



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

FINAL THESIS

Bachelor's degree in Biomedical Engineering

**IMPLEMENTATION OF A 3D CNN FOR COPD
CLASSIFICATION**



Report and Annexes

Author: Elisabet-Vera Matamala Avrova
Director: Juan Ramón González
(FUNDACION PRIVADA INSTITUTO DE SALUD GLOBAL
BARCELONA)
Co-Director: Christian Mata Miquel
Call: June 2023

Resum

Segons les prediccions de la Organització Mundial de la Salut (OMS) pels voltants del 2030 la Malaltia Pulmonar Obstructiva Crònica (MPOC) es convertirà en la tercera causa de mort en tot el món.

L'MPOC és una patologia que afecta a les vies respiratòries i als pulmons. Avui en dia esdevé crònica i incurable però, és una malaltia tractable i prevenible.

Fins ara les proves de diagnòstic usades per a detectar l'MPOC es basen en l'espirometria, aquesta prova, tot i indicar el grau d'obstrucció al pas de l'aire que es produeix en els pulmons, sovint no és molt fiable. És per aquest motiu que s'estan començant a usar tècniques basades en algorismes de Deep Learning per a la classificació més acurada d'aquesta patologia, basant-se en imatges tomogràfiques de pacients malalts d'MPOC. Les xarxes neuronals convolucionals en tres dimensions (3D-CNN) en són un exemple.

A partir de les dades i les imatges obtingudes en l'estudi observacional d'ECLIPSE proporcionades per l'equip de recerca de BRGE de ISGlobal, s'implementa una 3D-CNN per a la classificació de pacients amb risc d'MPOC.

Aquest treball té com a objectiu desenvolupar una recerca extensa sobre la recerca actual en aquest àmbit i proposa millores per a l'optimització i reducció del cost computacional d'una 3D-CNN per aquest cas d'estudi concret.

Resumen

Según las predicciones de la Organización Mundial de la Salud (OMS), para alrededor del 2030, la Enfermedad Pulmonar Obstructiva Crónica (EPOC) se convertirá en la tercera causa de muerte en todo el mundo.

La EPOC es una enfermedad que afecta las vías respiratorias y los pulmones. En la actualidad, se considera crónica e incurable, pero es una enfermedad tratable y prevenible.

Hasta ahora, las pruebas de diagnóstico utilizadas para detectar la EPOC se basan en la espirometría. Esta prueba, a pesar de indicar el grado de obstrucción en el flujo de aire que ocurre en los pulmones, a menudo no es muy confiable. Es por esta razón que se están empezando a utilizar técnicas basadas en algoritmos de Deep Learning para una clasificación más precisa de esta patología, utilizando imágenes tomográficas de pacientes enfermos de EPOC. Las redes neuronales convolucionales en tres dimensiones (3D-CNN) son un ejemplo de esto.

A partir de los datos y las imágenes obtenidas en el estudio observacional ECLIPSE proporcionado por el equipo de investigación de BRGE de ISGlobal, se implementa una 3D-CNN para la clasificación de pacientes con riesgo de EPOC.

Este trabajo tiene como objetivo desarrollar una investigación exhaustiva sobre la investigación actual en este campo y propone mejoras para la optimización y reducción del costo computacional de una 3D-CNN para este caso de estudio concreto.

Abstract

According to predictions by the World Health Organization (WHO), by around 2030, Chronic Obstructive Pulmonary Disease (COPD) will become the third leading cause of death worldwide.

COPD is a condition that affects the respiratory tract and lungs. Currently, it is considered chronic and incurable, but it is a treatable and preventable disease.

Up to now, diagnostic tests used to detect COPD have been based on spirometry. Despite indicating the degree of airflow obstruction in the lungs, this test is often not very reliable. That is why techniques based on Deep Learning algorithms are being increasingly used for more accurate classification of this pathology, based on tomographic images of COPD patients. Three-dimensional Convolutional Neural Networks (3D-CNN) are an example of such techniques.

Based on the data and images obtained in the observational study called ECLIPSE, provided by the research team at BRGE of ISGlobal, a 3D-CNN is implemented for the classification of patients at risk of COPD.

This work aims to conduct extensive research on the current state of research in this field and proposes improvements for the optimization and reduction of the computational cost of a 3D-CNN for this specific case study.



Acknowledgements

M'agradaria agrair primer de tot a en Christian Mata, el tutor d'aquest projecte, per la seva dedicació i esforç persistent, malgrat els obstacles. Li agraeixo els ànims i la motivació que em va saber transmetre un cop a classe i l'interès que va saber despertar en mi envers el món del processament d'imatge biomèdica.

També a en Juan Ramón González, director del grup de recerca BRGE de ISGlobal, per deixar-me formar part del seu equip al llarg d'aquest darrer any de carrera.

Per altra banda, m'agradaria també agrair a en Ramón Mateo i en Xavier Escribà, dos companys del centre que han estat presents en tot moment i que m'han ajudat a completar els meus coneixements envers el Deep learning i les seves múltiples aplicacions. Sense la seva ajuda aquest projecte no hagués estat possible.

Finalment vull agrair-li a la meva mare i al meu germà, per tota la paciència i suport que m'han estat donant de manera incondicional al llarg dels meus estudis. A ells mai els hi podré estar prou agraïda.

I would like to first express my gratitude to Christian Mata, the supervisor of this project, for his dedication and persistent effort despite the obstacles. I appreciate his encouragement and the motivation he conveyed to me during our class sessions, as well as the interest he sparked in me towards the field of biomedical image processing.

I would also like to thank Juan Ramón González, director of the BRGE research group at ISGlobal, for allowing me to be part of his team during my final year of studies.

Furthermore, I would like to express my gratitude to Ramón Mateo and Xavier Escribà, two colleagues from the institute who have been there for me at all times and have helped me enhance my knowledge of Deep learning and its various applications. Without their assistance, this project would not have been possible.

Finally, I want to thank my mother and my brother for their unwavering patience and support throughout my studies. I can never be grateful enough to them.



Glossary

COPD: Chronic obstructive pulmonary disease

GOLD: Global Initiative for Chronic Obstructive Lung Disease

WHO: World Health Organisation

ARD: Acute Respiratory Disease

CRD: Chronic Respiratory Disease

CT: Computed Tomography

MRI: Magnetic Resonance Imaging

NifTI: Neuroimaging Informatics Technology Initiative

DICOM: Digital Imaging and Communications in Medicine

HU: Hounsfield Units.

FEV1: Forced Expiratory Volume in 1 second

FVC: Forced Vital Capacity

AI: Artificial Intelligence

DL: Deep Learning

SVM: Support Vector Machine

RBF: Radical Basis Function

ANN: Artificial Neural Network

CNN: Convolutional Neural Network

ReLU: Rectified Linear Unit

CPU: Central Processing Unit

GPU: Graphics Processing Unit

API: Application Programming Interfase

RAM: Random-Access Memory

HDD: Hard Disk Drive

Index

RESUM	I
RESUMEN	II
ABSTRACT	III
ACKNOWLEDGEMENTS	V
GLOSSARY	VII
1. INTRODUCTION	5
1.1. Origin of this project	5
1.2. Motivation.....	5
1.3. Background and problematic	5
1.4. Objectives.....	6
1.4.1. General objective	6
1.4.2. Specific objectives	7
1.5. Project schedule.....	7
2. THEORETICAL FRAMEWORK	9
2.1. Chronic respiratory diseases (CRDs).....	9
2.2. Chronic Obstructive Pulmonary Disease (COPD)	9
2.3. The CT scan.....	12
2.4. Artificial intelligence in medicine.....	16
2.4.1. Machine learning and deep learning approaches	17
2.5. What is an Artificial Neural Network?	20
2.6. Convolutional Neural Networks (CNNs)	23
2.6.1. Overall architecture.....	25
2.6.2. Activation functions.....	28
2.6.3. Hyperparameters	29
2.7. Case of study	30
3. STATE OF THE ART	32
4. MATERIALS AND METHODS	37
4.1. Database.....	37
4.2. Requirements	40
4.3. 3D-CNN implementation	42

4.4.	3D-CNN structure	43
4.4.1.	Loading data and preprocessing.....	44
4.4.2.	Build train validation datasets	46
4.4.3.	Data augmentation	47
4.4.4.	Defining the 3D-CNN model	48
4.5.	Work process	49
4.5.1.	Beginning stages	50
4.5.2.	Final stages.....	50
4.5.3.	Hyperparameters.....	51
5.	RESULTS AND DISCUSSION _____	57
5.1.	Further works.....	62
6.	ENVIRONMENTAL IMPACT ANALYSIS _____	64
7.	BUDGET _____	65
8.	CONCLUSIONS _____	67
9.	REFERENCES _____	68
10.	ANNEX _____	72

INDEX OF TABLES

Table 1 - Diagram of the hour distribution for the project _____	8
Table 2 - Classification of airflow limitation severity in COPD (Based on post-bronchodilator FEV1) [11] _____	12
Table 3 - Overview of the 3D-CNN literature _____	32
Table 4 - ECLIPSE dataset, consent group: COPD patients [1] _____	37
Table 5 - Number of CT scans for COPD patients and for Non-smoker control patients _____	40
Table 6 - PowerEdge R640 Server main features _____	42
Table 7 - Basic libraries implemented in the code. _____	44
Table 8 - Basic structure of the 3D CNN models employed in this project. _____	49
Table 9 - Hyperparameters studied to optimize the performance of the 3D-CNN model. _____	54
Table 10 - Gridsearch results for the 3D-CNN of 4 layers. _____	55
Table 11 - Gridsearch results for the 3D-CNN of 2 layers. _____	56
Table 12 - Results of the models. _____	59
Table 13 - Results of the models with optimizers. _____	61
Table 14 - Personnel costs _____	66
Table 15 - Material costs _____	66

INDEX OF FIGURES

Figure 1 - COPD symptoms and their main risk causes [own source] [55] [56] [57]	10
Figure 2 - Flow diagram that shows the relationship between dyspnea, depression/anxiety, reduction in physical activity, impact on quality of life, and disease prognosis. COPD chronic obstructive pulmonary disease; QoL, quality of life [9].	11
Figure 3 - Thoracic axial CT-scan [13]	13
Figure 4 - How does the diagnostic test of a CT-scan procedure looks like [14]	14
Figure 5 - Structural thoracic X-ray computed tomography images	15
Figure 6 - Artificial intelligence in medicine, specifically on lung diseases screening [58]	16
Figure 7 - Supervised learning and unsupervised learning [own source]	18
Figure 8 - Machine learning overview.	19
Figure 9 - Neuron function. Flow diagram of the process and functionality of a neuron or node from a ANN [own source]	21
Figure 10 - Basic functionality of an ANN. Structure of an ANN, three main parts: (1) Input layer, (2) Hidden layer and (3) Output layer [own source]	21
Figure 11 - Representation of the loss function of a basic ANN (called J in this diagram). J is a function defined by two parameters: The predicted output and the true output.[48]	23
Figure 12 - Evolution of ANN through time, from 1943 to 2014. [own source]	24
Figure 13 - CNN structure. The functionality of a basic CNN can be down into four key components: (1) local receptive field, (2) sharing weights, (3) pooling and (4) fully connected (fc) layers. [own source]	25
Figure 14 - Convolution kernel [46]	26

Figure 15 - Representation of stride in ANNs [own source]	27
Figure 16 - Common activation functions [47]	29
Figure 17 - Basic workflow in machine learning for the selection of hyperparameters [own source].	30
Figure 18 - Statistical summary of the ECLIPSE dataset, focusing on the variable GOLD [1]	38
Figure 19 - Visualization montage of slices, 4 rows and 10 columns for 100 slices of a CT scan [own source]	39
Figure 20 - Identification of each NIfTI image for each patient.	39
Figure 21 - Jupyter and Python logos [26],[27]	41
Figure 22 - PowerEdge R640 [28]	41
Figure 23 - Main steps for the 3D-CNN implemented algorithm in Jupyter Notebooks in Python [own source].	44
Figure 24 - Graphic representation on how data is segregated into two folders [own source].	46
Figure 25 - Graphic representation of the train, validation and test dataset splits [53].	47
Figure 26 - Dimensionality of CT scan: (128, 128, 64, 1) [own source] [1]	48
Figure 27 - Beginning and final stages of the work process [own source]	50
Figure 28 - How does the learning rate affect the CNN performance. [59]	52
Figure 29 - Relu activation function [60]	52
Figure 30 - Graphic representation of model overfitting [54].	59
Figure 31 - Screenshots of the BRGE group repository on GitHub	72

1. Introduction

1.1. Origin of this project

The origin of this project comes from the observational study ECLIPSE, where 2164 COPD subjects and a smaller number of smoking controls (337) and nonsmoking controls (245) were induced in a regular observation for three years, with three CT scans (at baseline, one year, and three years) [1]. The ECLIPSE CT scans have been analysed with the VIDA software for emphysema and airway phenotypes. ECLIPSE has provided key insights into the clinical epidemiology of COPD and several papers have been published based on the ECLIPSE dataset [2].

Therefore, the purpose of this study is to expand upon and enhance the existing research by conducting an updated literature review and embarking on a self-directed learning journey.

1.2. Motivation

The motivation to focus my final thesis on biomedical images rised after taking a class called Biomedical Image Processing imparted by Dr. Christian Mata, the project's supervisor. This topic immediately piqued my interest and made me want to know more about it. Once I realized that I wanted to go deeper into this research subject, I searched for suitable internship positions, aiming to see more professional applications of the concepts that have fascinated me so much over the last year. In addition, I did an internship last summer in a project called Mobilise-D in the research center ISGlobal (Barcelona) which led me to the Bioinformatics Research Group for Epidemiology (BRGE), a research group specialized in medical data analysis and also medical imaging, where I ended up doing my internship and my final degree project focusing on applying 3D-CNN for the classification of COPD patients.

It is important to highlight that the Bachelor's degree program in Biomedical Engineering offers only one course dedicated to teaching the Python programming language and another course focused on image processing. Consequently, although my previous knowledge and experience in this specific field were limited, I possessed a strong desire to explore and gain a more profound comprehension of bioinformatics and its role in advancing healthcare.

1.3. Background and problematic

The World Health Organization has projected that COPD will become the third leading cause of death worldwide by 2030. However, there remains a lack of effective therapies that can significantly impact the outcomes of COPD. This deficiency can be partly attributed to the overreliance on spirometry

measurements. While spirometry offers a convenient and reliable method to assess lung function, it often fails to reveal distinct COPD phenotypes, as it represents a comprehensive summary of various potential pathologies associated with the disease. Consequently, the unique contributions of airway and parenchymal abnormalities may go unnoticed.

Fortunately, recent advancements in thoracic X-ray computed tomography (CT) and magnetic resonance imaging (MRI) have allowed for the identification and measurement of lung structure and function abnormalities on a regional basis. These imaging endpoints have the potential to serve as biomarkers for COPD, aiding in the characterization of patients with specific underlying pathological phenotypes and ultimately improving patient care and outcomes [3].

The progress in pulmonary imaging techniques has revealed growing evidence that different COPD patients, despite having similar pulmonary function measurements, experience the disease in diverse ways, with variations in symptoms and exercise capacities. As treatment approaches become more personalized, imaging techniques may become indispensable in quantifying treatment responses. Furthermore, imaging provides a unique opportunity to visually comprehend the true nature of COPD.

Therefore, one of the primary objectives in COPD research is to explore approaches that can identify patients based on specific underlying pathological phenotypes, leveraging the potential of advanced imaging techniques to enhance patient care and outcomes.

Hence, it is crucial to advance research in this field to achieve more precise detection of COPD in patients. This project endeavors to explore the potential of developing an algorithm that can identify this condition solely by analyzing a patient's CT scans, employing a 3D-Convolutional Neural Network. This approach holds promise even in the presence of diverse and complex characteristics associated with the pathology.

1.4. Objectives

1.4.1. General objective

The main objective of this BSc thesis is to classify digital thorax CT scans from pathological COPD cases to healthy lungs. The motivation of this objective is born under the following points:

- It is the most common lung disease for middle-aged or older adults that smoke or that live in polluted areas, such as big cities.
- It is the third-leading cause of mortality worldwide [4].
- Its early and accurate diagnosis would improve the cost of living for many patients.

- Its damage to the lungs caused is permanent, but early treatment can help slow down the progression of the condition.

Therefore, COPD has become a significant concern in today's world, particularly for individuals residing in urban areas exposed to high levels of air pollution and engaging in frequent smoking. It is crucial to establish standardized preventive measures to detect this disease at early stages and prevent it from progressing to severe states. Additionally, COPD is currently a chronic condition with no known cure, leading to a decrease in average life expectancy by approximately eight to nine years. Hence, there is a compelling need to develop a computer-aided diagnosis (CAD) system capable of assisting physicians in the classification of CT scan images.

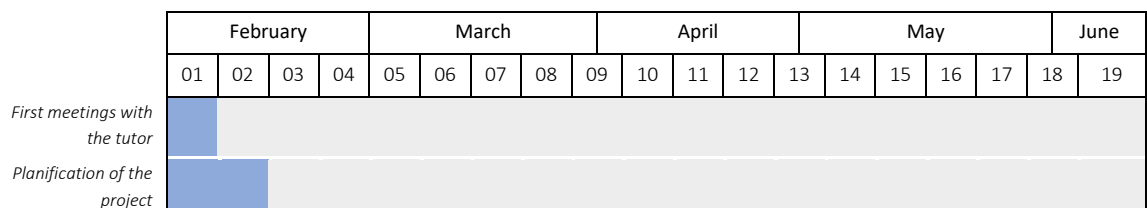
1.4.2. Specific objectives

In order to accomplish this ultimate objective, the preceding steps and objectives of this project are outlined as follows:

- Review state of the art on 3D-CNN classifications (mainly for thoracic images, CT scans, to detect COPD, COVID-19 or other respiratory diseases).
- Conduct a literature review to obtain a basis and fundamentals on the topic and classification methods in the literature to study them.
- Implement the code in Python.
- Create different 3D-CNN architectures with Python and compare the results and accuracy for the classification.
- Implement a big number of CT scan images from COPD patients from the dataset of ECLIPSE study.
- Discuss and find further improvements connected to this research work.

1.5. Project schedule

The following diagram illustrates the revised and final version of the project development steps that have been modified throughout the course of the project:



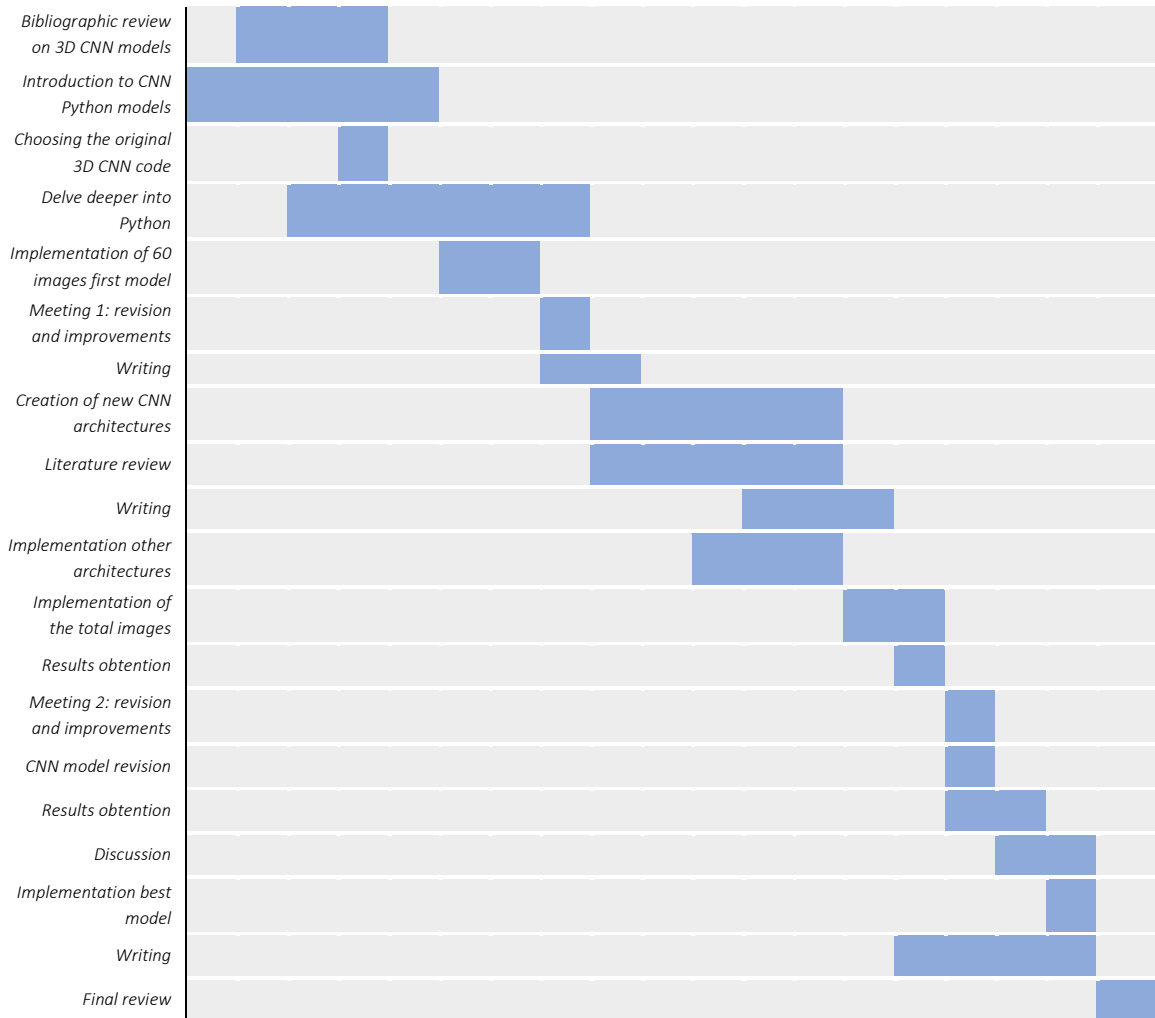


Table 1 - Diagram of the hour distribution for the project

2. Theoretical framework

2.1. Chronic respiratory diseases (CRDs)

Chronic respiratory diseases (CRDs) are a group of conditions that affect the respiratory system and cause long-term breathing problems. These diseases include chronic obstructive pulmonary disease (COPD), asthma, pulmonary fibrosis, bronchiectasis, and others [5].

CRDs are characterized by symptoms such as shortness of breath, wheezing, coughing, and chest tightness [6]. They can also lead to a decreased ability to perform physical activities, which can affect quality of life.

Risk factors for developing CRDs include exposure to environmental pollutants (such as tobacco smoke and air pollution), genetic factors, and lifestyle factors such as poor nutrition and lack of physical activity.

While CRDs cannot be cured, they can be managed with appropriate treatment and lifestyle changes. Treatment may include medications, pulmonary rehabilitation, oxygen therapy, and surgery in some cases. Lifestyle changes may include quitting smoking, avoiding environmental triggers, and engaging in regular exercise.

2.2. Chronic Obstructive Pulmonary Disease (COPD)

By 2015, the global population had reached an unprecedented 7.3 billion individuals. Experts in population studies emphasize that due to population growth and advancements in hygiene, there is a projected rise in the prevalence of chronic health conditions related to aging and smoking, including chronic obstructive pulmonary disease (COPD) [7].

COPD is a chronic respiratory disease that affects the lungs and makes it difficult to breathe. It is characterized by airflow limitation that is not fully reversible, which means that the airways in the lungs become narrowed and it becomes difficult to move air in and out of the lungs. COPD is a progressive disease, meaning that it tends to get worse over time.

The most common cause of COPD is smoking, although other factors such as exposure to air pollution, occupational dust and chemicals, and genetic factors may also contribute to its development. COPD is more common in older adults, particularly those who have smoked for many years.

Symptoms of COPD include coughing, wheezing, shortness of breath, and chest tightness. These symptoms may worsen with physical activity or exposure to irritants such as smoke or dust [8]. In advanced stages of the disease, individuals with COPD may experience difficulty performing daily activities and may require oxygen therapy.

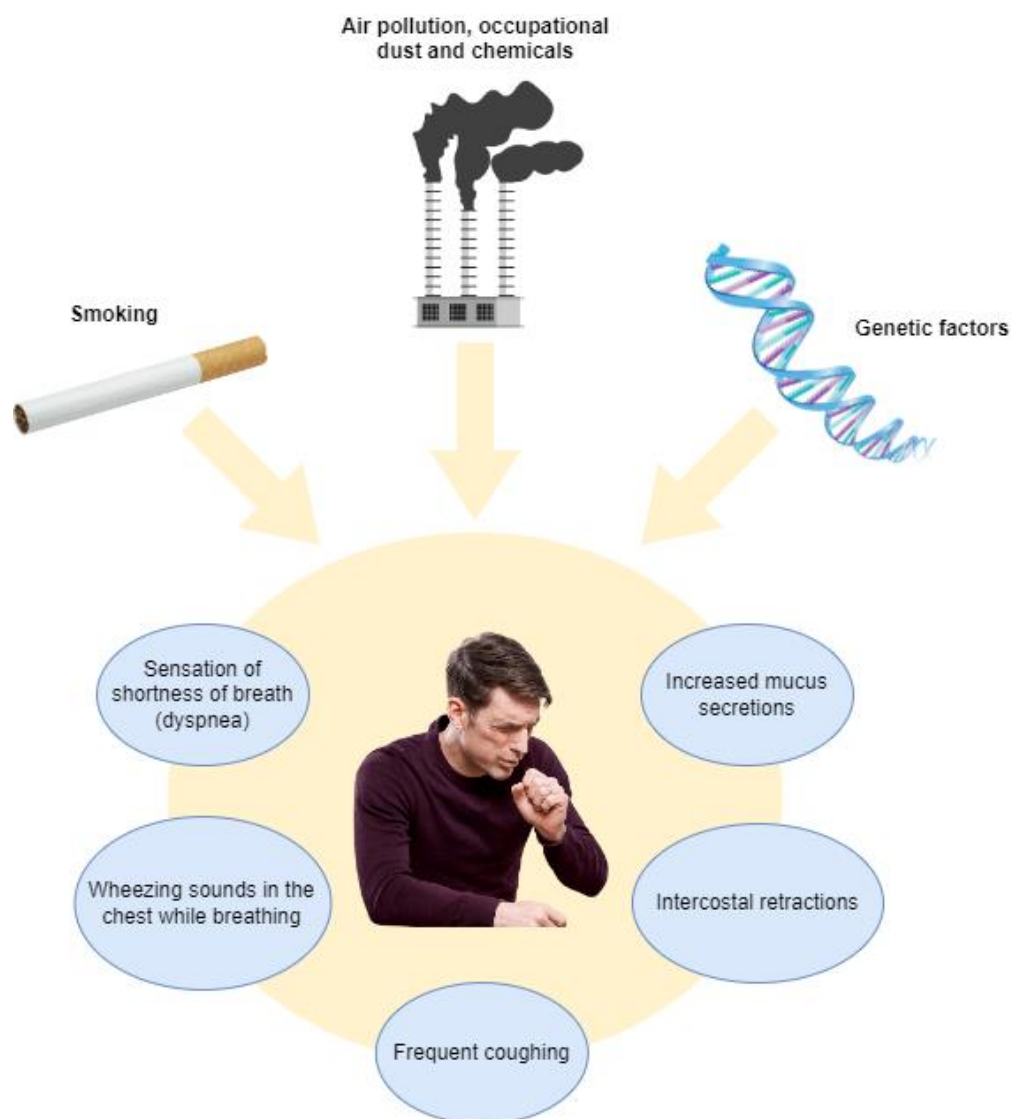


Figure 1 - COPD symptoms and their main risk causes [own source] [55] [56] [57]

The symptoms of COPD have a significant impact on the quality of life, overall health, and future outlook for individuals affected by the disease. These symptoms gradually limit a patient's ability to perform daily activities and engage in physical exertion, and they can also disrupt sleep quality. Additionally, a higher burden of COPD symptoms is linked to coexisting anxiety and depression. Moreover, the presence of COPD symptoms is associated with an elevated risk of exacerbations and a poorer prognosis for the disease [9].

The following diagram, **Figure 1**, shows how COPD respiratory symptoms, reduction in physical activity, reduction in quality of life (QoL) and mental illness such as anxiety and depression are linked to the disease prognosis of this disease.

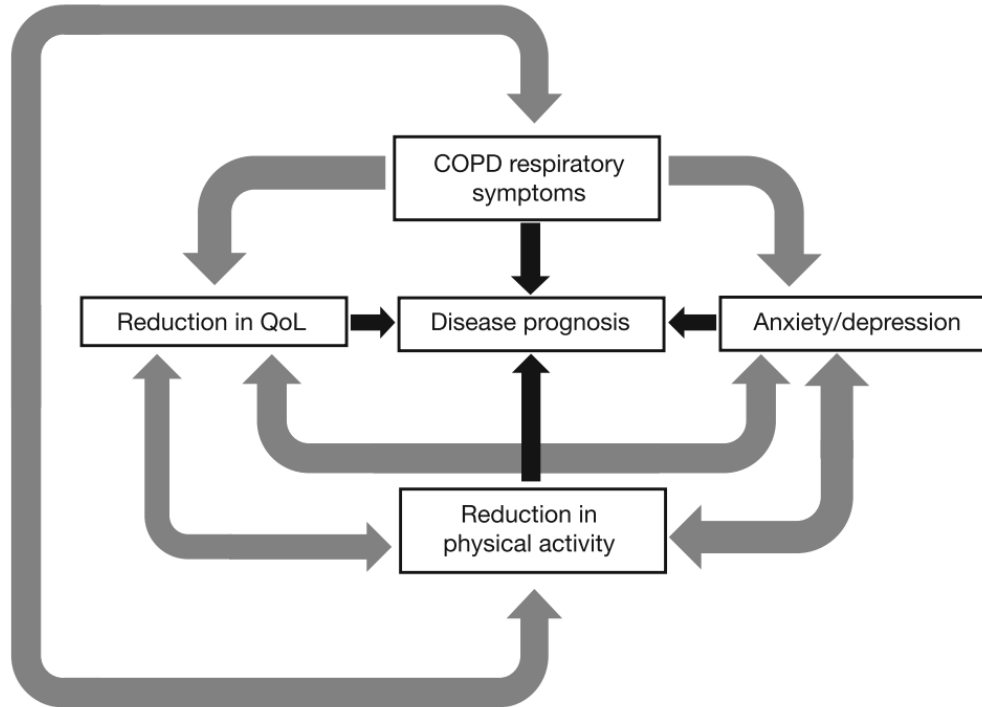


Figure 2 - Flow diagram that shows the relationship between dyspnea, depression/anxiety, reduction in physical activity, impact on quality of life, and disease prognosis. COPD chronic obstructive pulmonary disease; QoL, quality of life [9].

For now, there is no cure for COPD, but there are treatments available that can help manage symptoms and slow down the progression of the disease. These may include medications such as bronchodilators and steroids, pulmonary rehabilitation, oxygen therapy, and in some cases, surgery. Quitting smoking is also an important part of managing COPD, as continued smoking can worsen the disease and increase the risk of complications.

According to GOLD (Global Initiative for Chronic Obstructive Lung Disease) 2023, COPD is a diverse lung condition characterized by long-term respiratory symptoms (such as shortness of breath, cough, mucus production, and exacerbations) resulting from abnormalities in the airways (such as bronchitis, bronchiolitis) and/or alveoli (emphysema). These abnormalities lead to persistent, and often worsening, restriction of airflow [10].

When conducting pulmonary function testing, a ratio of forced expiratory volume in one second (FEV1) to forced vital capacity (FVC) of less than 0.7 is commonly used as a diagnostic criterion for COPD. The GOLD system, developed by the Global Initiative for Chronic Obstructive Lung Disease, classifies the degree of airflow limitation into various stages. For patients with an FEV1/FVC ratio less than 0.7.

On the following table it's shown the different stages of COPD standardized by the GOLD protocol:

CLASSIFICATION OF AIRFLOW LIMITATION SEVERITY IN COPD		
IN PATIENTS WITH FEV1/FVC < 0.70		
STAGE	Severity	FEV1 %
GOLD 1	Mild	≥ 80
GOLD 2	Moderate	50 - 79
GOLD 3	Severe	30 - 49
GOLD 4	Very severe	< 30

Table 2 - Classification of airflow limitation severity in COPD (Based on post-bronchodilator FEV1) [11]

The Global Initiative for Chronic Obstructive Lung Disease (GOLD) strategy does not routinely recommend CT scanning in COPD and only advises that it may be helpful in differential diagnosis or when surgical options are being considered. But most clinical experts recommend patients with severe COPD to have a CT scan [12].

2.3. The CT scan

A CT (computed tomography) scan works by using X-rays and computer technology to create detailed images of the inside of the body and to determine the exact size of three-dimensional objects.

During a CT scan, the individual lies on a table that moves slowly through a large, doughnut-shaped machine called a CT scanner. As the table moves through the scanner, X-ray beams are directed at the body from different angles. Detectors inside the scanner measure the amount of X-rays that pass through the body and produce a series of cross-sectional images of the body.

Because of its complexity and use of X-ray beams they are carried out in hospital by specially trained operators called radiographers.

These images are then processed by a computer to create detailed, three-dimensional images of the area being scanned. This allows healthcare providers to see the internal structures of the body, such as the lungs or other organs, in great detail.

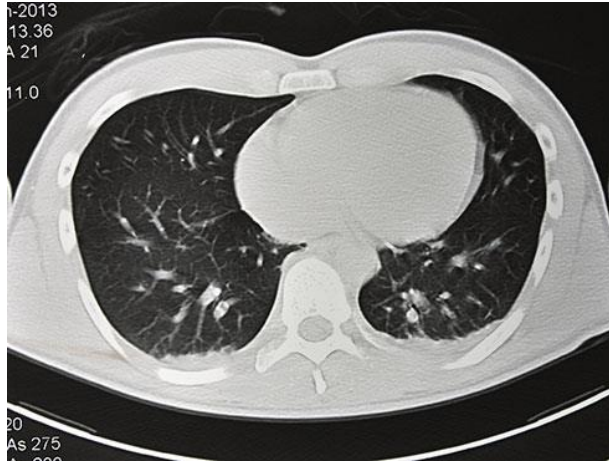


Figure 3 - Thoracic axial CT-scan [13]

In some cases, a contrast material may be injected into the individual's bloodstream to help highlight certain structures or tissues. This can be especially helpful in detecting abnormalities such as tumors or blood clots.

A CT scan of the thorax, or chest, works in the same way as a CT scan of any other part of the body. However, it is specifically focused on creating detailed images of the structures within the chest, including the lungs, heart, and surrounding tissues.

Thoracic CT scans are often used to diagnose and monitor lung diseases, such as COPD, pulmonary fibrosis, and lung cancer. They can also help detect heart and blood vessel problems, as well as abnormalities in the chest wall or other surrounding tissues.

CT scans are generally safe and non-invasive, but they do involve exposure to ionizing radiation, which can increase the risk of cancer. The amount of radiation exposure during a CT scan is generally low, and the benefits of the scan usually outweigh the risks. However, individuals who are pregnant or who have kidney problems may be advised to avoid CT scans or to take special precautions to minimize radiation exposure.



Figure 4 - How does the diagnostic test of a CT-scan procedure looks like [14]

In COPD, CT scans are used to recognise the key morphological features of emphysema, bronchial wall thickening and gas trapping. COPD is a complex and heterogeneous condition with many clinical features and a large number of different phenotypes, so relying on spirometric markers to diagnose the disease is causing some limitations on understanding the management of this condition [15].

In individuals with COPD, CT scans may show signs of emphysema, which is a type of lung damage where the air sacs in the lungs become enlarged and lose their elasticity. CT scans can also be used to identify other lung conditions that may be contributing to breathing difficulties, such as bronchiectasis or pulmonary fibrosis.

However, CT scans are not typically used as the first-line diagnostic tool for COPD. Instead, diagnosis is usually based on symptoms, lung function tests, and a physical exam. CT scans may be ordered if there is a suspicion of other lung conditions or if the individual is being considered for surgery to treat their COPD.

It's important to note that CT scans involve exposure to radiation, so they are generally reserved for situations where the benefits outweigh the risks. Your healthcare provider will discuss the risks and benefits of any diagnostic tests with you before ordering them.

In regards to CT-based imaging findings, COPD is characterized by the following abnormalities:

- Emphysematous lung
- Airway narrow-ing

- Functional small-airway disease
- Reduced lung deformation

The following **Figure 4** shows two different forms to present CT-scans; Axial CT and Coronal CT. This project will work with thorax lung images disposed in an axial form.

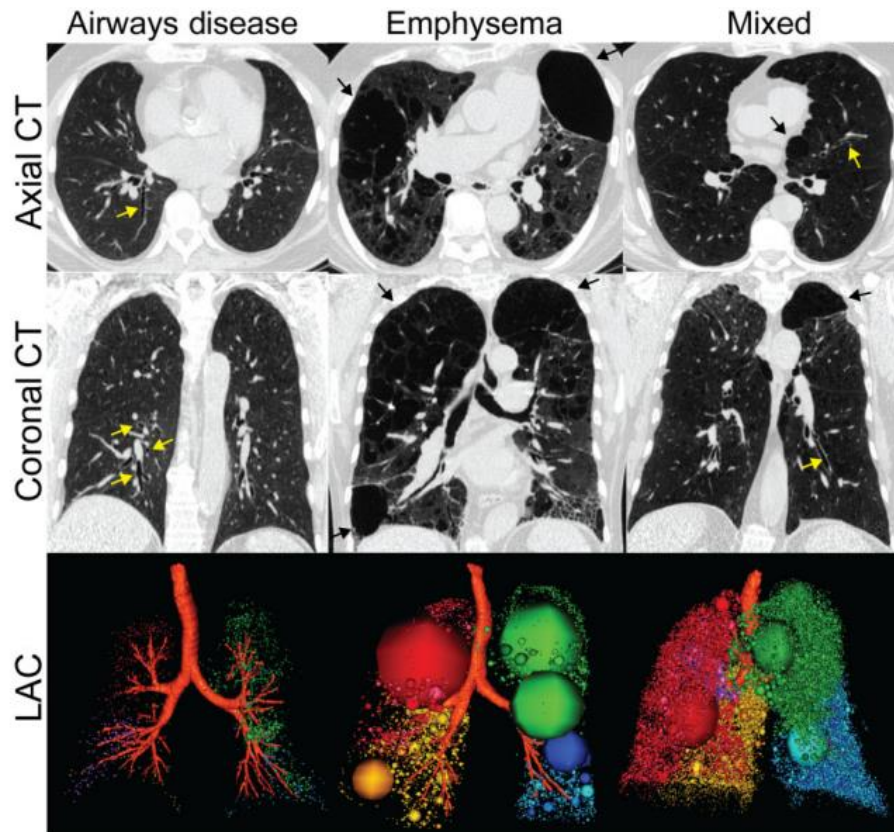


Figure 5 - Structural thoracic X-ray computed tomography images

Axial and coronal CT and low attenuation clusters (LAC) of emphysema, represented as spheres with CT densitometry values below -950 HU registered to the three-dimensional reconstruction of the airway tree shown below. Yellow arrows indicate airways disease and black arrows indicate emphysematous regions. From left to right: airways disease dominant, emphysema dominant and mixed phenotype. Airways disease dominant: $FEV1 = 103\%pred$, $DLC0 = 119\%pred$, $RA950 = 1\%$; Emphysema dominant: $FEV1 = 65\%pred$, $DLC0 = 17\%pred$, $RA950 = 37\%$; Mixed: $FEV1 = 47\%pred$, $DLC0 = 62\%pred$, $RA950 = 15\%$ [3]

In recent decades, the utilization of CT and MRI in COPD research has undergone significant advancements, offering novel perspectives on the pathophysiology of the disease through visual and quantitative representations. CT, in particular, boasts numerous advantages such as widespread availability, quick image acquisition, and the ability to provide high-resolution measurements of lung structure due to the inherent physics of X-ray-based techniques. These capabilities surpass those of other functional imaging methods.

Nevertheless, the issue of radiation dose and long-term risks continues to be a concern for healthcare providers and younger patients, despite the implementation of low-dose adaptive iteration methods.

The potential burden and associated risks associated with radiation exposure remain an ongoing consideration for both healthcare providers and patients themselves [3].

2.4. Artificial intelligence in medicine

In this project, the primary objective is to implement a 3D-CNN for the classification of COPD cases. However, in order to gain a comprehensive understanding of what we are doing and how we can enhance it, it is necessary to delve into some fundamental concepts. This includes exploring artificial intelligence and its implications in medicine, as well as delving into machine learning and deep learning approaches.

Artificial intelligence (AI) refers to the use of computers and technology to imitate intelligent behavior and critical thinking similar to that of humans. With advancements in processing power, memory, storage, and the availability of vast amounts of data, computers are increasingly tasked with complex learning endeavors. One significant area where AI finds application is in the field of medicine [29].

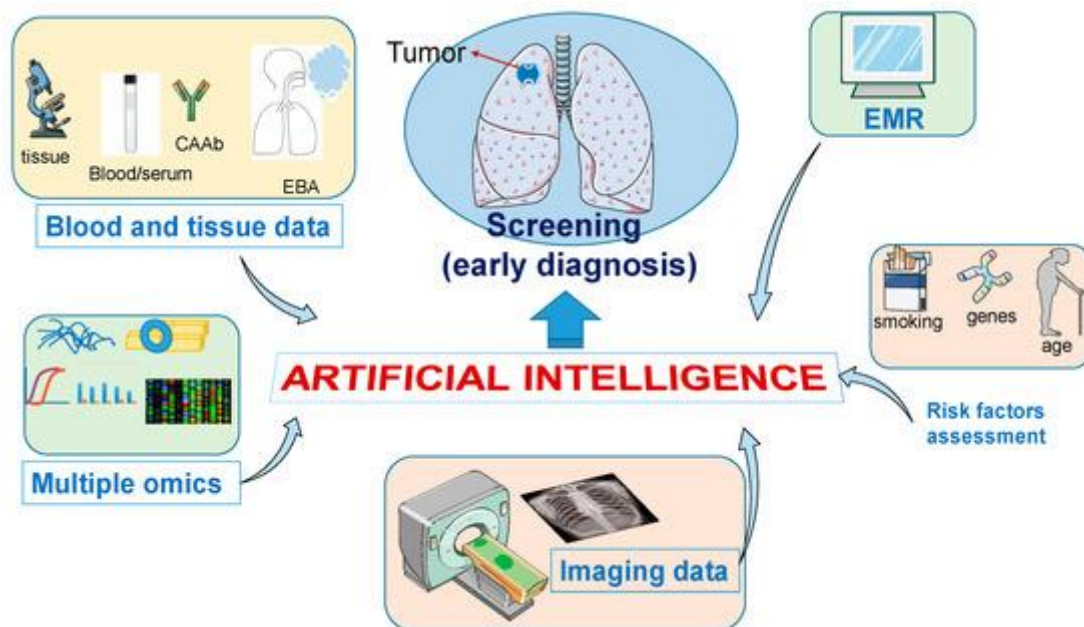


Figure 6 - Artificial intelligence in medicine, specifically on lung diseases screening [58]

The application of AI in healthcare holds the potential to decrease variability in care, enhance precision, expedite discoveries, and mitigate disparities. It enables patients to be empowered while supporting healthcare professionals in providing treatment based on the combined knowledge derived from the best medical research and analytical technology [30].

Recent studies indicate that AI technology has the potential to outperform humans in certain tasks involving visual and auditory recognition. This suggests promising applications in the medical field, particularly in medical imaging, in the near future.

Machine learning and deep learning have already made significant contributions to medical imaging and are expected to exert even greater influence in the coming years. Some examples of machine learning and deep learning implementation in the medical field are mentioned in the state-of-the-art **Section 3**, such as [20], [21],[22] .

2.4.1. Machine learning and deep learning approaches

Machine learning and **deep learning** are often mistaken and used interchangeably, but they are all sub-fields to artificial intelligence (AI) and also, two different concepts.

The model or algorithms based on machine learning are provided with the ability to learn and improve, so they will make decisions or predictions based on the data provided [31].

Within **machine learning** and artificial intelligence there are two main approaches: supervised and unsupervised learning. The first one uses labeled data to help predict outcomes, while the other does not, that's the main difference between supervised and unsupervised learning.

On one hand, as it has been said before, **supervised machine learning** is defined by its use of labeled datasets. It uses the data that has been learned in the past to new data predicting events in the future. So, by using labeled inputs and outputs, the model can measure its accuracy and learn over time. The goal is to predict outcomes for new data. This approach can be separated into two types of problems when data mining, such as classification and regression.

On the other hand, **unsupervised machine learning** differs in that the training set does not include any labels. The goal is to get insights from large volumes of new data, the algorithm determines what is different or special from the dataset. These algorithms discover hidden patterns in data without the need of human intervention. Unsupervised learning models are computationally complex because they need a large training set to produce intended outcomes. These models are used for three main tasks: clustering, association and dimensionality reduction.

The following figure shows two diagrams (scatter plots) to better understand these to concepts.

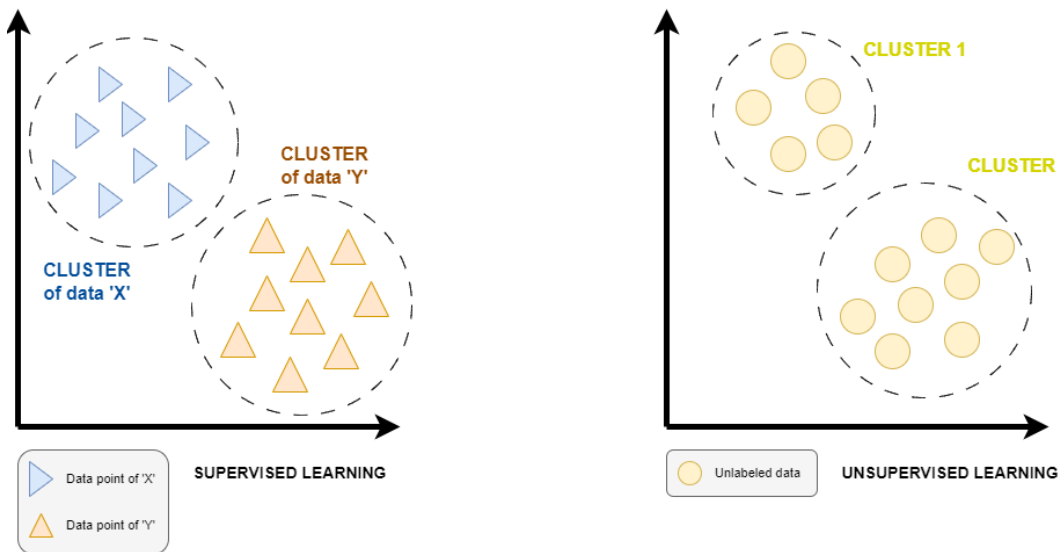


Figure 7 - Supervised learning and unsupervised learning [own source]

It is known that most image-focused pattern-recognition tasks usually depend on classification using *supervised learning*.

The most common machine learning algorithms are: Neural networks, Linear and Logistic regression, Clustering, Decision trees and Random forests.

Deep learning is a machine learning technique that teaches computers to learn to perform classification tasks directly from raw data (for example images, text and sound). Deep learning models can achieve state-of-the-art accuracy, sometimes overcoming human-level performance. Normally, models are trained using a large set of labeled data and neural network architectures that contain many layers. As a matter of course, Deep learning models require large amounts of labeled data and substantial computing power [32].

Nowadays, Deep learning algorithms drive many artificial intelligence (AI) applications and services that everyone uses in their daily day life like hands-free speakers, voice control devices and driveless cars.

To assess and then describe briefly, deep learning and machine learning differ on the way each algorithm “learns”. Deep learning can manage unstructured datasets in its raw and it can automatically determine the set of features which categorises the group of data in small subgroups, without any human intervention along the process. Whereas machine learning is more dependent on human intervention to correctly classify data. So in this last case, human experts must be trained to determine the set of dominant features to understand the differences between data inputs, usually requiring more structured data to learn [33].

In fact, neural networks are a subset of machine learning, and deep learning is also a subset of neural networks. Thanks to their potential and scalability, they have become the main tool for deep learning.

Next, is a graphic machine learning and deep learning overview.

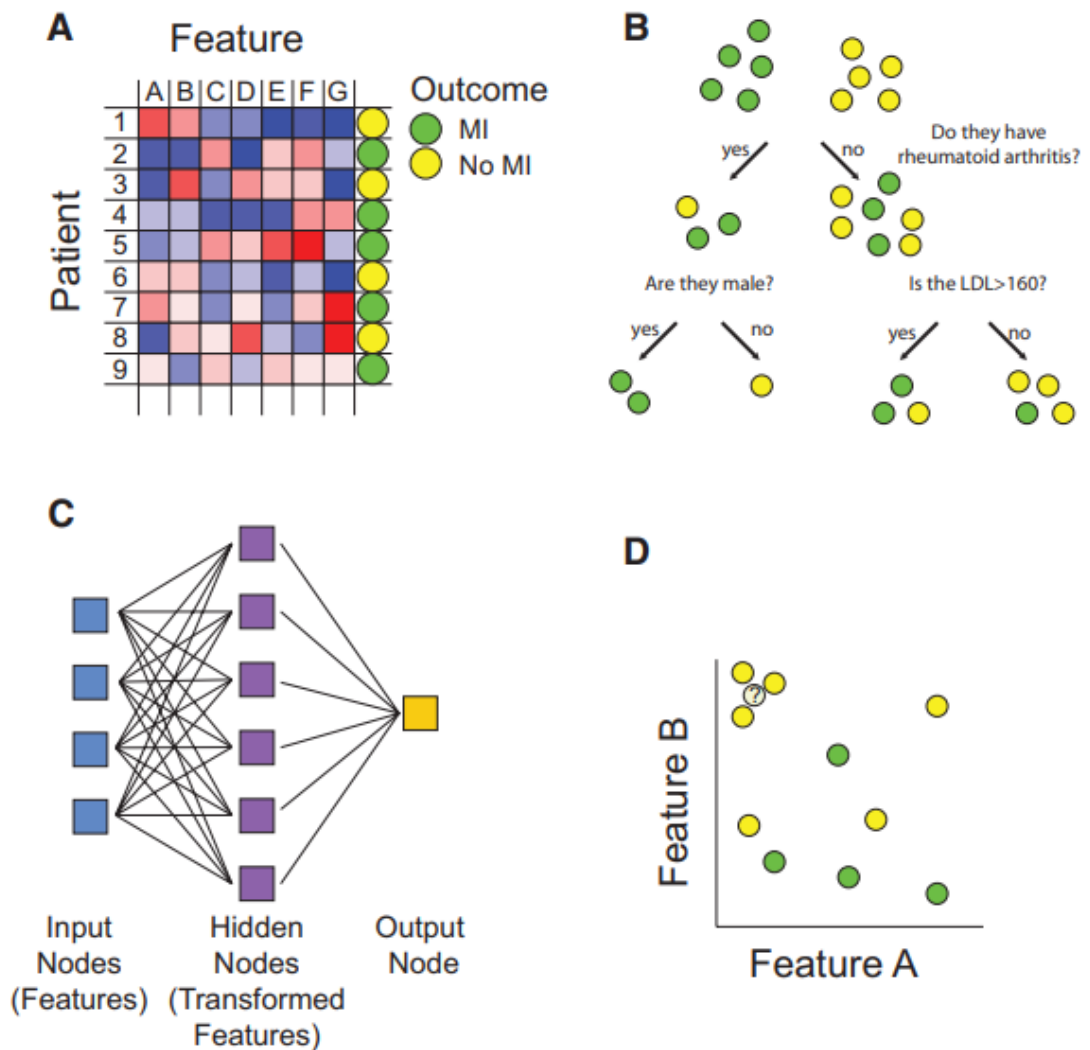


Figure 8 - Machine learning overview.

A, Matrix representation of the supervised and unsupervised learning problem. For this example the data is used in developing a model for predicting myocardial infarction (MI). For training data, we have patients, each characterized by an outcome (positive or negative training examples), denoted by the circle in the right-hand column, and by values of predictive features, as well, denoted by blue to red coloring of squares. Multiple types of functions can be used for mapping features to outcome (B through D).

B, Decision trees map features to outcome. At each node or branch point, training examples are partitioned based on the value of a particular feature. Additional branches are introduced with the goal of completely separating positive and negative training examples.

C, Neural networks predict outcome based on transformed representations of features. A hidden layer of nodes integrates the value of multiple input nodes (raw features) to derive transformed features. The output node then uses values of these transformed features in a model to predict outcome.

D, The k-nearest neighbor algorithm assigns class based on the values of the most similar training examples. The distance between patients is computed based on comparing multidimensional vectors of feature values. In this case, where there are only 2 features, if we consider the

outcome class of the 3 nearest neighbors, the unknown data instance would be assigned a "no MI" class. LDL indicates low-density lipoprotein; and MI, myocardial infarction. [34]

2.5. What is an Artificial Neural Network?

The first thing that comes to mind when hearing the name neural network are concepts related to the brain and neurons and that's exactly what it has to do. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Artificial Neural Network (ANNs) is a computational system that consists primarily of a large number of interconnected nodes, also known as neurons. Nodes interact in a distributed fashion to collectively learn from their inputs and optimize their final output [35].

The functioning of a neuron in an ANN can be summarized in the following steps:

1. **Weighted Sum:** Each input to the neuron is multiplied by its corresponding weight. These weighted inputs are then summed together.
2. **Activation Function:** The weighted sum is passed through an activation function, which introduces non-linearity to the output of the neuron. Common activation functions include sigmoid, ReLU, and tanh. The activation function determines the output value or activation level of the neuron.
3. **Output:** The output of the activation function becomes the output of the neuron and is typically passed as input to other neurons in the subsequent layers of the network.
4. **Bias:** In addition to weights, neurons often have an associated bias term. The bias is a constant value that is added to the weighted sum before passing through the activation function. It helps in shifting the activation function and introducing flexibility in the neuron's behavior.
5. **Learning:** The weights and biases of the neuron are initially assigned random values and then updated during the learning process. This is typically done through a process called backpropagation, where the network adjusts its weights based on the calculated error and the desired output.

To really understand the concepts the following figure shows how does a neuron function:

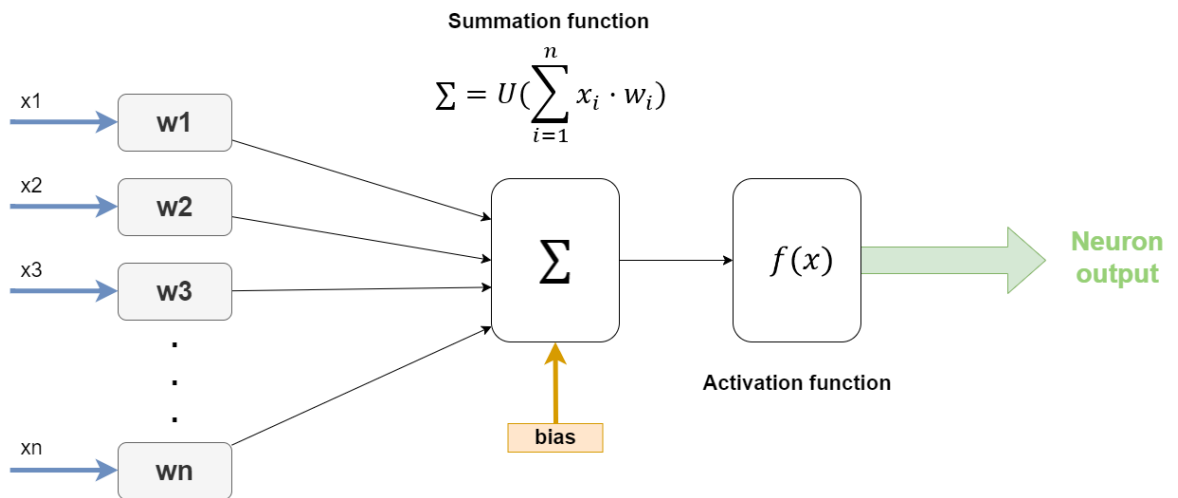


Figure 9 - Neuron function. Flow diagram of the process and functionality of a neuron or node from a ANN [own source]

The combined behavior of multiple neurons in the network, along with their connections and weights, allows an ANN to learn and make predictions based on the given input data. The training process aims to adjust the neuron's weights and biases to minimize the overall error of the network and improve its performance on specific tasks, such as classification or regression.

Neural networks are commonly structured into layers consisting of interconnected neurons. Specifically, layers positioned between the input and output layers are referred to as hidden layers, as their outputs are not directly observed or accessible.

The following figure shows the basic functionality of an ANN [36]:

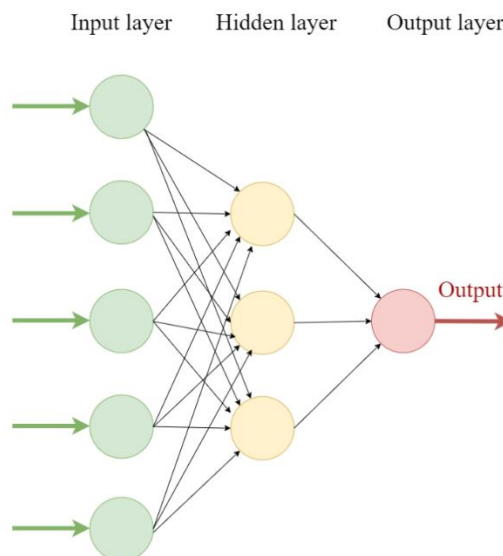


Figure 10 - Basic functionality of an ANN. Structure of an ANN, three main parts: (1) Input layer, (2) Hidden layer and (3) Output layer [own source]

Firstly, the input in the form of a multifimensional vector would be loaded to the input layer. Each neuron has an associated weight and threshold value and is connected to another neuron. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

So, this first information would be distributed to the hidden layers, the layers in between. These layers will then make decisions from the previous layer and weigh up how a stochastic change within itself detracts or improves the final output. This process is what we refer as learning.

Having multiple hidden layers stacked upon each-other is commonly called deep learning.

A neural network can have different architectures modifying the types of neurons and adjusting the number of layers.

Training a neural network involves modifying the weights in order to transform the given input into the desired output. In order to alter the behavior of the input/output relationship, it is necessary to make adjustments to the weights. So, concepts like optimizers and loss function must be introduced to truly understand this.

The **optimizers** you use in a ANN determine how you should adjust the weights or learning rates of your neural network in order to decrease the losses. Optimizer algorithms are methods of optimization that have a significantly impact both the accuracy and speed of training the deep learning model [37].

During the training of a deep learning model, optimizers modify the weights of each epoch and aim to minimize the loss function. An optimizer serves as a function or algorithm that fine-tunes the attributes of the neural network, such as weights and learning rates. By doing so, it effectively reduces the overall loss and improves accuracy. Selecting the appropriate weights for a model can be a daunting task, given that deep learning models typically consist of millions of parameters. Therefore, it becomes essential to choose a suitable optimization algorithm for your specific application. Consequently, comprehending these machine learning algorithms is crucial for data scientists before delving deeper into the field [38] [39]. Some examples of optimizers used in ANNs are: Adam, Adamax, RMSprop, SGD (Stochastic Gradient Descent) among others.

In an ANN, a **loss function** is a measure of how well the network is performing on a given task. It quantifies the difference between the predicted output of the network and the actual or expected output. The goal of training an ANN is to minimize this loss function, indicating that the network is making accurate predictions.

As mentioned before, optimizers play a crucial role in minimizing the loss function of an ANN. The optimizer algorithm calculates the gradients of the loss function with respect to the network parameters and adjusts them accordingly [40].

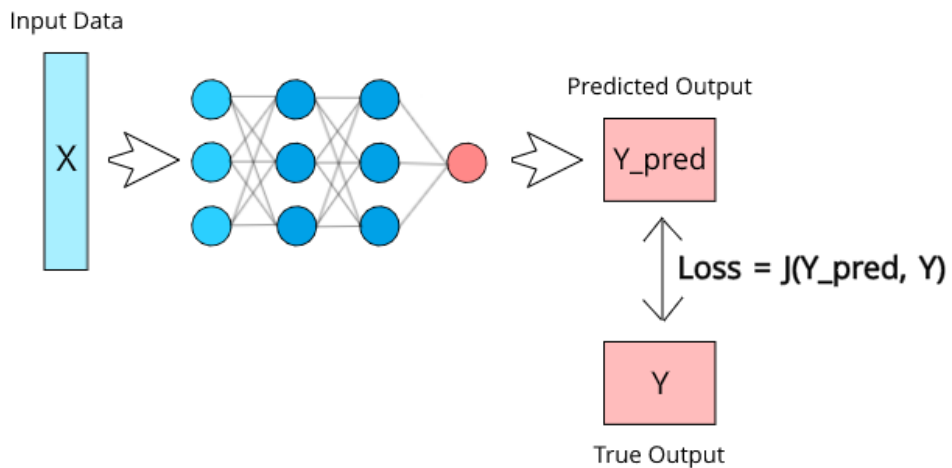


Figure 11 - Representation of the loss function of a basic ANN (called J in this diagram). J is a function defined by two parameters: The predicted output and the true output.[48]

The relationship between loss functions and optimizers lies in their combined effort to optimize the performance of the neural network. The loss function provides a measure of the network's performance, while the optimizer algorithm determines how to update the network's parameters to minimize this loss. By iteratively adjusting the weights and biases through optimization, the network gradually improves its ability to make accurate predictions, ultimately achieving better performance on the given task.

2.6. Convolutional Neural Networks (CNNs)

The rapid growth of deep learning models in the medical field has been driven by advancements in machine learning, graphics processing technologies, and the availability of medical imaging data. Convolutional neural networks (CNNs) are being used by medical imaging professionals to aid clinicians in diagnosing diseases. In deep learning, CNNs are similar to the traditional ANNs architectures. CNNs use a mathematical operation, called convolution, in place of general matrix multiplication in at least one of their layers [42].

These type of algorithms are specifically used in patron and image recognition. There are many well known applications for CNNs such as video regonition, image classification, image segmentation, medical image analysis, natural language processing, and others.

Through automated learning, CNNs learn how to optimize the filters (or kernels) and that is why their preprocessing process is smaller compared to other image classification algorithms. Other tradiotional systems depend on human supervision.

CNNs generate network features by convolving kernels in a layer with outputs of previous layers. Early hidden layers in a CNN capture simple features such as shapes and edges, while deeper layers capture more complex features [41].

Depending on the target organ medical images have varying characteristics and suspected diagnoses. Three-dimensional CNNs have been recently employed to analyze medical images such as X-ray, computed tomography (CT) scans, diffusion tensor imaging (DTI), positron emission tomography (PET), magnetic resonance imaging (MRI), and functional MRI (fMRI). While improvements in radiological imaging technology have made image acquisition faster, of higher quality and more affordable, trained human radiologists are still necessary to analyze and diagnose the images. However, due to the increasing number of radiological images and the limitations faced by radiologists in terms of time, experience, and fatigue, automated machine learning algorithms can provide vital assistance to clinicians.

Deep learning involves using neural networks containing multiple interconnected layers of artificial neurons to identify patterns in data samples. Different activation functions, such as hyperbolic tangent (tanh), the sigmoid transformation and the rectified linear unit (ReLU) and their variants are used in CNN architectures.

The development of deep learning can be traced back to Walter Pitts and Warren McCulloch in 1943, and since then, significant advancements have been made, such as the backpropagation model, CNN, LSTM, ImageNet, AlexNet, and GoogLeNet [43].

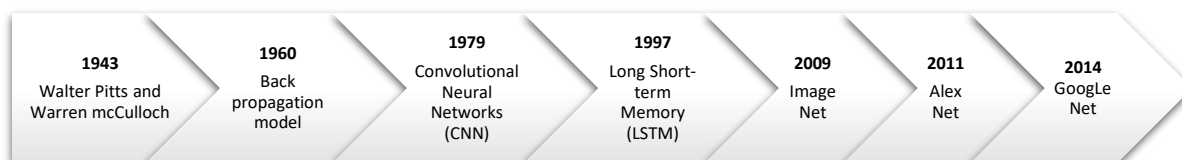


Figure 12 - Evolution of ANN through time, from 1943 to 2014. [own source]

In 2014, Google presented GoogLeNet (Winner of ILSVRC 2014 challenge), which introduced the concept of inception modules that drastically reduced the computational complexity of CNN.

However, the application of 3D deep learning on medical images is limited by the scarcity of data and high computational costs, as well as the curse of dimensionality.

Although there are certain limitations, advancements in neural network architectures, data augmentation methods, and powerful GPUs have made it progressively more viable to analyze volumetric medical data through 3D deep learning. Consequently, the utilization of 3D deep learning in diverse medical imaging modalities has witnessed a substantial surge since 2012 [41].

2.6.1. Overall architecture

The mathematical approach of 3D-CNN is very similar that of 2D-CNN, differing only in the added dimension.

The basic architecture of Deep CNN consists of an input layer, an output layer, and one or more hidden layers. A hidden layer is a combination of a convolution layer, a pooling layer, a normalization layer, and a fully connected layer.

A 3D-CNN simultaneously extracts both spectral and spatial features from an input volume. This difference makes use of CNN in 1D and 2D. This advantage also makes it a very useful tool for analyzing volumetric data in medical imaging.

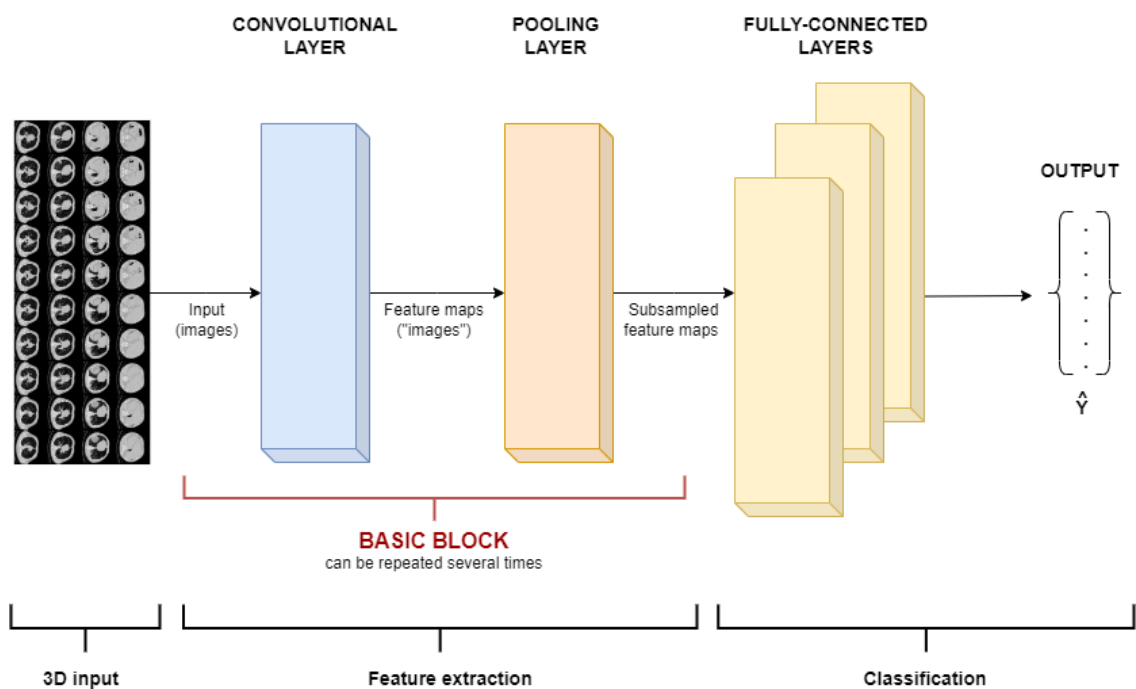


Figure 13 - CNN structure. The functionality of a basic CNN can be down into four key components: (1) local receptive field, (2) sharing weights, (3) pooling and (4) fully connected (fc) layers. [own source]

2.6.1.1. Local receptive field

Is a defined segmented area that consists on the input layer that will hold the pixel or voxel values of the image. This layer is occupied by the content of input data that a neuron within a convolutional layer is exposed to during the process of convolution [44].

2.6.1.2. Convolutional layer

The convolutional layer plays a major role in how the CNN operates, that's why it gives the name to the neural network. Mathematically, a convolutional network differs from a regular network in the basis

that the neurons in its layers are arranged in a three dimension disposition (width, height, and depth) [45].

This layer will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume.

The size of the output of a CNN depends on **filter** (kernel) size, **depth**, **stride** and **zero-padding**. The optimisation of these values, often called hyperparameters, can significantly reduce the complexity of the model.

In a convolutional neural network (CNN), the **kernel** acts as a filter that is employed to extract features from images. The network's layer parameters primarily revolve around the utilization of learnable kernels. These kernels typically possess a small spatial dimensionality, as depicted in the accompanying figure, but extend across the entire depth of the input.

The result of applying a filter is a feature map. When the data passes through a convolutional layer, the layer applies convolution to each filter across the spatial dimension of the input, resulting in the generation of a 2D feature map. As the network progresses through the input, a scalar product is computed for each value within the kernel. Through this process, the network learns kernels that become activated when they detect a specific feature at a particular spatial position in the input. These activations are commonly referred to as activations or feature activations.

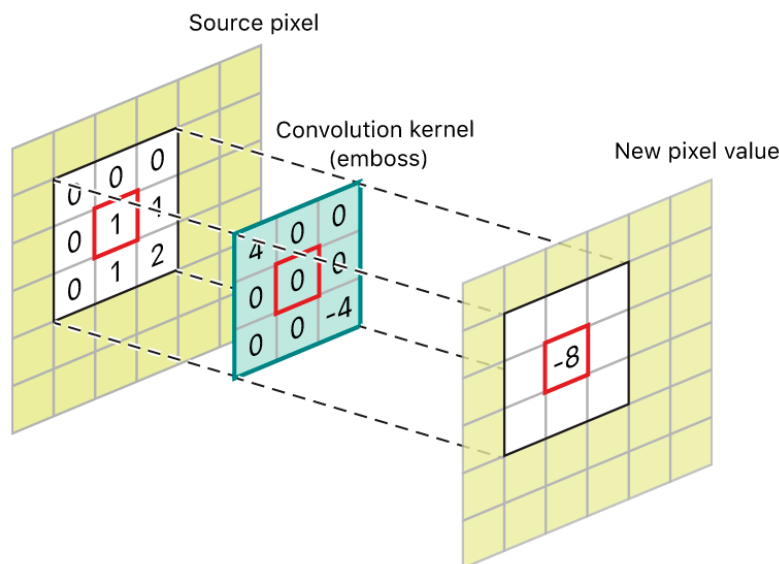


Figure 14 - Convolution kernel [46]

Each kernel in the convolutional layer will generate a corresponding activation map. These activation maps are then stacked along the depth dimension to create the complete output volume of the

convolutional layer. It is worth noting that the level of connectivity across the depth dimension is typically equal to the depth of the input volume.

One way to manipulate the **depth** of the output volume generated by the convolutional layers is by adjusting the number of neurons within the layer to match a specific region of the input. However, it is important to note that reducing the depth of the output volume can lead to a significant decrease in the model's ability to recognize patterns.

The **stride**, on the other hand, represents the number of cells by which the filter is shifted. Setting the stride to a greater number will reduce the amount of overlapping and produce an output of lower spatial dimensions.

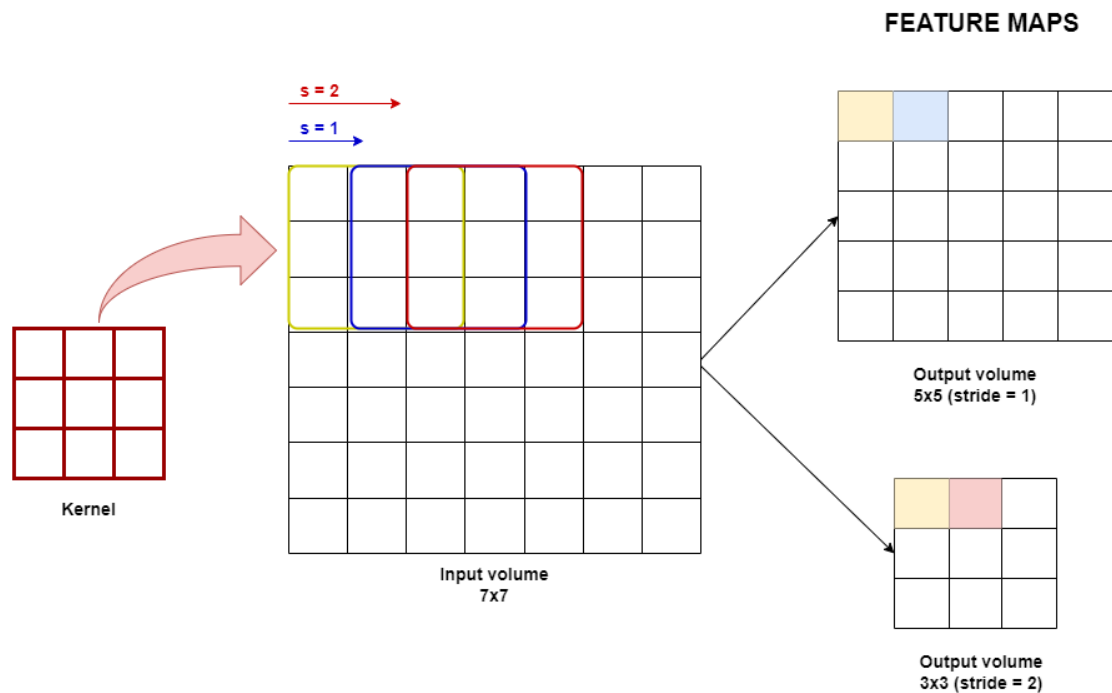


Figure 15 - Representation of stride in ANNs [own source]

Lastly, **zero-padding** consists in adding zeros all around the tensor (input). This method is used to prevent the decreasing of the data and is an effective method to give further control as to the dimensionality of the output volumes.

$$O = \frac{(n - f)}{s} + 1 \quad ; \quad O = \frac{(n + 2p - f)}{s} + 1$$

n = input tensor size

f = filter (kernel) size

p = padding

s = stride

The rectified linear unit (commonly shortened to ReLu) aims to apply an 'elementwise' activation function such as sigmoid to the output of the activation produced by the previous layer [42].

2.6.1.3. Pooling layer

This layer is added after the convolutional layer or layers. The pooling layer will then simply perform downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation. The feature maps from the convolutional layers are downsampled by the pooling layer giving a new map with a reduced size.

For that reason, it reduces the number of parameters and the computational complexity of the model.

In the majority of CNNs, these operations are typically implemented using max-pooling layers. These layers employ kernels of size 2×2 with a stride of 2 along the spatial dimensions of the input. As a result, the activation map is reduced to 25% of its original size while retaining the depth volume at its original dimensions.

Due to the destructive nature of pooling, having a kernel size above 3 will usually greatly decrease the performance of the model.

2.6.1.4. Fully-connected layers

These layers will then perform the same duties found in standard ANNs, attempting to produce class scores from the activations, to be used for classification. The fully-connected layer contains neurons of which are directly connected to the neurons in the two adjacent layers, without being connected to any layers within them. It is also suggested that ReLU may be used between these layers, as to improve performance.

2.6.2. Activation functions

Activation functions play a crucial role in ANNs by transforming linear functions into nonlinear operations. This allows ANNs to effectively address more intricate problems, including nonlinear classification tasks. Several commonly used activation functions include sigmoid, tanh, and rectified linear units (ReLU).

Out of these options, the ReLU function has gained significant popularity due to its demonstrated performance, which is comparable or even superior to sigmoid and tanh functions. Notably, the ReLU function not only mitigates the issue of vanishing gradients but also enhances computational efficiency by optimizing time and space complexity. For neural networks that use images as a dataset, ReLU is the most common activation function.

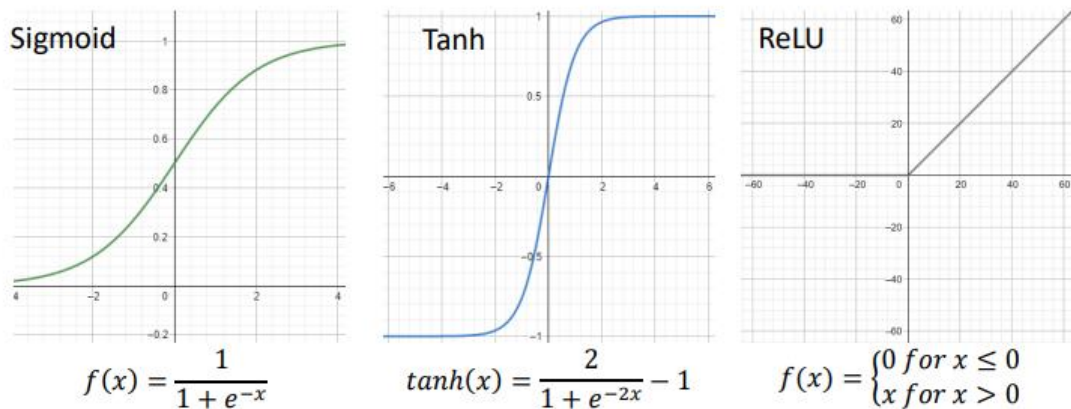


Figure 16 - Common activation functions [47]

It's important to note that activation functions are not limited to these options mentioned. There are other activation functions such as Leaky ReLU, exponential linear units (ELU), and softmax, each designed to address specific requirements of different tasks and neural network architectures. The choice of activation function depends on the specific problem being addressed and the characteristics of the data. Researchers and practitioners continually explore and develop new activation functions to further enhance the capabilities of ANNs [47].

2.6.3. Hyperparameters

Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up with. So, the performance of every machine learning model depends on these values. A few examples of main hyperparameters are: batch size, number of epochs, activation function, dense units, learning rate...

An effective set of hyperparameters should, at the very least, minimize the validation error. Hyperparameter optimization presents a unique challenge as it is a nested optimization problem in machine learning, wherein each iteration involves training and validating a machine learning model.

Therefore, in the absence of an alternative automation method, hyperparameters need to be manually set through a trial-and-error approach, making it a time-consuming and arduous component of the machine learning workflow [49].

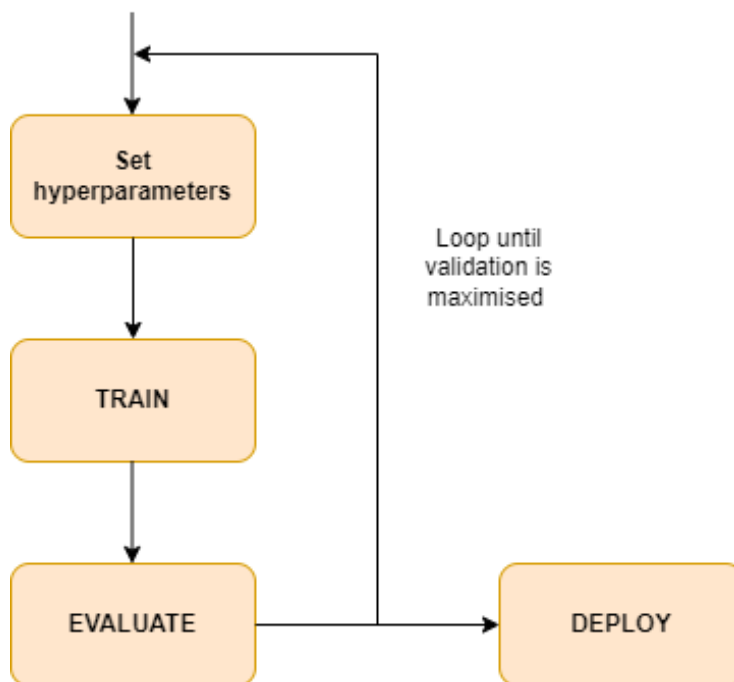


Figure 17 - Basic workflow in machine learning for the selection of hyperparameters [own source].

Several heuristic approaches, such as a grid search, randomised search, Bayesian optimisation, and gradient-based optimisation, have been developed for hyperparameter optimisation. The drawback of the above methods is that they are less efficient in a high-dimensional space because the number of evaluations increases exponentially as the number of hyperparameters increases [51].

Grid search is a technique used to optimize hyperparameters, involving an exhaustive exploration of a manually specified subset of the hyperparameter space for a targeted algorithm. Specifically, when working with a given model like a 3D-CNN and a test dataset, grid search helps identify the optimal combination of hyperparameters (such as the learning rate of the optimizer). This method offers several advantages, including an exhaustive search that determines the best way to fine-tune hyperparameters based on the training set. However, it also has drawbacks, notably being time-consuming and posing a risk of overfitting [50].

After careful consideration, we decided to employ the **grid search** method to determine the optimal hyperparameters for our 3D-CNN in this particular project.

2.7. Case of study

After conducting a theoretical review of the basic medical concepts around which this work is focused, it is necessary to perform a literature review to investigate the projects and research that have been developed in recent years. The impact of COPD on society currently reaches alarming levels,

significantly reducing the quality of life for many individuals. Furthermore, due to its heterogeneous nature and unique characteristics for each patient, traditional detection methods prove to be less accurate for a proper preventive diagnosis. The most suitable solution to this situation is the implementation of Deep learning algorithms, such as neural networks, for the detection of this disease and others with similar characteristics.

In the following section, a state of the art overview is presented, chronologically organized, highlighting some of the recent works in this field.

3. State of the art

During the last years, different approaches have been proposed to deal with the classification of CT scan images. These articles were chosen because they rely on the same type of diseases, respiratory diseases, such as COVID-19, lung cancer, emphysema abnormalities and COPD. Hence, these works exploit 3D-CNN classification for chest CT scans.

The following table (**Table 2**), gathers the previous works that have addressed classifying patients with COVID-19 and COPD using different 3D-CNN architectures.

Table 3 - Overview of the 3D-CNN literature

<i>Reference</i>	<i>Methodology</i>	<i>Imaging Modality</i>	<i>Dataset</i>	<i>Objective</i>	<i>Classification Model</i>	<i>Performance Measures</i>
<i>Xu et al. [16]</i> <i>(2022, January)</i>	Deep learning	CT	A total of 618 CT simples were collected: 219 samples from 110 patients with COVID-19 and 224 samples from 224 patients with influenza-A viral pneumonia (IAVP).	Early screening model to distinguish COVID-19 from IAVP and healthy cases through pulmonary CT images.	3D-CNN	Accuracy: 86,7%
<i>Ho et al. [17]</i> <i>(2021, January)</i>	Deep learning	CT	596 subjects (204 COPD, 392 non-COPD) from KNU and JNU Hospitals.	Introduction of a new classification method for COPD based on deep learning and a parametric-response mapping (PRM) method.	The integrated 3D-CNN and PRM (3D-cPRM)	Accuracy: 89.3%; Sensitivity: 88.3%, in five-fold cross-validation.
<i>Nguyen et al. [18]</i> <i>(2021, June)</i>	Deep learning	CT	1. 101 patients with COVID-19, 118 with pneumonia and 118 with non-pneumonia abnormalities.	Normal, COVID-19, and pneumonia classification.	3D-CNN	Accuracy: 1. 87%; 2. 97%; 3. 86% ROC curve: 1. 0,83; 2. 0,99;

			2. 1544 patients with COVID-19, 1556 with pneumonia and 118 with non-pneumonia abnormalities.			3. 0,87

			3. 281 patients with COVID-19, 1068 with non-pneumonia abnormalities.			
Ahmed et al. [19] (2020, January)	Deep learning	CT (DICOM format)	CT images from two data sets called SI-I and SI-II. SI-II data set consists of 1000 patients with at least 2 scans per patient.	A proposed variant of 3D VoxResNet for COPD and emphysema classification	3D VoxResNet	Accuracy: 74,3%; AUC: 0,820
Ko et al. [20] (2020, June)	Deep learning	CT	1194 chest CT COVID-19 images and 1357 chest CT images with non-COVID-19 pneumonia.	Classification of COVID-19 patients.	FCONet developed by transfer learning using one of four state-of-the-art pretrained deep learning models (VGG16, ResNet-50, Inception-v3, or Xception).	Accuracy: 99,87 % Specificity: 100 %
Polat et al. [21] (2019, March)	Deep learning	CT (DICOM format)	CT scan images of Data Science Bowl and Kaggle. This dataset consists of 2101 high-risk patients with CT scans of chest cavity which includes multiple 2D slices (Figure 3) in an Imaging and Communications in Medicine (DICOM) format.	To investigate the performance of the two proposed models.	Straight 3D-CNN with conventional softmax and hybrid 3D-CNN with Radial Basis Function (RBF)-based SVM, the altered models of two-well known CNN architectures (3D-AlexNet and 3D-GoogleNet)	Straight 3D-CNN: Sensitivity: 86,40%, Precision: 90,37%, Specificity: 93,09%, Accuracy: 90,23%. Hybrid 3D-CNN:

					were considered.	Sensitivity: 88,53%, Precision: 91,91%. Specificity: 94,23%, Accuracy: 91,81%.
González et al. [22] (2018, January)	Deep learning	CT	Genetic Epidemiology of COPD(COPDGene) dataset and ECLIPSE dataset	CNN analysis, for detection and stage of COPD and prediction of respiratory disease (ARD) events and mortality in smokers.	CNN	AUC: 0,856

Since large image datasets are difficult to collect and deep learning methods are starting to take part on the diagnosis of diseases by analysing medical images, there are some researches on the detection of lung diseases. As we just have seen in **Table 2**, there is a wide variety of studies in the field of CT-scan classification on COVID-19, lung cancer and precisely on COPD classification using different CNN architectures.

Some of these articles rely on the diagnose of COVID-19 patients, such as in **Xu et al.** where cases of coronavirus disease were segregated based on CT images. Such assessments are crucial for estimating intensive care unit events and making clinical decisions regarding treatment planning. To improve the performance and generalization of the classification model, the authors propose a novel domain adaptation (DA) method that addresses several challenges. This method consists of a stochastic class-balanced boosting sampling strategy and representation learning with domain-transferability, discriminant properties, and completeness. The experiments conducted on COVID-19 severity assessment show the effectiveness of the proposed method in overcoming challenges associated with imbalanced learning and domain distribution discrepancy. The performance measurements for this research is a great accuracy value (86,7%) [16].

In the article **Nguyen et al.**, nine identical DL-based classification models are trained using different combinations of datasets. The models trained on individual datasets achieve accuracy and area under the receiver operating characteristic curve (AUC) values specific to their respective datasets. This study highlights the challenge of generalizability in artificial intelligence technologies used for identifying and classifying COVID-19 patients. The performance of deep learning models trained on specific datasets tends to decrease significantly when evaluated on datasets from different countries, indicating the presence of heterogeneous factors. Factors such as patient demographics and differences in image

acquisition or reconstruction contribute to this problem, emphasizing the need for robust and generalized AI models in the context of COVID-19 diagnosis and classification. The datasets used include one from UT Southwestern (UTSW) and three external datasets from China, Iran, and Russia [18].

In **Ko et al.** a simple 2D deep learning framework called the fast-track COVID-19 classification network (FCONet) was studied to diagnose COVID-19 pneumonia using a single chest CT image. FCONet utilizes transfer learning with four state-of-the-art pretrained deep learning models (VGG16, ResNet-50, Inception-v3, and Xception) as backbones. Among these models, ResNet-50 demonstrates outstanding diagnostic performance in the testing dataset, with a sensitivity of 99.58%, specificity of 100.00%, and accuracy of 99.87%. On the other hand, FCONet, a straightforward 2D deep learning framework based on a single chest CT image, exhibits excellent diagnostic performance in detecting COVID-19 pneumonia. The ResNet-50 model proves to be the most effective among the pretrained models evaluated in this study [20].

Because early diagnosis in lung cancer is really important, several studies using AI have been developed. That's the main reason why it was important to include at least one study that analysed lung cancer CT scan image patients. **Polat et al.** highlights the potential of deep learning methods for diagnosing lung cancer on CT images. The proposed CNN-based models, particularly the hybrid 3D-CNN with SVM, exhibit superior performance compared to traditional CNN architectures, emphasizing the effectiveness of automatic feature extraction in lung cancer diagnosis. Experimental results demonstrate that the performance of the proposed models exceeds that of 3D-AlexNet and 3D-GoogleNet. Specifically, the hybrid 3D-CNN with Support Vector Machine (SVM) achieves the most satisfying results, with accuracy rate, sensitivity, and precision of 91.81%, 88.53%, and 91.91%, respectively, in the diagnosis of lung cancer. These results outperform the straight 3D-CNN with softmax [21].

In other cases, for obvious reasons, COPD research papers that use Deep learning for CT image classification were explored.

Ho et al. study focuses on COPD. In this paper, the authors propose a new classification method for grouping COPD based on deep learning and a parametric-response mapping (PRM) technique.

The authors use a 3D convolutional neural network (CNN) to classify COPD based on input parameters extracted through an image registration technique. Specifically, they extract functional variables related to functional small airway disease percentage (fSAD%) and emphysema percentage (Emph%). These variables are provided as input to the 3D-CNN combined with PRM (referred to as 3D-cPRM). The classification accuracy of the 3D-cPRM model is 89.3% with a sensitivity of 88.3% in five-fold cross-validation. Comparative analysis reveals that the proposed 3D-cPRM model outperforms both 2D

models and traditional 3D CNNs using the same neural network. Overall, the 3D-cPRM model successfully represents the parenchymal abnormalities associated with COPD and aligns with CT-based COPD diagnoses. The integration of deep learning and PRM techniques offers an effective approach for classifying COPD and provides insights into the regions of lung affected by the disease [17].

Ahmed et al. introduces a variant of the 3D VoxResNet model for the classification of Chronic Obstructive Pulmonary Disease (COPD) and emphysema. The model leverages volume-wise annotations without additional feature enhancement or inclusion of meta-data. For COPD classification, the network achieves similar results as other approaches. However, for emphysema classification, the authors fine-tune the COPD classification network, resulting in a significant improvement in model performance. As future work, the authors suggest validating the model on larger and balanced datasets. They also recommend conducting a thorough comparison with other methods, preferably using k-fold cross-validation, to further assess the model's effectiveness and compare its performance against existing approaches [19].

González et al. aimed to investigate the potential of deep learning, specifically CNN analysis, in detecting and staging COPD and predicting acute respiratory disease (ARD) events and mortality in smokers. For the diagnosis of COPD and prediction of ARD events, logistic regression was employed. In ECLIPSE, 29.4% were accurately staged and 74.6% were within one stage. The findings of this study suggest that CNN analysis through deep learning holds promise in detecting and staging COPD, as well as predicting ARD events and mortality in smokers. The specific performance metrics and statistical tests used indicate the effectiveness of the CNN model in these tasks [22].

This last article was chosen because the dataset of ECLIPSE is the same one used in this project. It is important to remark that ECLIPSE data has been used in a number of genetic studies and also, as this last article shows, there are a few studies for the classification of disease staging and prognosis in smokers using a CNN. However, they mostly rely on simple CNNs.

Therefore, what makes this project unique is the application of a 3D Convolutional Neural Network on this specific dataset. And the main goal is not only trying to achieve the best accuracy for the classification, it is also to optimize the algorithm and reduce the computational cost of the process.

After this project, a 3D-CNN architecture implemented on Python and COVID-19 CT scans images will be readapted and improved for COPD patients.

The strategies and the methodology used in this project will be explained in the following sections (mainly Section 4.).

4. Materials and methods

4.1. Database

As mentioned in the origin section of this project the database used in this project comes from observational study ECLIPSE. The release date of this study is from July 25, 2017.

Consent group	Is IRB required?	Data Access Committee	Number of participants
Disease-Specific (Chronic Obstructive Pulmonary Disease, RD) 🔗	No	National Heart, Lung, and Blood Institute DAC (nhlbigenicdata@nhlbi.nih.gov)	2746

Table 4 - ECLIPSE dataset, consent group: COPD patients [1]

The ECLIPSE dataset is provided by the BRGE research group at ISGlobal, led by Juan Ramón González, who is not only one of the leaders of this research team but also the co-director of this final bachelor thesis.

As mentioned at the beginning of this project the Evaluation of COPD Longitudinally to Identify Predictive Surrogate Endpoints (also known as the ECLIPSE study), which involved **2164 individuals with COPD** along with a smaller group of **smoking controls (337)** and **nonsmoking controls (245)**, aimed to assess and predict outcomes over a three-year period. Regular observations were conducted, and three CT scans were performed at baseline, one year, and three years. These scans were analyzed using VIDA software to identify emphysema and airway phenotypes.

The study's inclusion criteria encompassed individuals between the ages of 40 and 75, who had a smoking history of at least 10 pack-years, and exhibited spirometry results corresponding to GOLD grades 2-4 for COPD cases or normal spirometry results with post-bronchodilator FEV1 >85% predicted and FEV1/FVC >0.7 for controls. Study visits were conducted upon enrollment, at three months, and every six months thereafter, involving spirometry tests, questionnaires, and other clinical assessments. The VIDA software was utilized to analyze the ECLIPSE CT scans, specifically focusing on identifying emphysema and airway phenotypes. The ECLIPSE study has yielded significant insights into the clinical epidemiology of COPD, such as exacerbations of COPD (*Hurst, NEJM 2010; 363: 1128*) and the decline of lung function in COPD (*Vestbo, NEJM 2011; 365: 1184*). Additionally, ECLIPSE has been employed in various genetic studies examining COPD susceptibility and protein biomarkers [1].

There are 96 variables related to the ECLIPSE dataset.

The dataset description is:

Subject ID, age, race, sex, airway, BORG, BORG SOB assessment, BMI, body water, COPD, CV, distance walked, exhaled carbon monoxide, fat free mass, forced expiratory volume, FEV/FVC, FACIT, functional residual capacity, forced vital capacity, GOLD stage, inspiratory capacity, impedance, low frequency, lung capacity, MRC dyspnoea score, MNR, percent fat, prognostic index, predicted normal RV, percent oxygen in blood, resonant frequency, respiration system, residual volume, specific conductance, SGRQ-C, subject group, TLC, tidal volume, vital capacity, weight, and smoking status of participants with or without lung disease and involved in the "Evaluation of COPD Longitudinally to Identify Predictive Surrogate Endpoints (ECLIPSE)" project [1].

The following chart shows one of the 96 variables of this dataset: the GOLD stages of the participants in ECLIPSE study. For this project, that its main goal is to develop a 3D-CNN, it is interesting to be aware of the GOLD stages of our subjects. It is important to point at the fact that the majority of the participants that have COPD are found between Stage II and III (moderate and severe stages). Therefore, for Stage II, III and IV patient CT-scans will clearly show signs of COPD.

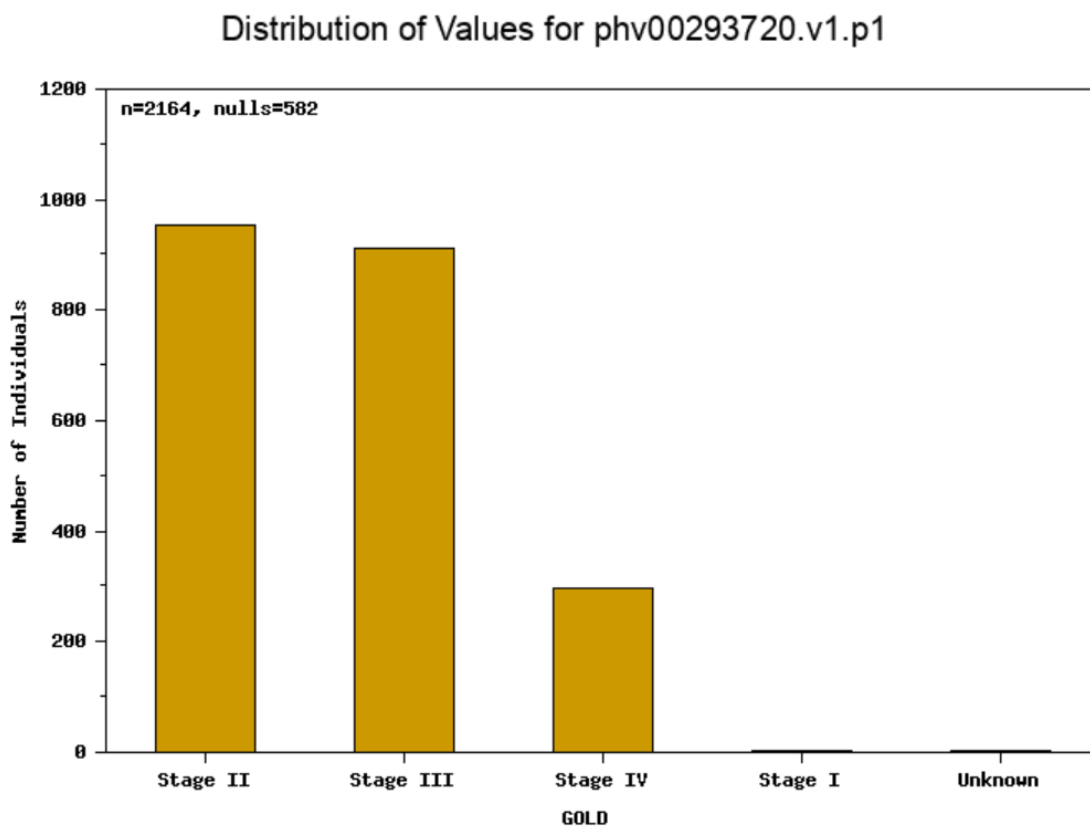


Figure 18 - Statistical summary of the ECLIPSE dataset, focusing on the variable GOLD [1]

The image files for this project are provided in NIfTI format with the extension .nii.



Figure 19 - Visualization montage of slices, 4 rows and 10 columns for 100 slices of a CT scan [own source]

The Neuroimaging Informatics Technology Initiative (NIfTI) is a widely used open file format designed to store brain imaging data acquired through Magnetic Resonance Imaging techniques [23]. Conversely, the DICOM image format (Digital Imaging and Communications in Medicine) is another method employed for storing medical images. The creation of the DICOM standard aimed to facilitate various processes in medical imaging and has contributed significantly to advancements in the field, particularly the digitalization of radiology. This standard allows for different resolutions and bit-depth allocations to be supported.

Typically, CT and MRI images are acquired and stored in the DICOM format. However, when working with 3D images, NIfTI files offer a superior alternative to DICOM files [24]. The primary distinction between DICOM and NIfTI lies in the representation of raw image data. NIfTI saves the data as a 3D image, while DICOM stores it as 2D image slices. This characteristic makes NIfTI more advantageous for certain machine learning applications since it is modeled as a 3D image [25].

As mentioned earlier, the images for this project are axial CT scans obtained from the patients.

In the ECLIPSE study, three CT scans were conducted, one each year throughout the study, resulting in a total of three NIfTI files per patient. For this project, we will be focusing on the first CT scan of each patient.

 [02314000001_INSP_STD_L1_ECLIPSE.nii.gz](#)

 [02314000002_INSP_STD_L1_ECLIPSE.nii.gz](#)


 [02314000003_INSP_STD_L1_ECLIPSE.nii.gz](#)

Figure 20 - Identification of each NIfTI image for each patient.

For this 3D-CNN implementation the dataset was build around the nonsmoking controls versus the individuals with COPD. Ensuring a balanced quantity of data for each class is crucial as an imbalanced dataset may lead the 3D-CNN model to favor the majority class. When implementing neural networks, balancing the dataset becomes essential to facilitate model training and prevent biases towards any specific class. By achieving a balanced distribution, the model is provided with a fair representation of each class, promoting more accurate and unbiased learning.

Hence, our dataset consisted on the following:

TYPE OF CT SCANS	NUMBER
CT SCANS FOR COPD PATIENTS	210
CT SCANS FOR NON-SMOKER CONTROL PATIENTS	214
	Total: 424 CT scans

Table 5 - Number of CT scans for COPD patients and for Non-smoker control patients

4.2. Requirements

To ensure the successful implementation of this project, we used the JupyterHub programming tools. However, due to the confidential nature of the ECLIPSE dataset, it was not feasible to freely access the data from external or personal computer devices. As a result, we needed to rely on a computer within the research group (BRGE, ISGlobal) to gain access to the dataset.

JupyterHub is an application designed to help you manage multiple-users sessions of interactive computing. JupyterHub provides connectivity that enables you to connect users with the infrastructure required for their sessions. JupyterHub enables remote access to JupyterLab as well as Jupyter Notebooks.

For this project, jupyter notebooks where used to code the 3D-CNN and implement the images from the ECLIPSE dataset.

Jupyter notebook is a free and open-source coding application that provides an interactive programming environment. It's a web application for creating and sharing computational documents and it offers a simple, streamlined, document-centric experience.

The implementation of the code used in this project has been written in Jupyter notebook using **Python**, a well-known programming language. It is a cross-platform, open-source (free) language that enables the development of all kinds of software. Currently, this language is used in various fields such as data analysis, data science, artificial intelligence, machine learning, and web development. For this reason, it is one of the most in-demand languages in the job market. Python was chosen as the main programming language for this project based on the fact that it one of the most common



Figure 21 - Jupyter and Python logos [26],[27]

Nowadays, Python programming language finds application in diverse domains including data analysis, data science, artificial intelligence, machine learning, and web development. As a result, it has become one of the highly sought-after languages in the job market.

The primary challenge associated with neural networks is their demanding computational requirements. Specifically, when working with image data, neural networks incur significant computational and memory expenses due to the multitude of variables and information contained within raw images. The computational cost measures the amount of resources utilized by the neural network during training or inference stages. Hence, for this project, a PowerEdge R640 is employed.

The **PowerEdge R640** server is designed to optimize workload performance and data center density. It consists in a dual-socket platform designed for high-density scale-out computing in data centers. It offers a perfect balance of density, performance, and scalability to optimize application performance and data center efficiency. With its fast performance and reduced latency, the PowerEdge R640 contributes to meeting project requirements.

The PowerEdge modular portfolio provides the flexibility to customize, deploy, and manage your infrastructure easily, resulting in lower operating costs [28].



Figure 22 - PowerEdge R640 [28]

Main features of the **PowerEdge R640** :

<p>CPU (Central Processing Unit)</p>	<p>2 x Intel Xeon Gold 6240 processors are utilized in this setup. Each processor operates at 2.6 GHz, with 18 cores and 36 threads, offering a 10.4 GT/s data transfer rate and a 24.75 MB cache. The processors support Turbo Boost technology and Hyper-Threading (HT), and have a power consumption of 150 W. Additionally, the system uses DDR4 memory running at 2933 MHz, providing a total of 36 cores and 72 threads.</p>
<p>RAM (Random-Access Memory)</p>	<p>The system is equipped with 4 x 64 GB RDIMM memory modules, operating at a speed of 3200 MHz. This results in a total memory capacity of 256 GB.</p>
<p>HDD (Hard Disk Drive)</p>	<p>The system includes 7 x 2.4 TB SAS hard drives operating at a speed of 12 Gb/s and spinning at 10,000 revolutions per minute (rpm). These hard drives are in a 2.5" form factor and are configured in a RAID setup. As a result, the total storage capacity of the system amounts to 8.8 TB.</p>

Table 6 - PowerEdge R640 Server main features

The scripts and the 3D-CNN implementation was done in a regular computer, specifically a Microsoft Windows 10 Pro. Although this is a great computer it can take quite some time to run along all the images from the dataset (more than 12 hours). So, by connecting remotely to this more powerful server (PowerEdge R640) we can run all the images from the database faster and easier.

4.3. 3D-CNN implementation

This section aims to provide a detailed explanation of the step-by-step implementation of a 3D-CNN for the classification of CT scans in thoracic patients with COPD. Within this section, we will explore the modifications and the work conducted over the past several months.

Creating an optimal 3D-CNN that produces accurate results may appear deceptively simple, but in reality, it is a challenging task. The computational processes involved are time-consuming, and obtaining reliable outcomes can be difficult.

To establish the fundamental coding structure for our neural network, extensive research has been conducted. As a crucial starting point, we employed a code example from September 23, 2020, which effectively implements a 3D-CNN for classifying patients with pneumonia based on CT scans [52].

In this particular example, a subset of the MosMedData called "Chest CT Scans with COVID-19 Related Findings" was utilized. This dataset comprises both lung CT scans displaying COVID-19 related findings and those without such findings.

The reason behind selecting this specific code example lies in its ability to handle images similar to those encountered in the ECLIPSE project, especially axial CT scans. By leveraging this code, we can effectively tackle the challenges posed by the ECLIPSE project's imaging data.

4.4. 3D-CNN structure

This section focuses on providing a step-by-step description of the process involved in developing the code for the 3D-CNN. It also covers the essential libraries required for importing the data, as well as various processes involved in classifying cases of COPD.

To begin, we assimilate the workflow of the neural network, gaining a comprehensive understanding of its structure and functioning. This understanding serves as a foundation for further analysis and optimization. By meticulously following this step-by-step process, we aim to develop a well-optimized and effective classification system for CT scans, specifically targeting COPD cases.

For the purposes of our implementation the required libraries to import are:

PYTHON LIBRARIES	DESCRIPTION
OS (OPERATING SYSTEM)	Provides easy functions that allow us to interact and get Operating System information and even control processes up to a limit.
KERAS	Is a high level, deep learning Application Programming Interfase (API) developed by Google written in Python language to implement neural networks.
TENSORFLOW	Helps to implement best practices for data automation, model tracking, performance monitoring, and model retraining. It is a library for fast numerical computing created and also released by Google.
NUMPY	It is used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices

MATPLOTLIB.PYPLOT	Is a collection of functions that make matplotlib work like MATLAB. It allows users to create numerous and diverse plot types.
--------------------------	--

Table 7 - Basic libraries implemented in the code.

The main steps for the 3D-CNN algorithm are:

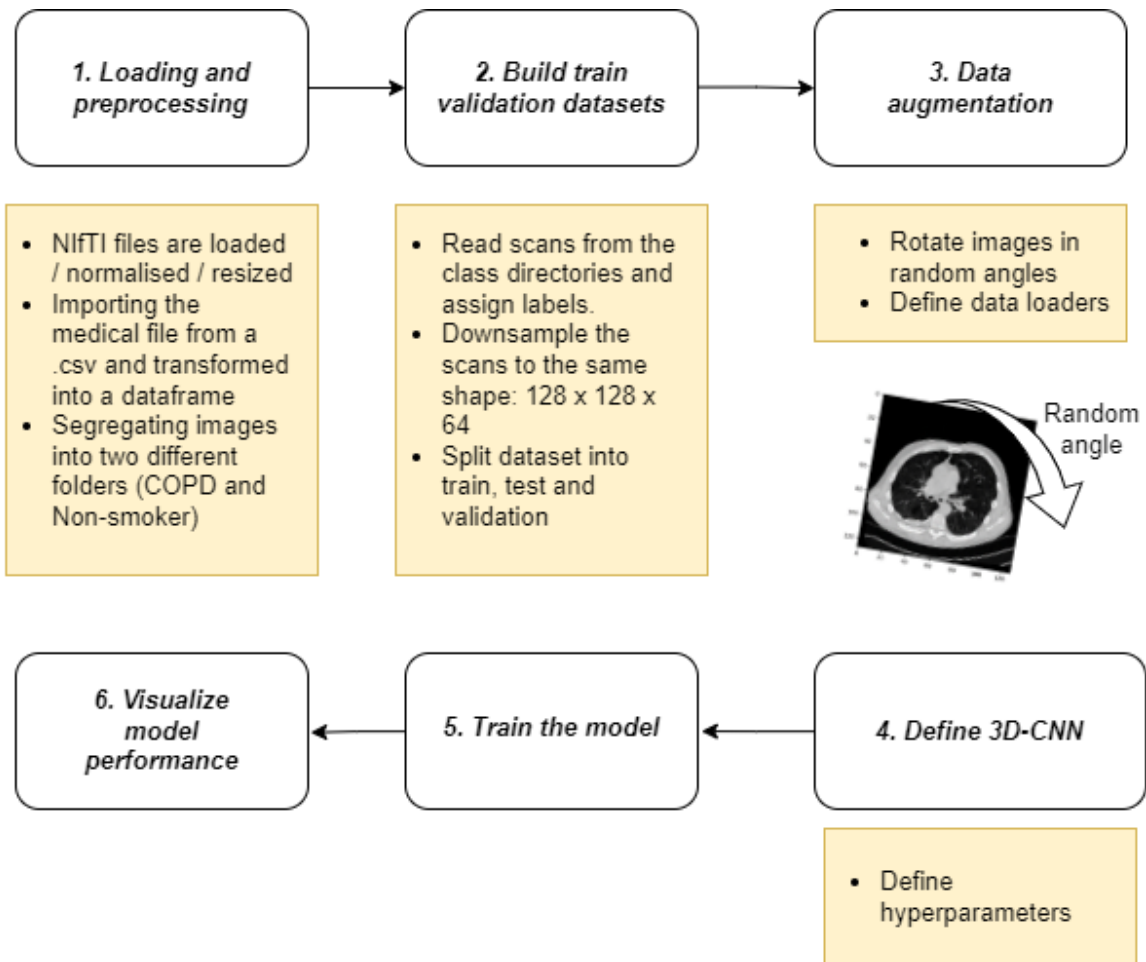


Figure 23 - Main steps for the 3D-CNN implemented algorithm in Jupyter Notebooks in Python [own source].

Next, on the following subsections, these steps are more well explained.

4.4.1. Loading data and preprocessing

The initial step in our workflow involves the loading of the data and CT images from our dataset into the JupyterHub environment. As previously mentioned, these files are stored in the NIfTI format,

denoted by the .nii extension. To facilitate the reading and processing of these scans, we rely on the nibabel package, which provides a robust framework for working with NIfTI files.

CT scans, as a fundamental component of our dataset, employ Hounsfield units (HU) to quantify voxel intensity. HU values play a crucial role in characterizing radiodensity, enabling radiologists and medical professionals to interpret images and diagnose diseases across various medical specialties. This quantifiable scale serves as a reliable reference for understanding tissue composition and pathologies within the scanned anatomical regions.

In our specific dataset, the HU values span a range from -1024 to values exceeding 2000. However, it is important to establish an upper bound to ensure effective normalization. Values above 400 typically correspond to bones with varying radiointensity, thus serving as a practical upper threshold. To achieve the normalization of CT scans, a commonly employed approach involves applying a threshold within the range of -1000 to 400. This range serves as a reference to rescale the HU values and facilitate consistent comparisons and analysis across the dataset.

To facilitate the data preprocessing procedures, we adhere to a systematic sequence of steps outlined below:

Firstly, **the volumes undergo an initial rotation of 90 degrees** to ensure a standardized orientation throughout the dataset. Subsequently, the HU (Hounsfield Unit) values pertaining to the scans are carefully scaled, constraining them within the uniform range of 0 to 1. This normalization process allows for consistent and meaningful comparisons among different scans. Following this, we proceed with the crucial task of resizing the width, height, and depth dimensions of the scans. To streamline this process, we have devised a set of auxiliary functions that serve as invaluable tools for data processing. These specialized functions will be leveraged extensively during the construction of training and validation datasets, ensuring their optimal efficacy.

Now, we will move forward by **extracting the file paths of the CT scans** from their corresponding class directories. Then, we rely on a text file (.txt) that is the medical file where it shows the essential information for classifying the scans into two categories. This text is converted into a dataframe that uses unique patient ID numbers to determine the classification of patients as either COPD or non-smoker controls.

Next, we will proceed to **segregate the images into two distinct folders**: one for COPD subjects and another for non-smoker controls. To differentiate between the two categories, we will assign the label 0 to COPD subjects and the label 1 to non-smoker controls.

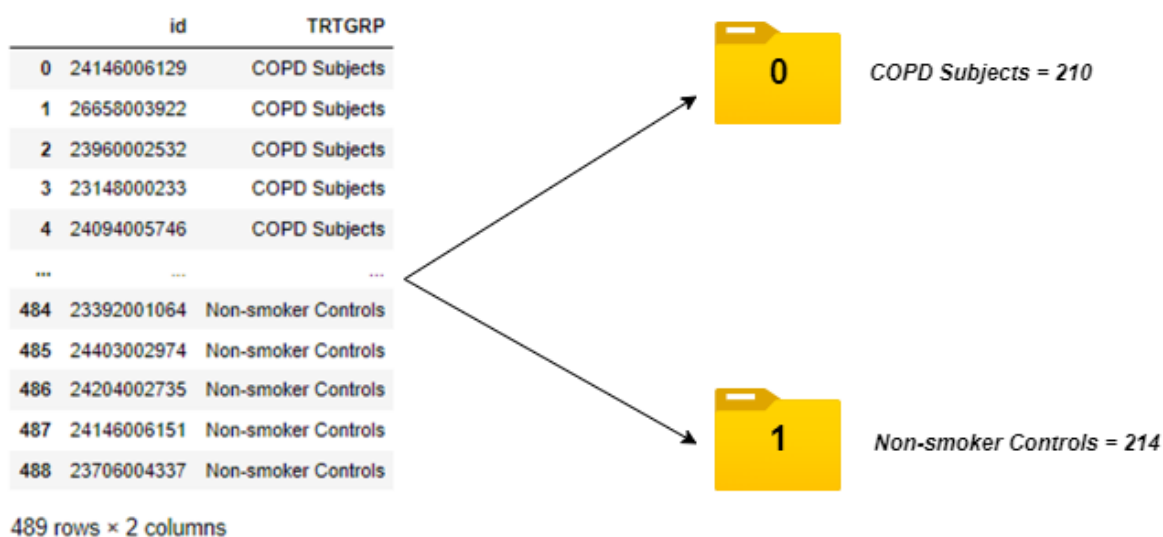


Figure 24 - Graphic representation on how data is segregated into two folders [own source].

4.4.2. Build train validation datasets

The data processing workflow encompasses several pivotal stages, commencing with the extraction of scans from their respective class directories, followed by the assignment of corresponding labels. Subsequently, the scans undergo downsampling, ensuring their conformity to a standardized shape of 128x128x64. To further enhance the data quality, a rescaling operation is performed on the raw HU values, effectively mapping them to a range spanning from 0 to 1.

As the process nears completion, the dataset is judiciously partitioned into two distinct subsets, namely the training dataset and the validation dataset. Later on, we coded the test dataset also. It is worth reiterating the distinguishing characteristics of these datasets:

The **training dataset** represents a sample of data utilized to fit the model. It serves as the foundation for the model's development and acquisition of the desired predictive capabilities.

Conversely, the **validation dataset** assumes the role of an impartial arbiter, facilitating an unbiased evaluation of the model's performance. This subset is instrumental in fine-tuning the model's hyperparameters, and its evaluation provides valuable insights. It is important to note that the model configuration may inadvertently incorporate certain biases if the validation dataset's characteristics significantly influence the final model.

Lastly, the **test dataset** assumes the paramount role of offering an unbiased evaluation of the final model. This dataset serves as the gold standard against which the model's performance is assessed, independent of the training process. The test dataset plays a pivotal role in determining the model's true efficacy and generalizability.

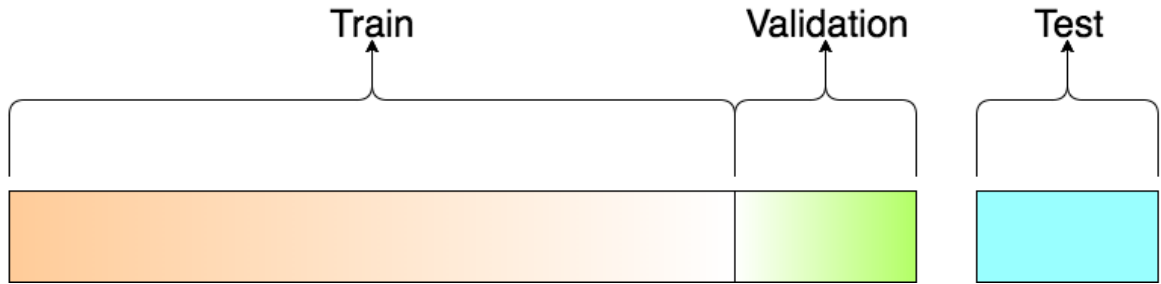


Figure 25 - Graphic representation of the train, validation and test dataset splits [53].

In the context of our 3D-CNN, the allocation of samples across the different datasets is as follows: **the training dataset** consists of **296 samples**, **the test dataset** comprises **43 samples**, and **the validation dataset** encompasses **85 samples**. These figures denote the number of instances available for training, evaluating, and validating our model's performance.

It is worth emphasizing that this particular phase, involving the allocation of samples to the respective datasets, is a time-intensive process. Given the significance of dataset composition in model training and evaluation, careful consideration and meticulousness are paramount. Consequently, the time spent on this task tends to be relatively extensive, as it necessitates meticulous attention to detail and precision to ensure optimal performance and accurate assessment of the model's capabilities.

4.4.3. Data augmentation

Throughout the training procedure, an essential aspect involves subjecting the CT scans to augmentation techniques. One such technique employed is the random rotation of the scans at diverse angles, thereby enhancing the diversity and robustness of the training data. The CT scans are stored as a rank-3 tensor, characterized by dimensions (samples, height, width, depth). To facilitate efficient 3D convolutions on the data, we introduce an additional dimension of size 1 at axis 4, thereby transforming the shape of the tensor to (samples, height, width, depth, 1).

It's worth noting the significance of preprocessing and augmentation techniques that are available to us. These techniques form the basis and starting point for further exploration and improvement. While we employed only one technique in this specific project, there are multiple others that could be implemented to increase the range of values examined by the 3D-CNN. By using these additional techniques, we can expand the possibilities for studying and enhancing the performance of our 3D-CNN model.

When constructing the train and validation data loaders, the training data is subjected to an augmentation function. This function performs random rotations on the volumes, introducing variations at various angles. It is noteworthy that both the training and validation data have already undergone rescaling, ensuring their respective values fall within the range of 0 to 1. This normalization

step guarantees consistent and meaningful comparisons during the subsequent training and evaluation phases.

Consequently, the dimensionality of the CT scan is as follows: (128, 128, 64, 1). This configuration, comprising a height of 128, width of 128, depth of 64, and a single channel, facilitates the effective utilization of the 3D-CNN model and accommodates the augmentation processes employed during the training phase.

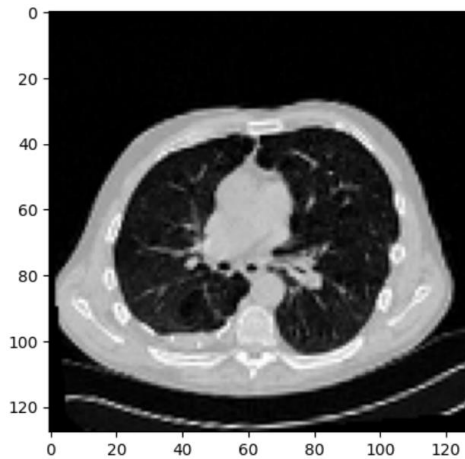


Figure 26 - Dimensionality of CT scan: (128, 128, 64, 1) [own source] [1]

4.4.4. Defining the 3D-CNN model

The process of defining a 3D-CNN model consists in making critical decisions regarding the architecture, including the number of layers, the quantity of dense units, and other essential hyperparameters. As mentioned before, these hyperparameters play a pivotal role in shaping the model's behavior and ultimately influencing its predictive capabilities.

The **basic structure of the 3D-CNN** models employed in this project consists of four layers:

4 LAYERS

WIDTH = 128, HEIGHT = 128, DEPTH = 64

LAYERS	Hyperparameters
CONVOLUTIONAL LAYER (C1)	Number of kernels: 64 Size of kernels: 3 x 3 Activation: relu
MAXPOOLING LAYER	Size of pooling: 2 x 2
CONVOLUTIONAL LAYER (C2)	Number of kernels: 64 Size of kernels: 3 x 3 Activation: relu

MAXPOOLING LAYER	Size of pooling: 2 x 2
CONVOLUTIONAL LAYER (C3)	Number of kernels: 128 Size of kernels: 3 x 3 Activation: relu
MAXPOOLING LAYER	Size of pooling: 2 x 2
CONVOLUTIONAL LAYER (C4)	Number of kernels: 256 Size of kernels: 3 x 3 Activation: relu
MAXPOOLING LAYER	Size of pooling: 2 x 2
FULLY CONNECTED LAYER	[256, 512] units Activation: relu
OUTPUT LAYER	Classification: 2 (COPD and Non-smoker subjects) Activation: sigmoid

Table 8 - Basic structure of the 3D CNN models employed in this project.

In addition, we conducted experiments with two-layered models to investigate the impact of model complexity on performance. These models shared the same structure as the four-layered 3D-CNN, except for the exclusion of the last two convolutional layers, namely C3 and C4.

By simplifying the model's architecture and removing unnecessary layers, we reduced the risk of overfitting and helped it better understand new and unseen data. This optimization strategy aimed to make the model more effective in real-world situations where the data may be different from what it was trained on. Overall, the focus was on enhancing the model's ability to capture important patterns and improve its performance on unseen data.

4.5. Work process

The work process I followed throughout this project can be visualized through the steps outlined in the accompanying figure. The figure provides a clear graphical representation of the sequential progression and interconnectedness of the various stages undertaken in this endeavor. By adhering to this structured framework, I ensured a systematic and organized approach to achieving the project objectives. Each step played a crucial role in advancing the project towards its ultimate goal, fostering clarity and coherence in the overall workflow.

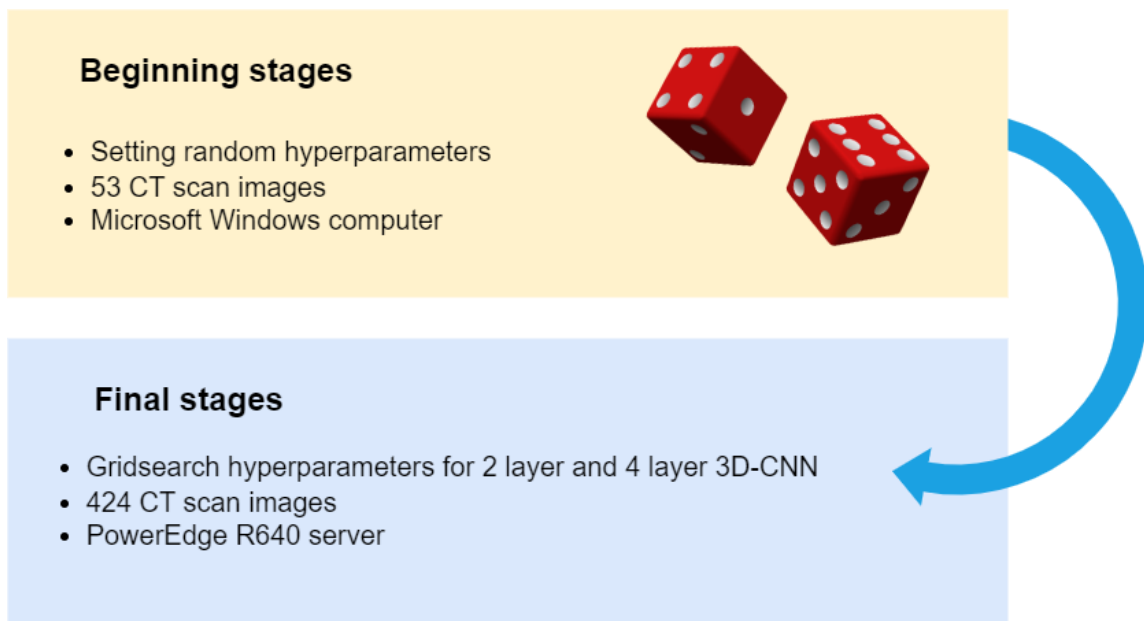


Figure 27 - Beginning and final stages of the work process [own source]

4.5.1. Beginning stages

In the initial phase of this project, I conducted hands-on experiments by manually adjusting different aspects of the CNN. I started with a limited dataset of only 53 images, which resulted in poor accuracy due to the insufficient training data. It was almost like flipping a coin. These initial trials consumed a significant amount of time as I developed the first 10 models on my work computer, a Microsoft Windows 10 Pro system, with each model taking approximately 30 minutes to compute.

The objective of this initial stage was to observe and analyze how the changes in values affected the behavior and results of the model. Through multiple rounds of trial and error, I discovered that these modifications had a notable impact on the predictive performance of the CNN. This realization emphasized the importance of carefully selecting appropriate hyperparameters for achieving optimal model performance.

As a result, I recognized the need to search for the best hyperparameters to improve the accuracy of my model. Consequently, in the subsequent steps, I implemented a method to identify and determine these hyperparameters.

4.5.2. Final stages

Having conducted initial experiments with random values, I determined it necessary to employ a search method to identify the optimal hyperparameters for my model. The chosen method for this task was

gridsearch, which constructs a model for every specified combination of hyperparameters and evaluates each model accordingly.

Moving on to the final stage of the project, I used a total of 424 CT scan images. Due to the increased data size, I required a more powerful server than my personal computer. While the computational process was faster on the server, the gridsearch method still entailed a considerable amount of time to complete.

It is important to highlight that the pursuit of optimal hyperparameters in CNN implementation remains a complex undertaking. The process encompasses various approaches and methodologies, with a central focus on identifying the most suitable hyperparameters. Extensive studies have been conducted to meticulously adjust and assess different parameters in order to determine the optimal configuration that achieves the desired performance.

The exploration of hyperparameters represents a crucial aspect of CNN implementation, necessitating thorough investigation and careful experimentation to identify the settings that yield the most favorable outcomes.

4.5.3. Hyperparameters

This subsection explains the process of selecting the hyperparameters for the 3D-CNN. Several hyperparameters have been predetermined from the outset, and this deliberate decision is not without justification.

4.5.3.1. Fixed hyperparameters

The learning rate and the activation function of the model were predetermined and fixed from the beginning of the experimentation process. The learning rate was set to 0.0001, while the activation function chosen was relu. The decision to settle on these specific values for the learning rate and activation function was driven by sound reasoning and considerations. Let's explore the reasons behind these choices:

1. **Learning rate:** The learning rate determines the step size at which the model adjusts its weights during the training process. A smaller learning rate ensures cautious and incremental weight updates, preventing overshooting and convergence issues. By setting a learning rate of 0.0001, we prioritize more conservative weight adjustments, aiming for a stable and controlled learning process.

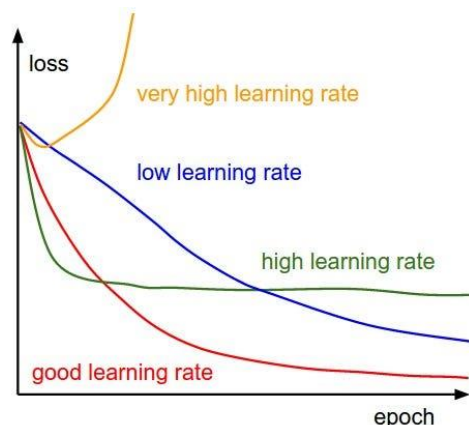


Figure 28 - How does the learning rate affect the CNN performance. [59]

2. **Activation function:** The activation function introduces non-linearity to the model, allowing it to learn complex patterns and relationships within the data. Relu (Rectified Linear Unit) is a commonly used activation function that offers simplicity and computational efficiency. It has been widely adopted in deep learning due to its ability to handle the vanishing gradient problem and promote faster convergence.

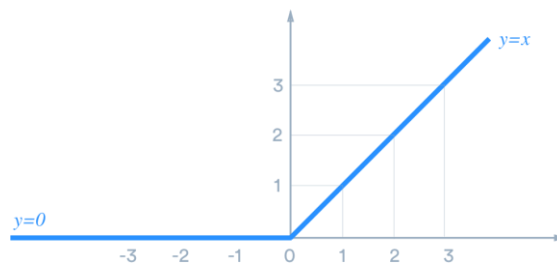


Figure 29 - Relu activation function [60]

The selection of relu as the activation function aligns with its proven effectiveness in various deep learning applications. Its piecewise linear nature and efficient computation make it a reliable choice for promoting the model's ability to capture intricate patterns and enhance training efficiency.

By settling on the learning rate of 0.0001 and the relu activation function, we aim to establish a starting point that balances stability, learning efficiency, and computational effectiveness for the model. These choices are informed by previous research, empirical evidence, and best practices in the field of deep learning.

During the initial grid search, the optimizer selected for the 3D-CNN model was **Adam**. Adam is a popular optimization algorithm widely used in deep learning due to its adaptive learning rate and efficient handling of large datasets. By utilizing Adam as the optimizer, the model underwent training and evaluation with various hyperparameter combinations to identify the optimal configuration for achieving the desired performance.

The choice of Adam as the initial optimizer was based on its effectiveness in many deep learning applications and its potential to yield good results for the 3D-CNN model. However, it is important to note that this was just the starting point, and further exploration and experimentation were conducted to determine if other optimizers could potentially produce even better outcomes.

To enhance the performance of the selected model, I focused on making adjustments to the optimizer, a crucial component in compiling a Keras model. In order to unlock potential improvements in the performance of the 3D-CNN, I conducted a series of experiments involving different optimizers such as **Adamax, SGD, and RMSprop**. Through a thorough evaluation process considering the unique characteristics and performance of each optimizer, my objective was to determine the optimizer that yielded the most favorable outcomes for the 3D-CNN model. By optimizing the model's performance and maximizing its effectiveness in the given context, I aimed to enhance its overall performance.

4.5.3.2. Best hyperparameters

As mentioned previously, the grid search method was employed in this project to optimize the values of specific hyperparameters. These may include parameters related to model architecture, learning rate, regularization techniques, or other crucial settings that significantly impact the model's performance.

By systematically varying these hyperparameters within the specified grid, the grid search method allows for a comprehensive exploration of the parameter space. For each combination, the model is trained and evaluated, and the performance metrics are recorded. This process helps identify the best-performing hyperparameter values that lead to optimal model performance.

Throughout the development of this project, various hyperparameters were thoroughly studied in order to optimize the performance of the 3D-CNN model.

NAME	TYPE	HYPERPARAMETER RANGE	APPLIED GRIDSEARCH	
TRAINING EPOCHS	Integer	[10, 20, 50, 100]	Yes	It defines the number of times the entire data set has to be worked through the learning algorithm.

NUMBER OF LAYERS	Integer	[2, 4]	No	Reducing the number of layers the model's complexity is reduced.
BATCH SIZE	Integer	[16, 32]	Yes	It defines the number of samples to work through before updating the internal model parameters
DENSE UNITS	Integer	[256, 512]	Yes	They define the size of the output from the dense layer
OPTIMIZER	Categorical	{Adam, Adamax, RMSpropz, SGD}	No	Algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses.

Table 9 - Hyperparameters studied to optimize the performance of the 3D-CNN model.

Once the acceptable hyperparameter ranges for the 3D Convolutional Neural Network (3D-CNN) were determined, a gridsearch was conducted. The results, presented in tables, helped identify the best hyperparameter configuration for the 3D-CNN. Two gridsearch processes were performed, one for a 4-layer 3D-CNN structure and another for a 2-layer structure. Each process took approximately 8 hours to complete.

Testing for 4 layered 3D CNN

Best val_loss So Far: 0.00235

Total elapsed time: 8h

Number of CT scans: 424

Best 5 combinations of hyperparameters

	TRIAL	HYPERPARAMETERS
1	0011	units: 512 batch_size: 16

2	0002	epochs: 100 Score: 0.002345 units: 256 batch_size: 16
3	0007	epochs: 50 Score: 0.0093 units: 256 batch_size: 32
4	0003	epochs: 100 Score: 0.0183 units: 256 batch_size: 16
5	0015	epochs: 100 Score: 0.0187 units: 512 batch_size: 32
		epochs: 100 Score: 0.0209

Table 10 - Gridsearch results for the 3D-CNN of 4 layers.

Testing for 2 layered 3D CNN

Best val_loss So Far: 0.000938

Total elapsed time: 8h

Number of CT scans: 424

Best 5 combinations of hyperparameters

	TRIAL	HYPERPARAMETERS
1	0005	units: 512 batch_size: 16 epochs: 100 Score: 0.000938
2	0006	units: 512 batch_size: 32 epochs: 50 Score: 0.00228
3	0001	units: 256 batch_size: 16 epochs: 100 Score: 0.0067
4	0007	units: 512 batch_size: 32 epochs: 100 Score: 0.0258
5	0002	units: 256



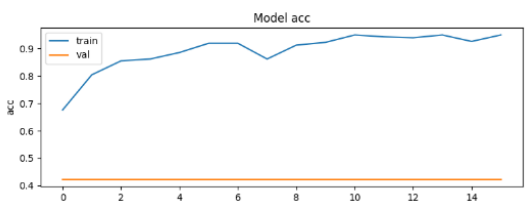
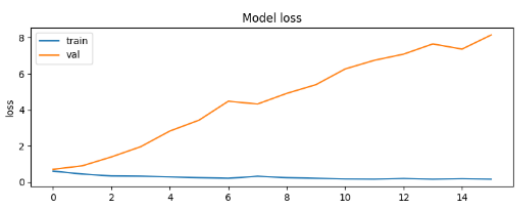
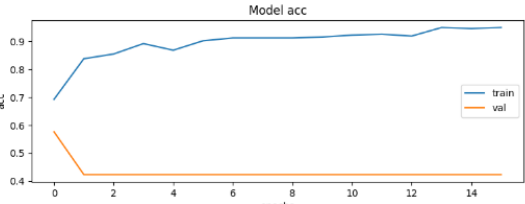
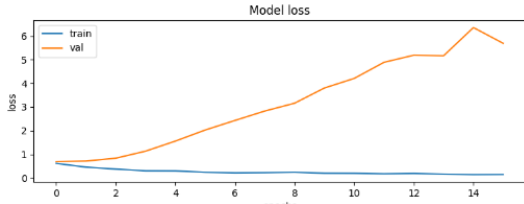
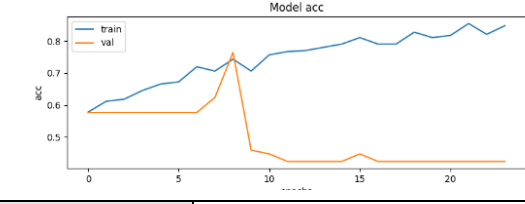
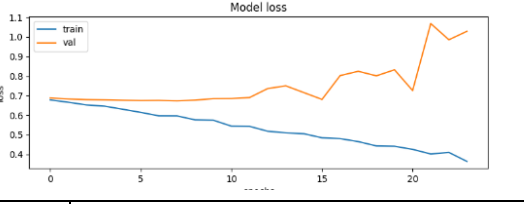
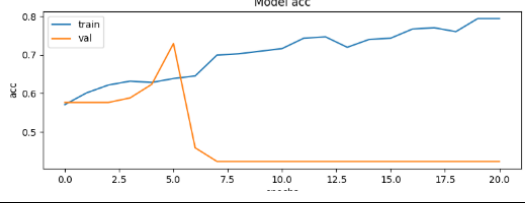
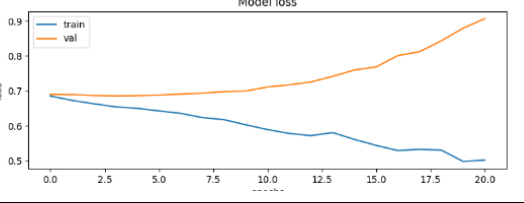
	batch_size: 32 epochs: 50 Score: 0.0317
--	---

Table 11 - Gridsearch results for the 3D-CNN of 2 layers.

5. Results and discussion

Below, you will find tables presenting the results obtained from the models with the selected hyperparameters, which were obtained using grid search. The tables include the performance metrics for both the 4-layer and 2-layer models:

4 layers		
Trials	Test loss	Test acc
1	5.608	0.4235
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Model acc</p> </div> <div style="text-align: center;"> <p>Model loss</p> </div> </div>		
2	8.2331	0.42353
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Model acc</p> </div> <div style="text-align: center;"> <p>Model loss</p> </div> </div>		
3	5.4185	0.4235
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Model acc</p> </div> <div style="text-align: center;"> <p>Model loss</p> </div> </div>		
4	8.1500	0.4235

			
5	5.68738	0.4235	
			
2 layers			
Trials	Test loss	Test acc	
1	1.0291	0.4235	
			
2	0.9070	0.4235	
			
3	1.2284	0.4235	

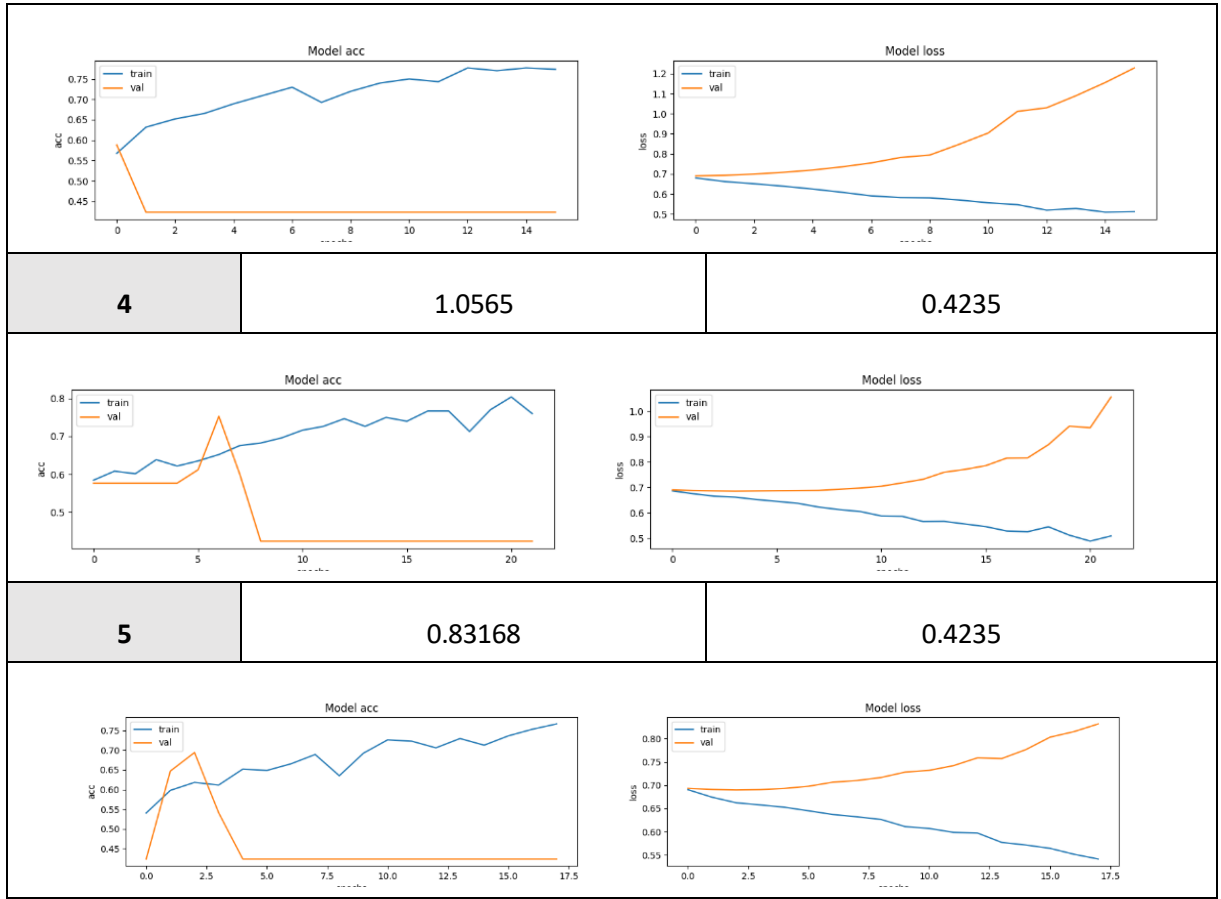


Table 12 - Results of the models.

Based on the obtained results, we can clearly observe that the models tend to **overfit**. This can be observed by looking at the accuracy graph, where the training curve (marked in blue) increases while the validation curve decreases (marked in orange).

Overfitting occurs when a model learns the training data too well, capturing specific details that may not generalize well to new data. This is due to the model being excessively trained and considering only the data identical to the training set, including its flaws.

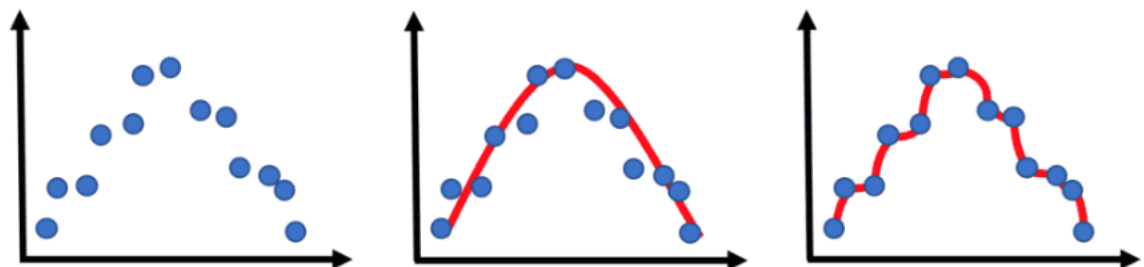


Figure 30 - Graphic representation of model overfitting [54].

The situation of overfitting can be attributed to two factors: Firstly, the model may be too complex and requires simplification. Secondly, the data used for implementing this neural network is very limited, necessitating data augmentation to generate more data variation.

The current situation is less than desirable, as indicated by the accuracy and loss values, which demonstrate that the model has not yet achieved a satisfactory level of performance. So, to make the model better, after the last obtained results we choose the best model evaluation, in this case is trial number 5 with 2 layers, with the following hyperparameters:

Number of layers: 2
 units: 256
 batch_size: 32
 epochs: 50
 Optimizer: Adam

Now we observe the results applying different optimizers:

Optimizers	Test loss	Test acc
Adam	0.83168	0.4235
Adamax	0.6783	0.5176
RMSprop	0.7797	0.4235

SGD	0.6969	0.5765

Table 13 - Results of the models with optimizers.

The modification of the optimizer in the CNN has shown slight improvements in the results. In certain cases, such as with the chosen optimizers Adamax and SGD, there is a noticeable enhancement in both the accuracy and loss values. Specifically, the loss value decreases, while the accuracy value increases to approximately 0.58.

It is important to note that the process of hyperparameter optimization in 3D-CNN primarily relies on a trial and error approach. As we observe incremental progress, we make adjustments and decisions accordingly.

After searching the best optimizer the best model is:

Number of layers: 2
 units: 256
 batch_size: 32
 epochs: 50
 Optimizer: SGD
 Test loss: 0.6969 Test acc: 0.5765

It is also important to acknowledge that grid search method has certain drawbacks as well. One notable drawback is its time-consuming nature. Since grid search evaluates all possible combinations of hyperparameters, it can significantly extend the overall training time. This can be a considerable disadvantage, particularly when working with large datasets or complex models.

Additionally, grid search carries the risk of overfitting. The exhaustive search may result in hyperparameters that are specifically tailored to the training set, leading to reduced generalization performance on unseen data. This overfitting can limit the model's ability to perform well on new, unseen instances, ultimately compromising its effectiveness.

Therefore, while grid search offers the advantage of thorough exploration, it is essential to weigh its benefits against the associated time requirements and potential risks of overfitting. Prudent consideration and careful analysis are crucial when deciding whether to employ grid search or explore alternative methods for hyperparameter tuning.

5.1. Further works

Although an exact 3D-CNN model that accurately predicts whether a CT-scan belongs to a patient with COPD or not could not be achieved, an accuracy of 0.58 has been attained. Through an extensive and lengthy study on three-dimensional convolutional neural networks, I have gained significant knowledge in this field and am confident that I could continue working on this project to refine a model with a higher accuracy. However, due to time constraints and various limitations, we can only speculate about the future work that this project could entail.

Regarding the architectures of the models, several strategies can be employed to mitigate or reduce the overfitting situation:

- **Data augmentation:** By generating new training samples through data augmentation techniques such as translation, zooming, horizontal flipping, brightness/contrast adjustment, random cropping, among others, we could obtain a wider variety of parameters. This would increase the diversity of the training data and would help the model learn more robust patterns.

Another option for data augmentation would be to use the entire ECLIPSE dataset, encompassing all the available images, rather than only relying on the 424 CT scans from the first year. The complete dataset includes CT scans for each patient across all three years of the study. In this project, we only utilized the CT scans from the first year. However, if we had access to the entire dataset, we would be able to work with a total of 1,272 CT scans.

- **Model simplification:** By reducing the number of layers, the number of units per layer, or the overall model capacity, we can decrease its complexity and potentially alleviate overfitting.
- **Proper data splitting and cross-validation:** Employing cross-validation techniques, such as dividing the data into training, validation, and test sets, helps to evaluate the model's performance more accurately. It also helps identify if the model is overfitting to the training data.

As for the limitations, future work could involve:

- Having direct access to medical data, eliminating the dependence on third parties for model implementation.
- Accessing a powerful GPU server to optimize the computational time required for these operations. This would allow for faster generation of results, facilitating modifications and improvements as desired.

It would also be interesting to upgrade the code of the 3D-CNN model by making the classification for GOLD stages. By integrating this classification aspect into the code, the model would gain the ability to provide valuable insights and predictions pertaining to the severity and progression of chronic

obstructive lung disease. Such an upgrade would significantly expand the model's utility and make it more relevant and impactful in the field of respiratory health.



6. Environmental impact analysis

This section outlines the assessment of the environmental consequences stemming from the implementation of this project. It should be noted that the entire workflow relies solely on a computer as the primary tool. While the computer in use is of high quality, processing all the images within the dataset can be time-consuming, taking more than 12 hours. To address this, we opted to connect remotely to a more powerful server, specifically the PowerEdge R640, enabling faster image processing [28].

The computer used for this project operates on the Microsoft Windows 10 Pro system model. It is important to clarify that neither this device nor the PowerEdge R640 server was procured specifically for this project. Instead, they are assets owned by the research group center, ISGlobal. Consequently, in this environmental analysis, we solely consider the electrical energy consumed by these devices during their operation, excluding any potential ecological impact associated with their manufacturing.

To estimate the environmental impact resulting from the computer's electricity consumption, it is necessary to take into account the CO₂ emissions associated with energy production. Considering that the project has been conducted over a total duration of 650 hours, the energy consumption can be calculated as follows:

Electricity consumed by (Microsoft Windows 10 Pro) = $42\text{W} * 650\text{h} = 27300 \text{ Wh} = 27,30 \text{ KWh}$

Electricity consumed by (PowerEdge R640) = $42\text{W} * 168\text{h} = 4032 \text{ Wh} = 4,032 \text{ KWh}$

Total = 27,30 KWh + 4,032 KWh = 31,332 KWh

It is assumed that the computer has been continuously connected during the 650 working hours. Additionally, it is estimated that the server has been running with all the images for approximately 168 consecutive hours.

Considering the guidelines set by the National Commission of Markets and Competition in April 2022, which state that the electricity grid mix emits 0.259kg of CO₂ per kWh, the CO₂ emissions resulting from the execution of this work are as follows:

CO₂ emissions = 31,332 KWh * 0,259 Kg/KWh = 8114,9Kg

7. Budget

This section outlines the projected expenses associated with the development of the project. The budget is subsequently divided into two main categories: personnel costs and material costs.

The average hourly pay for a Bioinformatics Scientist in Spain is €33, while for data analysts it is €27. For the purpose of this hypothetical budget, a rate of €30 per hour was considered in the calculation of the hourly cost.

Personnel costs:

TASKS	WORKING HOURS	COST (€)
FIRST MEETINGS WITH THE TUTOR	2	60
PLANIFICATION OF THE PROJECT	10	300
BIBLIOGRAPHIC REVIEW ON 3D CNN MODELS	38	1140
INTRODUCTION TO CNN PYTHON MODELS	15	450
CHOOSING THE ORIGINAL 3D CNN CODE	5	150
DELVE DEEPER INTO PYTHON	30	900
IMPLEMENTATION OF 60 IMAGES FIRST MODEL	60	1800
MEETING 1: REVISION AND IMPROVEMENTS	2	60
WRITING	23	690
CREATION OF NEW CNN ARCHITECTURES	70	2100
LITERATURE REVIEW	40	1200
WRITING	55	1650
IMPLEMENTATION OTHER ARCHITECTURES	29	870
IMPLEMENTATION OF THE TOTAL IMAGES	50	1500

RESULTS OBTENTION	20	600
MEETING 2: REVISION AND IMPROVEMENTS	1	30
CNN MODEL REVISION	35	1050
RESULTS OBTENTION	40	1200
DISCUSSION	13	390
IMPLEMENTATION BEST MODEL	12	360
WRITING	45	1350
FINAL REVIEW	35	1050
TOTAL	360 h	18,900 €

Table 14 - Personnel costs

Material costs:

MATERIALS	COST (€)
POWEREDGE R640	5536,96
MICROSOFT WINDOWS 10 PRO	209
TOTAL	5,745,96 €

Table 15 - Material costs

In conclusion, the total cost of the project amounts to 18,900 € plus 5,745.96 €, resulting in a sum of 24,645.96 €. This calculation does not include the additional expense of hiring a project supervisor or covering the costs of project implementation.

8. Conclusions

The primary goal of this project was to use a significant number of CT scan images from the ECLIPSE dataset in order to develop a 3D-CNN model for classification purposes. While we can confirm that the objective was achieved, the resulting accuracy of only 0.57 falls short. Implementing this code in real-world scenarios is currently ill-advised, as it cannot accurately discern whether a patient has COPD or not. Therefore, it lacks reliability at this stage. Despite the suboptimal results, it serves as a starting point for further progress.

As the first person in the research group to access and implement the ECLIPSE data into a 3D-CNN, this project marks a significant milestone. From this point, there is ample room for advancement and the eventual attainment of an exceptional model capable of accurate predictions.

Throughout this project, I faced a significant limitation in terms of data confidentiality and computational resources. The lack of a GPU in the available machine hindered the efficiency of building 3D-CNN models, resulting in slower processes. However, the research group has recognized this limitation and plans to acquire a GPU-equipped machine soon, which would greatly enhance the speed and effectiveness of model training, evaluation, and experimentation.

Despite the absence of a GPU, I used the resources at hand to explore and develop 3D-CNN models. However, the inability to extract data from a specific environment limited comprehensive testing, training, evaluation, and the search for optimal parameters with various models.

Nevertheless, these models have shown promising results, demonstrating the potential for positive outcomes. Further experimentation, testing, and the inclusion of a larger volume of data can lead to optimization and meaningful results.

Despite initial knowledge gaps and resource constraints, my dedication to self-learning and the support of the research group have allowed me to gain a deeper understanding of neural networks and successfully apply them in the context of 3D-CNNs.

In summary, although data confidentiality and limited computational resources presented challenges, the initial exploration and experimentation with 3D-CNN models have shown promise. Continued exploration, refinement, and expansion of the dataset based on observed outcomes hold the potential for achieving optimal and meaningful results.

9. References

- [1] *dbGaP Study*. (n.d.). Nih.gov. Retrieved May 29, 2023, from https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs001252.v1.p1&phv=293753&phd=&pha=&pht=6366&phvf=&phdf=&phaf=&phtf=&dssp=1&consent=&temp=1
- [2] Vestbo J, Anderson W, Coxson HO, Crim C, Dawber F, Edwards L, Hagan G, Knobil K, Lomas DA, MacNee W, Silverman EK, Tal-Singer R; ECLIPSE investigators. Evaluation of COPD Longitudinally to Identify Predictive Surrogate End-points (ECLIPSE). *Eur Respir J*. 2008 Apr;31(4):869-73. doi: 10.1183/09031936.00111707. Epub 2008 Jan 23. PMID: 18216052.
- [3] Sheikh K, Coxson HO, Parraga G. This is what COPD looks like. *Respirology*. 2016 Feb;21(2):224-36. doi: 10.1111/resp.12611. Epub 2015 Aug 26. PMID: 26333307.
- [4] Patel AR, Patel AR, Singh S, Singh S, Khawaja I. Global Initiative for Chronic Obstructive Lung Disease: The Changes Made. *Cureus*. 2019 Jun 24;11(6):e4985. doi: 10.7759/cureus.4985. PMID: 31453045; PMCID: PMC6701900.
- [5] Lung disease. (n.d.). Pennmedicine.org. Retrieved May 29, 2023, from <https://www.pennmedicine.org/for-patients-and-visitors/patient-information/conditions-treated-a-to-z/lung-disease>
- [6] World Health Organization. Chronic respiratory diseases. Link: https://www.who.int/health-topics/chronic-respiratory-diseases#tab=tab_1
- [7] López-Campos JL, Tan W, Soriano JB. Global burden of COPD. *Respirology*. 2016 Jan;21(1):14-23. doi: 10.1111/resp.12660. Epub 2015 Oct 23. PMID: 26494423.
- [8] Vogelmeier CF, Román-Rodríguez M, Singh D, Han MK, Rodríguez-Roisin R, Ferguson GT. Goals of COPD treatment: Focus on symptoms and exacerbations. *Respir Med*. 2020 May;166:105938. doi: 10.1016/j.rmed.2020.105938. Epub 2020 Mar 21. PMID: 32250871.
- [9] Miravittles M, Ribera A. Understanding the impact of symptoms on the burden of COPD. *Respir Res*. 2017 Apr 21;18(1):67. doi: 10.1186/s12931-017-0548-3. PMID: 28431503; PMCID: PMC5399825.
- [10] Tamondong-Lachica DR, Skolnik N, Hurst JR, Marchetti N, Rabe APJ, Montes de Oca M, Celli BR. GOLD 2023 Update: Implications for Clinical Practice. *Int J Chron Obstruct Pulmon Dis*. 2023 May 5;18:745-754. doi: 10.2147/COPD.S404690. PMID: 37180752; PMCID: PMC10168197.
- [11] Laisure, M., PharmD Candidate 2022 Western New England University College of Pharmacy, Health Sciences Nicole Covill, PharmD Candidate 2022 Western New England University College of Pharmacy, Health Sciences Marissa L. Ostroff, BCGP Clinical Associate Professor Department of Pharmacy Practice Western New England University College of Pharmacy, Health Sciences Springfield, & BCGP Clinical Pharmacy Coordinator Baystate Health—Department of Acute Care Pharmacy Services Springfield. (2021, July 21). Summarizing the 2021 updated GOLD guidelines for COPD. [Uspharmacist.com. https://www.uspharmacist.com/article/summarizing-the-2021-updated-gold-guidelines-for-copd](https://www.uspharmacist.com/article/summarizing-the-2021-updated-gold-guidelines-for-copd)
- [12] Ostridge, K., & Wilkinson, T. M. A. (2016). Present and future utility of computed tomography scanning in the assessment and management of COPD. *The European Respiratory Journal: Official Journal of the European Society for Clinical Respiratory Physiology*, 48(1), 216–228. <https://doi.org/10.1183/13993003.00041-2016>

- [13] Rohan, A. (2015, February 19). Medicare to cover lung cancer screening with CT scan. UAB News; News. <https://www.uab.edu/news/campus/item/5760-medicare-to-cover-lung-cancer-screening-with-ct-scan>
- [14] CT scan. (2022, January 6). MayoClinic.org. <https://www.mayoclinic.org/tests-procedures/ct-scan/about/pac-20393675>
- [15] Choromańska A, Macura KJ. Role of computed tomography in quantitative assessment of emphysema. Pol J Radiol. 2012 Jan;77(1):28-36. doi: 10.12659/pjr.882578. PMID: 22802863; PMCID: PMC3389962.
- [16] Xu et al., "Cross-Site Severity Assessment of COVID-19 From CT Images via Domain Adaptation," in IEEE Transactions on Medical Imaging, vol. 41, no. 1, pp. 88-102, Jan. 2022, doi: 10.1109/TMI.2021.3104474.
- [17] Ho, T.T., Kim, T., Kim, W.J. et al. A 3D-CNN model with CT-based parametric response mapping for classifying COPD subjects. Sci Rep 11, 34 (2021). <https://doi.org/10.1038/s41598-020-79336-5>
- [18] Nguyen D, Kay F, Tan J, Yan Y, Ng YS, Iyengar P, Peshock R, Jiang S. Deep Learning-Based COVID-19 Pneumonia Classification Using Chest CT Images: Model Generalizability. Front Artif Intell. 2021 Jun 29;4:694875. doi: 10.3389/frai.2021.694875. PMID: 34268489; PMCID: PMC8275994.
- [19] Ahmed, J., Vesal, S., Durlak, F., Kaergel, R., Ravikumar, N., Remy-Jardin, M., & Maier, A. (2020). COPD Classification in CT Images Using a 3D Convolutional Neural Network. In arXiv [eess.IV]. <http://arxiv.org/abs/2001.01100>
- [20] Ko, H.; Chung, H.; Kang, W.S.; Kim, K.W.; Shin, Y.; Kang, S.J.; Lee, J.H.; Kim, Y.J.; Kim, N.Y.; Jung, H.; et al. COVID-19 Pneumonia Diagnosis Using a Simple 2D Deep Learning Framework With a Single Chest CT Image: Model Development and Validation. J. Med. Internet Res. 2020, 22, e19569. [CrossRef] [PubMed]
- [21] Polat, H.; Danaei Mehr, H. Classification of Pulmonary CT Images by Using Hybrid 3D-Deep Convolutional Neural Network Architecture. Appl. Sci. 2019, 9, 940. <https://doi.org/10.3390/app9050940>
- [22] G. Gonzalez, S. Y. Ash, G. Vegas-Sánchez-Ferrero, J. O. Onieva, F. N. Rahaghi, J. C. Ross, A. D'áz, R. S. Jos'e Est'epar, and G. R. Washko. Disease staging and prognosis in smokers using deep learning in chest computed tomography, 2018.
- [23] Knipe, H., & Moore, C. (2019). NIFTI (file format). In Radiopaedia.org. Radiopaedia.org.
- [24] Prevedello, L. M., Halabi, S. S., Shih, G., Wu, C. C., Kohli, M. D., Chokshi, F. H., Erickson, B. J., Kalpathy-Cramer, J., Andriole, K. P., & Flanders, A. E. (2019). Challenges related to artificial intelligence research in medical imaging and the importance of image analysis competitions. Radiology. Artificial Intelligence, 1(1), e180031. <https://doi.org/10.1148/ryai.2019180031>
- [25] Medical image analysis with deep learning , part 4. (n.d.). KDnuggets. Retrieved June 1, 2023, from <https://www.kdnuggets.com/2017/07/medical-image-analysis-deep-learning-part-4.html>
- [26] Project jupyter. (n.d.). Jupyter.org. Retrieved June 1, 2023, from <https://jupyter.org/hub>
- [27] File:Python-logo-notext.Svg. (n.d.). Wikimedia.org. Retrieved June 1, 2023, from <https://commons.wikimedia.org/wiki/File:Python-logo-notext.svg>
- [28] Delltechnologies.com. Retrieved June 1, 2023, from <https://www.delltechnologies.com/asset/en-my/products/servers/technical-support/poweredge-r640-spec-sheet.pdf>

- [29] Amisha, Malik P, Pathania M, Rathaur VK. Overview of artificial intelligence in medicine. *J Family Med Prim Care*. 2019 Jul;8(7):2328-2331. doi: 10.4103/jfmpc.jfmpc_440_19. PMID: 31463251; PMCID: PMC6691444.
- [30] Koski E, Murphy J. AI in Healthcare. *Stud Health Technol Inform*. 2021 Dec 15;284:295-299. doi: 10.3233/SHTI210726. PMID: 34920529.
- [31] Supervised vs. Unsupervised learning: What's the difference? (2021, March 12). *IBM.com*. <https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning>
- [32] Bisong, E. (2019). What is deep learning? In *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 327–329). Apress.
- [33] Bisong, E. (2019b). What is machine learning? In *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 169–170). Apress.
- [34] Deo RC. Machine Learning in Medicine. *Circulation*. 2015 Nov 17;132(20):1920-30. doi: 10.1161/CIRCULATIONAHA.115.001593. PMID: 26572668; PMCID: PMC5831252.
- [35] McNelis, P. D. (2005). What Are Neural Networks? In *Neural Networks in Finance* (pp. 13–58). Elsevier.
- [36] Agrawal, S. K. (2021, July 19). Understanding the basics of artificial Neural Network. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/07/understanding-the-basics-of-artificial-neural-network-ann/>
- [37] Doshi, S. (2019, January 13). Various optimization algorithms for training neural network. *Towards Data Science*. <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>
- [38] Gupta, A. (2021, October 7). A comprehensive guide on optimizers in deep learning. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/>
- [39] Optimizers. (n.d.). *Keras.io*. Retrieved June 1, 2023, from <https://keras.io/api/optimizers/>
- [40] Seo H, Bassenne M, Xing L. Closing the Gap Between Deep Neural Network Modeling and Biomedical Decision-Making Metrics in Segmentation via Adaptive Loss Functions. *IEEE Trans Med Imaging*. 2021 Feb;40(2):585-593. doi: 10.1109/TMI.2020.3031913. Epub 2021 Feb 2. PMID: 33074800; PMCID: PMC7858236.
- [41] Singh SP, Wang L, Gupta S, Goli H, Padmanabhan P, Gulyás B. 3D Deep Learning on Medical Images: A Review. *Sensors (Basel)*. 2020 Sep 7;20(18):5097. doi: 10.3390/s20185097. PMID: 32906819; PMCID: PMC7570704.
- [42] O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. In *arXiv [cs.NE]*. <http://arxiv.org/abs/1511.08458>
- [43] Nips.Cc. Retrieved June 1, 2023, from https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html
- [44] Alake, R. (2020, June 12). Understand Local Receptive Fields In Convolutional Neural Networks. *Towards Data Science*. <https://towardsdatascience.com/understand-local-receptive-fields-in-convolutional-neural-networks-f26d700be16c>

- [45] Convolutional Neural Network (CNN). (2018, April 23). NVIDIA Developer. <https://developer.nvidia.com/discover/convolutional-neural-network>
- [46] Blurring an image. (n.d.). Apple Developer Documentation. Retrieved June 1, 2023, from https://developer.apple.com/documentation/accelerate/vimage/blurring_an_image
- [47] Yeh, W.-C., Lin, Y.-P., Liang, Y.-C., Lai, C.-M., & Gao, X.-Z. (n.d.). Simplified swarm optimisation for the hyperparameters of a convolutional neural network. Arxiv.org. Retrieved May 29, 2023, from <https://arxiv.org/ftp/arxiv/papers/2103/2103.03995.pdf#:~:text=The%20CNN%20structure%20consists%20of,hyperparameter%20settings%20of%20the%20CNN.>
- [48] D_08. [recap] Loss Functions explained - EN. (n.d.). 위키독스. Retrieved June 8, 2023, from <https://wikidocs.net/179999>
- [49] 19. Hyperparameter Optimization — Dive into Deep Learning 1.0.0-beta0 documentation. (n.d.). D2l.Ai. Retrieved June 8, 2023, from https://d2l.ai/chapter_hyperparameter-optimization/
- [50] Hyperparameter tuning with GridSearchCV. (2023, May 30). Great Learning Blog: Free Resources What Matters to Shape Your Career! <https://www.mygreatlearning.com/blog/gridsearchcv/>
- [51] 19. Hyperparameter Optimization — Dive into Deep Learning 1.0.0-beta0 documentation. (n.d.). D2l.Ai. Retrieved June 8, 2023, from https://d2l.ai/chapter_hyperparameter-optimization/
- [52] 3D image classification from CT scans. (n.d.). Keras.io. Retrieved June 8, 2023, from https://keras.io/examples/vision/3d_image_classification/
- [53] Shah, T. (2017, December 6). About train, validation and test sets in Machine Learning. Towards Data Science. <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>
- [54] Torres, J. (2019, September 1). Datos y overfitting. Jordi TORRES.AI; Jordi Torres. <https://torres.ai/datos-y-overfitting-keras-tensorflow/>
- [55] Cough PNG Photos. (n.d.). Pngplay.com. Retrieved June 11, 2023, from <https://www.pngplay.com/image/128196>
- [56] Download free vectors, clipart graphics, vector art & design templates. (n.d.). Vecteezy. Retrieved June 11, 2023, from <https://www.vecteezy.com/free-png/pollution>
- [57] DNA Transparent PNG, Biological DNA, DNA genetic clipart images. (n.d.). DNA Transparent PNG, Biological Dna, Dna Genetic Clipart Images - Free Transparent PNG Logos. Retrieved June 11, 2023, from <https://www.freepnglogos.com/pics/dna>
- [58] Espinoza, J.L.; Dong, L.T. Artificial Intelligence Tools for Refining Lung Cancer Screening. J. Clin. Med. 2020, 9, 3860. <https://doi.org/10.3390/jcm9123860>
- [59] Rakhecha, A. (2019, June 28). Understanding learning rate. Towards Data Science. <https://towardsdatascience.com/https-medium-com-dashingaditya-rakhecha-understanding-learning-rate-dd5da26bb6de>
- [60] Liu, D. (2017, November 30). A practical guide to ReLU - danqing Liu. Medium. <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>

10. ANNEX

In this section, the GitHub repository is displayed with the necessary files to make the implementation of the 3D-CNN work.

You may access to the GitHub repository via this link: https://github.com/isglobal-brge/master_thesis

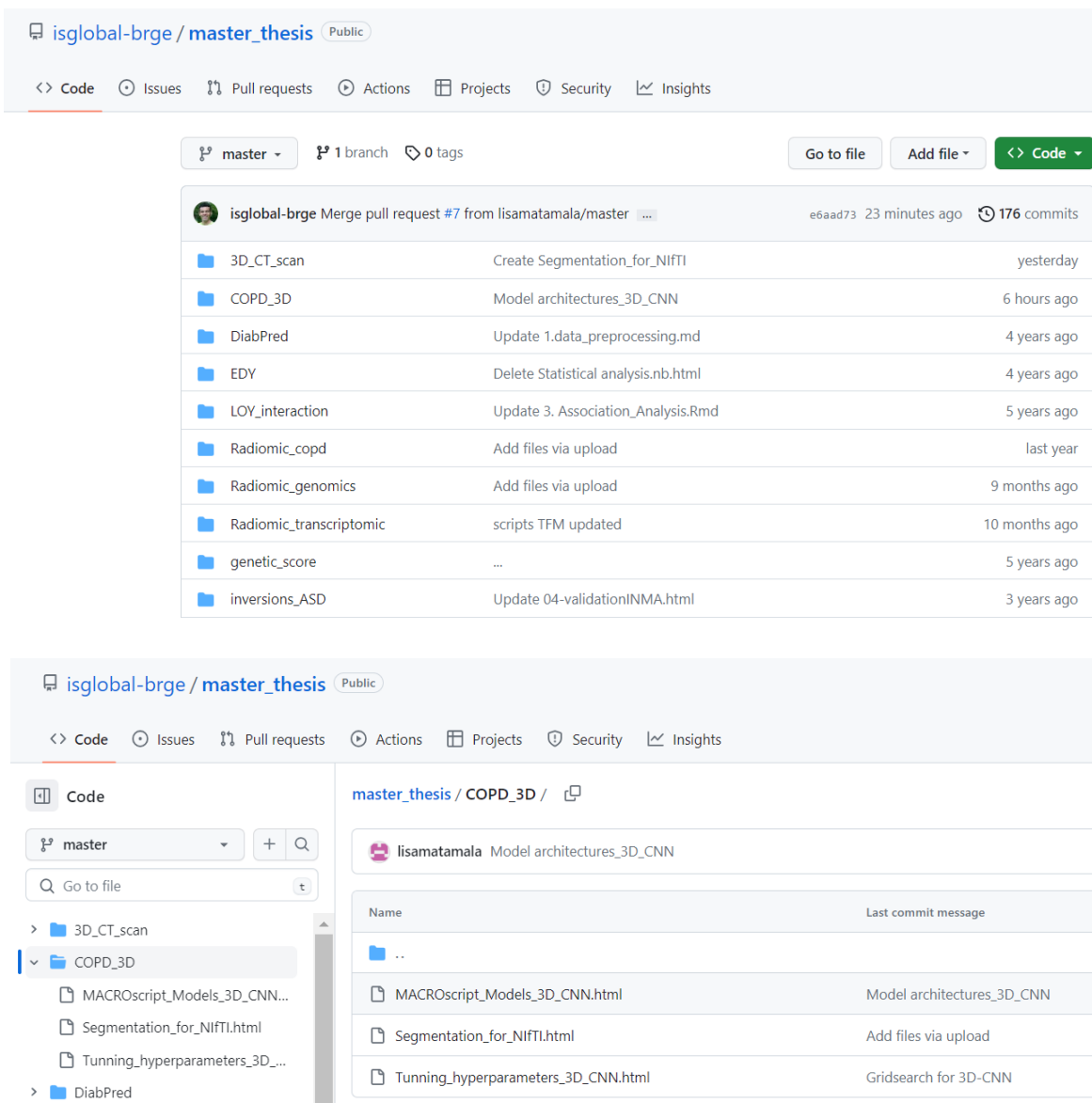


Figure 31 - Screenshots of the BRGE group repository on GitHub

