnn and LSTM.

II.2. MACHINE LEARNING VS DEEP LEARNING

A lot of people are worried about AI because they couldn't believe that today's computers can learn and make smart decisions. However, the basics of artificial intelligence are now available to everyone.

ML machine learning and DL deep learning are the two most important concepts that make artificial intelligence possible. These two terms are often confused, even though they designate two very distinct methods used in different fields of application.

ML and DL are part of artificial intelligence as both approaches lead computers to make smart decisions. and deep learning is a subcategory of machine learning as shown in the picture below, the use of ML and DL has become in many areas and conditions, in which the machine can take Intelligent decisions similar to those made by humans, both technologies definitely require quantities of data, which it uses as a basis for learning, the similarity ends there [7].

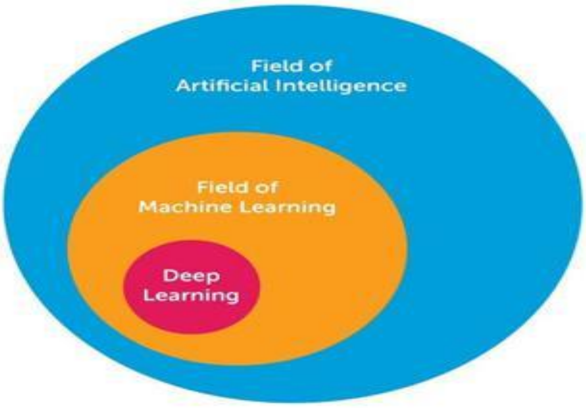


Fig. II.1.the relationship between AI, ML and DL

II.2.1.What are their differences?

Machine learning or ML is the oldest and simplest technique than DL, it depends on an algorithm that adapts the system itself based on the network of the human brain, The application of this technology involves the presence of structured and classified data because the system is fed with this data which allows it to understand how to classify new similar data, Based on this classification, the system then performs the programmed actions, It can for example, determine whether the image shows a dog or a cat, after the first step of use, the algorithm is improved on the basis of the opinions of the developers, which inform the system of incorrect ratings and indicate the correct categories [7].

While deep learning or DL does not require structured data, as the system works on multiple layers of neural networks that combine different algorithms that depend on the human brain. Thus, the system can operate from unstructured data, this approach is particularly suitable for complex tasks, when all aspects of the objects to be processed cannot be classified in the direction of the current. The deep learning system itself defines the discriminatory characteristics. On each layer, you look for a new, specific standard for the object, which serves as the basis for deciding on the classification of the object's retention at the end of the process.

In addition, the technology required for deep learning is more complex. It requires more computing resources and is much more expensive than machine learning, so it is not attractive, at least for the moment, for group use by companies.

In addition, deep learning requires a lot more data, the system must have more than 100 million entries to get reliable results, but ML works when there is a small amount of data to get reliable results [7]. In the table below

Table II.1 [7].

We have put the most important differences between ML and DL techniques:

In terms of	***Machine learning***	***Deep Learning***
Organization of data	Structured data	Unstructured data
amount of data	high performance on a small and medium dataset	works great on a big data
Engineering features	carefully understand the features of how it represents the data	necessary to understand the best functionality specific which represents the data
coaching	Human training necessary	Learning system autonomous
algorithm	Algorithm editable	Neuronal networks algorithms
Necessary material	works on a low machine range	preferably requires a machine with GPU
execution time	from a few minutes to a few time	it takes a long time up to 2-3 weeks
interoperability	some algorithms are easy to interpret as, logistics and the decision tree, and rare where as some are almost difficult like SVM	can be difficult or impossible in most case
Field application	Simple routine actions	Complex stain

Table II.1. Synthesis of the differences between ML and DL

II.2.2.The fields of application

Machine learning can be considered as a precursor technology to Deeplearning. Concretely, all the tasks accomplished using machine learning can be solved

with Deep learning. So deep learning and machine learning should not necessarily be opposed.
Deep learning mobilizes a lot more resources and is therefore not an efficient process. The fields of application of the two technologies are therefore in principle well delimited: any task that machine learning can perform must be handled by this same technique.
For companies, using these technologies represents a huge competitive advantage, because the two types of learning, automatic or deep, are not yet the norm in everyday professional life [7].

II.2.2.1.The fields of application of machine learning

Online marketing: which marketing measures bring results? Humans are generally bad at sifting through large amounts of data and providing reliable estimates. In this case, it is better to use marketing analysis tools that rely on machine learning. They evaluate defined data and can provide reliable diagnoses about the type of content capable of converting, the content customers want to read, and the most effective marketing channels to make a sale [7].
Customer Support: Sculpins can rely on Machine learning. They orient themselves according to the keywords found in the user's question and, through questions to obtain more information or make decisions, dialogue with the user until they provide the desired answer.
Sale: What works for Netflix and Amazon is also great for sale. Using machine learning, systems can accurately anticipate the products and services that might interest customers on their site. They can thus make detailed recommendations, which makes it easier to sell with very wide product lines or highly customizable products.
Business intelligence : Machine learning can also be used to visualize important company data and to make various forecasts understandable for human decision-makers.

II.2.2.2.The fields of application of deep learning

IT security : unlike machine learning-based solutions, IT and cybersecurity systems that rely on Deeplearning can also identify both documented dangers and hitherto unknown risks thanks to their ability to detect abnormalities in known patterns of the neural network. Deep learning increases the effectiveness of security measures.

Customer Support: Deeplearning-based chatbots understand people's natural expression and aren't limited to using specific keywords. Dialogue is clearly more effective and the solutions proposed are more likely to meet demand.

Content creation: Deeplearning can be used to automate content creation. From a sufficiently large content database, the system can create new content or perform translations independently.

Voice assistant: digital assistants, like Siri, Alexa or Google, rely on Deeplearning. The first digital assistants are starting to appear in business. Users can express themselves naturally to request, for example, to abandon an order, send an email, create a report or start a search.

In addition to the fields of application mentioned, the two technologies are used in many areas of everyday life, such as medicine, science or mobility [7].

II.3 WHY DEEP LEARNING?

The ML algorithms described in Part 1 work well for a wide variety of problems. However, they failed to solve some major AI problems such as speech recognition and object recognition.

The development of deeplearning was motivated in part by the failure of traditional algorithms in such AI task.

But it was only after larger amounts of data became available, thanks in particular to Big Data and connected objects, and that computing machines had become more powerful, that we could understand the real potential of Deep Learning.

One of the big differences between Deep Learning and traditional ML algorithms is that they adapt well, the greater the amount of data provided the better the performance of a Deep Learning algorithm. Unlike many classic ML algorithms which have an upper bound on the amount of data they can receive sometimes called a "performance plateau", Deep Learning models do not have such limitations (theoretically) and they have even gone so far. to exceed human performance in areas such as image processing [8].

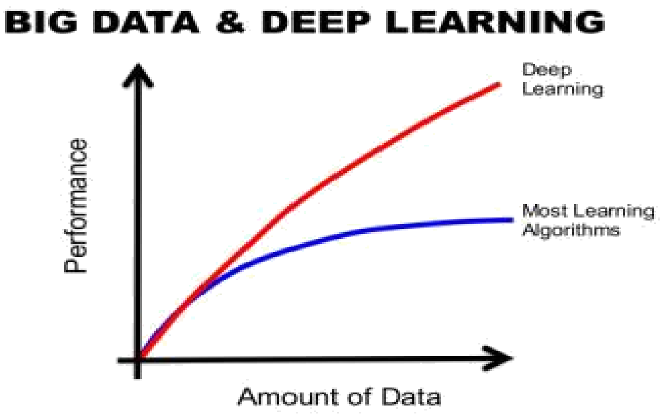


Fig. II.2. The difference in performance between Deep Learning and most ML algorithms depends on the amount of data

Another difference between traditional ML algorithms and Deep Learning algorithms is the feature extraction stage. In traditional ML algorithms the feature extraction is done manually, it is a difficult and time consuming step and requires a specialist in the field while in Deep Learning this step is performed automatically by the algorithm [8] .

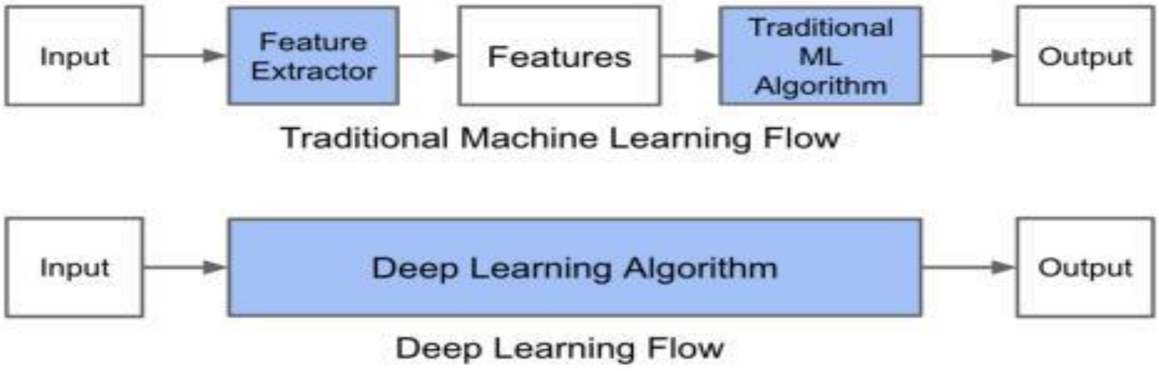


Fig. II.3. The process of classic ML compared to that of Deep Learning

Deep learning provides a very flexible, (almost?) Universal, and learnable to represent global information, visual and linguistic, by deeplearning Can learn both unattended and under supervision.

II.4. DEFINITIONS OF DEEP LEARNING

Deep learning is a type of machine learning (ML) and artificial intelligence (AI) that mimics how humans acquire certain types of knowledge, and it's an important part of data science, which includes from predictive statistics and modeling, and is also very useful for scientists tasked with collecting, analyzing and interpreting large amounts of data, faster and more easily.

In its simplest form, deep learning can be seen as a way to automate predictive analytics. While traditional machine learning algorithms are linear, deep learning algorithms are stacked in a hierarchy to increase complexity and abstraction, allowing them to perform nonlinear and more complex tasks [9].

II.4.1.How Deep Learning works

Computer programs that use Deep Learning go through the same process as a young child's learning to identify the dog. Each algorithm in the hierarchy applies a nonlinear transformation to its input and uses what it learns to create a statistical model for the output. Iterations continue until the output has reached an acceptable level of precision. The number of processing layers the data must pass through is what inspired the <<deep>> [9].

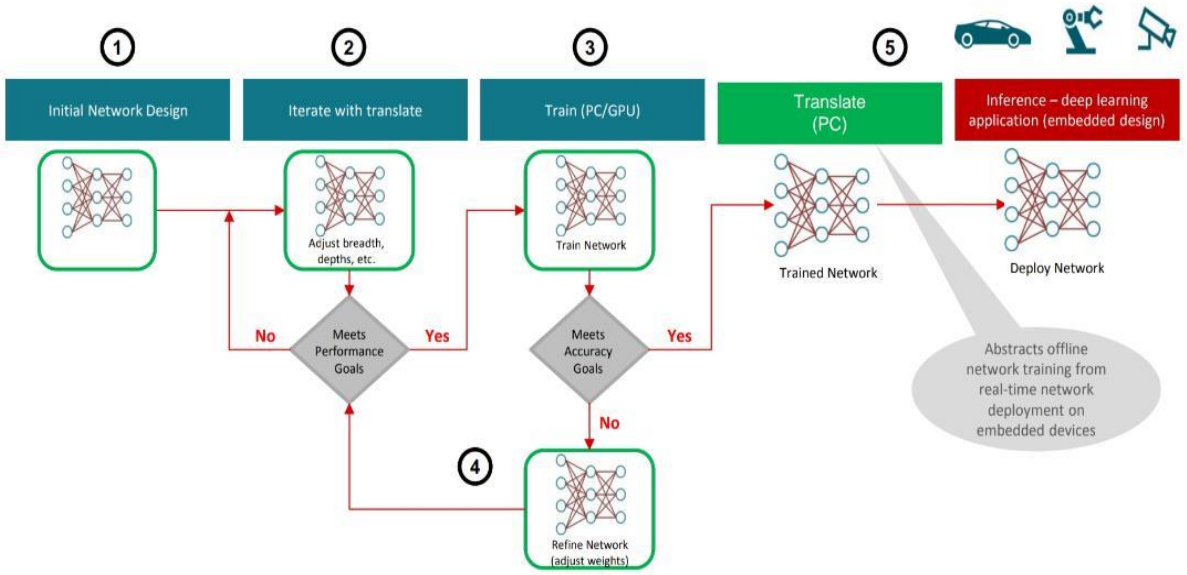

1
2
3
5
Initial Network Design
Iterate with translate
Train (PC/GPU)
Translate (PC)
Inference – deep learning application (embedded design)
Adjust breadth, depths, etc.
Train Network
Trained Network
Deploy Network
No
Meets Performance Goals
Yes
Meets Accuracy Goals
Yes
No
Abstracts offline network training from real-time network deployment on embedded devices
4
Refine Network (adjust weights)

Fig. II.4. Pattern recognition system.

In traditional machine learning, the learning process is supervised, and the programmer has to be extremely precise when telling the computer what kind of things to look for in deciding whether an image contains a dog or not. This is a laborious process called feature extraction, and the success rate of the computer depends entirely on the ability of the programmer to precisely define a set of features.

characteristics for "dog". The advantage of Deep Learning is that the program itself builds the feature set without supervision. Unsupervised learning is not only faster, it is also generally more precise.

Initially, the computer program may receive training data - a set of images for which a human has tagged each image "dog" or "not dog" with meta-tags. The program uses the information it receives from the training data to create a set of "dog" characteristics and build a predictive model. In this case, the model that the computer creates in the first place can predict that anything in an image that has four legs and a tail should be labeled "dog." Of course, the program is not aware of the “four legs” or “tail” labels. It will just look for pixel patterns in the digital data. With each iteration, the predictive model becomes more complex and more precise.

Unlike the young child, who will take weeks, if not months, to grasp the concept of "dog", a computer program that uses deep learning algorithms can be shown a training set and sort through millions of images, accurately identifying images that contain dogs within minutes.

To achieve an acceptable level of accuracy, deep learning programs require access to immense amounts of training data and processing power, which were not readily available to programmers before the era of big data and cloud computing. Because deeplearning programs can create complex statistical models directly from their own iterative results, they are able to create accurate predictive models from large amounts of unlabeled and unstructured data. This is important as the Internet of Things (IoT) continues to become more ubiquitous, as much of the data humans and machines create is unstructured and unlabeled [9].

II.4.2. What are neural networks for deep learning?

A type of advanced machine learning algorithm known as artificial neural networks is the basis of most deep learning paradigms, so deep learning can sometimes be referred to as neural learning. deep or deep neural network [9].

Neural networks come in many different forms, including recurrent neural networks, convolutional neural networks, artificial neurons and neural networks for early action, each with specific use cases (we'll see the explanation for each later). However, they all work in a somewhat similar fashion, populating the data, and leaving the form to determine for itself whether it provided the right interpretation or made the right decision for a particular piece of data.

- In the case of the application (classification), we usually have the path of our computer program written by all algorithms (DL) from start to finish as follows: • List of libraries used
- View our data
- Data Division (training and testing)
- Training of our model (creation of a trained model)
- tests
- Result (precision and loss with curve plotted for each)

Neural networks include a process to test the amount of error resulting from the decision of the network as such, so it needs huge amounts of data to train in it.

It is no coincidence that neural networks only became popular after most companies adopted in-depth data analytics and the accumulation of large amounts of data since the early iterations of the model included assumptions some. Little trained on the content of an image or parts of speech, the data used must be classified during the training phase so that the model can see if its hypothesis is correct. This means that while many companies that use large amounts of data have a large amount of data, unorganized data is less useful because unorganized data can only be analyzed through a deep learning model after training. and an acceptable level of precision,

II.4.3. Deep learning methods

Various methods can be used to create powerful deep learning models. These techniques include reducing the learning rate, transferring learning, scratching training and dropping out.

- **A decrease in the learning rate:** the learning rate is an excessive parameter - a factor that determines the system or corrects the conditions of its work before the learning process, where it controls the size of the change to which the model is subjected in response to the estimated error each time the model weights are changed. Too high learning rates can lead to unstable learning processes or a suboptimal weight group, and very low learning rates can lead to a long training process that can get clogged.

- **The method of lowering the learning rate:**Also called softening the learning rate or adaptive learning rate - involves adjusting the learning rate to increase performance and reduce training time. The simplest and most common adjustments to the learning rate during training include techniques to reduce the learning rate over time.
- **Learning transfer:**This process involves mastering an already created model; Requires an interface from an existing network. First, users feed the existing network with new data containing labels that were previously unknown. Once changes are made to the grid, new tasks can be implemented with more specific classification capabilities. This method has a feature that requires much less data than the others, thus reducing the computation time to a few minutes or hours.
- **Training from scratch:**This method requires the developer to collect a large collection of disaggregated data and build a network architecture capable of learning the functionality and the model. This technology is particularly useful for new applications, as well as for applications that have a large number of output categories. In general, this is a less common approach, as it requires excessive amounts of data, causing training to take days or weeks.

To improve the result (accuracy and loss), we generally tried to adjust the following factors:

- Amount of data for conducting the training (modification of the division between training and test)
- The number of layers in our model
- The number of neurons in each layer
- Epochs (number of training sessions
- the value of Batch_size (display the rotation speed)

II.4.4. Applications of deep learning

Because deep learning models process information similarly to the human brain, they can be applied to many tasks that people perform. Deep Learning is currently used in most image recognition, natural language processing, and speech recognition software. Thus, by deeply training facial recognition algorithms on 200 million face images, the Google Company's FaceNet system achieves a correct identification rate of 99.63%. The number of potential applications of deep learning is immense. This is the reason why this method of learning has imposed itself in recent years. These techniques make it possible to improve image recognition in general and to create applications for biometrics (recognition of fingerprints or iris), medicine (with, for example, the diagnosis of melanoma from images. moles and X-ray analysis), the autonomous car (recognition of obstacles, vehicles, traffic signs, etc.), for example. They also make it possible to improve speech recognition, with systems such as Siri, or the profiling of individuals, for recommendation and targeted advertising, or even game software, as we saw in March 2016 when the AlphaGo computer program won out over Lee Sedol, one of the world's top go players, using deep learning and reinforcement learning. Last but not least, supervised learning techniques help to anticipate the future on the basis of the past, which makes it possible to assess, with a precision unknown before, the risks of investments, accidents, illnesses, etc. However, prediction helps to make decisions by calculating the most likely consequences of each action. As a result, predictive systems using deep learning play an increasingly important role in the contemporary world where they are used to decide in delicate situations instead of men. prediction helps in making decisions by calculating the most likely consequences of each action. As a result, predictive systems using deep learning play an increasingly important role in the contemporary world where they are used to decide in delicate situations instead of men. prediction helps in making decisions by calculating the most likely consequences of each action. As a result, predictive systems using deep learning play an increasingly important role in the contemporary world where they are used to decide in delicate situations instead of men.

II.5. DIFFERENT TYPES OF DEEP LEARNING MODELS EXPLAINED

Deep Learning is a class of machine learning techniques that exploit many layers of nonlinear information processing for the extraction and transformation of supervised and unsupervised entities, for the analysis and classification of models. It is consists of many hierarchical layers to process information in a non-linear way, where a lower level concept helps define higher level concepts. Shallow artificial neural networks are unable to handle a large amount of complex data, which is evident in many routine applications such as natural speech, images, information retrieval and other processing applications. human-like information. Deep learning is suitable for such applications. With deep learning, it is possible to recognize, classify, and classify data models for a machine with relatively less effort. Google is a pioneer in the deep learning experiment initiated by Andrew Ng.

Deep Learning offers human-like layered processing over shallow architecture. The basic idea of deep learning is to use hierarchical processing using many layers of architecture. The architecture layers are organized in a hierarchical fashion. After several pre-workouts, the entry of each layer goes to its adjacent layer. Most often, such pre-formation of a selected layer performed in an unsupervised manner.

Deep Learning follows a distributed approach to big data management. The method assumes that the data is generated taking into account many factors, different times and different levels. Deeplearning makes it easier to organize and process data in different layers according to their duration (occurrence), scale or nature.

II.6. DEEP LEARNING ARCHITECTURES

The number of architectures and algorithms used in deep learning is large and varied, in this part we will look at the five most important deeplearning architectures, which have been one of the most used methods in different applications in the world. over the past twenty years and are still used even today and heavily in various fields [10].

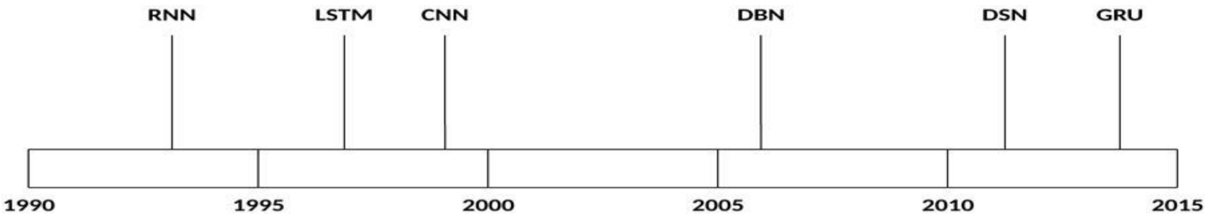


Fig. II.5. Graphical timeline showing the development of the top five deep learning architectures by date, 1990 to 2015

In the following table, we put some applications for each algorithm:

Architecture	***Application***
DNN	General use
DFFnn	computer vision and recognition facial
CNN	Image recognition, video analysis, natural language processing
RNN	in natural language, recognition of handwriting, voice recognition, gesture recognition, subtitling of pictures
LSTM	

Table II.2.some applications for a group of architects

Now let's explore these architectures and methods used to train them.

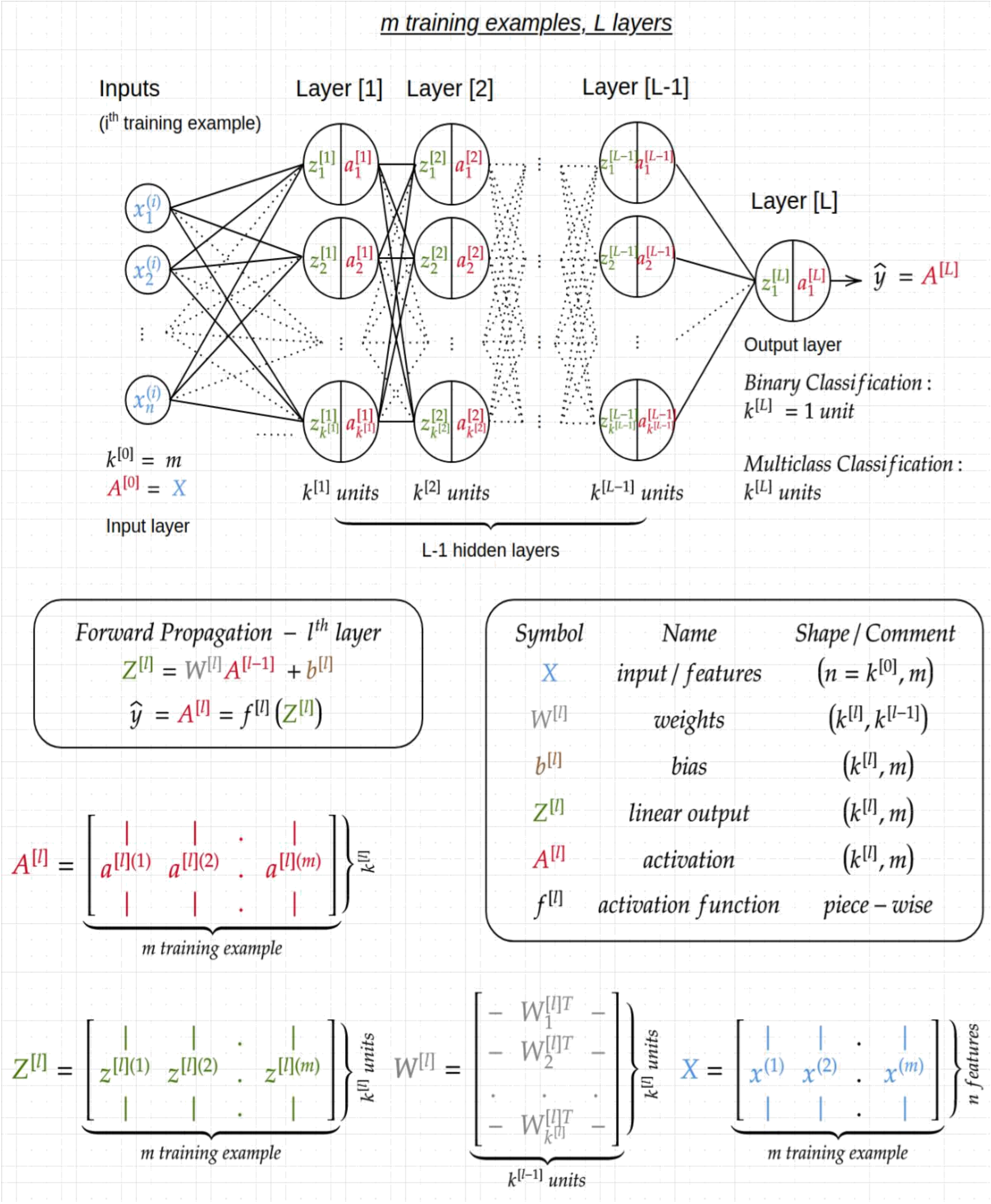
II.6.1. DEEP NEURON NETWORK __DNN__:

These multilayer neural networks can include millions of neurons, divided into several dozen layers. They are used in deep learning to design supervised and unsupervised learning mechanisms.

In these mathematical architectures, each neuron performs simple calculations but the input data passes through several computational layers before producing an output. The results of the first layer of neurons are used as input for the calculation of the next layer and so on. It is possible to play on the different parameters of the network architecture: the number of layers, the type of each layer, the number of neurons that make up each layer, etc.

The multilayer perceptron, capable of processing non-linear phenomena, is an example of this type of network. Concretely, the first layers will make it possible to extract simple characteristics (such as outlines) that the following layers will combine to form increasingly complex and abstract concepts: assemblies of outlines in patterns, of patterns in parts of objects, of parts from object to object, etc.

The hope is that the more we increase the number of layers, the more neural networks learn complicated, abstract things, corresponding more and more to the way a human reasons. However, it is difficult to develop effective learning mechanisms for each of the intermediate layers (called deep layers or hidden layers). The solution is to combine supervised learning (as is done for a two-layer network) and unsupervised learning, using examples whose output value is unknown, (inspired by an article in Interstices) [11] .



Forward Propagation – l^{th} layer

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

$$\hat{y} = A^{[l]} = f^{[l]}\left(Z^{[l]}\right)$$

Symbol	Name	Shape / Comment
X	input / features	$(n = k^{[0]}, m)$
$W^{[l]}$	weights	$(k^{[l]}, k^{[l-1]})$
$b^{[l]}$	bias	$(k^{[l]}, m)$
$Z^{[l]}$	linear output	$(k^{[l]}, m)$
$A^{[l]}$	activation	$(k^{[l]}, m)$
$f^{[l]}$	activation function	piece – wise

$$A^{[l]} = \underbrace{\begin{bmatrix} | & | & . & | \\ a^{[l](1)} & a^{[l](2)} & . & a^{[l](m)} \\ | & | & . & | \end{bmatrix}}_{m \text{ training example}} \Big\} k^{[l]}$$

$$Z^{[l]} = \underbrace{\begin{bmatrix} | & | & . & | \\ z^{[l](1)} & z^{[l](2)} & . & z^{[l](m)} \\ | & | & . & | \end{bmatrix}}_{m \text{ training example}} \Big\} k^{[l]} \text{ units} \quad W^{[l]} = \underbrace{\begin{bmatrix} - & W_1^{[l]T} & - \\ - & W_2^{[l]T} & - \\ . & . & . \\ - & W_{k^{[l]}}^{[l]T} & - \end{bmatrix}}_{k^{[l-1]} \text{ units}} \Big\} k^{[l]} \text{ units} \quad X = \underbrace{\begin{bmatrix} | & | & . & | \\ x^{(1)} & x^{(2)} & . & x^{(m)} \\ | & | & . & | \end{bmatrix}}_{m \text{ training example}} \Big\} n \text{ features}$$

Fig. II.6.the functioning of a DNN deep neural network.

II.6.2 CONVOLUTIONAL NEURON NETWORKS:

II.6.2.1.History:

In recent years, convolutional neural networks (CNNs) have been widely used in various visual recognition tasks by computer and have achieved good results.

compared to traditional methods.Classification of images is one of the fundamental and important tasks of visual recognition, the CNN method is also applied to other visual recognition tasks (such as object detection, localization of objects). 'objects and semantic segmentation ..etc), its architecture generally derived from the human network architecture in image classification [12].

II.6.2.2.General principle

We will see together how the convolutional network works, by briefly approaching the main mathematical tool hidden behind, as well as all the stages of the analysis made by the algorithm. As human beings we can distinguish images of different shapes and colors, but the computer does not have this ability, on the contrary it has the ability to perform hundreds of calculations in a fraction of a second, this that the human mind is unable to do, The computer sees the universe only by numbers, if we damage them, The image below as humans we see it as an image containing three dogs, As for the computer, it sees it as a three-dimensional matrix nw × nh × nc (nh: height, nw: Width, nc:

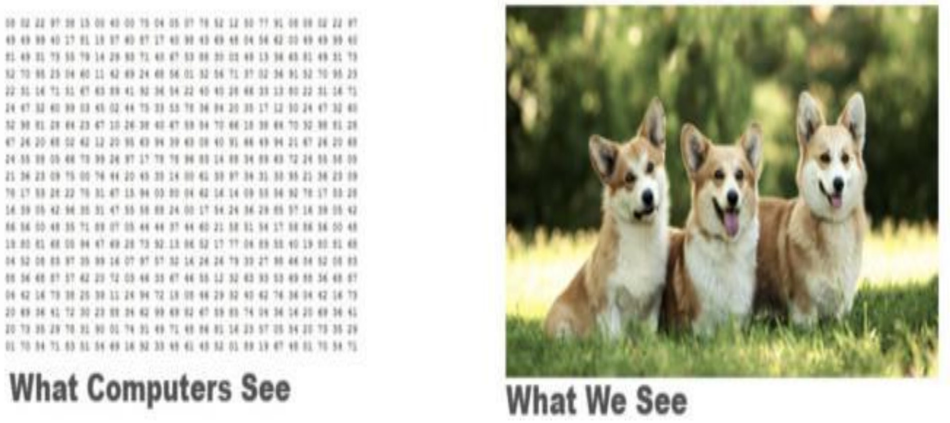


Fig. II.7. Thedifference between computer vision and human vision

Thus, convolutional neural networks will receive this image (Figure 1), and a set of processes will be applied to it to identify the shapes it contains for use in a particular application (Figure 2), such that these processes are divided into two main categories: Filters (extracting information): to bring out, for example, vertical edges, variations in coloring, etc ..., knowing that the more filters intervene later in the sequence, the more they make it possible to detect complex (and more abstract) shapes Simplifications (information analysis): to lighten the calculations and identify the main information Then, it will give the probability that this image represents a robot, a dog, a human… etc, thanks to an artificial neural network ANN [12].

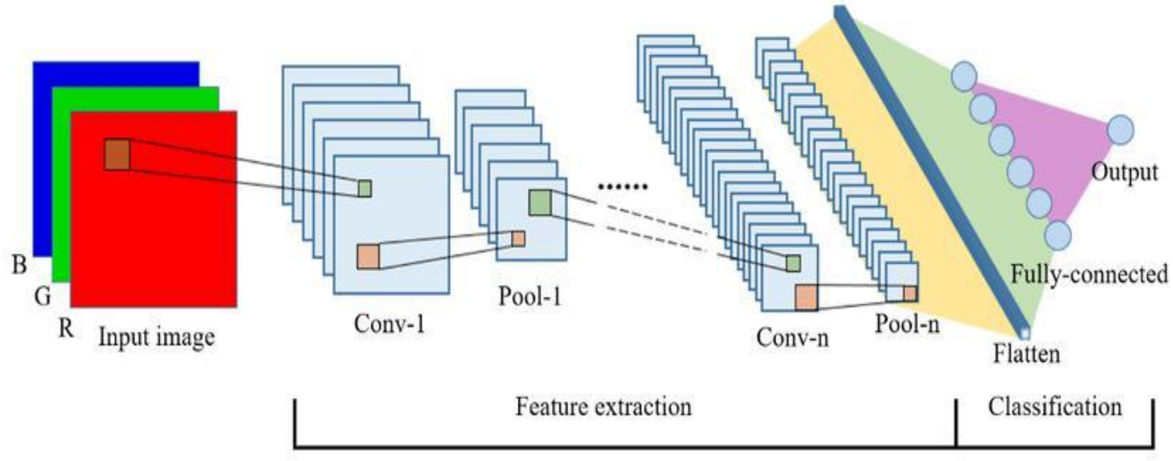


Fig. II.8.the general architecture of CNN

II.6.2.3.The convolution operation:

Convolution is the heart of the convolutional neural network and originally it is a mathematical operation based on multiplying the matrix of numbers with a filter (also known as a kernel), which would determine the presence of certain characteristics. or patterns in the original image (income), then it can be used multiple filters to extract Different functionality and the filter is small to erase the whole image and apply the appropriate calculations between the values of the filter and the color points (pixels) in the image, in order to extract functionality This process results in a new matrix with smaller dimensions compared to the original image [12].

The relation of the output image size:

nh ' : output image size (width).

nh : size of the input image (width).

k : size of the kernel used (width)

p : the applied padding

s : stride applied

The calculation is relatively simple and can be done manually, but the larger the image size, the larger the filter (kernel), the more complex it becomes, which requires computer intervention, Assuming we have a matrix dimensional 3 × 4 and a kernel2 × 2 and the stride = 1, the calculation will be as shown in the image below (figure 03). Then, to calculate the second value of the output image, we will use the stride parameter of the convolution which represents the number of pixels of the input image that we are going to convert to reapply the string. But it can happen that the convolution exceeds the image if we choose the kernel and stride incorrectly, For example if we take the same previous example, but we modify the value of stride to 3,

« padding ”which is used in convolutions and which, simply, adds 0s around the input image, to increase the size of the output image or to avoid overflows [12].

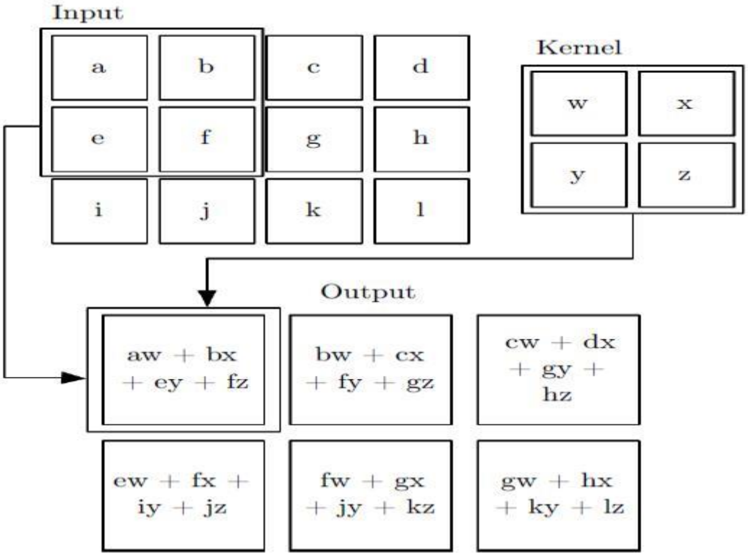


Fig. II.9.how does convolution work

II.6.2.4.The ReLU layer (linear rectification):

The ReLU activation function is the so-called "correction" function widely used in deeplearning, it is rather one of the most used functions nowadays due to the most important characteristics:

1- The fact of not activating all the neurons at the same time also makes it possible to accelerate the calculations because the negative values are set to zero, which means that all values equal to or less than zero become zero, while positive values remain the same.

2- As we have seen, a convolution will perform addition / multiplication operations: the output values are therefore linear with respect to those at the input.

3- in an image, linearity is not very present or important (for example, the variations between pixel values can be important in a region, the image has corners…).

4- By not modifying the positive data, ReLU does not impact the characteristics highlighted by the convolution, on the contrary: it highlights them more by widening the gap (negative values) "between" two characteristics (for example nose and eyes).

Note that there are other ReLUs like for example LeakyReLU which break less the linearity of the data, even if ReLU is very widely preferred and used [12].

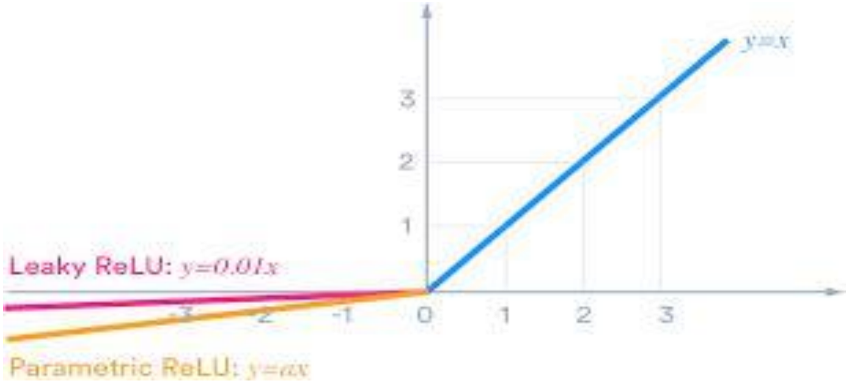


Fig. II.10.the ReLU and LeakyReLU activation functions and the difference between them

II.6.2.5.Pooling:

Pooling is a simple operation that involves replacing a square of pixels (usually 2 × 2 or 3 × 3) with a single value. In this way, the image decreases in size and becomes simplified (smoothed). Not only does this reduce the number of calculations required, but it also prevents overfitting [12].

To apply the pooling, we start by selecting a square of pixels of size 2 × 2 then we calculate the value that will replace this square, Then we shift this square to the right by 1 squares if the stride (= step) is equal to 1 for example (generally, it is worth 1 or 2), Once arrived at the end on the right, one starts again all on the left by shifting once down (of a step equal to the stride) and going again towards the right [13].

There are several types of pooling:

Maxpooling:which amounts to taking the maximum value of the selection. It is the most used type because it is quick to calculate (immediate), and allows to efficiently simplify the image [12].

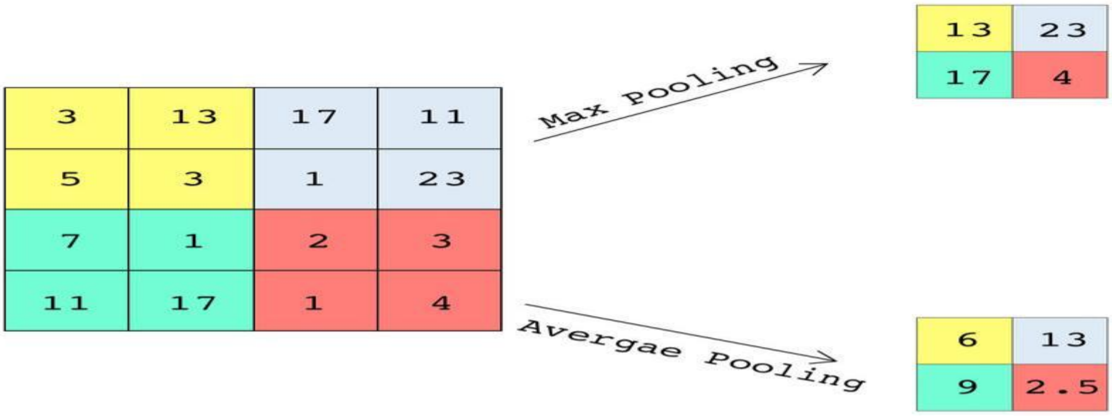
Meanpooling (or averagepooling):or the average of the pixels of the selection: we calculate the sum of all the values and we divide by the number of values. We thus obtain an intermediate value to represent this batch of pixels [12].

Sumpooling: it is the average without having divided by the number of values (only their sum is calculated) [12].

The difference between aux:

the Mean and the sum are almost identical, We can simply note that the sum increases the value of the pixels of the image to widen the gaps between regions of the image which can highlight certain characteristics, while the mean allows to remain limited, while max-pooling will tend to retain the most marked and simple characteristics of the selection of pixels, such as for example a vertical edge. Conversely, mean being an average, only the less marked features will stand out.

In general, it is recommended to use max-pooling, because it differs from mean-pooling in extreme cases and is almost equivalent to mean-pooling in other cases [12].



3	13	17	11
5	3	1	23
7	1	2	3
11	17	1	4

13	23
17	4

6	13
9	2.5

Fig. II.11.the difference between Max pooling and Averagepooling

II.6.2.6.The flattening:

This is the last step of the "information extraction" part. The input to this layer is a matrix which includes information about the locations of specific complex shapes (patterns) in the image, A flattening operation (flattening) makes the input (the matrix) that we have to make it a (long) vector of values, each of these values representing the probability of a class of the set of classifications on which the network is formed and then passed to a neural network to predict the probability of exit [12].

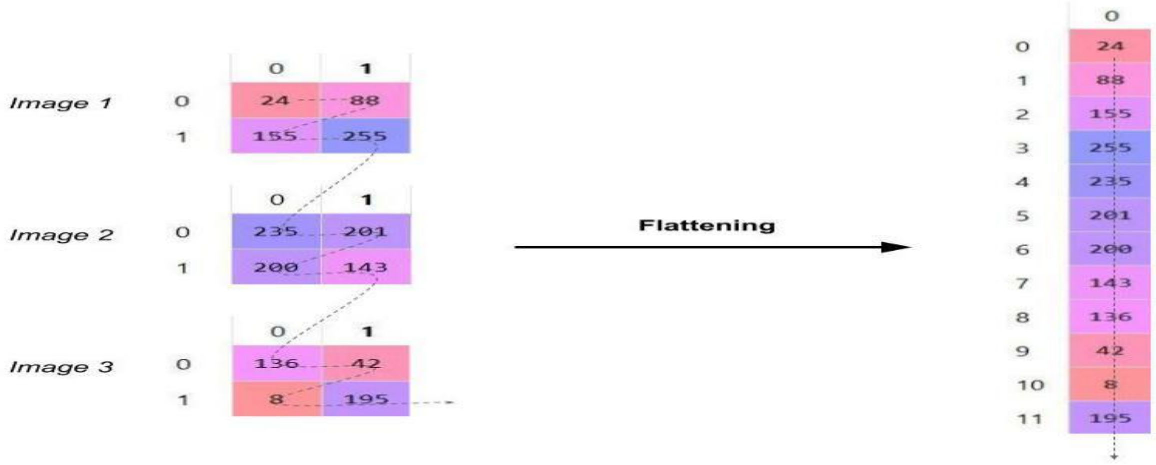


Fig. II.12.how does flattening work

II.6.2.7.fully connected:

Artificial neural network is a network that contains layers called fullyconnectedlayers or dense, these layers that receives a vector of values and returns probabilities for each prediction class, this layer contains neurons, There are different types of these neurons [12]:

the input neurons: which receive the sum of the weighted signals from the flatting couch which send their value to all the neurons of the next layer, In the case of CNN this will be the value of a precise pixel for each neuron.

hidden neurons: organized in layers, which will send the sum of the signals they receive to the neurons of the next layer.

the output neurons:which receive the sum of the weighted signals from the last hidden layer which predict the probability of the output, and Each output neuron then represents a specific prediction. For example, the first neuron represents the prediction "it's a cat", the second "it's a dog" ... etc.

generally the number of neurons in this layer (output layer) is equal to the number of classifications, and that according to the problem to which we are going to find a solution, and the conclusion of our network depends on which output neuron has the strongest signal [14].

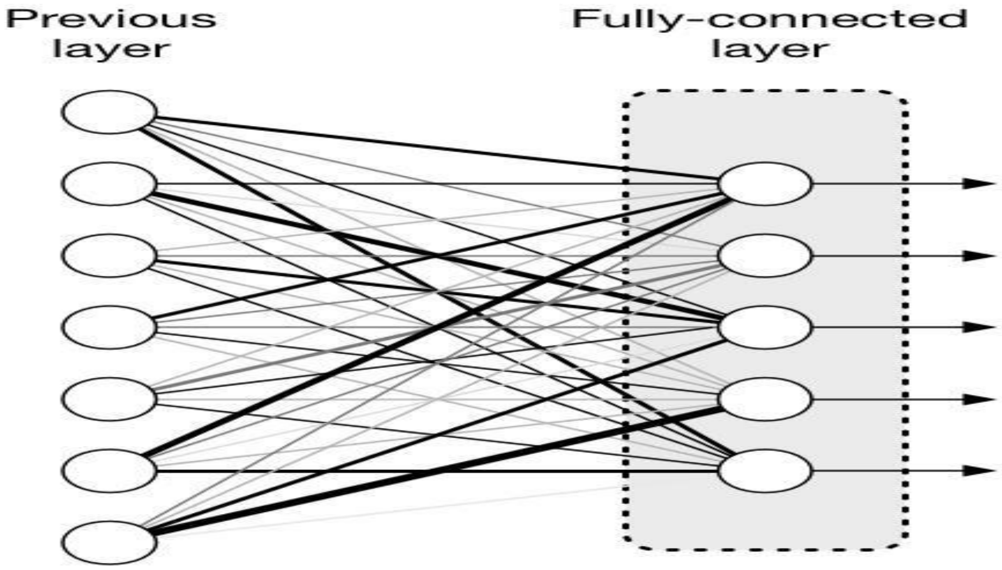


Fig. II.13.the distribution of fully_connceted layers

II.6.2.8. The padding:

Each of the convolution and pooling can make the input matrix smaller and this can make the information disappear from the image and as a solution for this there is the padding method it is is simply to add zeros all around a matrix to increase the image size as shown in the image below (figure 8) [12].

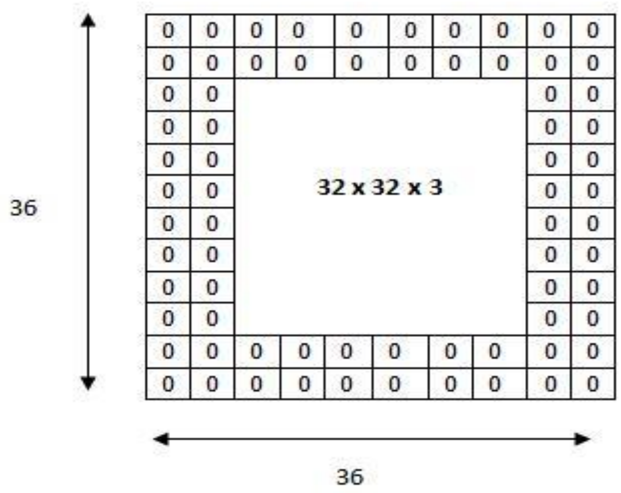


Fig. II.14.a padding of 2 on a matrix of size 32 × 32

II.6.3 DIRECT ACTION NEURON NETWORKS (DFFnn):

II.6.3.1.History:

In 1943, McCulloch and Pitts (1943) proposed a computational model inspired by the human brain, which initiated research on the artificial neural network (ANN), these are able to learn and recognize and can solve a wide range of complex problems, Feedforward Neural Networks (FFNNs) are the special type of ANN models, and its structural representation makes it attractive as it allows to perceive a computational model (a function) in a structure / network, form of plus this structure which makes it a universal function approximator, which has the capacity to approximate any continuous function (Hornik, 1991).

Therefore, Feedforward Neural Networks (FFNNs) can solve a wide range of problems, some of them below:

. like pattern recognition (Jain et al., 2000).

. grouping and classification (Zhang, 2000).

. approximation of the function (Selmic and Lewis, 2002).

. control (Lamand Leung, 2006).

. bioinformatics (Mitra and Hayashi, 2006).

. signal processing (Niranjan and Principe, 1997).

. speech processing (Gorin and Mammone, 1994).

The structure of an FFNN consists of several neurons (processing units) arranged layer by layer and the neurons of one layer have connections (weights) from the neurons of its previous layer [15].

II.6.3.2. General principle:

Deep feedforward networks, also often referred to as feedforward neural networks, or multilayer perceptrons (MLPs), are the quintessential deep learning models, and it's called that because information only flows through the neural network via the input signals, then to the hidden layers, (one or more layers) and finally through the output layer, and there are no feedback connections where the outputs of the model are fed back into itself , when feed-forward neural networks are extended to include feedback connections, but before we talk about feedforward neural networks and how they work, let's understand why we needed these kind of neural networks.

The purpose of a feedforward network is to approximate a function f *, for example, for a classifier y = f * (x) maps an input x to a category y, a feedforward network defines a mapping y = f (x; θ) and learns the value of the parameters θ which give the best approximation of function [16].

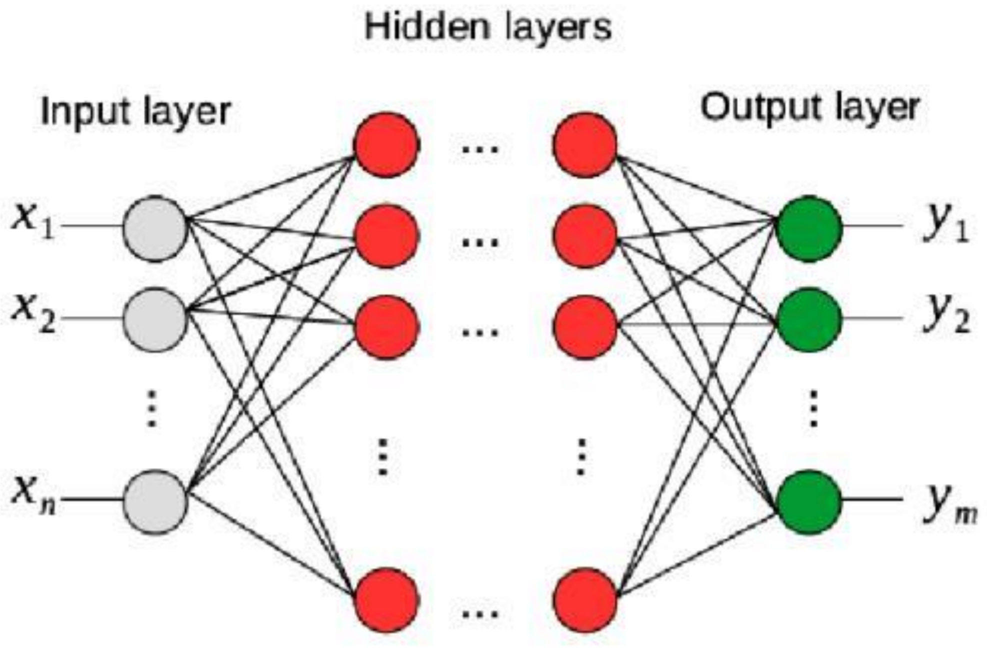


Fig. II.15. the distribution of a Direct acting neural networks

II.6.3.3 The perceptron (neuron):

The neural network was born from a very popular machine learning algorithm named perceptron, they were developed in the 1950s and 1960s by scientist Frank Rosenblatt, inspired by earlier work by Warren McCulloch and Walter Pitts, traditional models like the perceptron (the perceptron is a fundamental unit of the neural network) which takes actual inputs and gives a Boolean output only works if the data is linearly separable.

Then we have the sigmoid neuron model which is similar to the perceptron, but the sigmoid model is slightly modified so that the output of the sigmoid neuron is much smoother than the functional stepwise output of the perceptron. Although we have introduced the nonlinear sigmoid neuron function, it is still not able to effectively separate nonlinear data, but if we connect multiple neurons sigmoid efficiently, we can approximate the combination of neurons to any complex relationship between input and output needed to process nonlinear data [17].

The relationship between input and output in the case of a neuron:

xi: Input signals.

wi: The weights between the input signals and neuron.

b: through a neuron.

f (): the activation function of a neuron.

y: the output (decision).

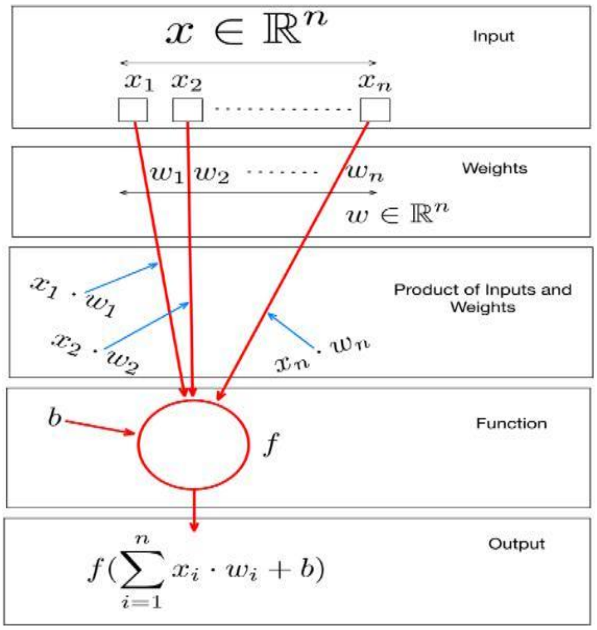


Fig. II.16.how does a neuron work

II.6.3.4 Artificial neural networks:

The multilayer neural network is made up of many sigmoid neurons and it is able to handle nonlinearly separable data, the layers present between the input and output layers are called hidden layers and used to manage the complex nonlinearly separable relationships between input and output [14].

The relationship between input and output in the case of neural networks with a single hidden layer:

xi: Input signals directed towards the invisible layer.

xj: Signals from the invisible layer directed to the output layer.

wi: The weights between the input layer and the hidden layer.

wj: The weights between the hidden layer and the output layer.

bj: through the hidden layer.

bj: through the output layer.

g (): the activation function of the hidden layer.

f (): the activation function of the output layer.

y: the output (decision).

II.6.3.5.Softmax activation function:

Softmax function is the most frequently used activation function in deep learning for classification problems as an output activation function, which makes it better than others in this case is that the output is always positive, whatever the input, by applying the function softmax we would obtain a predicted probability distribution and our real output is also a probability distribution, we can compare these two distributions to calculate the loss of the network [17].

If: j = 1 to k

II.6.3.6. loss_function:

In this section, we will talk about the loss function for binary and multi-class classification. The purpose of this function is to indicate to the model that a correction should be made in the learning process.

In general, the number of neurons in the output layer would be equal to the number of classes, and in the case of a binary classification, we can only use one sigmoid neuron which generates the probability P (Y = 1) therefore we can get P (Y = 0) = 1-P (Y = 1), but in the case of classification we will use the cross-entropy loss to compare the predicted probability distribution and the real probability distribution [17].

The cross-entropy loss for binary classification is given by:

The cross-entropy loss for the multi-class classification is given by:

II.6.4. RNN Recurrent Neural Networks:

RNN is one of the core network architectures from which other deep learning architectures are built. The main difference between a typical multilayer network and a recurring network is that, rather than full feedback connections, a recurring network can have connections that come back to the previous layers (or the same layer). This feedback allows RNNs to retain memory of past entries and to model problems over time.

RNNs consist of a rich set of architectures (next we'll look at a popular topology called LSTM). The key differentiator is feedback within the network, which could manifest from a hidden layer, the output layer, or a combination of these [10].

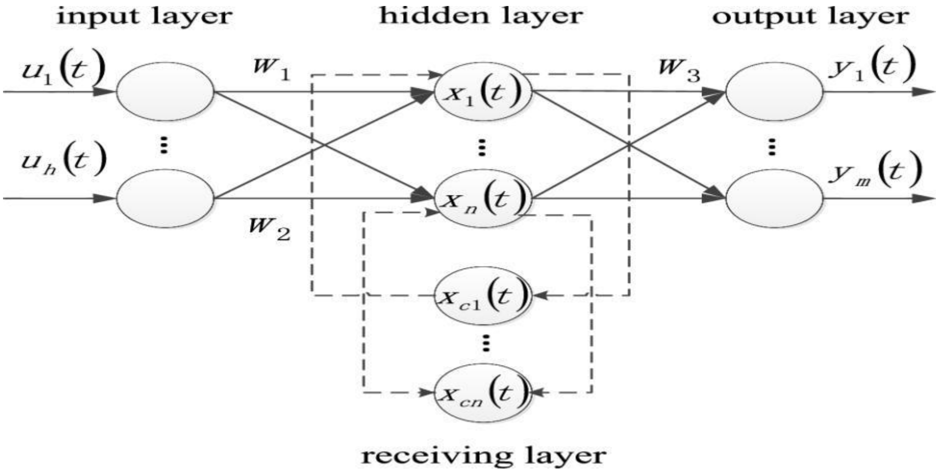


Fig. II.17. Image with circles and arrows demonstrating the interrelationship between input, output, hidden, and context network layers

RNNs can be time-unfolded and trained with standard back-propagation or using a variant of back-propagation called time-back-propagation (BPTT) [10].

II.6.5. LSTM networks:

The LSTM was created in 1997 by Hochreiter and Schimdhuber, but has gained popularity in recent years as an RNN architecture for various applications. You'll find LSTMs in products you use every day, such as smartphones. IBM has applied LSTMs in IBM Watson® for conversational speech recognition setting milestones.

The LSTM moved away from typical neural-based neural network architectures and instead introduced the concept of a memory cell. The memory cell can hold its value for a short or long time depending on its inputs, allowing the cell to remember what is important and not just its last calculated value.

The LSTM memory cell contains three gates that control how information enters or leaves the cell. The front door controls when new information can flow into memory. The forgetting gate controls when existing information is forgotten, allowing the cell to remember new data. Finally, the output gate controls when the information in the cell is used in the output of the cell. The cell also contains weights that control each gate. The training algorithm, usually BPTT, optimizes these weights based on the resulting network output error [10].

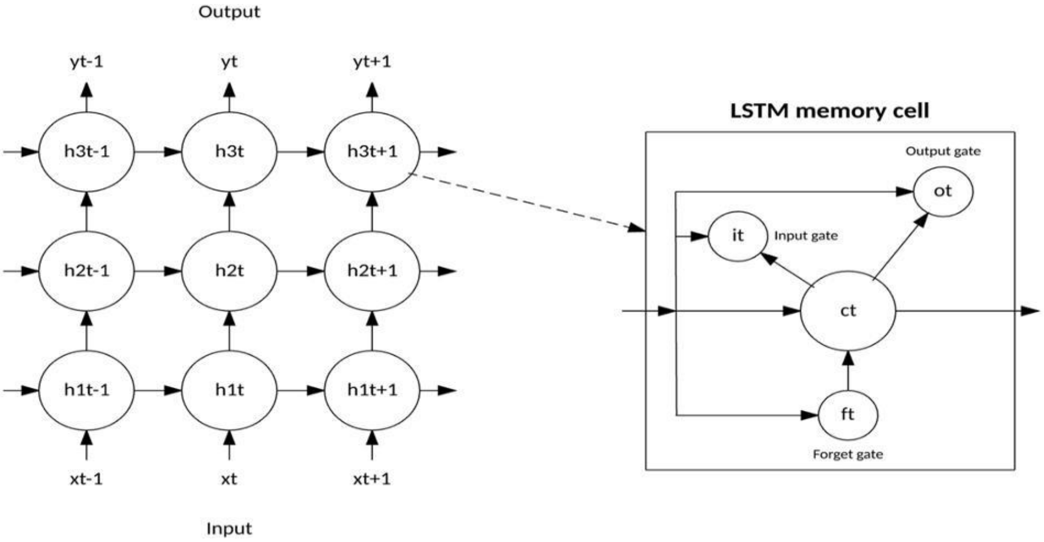


Fig. II.18. Image with circles and arrows showing the LSTM memory cell and the flow of information through the various gates

In 2014, a simplification of the LSTM was introduced, called the closed recurring unit. This model has two doors, getting rid of the exit door present in the LSTM model. For many applications, the GRU has similar performance to the LSTM, but its simplicity means less weight and faster execution.

The GRU has two gates: an update gate and a reset gate. The update gate indicates the amount of previous cell content to keep. The door reset defines how to incorporate the new entry with the previous cell content. A GRU can model a standard RNN simply by setting the reset gate to 1 and the update gate to 0.

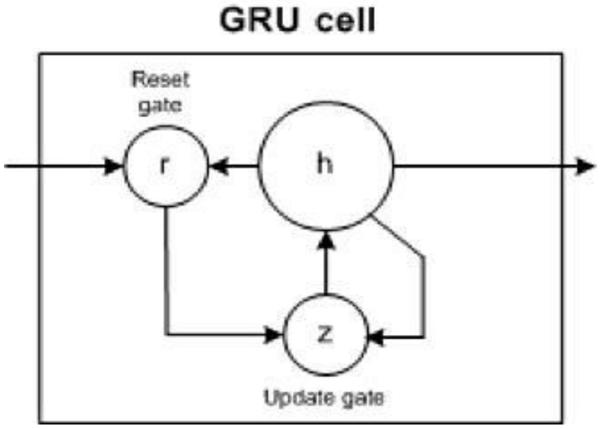


Fig. II.19. Diagram of a typical GRU cell

GRU is simpler than LSTM, can be trained faster, and can be more efficient in its execution. However, LSTM can be more expressive and with more data, can lead to better results [10].

II.7. ACTIVATION FUNCTIONS

They allow each layer of a neural network to operate in a non-linear fashion.

There are a very large number of them and they can be combined.

A judicious combination of appropriate activation functions of the input to output layers improves the extraction of characteristics.

The active sigmoid and reread function is the most used in our work (for

the classification application) and the softmax function when there is more than one class (in our case of Alzheimer's) [22].

Examples of activation functions (represented by g (x) with x = value of the input layer):

II.7.1 Sigmoid

A sigmoid function transforms variables into values ranging from 0 to 1 and it is commonly used to generate a Bernoulli distribution.

II.7.2 Hyperbolic tangent

The derivative g '= (1 - g2) facilitates the procedure with back-propagation algorithms. It is commonly used in the output layer.

II.7.3 Softmax

The Softmax output can be considered as a probability distribution over the categories.

It is commonly used in the output layer.

II.7.4 RectifiedLinear Unit (ReLU)

This activation function (and its variants) is frequently used in deep learning.

Function used for the hidden layers ("hiddenlayers") of neural networks.

Advantages and disadvantages :

- Low computational cost and easy to optimize function.
- Convergence faster than sigmoid or tanh.
- Improves gradient propagation.
- If x <0 => weights not updated => no learning.
- Not differentiable at zero.
- Not zero-centered and unbounded.

II.7.5 Variants of ReLU

Softplus or SmoothReLU:

Other variations of ReLU: LeakyReLUs, Noisy ReLUs, PReLUs ("ParametricReLUs"), ELUs ("ExponentialLinear Unit")

II.8.6 Maxout

The weighting matrix of this function is a 3-dimensional array in which the 3rd dimension corresponds to the connection between neighboring layers [22].

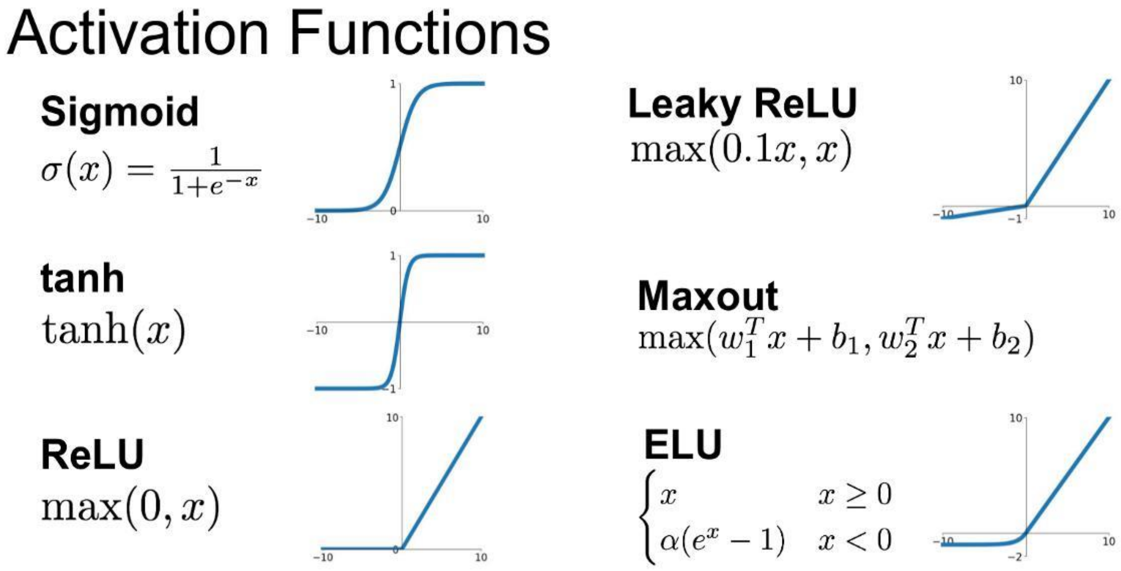
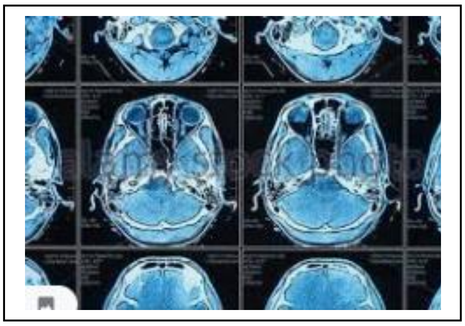


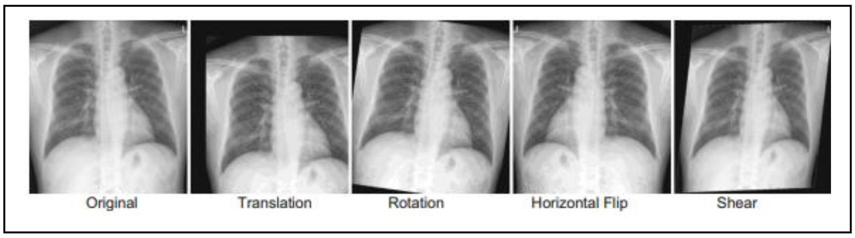
Fig. II.20.representation of different types of activation functions

II.8. CONCLUSION:

In this chapter, different deep learning techniques applied for the classification that constitutes our disease recognition system are highlighted, We have made a comparison between the deep learning techniques and the machine learning techniques that we explained in the previous chapter. Next, we mentioned what makes deep learning unique to use instead of machine learning techniques, and we covered in detail what the most important deep learning algorithms are and how they work.

The following chapter is entirely devoted to simulation, and is dedicated to the evaluation of the performance of the deeplearning techniques that we have studied in this chapter, and the objective of this study is the contribution of these methods for their application. in the field of disease identification, where many experiments will be carried out to maximize the performance of the system and choose the technique which has the highest performance.




Original
Translation
Rotation
Horizontal Flip
Shear

III.1. INTRODUCTION

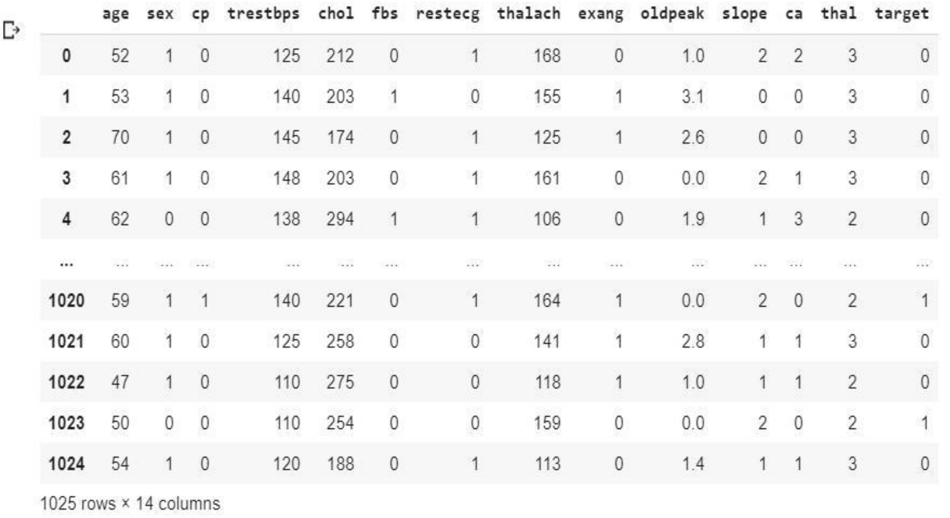
This last chapter is specifically dedicated to the implementation and evaluation of our system using many deep learning techniques (DNN, DFF, LSTM) and this by applying them to different real biomedical databases (heart disease , breast cancer, diabetes, Alzheimer's, rice disease, etc.).

III.2. DATABASE USED

In order to properly evaluate our proposed system, there are four different databases used to extract the overall performance, These data come from real biomedical sources, here is a description of the properties of each database.

III.2.1. Heart Disease

This dataset dates from 1988 and consists of four databases: Cleveland, Hungary, Switzerland and Long Beach V, It contains 76 attributes, including the predicted attribute, but all published experiments refer to the use of 'a subset of 14 of them. The "target" field refers to the presence of heart disease in the patient. It has an integer value: 0: no disease and 1: disease.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

Fig.III.1. the form of cardiac patient data

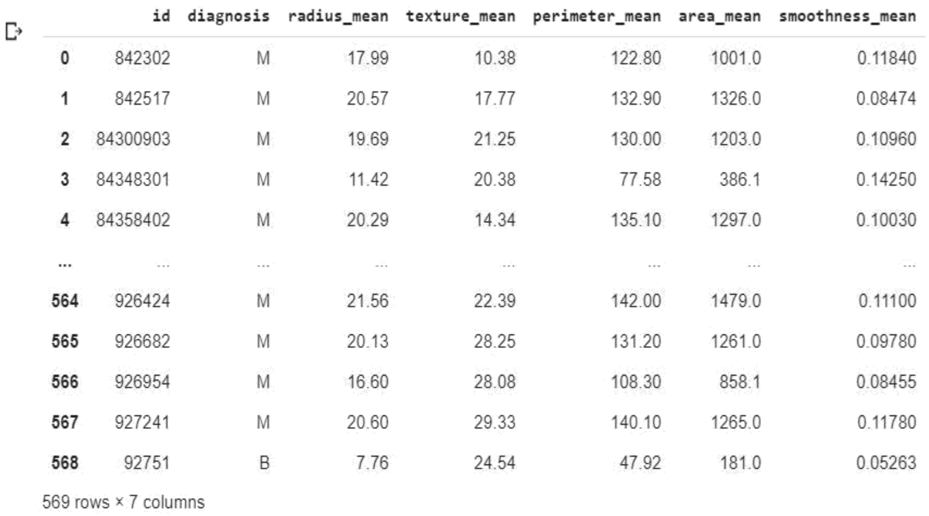
Parameter no.	Abbreviation	meaning
1	Age	age in years
2	sex	1 = male; 0 = female
3	cp	type of chest pain
4	trestbps	resting blood pressure (in mm Hg on admission to hospital)

5	chol	serum cholesterol in mg / dl
6	fbs	fasting blood sugar> 120 mg / dl) (1 = true; 0 = false
7	restecg	Electrocardiographic results at rest
8	thalach	maximum heart rate reached
9	exang	exercise-induced angina (1 = yes; 0 = no)
10	oldpeak	Exercise-induced ST depression relative to rest
11	Slope	the slope of the peak exercise ST segment
12	that	number of major vessels (0-3)
13	thal	1 = normal; 2 = fixed fault; 3 = reversible fault
14	target	0 = no disease and 1 = disease

Table III.1. Cardiac Patient Database attribute information.

III.2.2. Breast cancer

Globally, breast cancer is the most common type of cancer in women and the second in terms of death rate.Breast cancer diagnosis is made when an abnormal lump is detected (by self-examination or x-ray) or a small speck of calcium is seen (on an x-ray), After a suspicious lump is found, the doctor will make a diagnosis to determine if it is cancerous and, if so, if it has spread to other parts of the body. This breast cancer data set was obtained from the Hospitals of the University of Wisconsin, Madison by Dr. William H. Wolberg.



```
           id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean
0      842302         M        17.99         10.38          122.80     1001.0          0.11840
1      842517         M        20.57         17.77          132.90     1326.0          0.08474
2    84300903         M        19.69         21.25          130.00     1203.0          0.10960
3    84348301         M        11.42         20.38           77.58      386.1          0.14250
4    84358402         M        20.29         14.34          135.10     1297.0          0.10030
...       ...       ...          ...           ...             ...        ...              ...
564    926424         M        21.56         22.39          142.00     1479.0          0.11100
565    926682         M        20.13         28.25          131.20     1261.0          0.09780
566    926954         M        16.60         28.08          108.30      858.1          0.08455
567    927241         M        20.60         29.33          140.10     1265.0          0.11780
568     92751         B         7.76         24.54           47.92      181.0          0.05263
569 rows × 7 columns
```

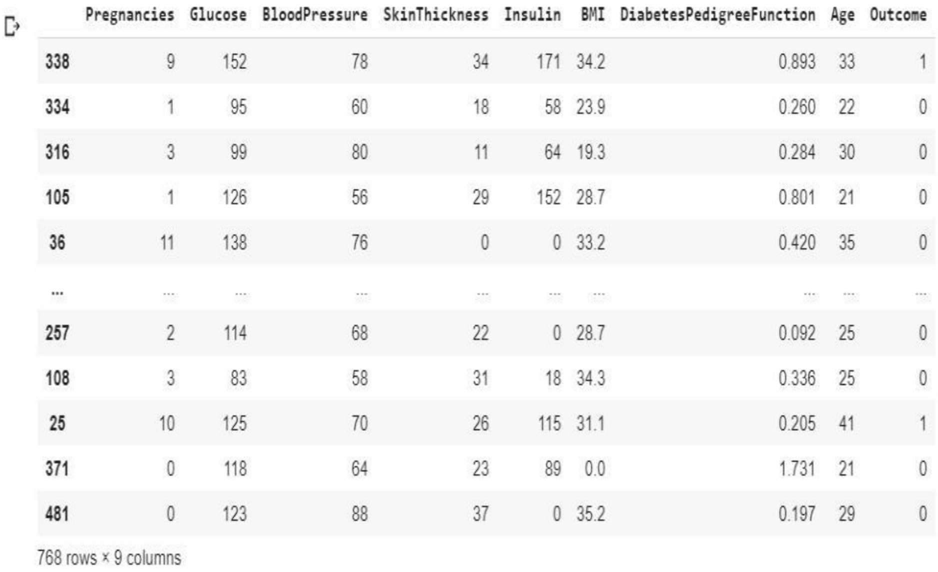
Fig.III.2. the form of breast cancer patient data.

No. setting	**Meaning**	**meaning**
0	id	ID number
1	mean_radius	mean of the distances from the center to the points of the perimeter
2	mean_texture	standard deviation of gray scale values
3	mean_perimeter	mean central tumor size
4	mean_area	
5	mean_smoothness	standard error for local variation of spoke lengths
6	diagnosis	Diagnosis of breast tissue (M = malignant, B = benign)

Table III.2. Breast cancer patient database attribute information.

III.2. 3.diabetes

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases, The purpose of the dataset is to predict by diagnosis whether or not a patient has diabetes, based on some diagnostic measures included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all of the patients here are women of at least 21 years of Pima Indian origin, this data consists of several medical predictor variables and one target variable, the outcome. Predictor variables include the number of pregnancies the patient has had, BMI, insulin level, age, etc.



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
338	9	152	78	34	171	34.2	0.893	33	1
334	1	95	60	18	58	23.9	0.260	22	0
316	3	99	80	11	64	19.3	0.284	30	0
105	1	126	56	29	152	28.7	0.801	21	0
36	11	138	76	0	0	33.2	0.420	35	0
...	...	...	...	...	...	...	...	...	...
257	2	114	68	22	0	28.7	0.092	25	0
108	3	83	58	31	18	34.3	0.336	25	0
25	10	125	70	26	115	31.1	0.205	41	1
371	0	118	64	23	89	0.0	1.731	21	0
481	0	123	88	37	0	35.2	0.197	29	0

768 rows × 9 columns

Fig.III.3. the form of diabetes patient data

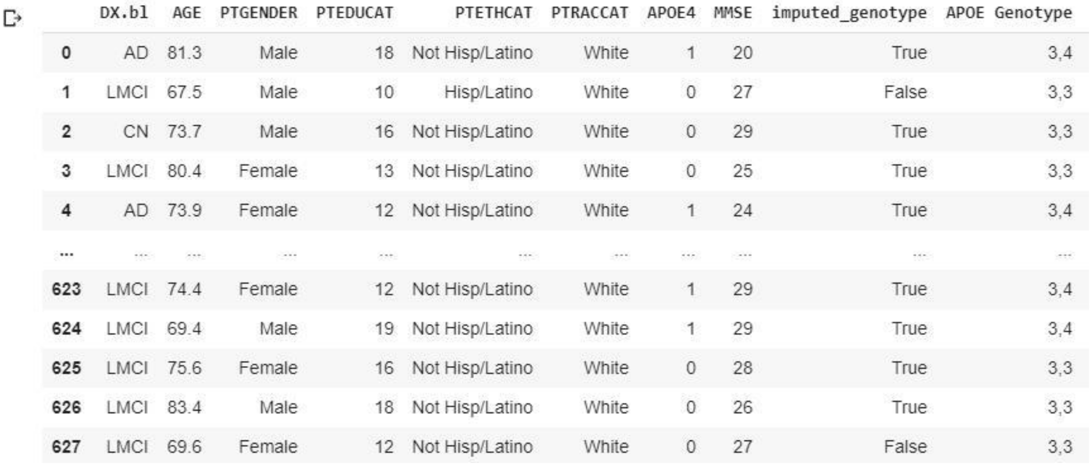
Parameter no.	Abbreviation	meaning
1	Pregnancies	Number of times pregnant

2	Glucose	Plasma glucose concentration at 2 hours in a test oral glucose tolerance
4	BloodPressure	Diastolic blood pressure (mm Hg)
5	SkinThickness	Triceps skin fold thickness (mm)
6	Insulin	Serum insulin 2 hours (mu U / ml)
7	BMI	Body mass index (weight in kg / (height in m) ^ 2)
8	DiabetesPedigreeFunction	Diabetes pedigree
9	Age	age in years
10	Outcome	Class variable (0 or 1) 268 out of 768 are 1, the others are 0

Table III.3. Diabetes Patient Database Attribute Information.

III.2. 4. Alzheimer's

Alzheimer's is the most common cause of dementia because it impairs mental and social skills, which leads to a more precise disruption of daily functioning in normal life, it is also a degeneration of healthy brain cells which leads to a continuous decline in memory as well as intellectual capacities, we will use the data provided by the ADNI project to develop a computer model that helps in the diagnosis of the disease using a deep neural network model.



```
     DX.bl   AGE  PTGENDER  PTEDUCAT         PTETHCAT  PTRACCAT  APOE4  MMSE  imputed_genotype  APOE Genotype
0       AD  81.3      Male        18  Not Hisp/Latino     White      1    20              True            3,4
1     LMCI  67.5      Male        10      Hisp/Latino     White      0    27             False            3,3
2       CN  73.7      Male        16  Not Hisp/Latino     White      0    29              True            3,3
3     LMCI  80.4    Female        13  Not Hisp/Latino     White      0    25              True            3,3
4       AD  73.9    Female        12  Not Hisp/Latino     White      1    24              True            3,4
...    ...   ...       ...       ...              ...       ...    ...   ...               ...            ...
623   LMCI  74.4    Female        12  Not Hisp/Latino     White      1    29              True            3,4
624   LMCI  69.4      Male        19  Not Hisp/Latino     White      1    29              True            3,4
625   LMCI  75.6    Female        16  Not Hisp/Latino     White      0    28              True            3,3
626   LMCI  83.4      Male        18  Not Hisp/Latino     White      0    26              True            3,3
627   LMCI  69.6    Female        12  Not Hisp/Latino     White      0    27             False            3,3
```

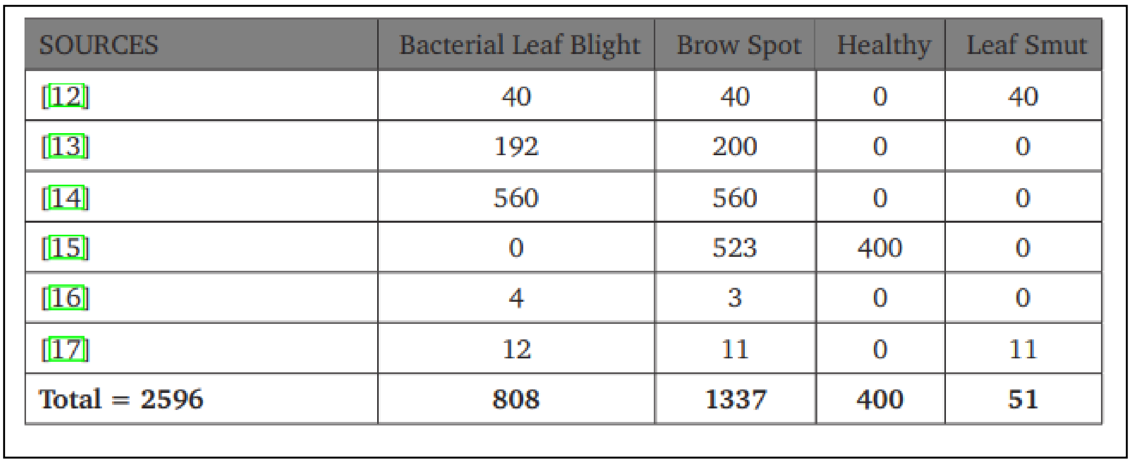
Fig.III.4. the form of data from Alzheimer's patients

No. setting	Abbreviation	meaning
1	Age	Age at departure
2	PTGENDER	Sex
4	PTEDUCAT	Years of study
5	PTETHCAT	Ethnicity
6	PTRACCAT	Race
7	APOE4	APOE4 genotype
8	MMSE	MMSE score on cognitive test
9	imputed_genotype	Designation specific to the challenge, TRUE = has imputed genotypes
10	DX.bl	Diagnosis at inclusion

Table III.4. Alzheimer's patient database attribute information.

III.2.5. Rice disease

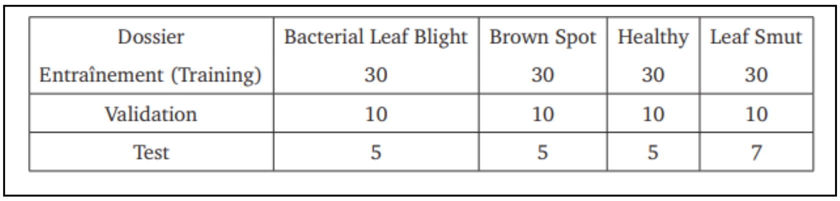
For the constitution of our dataset, several data sources were visited.
We have among others: They are presented in table 3.1.



SOURCES	Bacterial Leaf Blight	Brow Spot	Healthy	Leaf Smut
[12]	40	40	0	40
[13]	192	200	0	0
[14]	560	560	0	0
[15]	0	523	400	0
[16]	4	3	0	0
[17]	12	11	0	11
Total = 2596	**808**	**1337**	**400**	**51**

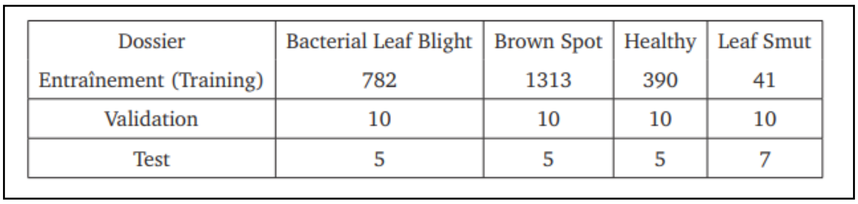
TABLE 3.1 - The different sources that allowed the creation of the Dataset

We have divided the data between 20% and 80% between the validation and training stages. We have also added another test folder to perform prediction
from the generated model. We worked on two types of dataset:
1.a first dataset (Table 3.2) containing a balanced sample of images for
every disease. This was used in the construction of the simple convolutional neural network model;
2.a second set of data (Table 3.3) containing an unbalanced sample
number of images for each disease. This was used for the learning transfer



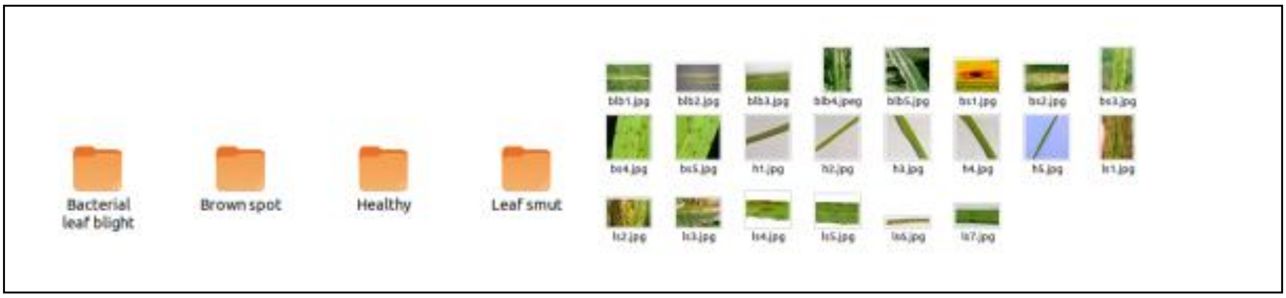
Dossier	Bacterial Leaf Blight	Brown Spot	Healthy	Leaf Smut
Entraînement (Training)	30	30	30	30
Validation	10	10	10	10
Test	5	5	5	7

TABLE 3.2 -
Detail of the small dataset [12]



Dossier	Bacterial Leaf Blight	Brown Spot	Healthy	Leaf Smut
Entraînement (Training)	782	1313	390	41
Validation	10	10	10	10
Test	5	5	5	7

TABLE 3.3 -
Detail of the large dataset: (Table 3.1)



(a) Training and Validation files (b) Test file FIGURE 3.1 - Content of the Training, Validation and Test files

III.3. EXPERIMENTAL PROTOCOL

After having described the characteristics of each database. Now the next operation is to implement a model and get results as a probability or a class. The next step is to determine the effectiveness of the metric-based model using test data sets. Different performance metrics are used to evaluate different machine learning algorithms, metric for evaluating machine learning algorithms is precision (Accuracy), errors, let's step in to see what these metrics are.

III.3.1. Classification rate

The classification rate is the number of correct predictions made by the model on all kinds of predictions made.

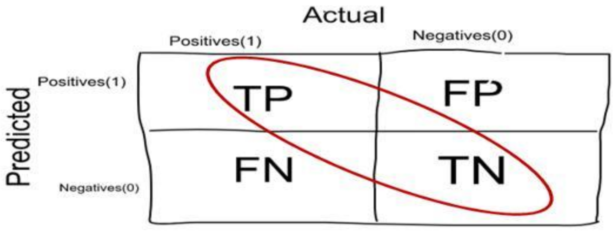


Fig.III.5. Classification rate shown in the confusion matrix.

(III.1)

In the numerator, our correct predictions (true positives and true negatives) (marked in red in the figure above) and in the denominator, are the kind of all predictions made by the algorithm (correct or not).

III.3.2. Error rate

The error in classification problems is the number of incorrect predictions made by the model over all types of predictions made.

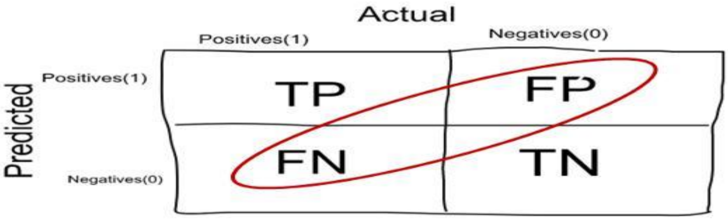


Fig. III.6. Error rate shown in the confusion matrix.

(III.2)

In the numerator, our incorrect predictions (false positives and false negatives) (marked in red in the figure above) and in the denominator, are the kind of all predictions made by the algorithm (correct or not).

III.4. SIMULATION RESULTS

III.4.1. With the Heart Disease database

In this part we show the results obtained with the database of cardiac patients with many deep learning models and also the results of the last years with many machine learning classifiers (svm, knn, ... etc), Table III.5. summarizes the performance of all these classifiers.

Model	epochs	Accuracy Train%	Loss Train%	Time train (us)	Accuracy Test%	Loss Test%	Time test (us)
DNN	200	90.98	26	31	**98.53**	**10**	39
DFFnn	200	81	47	4ms	82.9	36	1ms
LSTM	100	99.8	2	1ms	96.5	22	196
SVM					85		
Decision tree					71		
Logistic Regression					81		
Random Forest					80		
KNN					87		

Table III.5. Results obtained with the database of cardiac patients.

After obtaining the results (accurcy and loss) with all the models, we compared these results and as we note in the table above Table III.5. , the results we got with this data using the DNN model (accuracy_testing = 98.53 and loss_testing = 10) is better than all the results we got with the other models and classifiers, and the results were excellent as the shown in Fig. III.7. And Fig. III.8. .

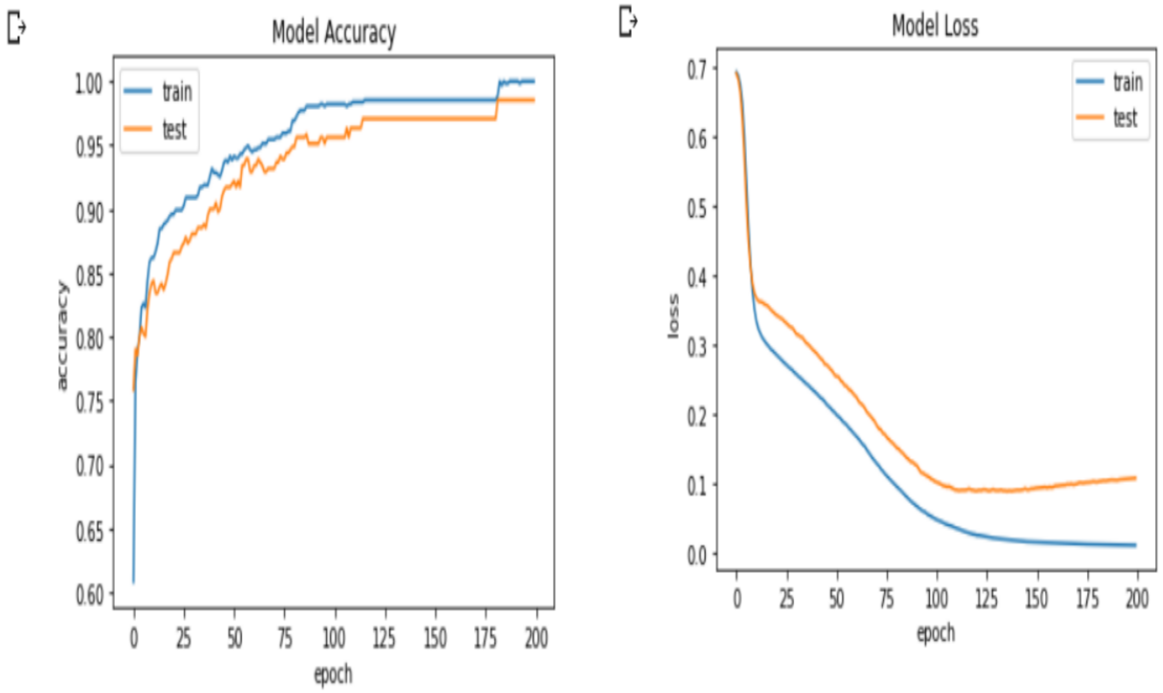
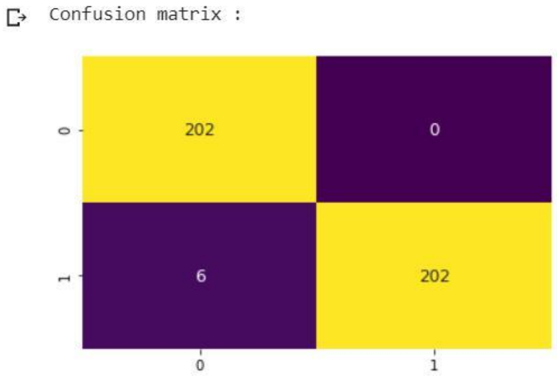


Fig. III.7. the difference between accuracy and loss of learning and testing with the data set of cardiac patients with the DNN model.

We also have the confusion matrix which also shows how much our model has been trained:



```
Confusion matrix :
```

Fig. III.8. the confusion matrix of cardiac patients with the DNN model.

III.4.2. With the Breast Cancer base

In this part we show the results obtained with the database of breast cancer patients with many deep learning models and also the results of the last years with many machine learning classifiers (svm, knn,. ..etc), Table III.6. summarizes the performance of all these classifiers.

Model	**epochs**	**Accuracy Train%**	**Loss Train%**	**Time train (us)**	**Accuracy Test%**	**Loss Test%**	**Time test (us)**
DNN	200	99.6	0.7	33	**100**	**0.28**	91
DFFnn	200	95.5	12	3ms	92.9	14	2ms
LSTM	50	97	13	4ms	91.6	31	387
SVM					63		
Decision tree					99		
Logistic Regression					64		
Random Forest					99		
KNN					72		

Table III.6. Results obtained with the database of breast cancer patients.

After obtaining the results (accurcy and loss) with all the models, we compared these results and as we note in the table above Table III.6. , the results we got with this data using the DNN model (accuracy_testing = 100 and loss_testing = 0.28) is better than any results we got with the other models and classifiers, and the results were excellent as shown in Fig. III.9. And Fig. III.10. .

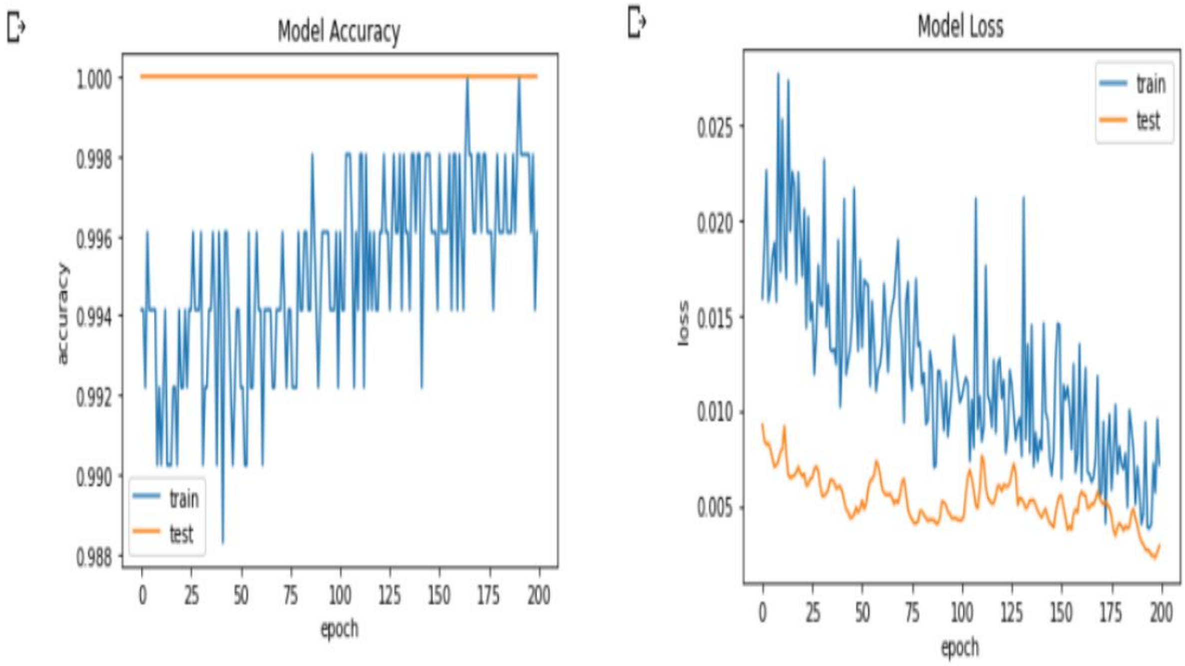
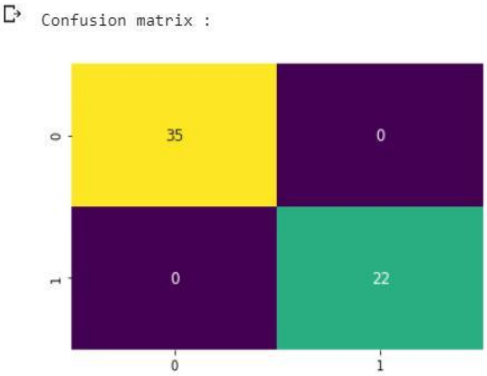


Fig. III.9. the difference between learning and testing accuracy and loss with the breast cancer patient dataset with the DNN model.

We also have the confusion matrix which also shows how much our model has been trained:



```
Confusion matrix :
```

Fig. III.10. the confusion matrix of breast cancer patients with the DNN model.

III.4.3. With basic diabetes

In this part we show the results obtained with the diabetes patient database with many deep learning models and also the results of the last years with many machine learning classifiers (svm, knn, ... etc), Table III.7. summarizes the performance of all these classifiers.

Model	epochs	Accuracy Train%	Loss Train%	Time train (us)	Accuracy Test%	Loss Test%	Time test (us)
DNN	500	87	27	180	**92**	20	137
DFFnn	200	80.9	42	4ms	74	61	2ms
LSTM	50	85	31	307	67	59	796
SVM					77.6		
Decision tree					**100**		
Logistic Regression					**100**		
Random Forest					**100**		
KNN					**100**		

Table III.7. Results obtained with the database of diabetes patients.

After obtaining the results (accuracy and loss) with all the models, we compared these results and as we note in the table above Table III.7. , The results we got with this data with the DNN model (accuracy_ testing = 92 and loss_testing = 20) It was better than the results we got from the LSTM and DFF models, while it was so close to the results with machine learning classifiers (M_L) the results are good as shown in Fig. III.11. And Fig. III.12. .

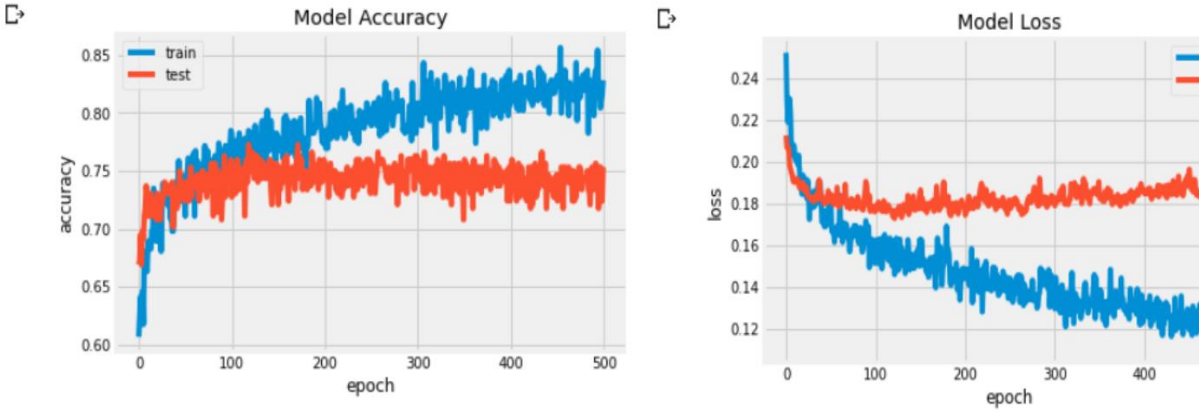
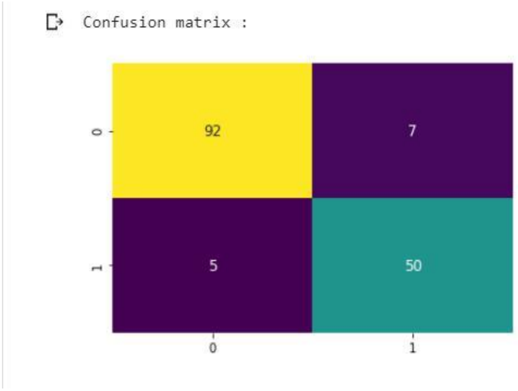


Fig. III.11. the difference between learning and testing accuracy and loss with the diabetes patient dataset with the DNN model.

We also have the confusion matrix which also shows how much our model has been trained:



```
Confusion matrix :
```

Fig. III.12. the confusion matrix of diabetes patients with the DNN model.

III.4.4. With the Alzheimer base

In this part we show the results obtained with the database of Alzheimer's patients with many deep learning models and also the results of the last years with many machine learning classifiers (svm, knn, .. .etc), Table III.8. summarizes the performance of all these classifiers.

Model	**epochs**	**Accuracy**	**Loss**	**Time**	**Accuracy**	**Loss**	**Time**
		Train%	**Train%**	**train (us)**	**Test%**	**Test%**	**test (us)**
DNN	200	78.6	48	80	**74.6**	**56**	81
DFFnn	500	68	63	9ms	73	66	2ms
LSTM	100	81	48	563	70	64	191
SVM					55		
Decision tree					63		
Logistic Regression					**75**		
Random Forest					66		
KNN					73		

Chart **III.8.** Results obtained with the database of Alzheimer's patients.

After obtaining the results (accuracy and loss) with all the models, we compared these results and as we note in the table above Table III.7. , The results we got with this data with the DNN model (accuracy_ testing = 92 and loss_testing = 20) It was better than the results we got from the LSTM and DFF models, while it was so close to the results with machine learning classifiers (M_L) the results are good as shown in Fig. III.13.

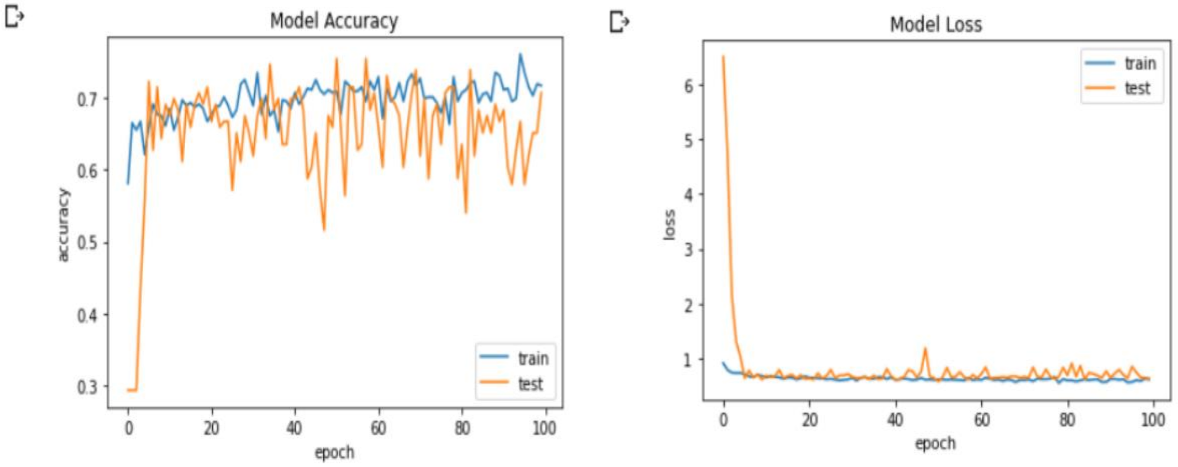
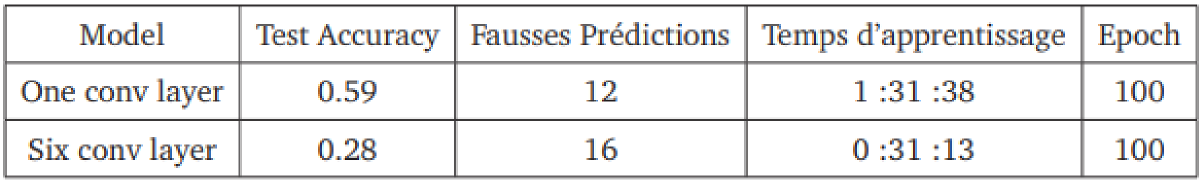


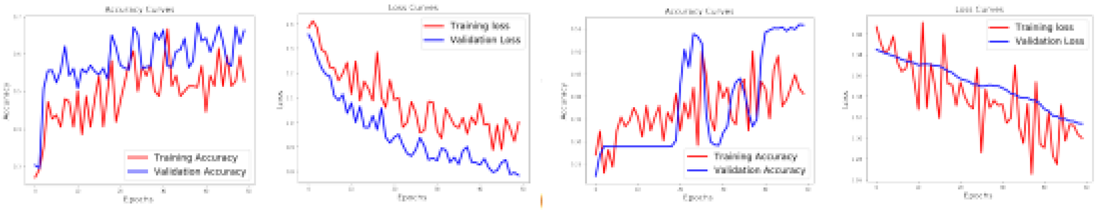
Fig. III.13. the difference between accuracy and loss of learning and testing with the dataset of Alzheimer's patients with the DNN model.

3.2 Results of training carried out on small data sets of 40 images per disease taken from [12] 1. Simple convolutional neural networks without a regularization method



Model	Test Accuracy	Fausses Prédictions	Temps d'apprentissage	Epoch
One conv layer	0.59	12	1 :31 :38	100
Six conv layer	0.28	16	0 :31 :13	100

TABLE 3.4 - CNN result without regularization method on small data sets

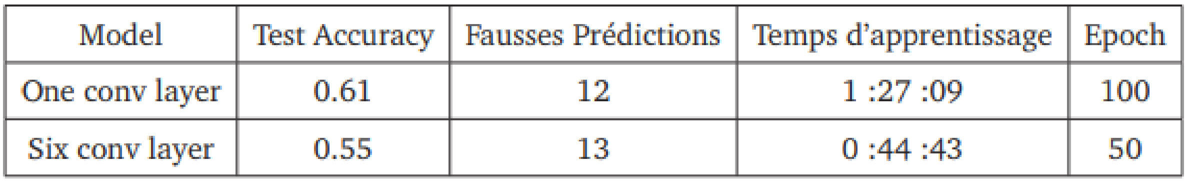


a) Training phase: b) Validation phase: c) Training phase: d) Six-layer validation phase
one layer one layer six layers

FIGURE 3.2 - Results of training on one and six convolutional layers without regularization method

Figures 3.2 (a), (b), (c) and (d) show that our model has an Underfitting problem. That is to say, the precision on the validation data sets is greater than that on the training data sets. The collection of additional data, or the application of one of the regularization methods can be considered to solve the problems related to this underfitting (see Annex 2)

2. Convolutional neuron network with 'Dropout' regularization method

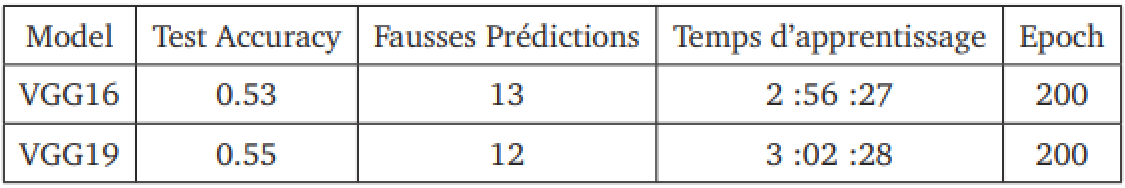


Model	Test Accuracy	Fausses Prédictions	Temps d'apprentissage	Epoch
One conv layer	0.61	12	1 :27 :09	100
Six conv layer	0.55	13	0 :44 :43	50

TABLE 3.5 - CNN result with Dropout regularization method on [12]

Table 3.5 shows us that despite the application of the regularization method, the problem of underfitting still persists. We therefore carried out further research on the internet. This allowed us to understand that faced with a really small number of data sets, we have to think about doing Transfer learning based on the different strategies to make our choice according to our dataset.

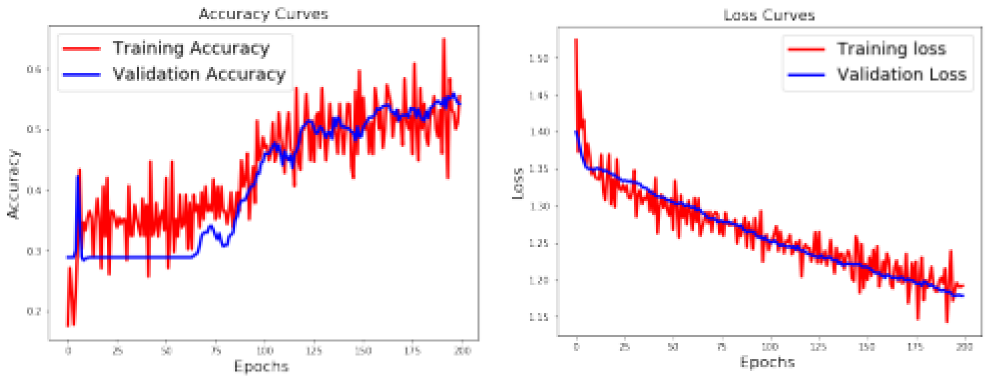
4. Transfer learning with knowledge extraction



Model	Test Accuracy	Fausses Prédictions	Temps d'apprentissage	Epoch
VGG16	0.53	13	2 :56 :27	200
VGG19	0.55	12	3 :02 :28	200

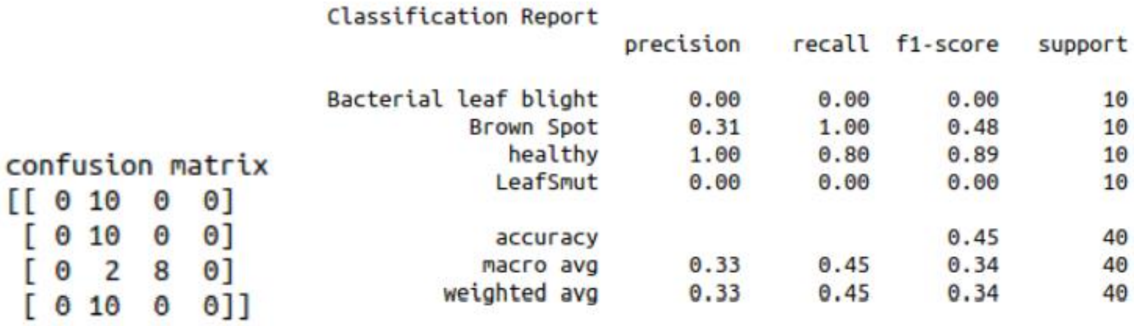
TABLE 3.7 - Knowledge extraction result on small data sets

Tables 3.6 and 3.7 present the results of transfer learning with fine-tuning and knowledge extraction. Faced with these two strategies, the problem of underfitting still persists but with a small improvement (see figure 3.3). However, by comparing the two tables, we see that the fine-tuning strategy promises us good results if we add other images to our datasets.



(a) Training phase b) Validation phase

FIGURE 3.3 - Results of the training and validation phases with knowledge extraction VGG- -16 on [12]



```
confusion matrix
[[ 0 10  0  0]
 [ 0 10  0  0]
 [ 0  2  8  0]
 [ 0 10  0  0]]
```

```
Classification Report
                      precision    recall  f1-score   support

Bacterial leaf blight      0.00      0.00      0.00        10
           Brown Spot      0.31      1.00      0.48        10
              healthy      1.00      0.80      0.89        10
             LeafSmut      0.00      0.00      0.00        10

             accuracy                          0.45        40
            macro avg      0.33      0.45      0.34        40
         weighted avg      0.33      0.45      0.34        40
```

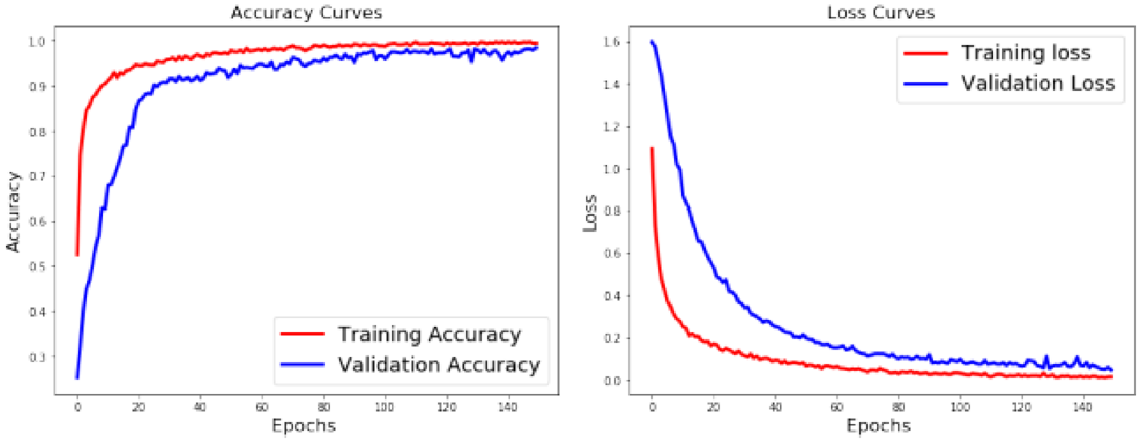
a) Confusion matrix b) Classification report

FIGURE 3.4 - Confusion matrix and classification report of performance measures: knowledge extraction VGG-16 on [12

Figure 3.4a shows that the model made too many false predictions. Figure 3.4b shows us the performance measures, we also see that the model reached 0.45% on the training datasets. This model is not at all reliable, nor precise.

3.3 Results of training sessions on the large data set 3.1 1. Transfer teaching with fine-tuning on the large data set

1. **Transfer learning with fine-tuning on the large data set**



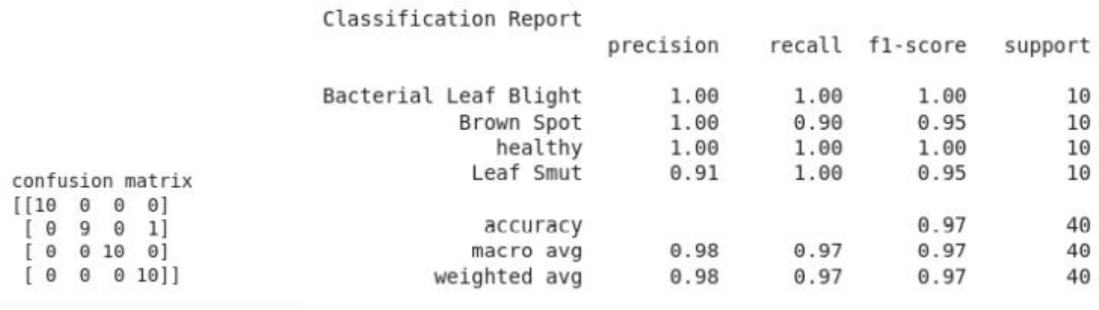
a) Training and validation accuracy b) Training and validation loss

FIGURE 3.5 - Transfer learning Strategy 1 with the Inception-v3 model

Model	Test Accuracy	Fausses Prédictions	Temps d'apprentissage	Epoch
VGG16	0,95	5	5 :36 :44	150
	0.975	4	7 :45 :75	200
	0.925	4	12 :35 :54	300
VGG19	0.975	4	5 :30 :49	200
	1.0	4	7 :30 :41	200
Mobilenet	0.95	4	5 :19 :26	150
	0.975	4	9 :48 :43	250
InceptionResNetV2	1.0	7	7 :09 :08	150
InceptionV3	1.0	3	8 :42 :08	150
	0.95	1	9 :03 :58	200
	0.975	0	11 :52 :42	250
Resnet152V2	0.925	4	5 :50 :17	150
Xception	0.975	6	9 :53 :38	200
MobilenetV2	0.9	5	5 :29 :45	150
	0.95	6	9 :19 :11	250
Densenet121	0.975	5	5 :32 :35	150
Densenet201	0.95	6	5 :59 :17	150
Resnet50	0.92	7	6 :15 :27	200

TABLE 3.8 - Strategy1 results: FINE-TUNING out of 3.1

Figures 3.5 (a) and (b) show that the precision of training and validation increases with the number of epochs. This reflects that with each epoch the model learns more information. Likewise, the learning and validation error decreases with the number of epochs.



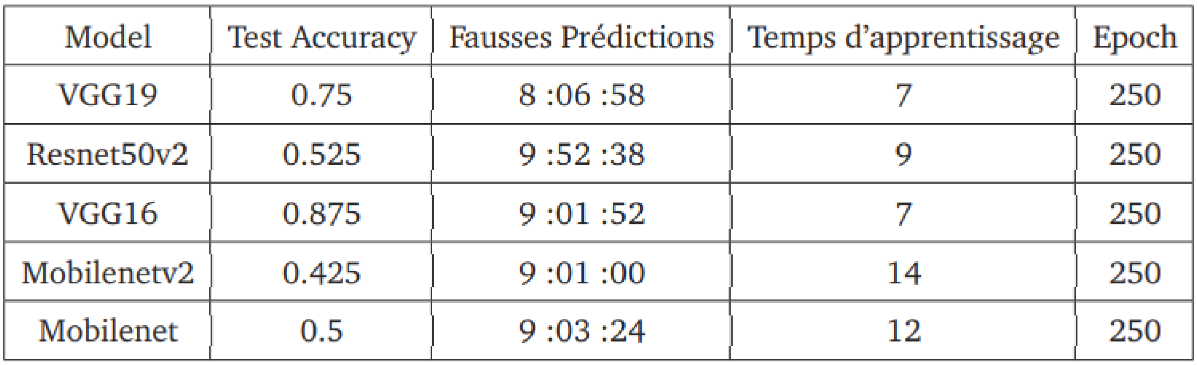
```
confusion matrix
[[10  0  0  0]
 [ 0  9  0  1]
 [ 0  0 10  0]
 [ 0  0  0 10]]
```

```
Classification Report
                       precision    recall  f1-score   support

Bacterial Leaf Blight       1.00      1.00      1.00        10
           Brown Spot       1.00      0.90      0.95        10
              healthy       1.00      1.00      1.00        10
            Leaf Smut       0.91      1.00      0.95        10

             accuracy                           0.97        40
            macro avg       0.98      0.97      0.97        40
         weighted avg       0.98      0.97      0.97        40
```

a) Confusion matrix b) Classification report

FIGURE 3.6 - Confusion matrix and classification report of performance measures

Figure 3.6 (a) shows that there is a false prediction (i.e. the model predicted an image to be Leaf Smut when it is a Brown Spot.



Model	Test Accuracy	Fausses Prédictions	Temps d'apprentissage	Epoch
VGG19	0.75	8 :06 :58	7	250
Resnet50v2	0.525	9 :52 :38	9	250
VGG16	0.875	9 :01 :52	7	250
Mobilenetv2	0.425	9 :01 :00	14	250
Mobilenet	0.5	9 :03 :24	12	250

TABLE 3.9 - Strategy2 results: Knowledge extraction on the large dataset

It should be noted that the training times for models with the fine-tuning strategy are much longer compared to those for the knowledge extraction strategy. Getting 100% accuracy on test data would not mean that our

model is 100% reliable. To select the model that we are going to use in our mobile application, we mainly relied on the number of false
predictions made on the 22 images and on the precision obtained on the validation datasets. Tables 3.8 and 3.9, it emerges that only the InceptionV3 model obtained zero (0) false prediction. He could
correctly classify the 22 test images into their respective classes without making false predictions and an accuracy of 0.975% on the validation datasets.

III.5. PARTIAL CONCLUSION

This last chapter was the subject of a simulation study related to the implementation of deep learning techniques applied to the science of disease identification, this study having made it possible to take a look at most of the deep learning techniques (DNN, LSTM, DFF, ... etc) and by studying them, learn more about them and separately assess the performance of each of them.

This detailed study was carried out with the aim of decisively choosing the proposed approach, the parameters associated with the recognition rate (precision and loss) being the relevant factors which made it possible to evaluate the studied methods and to make the comparison possible. between their performance, where after this comparison we found that in general this deep neural network model performs better than other models and classifiers, especially in terms of disease identification and when we use this data type specially.

GENERAL CONCLUSION

At the stage of the selection of technologies in our system, the criterion that we took into consideration is the form of the data to be studied, because the data has several forms (1D, 2D, 3D), and the deep learning techniques differ. depending on how they use this data, for example, one of them uses only one shape of these shapes or gives us the best performance with it, but another can use with all of these shapes of data.

We then tested deep learning techniques using four real biomedical databases with a dimension of different diseases to verify the validity of the proposed system, as it allows us to know the advantages and disadvantages of each technique. and the differences between one technique and another.

If this system is applied in hospitals, the number of treatments will be reduced which will take place will contribute to reducing the intervention of the doctor, especially in certain cases which do not require his intervention at all, which will reduce the medical equipment and the costs. persons concerned.

In this case, DNN has proven its effectiveness through its high performance compared to other deep learning techniques and machine learning techniques as illustrated by the set of tables in the previous chapter. Artificial Intelligence is a tool developed to imitate human intelligence, and more precisely to replace certain actions performed by humans so that the activity is more efficient and faster. In the field of health and in Clinical Research, Artificial Intelligence would represent a great help to doctors, because it would allow them to process clinical data as well as to offer adapted diagnoses and treatments more quickly. It would save time and advance Research exponentially. The feared risks it entails would be that one day the machine would replace humans and fall into malicious hands through hacking. This has given rise to a great many ethical questions which to this day remain unanswered. In the field of health and in medical research, it would be impossible for the machine to take the place of man. Medicine is an inexact science and it is necessary to investigate the advantages and not to rely on a simple diagnosis made. The machine suggests diagnostic proposals thanks to its knowledge library and machine learning algorithm. But the doctor will always be present to verify and validate the diagnosis made. Like any other technology, Artificial Intelligence must be equipped with a system of protection against the arrival of intrusions, or external attacks thus causing a modification of the coding included in its algorithm. It is therefore necessary to constantly renew accessibility, for example, by regularly changing passwords making them more complex, and to have a secure containment zone dedicated to the centralization of data. Artificial Intelligence at the service of medical research is still in progress due to certain factors hindering its deployment. Although it must be a powerful tool for the development of medical research, its implementation takes time because of a regulatory and ethical vacuum. Many structures, such as F-CRIN and INSERM, are starting to set up projects combining Clinical Research and Artificial Intelligence,

Interesting prospects are envisaged. We could, for example,:

1. Collect other images of different types of diseases in order to make our model much more reliable and precise;
2. Broaden the field of identification of the system to other pathologies affecting not only human beings but also animals and plants
3. to propose an image segmentation system with a view to allowing the detection of several pathologies on a single image.

Annex1: Image characteristics

The image is a structured set of information characterized by the following parameters:

- **Pixel:**The pixel is the abbreviation of the word "Picture element", is a unit of area allowing to define the basis of a digital image. It materializes a given point (x, y) on the plane of the image. The information presented by the pixel is the gray level (or color) taken from the corresponding location in the real image. The difference between monochrome image and color image lies in the quantity of information contained in each pixel, for example in a color image (RGB: Red, Green, Blue) the value of a pixel is represented on three bytes for each color.
- **Dimension & Resolution:**The dimension is the size of the image. It is in the form of a matrix whose elements are digital values representative of light intensities (pixels). The number of rows in this matrix multiplied by the number of columns gives us the total number of pixels in an image.

  In contrast, resolution is the clarity or finesse of detail achieved by a monitor or printer in producing images. On computer monitors, resolution is expressed as a number of pixels per unit of measurement (inch or centimeter). The word resolution is also used to refer to the total number of horizontal and vertical pixels on a monitor. The larger this number, the better the resolution.
- **Neighborhood:**The image plane is divided in terms of rectangular or hexagonal shapes thus allowing the exploitation of the notion of neighborhood. The neighborhood of a pixel is formed by the set of pixels which are located around this same pixel. We define

also the plate as being the set of pixels defining the neighborhood taken into account around a pixel. There are two types of neighborhood:

Neighborhood at 4: We only take into account the pixels which have a common side with the pixel considered.

Neighborhood at 8: We take into account all the pixels which have at least one point in connection with the pixel considered.

- **Shades of grey :**It is the light intensity value of a pixel. This value can range from black (0) to white (255) passing through the shades that are contained in the interval [0, 255]. It actually corresponds to the amount of reflected light.

  For 8 bits, there are 256 gray levels, 40 of which are recognized by the naked eye. The greater the number of bits, the more the levels are numerous and the more faithful the representation.

- **Contrast:**It is the marked opposition between two regions of an image. A contrasted image presents a good dynamic of the distribution of gray values over the entire range of possible values, with very clear whites and deep blacks. On the contrary, a low contrast image has a low dynamic range, most of the pixels having very similar gray values. If L1 and L2 are the degrees of luminosity respectively of two neighboring zones A1 and A2 of an image, the contrast is defined by the ratio: $C = L1 - L2 / L1 + L2$
- **Luminance:**It is the degree of brightness of the points of the image. It is also defined as being the quotient of the luminous intensity of a surface by the apparent area of this surface, for a distant observer, the word luminance is substituted for the word brilliance, which corresponds to the brightness of an object . Good luminance is characterized by:

  - Bright (brilliant) images;
  - Good contrast: images should be avoided where the range of contrast tends towards white or black;
  - The absence of parasites.

- **Noise :**A noise (parasite) in an image is considered to be a phenomenon of sudden variation in the intensity of a pixel compared to its neighbors, it comes from the lighting of the optical and electronic devices of the sensor. It is a parasite which represents certain faults (dust, small clouds, momentary drop in electrical intensity on the sensors, etc.). It results in small-dimension spots whose distribution on the image is random.

- **Outline:**Contours represent the border between objects in the image, or the limit between two pixels whose gray levels represent a significant difference. In a digital image, the contours lie between the pixels belonging to regions having different average intensities; these are contours of the “amplitude jump” type. An outline can also correspond to a local variation in intensity exhibiting a maximum or a minimum; it is then a question of “roof” contour"[7].

Annex 2: Regularization methods

To avoid falling into the problem of over-learning (overfitting) and under-learning (underfitting), there are regularization methods to use.

- Empirical method

1. **Dropout:**the "FC" (Fully Connected) layers take up most of CNN's memory. Moreover, the concept of FC creates an exponential memory problem called "overfitting" ("over-connection" leading to over-learning) slowing down the processing of information. To prevent this, the dro-pout method is used to randomly "turn off" neurons (with a predefined probability, often every other neuron) as well as peripheral neurons. Thus, with fewer neurons, the network is more responsive and can therefore learn faster. At the end of the learning session, the "turned off" neurons are "turned back on" (with their original weights). The closer the FC layer is to the source image, the fewer neurons will be extinguished.

   The goal is to switch neurons off and on again at random, as part of successive trainings. Once the training series is finished, we turn all the neurons back on and use the network as usual. This technique has shown not only a gain in the speed of learning, but by disconnecting the neurons, we have also limited marginal effects, making the network more robust and able to better generalize the concepts learned.

2. **DropConnect:**it is an evolution of dropout, where we are no longer going to switch off a neuron, but a connection (the equivalent of the synapse), and always in a random manner. The results are similar (speed, ability to generalize learning), but show a difference in terms of the evolution of the weights of connections. An FC layer with a DropConnect can be compared to a "diffuse" connection layer.
3. **Stochastic pooling:**It uses the same principle as Max-pooling, but the chosen output will be taken at random, according to a multinomial distribution defined according to the activity of the zone addressed by the pool. In fact, this system is similar to doing Max-pooling with a large number of similar images, which vary only by localized deformations. We can also consider this method as an adaptation to elastic deformations of the image. This is why this method is very effective on MNIST images (database of images representing handwritten digits). The strength of stochastic pooling is to see its performance grow exponentially with the number of layers in the network.

- **Explicit method**

1. **Network size:**the easiest way to limit over-learning is to limit the number of layers in the network and free up free network parameters (connections). This directly reduces the power and predictive potential of the network. It is equivalent to having a "zero standard".
2. **Weight degradation:**The concept is to consider the vector of the weights of a neuron (list of weights associated with the incoming signals), and to add to it an error vector proportional to the sum of the weights (norm 1) or to the square of the weights (2 or Euclidean norm). This error vector can then be multiplied by a coefficient of proportionality which will be increased to further penalize the vectors of high weight.

   -Regularization by standard 1: The specificity of this regulation is to reduce the weight of random and weak entries and to increase the weight of "important" entries. The system becomes less sensitive to noise.

   - Regularization by norm 2: (Euclidean norm) The specificity of this regulation is to reduce the weight of the strong inputs, and to force the neuron to take more account of the weak inputs.

The regularizations by norm 1 and norm 2 can be combined: this is the "elastic net
regulation".

Annex 3: Difference between 'accuracy', 'loss'' validation_accuracy, validation_loss

1. **Accuracy or Accuracy**is a method of measuring the performance of a classification model. It is generally expressed as a percentage. Precision is the number of predictions where the predicted value equals the true value. It is binary (true / false) for a particular sample. Accuracy is often graphed and checked during the training phase, although the value is often associated with the overall or final accuracy of the model. Precision is easier to interpret than loss.

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total prediction}}$$

2. Loss: is a loss function, also known as a cost function, takes into account the probabilities or uncertainty of a prediction based on the difference (error) between the prediction value and the actual value. This gives us a more nuanced view of the model's performance. The most common loss function used in deep neural networks is cross-entropy.It is defined as:

Xn Xm
Cross - entropy = - yi, j log (Pi, j)
i
= 1 i = 1

where, yi, j denotes the true value, ie 1 if the sample i belongs to class j and 0 otherwise. Pi, j denotes the probability predicted by your sample model i belonging to class j.

During the training phase of our model on the training datasets, the 'accuracy' and 'loss' values may vary in different cases. Usually with each epoch the loss should decrease and the accuracy should increase. However, with the variables, 'val_loss' and 'val_acc', many cases can be possible.

1. val_loss starts to increase, val_acc starts to decrease. This means that the model crams values without learning;
2. val_loss starts to increase, val_acc also increases. This can be the case of over-learning or various probability values in cases where softmax is used in the output layer;
3. val_loss starts to decrease, val_acc starts to increase. This is also good because it means that the model built learns and performs well.

Annex 4: Implementation of strategy 1 and 2: Fine-tuning and knowledge extraction

---

1. **Import of Keras libraries**

```
        # import relevant libraries
import bone
import sys
import datetime
import glob as glob
import numpy as np
import cv2
# [Keras Models]
# import the Keras implementations of VGG16, VGG19, InceptionV3 and Xception models

# the model used here is VGG16
from keras.applications.inception_v3 import InceptionV3
from keras.models import Model
from keras.layers import Dense, GlobalAveragePooling2D
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
from keras.optimizers import SGD
#from scipy.interpolate import spline
import pandas as pd
import matplotlib.pyplot as plt
```

2. Loading training and validation datasets

```
# [Dataset]
# image dimensions for VGG16, VGG19 are 224, 224
# image dimensions for InceptionV3 and Xception are 299, 299
img_width, img_height = 299, 299

train_dir = '/ diseases / training'
validate_dir = '/ diseases / validation'
nb_epochs = 250
batch_size = 32
nb_classes = 4
```

3. Preprocessing and extension of data sets

```
# data pre-processing for training
train_datagen =               ImageDataGenerator (
        rescale = 1./255,
        rotation_range = 20,
        width_shift_range = 0.2,
        height_shift_range = 0.2,
        shear_range = 0.2,
        zoom_range = 0.2,
        fill_mode = 'nearest',
        horizontal_flip = True)

# data pre-processing for validation
validate_datagen =               ImageDataGenerator (
        rescale = 1./255,
        rotation_range = 20,
        width_shift_range = 0.2,
        height_shift_range = 0.2,
        shear_range = 0.2,
        zoom_range = 0.2,
        fill_mode = 'nearest',
        horizontal_flip = True)
```

```
# generate and store training data
train_generator = train_datagen.flow_from_directory (
        train_dir,
        target_size = (img_width, img_height),
        batch_size = batch_size)

# generate and store              data validation
validate_generator =              validate_datagen.flow_from_directory (
        validate_dir,
        target_size = (img_width, img_height),
        batch_size = batch_size)
```

4. **Definition of the model**

```
# set up transfer learning on pre-trained ImageNet VGG19 model - remove fully connected layer and replace
# with softmax for classifying the number of classes in the dataset
inceptionV3_model = InceptionV3 (weights = 'imagenet', include_top = False)
x = inceptionV3_model.output
x = GlobalAveragePooling2D () (x)
x = Dense (1024, activation='reread') (x)
predictions = Dense (nb_classes, activation = 'softmax') (x)
model = Model (input = inceptionV3_model.input, output = predictions)
```

Strategy 1: Fine-tuning

```
# fine-tuning
for layer in inceptionV3_model.layers:
        layer.trainable = True
```

Strategy 2: Knowledge extraction

```
# fine-tuning
for layer in inceptionV3_model.layers:
        layer.trainable = True
```

5. **Model compilation**

```
# compile the new model using a RMSProp optimizer
model.compile(optimize = SGD (lr = 0.0001, momentum=0.9),
        loss = 'categorical_crossentropy',
        metrics = ['accuracy'])
```

6. Model training

```
# fit the model, log the results and the training time
now = datetime.datetime.now
t = now ()
transfer_learning_history = model.fit_generator (
        train_generator,
        nb_epoch = nb_epochs,
        samples_per_epoch = 2526,
        validation_data = validate_generator,
        nb_val_samples = 40,
        class_weight='auto')
print('Training time: %s' % (now () - t))
```

7. Model evaluation

```
# evaluate the performance the new model and report the results
score = model.evaluate_generator (validate_generator, 40/batch_size)
print("Test Score:", score[0])
print("Accuracy test:", score[1])

Test Score: 0.0390192016959l9034
Accuracy test: 0.975
```

Saving the model

```
# save transfer learning model for offline prediction purposes
model.save ('diseases_inceptionV3_model_ft.h5')
```

Graphical representation of the results

```
# Loss Curves
plt.figure (figsize= [8.6])
plt.plot(transfer_learning_history.history['loss'],'r', linewidth=3.0)
plt.plot(transfer_learning_history.history['val_loss'],'b', linewidth=3.0)
plt.legend(['Training loss', 'Validation Loss'], fontsize=18)
plt.xlabel('Epochs', fontsize=16)
plt.ylabel('Loss', fontsize=16)
plt.title ('Loss Curves', fontsize=16)

# Accuracy Curves
plt.figure (figsize= [8.6])
plt.plot(transfer_learning_history.history['acc'],'r', linewidth=3.0)
plt.plot(transfer_learning_history.history['val_acc'],'b', linewidth=3.0)
plt.legend(['Training Accuracy', 'Validation Accuracy'], fontsize=18)
plt.xlabel('Epochs', fontsize=16)
plt.ylabel('Accuracy', fontsize=16)
plt.title ('Accuracy Curves', fontsize=16)
```

8. **Test the model**

```
#predict images in test folder: 22 images
num_images = len(glob.glob ("/diseases/tester/*.*"))
predict_files = glob.glob ("/diseases/tester/*.*")

im = cv2.imread (predict_files[0])
im = cv2.cvtColor (im, cv2.COLOR_BGR2RGB)
im = cv2.resize (im, (256, 256)). astype (np.float32)
im = np.expand_dims (im, axis = 0)/255

predictor, image_id = [], []
for i in predict_files:
        im = cv2.imread (i)
        im = cv2.resize (cv2.cvtColor (im, cv2.COLOR_BGR2RGB), (256, 256)). astype (np.float32) / 255.0
        im = np.expand_dims (im, axis =0)
        outcome = [np.argmax (model.predict (im))]
        predictor.extend (list(outcome))
        image_id.extend ([i.rsplit ("\\")[-1]])

final = pd.DataFrame ()
final["id"] = image_id
final["Diseases"] = predictor

classes = train_generator.class_indices
classes = {value : key for key, value in classes.items ()}

final["Diseases"] = final["Diseases"].apply(lambda x: classes[x])
final.head (num_images)

#save the prediction results
final.to_csv ("
                                                  DiseasesTestFiles_with_pretrained_inceptionV3
                                                  .csv "index= False)
```

Appendix 5: Implementation of a simple convolutional neuron network

1. **Importing keras libraries**

```
# import relevant libraries

import bone
import sys
import datetime
import glob as glob
import numpy as np
import cv2
# [Keras Models]
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPooling2D, Dropout, Flatten
from keras.preprocessing.image import ImageDataGenerator, array_to_img,
                                                  img_to_array, load_img
from keras.optimizers import SGD
#from scipy.interpolate import spline
import pandas as pd
import matplotlib.pyplot as plt
```

Definition of Model1: a convolutional layer

```
#create model

def cnn_model ():
    model = Sequential ()
    model.add (Conv2D (32, (3, 3), padding='same',
                             input_shape=(224,224, 3),
                             activation='reread'))
    model.add (MaxPooling2D (pool_size=(2, 2)))
    model.add (Flatten ())
    model.add (Dense (1024, activation='reread'))
    model.add (Dropout (0.5))
    model.add (Dense (4, activation='softmax'))
    return model
```

Model definition2: Six layers of convolution

```
def convnet_model ():
    model = Sequential ()

    model.add (Conv2D (32, (3, 3), padding='same',
                             input_shape=(224,224,3),
                             activation='reread'))
    model.add (Conv2D (32, (3, 3), activation='reread'))
    model.add (MaxPooling2D (pool_size=(2, 2)))
    model.add (Dropout (0.2))

    model.add (Conv2D (64, (3, 3), padding='same',
                             activation='reread'))
    model.add (Conv2D (64, (3, 3), activation='reread'))
    model.add (MaxPooling2D (pool_size=(2, 2)))
    model.add (Dropout (0.2))

    model.add (Conv2D (128, (3, 3), padding='same',
                             activation='reread'))
    model.add (Conv2D (128, (3, 3), activation='reread'))
    model.add (MaxPooling2D (pool_size=(2, 2)))
    model.add (Dropout (0.2))

    model.add (Flatten ())
    model.add (Dense (512, activation='reread'))
    model.add (Dropout (0.5))
    model.add (Dense (4, activation='softmax'))
    return model
```

Annex 6: Conversion and integration of the model obtained in the android application

```
from keras.backend import clear_session
import numpy as np
import tensorflow as tf
clear_session ()
np.set_printoptions (suppress= True)
input_graph_name = "diseases_inceptionV3_model_250_ft.h5"
output_graph_name = input_graph_name[: -3] + '.tflite'
converter = tf.lite.TFLiteConverter.from_keras_model_file (model_file= input_graph_name)

converter.post_training_quantize = True
tflite_model = converter.convert ()
open(output_graph_name, "wb") .write (tflite_model)
print ("generate:", output_graph_name)

#obtain the labels.txt file
print (train_generator.class_indices)

labels = '\not'.join (sorted(train_generator.class_indices.keys ()))

with open('labels.txt', 'w') as f:
    f.write (labels)
```

Appendix 7 - Python code to create a DNN under TensorFlow

```
from __future__ import absolute_import from __future__ import
division
from __future__ import print_function

import os
import urllib
import numpy as np
import tensorflow as tf

# Data sets
Train = "Train.csv"
Test = "Test.csv"

def main ():
    # Load datasets
    training_set = tf.contrib.learn.datasets.base.load_csv_with_header (
            filename = Train,
            target_dtype = np.int,
            features_dtype = np.float32)
    test_set = tf.contrib.learn.datasets.base.load_csv_with_header (
            filename = Test,
            target_dtype = np.int,
            features_dtype = np.float32)

    # Specify that all features have real-value data
    feature_columns = [tf.feature_column.numeric_column ("x", shape = [28])]

    # Build 3 layer DNN with 10, 20, 10 units respectively
    classifier = tf.estimator.DNNClassifier (feature_columns = feature_columns,

hidden_units = [10,20,10],
n_classes = 2,

model_dir = "/ tmp / RadiomicsICC_model6")
```

```
#    Define the training inputs
     train_input_fn = tf.estimator.inputs.numpy_input_fn (x = {"x": np.array (training_set.data)}, y =
             np.array (training_set.target), num_epochs = None,
             shuffle = True)

     # Train model
     classifier.train (input_fn = train_input_fn, steps = 1000)

     # Define the test inputs
     test_input_fn = tf.estimator.inputs.numpy_input_fn (x = {"x": np.array (test_set.data)}, y = np.array

             (test_set.target),

             num_epochs = 3,
             shuffle = False)

     # Evaluate model
     accuracy_score = classifier.evaluate (input_fn = test_input_fn) ["accuracy"]

     print ("\ nAccuracy: {0: f} \ n" .format (accuracy_score))

if __name__ == "__main__":
       hand()
```

REFERENCES

[1] Methodological guide: How to evaluate a screening program a priori? , Technological evaluation service, Economic evaluation service May 2004 years.

[2] https://livre.fnac.com/a1854678/Marina-Carrere-d-Encausse-La-maladie-d-Alzheimer

[3] https://sante.journaldesfemmes.fr/fiches-maladies/2499922-diabete-definition-causes-symptoms-treatment /

[4] End of studies thesis, for obtaining the Master's degree in Computer Science,

Option: Intelligent Model and Decision (MID), Theme: Deep Learning for classification des images, Directed by: Moualek Djaloul Youcef Academic year: 2016-2017.

[5] Professional Master Thesis, Field: Computing and Information Technology, Sector: Computing, Specialty: fundamental, Presented by: Boughaba Mohammed and Boukhris Brahim, Theme: Deep Learning

for classification and search of images by content Academic year 2016/2017.

[6] End of study thesis, with a view to obtaining the Master's degree in

Computing Intelligent Systems Option, Theme: Development of an Architecture Based on Deep Learning for Network Intrusion Detection, Prepared by: Mr. MIMOUNE Zakarya. University year 2018/2019.

[7] https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/deep-learning-vs-machine-learning/

[8] End of study thesis, with a view to obtaining a Master's degree in computer science, Option: Intelligent Systems, Theme: The classification of satellite images by deep learning, Prepared by: Abdelaziz HABBA and Omar ISHAK

[9] https://actualiteinformatique.fr/intelligence-artificielle/definition-deep-learning

[10] https://developer.ibm.com/technologies/artificial-intelligence/articles/cc-machine-learning-deep-learning-architectures/

[11] https://dataanalyticspost.com/Lexique/reseau-de-neurones-profonds/

[12] https://penseeartificielle.fr/focus-reseau-neurones-convolutifs/

[13] https://www.aliens-sci.com/convolutional-neural-network-cnn/

[14] https://penseeartificielle.fr/focus-reseau-neurones-artificiels-perceptron-multicouche/

[15] https://www.sciencedirect.com/science/article/abs/pii/S0952197617300234

[16] https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7

[17] https://hackernoon.com/deep-learning-feedforward-neural-networks-explained-c34ae3f084f1

[18] https://hackernoon.com/building-a-feedforward-neural-network-from-scratch-in-python-d3526457156b

[19] https://medium.com/swlh/neural-networks-4b6f719f9d75

[20] https://archive.ics.uci.edu/ml/datasets/Heart+Disease, Accessed May 2019.

[21] https://www.kaggle.com/ronitf/heart-disease-uci, Accessed May 2019.

[22] http://biochimej.univangers.fr/Page2/COURS/Zsuite/6BiochMetabSUITE/5IntelligenceA
rtificial / 1Artificial Intelligence.htm

[23] https://alzheimer-recherche.org/la-maladie-alzheimer/quest-maladie-dalzheimer's /
definition-and-numbers /

[24] Data Set for Rice Diseases with labels https://github.com/aldrin233/RiceDis eases-DataSet

[25] rice diseases using cnn and svm https://www.kaggle.com/rajeshbhattacharj ee / rice-diseases-using-cnn-and-svm

[26] Rice Diseases Image Dataset https://www.kaggle.com/minhhuy2810/rice-di seases-image-dataset

[27] Rice Knowledge Bankhttp://www.knowledgebank.irri.org/step-by-step-
https://www.sipen-dakar.com/dematerialisation-a-lere-de-lintelligence-artificielle/

Robotics Process Automation Training - Senegal (theknowledgeacademy.com)

https://www.jeuneafrique.com/emploi-formation/843781/senegal-une-nouvelle-ecole-dediee-a-lintelligence-artificielle-a-dakar/

http://www.senegalautomation.com/?numpage=4&lien=31

http://www.senegalautomatio