Diagnosis with the assistance of Deep Learning: Application of Artificial Intelligence in the medical area

DT 228
BSc in Computer Science

Rui Ranito
C18753061

Dr Luis Miralles

School of Computer Science
Technological University, Dublin

08/04/2022

Abstract

There are some very strong popular beliefs that say that soon the world will be dominated by robots, that cars will be flying, and that artificial intelligence will dominate the human race: stories that can be found on any social network typed through an iPhone or Android. Technology, artificial intelligence, is not a popular belief, it's a reality. An object, a shape, a method that can be used to improve aspects of our daily lives.

When talking about the interference of artificial intelligence in the medical field, it is necessary to think: How many lives can be saved with the use of advanced technology? How many hours in the doctor x patient relationship could the time in improved diagnosis be better used with Deep Learning? How much about our own biology would be discoverable with the help of software and technological experiments?

This monograph proposes to discuss and analyse, through a proof of concept, the possibilities of Artificial Intelligence acting within the medical field, using Deep Learning to diagnose healthy lungs, with viral pneumonia and affected by COVID-19, by reading images. In addition to technical terms, it was sought to create a bridge between medicine and technology, proving that the way of conversation and research between them is not only innovative, but also extremely necessary for daily practices in search of health and quality of care. life. In this process, the history of diagnostic imaging and the history and structure of Machine Learning were researched, until understanding how Deep Learning works. Hypotheses and proof through the project were created for the applicability of these concepts in a practical way.

Keywords: Machine learning, Deep learning, Diagnostic imaging, Medicine. Artificial intelligence, Coronavirus.

Declaration

I hereby certify that the material which I now submit for assessment as Dissertation on the program of study leading to the award of BSc. (Hons) in Computer Science, is entirely my own work, except where fully and properly referenced. This Dissertation has not been submitted in whole or in part for assessment for any academic purpose other than in partial fulfilment for that stated above.

Signed: Rui Ranito Date: 08/04/2021

Acknowledgements

I would like to thank and dedicate this dissertation to the following people:

My family, my mother Isa, and my father José, two great supporters of my professional and academic career, for all the patience and love given to me throughout literally everything.

My wife Luana who accompanied me, listening to me, advising me, offering me help throughout the course, always brought me support and happiness in all situations.

I would like to thank Boss, my wonderful dog, who always cheers me up when I was down.

To my friend Jakub, who were also by my side throughout the course making this one more pleasant and for the leisure hours needed inside (and outside) the university.

I would like to thank both my project supervisor Dr Luis Miralles and the Final Year Project lecturer Damien Gordon for their guidance and help throughout the project and to the teachers who crossed paths with me, for all the teachings and experiences necessary for my academic growth.

List of Illustrations

List of Tables

Table of Contents

1. Introduction

Before considering any aspect of the concept of artificial intelligence, it is necessary to understand the concept of human intelligence, considering that the development of one occurred at the expense of observation and understanding of the other, and therefore, they are interconnected in concept, study, and new possibilities of theories.

In the light of Piaget's constructivist theory, learning comes from a balanced relationship between assimilation and accommodation, which allows the subject to act on the environment in a more elaborate way. throughout existence (Nogueira and Leal, 2018).

Thus, when it reports the experience of Artificial Intelligence, it is expected not only a system to execute previously programmed orders, but also to be a system to find new solutions as new challenges are imposed, becoming part of the environment's action.

The artificial intelligence system is not only capable of storing and manipulating data, but also acquiring, representing, and manipulating knowledge. Manipulation includes the ability to deduce or infer new knowledge or relationships about facts and concepts from existing knowledge and to use representation and manipulation methods to solve complex problems that are often non-quantitative in nature. (Silva; Lenz; et al, 2019 p. 14).

Although the concepts of intelligence transit between countless theories and there is still a considerable amount of doubt and questioning about a possible definition, as mentioned above, Artificial Intelligence walks together with human intelligence and, consequently, they are always in development, of their functions and Definitions.

It is also necessary to emphasize that despite the popular imagination and the futuristic tone of the idea, the concept of Artificial Intelligence is not new and has been worked on over many years of history, according to Silva et al, “this technology began to be developed in the 1950s, with the Dartmouth Summer Research Project Artificial Intelligence at Dartmouth College, Hanover, New Hampshire, USA”. Thus, the concept of AI (Artificial Intelligence) has been improved over decades of study, mainly in the areas of technology and exact sciences.

However, more recently, other areas, in addition to exact sciences and technologies, have started to use Artificial Intelligence to achieve more accurate and satisfactory results. This is the case of medicine, an area in which it will try to show in this study, the connection with AI and the results of this relationship.

For some time, there has been an attempt to develop computerized systems to support clinical diagnosis. Howard Bleish, for more than 50 years, already offered a system that, by evaluating data from a patient, suggested actions to re-establish his hydro-electrolyte balance. Several systems were developed with the aim of offering a list of possible diagnoses for a health problem, with the probability of correctness, using data on the genesis and evolution of this problem, evaluating signs and symptoms of patients, analysing the results of tests performed and proposing possibilities diagnostics (Lobo, 2017 p. 03).

Thus, it is understood that the use of technology has long since ceased to be a result of processes and has become an active agent of development, not only theoretical, but also personal and social in the environment where it operates.

1.1. Project Background

Currently, although imaging diagnosis has expanded its performance capacity, many exams become too costly or laborious: with the emergence of new diseases, such as COVID-19, the speed in diagnosing can mean a life saved. In a scenario like the one that we live in, where every minute accuracy in the result is just as important, it seeks to improve imaging diagnosis through Deep Learning technology, facilitating diagnosis and acting assertively on image reading.

1.2. Project Description

The use of technology is increasingly growing and present in contemporary society, from its use for day-to-day distractions, interaction between peers and use in different areas of knowledge. Technology, once material for study in the fields of physics, engineering, information systems and their peers, today also encompasses areas such as neuropsychology, marketing, advertising, gastronomy, and medicine. There is an interdependent relationship between culture, economy, politics, society, and technology, as can be seen through Diogo's thinking.

It is within this framework of appropriation of usable technical knowledge that it can understand the issue of valuing the technological system. The nineteenth century has a clearly optimistic view of technology insofar as it enhances material progress with a positive connotation. The machine works, physically and metaphorically, as the centre of the technical system, allowing access to the quantitative and qualitative dimensions present in the concept of progress. (Diogo, 2009 p. 8).

Technology has the connotation of functionality, maximizing knowledge. Throughout this monograph, it will be discussed the parallel between Deep Learning technology in the medical context, evaluating the different forms of technological intervention to experience the qualitative improvement in medical performance, consequently reflecting on the life expectancy and treatment of patients. One of the most characteristic aspects of medicine in the current period has been the exceptional inventions and technological innovations specialized in combating different diseases and their rapid, if not immediate, acceptance and incorporation into medical practices (Antas Jr. and Almeida, 2015)

The concept of Deep Learning, in turn, despite being recent, comes from Machine Learning, a concept that was incorporated into society in the last decade and is based on the use of algorithms to extract information from raw data and represent them through mathematical model (Goodfellow, Bengio and Courville, 2017). However, Deep Learning, arises from a deepening of this concept, it is the machine's intelligence capacity in a deeper way. This concept tends to have a strong impact on the health area. The idea that the impact will be great, with a likely transformation of daily clinical practice in some areas such as radiology, pathology, ophthalmology, and ophthalmological diagnoses has been supported by scientific publications (Amato, 2013 p. 47 - 58 apud Leite, 2019 p. 3).

It is known that medical decisions directly depend on the diagnostic hypotheses for assertive treatment, without these crucial data, uncertain decisions can be taken, compromising the patient's health. On the other hand, it also knows that the search for the most appropriate diagnosis requires a series of tests, generating high costs for the patient or government.

In 2009, it was found that 32% of medical errors in the USA resulted from a reduction in the time of interaction between the physician and the patients, producing mistaken diagnoses, non-recognition of the urgency or worsening of the patient's evolution, which would require prescribing or performing relevant actions. Even in hospitals that have electronic medical records, with the possibility of better data collection, it is assumed that 78.9% of medical errors would be related to problems in the doctor-patient relationship, deficient clinical examination, failure to evaluate data from the patient. patient or lack of tests to prove the diagnostic hypothesis. (Lobo, 2017 p.06).

The use of technology, in this sense, presents itself as a way out of this problem, because through its use, it makes the diagnosis faster and more assertive, quantitatively improving the relationship time between doctor and patient. Physicians and pharmacists benefit a lot from these systems that compare data and prescriptions, avoiding cases of inappropriate medications and misdiagnosis (Lobo, 2017). In order to involve technology as a solution to social problems, this project is justified.

1.3. Project Aims and Objectives

The overall aim of the project is a simple one:

- Apply Deep Learning theory in the medical field through proof of concept.
- Map images of lungs compromised by COVID 19 and compare them to other comorbidities to have a clear and accurate diagnosis.
- Build the proof-of-concept prototype for COVID 19 detection.
- Analyse the results obtained.
- Design technology usage scenario in future cases.

1.4. Project Scope

The objective of this project is to present an automatic method to detect and differentiate COVID-19 from Pneumonia in an image using Deep Learning.

Radiography and the most common exam performed in medicine with the analysis of this image can diagnose many diseases such as Pneumonia and Covid-19. In this context, the objective of this work is to propose a detection and classification model in which, through the use of the convolutional neural network, it can say if the analyzed image (the radiograph) is an image with pneumonia, one image with Covid-19 or a healthy radiograph image. The Neural Network Architecture is automatically generated through hyperparameter optimization, the networks will be trained and later validated in an image base that presents cases of the mentioned diseases.

As a person who directly witnessed a late medical examination result of a radiograph to detect Covid-19, which resulted in a family loss, I think it is important as a student to apply the knowledge I acquired throughout the course in the implementation of a Machine Learning in which it would streamline this process, I know that the accuracy will never be 100% accurate, but with the proper training, the accuracy of the results can be sufficiently satisfactory.

Diagnosis is a process that can only be confirmed by physicians and may result in different diagnoses depending on the specialist. Automating or speeding up the diagnostic process is something desirable, because of this, computational methods should be studied to reduce the workload of specialists and also offer a second opinion, increasing the amount of accurate and fast diagnostics.

1.5. Thesis Roadmap

This chapter will talk about how my final year project is organized, the design and development of my thesis also it will be talking about the researching applied in order to complete this project.

Research

This chapter explores background research related to the Covid-19 disease as the Pneumonia and the machine learning, also some research already made in this area. Finally, this chapter will discuss an array of another relevant research completed for this project.

Design

This chapter delves into the methodology chosen for this project and how these choices came to be. Finally, the designed technical architecture and software testing plans will be discussed.

Testing and Evaluation

This chapter describes how all the testing and evaluation of the system was executed. Each phase of testing will be described in detail, followed by an in-depth account of all user feedback received during user evaluation trials.

Conclusions and Future Work

This chapter will reflect on the entirety of the project and will discuss the conclusions drawn, personal reflections made, and the future work planned for the project. Planning for evaluation and for testing is discussed and what the goals going forward will be.

2. Related Works

2.1. Introduction

This Chapter describes studies that used computational techniques to determine the presence of disease in the lungs in patients.

2.2. Alternative Solutions

There are some working on the area of the use of Machine Learning for the detection of diseases trough images.

The work of Heckerling et al. (2004) made use of medical examinations to perform the automatic detection of pneumonia but differs from the way the information was coded. The data were not detailed information on the state of the lung, but binary information about the presence of symptoms. The technique used to perform the classification was artificial neural networks. This work also made use of genetic algorithms to automatically generate a network architecture. The result of this study showed high values of 94% AUC in test tests.

Continuing with methods that only use radiography to perform detection, (Rajpurkar et al. 2017) use convolutional neural networks to perform automatic extraction of characteristics and classification. The base used by this author is public and presents a total of 1,353 different pathological images in different image qualities. The author achieved an AUC of 76.8% with this technique.

Another work in this area belongs to (Kermany et al., 2018). This work also makes use of only radiographs and learning by convolutional neural networks, but presents a more robust and detailed database, presenting 3,883 images. The results of this study were excellent, presenting an AUC of 96.8% for the detection of pneumonia and 90.7% for the differentiation of type of pneumonia.

One of the most recent works in this area belongs to (Yang et al. 2021) he investigate and compare several deep learning enhanced techniques applied to X-ray and CT-scan medical images for the detection of COVID-19, he uses powerful pre-trained CNN models, VGG16, DenseNet121, ResNet50,and ResNet152, for the COVID-19 CT-scan binary classification task, The accuracy and F1-score were both above 96% in the diagnosis of COVID-19 using

CT-scan images and a high accuracy of 99% was achieved by enhanced VGG16 in the detection of X-ray images from COVID-19 and pneumonia.

Conclusion

The work to be presented will continue the exploration of convolutional neural networks for the extraction of characteristics and automatic classification of chest X-rays, but with the differential of automating the generation of the architecture of this network, like the work of Heckerling et al. (2004), applying this for a detection of Covid-19.

2.3. Existing Final Year Projects

A couple of Final Year Projects from previous years that use Machine learning and python.

Recommovie: Movie Recommender System - Ivan Yanez

Recommovie is web application that recommends movies based on the user's previous ratings and ratings from similar users on similar movies. the web application has characteristics similar to those used by Netflix and Amazon when suggesting their products.

The use of machine learning to guess the user's preferences, and the storage of this information and the development of the user-friendly web application and the interconnection with the database.

The web application is built using Python the data is stored in a PostgreSQL database. The web application was made using Django Framework and React.js.

***Auto-Efficiency – Machine Learning for Predictive Analysis and Recommender System -* Glory Pierce Eguare.**

The purpose of this project is to create a user-friendly computer application that utilises information from the user input and returns a feedback output in a graphic presentation for ease of interpretation.

It uses machine learning computer application that can predict the fuel efficiency in kilometres per litre and in miles per gallon of a car based on inputs such as car weight, horsepower, year made, displacement.

3. Literature Review

3.1 Introduction

In this chapter, it will be approach, through the literature review, the concepts that affect this work: imaging diagnosis, Deep Learning, the reading of radiographic images through Deep Learning technology, as well as the general parameters about SARS-COV-2 (COVID-19), viral pneumonia and healthy lung. The goal is to build a text in a clear and solid way, to serve as a basis for the construction of the proof of concept.

3.2. Diagnosis by Image

Diagnostic imaging has revolutionized the area of medicine and has still revolutionized as it is updated in an intimate relationship with technology. However, first, it will be necessary to understand a little of its history, in parallel, to understand the dimension of the importance of this segment within the health area.

According to the São Paulo Society of Radiology and Diagnostic Imaging (SPR), this story begins in 1985, when German physicist Wilhelm Conrad Röntgen, through radiation experiments, gets the first x-ray image in history, containing his wife's hand.

On November 8, 1895, Wilhelm Conrad Röntgen, then professor of physics at the University of Würzburg, Bavaria, Germany, discovers a new species of radiation produced by the passage of an electric current through a glass tube under vacuum, and which had the unique quality of, although invisible to the naked eye, producing fluorescence by focusing on a paper impregnated by barium and platinum cyanide. More impressive was the ability of these rays to cross solid bodies (wood, paper, parts of the human body), with greater or lesser intensity, depending on the nature of the material. On December 28, 1895, his 10-page work "Über eine neue Art von Strahlen" ("On a new kind of lightning"), was submitted to Publication and briefly and objectively describes its discovery. (Arruda, 1996).

In fact, Rontgen's radiation experiment brought the possibility of seeing the human body from the inside, without having to open it, as was usual at the time, even making it win the Nobel Prize. However, the effects of radiation treatments were extremely harmful to the population of the time. There are numerous reports of people who survived shooting but succumbed to radiation when doctors resorted to Roentgen's invention to locate the bullet. (Vale, 2009 P. 60).
Diagnostic imaging is the essence of technology applied to medicine: from the beginning, when physicist Wilhelm Conrad makes discoveries about x-rays (Brentano, Schnorr, Lima and Perez, 2015), it can already be associate the use of this object as one of technology as well. However, this same technology has gone through several levels of development.
The great evolution of Radiology occurred from the 70s-80s, with great technological advances, which allowed a much more accurate diagnosis. Since then, radiology has gained great space in medical practice. The advent of new diagnostic methods, such as ultrasound, digital mammography, bone densitometry, computed tomography and multislice, PT/CT and PET/MRI, 3T magnetic resonance imaging and digital radiology, gave rise to a new specialty, imaging. That is, a set of diagnostic techniques that provide the doctor with a visual image of the various parts of the human body, whatever the radiation or the way or the wave used to explore the patient. (Oliveira et al, 2008, p.02).
In the same period, along with this new possibility of diagnosis within medicine, specialists in the area also arise, in order to become a specialist in reading these images and facilitate the diagnosis of the patient.
Still, the medical movement began with the help of physicists and other specialists, to improve Wilhelm's invention, expanding the area of imaging diagnosis. Currently it has different tests, such as: Conventional Radiology, Computed Tomography, Magnetic Resonance Imaging and Ultrasonography (Brentano, Schnorr, Lima and Perez, 2015). Conventional radiology has already been highlighted previously in the description of this text, so it will not go deep into its aspects. Computed tomography, in turn, allows an expanded study of the brain, with images, as if they were small slices of the entire area.
The device consists of an X-ray source that is activated while performing a circular movement around the patient's head, emitting a fan-shaped X-ray beam. On the opposite side of this source, is located a series of detectors that transform radiation into an electrical signal that is converted into a digital image. Thus, the images correspond to sections ("slices") of the skull. The intensity (brightness) reflects the absorption of X-rays and can be measured on a scale (Hounsfield units) (Yamashita and Junior, 2001 p. 02).

Magnetic resonance imaging can be subdivided into functional or not. In some literature reviews, it was also allowed to find the subdivision as nuclear magnetic resonance imaging, or functional magnetic resonance imaging. In general, magnetic resonance imaging can be divided into three stages, the first of which concerns alignment.

Alignment refers to the magnetic property of nuclei of some atoms, which they tend to orient themselves parallel to a magnetic field (like a compass in relation to the earth's magnetic field). For physical reasons and for its abundance, the hydrogen nucleus (proton) is the element used to produce images of biological beings. (Yamashita and Junior, 2001 p. 04).

The second stage suggests excitement. “Each hydrogen nucleus is known to "vibrate" at a certain frequency proportional to the magnetic field in which it is located. Thus, at 1.5 T, hydrogen has a frequency of 63.8 MHz. The device then emits an electromagnetic wave at that same frequency. There is a transfer of energy from the wave emitted by the equipment to hydrogen atoms, a phenomenon known as resonance (Yamashita and Junior, 2001 p. 04).

And the third and final stage concerns radio frequency detection. When hydrogen nuclei received energy, they became unstable. Upon returning to their normal state, they emit electromagnetic waves at the same frequency (63.8 MHz - radio waveband). Then the equipment detects these waves and determines the position in space and the energy intensity. This intensity is shown as "brightness" in the image, using the nomenclature "signal strength". (Yamashita e Junior, 2001 p. 04).

It is also necessary to talk about ultrasound, this type of exam, unlike what it has seen so far, does not use radiation to obtain results. Ultrasound technology is based on the frequency of sound waves, allowing the verification of any type of anatomical variation that does not match the conventional one. Currently, it is possible to diagnose different areas of the body using ultrasonography.

Like technology, the principles of ultrasound are well known and its use as a therapeutic and primary diagnostic modality in radiological practice is well founded. In recent years, ultrasound technology has evolved and is now used outside the specialty of radiology. Ultrasonography is commonly used in obstetrics and cardiology. The application of ultrasound in other areas, such as emergency medicine and the intensive care unit, is growing rapidly. As its use expanded beyond the realm of traditional radiology, manufacturers had to adapt the equipment, building smaller, portable devices that could be used at the point of care. (Silva, 2017).

.

Understanding the operation and specifications of diagnostic imaging is crucial for the development of this work, because, through this understanding, it was able to define the main problems and how deep learning technology can work on them, contributing to the growth of the area. medical and above all, for the most appropriate treatment for patients. According to Briani, the influence of the labour market and the adoption of increasingly advanced technological resources in medical practice brings undeniable consequences for teaching (Briani, 2001, p. 25 -73 apud Oliveira, 2014, p. 21), therefore, the need for technology in the medical field becomes unquestionable.

According to Wanderlei (2020), there is a range of different exams based on image diagnosis, however, some better accept the interference of Deep Learning technology in their diagnostic process. Computed tomography stands out among them because, through it, doctors can visualize internal structures such as bones, organs, and other tissues. In addition, the images are taken from various angles. However, in the computed tomography process, an average of 320 imaging slides are generated, which makes the process very costly.

One cannot fail to mention the importance of ethical issues and professional responsibility that specialists in the field may face. Issues that Fenelon (2003) cited by Malpractice, Recklessness and Negligence. Also, according to the author, Malpractice is characterized by lack of practical preparation or insufficient knowledge. Imprudence, in turn, is characterized as an act of precipitation, nonsense through the image of the diagnosis. Negligence, as the name itself refers to, is the moment when the doctor fails to make certain decisions, causing harm to the patient. The use of Deep Learning technology applied to diagnostic imaging exams is intended to minimize, mainly, Imprudence and Malpractice errors.

3.3. Machine Learning

It arises from the question, can a machine learn on its own to perform a task? and its first origins date from 1830-1840 with the first general purpose computer, devised by Charles Babbage, although without that being the reason for creating it (for the beginning of AI and ML, it still had to wait more than 100 years). What he wanted was to find a way to automate certain calculations in the field of mathematical analysis, and the concept of general-purpose computing had not yet been invented. It is noteworthy that Babbage never saw the machine built of him (which was totally analogue) although his design was absolutely valid, it can now be seen in the science museum in London.

In 1959, IBM scientist Arthur Samuel coined the term machine learning after making a program to play checkers that he was able to learn while playing against himself. Machine learning implies that computers are capable of discovering how to carry out tasks for which they are not specifically programmed. The goal is for computers to learn from the data provided, applying statistical techniques to carry out certain tasks. When the tasks are simple it is possible for a programmer to explain the algorithm needed to solve the problem, but when the task is more complex it may be impossible or very difficult for a human being to manually create the necessary algorithms. In practice, it may be more efficient to help the machine develop its own algorithm, rather than having human programmers specify each necessary step. Machine learning uses several approaches to get a machine to learn, the main ones are:

3.4. Deep Learning: Concepts, Applications and Structures

Deep Learning initially, it is not just a term used in the field of computer science, but a general term that tells us about the deepening of learning, knowledge and/or intelligence.

Thus, it is crucial that understanding what human intelligence/learning means, so that understanding the basis of artificial intelligence. Although human intelligence has gone through several concepts, from quantitative testing of the Intelligence Quotient (IQ), to analysis of affective dimensions, also passing through concepts such as Multiple Intelligences by Howard Gardner and the theory of Emotional Intelligence by Daniel Goleman and still that its definition is diverse and is supported by multiple factors, an important factor to be taken into account is the ability to solve problems, adapting the brain to new learning (SOBRAL, 2013). In this sense, with the historical evolution of the concept of intelligence, and its conceptions and typology, it is currently possible to perceive it as the human capacity to solve problems of different orders: affective, volitional, and cognitive. (Sobral, 2013 p. 14).

The concept of Artificial Intelligence began to be used and studied from 1956, when computer scientist John McCarthy and a group of fellow students proposed to think about the use of human language, the ability to solve problems, form abstractions and concepts by computer machines. That is, more than realising functions of which there is already a defined programming, with established functioning, John McCarthy and his colleagues believed in the ability of machines to reproduce aspects of human cognition. (Reis et al, 2019). Since then, this concept of intelligence within Computer Science has not stopped evolving, every year, new studies and improvements arise in the area, so that today it comes to concepts such as Machine Learning and Deep Learning.

The world called “technology” evolves at the speed of Big Data volume, which is data produced well beyond Petabytes, Exabytes or even Zettabyte – which represents a billion Terabyte. This evolution leads AI to evolve as well, forming Machine Learning “machine learning” which, in turn, is evolving into Deep Learning “deep graphics-based learning”. Although machine learning has become an integral part of data processing, one of the main differences from deep learning is that it requires manual intervention in the selection of resources to be processed, whereas deep learning is intuitive, residing in these subjects for future studies. (Reis et al, 2019 p. 10).

Thus, the concepts of Machine Learning and Deep Learning are layers of Artificial Intelligence. The first concept created by John McCarthy and his friends was transformed to such an extent that it develops the concept of Machine Learning, the concept of the latter evolved in such a way, with increasingly complex structures, that today it is developing Deep Learning.

The purpose of the computer scientist is to send data to machines, so that from this data, it is possible to create concepts, in a process of systemic comparison, where after countless approximations, trials and errors, it becomes possible to build a pattern of assertiveness, that is, learning. This pattern will generate estimates using several different algorithms. The classification through Machine Learning can be binary or not. Clearly, it is referring so far to a broad classification of what Machine Learning can be, however, based on this context, it will start discussing the specificities of this model within Artificial Intelligence.

An extremely important concept for understanding how Machine Learning works is the concept of Neural Networks. The principle of Neural Networks applied in Artificial Intelligence, more specifically in Machine Learning, is inspired by networks human neural pathways.

Human neural networks concern the functioning of the brain, knowing that, the brain is made up of millions of neurons. Neurons perform the movement it calls synapses: synapses are the exchange of information between these neurons, which happen at different levels, over and over again (Ferneda, 2006).

A neuron is a cell formed by three sections with specific and complementary functions: body, dendrites and axon. Dendrites capture the stimuli received in a certain period of time and transmit them to the neuron's body, where they are processed. When such stimuli reach a certain threshold, the cell body sends a new impulse that propagates through the axon and is transmitted to neighbouring cells through synapses. This process can repeat itself in several layers of neurons. As a result, incoming information is processed, which can prompt the brain to command physical reactions. (Ferneda, 2006 P. 02).

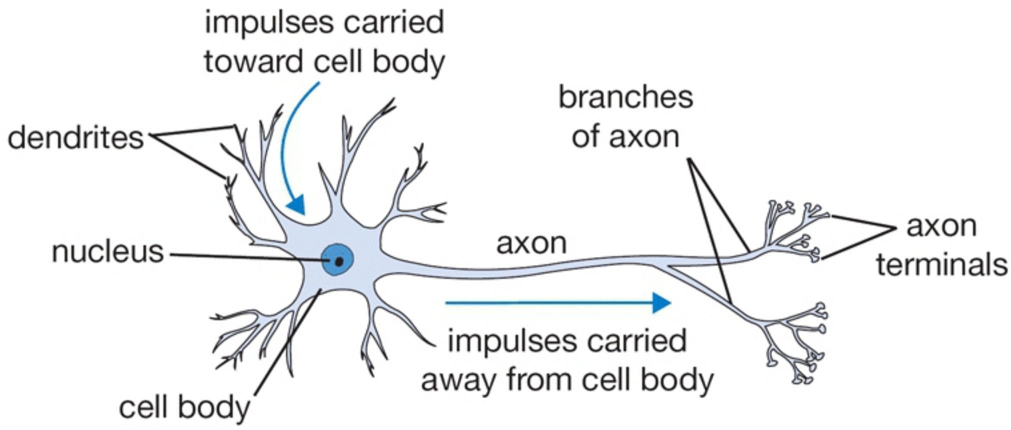


Figure 1 - Structure of a human neuron

Source: cs231n.github.io

Thus, the ability to perform complex functions, learning and cognitive ability of human beings comes from the communication structure between neurons. Therefore, the best way to create an artificial intelligence system is based on the functioning of the human brain. The shape and processing of an artificial neuron is very similar to a human neuron, as you can see in the figure below, its morphology and structure through a mathematical model.

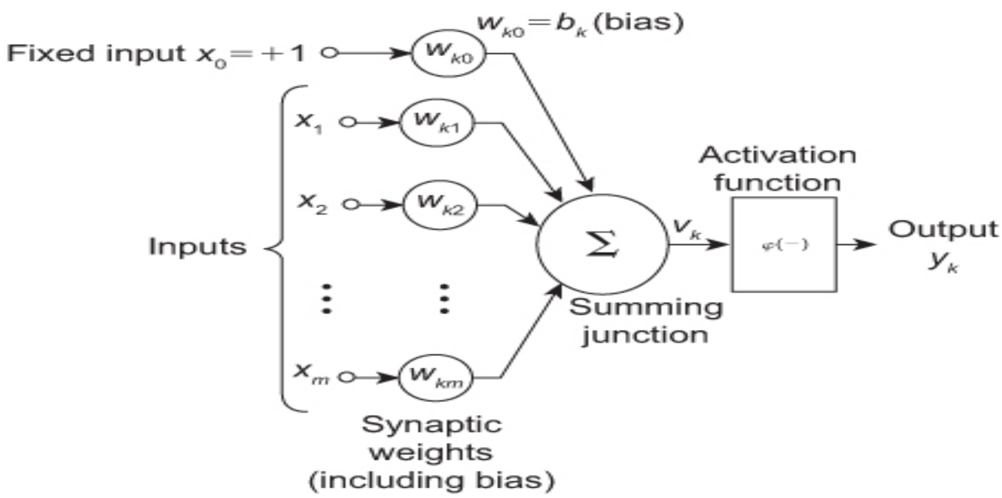


Figure 2 - Mathematical model of a neuron

Source: MATLAB® and Its Applications in Engineering

This model has three basic elements in the composition that need to be understood: the sets of input connections (represented by the factors X1, X2 and Xn), each of these connections represented by different weights (P1, P2 and Pn). An adder, which accumulates and sums the weights, represented by the symbol $\sum$ and the activation function, φ, which delimits the frequency of propagation of information in the output (y) with a fixed value (Ferneda, 2006).
The behaviour of connections between neurons is simulated through their weights. The values of such weights can be negative or positive, depending on whether the connections are inhibitory or excitatory. The effect of a signal coming from another neuron is determined by multiplying the value (intensity) of the received signal by the weight of the corresponding connection (X1 P1). The sum of the X1 P1 values of all connections is performed, and the resulting value is sent to the activation function, which defines the output (y) of the neuron. (Ferneda, 2006 p. 26).
However, although the inspiration of the functions of Machine Learning and Deep Learning comes from the functioning of the human brain, it cannot say that its structure is exactly the same, both in morphology and functionality. The task of trying to create a copy of the human brain is the responsibility of computational neuroscience (Oliveira, 2019). Machine Learning and later Deep Learning, use the concept of information sharing through neural networks, in order to solve complex problems, with or without human intervention during this process. Therefore, it can deduce, as Oliveira said, that an artificial neural network is mainly a data processor.
“A neural network is a parallel distributed processor made up of simple processing units. These units store experimental knowledge. The process to carry out the learning is through a Machine Learning algorithm, whose function is to modify the network weights to reach a final objective.” (Oliveira, 2019 p. 02).

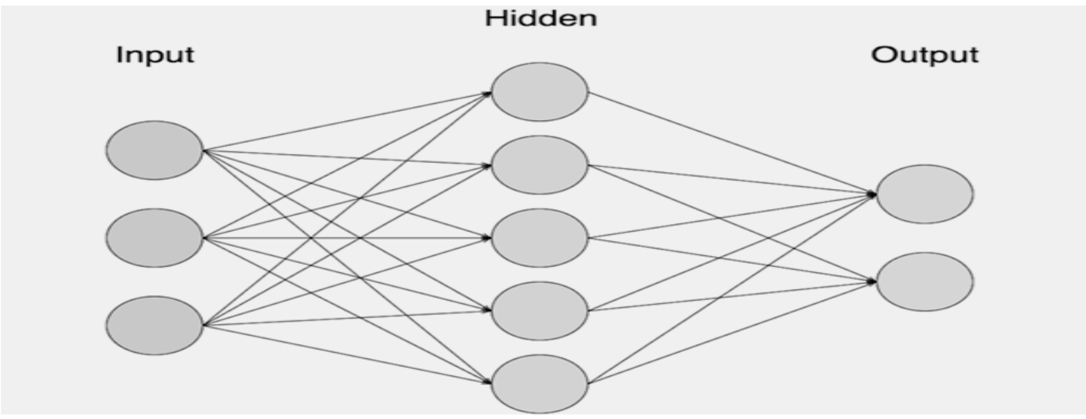


Figure 3- Representative model of an artificial neural network, with the Input, Hidden and Output

Source: Oliveira, 2019

The image above can be seen as an artificial neuron, but more developed to perform the designated functions. As with the neuronal model, there is a channel for data entry, one for processing and one for final value, or results. In this case, the Input is responsible for receiving the information that will be decoded, the sector called Hidden, is responsible for processing this data initiated in the Input, the more Hidden layers a network presents, the greater its ability to extract more elaborate statistics, and for Finally, the Output sector, based on all the statistics extracted from Hidden, is able to respond with a final value, or with a final estimate. It is important to emphasize that there is a latent difference between this neural network model and the deeper neural network models, which is the case of Deep Learning technology, as it will see below.

The Deep Learning technology emerges as an improvement of Machine Learning: a more sophisticated type of Machine Learning algorithm, built on the principle of neural networks. However, unlike the first existing algorithms, it can support and work with a large flow of data, called Big Data, and function as a mind of its own through overlapping non-linear data processing layers. While Machine Learning is a technology studied and applied since the 1980s, Deep Learning emerged around the 2000s, when it had more sophisticated computers, with a large number of accessible data (Chagas, 2019). In simple terms, it can say that Deep Learning are complex algorithms built from a stack of several layers of "neurons", fed by huge amounts of data, which are capable of recognizing images and speech, processing natural language and learning to perform extremely advanced tasks without human interference. The main application of Deep Learning algorithms are classification tasks, especially image recognition.

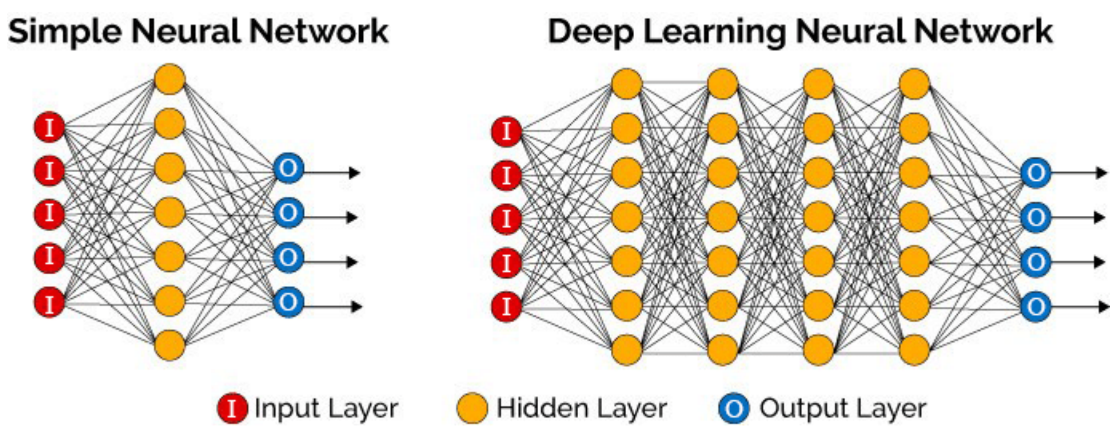


Figure 4 - Comparisons between a simple neural network and Deep Learning Neural Network

Source: Vázquez, 2017.

Clearly shown in Figure 4, it is possible to see the evolution of the artificial neural network, if before it was necessary to compute a lot of data to get a result, now the neural network has become faster and more accurate.

Given these circumstances, one family of algorithms thrived most: neural networks. In the past, some of the disadvantages of neural networks were the big amount of data needed to get good results and low training speed. Recent events have helped to mitigate these disadvantages, enabling the development of multilayer neural networks, hence the name Deep Learning. Deep Learning itself is not a technique, but a multi-layered neural network. More interestingly, there are some specializations. Some of them had very promising results: Convolutional Neural Networks (Convolutional Neural Networks or CNNs) and Generative Adversarial Networks (GANs). (Gulli, Kappor And Pal, 2019)

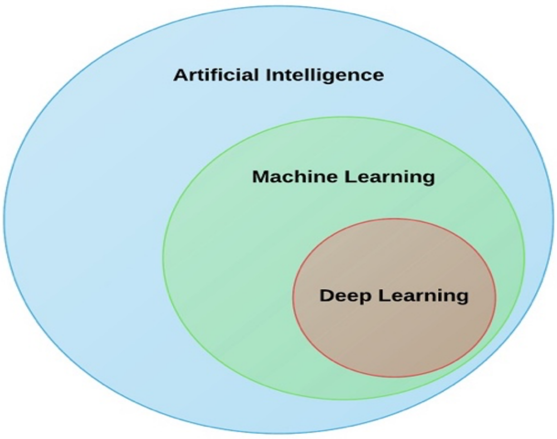


Figure 5 - Deep Learning Situational Graph

Source: (Gulli, 2019).

3.5 CNNs

Although within deep learning there are other types of neuron networks, such as recurrent neural networks (RNNs), this work will be focus on the Convolutional Neural Networks (CNN), are a variation of a multilayer perceptron and its application is carried out in two-dimensional arrays, being very effective for artificial vision tasks, such as image classification and segmentation.

The foundations of convolutional neural networks are based on the Noncognition, introduced by Kunihiko Fukushima in 1980 (Fukushima). This model was later improved by Yann LeCun in 1998 by introducing a learning method based on back propagation in order to train the system correctly. In 2012, they were refined by Dan Ciresan and implemented for a graphics processing unit (GPU) with impressive results.

A CNN is a network that is passed an image as input, assigns an importance (learned based on weights and biases) to various features or objects in the image, and can differentiate one from another. The pre-processing required in a CNN is much less compared to other classification algorithms. In addition, it can capture both the spatial and temporal dependencies of an image, through the application of relevant filters. The main goal of CNNs is to reduce images in a way that makes them easier to process without losing features, which are critical for good prediction. This is important when designing an architecture that is not only good at learning features, but also scalable to massive data sets.

Its topology is divided into multiple learning stages made up of a combination of: convolutional layers, non-linear processing units (Relu) and data layers under sampling. The convolution operation helps in extracting useful features from locally correlated data points. The output of the convolution is then assigned to the nonlinear processing unit (activation function), which not only helps to learn abstractions but also integrates nonlinearity into the feature space. This nonlinearity generates different activation patterns for different responses and thus facilitates the learning of semantic differences in images. The output of the nonlinear activation function is often followed by subsampling, which helps to summarize the results and also makes the input invariant to geometric distortions. It does not need a separate feature extractor, as it has this capability automatically, so a CNN does not need extensive processing to learn good representations from raw pixels.

Deep architectures often have an advantage over shallow architectures when it comes to complex learning problems. Stacking multiple linear and nonlinear processing units in layers provides the ability to learn complex representations at different levels of abstraction. Consequently, in recognition tasks consisting of hundreds of image categories, deep CNNs have shown substantial performance improvement over conventional models.

It has been shown that given enough training data, deep CNNs can learn the invariant representations and can achieve human-level performance. In addition to its use as a supervised learning mechanism, the potential of deep CNNs can be harnessed to extract useful representations from large scale unlabelled data. It has also been shown that different levels of

features, both low-level and high-level, can be transferred to a generic recognition task by exploiting the concept of transfer learning.
From the late 1990s to 2000, several improvements were made to the CNNs learning methodology and architecture to make them more scalable to large, heterogeneous, complex, and multiclass problems. Innovations in CNNs include different aspects such as modification of processing units, parameter and hyperparameter optimization strategies, design patterns and layer connectivity, etc. Applications based on CNNs became prevalent after the incredible performance of AlexNet on the ImageNet dataset in 2012 (Krizhevsky). Major innovations have been proposed at CNN since then and are largely attributed to the restructuring of processing units and the design of new blocks.

3.5.1 Basic components CNNs

Convolutional layer

The convolutional layer is composed of a set of kernels or convolutional filters that perform convolution operations while scanning the input (image) with respect to its dimensions. Its hyperparameters include filter size and stride. The resulting output is called a feature map or activation map. The kernel contains a specific set of weights that it multiplies with the corresponding elements of the receptive field. The convolution operation can be expressed as follows:

$$f_l^k(p,q) = \sum_c \sum_{x,y} i_c(x,y) \cdot e_l^k(u,v)$$

where, ic(x, y) is an element of the input tensor of the image ic, this is multiplied by the index ekl(u, v) of the k-th convolutional kernel kl of the l-th layer. While the output feature map of the k-th convolutional operation can be expressed as $F_l^k = [f_l^k(1,1), ..., f_l^k(p,q), ..., f_l^k(P,Q)]$.
Due to the weight-sharing ability of the convolutional operation, different sets of features within an image can be extracted by sliding the kernel with the same set of weights into the image, and thus makes the CNN parameter efficient in compared to fully connected networks. The convolution operation can be classified according to the type and size of the filters, the type of padding, and the direction of the convolution.

Grouping layer

Once features are mined, their exact location becomes less important if their approximate position relative to other features is preserved. Clustering or subsampling is a local operation that summarizes similar information in the vicinity of the receptive field and generates the dominant response within this local region.

$$Z_l^k = g_p(F_l^k)$$

The above equation shows the grouping operation, where Zkl represents the feature map clustering at the lth layer for the kth feature map entry Fkl. gp(·) defines the type of grouping operation. The use of the grouping operation helps to extract a combination of features, which are invariant to translational changes and small distortions. The reduction of the size of the feature map to the set of invariant features not only regulates network complexity, but also helps increase generalizability by reducing overfitting. Different clustering types are used, such as maximum, average, L2, overlap, spatial pyramid clustering.

Activation function

The activation function serves as a decision function and helps to learn intricate patterns. Selecting a suitable activation function can speed up the learning process. The activation function for a convolved feature map is defined:

$$T_l^k = g_a(F_l^k)$$

Equation above shows that F^k_l is an output of a convolution, which is assigned to the activation function $g_a(\cdot)$. This function adds the nonlinearity and returns a transformed output F^k_l for the l-th layer. In the literature, different activation functions are used such as sigmoid, tanh, maxout, SWISH, ReLU and ReLU variants such as leaky ReLU, ELU and PReLU. However, ReLU and its variants are preferred as they help to overcome the vanishing gradient problem. One of the proposed activation functions recently is MISH, which has shown better performance than ReLU in most recently proposed deep networks in reference data sets.

Batch normalization

Iofe and Szegedy (Iofe) observed that the change in the distribution of network activations due to the change in hyperparameters during training caused a longer time
of training and called this change of internal covariance. To address issues related to internal covariance change within feature maps, they introduced batch normalization, which unifies the distribution of feature map values by setting them to zero mean and unit variance. Furthermore, it smoothers the gradient flow and acts as a regulating factor, which helps to improve the generalization of the network. The internal covariance shift is a shift in the distribution of hidden unit values, which slows down convergence (by forcing the learning rate to a small value) and requires careful parameter initialization. Batch normalization is shown for a transformed feature map F^k_l in the equation.

$$N_l^k = \frac{F_l^k - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Equation above shows that N^k_l represents the normalized feature map, F^k_l is the map of input characteristics, μB y σ^2_B correspond to the mean and variance of the feature map for a minibatch. ϵ is added to avoid division by zero. Iofe and Zegedy observed in their experiments that the use of batch normalization, part of shortening network convergence time, it also reduced the need for dropout.

Dropout

Dropout introduces regularization within the network, which ultimately improves generalization by randomly skipping some units or connections with a certain probability. In CNNs, multiple connections that learn a nonlinear relationship sometimes adapts causing overfitting. This random dropout of a few connections or units produces several slimmed-down network architectures, and finally a representative network with small weights is selected. This selected architecture is then considered as an approximation of all proposed networks.

Fully connected layer

The fully connected layer is mainly used at the end of the network for classification. Unlike pooling and convolution, it is a global operation. It takes information from the feature extraction stages and globally analyses the output of all previous layers. Consequently, it performs a non-linear combination of selected features, which are used for data classification.

Loss function

The loss function or also called the objective function. Normally in CNNs it is a matter of minimizing the value of this function since it evaluates the difference between the output values of the network and the true values. Depending on the type of problem it is facing, it will use different types of loss functions. Therefore, it oversees managing the weight adjustments throughout the network. Before network training begins, the weights in the fully connected and convolution layers are given random values. Then, during training, the loss layer continually checks the assumptions of the fully connected layer against the actual values with the goal of minimizing the difference between the assumption and the actual value as much as possible. The loss layer does this by adjusting the weights on both the convolution and fully connected layers.

Optimization methods

Optimizers link the loss function and model parameters by updating the model in response to the output of the loss function. That is, the optimizers shape the model to be as accurate as possible when combining the weights. The stall function tells the optimizer when it is moving in the right or wrong direction.

- **SGD algorithm**: Stochastic gradient descent optimizer, SGD also known as incremental gradient descent, is an iterative procedure for optimizing a differential objective function, a stochastic approximation of gradient descent. The stochastic algorithm does not need to remember which examples were seen during the previous iterations; therefore it can process examples during the training stage.

- **SGDM algorithm**: The boosted stochastic gradient descent optimizer, SGDM, is a variation of stochastic gradient descent used for faster convergence of the loss function. Uses a single learning rate for full parameters. This gives gradient descent a boost to avoid getting caught in the local minimum. However, using a (too) large boost coupled with a large learning rate may result in a jump from a suitable solution (local minimum) due to a large update step.
- **RMSProp algorithm**: The root means square propagation optimizer, RMSProp, uses different learning rates for each parameter. The learning rates can automatically adapt to the loss function that is optimized to improve network training. The core idea of RMSprop is to keep the moving average of the squared gradients for each weight, and then divide the gradient by the square root of the mean square. That is why it is called RMSprop (root mean square).
- **Adam Algorithm**: The Adam algorithm is an optimization algorithm based on first-order gradients of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is simple to implement, is computationally efficient, has low memory requirements, is invariant to diagonal scaling of gradients, and is well suited to problems that are large in terms of data and/or parameters. The method estimates individual adaptive learning rates for various parameters from calculations of the first and second moments of the gradients. Some of the benefits of Adam are that the number of parameter updates does not vary with gradient scaling, the size of its steps is almost limited by the stride size hyperparameter.

3.5.2. Learning strategies

Learning from scratch

All parameters of the CNN are initialized randomly following random Gaussian distributions.

Transfer learning strategies

The advancement of deep learning in recent years has enabled great progress in many fields, by being able to address complex problems and obtain very good results, but the training time and the amount of data required for this type of system is much greater than for traditional machine learning systems. It tries to solve this problem by sharing the details of the most used deep learning networks trained for data sets so that other people can use them to train their models.

- **Pre-trained models ready to use as function extractors** The use of pre-trained models as function extractors consists of using the models without their last layer, where it can be replaced by another one that is more suited to the problem to be addressed. This is possible since deep learning systems are made up of layers that learn features in a hierarchical way, where as the model progresses, the layers are semantically richer. Therefore, a pre-trained network is used for another problem, but it is useful for extracting low-level features, which can then be refined to fit the problem to be addressed. For example, if AlexNet is used without its final classification layer, it will transform images of a new domain task into a 4096-dimensional vector based on their hidden states, allowing features to be extracted from a new domain task, using knowledge of a source domain task. This is one of the most widely used methods to perform transfer learning using deep neural networks.
- **Fine-tuning of ready-to-use pre-trained models** This is a more complicated technique, in which not only the final layer is replaced (for classification / regression), but also some of the previous layers are selectively re-trained. Deep neural networks are highly configurable architectures with various hyperparameters. The initial layers capture generic features, while the later layers focus more on the specific task at hand. Using this information, it can be freeze (set weights) certain layers while re-training them or adjust the rest of them to suit the needs. In this case, knowledge in terms of the general architecture of the network is used and its states are used as a starting point for the retraining step. This, in turn, helps to achieve better performance with less training time.

3.5.3 Architectures

Since they emerged in 1989, countless improvements have been made to their architecture. These improvements can be categorized into parameter optimization, regularization, structural reformulation, among others. However, the main improvement in the performance of the CNNs came from the restructuring of processing units. Most of the innovations in CNN architectures have been in relation to depth and spatial exploitation. Although it will not delve into them, depending on the type of architectural modifications, CNNs can be classified into seven different classes.

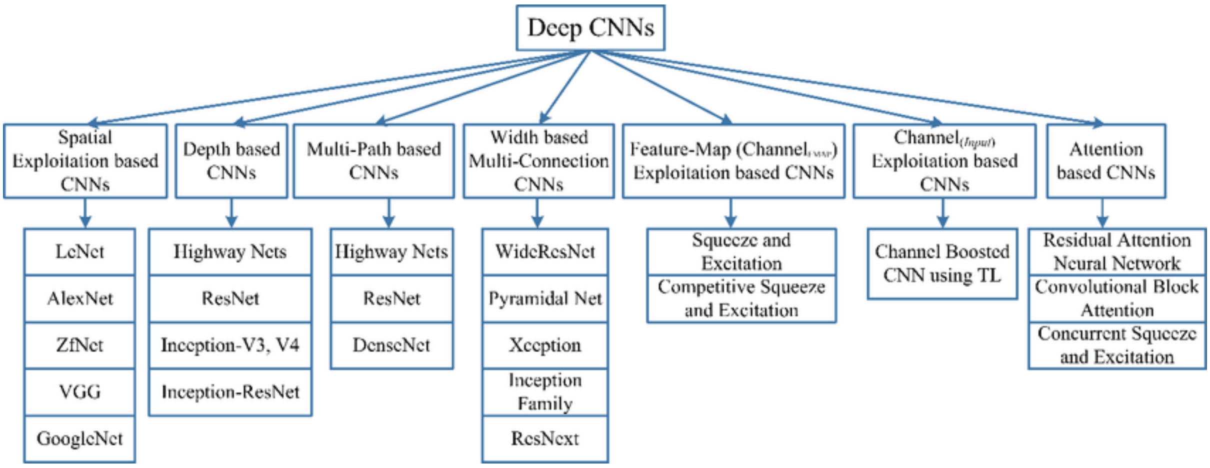


Figure 6 - Taxonomy of deep CNN architectures.

Source:(researchgate.net)

3.5.4 Training, Validation and Test Sets

To start training a model in deep learning, the data set must be divided into three groups or sets: train, validation, and test.

Figure 7 - Example of splitting a training, validation, and test data set.

Source:(towardsdatascience.com)

The training set is used, as its name suggests, to train the created model. Therefore, this data set is used so that the values of the model parameters are obtained by the learning algorithm used. It would be necessary to modify the value of some hyperparameter in case the network is not conditioned to the input data, retrain the model with the training data and evaluate with the validation set.

3.5.5. Iteration And Epoch

They are two hyperparameters that, in neural network terminology, can be defined as follows:

• Iteration: refers to the number of times a batch of data (established by the batch size) passes through the neural network. This pass is forward and backward

• Epoch (or epoch): number of times the neural network sees the entire training data set. Each time the algorithm sees all the samples, an epoch is completed. A good training tactic, which it will be seen later, is to increase the number of epochs until the validation accuracy starts to decrease (there it detects overfitting).

Let's see an example, in order to clarify these two concepts and the relationship between them:

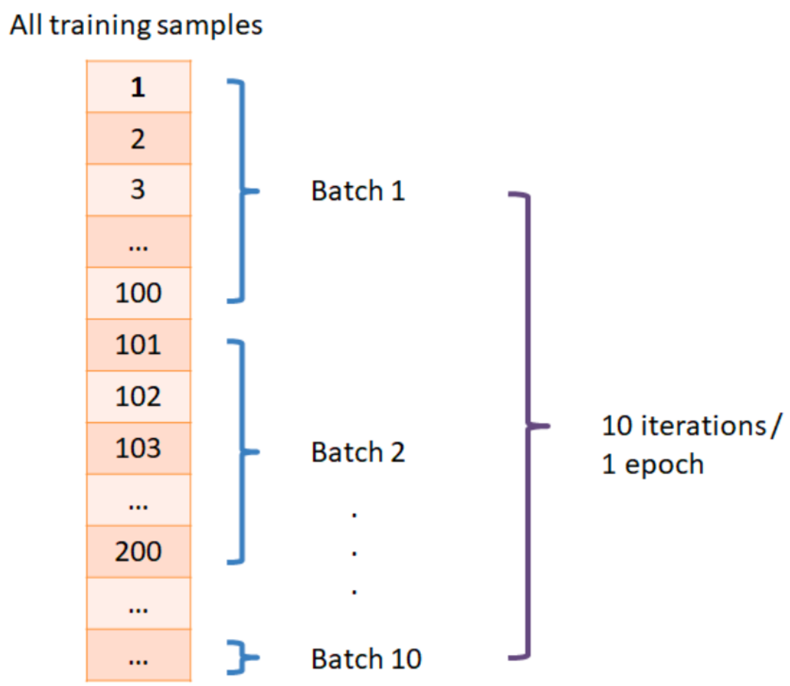


Figure 8 - Batch size, iteration, and epoch.

Source:(mediun.com)

It has a set of 1000 samples and a batch size of 100 training samples, and it states that it want the network to run 1 epoch (N = 100, batch size = 100, epochs = 1).
Therefore, in each epoch it has 10 batches (1000/100 = 10). Each of these batches goes through the neural network, so 10 iterations will be processed per Epoch and, as it has only specified 1 single epoch, after completing the 10 iterations, all the training samples will have.

3.5.6 SoftMax Classification

The last layer is the classifier layer that will have as many neurons as there are classes to predict. In it the SoftMax activation function is applied, as mentioned in the previous section. This function is based on calculating the evidence that an image can belong to a particular class and then converting it into a probability. To measure the evidence that an image belongs to a particular class, SoftMax performs a weighted sum of the evidence that each of its pixels belongs to that class.

Once this calculation is done, the evidence from each of the different classes turns the evidence into probabilities. To perform this transformation, develop the following formula, to know the probability of belonging to a given class (i), SoftMax uses the exponential value of the calculated evidence and normalizes it

$$f_i(x) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Figure 9 - Probability of belonging to class i.

Source(vitaflux.com)

Keras has a commonly used callback function (training data monitoring function) to avoid overfitting as well. Early Stopping allows you to specify a performance measure that will be monitored and when this function is triggered it will stop the training process.

This study, focus on convolutional neural networks, as these networks are suitable for developing the objective: image diagnosis. Convolutional networks are a neural network model, within Deep Learning, capable of performing actions without the need for the supervision of a data scientist, configuring itself as an autonomous neural network, allowing Artificial Intelligence to be trained with a logic that approaches the trial-and-error scheme (Aliger, 2020). This kind of neural network is very effective for reading images, for example, being widely used in autonomous car technologies, robotics, drones, security, image diagnosis and treatments for the visually impaired (Aggarwal, 2019)

However, it needs to have as a starting point, the certainty that convolutional neural networks do not see an image in the same way as a human being, and this makes all the difference when it is proposed to understand the logic of this neural network. To read images, convolutional networks make use of tensors, which are arrays of numbers with various dimensions, thus, images are perceived as volumes, as three-dimensional objects, being measured beyond the logical height x width. This is because digital colours have by default a colour coding: red – green – blue (RGB – Red - Green – Blue), mixing these three colours to create the spectrum of colours perceived by human vision. (Aggarwal, 2019)

As images move through a convolutional network, it describes them in terms of input and output volumes, expressing them mathematically as matrices of multiple dimensions in this way: 30x30x3. From layer to layer, their dimensions change as they traverse the convolutional neural network until they generate a series of probabilities in the output layer, one probability for each possible class of output. The one most likely will be the class defined for the input image, a bird for example (Aggarwal, 2019).

Next, in figure 6, it is clearly demonstrated how the process of each piece of the image is carried out by a neural network, where each small piece is extensively analysed before any result.

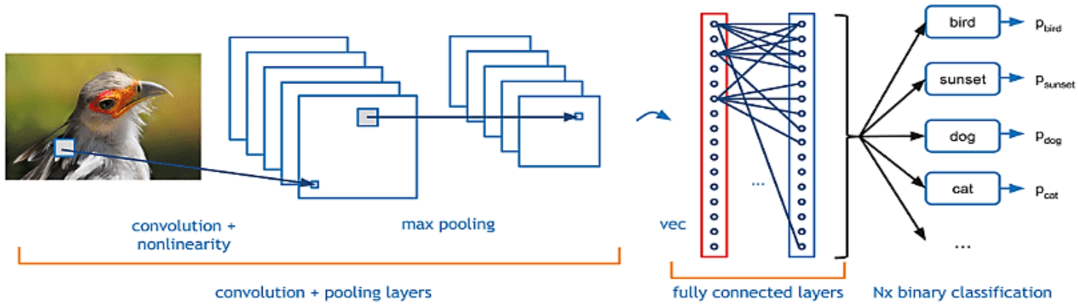


Figure 10 - Image Reading in Deep Learning through Convolutional Neural Networks

Source: (Data Science, 2016).

Within the image processing through convolutional networks, it also finds another important aspect, which are the grouping layers within these networks, which it is name as Pooling. The pooling layers are responsible for simplifying the information in the output of the convolutional layer. Pooling is, in other words, a process of reducing the size of the image, in a simple way, it could say that it is a reduction in the size of the image. The main objective of this operation is to reduce the variance to small changes, and reduce the number of parameters trained by the network (Mueller and Massaron, 2019).

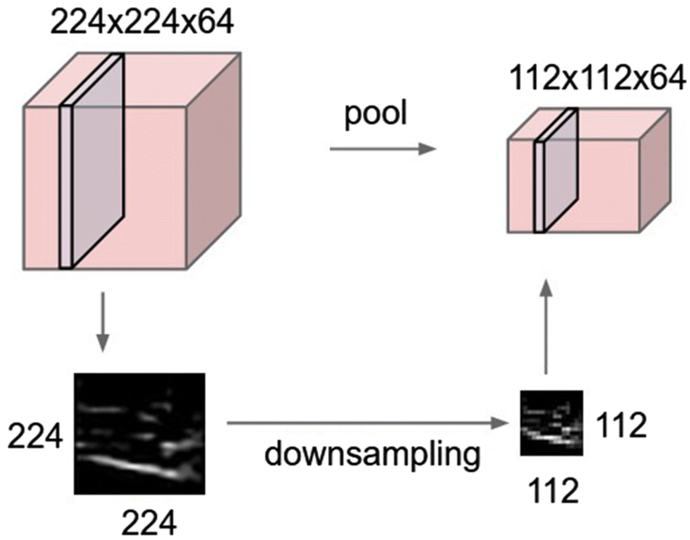


Figure 11 - Image showing an example of downsampling (pooling)

Source: (CLAPPIS, 2019)

Pooling processes can be divided into three ways according to their operations: Max Pooling, Sum Pooling and Average Pooling. All follow the same principle, differing only in the way they calculate the final value. The most used technique today is Max Pooling (Mueller and Massaron, 2019)

Max-Pooling can be seen as a way for the network to ask if a certain resource is found anywhere in a region of the image. It then eliminates exact positional information. The intuition is that once a feature has been found, its exact location is not as important as its approximate location relative to other features. A big benefit is that there are far fewer resources bundled together and therefore this helps to reduce the number of parameters needed in the later layers (Aggarwal, 2019).

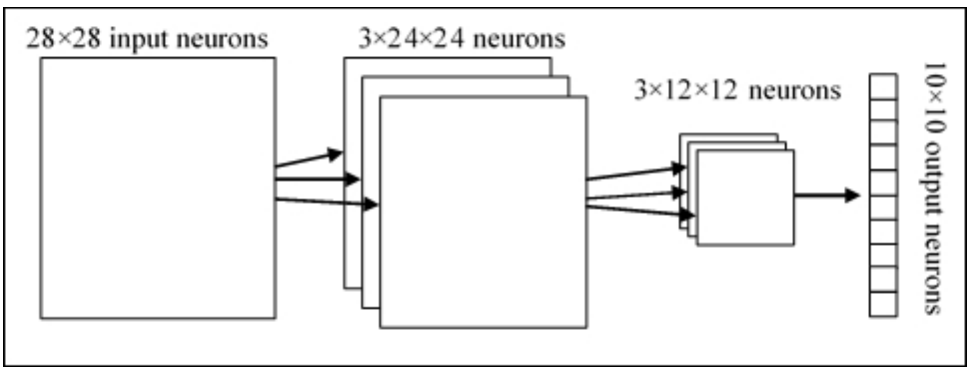


Figure 12 - Image showing a Complete Convolutional Neural Network

Source: (Data Science, 2016)

The network starts with 28 × 28 input neurons (each image of each digit of the MNIST dataset is 28 × 28 pixels), which are used to encode the pixel intensities for an image in the MNIST dataset. This is then followed by a convolutional layer using a 5 x 5 local receptive field and three feature maps. The result is a layer of 3 × 24 × 24 hidden neurons. The next step is a Max-Pooling layer, applied to 2 × 2 regions, in each of the three feature maps. The result is a layer of 3 × 12 × 12 hidden neurons. (Aggarwal, 2019)

Thus, based on this logic, it has access to a network composed of many simple units, where behaviours are determined by their weights and biases, in search of the greatest possible assertiveness in reading images.

3.6. Deep Learning in Imaging Diagnosis

The Artificial Intelligence technology has been improving several areas, specifically, in the medical area, the Deep Learning engineering, presents very satisfactory results, conquering more and more space, as they are faster and more assertive than waiting for the opinion of different specialists. First of all, that in recent years, the popularization of imaging exams has increased in importance, generating a large amount of data to be studied and understood (Paiva Antunes And Prevedello, 2017**)**. For each large amount of data, several professionals are needed for a more conclusive analysis of the exams, and even so, the diagnosis is not always propositional, and errors can occur in the treatment of patients.

The development of artificial neural networks allowed solving some problems of other machine learning techniques. In traditional methods, which are the basis of operation of some diagnostic aid software — for example, computer-aided diagnosis — the increase in the amount of data conditions an improvement in accuracy to a certain extent. With deep learning, a technique that uses hidden layers of artificial neural networks, accuracy tends to continue to increase as new data are added. This allowed, for example, that in 2015, for the first time, in an annual competition for object recognition in everyday images, the machine's performance surpassed the human one. (Paiva Antunes And Prevedello, 2017 p. 02).

However, the technology is still being implemented in the areas of medicine gradually. As an example, it will perform an analysis of a software commonly used in image reading: PAC. From the analysis of this model, it will build Deep Learning interference hypotheses for the improvement of the software process.

PACS (Picture Archiving and Communication System), the idea of this software is to create an environment where all medical images, in digital format, can be accessed from any space in the clinic or hospital. Thus, it can say that a PACS must provide access to a patient's digital images through remote access, using local networks. One of the main characteristics of PACS is that it is one of the most used systems in the medical field, creating a broad and integrated environment in the hospital, focused on internal communication through digital radiological spaces. (Marques Azevedo And Salomão Covas, 2009).

However, for this communication to happen clearly and fulfil its objectives, an organization is needed to distribute information. In this way, a hierarchical structure based on top-down distribution is followed, among other words, the information is propagated from the more general information system (HIS), passing through the intermediate information system (RIS) until reaching the system of more specific information (PACS) (MARQUES AZEVEDO AND SALOMÃO COVAS, 2009 P. 02). For this entire process to be possible, there are two prerequisites to be met, they are: an adequate network structure, which allows for fluid communication. It is a communication standard, the most common in digital radiology being called DICOM (Digital Imaging in Communications in Medicine).

Standardized	Compost
Patient	Computed radiography
Study	Computed tomography
Results	Digitized film image
Storage source	Image subtraction
Image annotations	Magnetic resonance imaging
	Nuclear medicine image
	Ultrasound image
	Graphics
	Curves

Table 1 - Table of Classes of DICOM information objects

Adapted from: (Marque Azevedo e Salomão Covas, 2009)

DICOM has become a global standard in the transfer of radiological images and has gradually been integrated into other areas of knowledge, such as dentistry. To the point of being maintained and updated by an international multidisciplinary committee.

The current DICOM, published in 1993 and commonly identified as 3.0, evolved from earlier versions of a standard developed by the American College of Radiology (ACR) in conjunction with NEMA (ACR-NEMA 1.0, 1985, and ACR -NEMA 2.0, from 1988). The connectivity provided by the standard is very important regarding the cost-benefit ratio for healthcare areas that make use of medical imaging. DICOM users can provide radiology services between facilities located in different geographic regions, take advantage of existing information technology resources, and keep costs low through the compatibility and interoperability of new equipment and systems (Marques Azevedo and Salomão Covas, 2009 p. 2).

To perform the necessary functions, DICOM makes use of existing languages, based mainly on the Reference Model for the Interconnection of Open Systems, also known as the OSI Model (Open Systems Interconnection Reference Model). The model in question is divided into seven layers, ranging from the composition of application interfaces with the user, considered to be the highest layer, to lower layers, such as wires and cables. When communication takes place within the same device, it is called a service. When communication takes place between two different devices, it is considered a protocol. (Marques Azevedo and Salomão Covas, 2009**).**

The following image exemplifies for us, through an imaginary action, how the DICOM structure would react by sending a range of computed tomography images: The scanner encodes all images in a DICOM object. (b) The scanner invokes a series of services to move the object to the physical layer of the OSI model. (c) WS uses a series of services to receive the object through the physical layer and then move it to higher-level layers. (d) WS decodes the DICOM object. (Marques Azevedo and Salomão Covas, 2009 p. 03).

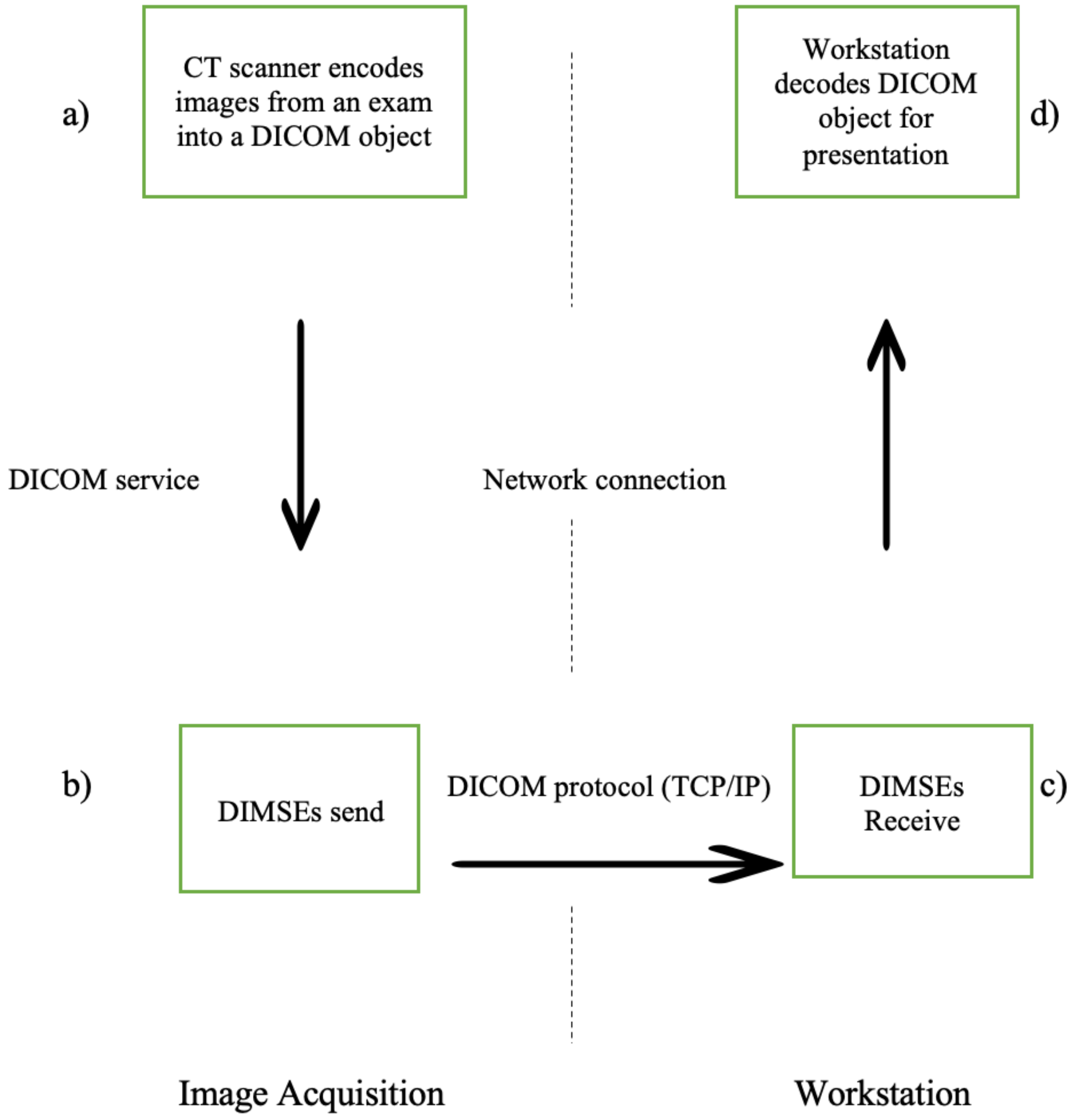


Figure 13 - Example of communication within the DICOM structure

Adapted from (Marques Azevedo e Salomão Covas, 2009)

Doctors and radiographers expect DICOM to be an efficient service when sending, receiving, storing, and retrieving images. However, PACS can also have some limitations, as seen earlier, it is an excellent system for reading images, based mainly on image size. However, some small nodules, for example, cannot be diagnosed via PACS, it is necessary to use specialized instruments to read images using colours and contrasts, in order to diagnose diseases prematurely, preparing the patient for the best treatment and even, intervening before the disease in question needs more aggressive interference.

This concern is extremely necessary, intervention in the software or the creation of new software supported by Artificial Intelligence technology can considerably reduce the diagnosis of false positives, as well as prevent the patient from being exposed to unnecessary radiation and surgeries. On the other hand, with the help of Artificial Intelligence, in particular, Deep Learning, it will be possible to organize image analysis, prioritizing cases from the most serious to the simplest in scales, optimizing productivity and work management by radiology professionals (Sabbatini, 2018).

It can exemplify, more clearly, below, with the case of an Artificial Intelligence software created by Google:

To assess AI's potential to support medical imaging diagnostics, in February 2018, Google announced that one of its subsidiary companies, Verily Life, had created an eye scanning (retinoscopy background) scan program. In it, a specific algorithm can identify retinal data that are used to detect risks of cardiovascular disease and other diseases noticeable through eye changes. According to data released by the organization, the tool registered 70% correct answers in tests of images of real people who could have a heart attack or stroke in five years. (Sabbatini, 2018 p.01).

Another example of the use of Deep Learning implied in the medical field is the detection process of Alzheimer's Disease prematurely. The early diagnosis of AD (Alzheimer's Disease) is essential for its control, while traditional methods are ineffective in making this diagnosis. (Lee, G., Nho, K., Kang, B., Sohn, K.A. and Kim, D., 2019).

Deep Learning, in this sense, can help through the processing of magnetic resonance images, along with physical assessment and memory tests, carried out by professionals relevant to the area.

The Portuguese company NeuroPsyCAD is an example of a pioneering start-up in the area that already carries out tests to detect the early stage of AD and other similar diseases through artificial intelligence. The objective is to gather a database with brain images of patients who have AD or other very similar diseases – such as Mild Cognitive Deficit. The challenge is to train the algorithms to analyse these data and find precisely a pattern that differentiates Alzheimer's Disease from Mild Cognitive Deficit, something very difficult to be distinguished, for example, in consultations and other medical exams. (NEURALMIND, 2020 p. 01).

Although this technology is under development and there is no set date to be implemented in medical services, software based on Artificial Intelligence represents a hope for humanity, for a safer future with more possibilities for cure.

3.7. Pneumonia

Each lung consists of several cavities called alveoli. These cavities are responsible for breathing and gas exchange in a healthy being. Pneumonia is an infectious disease that inflames the alveoli, resulting in pus and other fluids taking possession of the organs. The main symptoms caused by this are breathing pain and decreased amount of oxygen that the affected person can absorb. Such symptoms are lethal in patients with fragile immune systems or in early breathing environments.

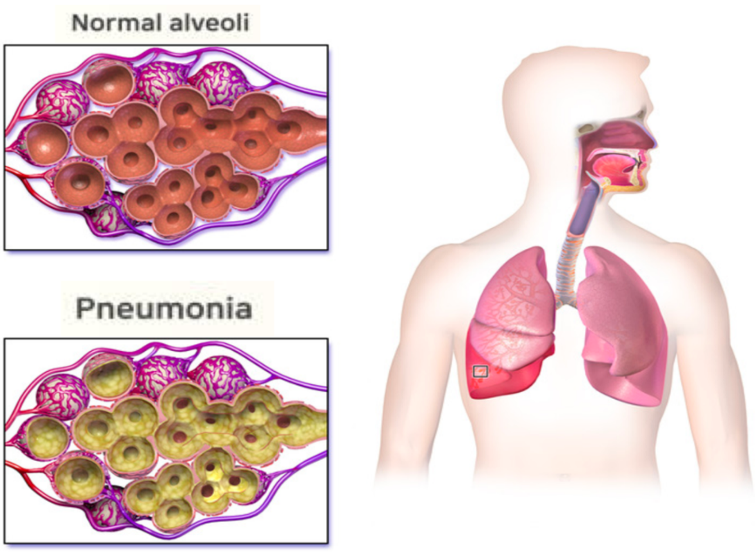


Figure 14 - Structure of normal alveolus and affected by pneumonia in the human body

Source - (DOVEMED, 2016)

The two largest transmitters of pneumonia are viruses and bacteria, and those who have the highest risk of being affected are children under 5 years of age and the elderly over 65 years of age. Pneumonia can be transferred through fluids expelled by a Infected individual (INFO, 2017). The bacterial version of the disease requires accelerated attention because it has to be treated by antibiotics, while the viral version is treated by intensive care (Goldbaum M, 2018), so the differentiation of the two types of the disease is highly beneficial to treatment.

Diagnosing pneumonia is a work done by specialists and usually requires the use of chest X-rays along with the patient's medical history.

3.8. SARS-COV-2 (COVID-19)

Since 2019, the world has gone through one of its most difficult phases: the pandemic caused by the new Coronavirus has claimed victims in all countries, generated health and economic crises, in addition to the collapse of health systems. Coronaviruses belong to a large viral family and have been known for 60 years to cause respiratory infections in humans and animals. It is estimated that the emergence of the new variant of the virus is related to a mutation in the coronavirus that infects bats, breaking the genetic barrier to adapt to a new species. The initial broadcast location was a seafood and live animal market in Wuhan City, China. (Medeiros, 2020). Among the most common symptoms of the virus, it can be listed: fever, dry cough, headache, and especially shortness of breath. These symptoms suggest an acute respiratory infection or radiologically detected pulmonary abnormalities.

Individuals seriously ill with COVID-19 have different pulmonary presentations. Gattinoni et al (2020) divide the pulmonary phenotypes into type L (low) and type H (high). Type L has low elasticity (normal compliance), low ventilation and perfusion ratio (VA/Q), low lung weight (even with ground-glass pattern, there is no major change in lung weight) and low alveolar recruitability (the alveoli are inflated, so recruitment manoeuvres have low impact). Type H, in turn, presents high elasticity (increased oedema that leads to decreased air volume), high right left shunt, high lung weight and high alveolar recruitability (increase in non-aerated tissues that can be effectively recruited) (Martinez et al. apud Nielsen and Silva, p 05, 2020).

Therefore, the force with which COVID-19 acts on the airways and lungs is indisputable, although to a greater or lesser degree, it is possible to perceive its effects on the organism of individuals affected by the virus. Below, it will be contemplated images of three lungs, one healthy, one diagnosed with SARS-COV-2 and the other with viral pneumonia, not from the COVID-19 virus. It will draw a parallel between these three examples, as a proof of concept will be supported by the classification of these diagnoses.

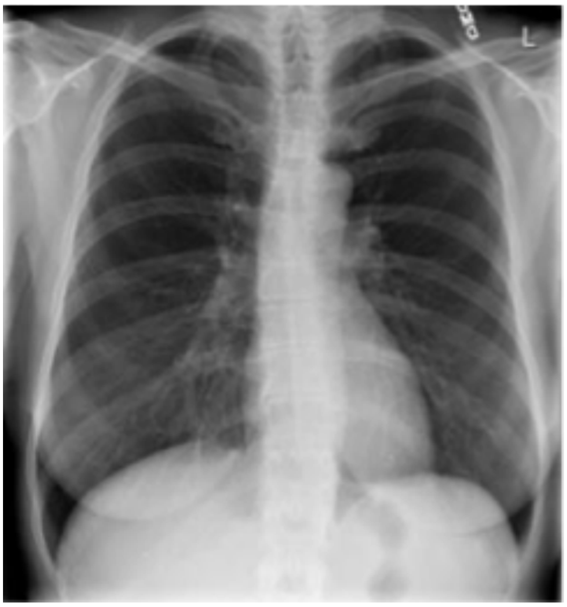


Figure 15 - Radiograph of a healthy lung

Source: (KAGGLE, 2021)

Figure 16 - Radiograph of a lung diagnosed with SARS-COV-2

Source: (Kaggle, 2021)

Figure 17 - Radiograph of a lung diagnosed with viral pneumonia, not from COVID-19

Source: (Kaggle, 2021)

In general, it is noticed that the radiographs with compromised lungs due to some comorbidity, present a white spot in its extension. This happens because the lung acquires more density due to infiltrations.

When there is an increase in pulmonary pressure, the margins of the vessels begin to have ill-defined limits due to the extravasation of fluid into the interstitium, the vessels of the apexes become wider, and the circulation is visible to the periphery. There is also a diffuse increase in density in hilar regions. The interlobular septa can thicken with the accumulation of fluid and short lines appear, horizontal and perpendicular to the pleura, the "Kerley lines", which indicate interstitial oedema. When there is alveolar oedema, the pulmonary vessels may no longer be seen, because the lung acquires fluid density, which is the same density as the vessels. (Bohrer Brentano, Schnorr et al. 2015, p 03).

This characteristic is very common in patients with SARS-COV-2, although the shape of this density is different, the specific characteristics of COVID 19 also change throughout the disease. The proof of concept will use this change in format and other characteristics characteristic of comorbidity to properly read the radiographs.

3.9. Conclusions

Along this study, Artificial Intelligence, especially Deep Learning, has a lot to add in the medical field, as well as influencing other different areas of human life. Although the studies are recent, they have represented a revolution in the health and life expectancy of patients, as well as the optimization of medical work.

4. Software Methodology

4.1 Waterfall Methodology

Waterfall Model is a software development and planning methodology, in which the entire project is performed like a waterfall, where the analysis, design, implementation, testing, integration and maintenance phases occur in a flow, in sequence. As it is a fluid planning, where one step leads to another in an integral way, the name Waterfall was used to define this methodology, as it is believed that the first to use this term was W.W. Royce, in an article published in the 1970s.

There is not enough information to determine who is the creator of the Waterfall methodology. However, this method has been used since World War II and gained strength in the market from the 1970s onwards, becoming the traditional methodology for planning and developing software.

W. W. Royce was the first author to speak of this method, which helped to spread the idea throughout the technological field. The author, however, was against this methodology. He criticized its efficiency by claiming that the interactive method would be a better option. Royce is responsible for the nomenclature given to the cascade method. The method outline, created by the author, has become popular among developers and is currently the basis for those who wish to use the waterfall methodology in their projects.

Waterfall methodology is a fluid planning, in which each step triggers the other. It's like water running through a waterfall. Therefore, this method does not allow errors or alterations. If the next steps do not develop properly, the entire project will have to be redone. This is one of the methodologies in which planning is essential, as it will be the basis for all software development.

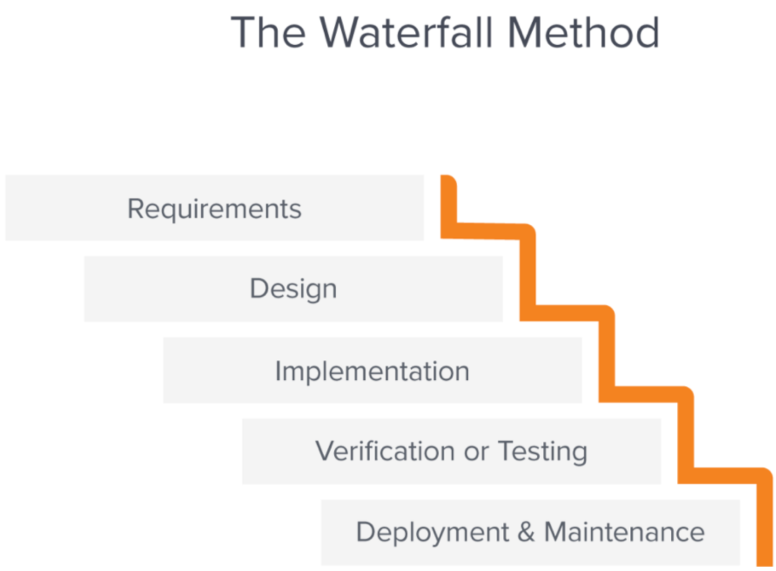


Figure 18 - Image representing the Waterfall Method

Source: (Pressman, R, 2016)

The cascade model proposes dividing each stage of software development into phases and completing each of them in a specific order, that is, you cannot start “phase 2” until you have completed “phase 1”.

Another feature of the cascade model is that you must do an analysis and check the operation of each of the phases at the end of them, before moving on to the next, thus detecting possible errors and correcting them before proceeding. This process allows you to relate each of the stages of the cascade model with the previous one, in this way, consider the elements that you must remove or add in the next phase.

Like all software methodologies there are advantages and disadvantages some advantages of this model are. It's very easy to implement and doesn't require a lot of resources to do so.

Allows departmentalization and specialization as each one has a specific role throughout the project. It facilitates management control as it produces an artefact at the end of each phase. Both the customer and the project manager are able to track progress with high visibility. Linear work also makes it possible to compare the end of each phase with the objective established at the beginning of the project. That way it's easy to know if we're “on schedule” for the project. The development team starts working (coding) much faster (or earlier) than in other models because they get what must be done defined from the previous phase. It reduces the time the

customer needs to devote to phases 2 and 3. As the model focuses on the production of documentation, the interaction with the client decreases as everything is documented in the artefact produced by phase 1.
The disadvantages are It does not allow for changes in requirements and, when it occurs, its costs and deadlines are very high. That's because the changes will occur in the Verification phase, which is where the customer sees their requirements translated into product. The actual product doesn't always make as much sense as the product that was written in the document. Faced with a change, it is necessary to go back to all the phases and redo them: specify, design, implement and test again.
As this model is strictly linear, interactions are not predicted and are treated “outside the model”. The confusion this can bring as the project progresses. Who has never heard the phrase “It's not in the requirements document but I asked for this.”
It brings uncertainty to both the technical team and the customer. As this model does not provide feedback, it is difficult to know whether, in addition to being on time, the actual software is in line with the customer's idea.
My thesis will have evolving needs as research and feedback will take place for the duration of the project – Waterfall Methodology will not be suitable for the project.

4.2 Agile Methodology

Agile methodology is a set of techniques and practices for project management that offers more speed, efficiency, and flexibility. Its initial objective was to streamline software development, but these methods have gone beyond the technology sector and today revolutionize the management of companies in all areas. Basically, the agile methodology makes processes simpler, more dynamic, and iterative, from idea conception to the final product. Among its main features, the following stand out:

- Incremental development, that is, continuous improvement
- Cooperation between team and customer (constant feedback loop)
- Fast and high-quality deliveries
- Project scope flexibility
- Progressive value creation according to customer needs
- Adaptability to change and high level of innovation.

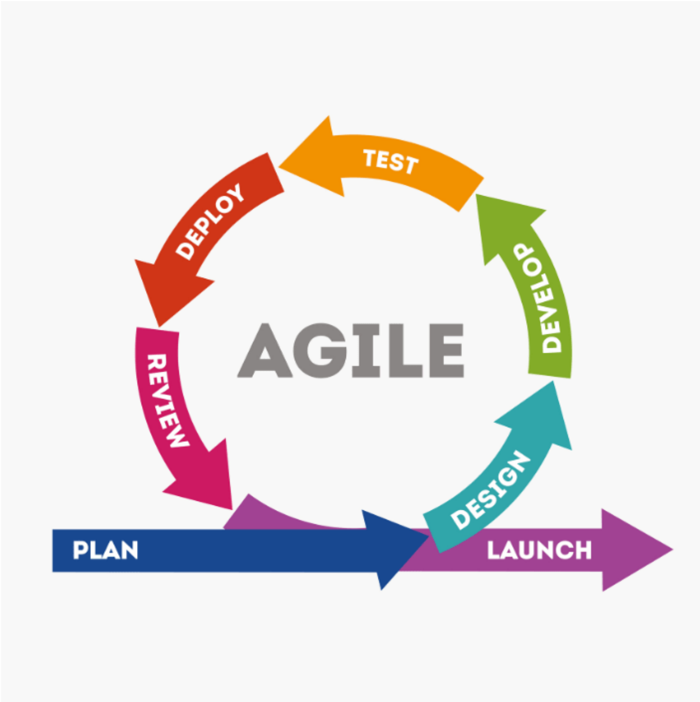


Figure 19 - Image representing the Agile Method

Source: (Barizon, Cláudio, 2020)

Agile methodologies differ from traditional ones in several ways, but one of the main ones is the speed of the process.

It is known as a traditional methodology, or waterfall methodology, is that predefined sequence of steps, such as requirements analysis, development, testing, production, and maintenance.

In this approach, projects are quite time-consuming, and the principle is to predict what the results will be in the final delivery. In the agile methodology, the focus is on adapting rather than planning. Therefore, projects are divided into small deliverables called iterations. Each iteration is a “mini-project”, that is, it includes all the steps mentioned in a fast and efficient cycle, which generates a partial delivery. Thus, the customer can see results quickly and give feedback throughout the entire process – hence the incremental characteristic of the method. As the cycles repeat, the product is continuously improved on an experimental basis and can be tested for each functionality. Thus, agile methods allow teams to deliver more value in less time. As a result, the final version is much completer and more assertive than the traditional methodology. Another fundamental difference is that, in the agile methodology, projects are always open to change, no matter how impactful they may be. On the other hand, in the traditional method, it tries to minimize changes so as not to compromise the planning. Finally, the participation of customers and employees in the process is another differential of the agile culture. That's because piecemeal deliveries allow everyone to assess the progress of the product or service, evolving the creation together. The agile methodology was created by a group of developers who got tired of the traditional methods and their plastered steps. Then, in 2011, they came together to craft the Agile Manifest, a statement that contains the principles of agile software development. The important thing is to deliver a quality product and create an excellent user experience, even if this requires bypassing some processes and changing plans several times. I liked the aspect of being adaptable as throughout the development phase certain factors can change the project and being able to adapt to those quickly is important. But I decided to use the V model due this is a small project and combine the Waterfall model and some aspects of agile model, so I think it will be more suitable for my project.

4.3 V Model

The V model is a conceptual model used in project management that in 1980, with the emergence of Software Engineering, became a standard concept in all domains of the software industry. The V-Model is a variation of the waterfall model, which demonstrates how testing activities are related to analysis and design. This model proposes that unit and integration testing can also be used to verify the design. That is, during unit and integration testing, programmers and the testing team must ensure that all aspects of the project have been correctly implemented in the code.

The connection between the left and right sides of the V model implies that if problems are encountered during verification and validation, the left side of the V can be rerun to correct and improve requirements, design and coding, before the execution of the test steps that are on the right side. In other words, model V makes some iterations and repetitions of the work more explicit, hidden in the waterfall model. While the waterfall model's focus is on documents and artefacts, V's focus is on activity and correction.

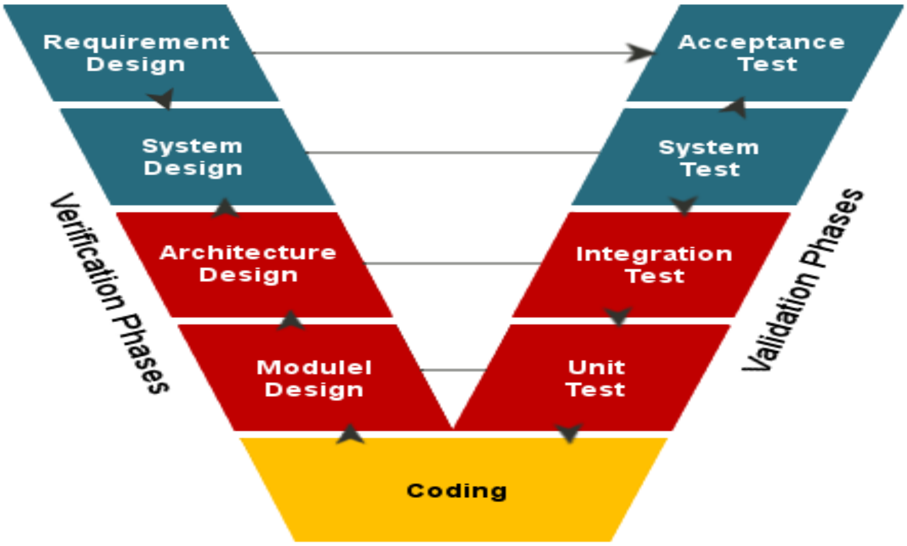


Figure 20 - Image representing the V Model Method

Source: (Barizon, Cláudio, 2020).

The V model is an SDLC model where the execution of processes occurs sequentially in a V format. It is also known as the verification and validation model. The V-Model can be considered an evolution of the Waterfall methodology. There are many similarities between the two models. The V-model takes a very rigid approach, with the next phase only starting when the previous phase is complete. The requirements must be very clear before the project begins because it is often expensive to go back and make changes. This model is used in the field of medical development, as it is a strictly disciplined domain.

Key features of this model are that it emphasizes the importance of considering testing activities during the process, rather than later testing after the process is finished, can get feedback faster, helps develop new requirements and improves quality of the resulting product. Some disadvantages of the V-model are the difficulty in following the sequential flow of the model and the difficulty for the customer to be able to specify the requirements explicitly.

5. Methodology

In this work, the objective is to research and apply techniques related to Artificial Intelligence. The research is classified as qualitative and focused on action research, as, in addition to studying and reviewing what has been produced in the area so far, a proof of concept that interacts and interferes will also be developed. in the environment that will be inserted, making us what Thiollent describes as agents of action.

It is a method, or a research strategy, aggregating several research methods or techniques, with which a collective, participatory, and active structure is established in terms of capturing information. “The methodology of the sciences considers action research like any other method. This means that it takes it as an object to analyse its qualities, potentials, limitations, and distortions. The methodology offers general knowledge subsidies to guide the design of action research and control its use“. (Thiollent, 1986, p. 25-26).

Initially, it seeks to review diagnostic imaging materials, understanding what it is about and how it is currently performed. Subsequently, it is proposing an in-depth study of Deep Learning, a sub-area of Artificial Intelligence, to understand its structures, the way it is applied and how it can work with this technology. Then, it is suggesting the interference of Deep Learning in the reading of medical images, aiming at the possibility of interference of Artificial Intelligence in the medical environment.

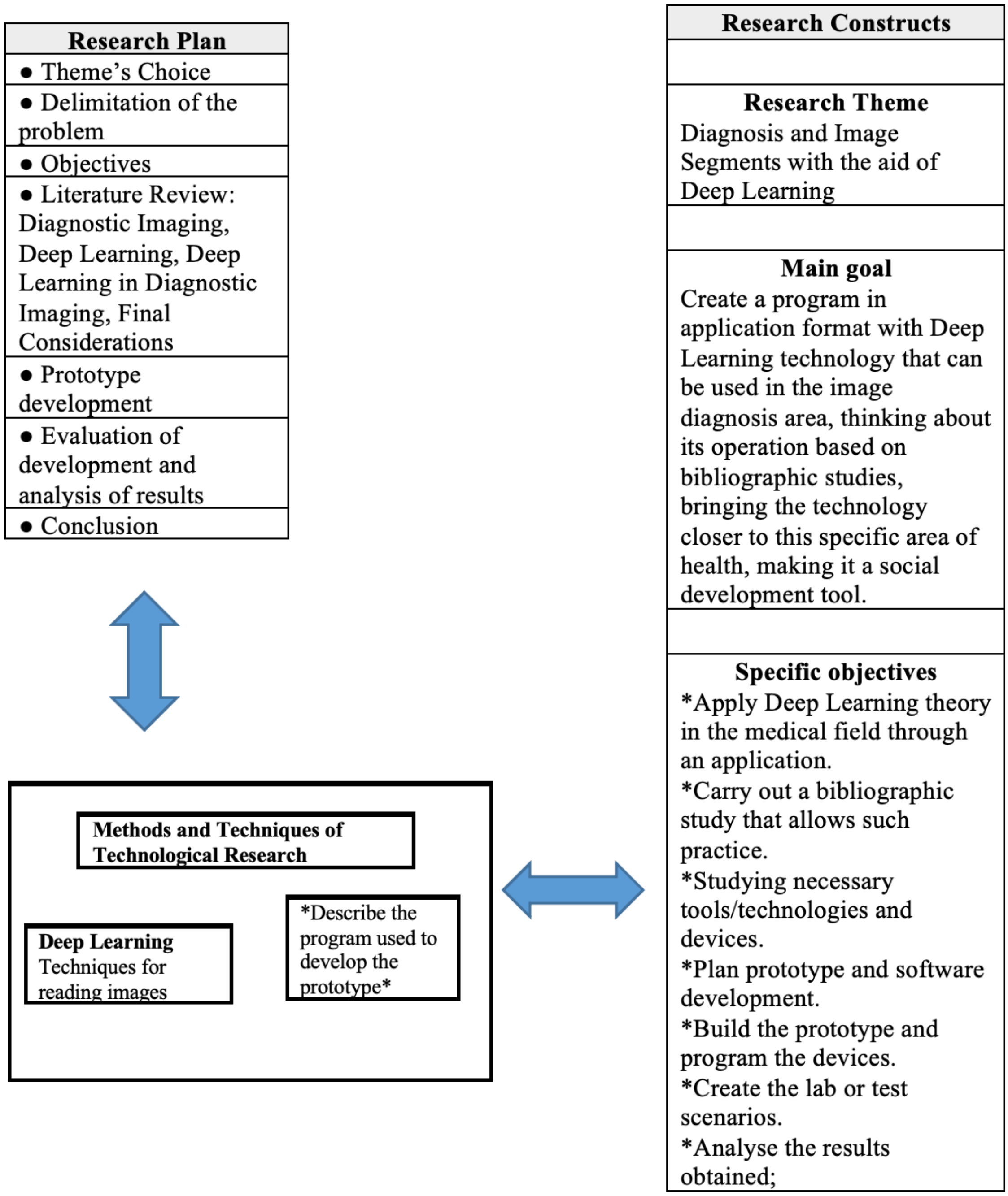
5.1. Research Characterization

In addition to theoretical production, in this work it has the ambition to act on the problem raised, influencing, through the creation of an Artificial Intelligence software, focusing on Deep Learning, in the segmentation of medical images. The aim is to rethink the quality of life of patients, aiming at technologies that can provide medical assistance, anticipating the occurrence and possibility of future diseases.

In this context, according to Silva “our research refers to applied research, which aims to generate knowledge for practical application and aimed at solving specific problems. It involves local truths and interests”.

Taking into account the objectives in this monograph, it can also be classified as exploratory, as through real case studies and biographical data collection, it seeks to provide an understanding of the subject, in a simple way. (Silva, 2005).

5.2. Methodological Activities



5.3. Study Object

The object of study of this thesis is to study specifically how Deep Learning can help physicians and radiology professionals to more easily detect possible diseases that are diagnosed by imaging.

5.4. Delimitation

This work was limited to the diagnostic analysis and image segment of only one software in the area of pattern recognition currently and freely distributed.

6. Solution Proposal

Throughout this chapter, the software development process, the main difficulties, language, and structure of the code, as well as the image bank and analysis of results obtained with the developed project were described. The main objective is to show how a software based on Deep Learning classifies an image and predicts comorbidities, specifically SARS-Cov-2 and Pneumonia, differentiating a healthy lung from a diseased lung.
It is also important to note that it does not focus on developing software that could be used by physicians or health professionals, the focus of this project is to demonstrate how Deep Learning technology can interfere in the Image Diagnosis process through a proof of concept (POC).

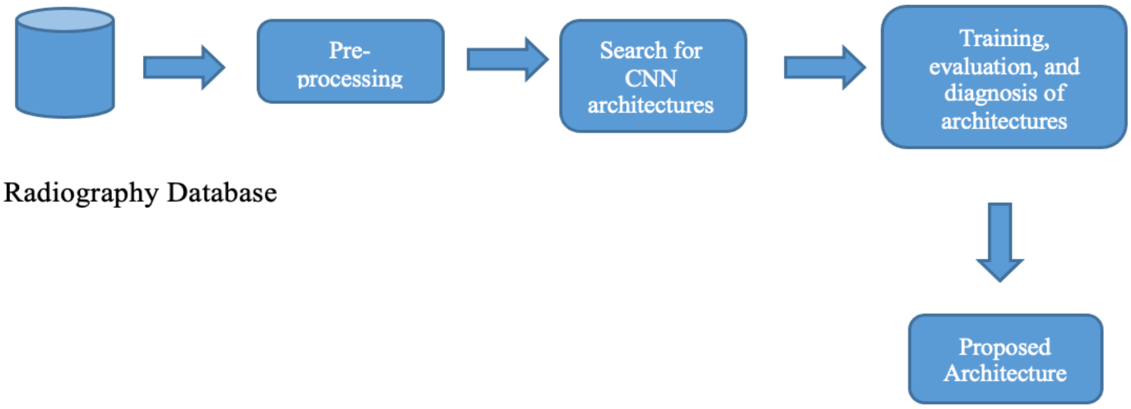


Figure 21 - Image of the Proposed methodology used in this project

Source - Author's Collection

6.1. Requirements

When this project began to be conceived, it was based on the principle of what would be the future in Data Science, in which way the concepts that were worked on daily within academic life transform reality to the point of becoming a part of human life. Clearly, this already happens at some level, through smartphones, tablets, and personal computers, however, the intention is to act directly together with other knowledge, showing that Data Science linked to other areas, such as health, acts directly on quality of individuals' lives.

Based on this information, a research work was started, to understand a little more about the emerging areas, among others, the Deep Learning technology was the one that showed the most promise, being used in several segments of areas in partnership with Science of Data.

In the 1950s, researchers wanted to build computers that had the same characteristics as the human brain (machines with artificial intelligence). Today artificial intelligence techniques that can help with various decision making. The AI market helps millions of companies use data to convert goals into action. IDC global released a recent study that says the artificial intelligence market will grow 46.2% annually (CAGR) to reach $52 billion in 2021. Spending on artificial intelligence will triple by 2022, so it has a forecast that the figure will reach $77.6 billion four years from now, three times more than the $24 billion in 2018. “Machine learning and deep learning will be trends that are expected to grow fastest in the growing technology category over the course of the year. of prediction, representing about 40% of all cognitive and artificial intelligence expenditures with a five-year CAGR of 43.1%”. (Montini, 2019 p. 01).

Like all software or proof of concept, it also developed a list of basic functional requirements for the development of this dissertation. Functional requirement stands out for being a set of capabilities that need to be implemented for the proof of concept to fully exercise its functionality. It indicated how the requirements works:

- Create neural network layers.
- Classify radiographic images according to preoperative comorbidities established.
- Generate graphs of the accuracy of training and validation data.
- Generate data on training loss and validation.
- Generate confusion matrix graph.
- Predict comorbidity from selected images.

6.1.1 Development of the Proof of Concept

Having defined the objectives of the proof of concept and having the project idealized, it sets out in search of diagnostic images of the comorbidities to be classified through Deep Learning. For the selection of these images, it was used the Kaggle library, which provides data scientists with a set of datasets. In this way, it was possible to find all the images analysed in the proof of concept. It was then selected the Visual Studio Code Integrated Development Environment (IDE) to perform the compilation and provide tools to support the inscription of the proof-of-concept code.

After these first two selections, it was necessary to find a code platform for Machine Learning, where it was selected TensorFlow. This platform, which was responsible for performing all the complex calculations needed in the proof of concept. To help build the code, the Keras library was used, which is a more sophisticated platform, which uses the complex calculations of TensorFlow, to quickly build neural networks, in an assertive way, reducing this long and complex mathematical process of TensorFlow. The figure below demonstrates the structure that was used to carry out the proof of concept.

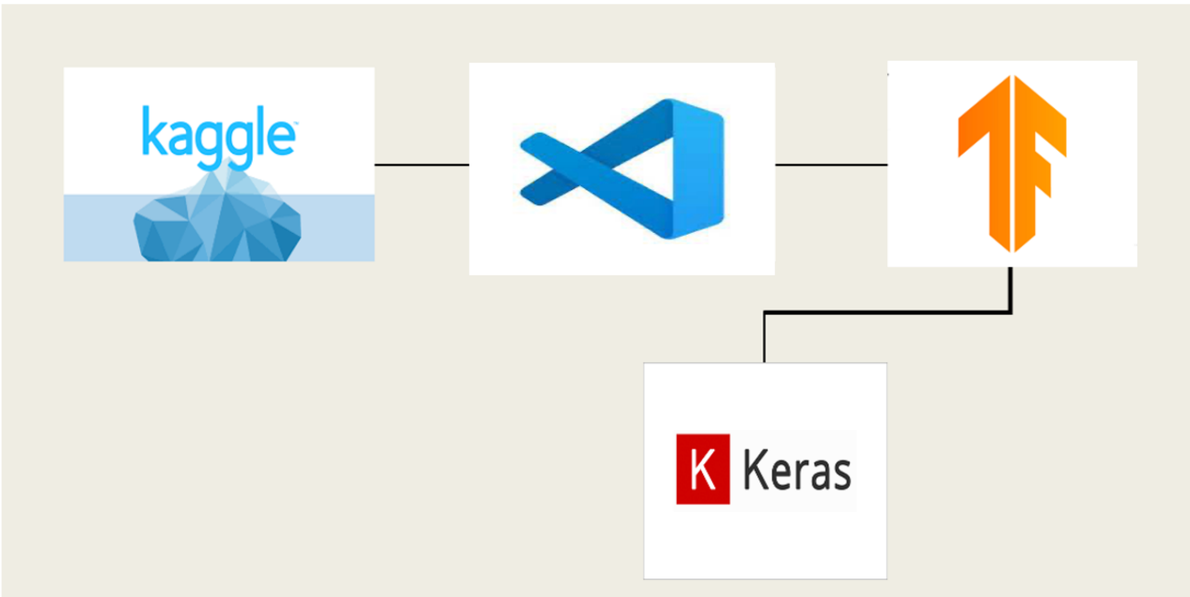


Figure 22 - Image of the Structure of the Proof of Concept with its technologies

6.2. Technologies Researched

Visual Studio Code

During the project development, Visual Studio Code tool will be use, an IDE (Integrated Development Environment), that facilitates the software development process. Among the features present in it that can be highlighted Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs.

Figure 23 - Image of Visual Studio Code logo

Source: (visual studio, 2021)

TENSORFLOW

TensorFlow, in turn, is a specific Google library for Deep Learning and Machine Learning projects. Its main function is to simplify complex calculations by graphing them. This library was chosen because it presented itself as the tool with the greatest documentation in its base and is easily accessible for use, has the capacity to interfere in various platforms, and is flexible enough to experiment with research aimed at Deep Learning.

Another important factor to be highlighted is that TensorFlow allows the easy creation of models, allowing the creation of clean code, which, according to Martin, is what allows us to develop in the long term, in other words, it is what allows the software remains and evolves from the moment that, in future projects, the code is clearly understood and can continue the development process. “In short, a programmer who writes clean code is an artist who can take a blank canvas and subject it to a series of transformations until it becomes a beautifully programmed system”. (Martin, 2011 p 07).

Figure 24 - Image of TensorFlow logo

Source: (TensorFlow, 2021)

KERAS

The Keras library, also has extensive documentation and developer guides, offers consistent and simple API's, minimizing the number of user actions. The main advantage of using this library is the speed in the execution of new experiments, allowing greater use of time during software programming.

Figure 25 - Image of Keras logo

Source:(Keras.io,2021)

KAGGLE

Kaggle is a learning and competition platform for data scientists. Kaggle allows users to publish, find datasets, build multiple models and work with data scientists.

Figure 26 - Image of Kaggle logo

Source: (Kaggle, 2021)

From this platform, it is possible to access a database that has a huge set of X-ray images of normal lungs and those with diseases such as: Covid-19 and Viral Pneumonia. This image database was COVID-19 dataset champion awarded by the Kaggle community, according to the platform users this image database was the best among the other datasets available on the platform in terms of image quality and availability. From this dataset on, the proof of concept will be carried out.

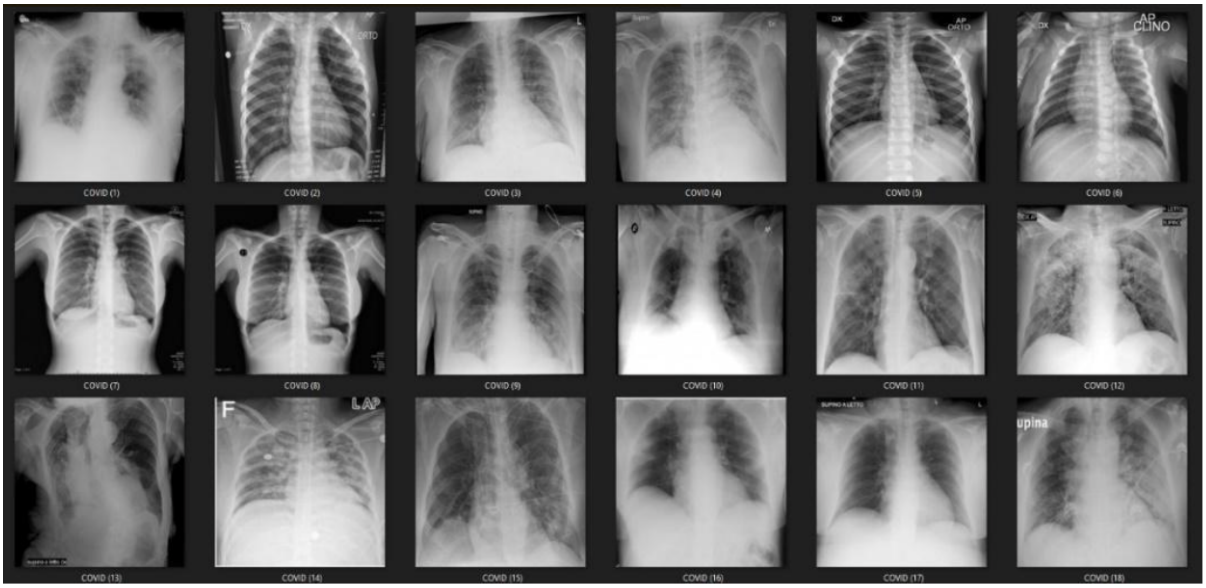


Figure 27 - Lung base images with Covid-19

Source: (Kaggle, 2022)

7. Testing

7.1. Introduction

The test aims to verify that the software is in accordance with what was planned, as to show that my program does what it intends to do and to discover the bugs or problems before releasing it.

Software testing, in theory, is a straightforward activity. Each input has a defined and known output. After the values are entered, a certain result is expected. If that happens, great. If not, there may be a bug and this is the difference for machine learning, where varied results are expected.

Machine learning, simply put, are prediction-based applications using ready-made models. These applications are built on top of learning algorithms that change according to data and interactions. From a testing point of view, the expected values after a predetermined battery will be different from the next results, as the machine will have learned from the first test.
Traditional testing techniques are done with fixed values. The common assumption in testing is that the value of input X will always result in output Y. With machine learning the output is not fixed, it will vary over time as the model has more data.

7.2. Plan for Testing

As said before the V model will be used for developing and test my program, V model is concerned with carrying out tests throughout the development cycle to find errors as quickly as possible. It means Defects can be found at an early stage because testing starts from requirement phase itself.
After finishing the Verification Phase (the requirements phase), the codification part of the dissertation will start and after that it is going to move to the Validation Phase. The tests that will be carried out are:

Unit testing is conducted at an early stage Here each unit of the system is tested individually.

For example: Here is where test are made to verify that the program recognize my databases.

Next text will be:

Integration test In this phase basically the units of the system are tested in a combined way, it tests the interaction between the integrated units.

System testing This test verifies that the components are compatible, that they interact correctly, that the right data is transferred at the right time.

Acceptance test This test ensures that the system operates as expected.

7.3. Conclusions

This chapter reviewed the testing. The testing consisted of performing V model test while developing the system. Well-planned and automated tests allow, quickly and accurately, detecting software non-conformities, starting corrections as soon as possible.

8. Project Development

8.1 Acquisition

The first step consisted in the acquisition of data, which were obtained through the previously mentioned Kaggle database. During the development of the tool, lung x-ray images were used in the experiments, as it is one of the modalities to detect COVID-19.
As a database, it were used the COVID-19 Radiography Database from Kaggle https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database , which is practically balanced, with 15.153 instances, being 3.616 referring to the lung with COVID-19, 1.345 referring to the lung with Pneumonia and 10.192 referring to the normal lungs of uninfected patients, however, who have other diseases that affect the lungs, as can be seen in the table, from this 1500 images will be used to test.

Lung images	Quantity
COVID-19	3.616
Pneumonia	1.345
Normal	10.192

Table 2 - Quantity of images and classification

8.2 Networks used in training

This section will look at the architecture of the four networks used in the dissertation. It starts with the pretrained networks VGG-16, and Xception, after the architecture of my network trained from scratch. These four solutions will be the tools to address the problem proposed in this dissertation and to choose the best based on the results obtained with each of them.

8.2.1 Pretrained network

A pretrained network is a network that has been previously trained and saved, so it can effectively act as a generic model in multiple problems. In this case, it is going to use a large convnet trained on the ImageNet dataset. The chosen pretrained network are the VGG16, and Xception architectures.

8.2.1.1 VGG16

As it can be seen in the Figure below, the VGG16 network is formed by the succession of the following convolutional layers with relu activation, max pooling, dense and softmax classifier in the last classifier layer. Its convolutional basis to which it will add the classifier would end up in the last layer coloured red. This image does not match with the model used in this dissertation since the input images are from another dimension.

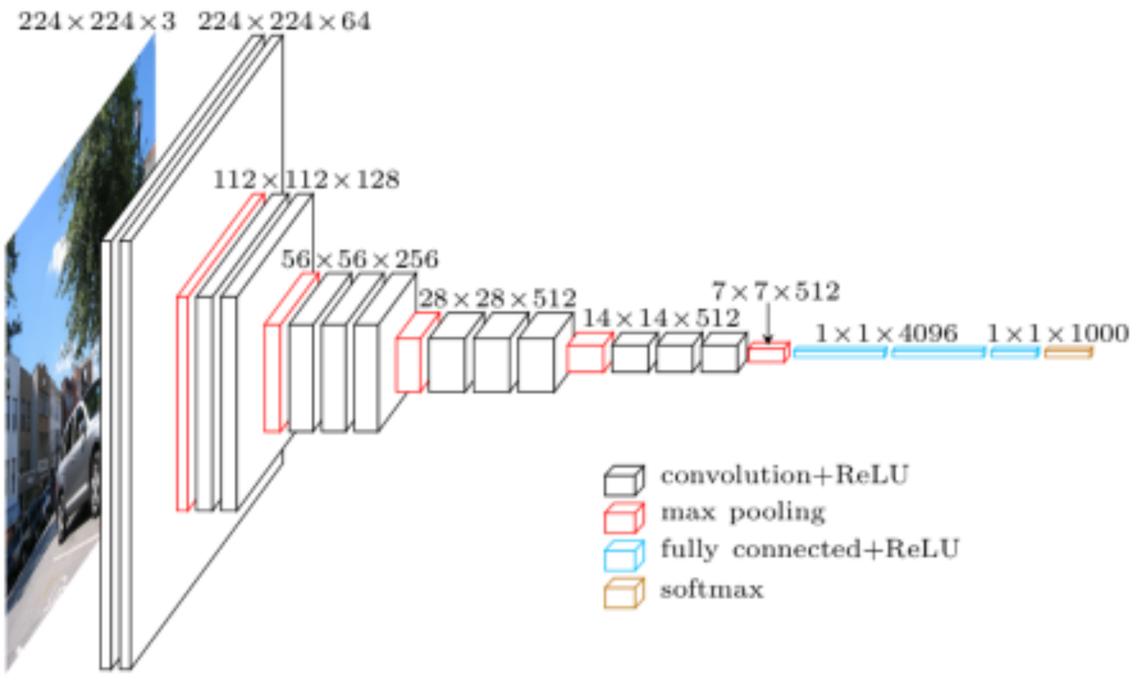


Figure 28 - Architecture of the VGG16 pretrained network.

Source:(towardsdatascience.com)

Tracking how the dimensions of the feature maps change in succession of the VGG16 layers:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_13 (InputLayer)       [(None, 200, 200, 3)]     0

 block1_conv1 (Conv2D)       (None, 200, 200, 64)      1792

 block1_conv2 (Conv2D)       (None, 200, 200, 64)      36928

 block1_pool (MaxPooling2D)  (None, 100, 100, 64)      0

 block2_conv1 (Conv2D)       (None, 100, 100, 128)     73856

 block2_conv2 (Conv2D)       (None, 100, 100, 128)     147584

 block2_pool (MaxPooling2D)  (None, 50, 50, 128)       0

 block3_conv1 (Conv2D)       (None, 50, 50, 256)       295168

 block3_conv2 (Conv2D)       (None, 50, 50, 256)       590080

 block3_conv3 (Conv2D)       (None, 50, 50, 256)       590080

 block3_pool (MaxPooling2D)  (None, 25, 25, 256)       0
show more (open the raw output data in a text editor) ...

=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
```

Figure 29 - Characteristics of VGG16.

As it can be seen (Figure above) that the output feature map has the form (2,3,512) It is on top of this feature that it will put the densely connected classifier. There would be several ways to do this, but it has settled on adding Dense layers on top of it and running everything from start to finish on the input data.

This choice will allow the application of the data augmentation technique, although it is a computationally expensive and slower process.

Code used to create the VGG-16 model:

```
tfVGG16_model = keras.Sequential([
    VGG16_model,
    layers.Flatten(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.15),
    layers.Dense(3, activation='softmax'),


])
tfVGG16_model.summary()
```

Figure 30 - Explanatory code. Creation of the VGG-16 model.

Whose sequence of feature maps will be as follows.

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 6, 6, 512)	14714688
flatten_26 (Flatten)	(None, 18432)	0
dense_52 (Dense)	(None, 256)	4718848
dropout_40 (Dropout)	(None, 256)	0
dense_53 (Dense)	(None, 3)	771

Figure 31 - Explanatory code. Characteristics of the VGG-16 model

Before carrying out the model learning process, it is important to freeze the convolutional base to prevent its weights from being updated during training and to prevent the representations previously learned by this network from being modified. This is possible with the following line of code.

```
for i in VGG16_model.layers:
    layers.trainable = False
```

Figure 32 - Explanatory code. Freezing of the convolutional basis of VGG16.

Next, the model is compiled, apply the Data Augmentation technique and execute the fit() method.

```
history_VGG16 = tfVGG16_model.fit(train_iter, epochs = 20,batch_size = 16,callbacks=[early_stopping],
                                  validation_data=test_iter)
```

Figure 33 - Explanatory code. Execution of VGG16 model.

8.2.1.2 Xception

Xception is an extension of the Inception architecture that replaces the Inception modules by point convolutions followed by depth convolutions, as shown in the figure below.

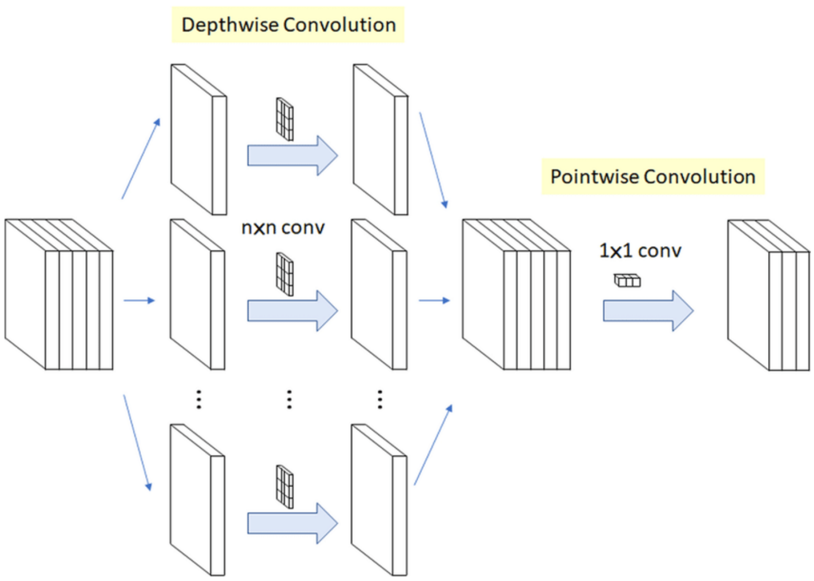


Figure 34 - Architecture of the Xception pretrained network.

Source:(researchgate.net)

The Xception module has 3 main parts. The Entry flow, the Middle flow, and the Exit flow. Tracking how the dimensions of the feature maps change in succession of the Xception layers:

```
Model: "xception"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_35 (InputLayer)          [(None, 200, 200, 3  0           []
                                )]

 block1_conv1 (Conv2D)          (None, 99, 99, 32)   864         ['input_35[0][0]']

 block1_conv1_bn (BatchNormaliz  (None, 99, 99, 32)  128         ['block1_conv1[0][0]']
 ation)

 block1_conv1_act (Activation)  (None, 99, 99, 32)   0           ['block1_conv1_bn[0][0]']

 block1_conv2 (Conv2D)          (None, 97, 97, 64)   18432       ['block1_conv1_act[0][0]']

 block1_conv2_bn (BatchNormaliz  (None, 97, 97, 64)  256         ['block1_conv2[0][0]']
 ation)

 block1_conv2_act (Activation)  (None, 97, 97, 64)   0           ['block1_conv2_bn[0][0]']

 block2_sepconv1 (SeparableConv  (None, 97, 97, 128)  8768       ['block1_conv2_act[0][0]']
 2D)

 block2_sepconv1_bn (BatchNorma  (None, 97, 97, 128)  512        ['block2_sepconv1[0][0]']
show more (open the raw output data in a text editor) ...
==================================================================================================
Total params: 20,861,480
Trainable params: 20,806,952
Non-trainable params: 54,528
```

Figure 35 - Characteristics of Xception

Code used to create the Xception model:

```
tfXception_model = keras.Sequential([
    xception_model,
    layers.Flatten(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.15),
    layers.Dense(3, activation='softmax'),
])
tfXception_model.summary()
```

Figure 36 - Explanatory code. Creation of the Xception model.

Whose sequence of feature maps will be as follows.

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 7, 7, 2048)	20861480
flatten_45 (Flatten)	(None, 100352)	0
dense_90 (Dense)	(None, 256)	25690368
dropout_69 (Dropout)	(None, 256)	0
dense_91 (Dense)	(None, 3)	771

```
Total params: 46,552,619
Trainable params: 46,498,091
Non-trainable params: 54,528
```

Figure 37 - Explanatory code. Characteristics of the Xception model.

Like the previous models the model has the convolutional base freeze to prevent its weights from being updated during training and to prevent the representations previously learned by this network from being modified, compile the model, apply the Data Augmentation technique and execute the fit() method.

8.2.1.3 Model Created from Scratch

To define the layers of the model, it was used Keras's Sequential class, which groups a linear stack of layers within a model. These layers are responsible for processing the input data, in the case of the prototype performed, these data are the radiographic images. During the development of the prototype, 14 layers were used to perform the image processing, each layer has its specific function and will impact the neural network in different ways. Some layers were used, among them some classes such as: Batch Normalization, Conv2D, MaxPooling2D, Dropout, Flatten and Dense. It will be seen in this chapter how these layers work within image processing. The following image shows the definition of the layers within the prototype.

```
#Now I create my own model
final_model = keras.Sequential([
        layers.Input(shape = train_iter.image_shape),
        layers.BatchNormalization(),
        layers.Conv2D(32, 3, activation='relu'),
        layers.MaxPooling2D(),
        layers.Conv2D(64, 3, activation='relu'),
        layers.MaxPooling2D(),
        layers.Dropout(0.3),
        layers.Conv2D(128, 3, activation='relu'),
        layers.MaxPooling2D(),
        layers.Dropout(0.2),
        layers.Flatten(),
        layers.Dense(256, activation='relu'),
        layers.Dropout(0.15),
        layers.Dense(3, activation='softmax'),


])
final_model.summary()
```

Figure 38 - Definition of layers for deep neural network.

The Batch Normalization layers were used to normalize the input layers and the hidden layers, in order to adjust the averaging and scaling of activations. This normalization effect makes the neural network able to use a higher learning rate without disappearing or exploding the gradients.

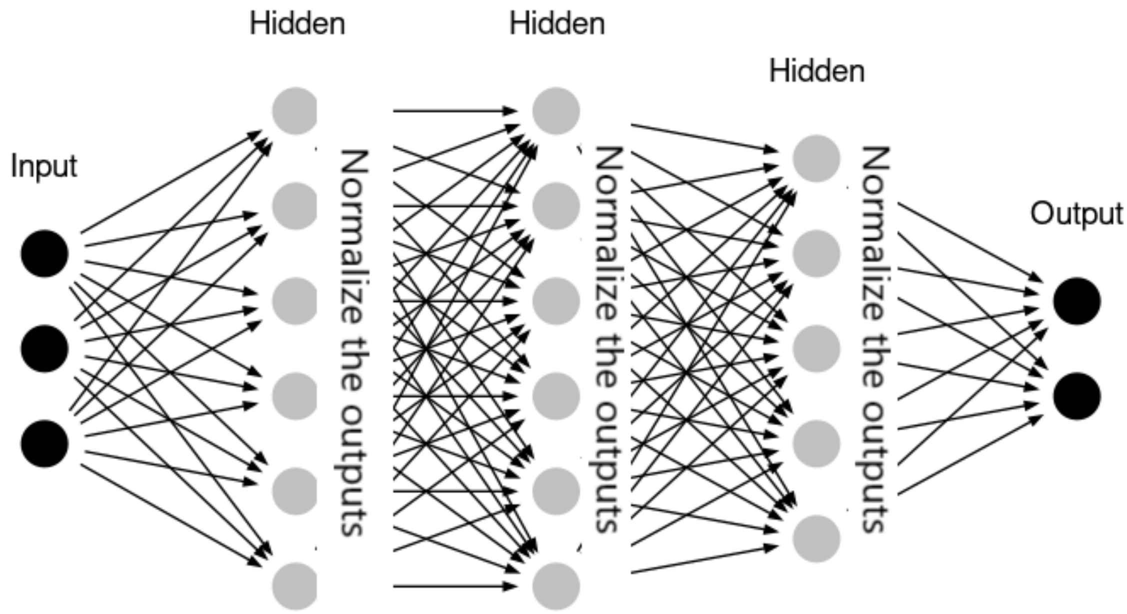


Figure 39 - Normalization layers

Source:(LaptrinhX, 2020)

To understand the Convolution layer (Conv2D), it needs to understand how the images are formed. The inputs are normally three-dimensional arrays with width and height and depth, a given number of colour channels.

Convolutions work like filters that see small squares and “slip” throughout the image, capturing the most striking features. To explain better, with a 32x32x3 image and a filter that covers a 5x5 area of the image with a 2-hop movement (called a stride), the filter will pass through the entire image, through each of the channels, forming in the end a feature map or activation 28x28x1 map. The output depth of a convolution is equal to the number of filters applied. The deeper the layers of the convolutions, the more detailed the traces identified with the activation map. (Ekman, 2021)

So, the convolution layer, in short, condenses the image matrix into a smaller matrix, making it leaner. It uses an activation function called “ReLU”, to be more computationally efficient and to avoid losses in accuracy.

The ReLU function returns 0 for all negative values, and the value itself for positive values. It is a computationally light function; however it is not zero-cantered. As its result is zero for negative values, it tends to “erase” some neurons during a forward step, which increases the training speed, but on the other hand it can cause these neurons to “die” and not learn anything if they only receive negative values. Additionally, it can produce exploded activations as it does not have a positive threshold. Even with its limitations, the ReLU function is today one of the most used activation functions in neural network training and is not usually used in the output layer. (Ceccon, 2020).

Another layer used in the deep artificial neural network is the pooling layer. This layer will serve to simplify the previous layer, similar to the convolution layer is selected a unit of area. This unit is responsible for summarizing the information into a single value. For example, if the previous layer is 24x24, after pooling the output will be 12x12. So, for that, it uses Keras' Max Pooling 2D function, in which it will go through the previous layer and just select the layer with the highest value and then go to the output. (Ekman, 2021)

According to Bianchi, this function becomes important not only to decrease the variances to small changes, but also to reduce the number of parameters. (Bianchi, 2020).

To avoid overfitting (statistic of when the model adjusts very well to the training data, but becomes ineffective when predicting new results), it uses the Dropout layer. This function is used to turn off some neurons and their connections randomly during training. During prediction these neurons are activated again. Therefore, Dropout will force a neural network to learn new, more robust features, avoiding generalization.

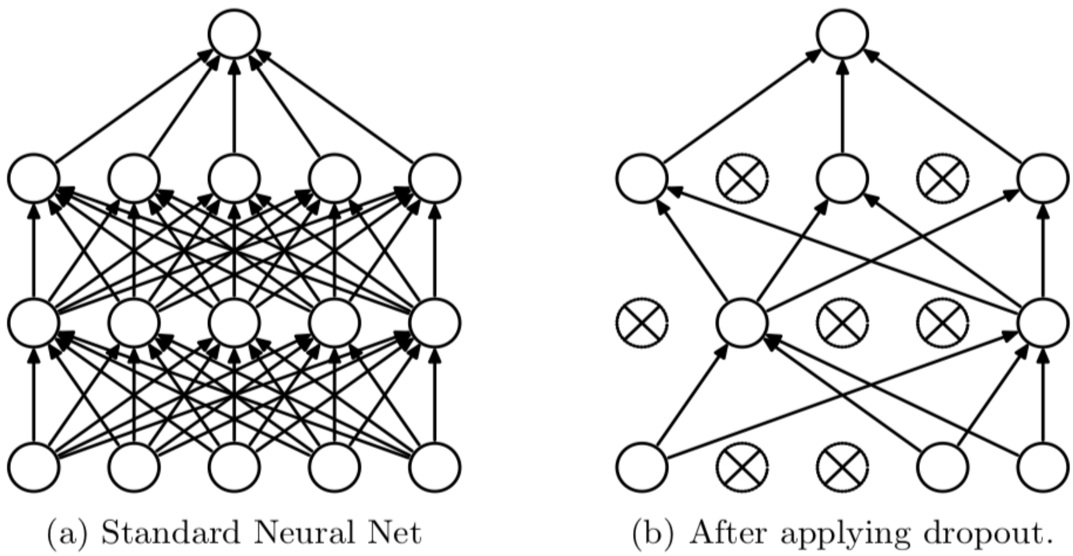


Figure 40 - Dropout: a simple way to prevent neural networks from overfitting

Source: (Freitas and Freitas, 2020)

The Flatten layer or flattened layer is responsible for flattening tensors, that is, it suppresses all dimensions except one. Change the shape of the tensor so that its shape is equal to the number of elements in the tensor. What the Flatten layer does is multiply the three dimensions (4*4*512=8192) and convert them into one so that it can go into the next Dense layer whose input is 1D tensors During the Flatten layer, arrays resulting from previous layers such as pooling and convolutions are scaled to become a linear, single-dimensional array. This step is a preparation to enter the main layer of the fully connected neural network.

Figure 41 - Flatten Operation

Source: (SuperDataScience APUD Clappis, 2018)

Finally, the Dense layer is used, which aims to calculate an activation function according to the data and weights, in this way each neuron within a dense layer will fully connect to each neuron in the previous layer. To activate this layer, the Softmax function was used, this mathematical function will convert the vector of numbers into a vector of probabilities, where each value is proportional to the relative scale of each value in the vector.

Keras has a commonly used callback function (training data monitoring function) to avoid overfitting as well. Early Stopping allows you to specify a performance measure that will be monitored and when this function is triggered it will stop the training process.

It will use two arguments in this method: the monitor and the patience. The monitor will be the function to monitor, the loss data from the validation set. So that way the training will stop when the monitored data stops improving.

The patience argument, will set a delay on the trigger of the Early Stopping function relative to the number of epochs in which it would not like to see any improvement. This is done because the first sign that there is no improvement may not be the best time to stop training, as there may be a loss at the beginning followed by a big improvement.

8.3 Evaluation and Metrics

The cost function adopted to perform the model evaluation is the *Categorical Crossentropy* due it is used as a loss function for multi-class classification model where there are two or more output labels, the result can be Covid, Normal or Pneumonia.

8.3.1 Accuracy

As evaluation metrics, the accuracy was adopted the accuracy provides information about the number of diagnoses that the model was able to get right out of the total. The equation below defines the accuracy, where TP is the number of true positives, TN are the true negatives, FP are the false positives and FN the false negatives.

$$\mathrm{A} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}$$

Figure 42 - Accuracy equation.

8.3.2 Precision

The precision is determined by the proportion of real sick individuals among those identified as infected by COVID-19

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Figure 43 - Precision equation.

8.3.3 Recall

The recall concerns the number of people diagnosed as infected among all sick individuals.

$$R = \frac{TP}{TP + FN}$$

Figure 44 - Recall equation.

8.3.4 F1-Score

The F-score encompasses the precision and recall metrics, seeking to maximize both simultaneously. The calculation of the F1-score is given by the harmonic mean between precision and recall.

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = 2 \cdot \frac{P \cdot R}{P + R}$$

Figure 45 - F1-score equation.

9. Experiments

The tool was developed in Python programming language with coda base, with the help of available libraries. TensorFlow and Keras were used as a machine learning library. The network architectures described in Section 8.2; the SARS-CoV-2 x-ray database detailed in Section 8.1. Initially, the segmentation networks were trained, all adopting the Adam optimizer with a batch size of 32 a dense layer 256 activation relu, a dropout of 0.15 and a dense layer 3 activation softmax, with a callback early stop function to stop the training if the next accuracy is lower than the previous one.

10. Results

The results of the adopted networks are presented in this section, addressing the network architectures VGG-16, Xception and one model made from scratch, respectively and the prediction made from the results.

10.1. Results from the VGG16 model

Next, the pre-trained model VGG16 will be analysed, as it can be seen, through the figure of the network being trained, the training process had 10 iterations, the epoch was set to 10 interactions due the time consuming to the program

```
history_VGG16 = tfVGG16_model.fit(train_iter, epochs = 10,batch_size = 16,callbacks=[early_stopping],
                          validation_data=test_iter)

Epoch 1/10
436/436 [==============================] - 210s 471ms/step - loss: 1.0473 - accuracy: 0.4943 - val_loss: 0.9660 - val_accuracy: 0.5007
Epoch 2/10
436/436 [==============================] - 201s 460ms/step - loss: 0.9007 - accuracy: 0.5382 - val_loss: 1.0367 - val_accuracy: 0.3788
Epoch 3/10
436/436 [==============================] - 199s 457ms/step - loss: 0.7562 - accuracy: 0.6574 - val_loss: 0.9836 - val_accuracy: 0.5532
Epoch 4/10
436/436 [==============================] - 199s 457ms/step - loss: 0.6711 - accuracy: 0.7142 - val_loss: 0.8375 - val_accuracy: 0.5974
Epoch 5/10
436/436 [==============================] - 199s 456ms/step - loss: 0.6220 - accuracy: 0.7477 - val_loss: 0.8100 - val_accuracy: 0.6181
Epoch 6/10
436/436 [==============================] - 198s 455ms/step - loss: 0.5500 - accuracy: 0.7791 - val_loss: 0.7903 - val_accuracy: 0.6222
Epoch 7/10
436/436 [==============================] - 198s 454ms/step - loss: 0.5458 - accuracy: 0.7833 - val_loss: 0.7984 - val_accuracy: 0.6262
Epoch 8/10
436/436 [==============================] - 197s 452ms/step - loss: 0.4953 - accuracy: 0.8028 - val_loss: 0.6810 - val_accuracy: 0.6784
Epoch 9/10
436/436 [==============================] - 198s 453ms/step - loss: 0.4714 - accuracy: 0.8090 - val_loss: 0.6457 - val_accuracy: 0.6801
Epoch 10/10
436/436 [==============================] - 198s 454ms/step - loss: 0.5549 - accuracy: 0.7737 - val_loss: 0.6960 - val_accuracy: 0.6610
```

Figure 46 - VGG-16 model training

Following the metrics presented during the iterations, the training data started with an average accuracy rate of 78%, but as the model underwent training, these rates showed a huge growth, going from an accuracy of only of 49% to up to 80%. The validation data also started with a rate of 50% and had a improvement arriving at 66%.

Regarding the data loss rate, that is, the numbers that indicate how bad the prediction of an image was, as it can be seen in the graph below that the training data started with a high rate of about 1.047 and over the iterations decreased, reaching numbers as low as 47% loss.

The validation data also started with a high number of losses, about 96%, and had a significant decline until 69%. It is possible to verify these results from the following graph generated through the prototype.

Blue is Accuracy and Loss.

Orange is validation accuracy and validation loss.

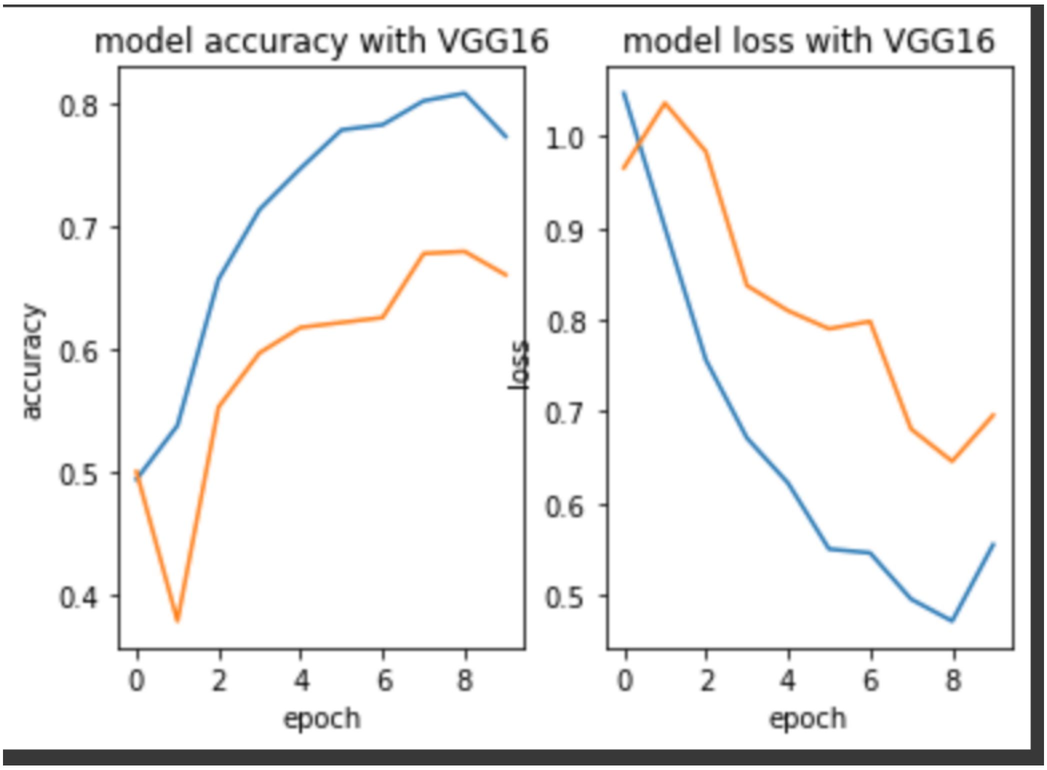


Figure 47 - Accuracy and Loss graph model VGG-16

Because the results showed a good accuracy a confusion matrix was generates.

	Prediction			
True Value		Covid	Normal	Pneumonia
	Covid	184	170	146
	Normal	158	156	186
	Pneumonia	168	172	160

Table 3 - VGG-16 model Confusion Matrix Data

10.2. Results from the Xception model

The last pre-trained model to be analysed is the Xception model, as it can be seen, through the figure of the network being trained, the training process ended with only 13 iterations, not 20 iterations. Again, this is due to the Early Stopping callback function that was put in the development of the proof of concept.

The accuracy of the validation data started to decline, so the callback function was included, completing the training.

```
history_Xception = tfXception_model.fit(train_iter, epochs = 20,batch_size = 16,callbacks=[early_stopping],
                    validation_data=test_iter)

Epoch 1/20
948/948 [==============================] - 593s 606ms/step - loss: 0.5276 - accuracy: 0.8444 - val_loss: 0.9290 - val_accuracy: 0.6907
Epoch 2/20
948/948 [==============================] - 573s 604ms/step - loss: 0.3089 - accuracy: 0.8916 - val_loss: 0.8582 - val_accuracy: 0.8227
Epoch 3/20
948/948 [==============================] - 572s 603ms/step - loss: 0.2192 - accuracy: 0.9228 - val_loss: 0.2247 - val_accuracy: 0.9247
Epoch 4/20
948/948 [==============================] - 571s 602ms/step - loss: 0.2001 - accuracy: 0.9312 - val_loss: 0.7561 - val_accuracy: 0.9220
Epoch 5/20
948/948 [==============================] - 571s 602ms/step - loss: 0.2834 - accuracy: 0.8972 - val_loss: 2.2175 - val_accuracy: 0.3993
Epoch 6/20
948/948 [==============================] - 571s 602ms/step - loss: 0.3729 - accuracy: 0.8646 - val_loss: 0.9659 - val_accuracy: 0.6260
Epoch 7/20
948/948 [==============================] - 568s 599ms/step - loss: 0.2526 - accuracy: 0.9110 - val_loss: 0.3506 - val_accuracy: 0.8800
Epoch 8/20
948/948 [==============================] - 566s 596ms/step - loss: 0.1736 - accuracy: 0.9394 - val_loss: 0.1777 - val_accuracy: 0.9340
Epoch 9/20
948/948 [==============================] - 566s 597ms/step - loss: 0.1651 - accuracy: 0.9446 - val_loss: 0.3222 - val_accuracy: 0.8927
Epoch 10/20
948/948 [==============================] - 565s 596ms/step - loss: 0.1391 - accuracy: 0.9539 - val_loss: 0.2265 - val_accuracy: 0.9247
Epoch 11/20
948/948 [==============================] - 566s 596ms/step - loss: 0.1380 - accuracy: 0.9533 - val_loss: 4.9929 - val_accuracy: 0.7627
Epoch 12/20
948/948 [==============================] - 565s 596ms/step - loss: 0.1234 - accuracy: 0.9580 - val_loss: 0.3729 - val_accuracy: 0.8547
Epoch 13/20
948/948 [==============================] - 566s 597ms/step - loss: 0.1188 - accuracy: 0.9604 - val_loss: 0.5669 - val_accuracy: 0.8953
```

Figure 48 - Xception training

Following the metrics presented during the iterations, the training data started with an average accuracy rate of 92%, but as the model underwent training, these rates showed significant growth, going from an accuracy of 84% to up to 96%. The validation data also started with a rate of 69% until they reached the accuracy range of 89% and remained with only slight changes. You can follow the evolution of accuracy from the graph below.

Regarding the data loss rate, that is, the numbers that indicate how bad the prediction of an image was, as it can see in the graph below that the training data started with a high rate of about 52% and over the iterations decreased, reaching numbers as low as 11% loss.

The validation data also started with a high number of losses, about 92%, and declined throughout the process, but on a smaller scale than the training data. The loss numbers in the validation data were higher than in the training data, staying in the range of 56%, as shown in the graph below, managed in code. It is possible to verify these results from the following graph generated through the prototype.

Blue is Accuracy and Loss.

Orange is validation accuracy and validation loss.

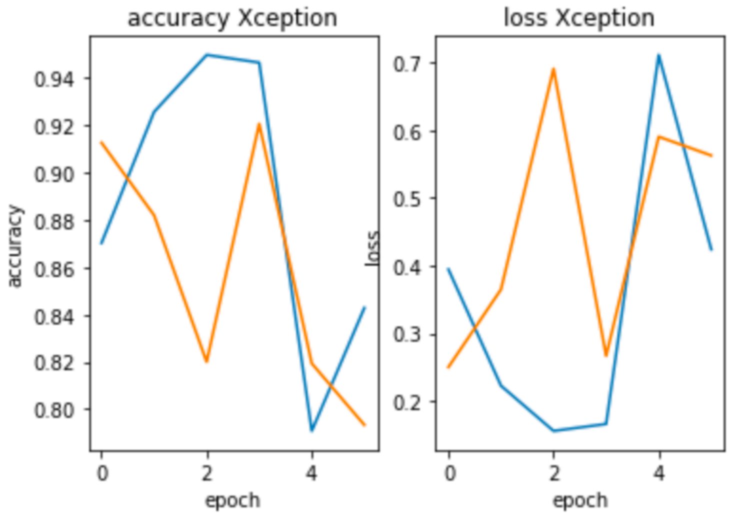


Figure 49 - Accuracy and Loss graph Xception

Because the results showed a good accuracy a confusion matrix was generates.

	Prediction			
True Value		Covid	Normal	Pneumonia
	Covid	146	170	184
	Normal	155	157	188
	Pneumonia	174	159	167

Table 4 - Xception model Confusion Matrix Data

The Xception model presented more favourable results with an accuracy rate around 95 percent as seen in the pictures above

10.3. Results from the model created from scratch

The last model to be trained is the model build from scratch, as it can be seen, through the figure of the network being trained, the training process ended with 20 iterations.

```
Epoch 1/20
948/948 [==============================] - 223s 224ms/step - loss: 0.6749 - accuracy: 0.7137 - val_loss: 0.8220 - val_accuracy: 0.5900
Epoch 2/20
948/948 [==============================] - 207s 218ms/step - loss: 0.5145 - accuracy: 0.7744 - val_loss: 0.5659 - val_accuracy: 0.7280
Epoch 3/20
948/948 [==============================] - 207s 218ms/step - loss: 0.4687 - accuracy: 0.7923 - val_loss: 0.5673 - val_accuracy: 0.7253
Epoch 4/20
948/948 [==============================] - 211s 223ms/step - loss: 0.4393 - accuracy: 0.8125 - val_loss: 0.5210 - val_accuracy: 0.7607
Epoch 5/20
948/948 [==============================] - 218s 230ms/step - loss: 0.4052 - accuracy: 0.8298 - val_loss: 0.4519 - val_accuracy: 0.8127
Epoch 6/20
948/948 [==============================] - 219s 231ms/step - loss: 0.3761 - accuracy: 0.8449 - val_loss: 0.4206 - val_accuracy: 0.8213
Epoch 7/20
948/948 [==============================] - 215s 227ms/step - loss: 0.3474 - accuracy: 0.8565 - val_loss: 0.4156 - val_accuracy: 0.8287
Epoch 8/20
948/948 [==============================] - 218s 230ms/step - loss: 0.3341 - accuracy: 0.8643 - val_loss: 0.3574 - val_accuracy: 0.8753
Epoch 9/20
948/948 [==============================] - 218s 229ms/step - loss: 0.3127 - accuracy: 0.8767 - val_loss: 0.4323 - val_accuracy: 0.8353
Epoch 10/20
948/948 [==============================] - 209s 220ms/step - loss: 0.3039 - accuracy: 0.8817 - val_loss: 0.4310 - val_accuracy: 0.8273
Epoch 11/20
948/948 [==============================] - 208s 220ms/step - loss: 0.2879 - accuracy: 0.8870 - val_loss: 0.4418 - val_accuracy: 0.8227
Epoch 12/20
948/948 [==============================] - 207s 219ms/step - loss: 0.2776 - accuracy: 0.8931 - val_loss: 0.3633 - val_accuracy: 0.8680
Epoch 13/20
948/948 [==============================] - 210s 221ms/step - loss: 0.2709 - accuracy: 0.8928 - val_loss: 0.3171 - val_accuracy: 0.8880
Epoch 14/20
948/948 [==============================] - 212s 223ms/step - loss: 0.2596 - accuracy: 0.9018 - val_loss: 0.3001 - val_accuracy: 0.8967
Epoch 15/20
948/948 [==============================] - 215s 226ms/step - loss: 0.2522 - accuracy: 0.9035 - val_loss: 0.3336 - val_accuracy: 0.8727
Epoch 16/20
948/948 [==============================] - 215s 226ms/step - loss: 0.2420 - accuracy: 0.9079 - val_loss: 0.3289 - val_accuracy: 0.8787
Epoch 17/20
948/948 [==============================] - 211s 222ms/step - loss: 0.2445 - accuracy: 0.9073 - val_loss: 0.3150 - val_accuracy: 0.8860
Epoch 18/20
948/948 [==============================] - 210s 221ms/step - loss: 0.2322 - accuracy: 0.9114 - val_loss: 0.3428 - val_accuracy: 0.8680
Epoch 19/20
948/948 [==============================] - 211s 223ms/step - loss: 0.2348 - accuracy: 0.9125 - val_loss: 0.2617 - val_accuracy: 0.9073
Epoch 20/20
948/948 [==============================] - 204s 215ms/step - loss: 0.2239 - accuracy: 0.9157 - val_loss: 0.2786 - val_accuracy: 0.9073
```

Figure 50 - Model from scratch training

Following the metrics presented during the iterations, the training data started with an average accuracy rate of 71%, but as the model underwent training, these rates showed significant growth, going from an accuracy of 71% to up to 91%. The validation data also started with a rate of 59% until they reached the accuracy range of 90% and remained with only slight changes. You can follow the evolution of accuracy from the graph below, also managed in proof-of-concept code.

Regarding the data loss rate, that is, the numbers that indicate how bad the prediction of an image was, it can be seen in the graph below that the training data started with a high rate of about 67% and over the iterations decreased, reaching numbers as low as 22% loss.

The validation data also started with a high number of losses, about 82%, and declined throughout the process, but on a smaller scale than the training data. The loss numbers in the validation data were higher than in the training data, staying in the range of 27%, as shown in the graph below, managed in code. It is possible to verify these results from the following graph generated through the prototype.

Blue is Accuracy and Loss.

Orange is validation accuracy and validation loss.

Figure 51 - Accuracy and Loss graph Model from scratch

The model made from scratch was the one that obtained a better result around 91 percent accuracy in the results.

	Prediction			
True Value		Covid	Normal	Pneumonia
	Covid	158	189	153
	Normal	160	189	151
	Pneumonia	149	211	140

Table 5 – model from scratch Confusion Matrix Data

10.4 Prediction

After training the data, the method to perform the prediction of the images was produced in which the image is transformed into a vector and its dimensions expanded, so that it is possible to use the Keras predict function, which embodies the final predictions. In this method, it was only necessary to pass the path of the image location that is desired to be predicted and the neural network processed and made available in the console which data class the image belongs to: normal/healthy lung, lung with COVID-19 or lung with viral pneumonia. Five predictions were performed to test the neural network. As a result, the model correctly corrected the four diseases only from the images, as shown in the image below, which shows the function that was created for the prediction of radiographic images:

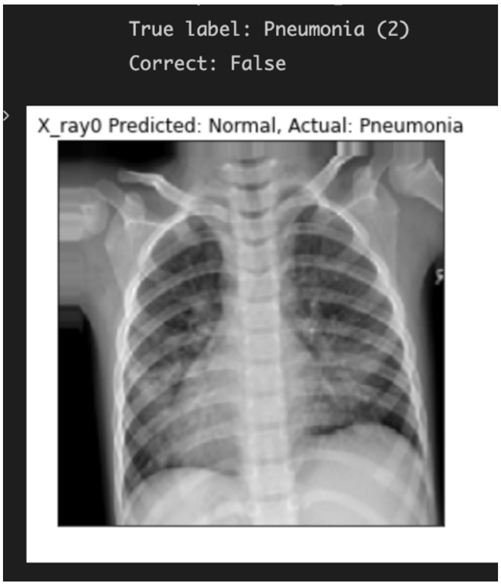


Figure 52 - Wrong prediction

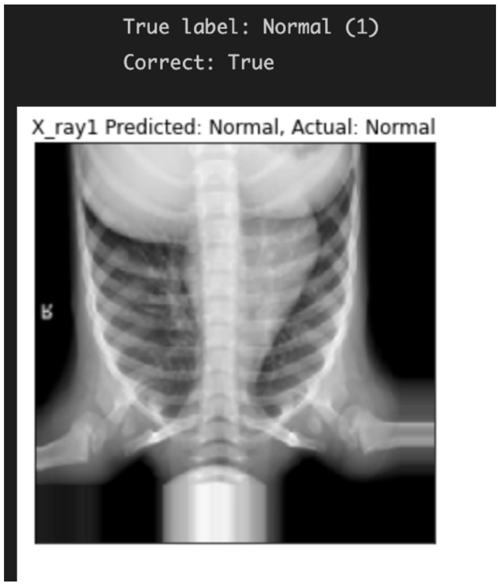


Figure 53 - Right prediction (normal lung)

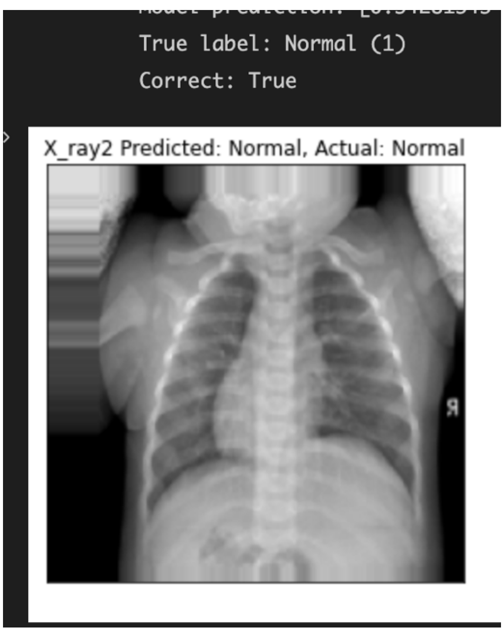


Figure 54 - Right prediction (normal lung)

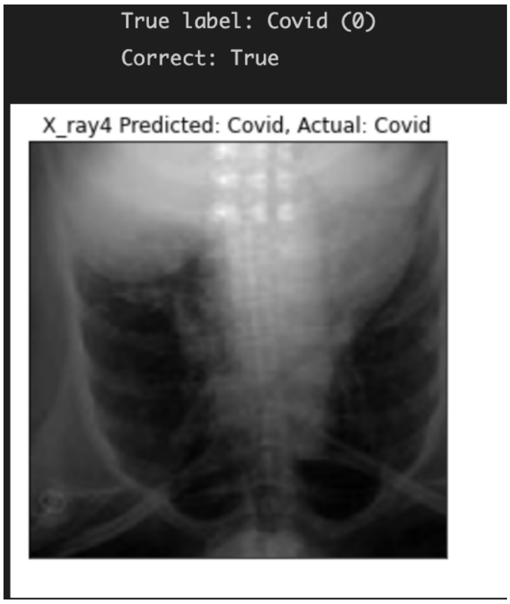


Figure 55 - Right prediction (Covid lung)

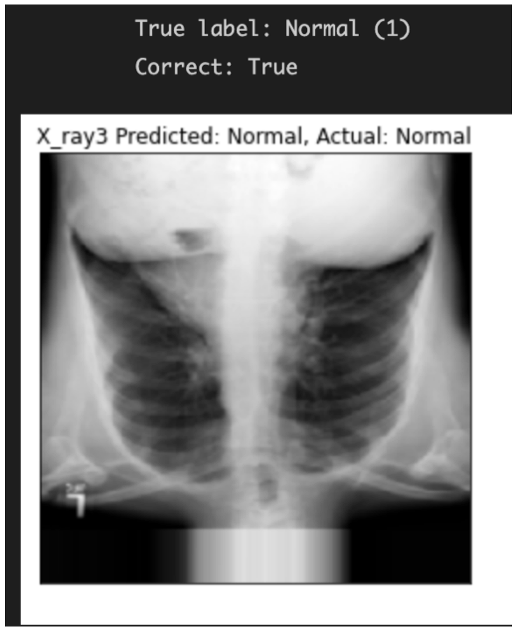
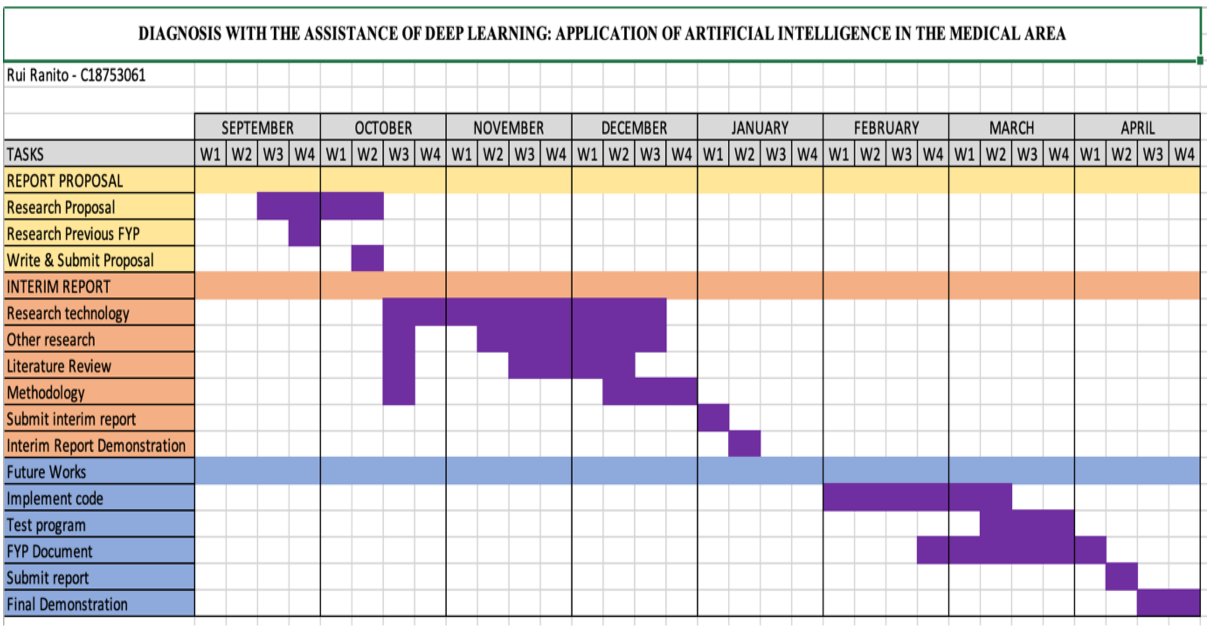


Figure 56 - Right prediction (normal lung)

11. GANTT Chart



12. Conclusions & Future works

12.1. Conclusions

In this work, four convolutional neural network architectures were applied to solve the problem of detecting COVID-19 through lung x-ray images. Among the networks discussed, two are pre-trained architectures, they are VGG16and Xception and one was created from scratch.
Among the pre-trained networks, VGG16 obtained an accuracy of 80%, and the Xception network obtained an accuracy of 96%, the network created from scratch obtained an accuracy of 91%.
At the end of this monograph, one can see the importance of Computer Science as a support to the medical field, recognizing the importance of knowledge of Artificial Intelligence as transversal and indispensable to everyone's daily life.
It highlights the evidence of Deep Learning structures in the reading of diagnostics by image, being assertively accurate the findings of the proof of concept, proving to be a reliable technology for medical use, which still has a lot to be developed, many possibilities of expansion and application, aiming always committed to everyone's quality of life.
Deep Learning structures can still be better explored, however, research, study, attempts are the bases that will carry forward the development of Artificial Intelligence.
Therefore, the effectiveness of Artificial Intelligence is proven, with a focus on Deep Learning for image diagnosis, with 96% accuracy according to proof of concept.
This project allowed the realization of important observations in relation to the investigated problem, allowing the development and application of deep machine learning and image processing techniques, as well as an in-depth study on segmentation architectures of biomedical images. The prototype carried out shows that it is possible to use Deep Learning technology for the classification of images, such as radiographs in the medical field. With results of these levels, the application in the health area is possible and that this technology still has a lot to contribute.

12.2. Future works

As suggestions for future work, it can be mentioned changes in image sizes and new tests with the networks used, increasing the number of epochs, number of filters, among other changes, as well as the study of new segmentation networks of biomedical images that may present similarly good results when applied to the problem. Another activity would be the pre-training of the networks for the biomedical image segmentation function to then carry out the transfer of learning, adding a classifier, after its last layer, to be trained for the problem studied. In addition, an important next step would be the use of other databases for training and testing, as a way of evaluating and ensuring a good generalization of the model.

This project enabled the realization of important observations regarding the investigated problem, allowing the development and application of deep machine learning and image processing techniques, as well as an in-depth study of biomedical image segmentation architectures. In this way, the application of Deep Learning in radiography reading, although revolutionary, is a primary process through everything that Artificial Intelligence can still propose to the health area. Within this monograph proposal, only seeks to prove the possibility of this intervention. Once this verification has been carried out, it also intends to develop a prototype that can be used in a practical way in medical daily life, not only for reading radiography images, but also for other types of image diagnosis, above all, not only dealing with the disease. in its development phase, but also, enabling predictions through routine exams, where the doctor who will use the technology, will be able to create an action plan on the possibility of developing the disease.

Bibliography

Aggarwal, C., 2019. *Neural networks and deep learning*. New York: Springer.

Antas Jr., R. and Almeida, R., 2015. Diagnóstico médico e uso corporativo do território brasileiro: uma análise do circuito espacial produtivo dos reagentes para diagnóstico. *Saúde e Sociedade*, 24(2), pp.674-690.

Ekman, M., 2021. *Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, NLP, and Transformers using TensorFlow*. 1st ed. Addison-Wesley Professional.

Arruda, W., 1996. Wilhelm Conrad Rontgen: 100 anos da descoberta dos raios x. *Arquivo de Neuropsiquiatria*, [online] Available at: <https://www.scielo.br/j/anp/a/xqdrMP8fqDvqqSgSy9k8Dbn/?format=pdf&lang=en> [Accessed 5 November 2021].

Barizon, C., 2020. *TRANSFORMAÇÃO ÁGIL: 8 hábitos para começar a Transformação Digital e conquistar resultados incríveis*. 4th ed.

Bártholo, T. and Bártholo, R., 2009. Pneumonia Viral. [online] Available at: <http://www.sopterj.com.br/wpcontent/themes/_sopterj_redesign_2017/_revista/2009/suplemento-pneumonia/pneumonia-viral.pdf> [Accessed 21 November 2021].

Brentano, V., Schnorr, A., Lima, M. and Perez, J., 2015. Interpretando A Radiografia De Tórax Na Emergência. *Acta Medica*, [online] Available at: <https://docs.bvsalud.org/biblioref/2018/04/882682/interpretando-a-radiografia-de-torax-na-emergencia.pdf> [Accessed 4 November 2021].

Bianchi, Alexandr Luiz. Arquiteturas de Redes Neurais Convulcionais para reconhecimento de imagens. Viceri. 2020. Available at: https://viceri.com.br/insights/arquiteturas-de-redes-neurais-convolucionais-para-reconhecimento-de-imagens/ Access on: 05 Nov 2021.

Bohrer Brentano, Vicente. Schnorr, Amanda. Et Al. Interpretando a Radiografia de Tórax na Emergência. Acta Médica. Porto Alegre, RS. 2015. Available at: https://docs.bvsalud.org/biblioref/2018/04/882682/interpretando-a-radiografia-de-torax-na-emergencia.pdf Access on: 20 Nov. 2021

Brentano, V., Schnorr, A., Lima, M. and Perez, J., 2015. Interpretando A Radiografia De Tórax Na Emergência. *Acta Medica*, [online] Available at: <https://docs.bvsalud.org/biblioref/2018/04/882682/interpretando-a-radiografia-de-torax-na-emergencia.pdf> [Accessed 4 November 2021].

Chagas, Edgar Thiago De Oliveira. Deep Learning e suas aplicações na atualidade. Revista Científica Multidisciplinar Núcleo do Conhecimento. Ano 04, Ed. 05, Vol. 04. São Paulo, SP. 2019. Available at: https://www.nucleodoconhecimento.com.br/administracao/deep-learning Access on: 20 Nov. 2021.

Cireşan, D, Meier, U and Schmidhuber, J, "Multi-column deep neural networks for image classification," 2012

Goldbaum M, Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 2018.

Diogo, Maria Paula. Tecnologia e Progresso. História da Tecnologia. Universidade Federal de Pelotas, RS. 2009. Available at: https://www.docsity.com/pt/historia-da-tecnologia/4706285/ Access on: 18 Nov. 2021.

Silva, I., 2017. *CBR - Tórax: Serie Diagnostico por Imagem*. Elsevier Brasil.

Fenelon, Sandro. A Evolução da Radiologia – Os Avanços da Imagiologia e Radiologia Diagnóstica. Available at: http://www.imaginologia.com.br/dow/A-evolucao-da-Radiologia.pdf Access on: 28 Nov. 2021.

Ferneda, Edberto**.** Redes Neurais e sua aplicação em sistemas de recuperação de informação. Ci. Inf., v. 35, n.1. Brasília, DF. Available at: https://www.scielo.br/j/ci/a/SQ9myjZWLxnyXfstXMgCdcH/?format=pdf&lang=pt Access on: 11 Nov 2021.

Freitas, M. and Freitas, G., 2020. Inteligência Artificial e Machine Learning: Teoria e Aplicações. 1st ed. Independently published.
(Freitas and Freitas, 2020)

K. Fukushima, “Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biol Cybern, vol. 36, no. 4, pp. 193–202, 1980.

Guedes, Danila. Linguagem de Programação Python e Arduino como ferramenta para motivar estudantes iniciantes em programação. UTFPR. Curitiba, Paraná. 2018. Available at: http://repositorio.utfpr.edu.br/jspui/bitstream/1/9709/1/CT_COTEL_2018_1_03.pdf Access on: 29 Oct 2021.

Gulli, A., Kapoor, A. And Pal, S., 2019. Deep Learning With Tensorflow 2 And Keras;Regression, Convnets, Gans, Rnns, Nlp, And More With Tensorflow 2 And The Keras API, 2ND. 2nd Ed. [Place Of Publication Not Identified]: Packt Publishing.

Heckerling, P.S., Gerber, B.S., Tape, T.G. and Wigton, R.S., 2004. Use of genetic algorithms for neural networks to predict community-acquired pneumonia. *Artificial intelligence in medicine*, *30*(1), pp.71-84.Iofe, S and Szegedy, C, “Batch normalization: Accelerating deep network training by reducing internal covariate shif,” 2015.

Kaggle. Imagem de logotipo. Kaggle.com. 2010 Available at: https://www.kaggle.com/

Keras. Imagem de logotipo. Keras.io. 2017. Available at: https://keras.io/about/ Access on: 10 Nov 2021.

Krizhevsky A., Sutskever I., and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS’12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.

Leite, Cláudia Da Costa**.** Inteligência artificial, radiologia, medicina de precisão e medicina personalizada. Radiol Bras vol. 52 no.6 São Paulo Nov./Dec. 2019. Epub Nov 25, 2019. https://www.scielo.br/j/rb/a/ys6VTJ75xZsQcnXwKyDbVMw/?lang=pt#B5 Access on: 02 Out 2021.

LeCun Y, Botou L, G. B. Orr, and K. R. Müller, Efficient Backprop. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 9–50.

Lee, G., Nho, K., Kang, B., Sohn, K.A. and Kim, D., 2019. Predicting Alzheimer's disease progression using multi-modal deep learning approach. *Scientific reports*, *9*(1), pp.1-12.

Lobo, Luiz Carlos. Inteligência Artificial e Medicina. Rev. Brasileira de Educação Médica. Vol 4. Rio de Janeiro. 2017. Available at: https://www.scielo.br/j/rbem/a/f3kqKJjVQJxB4985fDMVb8b/?format=pdf&lang=pt Access on: 02 Out 2021.

Luger, George F. Inteligência Artificial. 6a edição. Pearson. Novo México, 2013.

Maeda, Teresinha. Noronha, Arnaldo José. O pulmão e as infecções virais. Rev. Hospital Universitário Pedro Ernesto, UERJ. Rio de Janeiro, RJ. 2010. Available at: http://bjhbs.hupe.uerj.br/WebRoot/pdf/164_pt.pdf Access on: 04 oct. 2021

Marques Azevedo, Paulo. Salomão Covas, Samuel. PACS: Sistemas de Arquivamento e Distribuição de Imagens. Revista Brasileira de Física Médica. Vol 03. São Paulo, SP. 2009. Available at: https://edisciplinas.usp.br/pluginfile.php/75402/mod_resource/content/1/RBFMv4n3a_Paulo_Mazzoncini.pdf Access on: 22 Nov 2021

Martin, Robert. Código Limpo: Habilidades Práticas do Agile Software. Alta Books. 1a Ed. Rio de Janeiro, RJ. 2011.

Martins, Inês Tomas Marques. A medicina do futuro nos dias de hoje: aplicações da inteligência artificial. Trabalho Final de Mestrado Integrado. Universidade de Lisboa. Lisboa, Portugal. 2020. Available at: https://repositorio.ul.pt/bitstream/10451/46808/1/InesTMargarido.pdf Access on: 30 oct. 2021

Medeiros, Luciano Frontino De. Inteligência Artificial Aplicada: uma abordagem introdutória. Ed. Intersaberes. Curitiba, PR. 2018

Montini, Alessandra. Machine e Deep Learning: fortes tendências para o Brasil em 2019. 2019. Available at: https://noomis.febraban.org.br/especialista/alessandra-montini/machine-e-deep-learning-fortes-tendencias-para-o-brasil-em-2019 Access on: 24 Oct 2021.

Mining, E., 2020. *Machine Learning*.

Mueller, J. and Massaron, L., 2019. *Aprendizado de máquina para leigos*. Alta Books.

Nielsen, Camille Cardoso. SILVA, Cristiane Campos. Reabilitação pulmonar em pacientes após COVID-19: uma proposta**.** UFF, Rio de Janeiro, RJ. 2020. Available at: <https://bdex.eb.mil.br/jspui/bitstream/123456789/7498/1/Cap_Camille%20Cardoso%20Nielsen.pdf> Access on: 20 Oct. 2021

Nogueira, Makeliny Oliveira Gomes; LEAL, Daniela. Teorias da Aprendizagem: um encontro entre os pensamentos filosófico, pedagógico e psicológico. Ed. Intersaberes. Curitiba, PR. 2018.

Oliveira, Aparecido Ferreira ; Lederman, Henrique Manoel; Batista, Nildo Alves. O aprendizado sobre a tecnologia no diagnóstico por imagem. Radiol Bras**.** São Paulo, v.47, n.1. 2014. Available at: <https://www.scielo.br/j/rb/a/JrXZXPVGNRjnMvZCyjQbg8v/?format=pdf&lang=pt>Access on: 28 Out. 2021.

Falcão, João V. R. Redes Neurais Deep Learning Com Tensorflow. 2019. Available at: <https://revistas.unifenas.br/index.php/RE3C/article/view/232/179>Access on 11 Oct 2021.

Paiva Antunes, Omir. Prevedello, Luciano M. O potencial impacto da inteligência artificial na radiologia. Radiol. Bras. Vol 50. São Paulo, SP. Available at: <https://www.scielo.br/j/rb/a/7brY5YfCcTGcySkwyNXSZYJ/?lang=pt> Access on: 22 Oct 2021.

Pressman, R. Engenharia de Software: Uma abordagem Profissional. 8º edição. Editora Bookman.

Radiologia, Início da Sociedade Paulista de Radiologia e Diagnóstico por Imagem. 2018. Available at: https://www.spr.org.br/a-spr/historia-da-radiologia Access on: 04 Nov 2021.

SPR - Sociedade Paulista de Radiologia e Diagnóstico por Imagem. 2018. SPR - Sociedade Paulista de Radiologia e Diagnóstico por Imagem. [online] Available at: <https://spr.org.br/> [Accessed 5 January 2022].

Reis, Henrique Marcelo Guérin. Et Al. A Inteligência Artificial – IA: À disposição de Operadores do Direito. Revista do Curso de Direito do Centro Universitário Brazcubas. 2019. Available at: https://revistas.brazcubas.br/index.php/revdubc/article/view/695/725 Access on: 08 Nov 2021.

Sabbatini, Renato. Aplicações da Inteligência Artificial na Radiologia. 2018. São Paulo, SP. Available at: https://medium.com/intelig%C3%AAncia-artificial-em-medicina/como-a-intelig%C3%AAncia-artificial-pode-ser-usada-na-radiologia-c4483b14e425 Access on: 02 Nov 2021.

Santos Koenigkam, Marcel. Et. Al. Inteligência Artificial, aprendizado de máquina, diagnóstico auxiliado por computador e radiômica: avanços da imagem rumo à medicina de precisão. Radiol. Bras. Vol 52. São Paulo, SP. Available at: https://www.scielo.br/j/rb/a/8668GxBGMLncZY3W3n8nktr/?lang=pt Access on: 22 Nov. 2021.

Silva, Fabrício Machado; Lenz, Maikon Lucian; Et Al. Inteligência Artificial. Ed. Sagah. Porto Alegre, 2019.

Sobral, Osvaldo José. Inteligência Humana: Concepções e Possibilidades. Revista Científica FacMais, volume III, Número 1. 2013. Available at: https://revistacientifica.facmais.com.br/wp-content/uploads/2013/05/3.-INTELIG%C3%8ANCIA-HUMANA-CONCEP%C3%87%C3%95ES-E-POSSIBILIDADES-Osvaldo-Jos%C3%A9-Sobral.pdf Access on: 08 Oct. 2021.

Tensorflow. Imagem de logotipo. TensorFlow.Org. 2015. Available at: https://www.tensorflow.org/datasets?hl=pt-br Access on: 08 Oct 2021.

Thoma, M. Analysis and optimization of convolutional neural network architectures. CoRR, abs/1707.09725, 2017. Available at: http://arxiv.org/abs/1707.09725 Access on: 08 Oct 2021

Thiollent, Michel. Metodologia da Pesquisa-ação. 4. ed. São Paulo: Cortez, 1986. Available at: https://www.reciis.icict.fiocruz.br/index.php/reciis/article/view/888/1533 Access on: 29 Oct. 2021

Vale, Simone do. Pequena História da Radiografia. Contemporânea. Rio de Janeiro, RJ. 2009. Available at: https://dotlib.com/blog/raio-x-a-descoberta-que-revolucionou-a-historia-da-medicina Access on: 24 Oct 2021

Vázquez, Flavio. Deep Learning made easy with Deep Conection. Becoming Human: Artificial Intelligence Magazine. 2017. Available at: https://becominghuman.ai/deep-learning-made-easy-with-deep-cognition-403fbe445351 Access on: 15 Oct 2021

Yamashita, Helio. JÚNIOR, Edson Amaro. Aspectos Básicos de tomografia computadorizada e ressonância magnética. Ver. Brasileira de Psiquiatria, vol 23. São Paulo, SP. 2001. Available at:

https://www.scielo.br/j/rbp/a/QkL5KhdH3WcYXmmHvmR7RCz/?lang=pt Access on: 17 Oct 2021

Yang, D., Martinez, C., Visuña, L. *et al.* Detection and analysis of COVID-19 in medical images using deep learning techniques. *Sci Rep* **11,** 19638 (2021). Available at: https://doi.org/10.1038/s41598-021-99015-3 Access on: 17 Oct 2021.