# Enhancing Image Consistency in CT Scans: A PIX2PIX-Based Framework for Cross-Modality Transformation

Bruno Alberto Carvalho Malta
Bioinformática e Biologia Computacional
Departamento de Ciências da Computação
2023

**Orientador**
Doutor Hélder Oliveira, Prof. Auxiliar Convidado da Faculdade de Ciência da Universidade do Porto
**Coorientador**
Doutor Hélder Novais e Basto, Prof. Auxiliar Convidado da Faculdade de Medicina da Universidade do Porto

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

MSc

# Declaração de Honra

Eu, Bruno Alberto Carvalho Malta, inscrito(a) no Mestrado em Bioinformática e Biologia Computacional da Faculdade de Ciências da Universidade do Porto declaro, nos termos do disposto na alínea a) do artigo 14.º do Código Ético de Conduta Académica da U.Porto, que o conteúdo da presente dissertação reflete as perspetivas, o trabalho de investigação e as minhas interpretaçõesno momento da sua entrega.

Ao entregar esta dissertação declaro, ainda, que a mesma é resultado do meu próprio trabalho de investigação e contém contributos que não foram utilizados previamente noutros trabalhos apresentados a estaou outra instituição.

Mais declaro que todas as referências a outros autores respeitam escrupulosamente as regras da atribuição, encontrando-se devidamente citadas no corpo do texto e identificadas na secção de referências bibliográficas. Não são divulgados na presente dissertação quaisquer conteúdos cuja reprodução esteja vedada por direitos de autor.

Tenho consciência de que a prática de plágio e auto-plágio constitui um ilícito académico.

Bruno Alberto Carvalho Malta

Coimbra, 30-06-2023

# Abstract

CT imaging plays a crucial role in diagnosing various pathologies, especially in the lungs. It is considered a valuable and objective method for evaluating the presence and progression of diseases such as cancer or interstitial lung diseases. The utilization of CT scans, particularly the extraction of radiomic features, shows promise in aiding healthcare professionals in comprehending the extent of specific diseases. However, an inherent challenge arises from the fact that the parameters of CT scans can vary across different machines. This variability poses difficulties when attempting to address radiomic features in a more comprehensive and generalized manner.This thesis investigates the challenges associated with employing diverse reconstruction kernels in computed tomography (CT) imaging of the lungs. It explores the potential of deep learning methods, specifically generative/domain transfer models, to transform CT images reconstructed with one kernel to closely resemble those reconstructed with a different kernel.

The motivation for this thesis underscores the importance of CT imaging in the medical field and highlights the growing role of artificial intelligence (AI), particularly deep learning, in healthcare applications such as image analysis and disease diagnosis. Developing AI-based techniques for medical image transformation can address the challenges arising from different reconstruction kernels, leading to enhanced interpretability and comparability of CT scans.

The objectives of this thesis are to explore the potential of deep learning, particularly the PIX2PIX model, for transforming CT images from one kernel to another.Through training the model on a dataset containing paired images reconstructed with various kernels, it learns to generate images that closely resemble those acquired with the target kernel. The thesis examines the efficacy of the PIX2PIX model in accurately transforming CT images from one kernel to another.

The findings of the study highlight the capability of the PIX2PIX model to effectively tackle the challenges linked to diverse reconstruction kernels in lung CT imaging. The quantitative analysis reveals enhanced resolution in the domain transfer process between different CT kernels, while qualitative evaluation demonstrates favorable outcomes. Additionally, the thesis incorporates optimization techniques and fine-tuning procedures to optimize the quality of the generated images.

Keywords: Deep learning, Machine learning, Artificial Intelligence, Generative models, PIX2PIX, domain transformation, Lung CT, Computed Tomography, Fibrosis, Lung Cancer)]

# Resumo

A imagem por TAC desempenha um papel crucial no diagnóstico de várias patologias, especialmente nos pulmões. É considerado um método valioso e objetivo para avaliar a presença e progressão de doenças como o cancro ou doenças pulmonares intersticiais. A utilização da TAC, nomeadamente a extração de características radiómicas, mostra promessa em auxiliar os profissionais de saúde a compreender a extensão de doenças. No entanto, surge um desafio inerente devido ao facto de os parâmetros dos exames de TAC poderem variar entre diferentes máquinas. Essa variabilidade coloca dificuldades quando se tenta abordar as características radiómicas de forma mais abrangente e generalizada. Esta tese investiga os desafios associados à utilização de diversos núcleos de reconstrução na tomografia computadorizada (TAC) dos pulmões. Explora o potencial de métodos de machine learning, especificamente modelos generativos de transferência de domínio, para transformar as imagens de TAC reconstruídas com um núcleo de forma a assemelharem-se às imagens reconstruídas com um núcleo diferente.

A motivação para esta tese realça a importância da TAC no campo médico e destaca o papel crescente da inteligência artificial (IA), especialmente de Deep Learning, em aplicações de saúde como análise de imagens e diagnóstico de doenças. O desenvolvimento de técnicas baseadas em IA para transformação de imagens médicas aborda desafios decorrentes de diferentes núcleos de reconstrução, melhorando a interpretabilidade e comparabilidade dos exames de TAC.

Os objetivos desta tese passam por explorar o potencial do uso de Deep Learning, em particular do modelo PIX2PIX, para transformar imagens de TAC de um núcleo para outro. Através do treino do modelo com um conjunto de dados contendo imagens em pares reconstruídas com vários núcleos, o modelo aprende a gerar imagens que se assemelham às adquiridas com o núcleo alvo. A tese examina a eficácia do modelo PIX2PIX na transformação precisa de imagens de TAC de um núcleo para outro.

Os resultados do estudo destacam a capacidade do modelo PIX2PIX em lidar efetivamente com os desafios relacionados com os diversos núcleos de reconstrução na imagem de TC do pulmão. A análise quantitativa revela uma resolução aprimorada no processo de transferência de domínio entre diferentes núcleos de TC, enquanto a avaliação qualitativa demonstra resultados favoráveis. Além disso, a tese incorpora técnicas de otimização e procedimentos de ajuste fino para otimizar a qualidade das imagens geradas.

# Agradecimentos

Gostararia de expressar os meus sinceros agradecimentos aos meus Pais, por todo o apoio incondicional que me deram durante estes últimos 3 anos, que não foram nada fáceis, sem eles nunca conseguia finalizar este mestrado. Agradeço também aos meus amigos, os que sempre me apoiaram e me levantaram a cabeça nos momentos em que eu estive mais em baixo. Gostaria de agradecer à minha namorada Ana, por me ter sempre ajudado e dado a mão quando as coisas não pareciam ter solução, sem ela este processo todo não ia ser nada fácil. Queria agracer aos meus orientadores, Dr.Hélder Novais e Bastos e em especial ao Professor Hélder Oliveira, por me ter dado ferramentas que me permitiram levar esta tese avante e conseguir finalizar. Não foram 3 anos fáceis, em que enfrentei a dificuldade de ter a responsabilidade de ser monitor de estágio, de trabalhar e ao mesmo tempo estudar um conjunto de áreas que não faziam parte da minha zona de conforto, tive de ter muita resistência e resiliência para conseguir não só superar isto, como também superar todos os obstáculos que me foram colocando ao longo deste percurso. Consigo finalizar esta saga de 3 anos de forma bastante positiva e merecedora do esforço que fui apresentando.

# Contents

# List of Tables

# List of Figures

# Acronyms

| | |
|---|---|
| **AI** | Artificial Intelligence |
| **DL** | Deep Learning |
| **ANN** | Articicial Neural Networks |
| **CV** | Computer Vision |
| **ILDs** | Interstitial Lung Diseases |
| **HRCT** | High Resolution Computer Tomography |
| **ES** | Emphysema Score |
| **HU** | Hounsfield Unit |
| **COPD** | Chronic Obstrutive Pulmonary Disease |
| **KNN** | K-nearest neighbour |
| **SVM** | Support Vector Mahcines |
| **MR** | Magnetic Ressonance |
| **ARIMA** | Autoregressive Integrated Moving Average |
| **CED** | Convolutional Encoders-Decoders |
| **CNN** | Convolutional Neural Networks |
| **GAN** | Generative Adversarial Networks |
| **VAE** | Variational Auto-Encoders |
| **PCA** | Principal Component Analysis |
| **RNN** | Recurrent Neural Networks |
| **RMSE** | Root Mean Square Error |
| **MAPE** | Mean Absolute Percentage Error |
| **BTS** | Back-Propagation through the System |
| **NLP** | Natural Language Processing |
| **STM** | Short Term Memory |
| **LSTM** | Long Short Term Memory |
| **GAP** | Global Average Polling |
| **FC** | Fully Connected |
| **BGD** | Batch Gradient Descent |
| **SGD** | Stochastic Gradient Descent |
| **EAML** | Expert Augmented Mahcine Learning |
| **FAIR** | Findability, Accessibility,Interoperability, Reusability |
| **FID** | Fréchet Inception Distance |
| **SSIM** | Structural Similarity Index Measure |

# Chapter 1

# Introduction

Artificial intelligence (AI) has received a lot of attention in particular, deep learning (DL). This success is attributable to their remarkable ability to solve complicated problems and provide original solutions [1].The concept of AI dates back to the 40's with the term "artificial intelligence" itself being coined in 1956 by John McCarthy [2], in the Dartmouth workshop [3].

In the 50's and 60's AI research made a significant progress, including the development of natural language processing, computer vision and robotics. Between the 70's and the 80's AI was subject to critiques and financial setback, much because of the difficulty to solve the problems raised by the high expectations of the AI researches, thus, losing all the funding [4]. It was not until the 90's, with the boom of AI, mostly because of the successful use of machine learning models (ML), that AI regain the financial support. ML uses specific traits to identify patterns that can be used to solve more complex problems. The algorithm processes the data and apply that information to future similar scenarios [5].

Between the 00's and 10's, AI research continued to advance with the development of DL models, allowing computers to learn more complex and abstract representations of data. Artificial neural networks (ANN) were the advent of the DL models, being a class of algorithms inspired by the structure and function of the human brain. They use multiple layers to progressively extract higher-level features from the raw input of the data. The term" deep" comes from the multiple layers and the recursively feature extraction that occurs within the model [6].

In recent years, advancements in computational power and data availability have enabled unprecedent breakthroughs in AI applications based on ML algorithms, particularly in the field of computer vision (CV). The use of Big Data, an impressive quantity of data used for analysis, instead of the traditional simpler survey [7], had increase the potential of the AI models.

The medical industry has benefited from these advancements by developing AI systems that may assist clinical judgments or automate certain clinical practice steps[8]. In order to complete difficult tasks in image and data analysis/processing in general and multimodality medical imaging in particular, the unique ML/DL algorithms have demonstrated excellent learning potential from high dimensional/complex data. The benefits of applying AI in the medical industry include, but are not limited to, tasks like image segmentation, image classification, data correction, image interpretation, cross-modality, or image translation [9],[10].

Papers relying on AI and ML report promising results in a wide range of medical applications, including disease diagnosis, image segmentation and outcome prediction. Overall, the latest success of AI in medical imaging is a result of its ability to mimic features that are characteristic of human intelligence, such as problem solving or learning and it is expected to have a significant impact on the field [1].

## 1.1 Motivation

Interstitial Lung Diseases (ILDs) comprise a heterogeneous group of diffuse parenchymal lung disorders, characterized by the infiltration of immune effector cells, fibroblasts, myofibroblasts and extracellular matrix deposition, producing progressive scarring and destruction of the pulmonary tissue, which leads to shortness of breath and ultimately to respiratory failure and death. Symptoms like airflow limitation, often presents as dyspnea and is caused by airway disease (chronic bronchitis) or destruction of lung parenchyma (emphysema). There is a subjectivity in the diagnosis as it relies on the pulmonary function maneuvers and on high resolution computed tomography (HRCT), that demands great expertise of the radiologist, thus creating intra and inter-observer disagreement even among experienced radiologist and technicians. High resolution computed tomography (HRCT) allows visualization of pathologic changes in the lung parenchyma and classification of patients into different phenotypes according to the presence of bronchitis or emphysema.

Emphysema score (ES), which is the proportion of lung voxels below a specific Hounsfield Unit (HU) threshold, is a regularly used method for estimating the amount of emphysema in the lungs using CT imaging of lung attenuation. ES is a recognized method for assessing the severity of emphysema and has been shown to correlate well with pathology and pulmonary functions tests [11]. Emphysema is identifiable on CT scans by the presence of regions with decreased attenuation, creating a visual contrast with the normal attenuation of the surrounding lung tissue. The severity of emphysema in lung regions is categorized into different grades: no emphysema (score 0), <25% emphysema (score 1), <50% emphysema (score 2),<75% emphysema (score 3), and >75% emphysema (score 4). This grading system allows for the classification and quantification of emphysema based on the extent of abnormal attenuation observed in the lung parenchyma (Figure 1.1). Emphysema quantification has been demonstrated to be substantially influenced by the filter kernel that is used [11]. In general, sharper (higher spatial resolution) reconstruction kernels, like B50f, produce larger ES than smoother (lower spatial resolution) kernels, like B30f. As a result, it is hard to compare emphysema quantification from scans acquired with various parameters in a meaningful way. For multi-center and longitudinal investigations where it may be challenging or impossible to manage scan parameter settings this is a crucial problem.

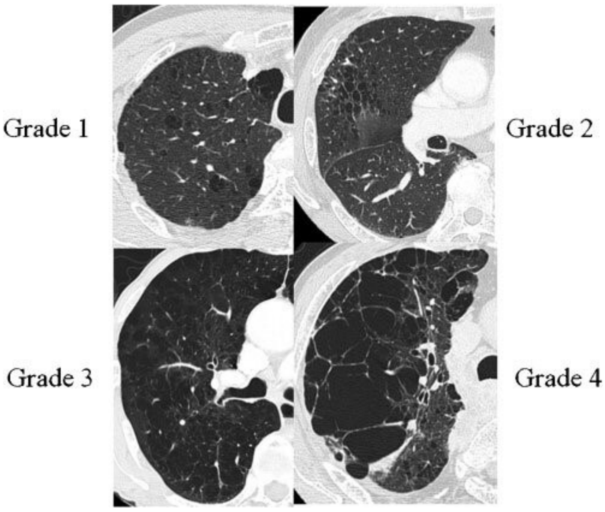


Figure 1.1: Representation of the various ES scores. Adapted from Tomoki Kimura et al *Radiation pneumonitis in patients with lung and mediastinal tumours: A retrospective study of risk factors focused on pulmonary emphysema*

Radiomic features play a crucial role not only in the evaluation of emphysema but also in other pathologies such as lung cancer or Chronic Obstructive Pulmonary Disease (COPD). In these cases, the utilization of radiomic features becomes essential for a comprehensive assessment. These features involve quantitative measurements derived from medical images using advanced image processing and analysis techniques. By capturing a wide range of information regarding spatial patterns, textures and intensities within the images, radiomic features provide valuable insights into the characterization of various diseases and tissues. Like said above the spatial relationship of voxels within different reconstruction kernels can vary, leading to inconsistent values for the same image slice obtained using different reconstruction kernels. This variability can affect the stability and reliability of radiomic features extracted from the images. Therefore, careful consideration and standardization of reconstruction kernels are necessary to ensure the consistency and reproducibility of radiomic measurements in clinical practice.

## 1.2 Goals and Contributions

The objective of this project is to develop an image-to-image translation model, based on PIX2PIX approach, specifically designed for converting CT scans acquired with the B30f kernel to CT scans acquired with the B50f kernel, and vice versa. The main contributions of this project include:

- Exploring a supervised approach for the conversion between B30f and B50f CT scans (and vice versa).

- Evaluating and optimizing the model's parameters to achieve accurate and reliable transformations.

- Conducting a comprehensive quantitative and qualitative analysis of the transformed images across different anatomical CT cuts.

- Investigating the preservation of high-level details in the lung parenchyma during the transformation process.

- Drawing conclusions regarding the practical usability and applicability of such transformations in real-world scenarios

## 1.3 Structure

The thesis commences by providing a comprehensive exposition on the foundational principles and fundamental concepts underlying machine learning and deep learning. It subsequently delves into significant topics pertaining to the treatment of medical images, elucidating their significance as well as the challenges associated with their analysis and interpretation in relation to artificial intelligence (AI) methodologies. Furthermore, a detailed explanation follows, providing a thorough understanding of the complexities of deep learning, covering important principles and various types of deep learning architectures. The thesis then proceeds to explore the application of convolutional neural networks (CNNs) and generative adversarial networks (GANs), emphasizing their notable characteristics and significant role in analyzing images and generative power. Afterward, the thesis conducts a thorough analysis of the main problem, examining its fundamental aspects and consequences. Finally, the research methodology, experimental procedures, and the resulting findings are presented, culminating in a comprehensive conclusion that summarizes the final insights and outcomes obtained from the study.

# Chapter 2

# Machine learning, Deep Learning and Medical Applications

## 2.1 Machine learning

In the ML methods, the system gained knowledge via experience. For instance, the type of data fed to the system learns the pattern and reacts at the output based on what it has learned, without human intervention, the model gets smarter and smarter over time. It employs a statistical learning system that continuously develops without assistance [12]. Relationships and patterns from the data, are some examples that ML methods use to codifying knowledge into computers [13]. The recent emergence of intelligent systems with cognitive abilities akin to humans has been made possible by advances in ML models and computer power. These systems are permeating both our personal and professional lives and shaping networked interactions on electronic markets in every imaginable way, with businesses enhancing decision making for productivity, engagement and employee retention [14].

In the ML algorithms, their classification is done in three major categories, which are supervised, unsupervised and reinforcement learning (Figure.2.2)

Supervised Learning uses an algorithm that requires processing from a human source. This means that after a laborious pre-processing of the data, this input is then separated into training and testing datasets. The output variable is predicted or classified from the training database. Algorithms try to learn some shapes during training of the database and implement these learned patterns to the testing database [15]. Training allows patterns to be found in previously collected data, whereas inference compares these patterns to new unseen data to then carry out a certain task like prediction or decision making [2].

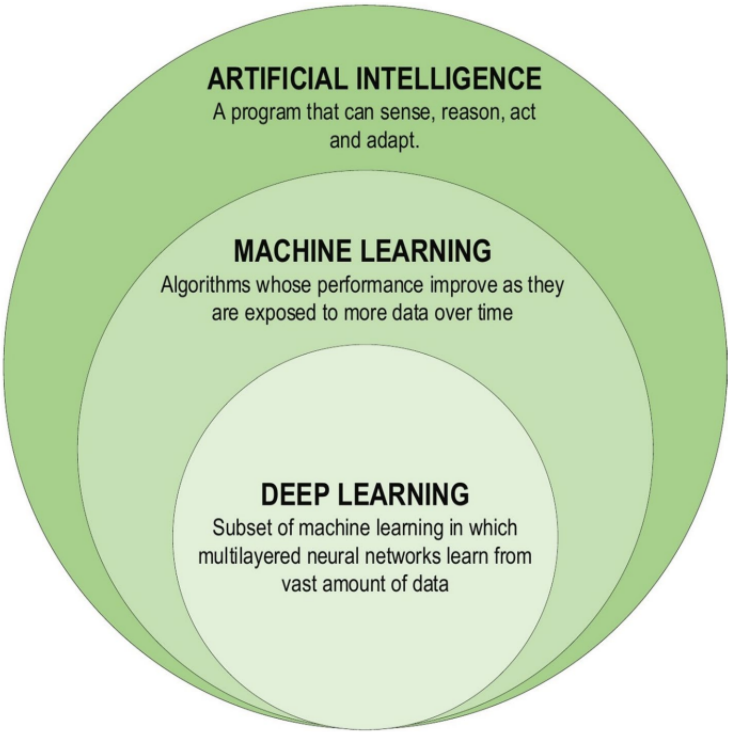

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act and adapt.
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amount of data

Figure 2.1: Constitution of AI. Adapted from Laith Alzubaidi et al *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*

Unsupervised learning is a machine learning algorithm that learns some characteristics of input information without label data. After providing a new database, it utilizes formerly learned characteristics for the identification of the class of data. It is mostly preferred for feature reduction also for clustering [2].

A type of machine learning called reinforcement learning teaches an agent how to operate in a given environment by having it execute certain behaviours and then getting rewards or punishments in response to those actions. Over time, the agent gains the ability to maximize rewards by figuring out the best course of action to pursue in each environmental condition. Numerous fields, including robotics, autonomous vehicles, and gaming, have adopted reinforcement learning. One of reinforcement learning's key benefits is that it may develop decision-making skills without explicit guidance or labelled data, which is advantageous in situations where labelled data is hard to come by or is prohibitively expensive [16].

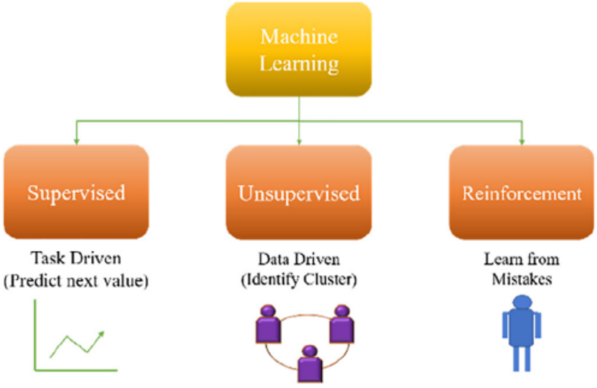


Figure 2.2: Major categories of ML.Adapted from Neha Sharma et al *Machine Learning and Deep Learning Applications-A Vision*

The most straightforward method that offers the best assurances and the tightest framework is supervised learning[2]. Logistic regression, K-nearest neighbour (KNN), Support Vector Machines (SVM), Decision Trees, Random Forest, ANN, Naïve Bayes are some algorithms that we can use to do some supervised regression or classification tasks. The choose of the algorithm, will depend on the complexity of the problem and on the nature of the data itself being use to feed the algorithms [17].

Classification can be binary, such as when evaluating whether a pathology is present in the patient image or not [17],[18], or multi-class, such as when identifying a specific pathology among multiple labels [19],[20]. On the regression side, the main goal is to predict a continuous output variable based on a set of input features. In regression the algorithm is trained on a labelled dataset, where each data point consists of a set of features and corresponding output value. The algorithm learns to model the relationship between the input features and the output value and can then make predictions on new, unseen data [21].

Contrarily, the majority of unsupervised tasks, such as clustering (identifying distinct groups of similar data), outlier detection, anomaly detection or manifold learning and dimensionality reduction, are related to the probability density estimates[2]. Although there are useful applications for medical imaging, such as domain adaptation (i.e., adapting a segmentation model trained on one image modality to work on a different image modality) [22],[23], generation of data [24][25]or even image segmentation [26], the unsupervised learning has been much more restricted than that of its supervised counterpart up to this point.

Semi-supervised learning uses a hybrid framework halfway between supervised and unsupervised. Unsupervised learning-based clusters can be utilized as potential class labels [2]. Semi-supervised learning can be also used in the translation of images from a specific class to another in a semi-supervised setting (i.e., generation of synthetic CTs from MR images) [[27],[28] and segmentation or classification of images with partially labelled data [29], [30]. Is widely known that data labelling in the medical areas is a laborious operation that requires expensive review by health care professionals. Semi-supervised learning approaches are a great option to supplementing small sets of correctly labelled data with huge quantities of unlabelled data acquired automatically, more and more academics and researchers are increasingly investigating them [30]. Even though the use of fully unsupervised learning in the medical field is still very limited, the future research will must probably focus on unsupervised techniques in order to unlock the full potential of ML methods [2]. The real limitations of ML/DL algorithms come from the scarce size of labelled data. Unsupervised models have lately outperformed supervised models for CV tasks [31], and the same is expected to occur for applications involving medical imaging.

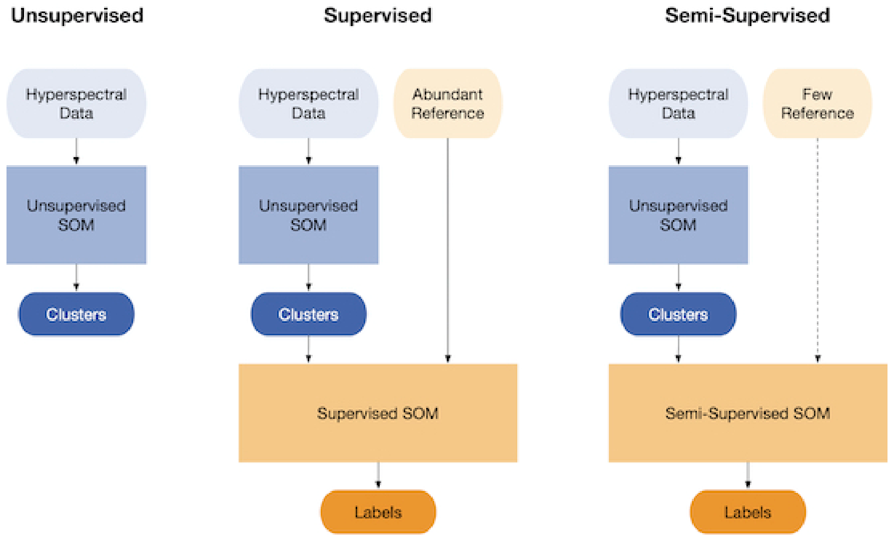


Figure 2.3: The differences between Unsupervised vs Supervised vs Semi-Supervised learning. Adapted from Felix M Riese et al *Supervised and Semi-Supervised Self-Organizing Maps for Regression and Classification Focusing on Hyperspectral Data*

Reinforcement learning is still not widely used in medical imaging, but in the past several years to certain exciting applications that enable emulating physician behaviours, like creation of a therapy [32].

## 2.2 Deep Learning

Deep neural networks, a subclass of ML models, in simple terms is a network model with layers between input and output and neurons with various properties. DL adopts the methodology of neural network structures. These artificial neurons are mathematical models of interconnected processing units that are inspired by the idea of information processing in biological systems, very similar to the synapses, where each link between neurons produces signals whose intensity may be boosted or decreased by weight that is continually modified during the learning process [33] (Figure 2.4). Only when a specific threshold is exceeded, as specified by an activation function, are signals processed by following neurons. Deep learning entire architecture is employed for the feature extraction and alteration procedure which puts it at odds with conventional machine learning techniques due to its potent process. Typically, an output layer creates the final result after receiving the data supplied from an input layer. The topology of the architecture is going to define the complexity of our model (e.g., to learn a non-linear mapping between input and output, there must be zero or more hidden layers in between) [34]. Other property choices, such learning rate or activation function, as well as the number of layers and neurons, cannot be learnt by the algorithm, and are considered the hyperparameters of the model that must be humanly chosen. Deep learning therefore is more appropriate for handling difficult problems and bigger amounts of data [35]

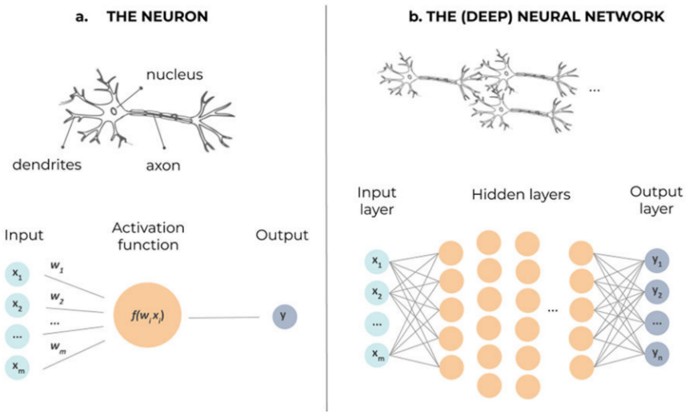


Figure 2.4: The comparison between a real neuron and an artificial neuron(a) and a real neuronal network with a deep learning network.Adapted from Ana Barragán-Montero et al *Artificial intelligence and machine learning for medical imaging: a technology review*

## 2.3 Medical Applications of Deep Learning and Machine learning

Bioinformatics and machine learning are two fields that have become increasingly intertwined in recent years. Bioinformatics involves the use of computational methods to analyse and interpret biological data, including genomic, transcriptomic, proteomic metabolic data [36]. The novel Covid-19, had show the importance of the ML and bioinformatics evolution. Diagnosing patients, identifying who is at most risk, better understand viruses, predict the spread of the disease, map from where the virus come, discovering drugs, are some examples of how Bioinformatics and ML models can help in the fight against some pathogenic agents. ML can be used to predict the behaviours of new cases to stop the disease from spreading as machine learning is trained with mathematical models for learning and analysing. After training the machine, an interesting pattern can be detected. Researche's Li et al [37], developed a prediction model with ML to detect the reported cases in China and the world and Kumar et al [38], applied the ARIMA (autoregressive integrated moving average) model to predict the coronavirus spread in 15 most infected countries.

DL methods are revolutionizing healthcare by enabling machines to recognize patterns in medical data and predict outcomes. This is due to various reasons, one of the reasons comes from the fact that the algorithms extract complex features from big data and make this feature valuable tools for medical diagnosis, treatment planning and drug development.

Convolutional Neural Networks have recently attracted more interest compared to some simpler ML algorithms like SVM's or Random Forest [2] (Figure 2.5). Additionally, since 2018, the use of DL techniques including reinforcement learning algorithms and generative adversarial networks has been growing quickly.

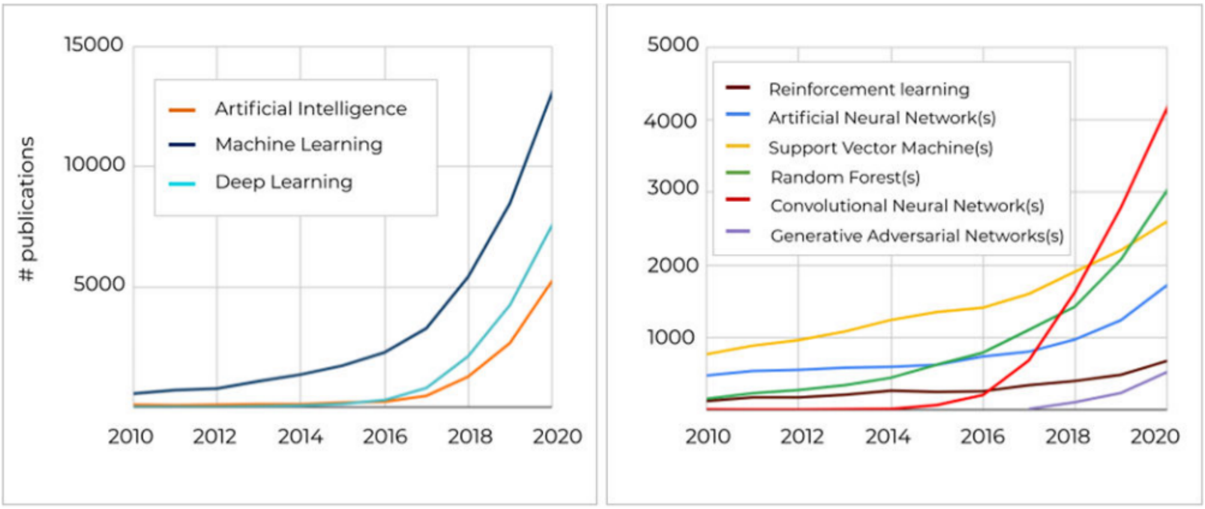


Figure 2.5: How many publications in the PubMed archive from 2010 to 2020 had titles or abstracts with keywords related to AI, ML, or DL techniques.Adapted from Ana Barragán-Montero et al *Artificial intelligence and machine learning for medical imaging: a technology review*

Convolutional encoders-decoders (CED) networks, consist of an encoder and decoder portions meant to transform input images into more abstract features and turn these abstract features into images again, being this one type of architecture that is very used in the field of medical image analysis [39]. GANs, are made up of two main parts: a generator, can be a CED network or a simple vanilla CNN, and a discriminator, that is going to distinguish between the real data and the fake image [39]. Image segmentation and image translation had been implemented with base on those architectures [40]. The use of variational auto-encoders (VAEs) is another option when trying to generate artificial medical images. This one is little different from the other ones, because uses a CED with the goal to represent the statistical distribution of an image in the encoder (probabilistic latent representation) and tries to use a reparameterization trick to computationally turn the model less expensive. The decoder uses the mean and the standard deviation train by the encoder to represent the image, and tries to create an image based on this distribution [41].

U-Net is one of the most popular architectures built upon the CED structure, adding some skip connections for context capturing and for creating a symmetric expanding path, which enables more efficient feature selection [42]. Upgrading networks with different modules, such as attention blocks/ components [43] for highlighting salient features in the input data and residual connections [44] to prevent vanishing gradient, are used to improve the overall performance of the networks. With the development of task-specific loss functions and various points of view in the DL techniques, the GANs models also suffer some evolution, with the use of conditional Gans (cGANs) [45], cycle consistency (Cycle-GAN) [46] and Pix2Pix [45] approach (the use of cGANs with different methodology).

The urge to automatize the diagnosis and segmentation of medical imaging is helping the development of DL software's. In order to classify some parts of the body as malignancies or traumas, DL is employed for X-Ray diagnosis. CNN base models like ResNet-50, Inception-V3 and Inception-ResNet-V2 used to predict Covid-19 patients with chest X-Ray images [47], [48] and reported that ResNet-50 had the best detection accuracy (98%). 95.38% of accuracy was reported using ResNet50 and SVM after using a variety of DL models[49].

Another area where deep learning is making significant contributions is in drug discovery. Deep learning algorithms can analyse large datasets of chemical compounds and predict which compounds are most likely to be effective against a particular disease. This can greatly speed up the drug discovery process, potentially leading to the development of new drugs more quickly and at lower cost.

Overall, deep learning is transforming healthcare by providing new tools for medical diagnosis, treatment planning, and drug development. As the field continues to advance, we can expect to see even more exciting applications of deep learning in healthcare in the years to come.

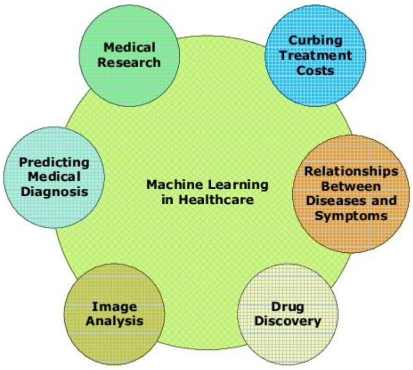

Medical Research
Curbing Treatment Costs
Predicting Medical Diagnosis
Machine Learning in Healthcare
Relationships Between Diseases and Symptoms
Image Analysis
Drug Discovery

Figure 2.6: The main characteristics of medical imaging and the potential use of DL techniques. Adapted from S. Kevin Zhou, Hayit Greenspan et al *A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises*

## 2.4 Medical imaging

Before the implementation of any ML/DL algorithm, is very important to pre-processing the data (images). Medical images have a particular set of important rules that doesn't exists in other sets of data. First the quality of the data acquisition (i.e., medical images as CT, MRI, PET scans), are acquired from the patients (random or chosen, accordingly to the main goal of the problem) and stored in a reliable format (i.e., NRRD/Dicom).

In the Data pre-processing, the acquired images may undergo various pre-processing steps to ensure their quality, such as normalization, filtering and noise reduction. Data extraction and feature selection are two important phases in this sector of the workflow in the medical imaging analysis[2] . We can define feature as being quantitative traits that condenses the information of the data into vector or arrays. General predictions models, are trained to carry out specific tasks using those features. This approach is demonstrated in the field of radiomics [50], [51], where radiomics properties are extracted from radiological images in order to forecast some important indicator, such as the severity of the disease or the prognostic of the patient. Feature engineering is the primary process for directing data to an ML approach [52]. There is the possibility to create the image features manually (more difficult, more time consuming and more prone to errors), or use deep learning (i.e., Autoencoders), to extract the features from the low-level ones to a more complex and abstract features called, high-level features. Low-level features are those that are particular to a small collection of pixels in the picture, while high-level features are those that characterize the whole image. Gabor or Laplace filters, edge detectors like Sobel operators, texture descriptors, Zernike moments or transforms like Fourier's or wavelet bases, are examples for extracting low-level features [2]. For the high-level features, the use of Principal Component Analysis (PCA) is an alternative. More specifically, the convolutional filters used in CNNs for images are comparable to the filters mentioned, because they extract local/low features but learn the parameters from data and stack them to reveal the global higher-level features[2] (Figure 2.7). Another important thing in the medical image's treatment is the annotation and labelling. Medical images are annotated and labelled with relevant information, such as the presence of abnormalities, the location and size of lesions and other important clinical information.

The typical holdout method is used, where the annotated images are split into training, validation and testing sets. The training set is used to train AI models, while the validation and testing sets are used to evaluate the model's performance and improve some regularization parameters. The model development is based on the AI algorithms and can be developed using various techniques such as convolutional neural networks (CNN), recurrent neural networks (RNN) or generative adversarial networks (GANs). To training the models is important to use the annotated training data to learn patterns and features that are useful for the given medical imaging task[2].

## 2.5 Model evaluation

To evaluate a model performance, is important to consider a number of factors, including computational resources and interpretability. Performance-based metrics measure how effectively a model achieves the learning task's purpose. There are recognized rules for this in the field of supervised learning. In these cases, it is usual practice to employ k-fold cross validation to avoid overfitting a model and assess how well it performs on data that was not included in the training sets (Figure 2.7).

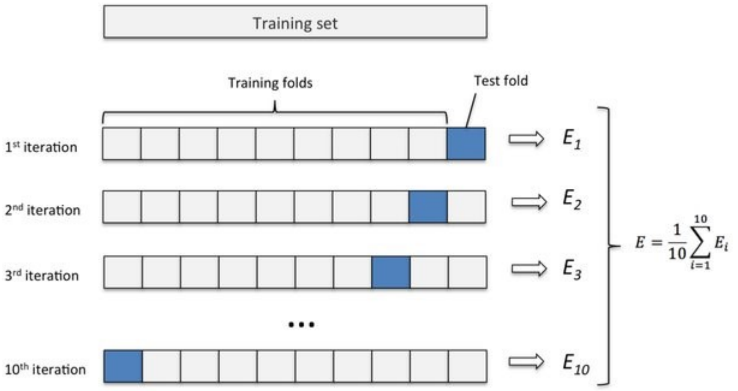


Figure 2.7: An example of a 10-Cross fold validation evaluation method.Adapted from Johar M. Ashfaque Maat et al *Introduction to Support Vector Machines and Kernel Methods*

By supplying several out-of-sample data examples that provide comparative statistical testing, cross-validation gives the possibility to examine the reliability of ML models.

While classification models are assessed by calculating various rations of correctly or incorrectly predicted instances, such as accuracy, recall, precision and F1 score, regression models are evaluated by measuring estimation errors such as the root mean square error (RMSE) or the mean absolute percentage error (MAPE).

In circumstances where prediction mistakes are linked to asymmetric cost structures, it is also usual to employ cost-sensitive metrics such average cost per projected observation [57], this is very usual in the financial market systems, where a cost of not seeing a fraudulent transaction are noticeably larger that the costs of misclassifying a non-fraudulent transaction. This is similar to what happens in the medical area, where the cost of do not classifying correctly a person with a fatal disease is way worst that classifying incorrectly (that is the reason why recall in the medical field is so important to evaluate the performance of the algorithms) [33].

It is appropriate to evaluate several models of various complexities, taking into account competing model classes as well as alternative versions of the same model class, in order to find a prediction model that is adequate for a given job. The number and kind of manually created or self-extracted features, the number of trainable parameters (such as network weights in ANNs) and type of learning processes, may all be used to describe how complicated a model really is.

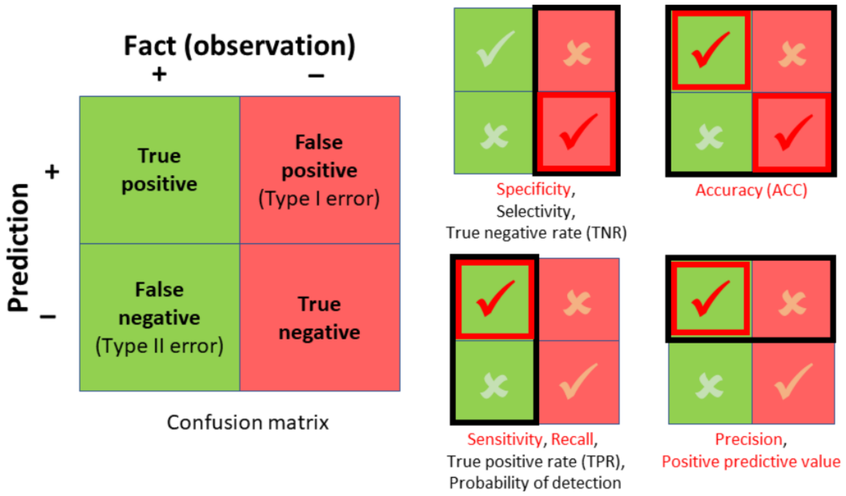

Fact (observation)
+
−
Prediction
+
−
True positive
False positive (Type I error)
False negative (Type II error)
True negative
Confusion matrix
Specificity, Selectivity, True negative rate (TNR)
Accuracy (ACC)
Sensitivity, Recall, True positive rate (TPR), Probability of detection
Precision, Positive predictive value

Figure 2.8: Confusion Matrix and respective metrics.Adapted from Adapted from Gabor Hrasko

Simpler models typically are not flexible enough to capture the non-linearity regularities and patterns important to learning process. On the other side, too complicated models carry a larger risk of overfitting, more ambiguous and computationally more costly. Memory requirements and the inference time to run a model on fresh data are two ways to represent computational costs. When judging deep neural networks, where a lot of parameters may be processed and stored, which places unique demands on hardware resources, these criteria are especially crucial. As a result, it is essential for business settings with limited resources, to choose a model that is not just at the ideal level of underfitting but also not too overfitting, thus creating a balanced workflow between the complexity of the problem, the complexity of the model and the resources that are used for running the model (accuracy vs memory consumption and speed) [58].

## 2.6 Bias and Drift on the data

When developing automated analytical models, is very important to be assure of the quality of the data, this comes from the idea of the cognitive biases from the human-generated data that can be introduced in any ML or DL algorithm. The definition of a cognitive bias is an incorrect assumption or opinion that people hold as a result of an inaccurate factual reporting or poor decision-making [55].

Data-introduced bias is not a brand-new idea, but when associated in the context of ML and DL, can have more implications. When training, if the data is improperly chosen or pre-processed, contains class imbalances or is not adequately evaluated after conclusions can amplify the bias done by the humans. Some examples include Google's Vision AI that produced different image labels based on skin tone or Amazon's AI recruiting software that displayed discrimination against woman [33].

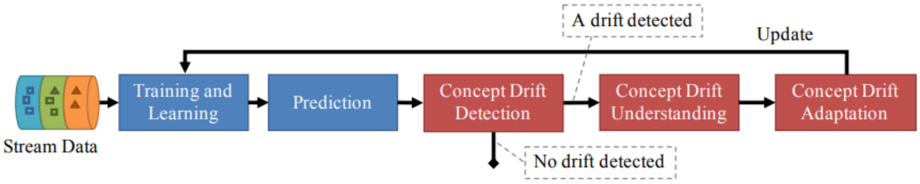


Figure 2.9: Framework for handling concept drift in machine learning.Adapted. Adapted from Jiu Lu et al. *Learning under Concept Drift: A Review*

Assuming that all bias effects can be explained in large datasets with high-dimensional data is unrealistic. However, it is crucial to identify and emphasize those effects that have an impact on predictions in order to better understand and have faith in ML models. A trained model is never complete since it is reasonable to suppose that persistent drift occurs in any real-world application. Companies must have measures in place to recognize, monitor and combat idea drift that affects the accuracy of the judgments made by their intelligent systems. Manual inspections and recurring model retraining with new data, are the norm right now.

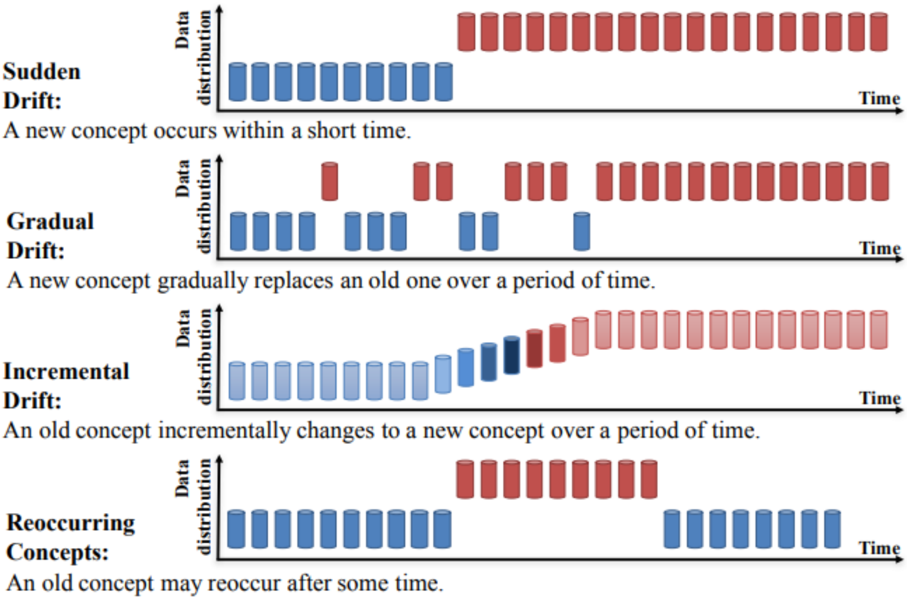

Sudden Drift:
Data distribution
Time
A new concept occurs within a short time.
Gradual Drift:
Data distribution
Time
A new concept gradually replaces an old one over a period of time.
Incremental Drift:
Data distribution
Time
An old concept incrementally changes to a new concept over a period of time.
Reoccurring Concepts:
Data distribution
Time
An old concept may reoccur after some time.

Figure 2.10: Types of drift that can affect the data.Adapted from. Adapted from Jiu Lu et al.*Learning under Concept Drift: A Review*

## 2.7 The Black-Box problem

It is hard to anticipate how DL models and some ML models will perform in a specific situation, this unpredictability is called "Black-Box" problem. Users might not be able to examine and comprehend the recommendations made by intelligent systems using these models. Additionally, this makes it extremely challenging to prepare for adversarial attacks, which can deceive and destroy DL models [59]. They may pose a risk to high-stakes applications such as disrupting traffic signs for autonomous vehicles [60]. Sometimes is important and necessary to explain the decision of the model.

Humans prefer straightforward explanation over complex ones, the goal for explainable AI (XAI) is to improve on current DL models by adding justifications for output predictions. In medical/biological field when searching for new drugs or trying to automatize the diagnosis of a specific disease, is important for the professionals to understand why the model get that conclusion, for future research on the topic. For image data, this entails highlighting the portions of the input image that led to a particular output choice [61]. Methods have been developed to emphasize the specific significant time steps that influence a forecast when dealing with time series [62]. Researchers need to examine the accountability and criticality of DL model applications in particular. They could need to choose for a white-box model that is easier to understand rather than a black-box model that is more precise [63] they might think about XAI augmentations to make the model's predictions easier for its users to understand [61].

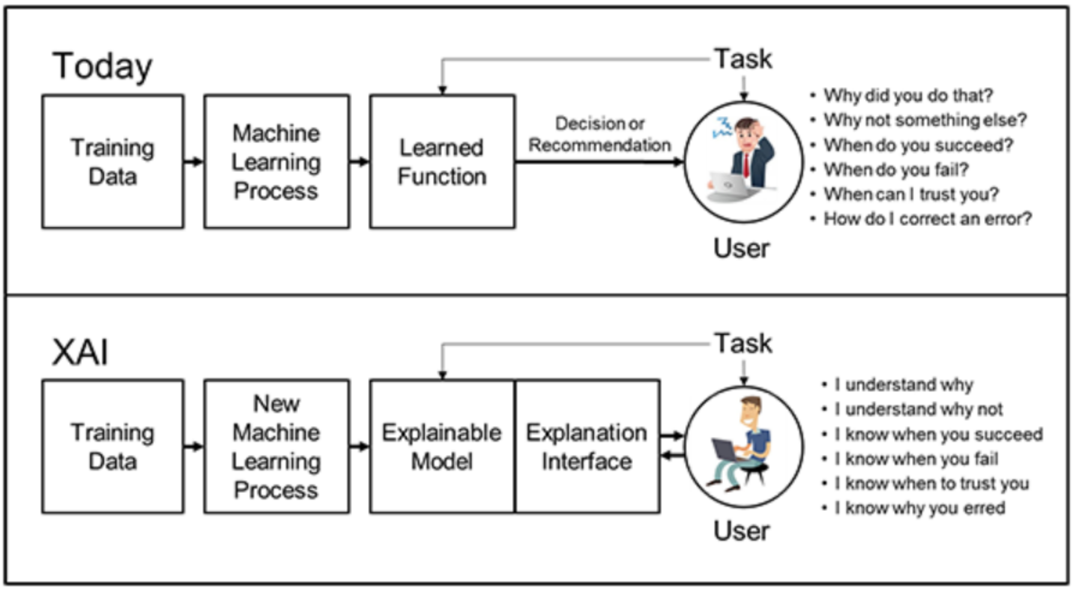


Figure 2.11: The importance of Explainable AI (XAI)- Source https://www.darpa.mil/program/explainable-artificial-intelligence

## 2.8 Summary

The chapter starts by introducing the concept of machine learning and its various categories, including supervised, unsupervised, reinforcement, and semi-supervised learning. These approaches are explained in terms of their ability to enable computers to learn patterns from data and make informed predictions or decisions. The chapter then moves on to deep learning, which is a subset of machine learning. It explores the architecture and capabilities of deep neural networks, emphasizing their capacity to learn hierarchical representations from complex data. The advantages of machine learning and deep learning in medical applications, such as disease diagnosis, medical image analysis, and drug discovery, are discussed. In particular, the chapter focuses on the challenges associated with handling medical images in deep learning models. It addresses issues related to large-scale medical image datasets, techniques for preprocessing the data, and the significance of appropriate training and validation strategies. The chapter also talks about the importance and what is in general the evaluation of the models. The chapter also delves into the topic of bias in models and how biases in data or algorithms can result in disparities in diagnosis or treatment. Finally, the chapter addresses the black box problem in deep learning, which refers to the challenge of interpreting the decision-making process of complex models. It explores the use of techniques like explainable artificial intelligence (XAI) to tackle this challenge. In summary, this chapter provides a comprehensive overview of machine learning and deep learning, their applications in the medical field, the challenges involved in handling medical images, the model evaluation and the issues of bias and the black box problem. It serves as a valuable resource for gaining an understanding of the fundamental concepts and implications of these technologies in medical settings.

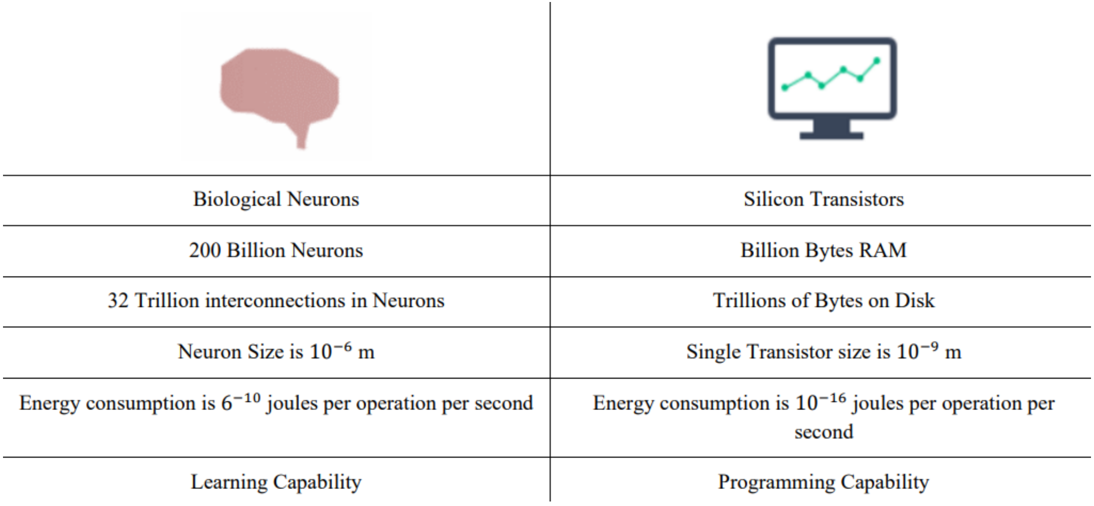
Chapter 3

# The anatomy of Deep Learning

## 3.1 ANN: The fundamentals

DL methods can be classified, like the ML methods, in 3 major categories: unsupervised, semi-supervised and supervised. Our goal in this chapter is not to talk deeply in these types of categories, since they are conceptually very similar to the homologous categories in ML methods, but more about the ANN basics and the different types of DL methods and their fundamentals (i.e., update of the weights, steps functions). Like said above, Biological neural networks (BNNs) served as the model for the algorithm-based Artificial Neural Networks (Table 3.1). ANNs attempt to abstract the enormous complexity of the actual, organic nervous system and concentrate on what may, in theory, matter most in terms of information processing [64].

Table 3.1: An Analogy of BNN and ANN.Adapted from Amey Thakur et al



| 🧠 | 🖥️ |
|---|---|
| Biological Neurons | Silicon Transistors |
| 200 Billion Neurons | Billion Bytes RAM |
| 32 Trillion interconnections in Neurons | Trillions of Bytes on Disk |
| Neuron Size is $10^{-6}$ m | Single Transistor size is $10^{-9}$ m |
| Energy consumption is $6^{-10}$ joules per operation per second | Energy consumption is $10^{-16}$ joules per operation per second |
| Learning Capability | Programming Capability |

We can characterize a neural network based on three crucial elements: its architecture, which determines the interconnections between neurons, the weight update technique used during training to optimize the algorithm, and, equally important, the activation functions employed. At the core of this intricate system lies the perceptron, depicted in Figure 3.1. The perceptron takes multiple binary inputs and produces a single binary output. The weights assigned, represent the importance, to the inputs received by each perceptron. Consequently, the perceptron's output is determined by the weighted sum of its input features. Figure 3.1 illustrates an instance of a perceptron with three inputs (x1, x2, x3) and a single output weight.

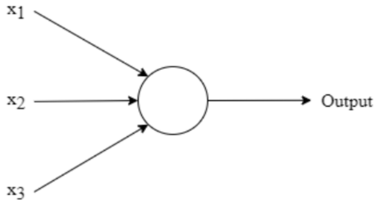


Figure 3.1: An example of a single perceptron (inputs and output).Adapted from Amey Thakur et al. *Fundamentals of Neural Networks*

The output of the neuron is either 0 or 1, this is going to be decided by whether the weighted sum is less or larger than some threshold number

$$output = \begin{cases} 0 \ \ if \sum w_j x_j \leq threshold \\ 1 \ \ if \sum w_j x_j > threshold \end{cases} \tag{3.1}$$

We can simplify this equation by expressing $\sum_j w_j x_j$ as a dot product, where $W$ and $X$ are vectors, whose components are the weights and inputs, respectively and shifting the threshold to opposite side of the inequality and replacing it with the perceptron's bias, $b$.

$$output = \begin{cases} 0 \ \ if \ \ w.x + b \leq 0 \\ 1 \ \ if \ \ w.x + b > 0 \end{cases} \tag{3.2}$$

For example, if we got 3 inputs, our equation will be something like, $X.W = x1w1 + x2w2 + x3w3 + b$. Conceptually, we can summarize the dot product of $X.W$ like being the significance and the importance ($W$) of each feature ($X$). The term $b$ is important to produce a shift to the output towards a desired range or to an specific value.

By creating numerous layers and incorporating numerous neurons, we can develop highly nonlinear and abstract systems that possess greater capabilities for handling complex problems. This is referred to as Deep Neural Networks, which mirrors the functioning of the human brain. The brain's vast number of layers and synapses enables it to not only receive sensory inputs but also process and generate responses, be they emotional or physical. In a neural network, the output layer nodes are interconnected with the input layer nodes, while the hidden layer nodes are interconnected with the input layer nodes. Each connection allows for signal transmission to neighboring neurons, forming a typical feedforward network. The hidden layer receives and processes the raw data from the input layer, and once the information is acquired, it is passed to the output layer. The output layer further analyzes the data from the hidden layer and produces the final output [13].

The signal transmitted through each connection is represented by a real number, while the output of each neuron is determined by a non-linear function applied to the sum of its inputs. As the learning process unfolds, the weights of neurons and connections are modified. These weights influence the strength of the signal transmitted through a connection[13].

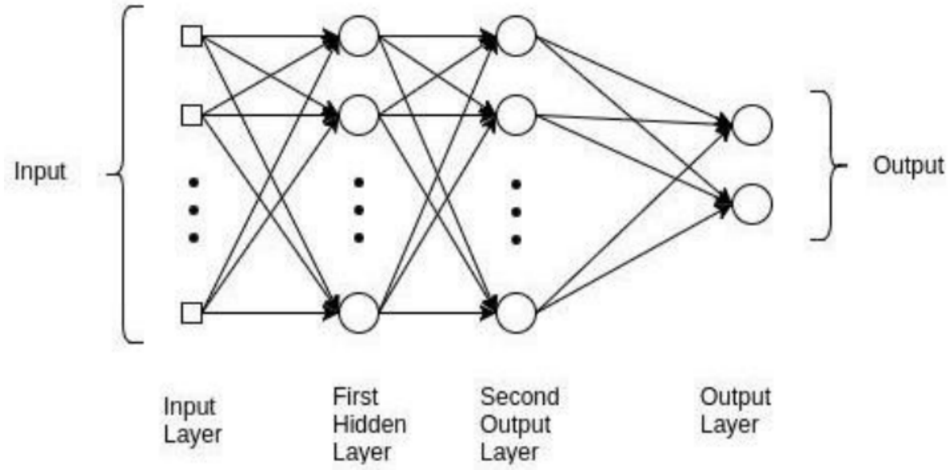


Figure 3.2: Multilayer Perceptron Adapted from Amey Thakur et al*Fundamentals of Neural Networks*

## 3.2 The importance of the architecture

An input layer, an output layer and one or more hidden layers are typically present in a neural network. Every neuron has an influence on every other neuron, making them interconnected. The network is able to recognize and keep track of every component of the dataset, as well as any potential relationship between the various bits of data. Neural networks are able to recognize extremely complex patterns in vast volumes of data in this way [64].

The transmission of information can take place in two different manners. In a feedforward network (Figure 3.3), signals flow unidirectionally towards the output layer. On the other hand, feedback networks operate using recurrent or interactive connections, allowing them to utilize their internal state or memory to process a series of inputs (Figure 3.3). These networks can facilitate bidirectional signal flow through loops within the hidden layers. Feedback networks are commonly employed in tasks that necessitate a specific sequence of events, such as chatbots that utilize recurrent neural networks (RNNs).

In a typical feedforward network, the output of the hidden layers, along with their respective weights, activation functions, and cost functions, is forwarded to the output layer. The strength of connections between neurons is expressed using numerical values, known as weights, which play a crucial role in the learning process of a neural network. During learning, these weights are initially assigned randomly and then adjusted through the learning process. The activation function is another important factor that influences the behavior and magnitude of the network's

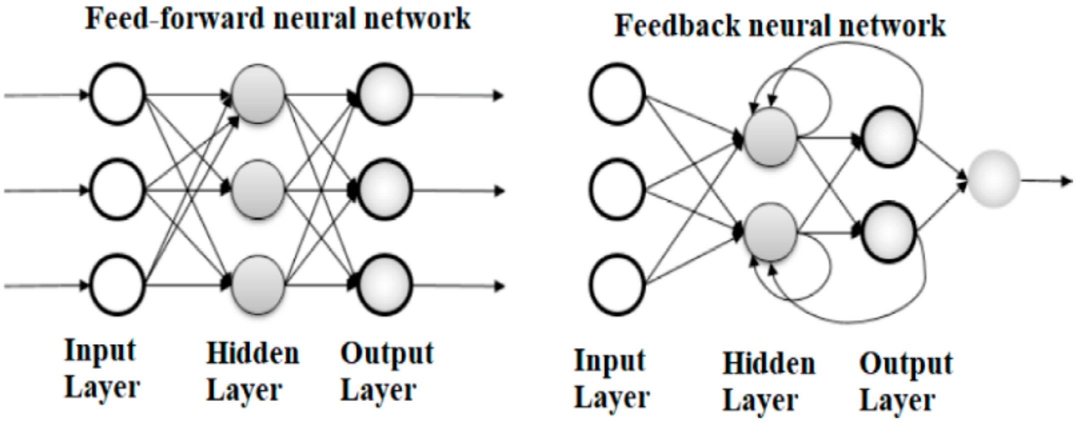


Figure 3.3: Right: Feedback neural network architecture / Left: Feedfoward neural network architecture.Adapted from Ali Y. Al-Bakri et al *Application of Artificial Neural Network (ANN) for Prediction and Optimization of Blast-Induced Impacts*

output. Activation functions are mathematical formulas used to compute the outputs of the neural network [13]. The data is standardized to ensure consistency across neurons, and this normalization process aids in achieving output values within the desired range, such as 0 to 1 with a sigmoid activation function or -1 to 1 with a tangent activation function.

The structure or architecture of neural networks plays a crucial role as it determines the final output. The connectivity pattern between neurons and the choice of activation functions greatly influence the network's behavior and performance.

## 3.3 Activations functions

Without the activation function, the output of a neural network would simply be a linear combination of inputs. There are several activations functions. The most common are, Sigmoid functions, Relu (Rectified Linear Unit), Tanh (Hyperbolic tangent) and Softmax.

The Sigmoid function receives any input between 0 and 1, and the output is provided by $(w.x + b)$ where $\sigma$ is known as the sigmoid function [13]. As a result, the output of a sigmoid neuron with inputs $x1$, $x2\ldots$, weights $w1$, $w2\ldots.$ and bias, equals to z = $\frac{1}{1+e^{-x}}$, this will map any input into a value between 0 and 1. It is commonly used in binary classification problems (Figure 3.4).

ReLu, which stands for Rectified Linear Unit, is a linear function that will output the input directly if is positive, else will output zero. Because it is way faster to train and typically produces better results, it has become the default activation function for many neural networks. It is provided by $f(x) = max(0, x)$, where x is the neuron's input (Figure 3.4) [65].

Tanh (Hyperbolic tangent), is similar to the sigmoid function, but it maps the input to a value between -1 and 1.

Softmax function, maps a vector of inputs to a probability distribution over the classes. Is typically used in multi-class classification problem in the output layer.

The step function is one of the simplest activation functions used in neural networks. It takes an input value and return an output of either 0 or 1, depending on whether the input is less than or greater that a certain threshold value, used in binary classification problems (Figure 3.4).

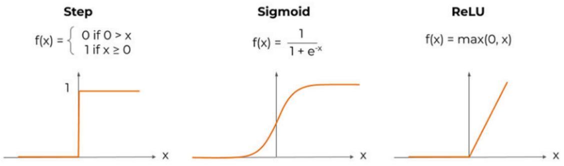


Figure 3.4: Left: Step function/ Middle: Sigmoid function / Right: ReLu function-Adapted from Ana Barragán-Montero et al. *Artificial intelligence and machine learning for medical imaging: a technology review*

## 3.4 The use of Gradient Descent and Backpropagation

An optimization technique commonly used to update weights based on the errors they generate is called gradient descent. In the context of neural networks, the focus is on the connection between the network's error and a single weight, specifically how modifying the weight affects the error. The goal is to find the weight value that minimizes the error. During the learning process, a neural network gradually adjusts multiple weights to accurately interpret signals. The relationship between the network's error and each weight can be expressed as a derivative $\frac{dE}{dW}$, indicating how a small change in weight corresponds to a small change in error. In a deep network with multiple complex transformations, each weight is just one component in a larger system. The weight signal accumulates across multiple layers and passes through activation functions. To identify the weight causing inaccuracies and understand its impact on overall performance, we need to traverse the network's activation's and outputs using the chain rule[3]. Chain rule states that:

$$\frac{dz}{dx} = \frac{dz}{dy} . \frac{dy}{dx} \tag{3.3}$$

Meaning that the relationship between the net's error and a single weight is:

$$\frac{dError}{dWeight} = \frac{dError}{dActivation} . \frac{dActivation}{dWeight} \tag{3.4}$$

In summary, when considering the variables Error and Weight, which are linked through the intermediary variable Activation, we can determine the impact of a weight change on the error by assessing the effect of an activation change on the error, followed by the influence of a weight change on the activation. This process is known as backpropagation[3].

By iteratively adjusting the parameters in the direction of the negative gradient, gradient descent can optimize the model's parameters to minimize the loss function. The weights are adjusted by the backpropagation and then used by the gradient descent to find the minimal local, this process is repeated until it reaches the minimum error.

## 3.5 Types of Deep Learning

As mentioned earlier, deep learning methods are advancing rapidly, and some of the most widely recognized ones include recursive neural networks (RvNNs), recurrent neural networks (RNNs), convolutional neural networks (CNNs), and generative adversarial networks (GANs).

Recursive neural networks (RvNNs) have the ability to make predictions in a hierarchical structure and classify outputs using compositional vectors. The RvNN architecture is specifically designed for handling objects with irregular structures, such as graphs or trees. It generates a fixed-width distributed representation from a variable-size recursive data structure. The network is trained using a technique called back-propagation through the system (BTS).

The BTS system utilizes a similar technique to the general back-propagation algorithm and is capable of handling tree-like structures. Auto-association is employed to train the network to reproduce the input pattern at the output layer.

In the context of natural language processing (NLP), Recursive neural networks (RvNNs) have demonstrated significant effectiveness.These authors showcase its application in categorizing natural language sentences, such as dividing each phrase into words or segmenting images into multiple regions of interest. RvNN constructs a syntactic tree and computes scores to determine the likelihood of merging pairs of units. Each pair's merge plausibility is evaluated by RvNN, and the pair with the highest score is combined into a composition vector. With each merge, RvNN generates a larger area consisting of multiple units, accompanied by a compositional vector for the area and a corresponding class label. The compositional vector representing the entire area serves as the root of the RvNN tree structure [2] (Figure3.5).

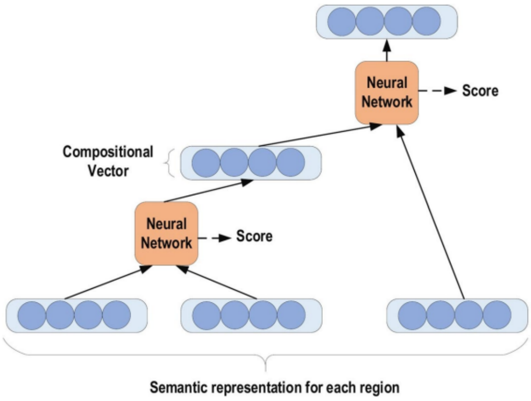


Figure 3.5: An example of RvNN tree.Adapted from Laith Alzubaidi et al.*Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*

In the text analysis field, Recurrent Neural Networks, are also an option (RNN). RNNs are widely used and well-known method [66]. Speech processing and NLP settings are where RNN is most commonly used. RNN employs sequential data in the network, as opposed to traditional networks. This property is essential to a variety of different applications because the inherent structure in the data's sequence provides useful information. For instance, it's critical to comprehend the sentence's context in order to interpret a particular word. RNN is though of as a short-term memory unit (STM). In the field of the neural networks, short-term memory refers to a type of RNN that can selectively remember or forget certain information over a short period of time. STM is important in processing sequential data, such as speech, text and video, where the current input depends on the previous inputs in the sequence. By maintaining a short-term memory of the recent inputs, an STM network can better predict the future outputs [67].

A typical unfolded RNN diagram for a particular input sequence is shown in next figure.

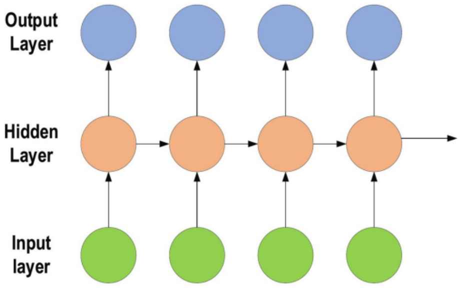


Figure 3.6: Typical unfolded RNN diagram. Source: https://devopedia.org/long-short-term-memory

One of the main challenges associated with this strategy is the problem of increasing gradient and vanishing gradients in RNNs. Specifically, during the training phase, a set of numerous derivatives may lead to the exponential explosion or decay of gradients. However, this sensitivity diminishes over time as the network starts disregarding the initial inputs when new ones are introduced. To address this issue, LSTM (Long-Short Term Memory) can be employed. LSTM provides recurrent connections to network memory blocks (Figure 3.7), each equipped with memory cells that store the network's temporal states and gated components for regulating information flow.

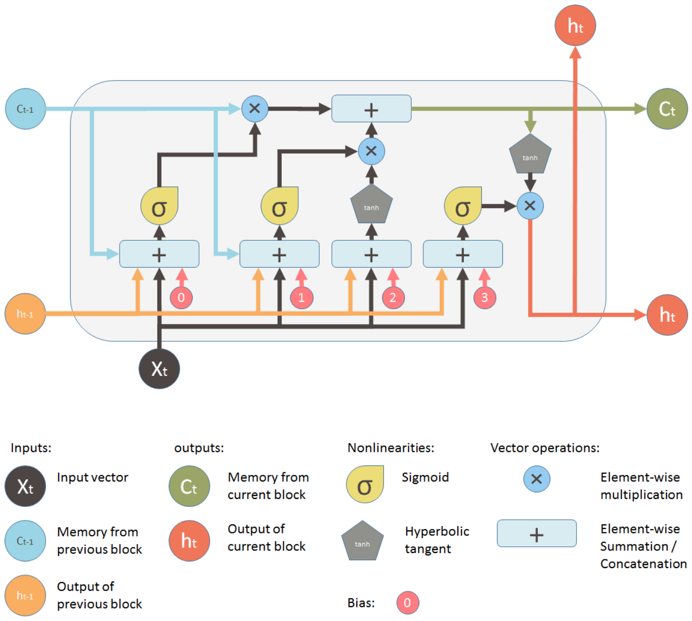


Figure 3.7: LSTM architecture. Source: //medium.com/@birla.deepak26/autoencoders-76bb49ae6a8f

Residual connections are particularly effective in mitigating the vanishing gradient problem in deep networks. Comparatively, CNNs are considered more powerful than RNNs, as they exhibit greater feature compatibility.

CNNs are primarily employed in computer vision and speech recognition tasks, aiming to autonomously and adaptively learn spatial hierarchies of features from input data. These networks comprise multiple layers, including convolutional pooling layers and fully connected layers. The convolutional layers facilitate feature extraction by applying filters to input images, convolving over the images, and generating feature maps. The pooling layers subsequently downsample the feature maps to reduce spatial dimensions in the output. Finally, the fully connected layers utilize the learned features to make final classification decisions.

CNNs have demonstrated exceptional efficacy across various computer vision tasks, such as image classification, object detection, and image segmentation. They are also extensively utilized in natural language processing (NLP) and other domains involving spatially-structured data [34].

CNNs are the building blocks for more complex image analysis systems like, Autoencoders (Figure 3.8). Autoencoders offer a detailed feature representation of the input data. They may, however be used with any kind of input. In a Autoencoder architecture, the input is often compressed into a low-dimensional representation during the encoding step and the network attempts to reconstruct the original input using the learnt characteristics during the decoding stage (having and encoder and a decoder). This forces the network to ignore irrelevant noise and retain useful information in the latent representation [34].

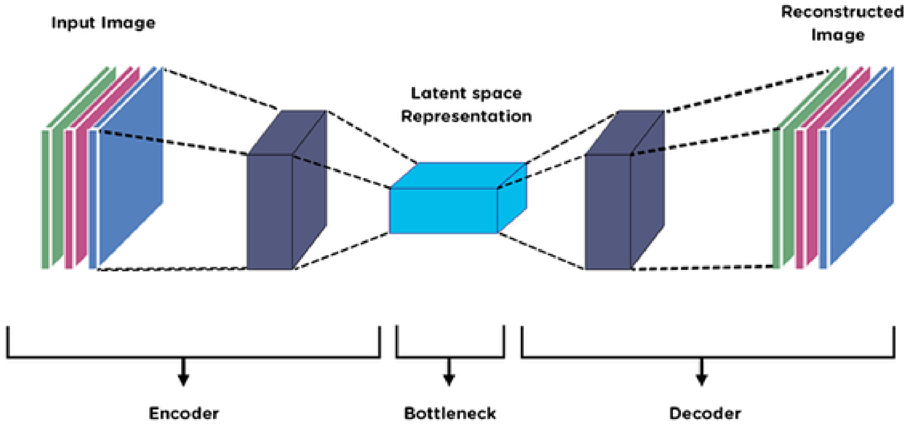


Figure 3.8: A typical Autoencoder architecture. Source: https://www.analyticsvidhya.com/blog/2019/04/top-5-interesting-applications-gans-deep-learning/

GANs, which stands for Generative Adversarial Networks, represent a cutting-edge architectural approach for generating synthetic data. These networks belong to the category of generative models and aim to learn the probability distribution of a given training dataset. The goal is to enable the network to generate new data samples that exhibit randomness and variation. GANs consist of two interconnected sub-networks that compete with each other. The first network is known as the generator network, which learns and models the distribution of the input data to produce fresh samples. The second network is the discriminator network, whose task is to distinguish between naturally occurring data samples and intentionally generated ones.

Through a non-cooperative zero-sum game, where one network's improvement comes at the expense of the other, both networks are simultaneously trained. The training continues until the discriminator becomes unable to differentiate between the two types of data. Essentially, the generator and the discriminator engage in a competitive process, driving the generator to produce artificially generated images that closely resemble real ones in order to deceive the discriminator. GANs have found significant applications in the fields of art and "deep-fakes" image generation. GANs play a crucial role in domains such as medicine, where ethical concerns surrounding the use of patient images are prominent. Various types of GANs exist, including conditional GANs (cGANs), cycleGANs, and Pix2Pix (image translation). In this particular work, the focus will primarily be on the Pix2Pix approach.

## 3.6 Summary

The link between neural networks and the brain as well as the basic principles of neural networks are thoroughly discussed in this chapter. We explore how neural networks draw inspiration

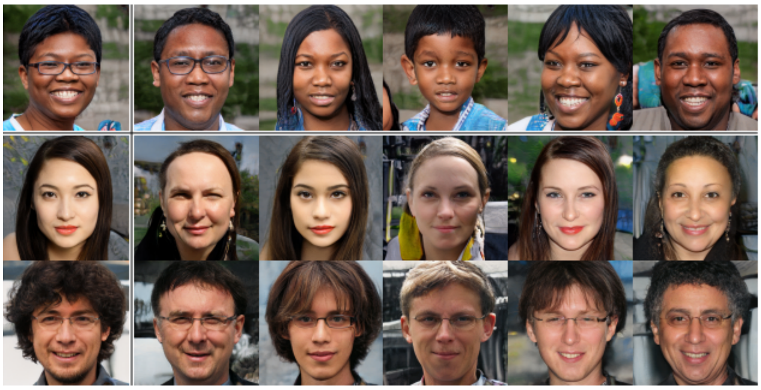


Figure 3.9: Deep-Fake face imagesAdapted from Christian Janiesch et al. *Machine learning and deep learning*

from the biological architecture of the brain and highlight the shared characteristics, such as the ability to learn and adapt through interconnected neurons. The chapter covers the fundamental components of neural networks, including neurons, activation functions, and layers. The discussion primarily focuses on the widely used feedforward neural network architecture. Throughout the chapter, there is a emphasize of the practical applications of neural networks in diverse fields like computer vision, natural language processing, and robotics. Overall, this chapter offers a valuable resource for gaining an understanding of the principles and real-world applications of neural networks in various domains.

# Chapter 4

# Exploring Key architectures: CNN and GANS

CNN is widely recognized as the most popular and extensively used algorithm in the field of deep learning (DL). One of the fundamental advantages of CNN is its ability to automatically identify relevant elements without requiring human intervention. CNNs have found applications in various domains such as computer vision, speech processing, and face recognition. Inspired by the biological architecture of the brain, CNNs share similarities with traditional neural networks. Goodfellow [34] first highlighted the key advantages of CNNs, which include similar representations, sparse interactions, and parameter sharing. Unlike typical fully connected (FC) networks, CNNs utilize shared weights and local connections to effectively utilize 2D input-data structures, particularly in visual signals. This technique reduces the number of parameters required, leading to faster network performance and easier training. In essence, CNN algorithms exhibit spatially local correlation in the input, functioning as local filters. Typically, CNN architectures resemble multi-layer perceptrons with multiple convolution layers preceding sub-sampling (pooling) layers and FC layers at the end, as depicted in figure 4.1.

In a CNN model, the input $X$ is structured in three dimensions: height, width, and depth ($m$ x $n$ x $r$). The height ($m$) and width ($n$) represent the spatial dimensions of the input, while the depth ($r$) refers to the number of channels, such as the RGB channels in an image, which is typically 3. Each convolutional layer in the CNN contains multiple kernels (filters), denoted as $K$, which also have three dimensions ($n$ x $n$ x $q$). These kernels serve as local connections and share the same set of parameters (bias and weights) to generate $k$ feature maps ($h$) with a specific size ($m$ - $n$ + 1) each. The feature maps are obtained by convolving the kernels with the input. Applying the nonlinearity or an activation function to the convolution-layer output, we get:

$$h^k = f(W^k * x + b^k) \tag{4.1}$$

In the CNN architecture, the feature maps are obtained by applying the weights ($W$) and bias

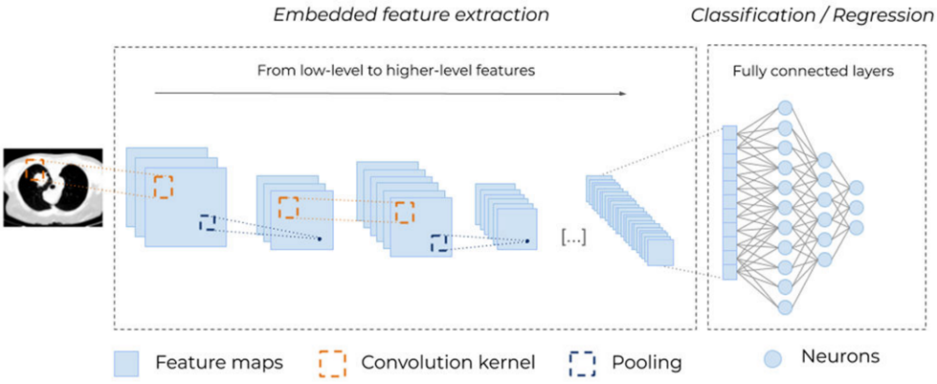

Embedded feature extraction
Classification / Regression
From low-level to higher-level features
Fully connected layers
[...]
Feature maps
Convolution kernel
Pooling
Neurons

Figure 4.1: Typical architecture for a (deep) Convolutional Neural Network (CNN). Adapted from Ana Barragán-Montero et al.*Artificial intelligence and machine learning for medical imaging: a technology review*

($b$) to the input ($x$) using activation functions ($f$). The number of feature maps ($h$) determines the output dimensionality.

Following this, the feature maps undergo down-sampling in the sub-sampling layers. This reduces the network parameters, speeds up the training process, and helps address overfitting issues. Each feature map is subjected to a pooling function (such as mean or max pooling) over a surrounding region of size $p$ x $p$, where $p$ represents the kernel size. Once the mid and low-level characteristics have been obtained, the fully connected (FC) layers generate high-level abstractions, which correspond to the final layers in a conventional neural network. The feature extraction layers and classification layer of the CNN are responsible for learning and organizing the extracted features, resulting in a highly structured model output. Additionally, CNNs offer advantages in terms of implementing large-scale networks compared to other types of neural networks, making their implementation more convenient.

The convolutional layer is a crucial component of CNN architecture, comprising multiple convolutional filters. These kernels convolve with the input image, which is represented as an N-dimensional matrix, to generate an output feature map.

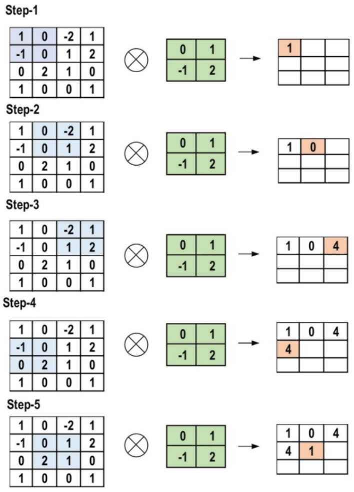
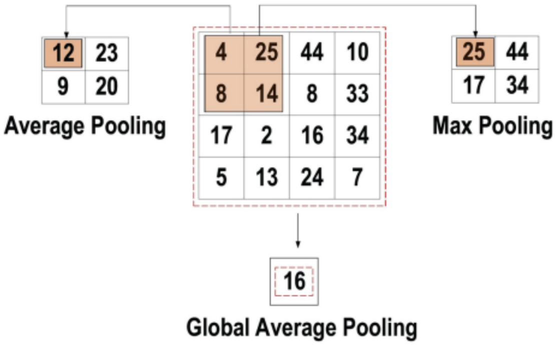


Figure 4.2: - The primary calculations executed at each step of convolutional layer.Adapted from Christian Janiesch et al. *Machine learning and deep learning*

In Figure 4.2, the feature map is obtained using a stride of 1 (moving 1 to the right) and a kernel of size 2x2. The right side of the figure shows the resulting feature map, while the steps for obtaining that feature map (convolutions) are illustrated. Padding is an important factor in determining the size of the input image's border. The border-side features tend to change rapidly. By applying padding, the input image size is increased, resulting in a larger feature map being produced. One of the key advantages of convolutional layers is their sparse connectivity. In a CNN, there are only a few weights or connections between adjacent layers. This leads to a smaller number of required weights and reduced memory storage for these weights. Weight sharing is another advantage, where no specific weights are allocated between neurons in neighboring layers. Instead, the entire set of weights operates on all pixels of the input matrix. Learning a single set of weights for the entire input significantly reduces training time and associated costs, as there is no need to learn additional weights for each neuron. The primary function of the pooling layer is the sub-sampling of the feature maps. The convolutional procedures are used to create these maps, in other words, using this method results in the creation of smaller versions of large-scale feature maps. At the same time, it keeps most of the dominating data (or characteristics) during the whole pooling stage. Both the stride and the kernel are originally size-assigned before the pooling process is carried out, much like the convolutional procedure. There are several pooling strategies that may be used in different pooling levels.

The primary function of the pooling layer is the sub-sampling of the feature map. The convolutional procedures are used to create these maps, in other words, using this method results in the creation of smaller versions of large-scale feature maps. At the same time, it keeps most of the dominating data (or characteristics) during the whole pooling stage. Both the stride and the kernel are originally size-assigned before the pooling process is carried out, much like the convolutional procedure. There are several pooling strategies that may be used in different pooling levels. These techniques include global average pooling (GAP), global max pooling and gated pooling. The max, min and GAP pooling techniques are the most well-know and widely used (Figure 4.3).



| | |
|---|---|
| 12 | 23 |
| 9 | 20 |

| | | | |
|---|---|---|---|
| 4 | 25 | 44 | 10 |
| 8 | 14 | 8 | 33 |
| 17 | 2 | 16 | 34 |
| 5 | 13 | 24 | 7 |

| | |
|---|---|
| 25 | 44 |
| 17 | 34 |

| |
|---|
| 16 |

Figure 4.3: Three types of pooling operations.Adapted from Laith Alzubaidi et al. *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*

The fully connected layer commonly located at the end of each CNN architecture. Inside this layer, each neuron is connected to all neurons of the previous layer, the so-called fully connected approach, it is utilized as the CNN classifier. It follows the basic method of the conventional multiple-layer perceptron neural network, as it is a type of feed-forward ANN. The input of the FC layer comes from the last pooling or convolutional layer. This input is in the form of a vector, which is created from the feature maps after flattening, the output of the FC layer represents the final CNN output [67].

The non-linear performance of the activation layers means that the mapping of input to output will be non-linear. Moreover, these layers give the CNN the ability to learn extra complicated things. The activation functions must also have the ability to differentiate which is an extremely significant feature, as it allows error back-propagation to be used to train the network [67].

Several well-known activation functions have been discussed previously, such as sigmoid, hyperbolic tangent, and ReLU. However, in CNN architectures, variations of the ReLU function are commonly used, such as LeakyReLU. Unlike ReLU, which scales down negative inputs to zero, LeakyReLU ensures that negative inputs are not completely ignored. This activation function is employed to address the issue known as "Dying ReLU," which occurs when the weights and biases of a neuron are adjusted in such a way that the neuron always produces negative outputs, causing ReLU to output zero consistently. Mathematically, this can be represented as:

$$f(x)LeakyRelu = \begin{cases} x, & \text{if } x > 0 \\ mx, & \text{if } x \leq 0 \end{cases} \tag{4.2}$$

The leak factor is denoted by $m$. It is commonly set to a very small value.[67]. The Noisy ReLu function employs a Gaussian distribution to make ReLu noisy. By adding noise to the output of the function, Noisy ReLu can ensure that a small number of units continue to receive non-zero gradients during training, which can improve the performance and stability of the network. It can be represented as:

$$f(x)Leakyrelu = max(x + Y), with\ Y \sim N(0, \sigma(x)))$$

Parametric Linear Units, is very similar to the LeakyReLu, the main difference is that the leak factor in this function is updated through the model training process.

$$f(x)Parametrization = \begin{cases} x,\ if\ x > 0 \\ \ ax,\ x \leq 0 \end{cases} \tag{4.3}$$

The main goal for any ML/DL algorithm is to minimize the loss function. The output layer of the CNN model uses a few loss functions to determine the anticipated error produced over the training samples. This mistake highlights the discrepancy between the actual and expected output. Will be then improved using CNN's learning method. The loss function however, uses two parameters to determine the mistake. The first parameter is the predicted output or anticipated output of CNN. The second argument is the actual output, sometimes referred to as the label. Different problems use distinct loss function types.

The Cross-Entropy or Softmax loss function is a widely used measure of performance in CNNs, and it is also known as the log loss function. It produces a probability value ($p$) ranging from 0 to 1. This loss function is commonly used instead of the squared error loss function in multi-class classification tasks. The softmax activation function is typically applied in the output layer to generate the output as a probability distribution. The mathematical representation of the probability of each output class is shown in the following equation.

$$pi = \frac{e^{a_i}}{\sum_k^N e_k^a, with\ k = 1} \tag{4.4}$$

Here, the numerator represents the non-normalized output from the preceding layer, while the denominator represents the number of neurons in the output layer, Finally the mathematical approach of cross-entropy loss functions is:

$$H(p, y) = -\sum y_i log(p_i)\ where\ i\ \epsilon\ [1, N] \tag{4.5}$$

Another loss function is the Euclidean loss function [67], this function is widely used in regression problems. The mathematical expression of the estimated Euclidean loss is:

$$H(p, y) = \frac{1}{2N} \sum_{i=1}^{N} (p_i - y_i)^2 \tag{4.6}$$

Finally, the hinge loss function is frequently utilized in binary classification problems, particularly in the context of maximum-margin-based classification. This is particularly relevant for Support Vector Machines (SVMs). In this case, the optimizer aims to maximize the margin between the two distinct classes in the dual objective.

$$H(p, y) = \frac{1}{2N} \sum_{i=1}^{N} max(0, m - (2y_i - 1)p_i) \tag{4.7}$$

The margin m is commonly set to 1. Moreover, the predicted output is denoted as $p_i$, while the desired output is denoted as $y_i$.

The primary challenge in creating CNN models with robust generalization is the issue of overfitting. Overfitting occurs when the model demonstrates exceptional performance on the training data but fails to perform well on unseen test data. Conversely, an underfitting model arises when it inadequately learns from the training data. A well-fitted model, on the other hand, performs effectively on both the training and testing datasets. Consequently, there exist various methods to address the problem of overfitting. Dropout is a widely employed technique that randomly deactivates neurons during each training session. This strategy compels the model to learn from a multitude of independent features and evenly distributes the responsibility of feature selection across all neurons. The deactivated neurons do not contribute to either backward or forward propagation during the training phase. However, during prediction using the testing set, the full-scale network is utilized.

Another similar approach is the weight drop, the main distinction between weight drops and dropout is that after each training epoch, the connections between neurons (weights) are discarded rather than the actual neurons themselves. Increasing the data is always a good option. This is the basis for the simplest technique to prevent overfitting. We can sample more data or using different types of data augmentation to increase the sample. The training dataset's size is artificially increased using a variety of approaches [67].

Batch normalization, ensures the effectiveness of the output activation by employing a univariate Gaussian distribution. This involves normalizing the output at each layer by subtracting the mean and dividing it by the standard deviation. It also serves to mitigate the impact of "internal covariance shift" within the activation layers. The internal covariance shift refers to the variation in the activation distribution within each layer, which can significantly increase during the weight update process if the training data samples come from diverse sources. However, incorporating batch normalization in the CNN architecture helps address this issue by reducing the convergence time and mitigating the problem of vanishing gradient. It also allows for better control of weight initialization and helps prevent overfitting by reducing changes during training.

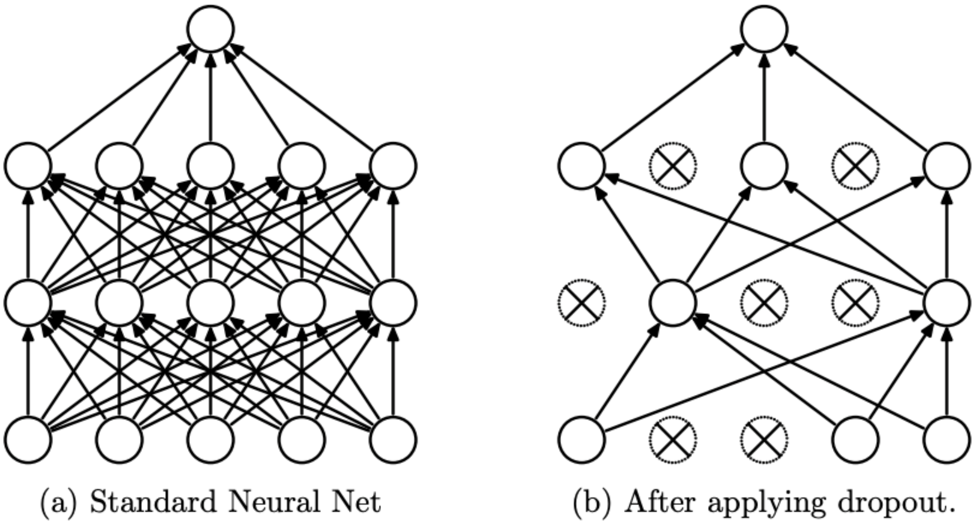


Figure 4.4: - The difference between a normal NN and after using dropout. Source : https://paperswithcode.com/method/dropout

In section 3, we discussed the approach for designing the architecture of neural networks to learn weights, specifically focusing on the gradient descent and backpropagation algorithm. The main objective of gradient descent is to minimize the training error by updating the network parameters during each training epoch. This involves calculating the gradient (slope) of the objective function using a first-order derivative with respect to the network parameters. The parameter is then updated in the opposite direction of the gradient to reduce the error. The backpropagation process involves propagating the gradient from each neuron to all neurons in the previous layer. The learning rate, which determines the step size of parameter updating, plays a crucial role in the learning process and needs to be chosen carefully. It is important to select an appropriate learning rate to avoid negatively impacting the learning process, as it is a hyperparameter.

When using batch gradient descent (BGD), the network parameters are updated by considering the entire training dataset together. In other words, the gradient is calculated using all the training samples, and this gradient is then used to adjust the parameters. This approach allows the CNN model to converge faster and provides a more stable gradient, particularly for small datasets. However, it's important to note that using BGD requires significant computational resources because the parameter updates are performed only once during each training cycle. On the other hand, when dealing with a large training dataset, convergence may take more time, and there is a possibility of converging to a local optimum rather than the global optimum. BGD is advantageous for small datasets due to its faster convergence and stable gradient, but it comes with increased resource requirements. Larger datasets may require more time for convergence and can be susceptible to local optima.

Stochastic gradient descent (SGD) is another optimization algorithm. It is a variation of BGD that updates the model's parameters after processing each individual training sample, rather than the entire dataset at once. This means that after randomly selecting and processing individual samples, the algorithm can escape from local minima and explore different areas of the parameter space. This make SGD particularly useful when dealing with large datasets.

Mini-batch gradient descent, combines the benefits of both batch gradient descent (BGD) and stochastic gradient descent (SGD). It provides a more stable gradient compared to SGD by considering a small group of samples with no overlap, which helps avoid the noisy updates of individual samples. At the same time, it is more computationally and memory-efficient than BGD, as it processes the data in smaller batches. However, mini-batch gradient descent has a constant convergence, meaning it may take more iterations to reach the minimum of the loss function compared to SGD. It also requires additional computational resources to process each mini-batch and may require more memory to store the gradients for parameter updates.

Adam optimizer (Adaptive Moment Estimation), is one of most widely optimizers used. Adam is an example of the most recent developments in deep learning optimization. The Hessian matrix, which uses a second-order derivative, serves as a representation for this. Adam is a learning technique that was created specifically for deep neural network training. Adam has the advantages of being more memory efficient and having less computational power. The way Adam works is by computing adaptative LR (learning rate) for each model parameter, it combines the advantages of RMSprop (Root Mean Squared Propagation) and momentum turn. Adam optimizer uses adaptive learning rates to update the model parameters during training. By doing this, it adjusts the learning rate for each weight separately based on its historical gradient information, making it a very effective and efficient optimization algorithm [67].

To finish the explanation of the fundamentals of CNN, is important to talk about some important architectures. ResNet, short for Residual Network, was developed to address the vanishing gradient problem encountered in earlier networks. The main objective was to create an exceptionally deep network that could effectively propagate gradients. ResNet introduced different variants based on the number of layers. One of the most popular variants is ResNet50, consisting of 49 convolutional layers and one fully connected (FC) layer. A key innovation of ResNet is the concept of a bypass pathway, also known as a skip connection. This idea allows the network to directly pass information from one layer to another, bypassing intermediate layers (Figure 4.5). This helps in preserving and propagating gradients effectively, enabling the network to train deep architectures more efficiently. ResNet is a groundbreaking CNN architecture designed to address the vanishing gradient problem by introducing skip connections. ResNet50, with its numerous convolutional layers and MAC operations, achieved notable success in various computer vision tasks.

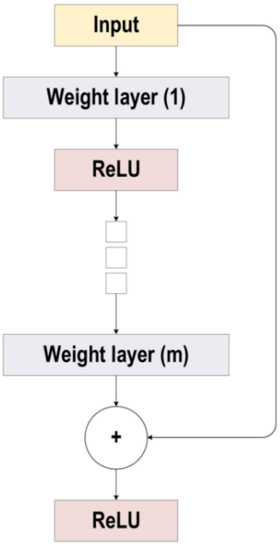


Figure 4.5: The block diagram for ResNet. Adapted from Laith Alzubaidi et al. *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*

This figure contains the fundamental ResNet block diagram. This is a conventional feedforward network plus a residual connection. These residual connections create the difference between the output of the layer and the input, which is then added back to the input to obtain the final output of the layer. The residual layer output can be identified as the ($I$ -1)$th$ outputs, which delivered from the preceding layer ($xl$ -1). After executing different operations, like convolution, using variable-size filters or batch normalization, the output is $F(xl$-1). The residual output is $xl$, which can be represented as:

$$x_l = F(x_l - 1) + x_l - 1 \tag{4.8}$$

In the medical images, U-Net is one of the most used architectures for doing segmentation. U-Net is a Fully Convolutional Network (FCN) applied to biomedical image segmentation, which is composed of the encoder, the bottleneck module and the decoder. The widely used U-Net meets the requirements of medical image segmentation for its U-shaped structure combined with context information, fast training speed and a small amount of data used [68].

In figure 4.6 is represented the architecture type of U-Net. In the left side of the U-shape is the encoding stage, also called contraction path with each layer consisting of two 3*3 convolutions with ReLu activation and a 2*2 maximum pooling layer. The right side of the U-Shape, also called expansion part, consists of the decoding stage and the upsampling process that is realized via 2*2 deconvolution to reduce the quantity of input channels by half. The U-Net also incorporates skip connections between the encoder and decoder layers to facilitate the transfer of low-level details between the layers. The skip connections enable the U-Net to produce segmentation masks with sharp and accurate boundaries.

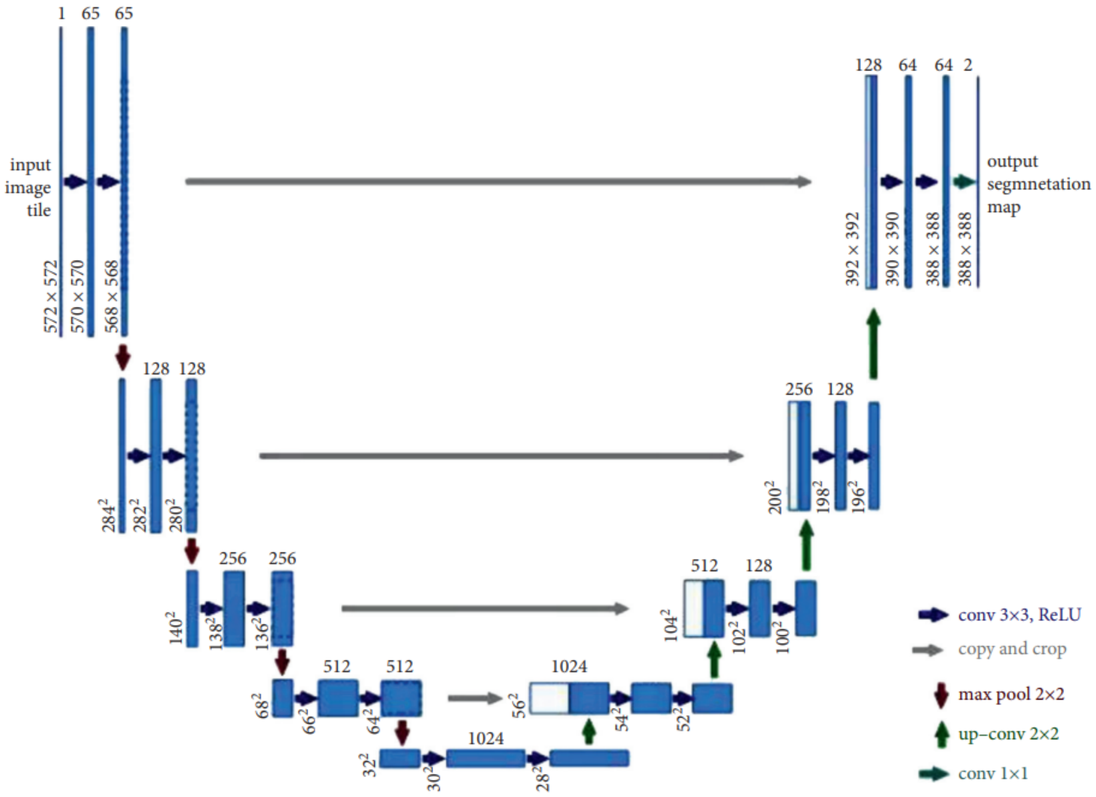


Figure 4.6: Illustration of U-Net convolution network structure. Adapted from Xiao-Xia et al. *U-Net-Based Medical Image Segmentation*

## 4.1 Generative adversarial Networks (GANs)

The discriminator (D) and generator (G) networks make up GANs. The idea is that G repeatedly tries to map a give data distribution in order to produce new data, which D tries to check if the generated data is true or not. G tends to minimize the loss between the two distributions, producing samples that comparable to the input data based on feedback from D. The idea is to get D to mistake created data for genuine data. While G learns to trick D, D is concurrently trained to better distinguish generated data from actual input data. Both networks are being trained to become better at their respective tasks. The discriminator and the generator are both CNN architecture types [2]. In particular, the conditional generative adversarial network (cGAN), a special kind of GANs, learns the conditional distribution of the source image $x$ given the target image $y$ and then performs image transference from one domain to another.

Among cGAN models, the Image-to-Image model performs the image-to-image transference from one domain to another concerning the given condition and it has become a widely recognized conditional image synthesis model (also known as PIX2PIX) [45].

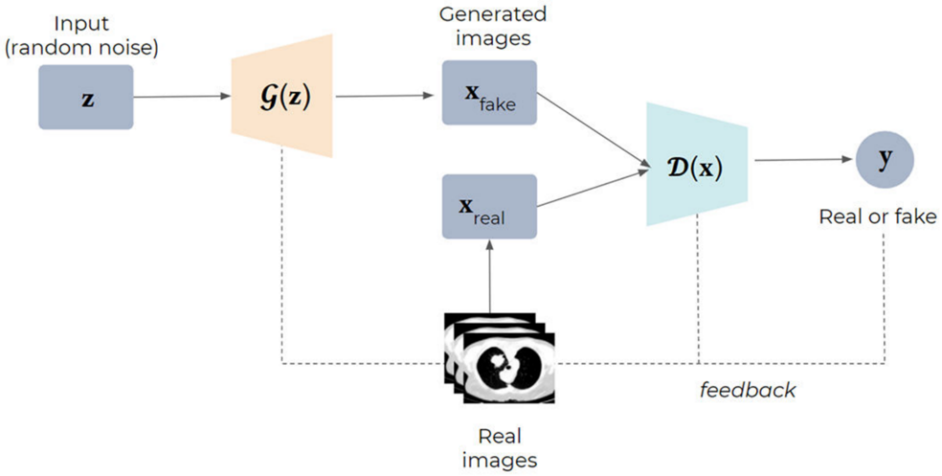

Input
(random noise)
z
$\mathcal{G}(\mathbf{z})$
Generated images
$\mathbf{x}_{\text{fake}}$
$\mathbf{x}_{\text{real}}$
$\mathcal{D}(\mathbf{x})$
y
Real or fake
feedback
Real images

Figure 4.7: Structure of Generative Adversarial Networks (GANs). Adapted from Ana Barragán-Montero et al. *Artificial intelligence and machine learning for medical imaging: a technology review*

To begin with, the generator receives a randomly generated noise vector $z$ and attempts to transform it into a generated sample called G(z). The generator is characterized by a function denoted as G(z; $\theta_G$), where $\theta_G$ represents the generator's parameters. The primary objective of the generator is to produce an image that appears as realistic as it can be [69]

The discriminator plays the role of the classifier, distinguishing between real and fake samples. Takes the input a sample (that can be real or generated) and outputs a probability indicating the likelihood of the same sample being real. It can be represented as a function D(x; $\theta_D$),where x is the input sample ( either real or generated) and $\theta_D$ represents the parameters of the discriminator network. The discriminator's main goal is to accurately classify real samples as real $(D(x) \approx 1)$ and generated samples as fake $(D(G(z)) \approx 0)$ [69]

The training procedure of GANs can be described as a game involving two players in a minimax setting. The generator's objective is to reduce the discriminator's capability to accurately classify the generated samples, while the discriminator's goal is to maximize its accuracy in classifying those samples. This competition between the two networks results in an adversarial learning process, where the generator strives to enhance its capacity to deceive the discriminator, and the discriminator aims to improve its ability to distinguish between real and fake samples [69]. It can be expressed as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{4.9}$$

Where $D$ represents the discriminator network, $G$ represents the generator network, $x$ represents real samples from the training data, $p_data(x)$ represents the data distribution, $z$ represents the random noise vector, $p_z(z)$ represents the noise distribution, $D(x)$ represents the discriminator's probability output for a real sample $x$ and $D(G(z))$ represents the discriminator's probability output for generated sample $G(z)$. Iteratively, the generator and discriminator are updated alternatively using gradient-based optimization algorithms [69].

## 4.2 Summary

This chapter presents a comprehensive overview of the fundamental concepts behind Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs) in the context of image generation. CNNs are a specialized type of neural network specifically designed to process visual data effectively by utilizing convolutional layers, pooling layers, and fully connected layers. These components enable CNNs to capture spatial relationships and hierarchical features within images. On the other hand, GANs are deep learning models that consist of two neural networks, namely the generator and discriminator, engaged in a competitive learning process. The generator's objective is to produce synthetic samples that closely resemble real images, while the discriminator's task is to differentiate between real and generated samples. This adversarial interplay drives the training process, leading to improvements in both networks. The integration of CNNs within GANs has revolutionized the field of image generation, enabling the creation of high-quality and realistic synthetic images. The chapter explains how CNNs are utilized in the architectures of both the generator and discriminator in GANs. This integration allows the generator to learn intricate patterns and structures from the input noise vector, while the discriminator learns to discriminate between real and generated images based on their distinctive features.

# Chapter 5

# B30f and B50f: An overview of the transformation problem

## 5.1 The Problem

One of the most widely used diagnostic imaging techniques is computed tomography (CT), which is frequently employed to evaluate anatomical tissue properties for disease treatment. CT scanners provide users the freedom to alter the acquisition and image reconstruction techniques to suit their unique clinical requirements, but doing so has drawbacks as well. While it enables physicians to capture critical image features towards personalized healthcare, it forms a barrier to analyzing CT image in a large scale, in that capturing CT images with non-standardized image protocols may result in inconsistent radiomic features [70]. Was revealed in a recent study, both intra-CT (by changing CT acquisition parameters) and inter-CT (by comparing different scanners with the same acquisition parameters) test have demonstrated low reproducibility regarding radiomic features, such as intensity, shape and texture for CT imaging [71].

CT protocol has an enormous influence in the results of the radiomic approach. Till today there is no recommendation for standardized chest CT protocol for evaluation of the radiomics of lung cancer. This limits the ability to compare studies. In relation to the CT protocol, one of the most important technical parameters is the reconstruction kernel. The reconstruction filter affects the distribution of pixel values or the noise pattern of a region of interest (ROI) while the mean pixel value remains relatively unchanged [72]. Given that the interpixel relationship is affected by different reconstruction kernels, measures using a radiomic approach will also change. This issue becomes a big problem when the use of algorithms for detecting lung cancer, fibrosis or other type of lung diseases are trained in specific kernels, with specific spatial pixel relationships [73].

There is a trade-off between the spatial resolution and noise for different kernels. Sharp kernels (B50f) maintain a high spatial resolution while also resulting in a noisier image. Smooth kernels (B30f), reduce noise at the cost of lowering spatial resolution. This tradeoff is fundamental not just for the optimal viewing of anatomical regions but for taking radiomic features [74].

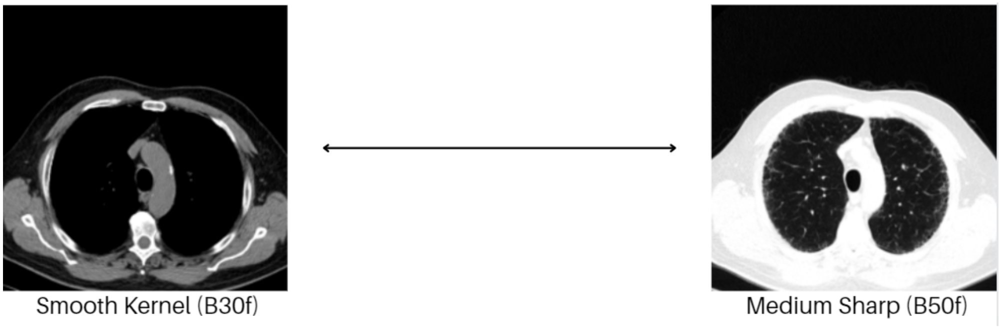


Figure 5.1: The main goal of the thesis, a bidirectional domain transformation of the CT reconstruction Kernels

## 5.2 Literature review

Kim et al [75], showed that using a convolutional neural network (CNN) to learn the differences between high-resolution and low-resolution images, the low-resolution images could be converted to high resolution images accurately and quickly. Recently, Gallardo-Estrella [11] demonstrated that normalization of CT data with different kernels using energy coefficients reduced variation during emphysema quantification. Ling G [76], used an optimized GAN architecture, called GANai, to create synthetic CT images. cGANs (a predecessor of PIX2PIX) is being widely used in medical field areas to create artificial images and in domain transfer, using paired data. For unpaired data cycleGANs [77], are way more used, due to its bidirectional mapping between two domains using unpaired input data. Shizuo et al [78], conclude that the use of DLL methods like cycleGANs or PIX2PIX approach is important for generating artificial images. So far, GANs have been mostly applied to synthetic image generation for data augmentation and multi-modality translation (i.e., MR to CT [79]). cGANs have achieved good results and cycleGANs usually outperforms in terms of accuracy, in addition to overcome the issues related to paired image training [2].

EAML (Expert Augmented Machine Learning), is being widely used in medical field. Using domain-specific knowledge in state-of-the-art models, to increase not just the interpretability but lowering the bias and mistakes created by the models. This knowledge comes from a panel of experts from a specific area [2]. Some studies compared the efficiency from a model without the human knowledge vs with the human knowledge, and conclude that the use of knowledge in the data for training the model, creates a more balanced training dataset, where the algorithm query the experts for more important points/data for training again, this creates an almost "human-in-the-loop" concept.

Like said in the previous sections, data is the most important feature for the success of the model, and despite the progress in AI methods, the collection of data remains, in the medical field, poorly automatized and the curation and processing takes a lot of time. When comparing the train of the models with the processing of large databases, we can get a ratio of almost weeks vs months. The FAIR (Findability, Accessibility, Interoperability and Reusability) Data Principles , are a set of guiding principles with the main goal of providing a framework for ensuring that the data is effectively managed and shared in a way that promotes openness, transparency and collaboration. The medical community should focus efforts on trying to adapt these principles to some specificities from the medical domain [2].

## 5.3 PIX2PIX Architecture

Image-to-image translation, is defined as the task of converting an input image from one domain to another domain, while preserving the structure of the image. This involves learning a mapping function between two different visual domains, allowing for the transformation of images from one style, representation or modality to another. In our case we have one domain, CT images with B30f reconstruction kernel and we want to transform these images to another domain, B50f reconstruction kernel and vice-versa. The use of Gans, specifically cGans, cycleGans and PIX2PIX, are the most DL methods used for this type of work.

PIX2PIX uses a U-Net as a generator, the use of this architecture has some advantages regarding the capture of important features in medical images. The use of skip-connections maintaining the spatial information of the homologous layers from the down sampling (encoder) to the up sampling (decoder), creates a more highly accurate feature retrieval from the images. However, to train our decoder to transfer one domain to another, we have to teach it, and to teach it we have to compare the ground truth vs the generated image. Here the discriminator is used. The main goal is to compare, for example, the mapping from the encoder (B30f) to the decoder (B50f) and using loss functions to assess how good are compared to the real images. The discriminator is trained using the real images.

Then, the discriminator will be trained to discriminate between a real pair $(x, y)$ of a photo and the corresponding source image($x$) and the generated target $f(x)$. The discriminator task is to classify, and using the classification loss to train the discriminator, in this case a BCE loss (binary cross entropy). The generator will also have a loss function, regarding the generated images that the discriminator wrongly assumes as real. In the case of PIX2PIX, the generator has two loss functions. One of the loss functions being the "vanilla" loss function talked before (BCE), where the generator knows how much it has fooled the discriminator, and a L1 (MAE) loss function, for comparing the generated target images with the ground true target images, with a ratio 1:100 [45].

The discriminator in PIX2PIX is not a simple CNN, it uses a "PatchGan" classifier. PatchGan classifier, focuses more on classifying small images patches, it operates by dividing the input image and the generated image into non-overlapping patches, and each patch is fed to this discriminator where independently classifies whether the corresponding patch comes from the real image or the generated one. By using this type of discriminator, the PIX2PIX can capture local details and enforce more fine-grained image-to-image translation, instead of evaluating the entire image, thus creating high-quality and more realistic images [45].

The loss function for the discriminator is calculated two times. One for real images (also known as auxiliary loss function) and another for the fake images (known as adversarial loss function). The comparison between the true images is used for enforce semantic meaning between the source image and the target image, getting more consistency and realistic results. The comparison between the target image vs the fake image is used to train the Gan itself, by simulating a competition between the generator and the discriminator [45].

Relatively to the generator architecture, the encoder consists of C64-C128-C256-C512-C512-C512-C512-C51 (with “C” standing for convolutional) and for the decoder CD512-CD512-CD512-C512-C256-C128-C64 (with “CD” standing for convolution and dropout). After the last layer of the decoder, a convolution is applied to map the number of output channels, followed by a Tang function. Batchnormalization is not applied to the first C64 layer in the encoder. All the activation function in the encoder are leaky ReLus with a slope of 0.2. The ReLu in the decoder are not leaky [45].

Regarding the discriminator architecture, is used a 70x70 PatchGan classifier with an architecture of C64-C128-C256-C512. After the last layer, a convolution is applied to map to a 1-dimensional output, followed by a sigmoid function. Like the generator, Batchnormalization is not applied to the first C64 layer and all the ReLus are leaky with a slope of 0.2 [45].

As a goal to optimize the model, it has been used the Adam solver and minibatch SGD, with a learning rate beginning with the original paper value (0.0002) and following an optimizing strategy, as for the momentum parameters, we stick to the original values used in the original PIX2PIX paper[45], $beta_1 = 0.5$ and $beta_2 = 0.999$.

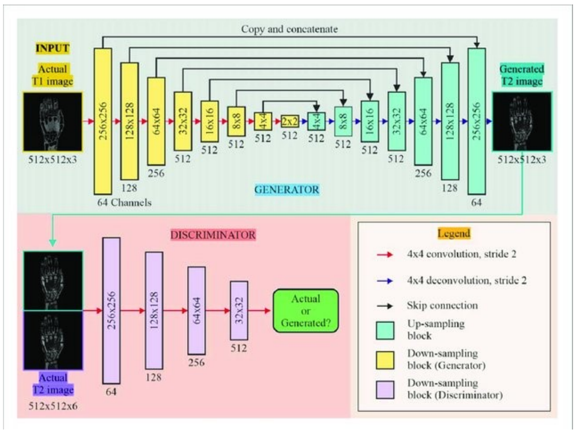


Figure 5.2: Architecture of PIX2PIX.Adopted from James A Grant-Jacob et al. *Exploring sequence transformation in magnetic resonance imaging via deep learning using data from a single asymptomatic patient*

## 5.4 Evaluation of generated images

For evaluating generated images, we have qualitative and quantitative methods. The qualitative measures are typically not numerical and often involve human subjective evaluation. Rapid scene categorization, rating and preference judgment and investigating and visualizing the internals of networks, are 3 qualitative methods for evaluating generated images [80]. In the "Rapid Scene Categorization" method, the human judges visualize the generated and the real image for a very limited amount of time, like a fraction of a second, as they have to classify if the image is real or fake. They tend to do an averaging of the scores from multiple human judges to reduce the variance. One limitation of using qualitative measures in this type of research is the requirement for human judges with expertise in a specific medical field. Additionally, evaluating generated images based on human performance introduces challenges related to inter-subjectivity and intra-subjectivity, as highlighted in previous studies [80], in this project, was asked a experienced pneumologist to evaluate pairs of fake and real images.

For quantitative measures, the evaluation refers to the calculation of specific numerical scores, being more objective. There are more twenty-four quantitative techniques for evaluating generated images. In this work, SSIM and FID had been chosen to evaluate the generate images.

FID (Fréchet Inception Distance), consist in two important components. The feature extraction, the Fid model uses a pre-trained Inception model to extract features from both the real and generated images and lastly compares the distribution from the extracted features from the real and the generated images, creating a multivariate Gaussian distribution, computing the mean and covariance for both the generated and real images. Then it calculates the distance between these two distributions (also called Wassertein-2 distance). This means that the less the value of FID, the more realistic are the generated images. We can define in mathematical terms as:

$$FID(r,g) = ||\mu_r - \mu_g||_2^2 \ + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}} \tag{5.1}$$

Where $(\mu_r, \Sigma_r)$ and $(\mu_g, \Sigma_g)$ are the mean and covariance of the real data and model distributions [80].

In terms of discriminability, robustness and computing efficiency, FID performs admirably. Even though it only considers the first two order moments of distributions, it seems to be a good metric. FID has been demonstrated to be more noise-resistant than IS (Inception Score) and to be compatible with human assessments [80].

Structural Similarity Index Measure (also known as SSIM), compares corresponding pixels and their neighbors in the real and the generated image using three quantities, luminance (I), contrast (C) and structure (S). Luminance considers the luminance or brightness information of the images. It evaluates how well the overall brightness and contrast of the image match. Contrast, looks at the local variations in contrast between different regions of the image. It assesses how well the contrast patterns align in corresponding areas. The structure analyzes the similarity between the images, how well the edges, textures and details align in the corresponding regions. In mathematical terms we can define as [80]:

$$I(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{5.2}$$

$$C(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{5.3}$$

$$S(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{5.4}$$

Where the variables $\mu_x$, $\mu_y$, $\sigma_x$, and $\sigma_y$ denote mean and standard deviations of pixel intensity in a local image patch. C1, C2 and C3 are numerical values added to confer some stability. The quantities are combined to form the SSIM score [80]:

$$SSIM(x,y) = I(x,y)^\alpha C(x,y)^\beta S(x,y)^\gamma \tag{5.5}$$

The SSIM values range between 0 and 1, where a value closer to 1 indicates higher similarity.

## 5.5 Materials and Methods

In this research, a CT images dataset was obtained from FIBRALUNG cohort (FCT grant — PTDC/MEC-RES/0158/2020), which includes patients with fibrotic interstitial lung diseases under follow-up at the Department of Pulmonology at Centro Hospitalar Universitário São João, Porto, Portugal. The dataset consisted of 2500 training images from the B50f and B30f (25 patients), resulting in a total of 5000 images and for testing 500 images(5 patients). The images were captured using Siemens Healthineers' CT scanners, specifically the 32-channel SOMATOM go.Up and 16-channel SOMATOM go.Now models. The scans were performed with patients lying in a supine position, using a slice thickness of 3 mm and a voxel matrix size of 512 x 512 x 3. The acquisition parameters included a voltage of 120 kV and 120 mA to 300 mA. For each specific image slice, two reconstructions were made using different kernels to obtain the exact same image from both domains.

All images underwent preprocessing to ensure they had appropriate pixel values for the PIX2PIX architecture. A paired-image dataset was created, consisting of aligned images from two domains: CT scans reconstructed with B30f and B50f kernels. The training and testing pairs are from different patients, 25 patients for the training data (2500 images) and 5 patients for the testing data (500 images). The goal was for the generator function, denoted as $f()$, to learn to convert a source image (B30f) represented by $x$ into a target image (B50f) represented by

$f(x)$. To enhance the transformation quality, a learning rate optimization strategy was employed. This involved training the model using the original learning rate (lr = 0.0002) and an extreme value (lr = 0.01), followed by gradually decreasing the learning rate after each complete model training (0.001/0.0008/0.0006, and finally 0.0001). Quantitative evaluation methods, namely SSIM and FID, and quantitative rating by an expert, were used to assess the quality of the generated images.

Initially, training was performed with 1000 3-channel images and 1-channel images, but no noticeable differences were observed. Subsequently, training was conducted using 500 1-channel images with different learning rates, allowing the identification of the optimal learning rates based on FID and SSIM metrics. These optimal learning rates were then used for training with the full sample.The SSIM and FID metrics were reassessed for the final models to determine the best model in terms of image quality and training time, and finally a qualitative analysis by an experienced pneumologist was made and using confusion matrices, analyzing the accuracy, using Cohen's Kappa test and McNemar's test ( with a p-value $< 0.005$ for statistical significance), we assess the reliability of our generated images.

The study utilized Python programming language version 3.8.5 along with various libraries to perform different tasks. The PyDicom library was used to load the data, the NRRD library was used to save the images, TensorFlow was employed for building the PIX2PIX model, Matplotlib library was used for data visualization, and Numpy library facilitated efficient data structure and manipulation. These libraries provided the necessary functionalities for the study's tasks. The models were executed using Nvidia Tesla V100-32 GB with 256 CPUs, which provided the computational resources for the training and evaluation processes.

## 5.6 Image Normalization and Pre-processing

The images in this study, were provided in DICOM format. Therefore, it is essential to use a helper function to load all the DICOM images from a folder into the Python environment. The pixel values in these images are in raw form and need to be converted into Hounsfield Units (HU). HU units, are a measurement scale used in CT scans to represent the radiodensity or attenuation of tissues and materials within the body. The HU units scale assigns numerical values to different substances based on their density, with water as the reference point (0 HU). The scale ranges from -1000 HU for air to +1000 HU for dense bone, with positive HU values indicating denser tissues and negative HU values representing less dense or air-filled structures. To convert the pixel values to HU, another helper function is needed that uses the slope and intercept information provided in the DICOM header for each image. This transformation is linear, allowing us to rescale the voxel values to their corresponding HU values. After applying the conversion, we should obtain a HU histogram representing the distribution of HU values similar to this:

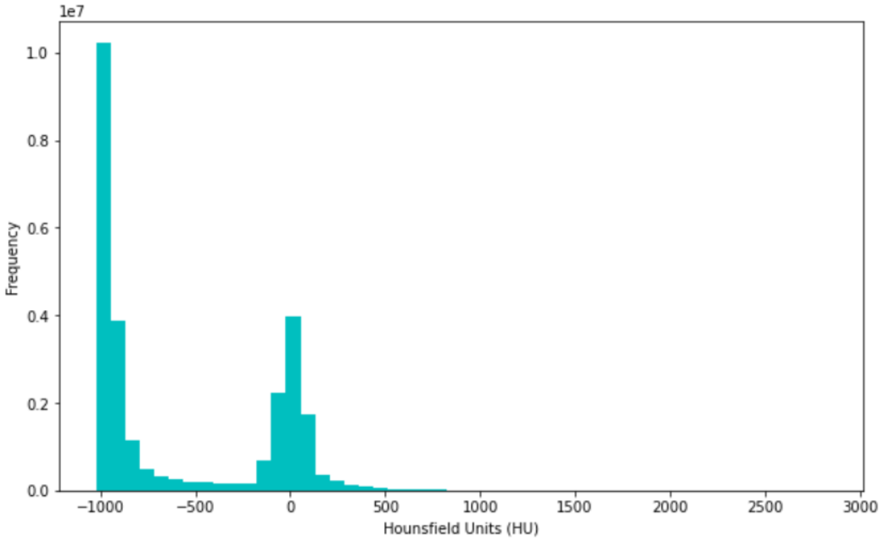


Figure 5.3: Normal distribution of Hu in a middle cut of a lung CT scan

The objective of the preprocessing is to minimize the loss of information from the CT scans. However, using a Tanh function in the final layer of the generator, which generates images in the range of [-1,1], poses a challenge for direct utilization of raw HU images in Pix2Pix due to convergence difficulties. To address this, a Min-Max normalization approach was employed. This normalization technique restricts the values between 0 and 1 and subsequently transforms them to fit the range of [-1,1] by multiplying by 2 and subtracting 1.

# Chapter 6

# Results and Discussion

The CT images used in this study have dimensions of 512 x 512 x 3, indicating that they are in 3-channel format, which is computationally more demanding compared to 1-channel images. The primary characteristics of these CT images lie in the intensity levels, distinguishing between brighter (B50) and darker (B30) regions. Theoretically, there is no significant difference between using 3-channel or 1-channel images in a PIX2PIX model, as both can preserve the essential image properties. Analyzing the plots (Figure 6.1/6.2) depicting the generator and discriminator losses during training, it is evident that there is no noticeable difference when using 3-channel or 1-channel images in the PIX2PIX model. However, employing 1-channel images offers the advantage of lower computational cost, faster convergence of the model and is less time consuming.

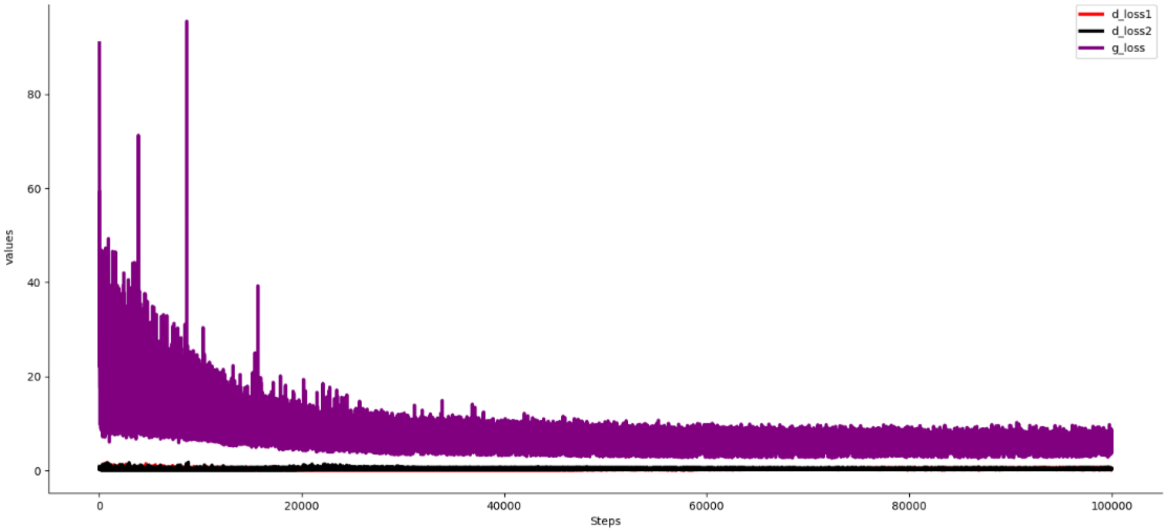


Figure 6.1: Loss plot of 1000 3-channel images in a PIX2PIX model with a LR of 0.0002. Legend: $d\_loss1$ - auxiliary loss function, $d\_loss2$ - adversarial loss function, $g\_loss$ - generator loss

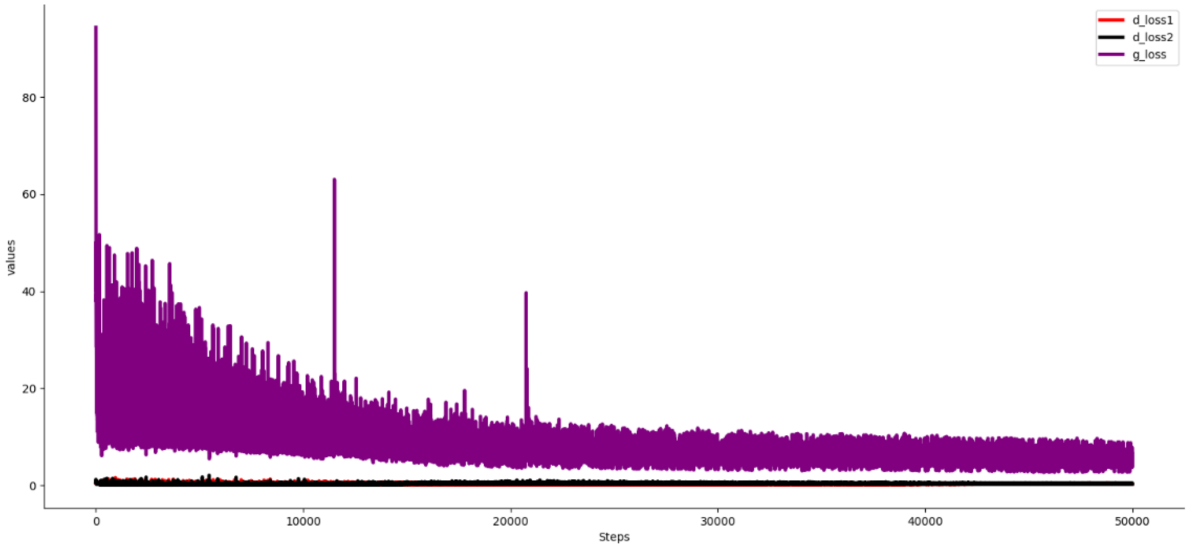


Figure 6.2: Loss plot of 500 1-channel scaled images in a PIX2PIX model with a LR of 0.0002. Legend: $d\_loss1$ - auxiliary loss function, $d\_loss2$ - adversarial loss function, $g\_loss$ - generator loss

Using higher learning rates, such as 0.01 (Figure 6.3), is generally not recommended for the PIX2PIX model. This is because larger learning rates can lead to unstable training dynamics, causing the model to overshoot and fail to converge to an optimal solution. It may result in fluctuating loss values, unstable gradients, and poor-quality generated images.

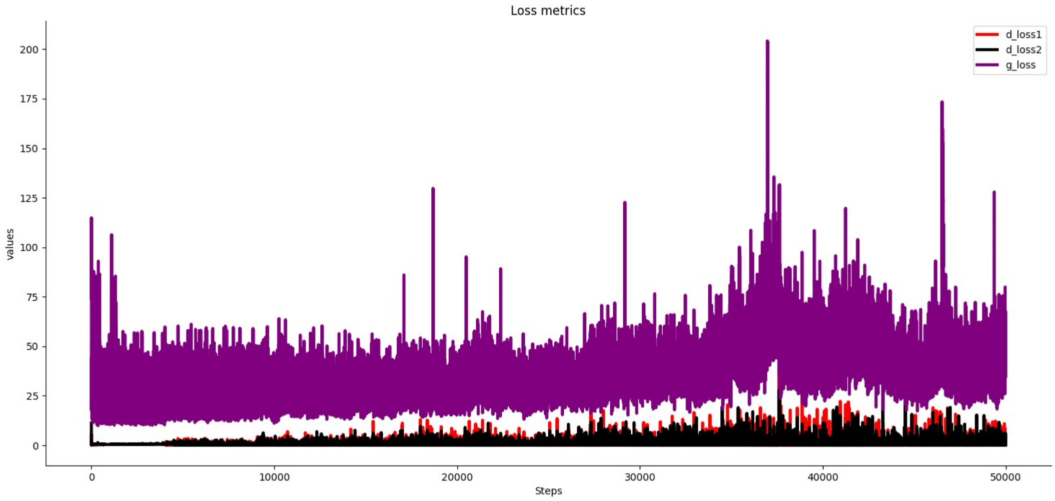


Figure 6.3: Loss plot of 500 1-channel images in a PIX2PIX model with a LR of 0.01. Legend: $d\_loss1$ - auxiliary loss function, $d\_loss2$ - adversarial loss function, $g\_loss$ - generator loss

The visual representation provided in the uppermost row of the Figure 6.4, displays the real B50f CT images, while the subsequent row presents the corresponding generated B50f images. These images were obtained using the PIX2PIX model with 1-channel images and a learning rate of 0.01. Analyzing the middle cut (first two images) and the abdominal cut (last image) of the CT scans, it becomes evident that the essential properties of the images, namely the preservation of the parenchyma and the delineation of the organs, are inadequately achieved.

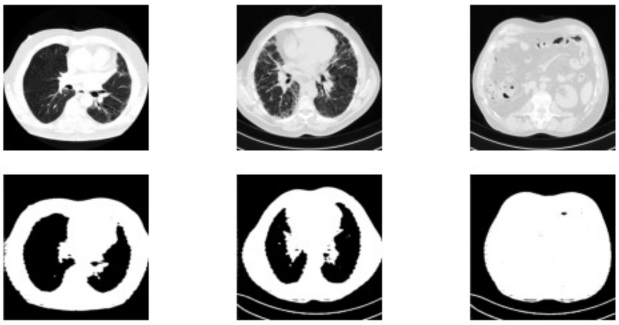
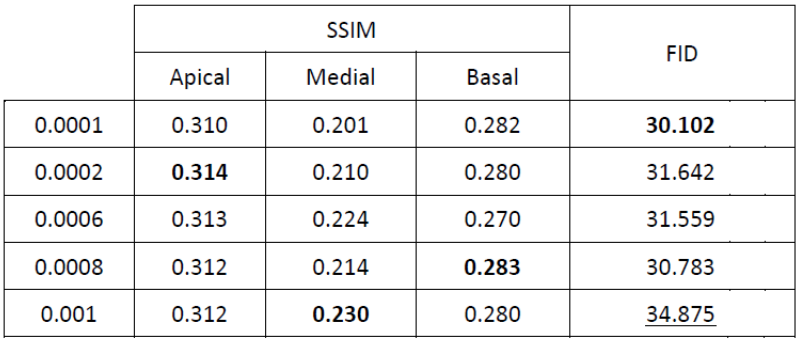


Figure 6.4: Uppermost row (Real B50f) compared with the last row (Generated B50f). The images were generated using a PIX2PIX model with a learning rate of 0.01

## 6.1 Conversion from B30f to B50f Kernel

Selecting an appropriate learning rate is crucial for achieving stable and high-quality results. Learning rates within the range of 0.001 to 0.0001 have been found to produce more consistent outcomes. Although the differences in performance among these learning rates may not be substantial on a global scale, is important to consider their impact on training time and image quality. Lower learning rates require more iterations to converge but often result in finer details and improved image quality. These models were trained in 500 images.

Table 6.1: Quantitative assessment with 500 testing images in the different learning rates for a PIX2PIX model (B30f -> B50f).Bold font highlights the best value among the listed options, while the underlined font denotes the worst value



| | SSIM | | | FID |
|---|---|---|---|---|
| | Apical | Medial | Basal | |
| 0.0001 | 0.310 | 0.201 | 0.282 | **30.102** |
| 0.0002 | **0.314** | 0.210 | 0.280 | 31.642 |
| 0.0006 | 0.313 | 0.224 | 0.270 | 31.559 |
| 0.0008 | 0.312 | 0.214 | **0.283** | 30.783 |
| 0.001 | 0.312 | **0.230** | 0.280 | 34.875 |

By employing 500 testing images for quantitative assessment, it becomes evident looking at the table (Tabel 6.1) that a learning rate of 0.001 yields the highest FID value, indicating poorer performance, while a learning rate of 0.0001 achieves the best FID value, suggesting superior performance. Moreover, by conducting a separate evaluation using three distinct test images (apical, medial, and basal), and applying the Structural Similarity Index (SSIM) metric across different learning rates, it was observed that the apical cut produced the most favorable outcome with a learning rate of 0.0002, the medial cut exhibited optimal results with a learning rate of 0.001, and the basal cut displayed improved performance with a learning rate of 0.0008. The models that were trained with learning rates of 0.0001, 0.0002, and 0.0008 demonstrated the best performance. As a result, we selected these learning rates for the final evaluation, which involved training the models with a dataset of 2500 images.

Table 6.2: Quantitative assessment with 500 testing images in the different learning rates for a PIX2PIX model (B30f -> B50f).Bold font highlights the best value among the listed options, while the underlined font denotes the worst value

| | SSIM | | | FID |
|---|---|---|---|---|
| | Apical | Medial | Basal | |
| 0.0001 | 0.32 | **0.40** | **0.31** | **25.12** |
| 0.0002 | 0.30 | 0.35 | 0.29 | 26.06 |
| 0.0008 | **0.35** | 0.33 | 0.28 | 35.81 |

In general, the model trained with a learning rate of 0.0001 achieved the best results in the FID evaluation and the SSIM evaluation for both the medial and basal cuts. On the other hand, the model trained with a learning rate of 0.0008 performed the worst, showing the highest FID value and the lowest SSIM scores for both cuts. The model trained with a learning rate of 0.0002 exhibited intermediate performance between these two models.

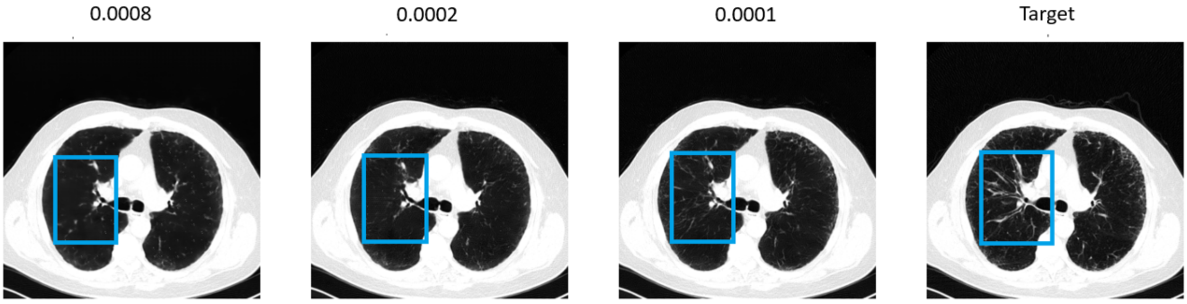


Figure 6.5: The comparison between the same cut and the trained models with different learning rates 0.0008,0.0002,0.0001 and the target image.Blue rectangle shows the region of interest, in this case the parenchymal tissue.

It can be observed (Figure 6.5/6.6) that the model trained with a learning rate of 0.0001 exhibits more intricate details, particularly in the parenchymal tissue (Blue rectangle), where it attempts to closely resemble the target parenchyma. When comparing the models trained with learning rates of 0.0002 and 0.0008, there is a slight improvement in the image quality and definition from the 0.0008 model to the 0.0002 model. The Figure 6.6, shows more detailed differences in a particular region of the parenchyma. The Figure 6.7, shows the different generated images with different models, showing an overall improvement in the image quality

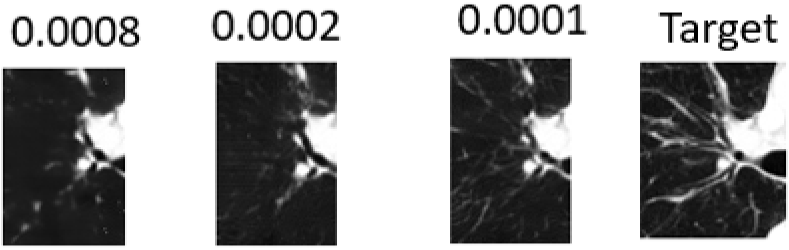


Figure 6.6: The comparison between the region of interest with different learning rates: First image represents the 0.0008, the second image represents the 0.0002, the third image represents the 0.0001 and the last image represents the target figure

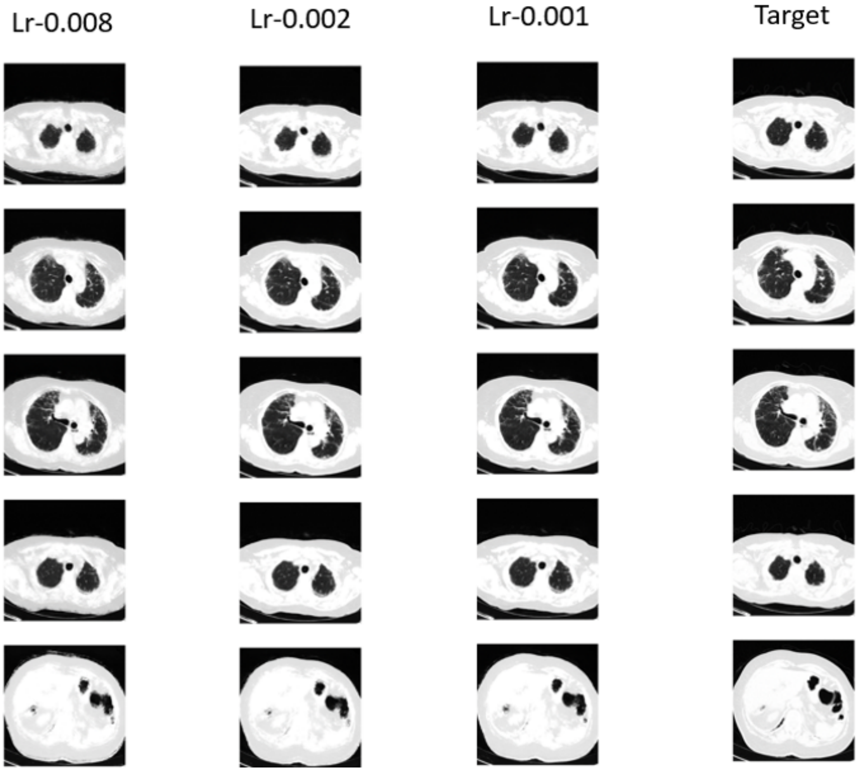
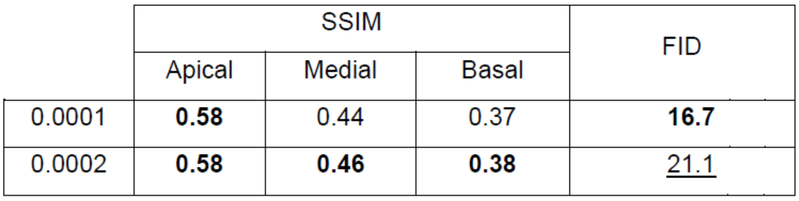


Figure 6.7: The comparison between the same cut and the trained models with different learning rates for B50f generated images: First column represents the 0.0008 generated CT, the second column represents de 0.0002, the third column represents the 0.0001 and the last column represents the target figure

## 6.2 Conversion from B50f to B30f Kernel

For transforming the B50f to B30f, we have discarded 0.0008 and used only 0.0002 against 0.0001 and train with 2500 images.

Table 6.3: Quantitative assessment with 500 testing images in the different learning rates for a PIX2PIX model (B50f -> B30f). Bold font highlights the best value among the listed options, while the underlined font denotes the worst value



| | SSIM | | | FID |
|---|---|---|---|---|
| | Apical | Medial | Basal | |
| 0.0001 | **0.58** | 0.44 | 0.37 | **16.7** |
| 0.0002 | **0.58** | **0.46** | **0.38** | 21.1 |

Overall, the model trained with a learning rate of 0.0001 demonstrates better consistency between the generated image and the target image. Although the model trained with a learning rate of 0.0002 shows slight quantitative improvements in the medial and basal cuts, the reliability of the FID evaluation is higher as it considers multiple images, and it aligns more closely with the visual examination of the images.

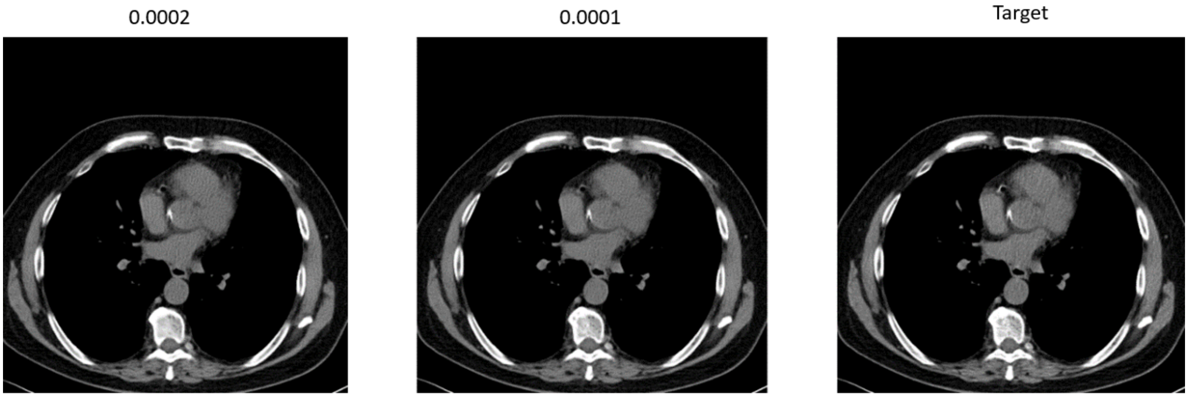


Figure 6.8: The comparison between the same cut and the trained models with different learning rates (0.0002/0.0001) and the target image

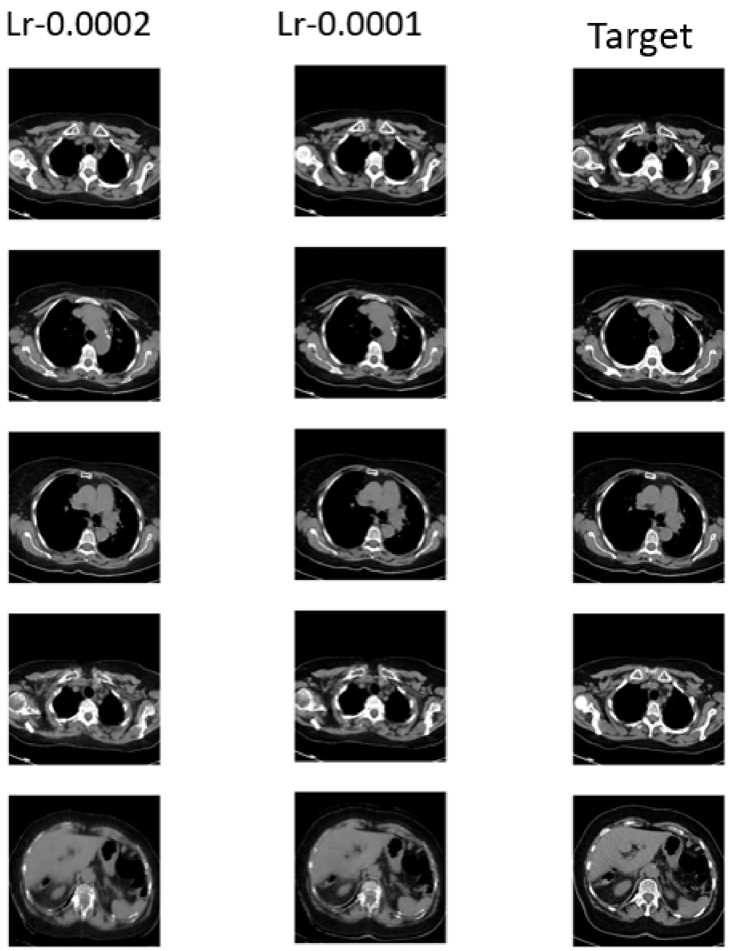


Figure 6.9: The comparison between the same cut and the trained models with different learning rates for B30f generated images: First column represents the 0.0002 generated CT, the second column represents the 0.0001, the third column represents the target figure

Considering that the B30f reconstruction kernel has lower resolution and fewer overall details compared to the B50f kernel, it appears to be more visually acceptable based on the images depicted in the Figures 6.8/6.9. The primary objective of this transformation is to maintain the dimension and relationships of the organs within the cut. The following table show that is not just more visual acceptable, but is proven by the direct comparison between the quantitative evaluation.

Table 6.4: Quantitative assessment with the model trained with a learning rate of 0.0001, and comparing the B50f and the B30f. Bold font highlights the best value among the listed options, while the underlined font denotes the worst value

| | SSIM | | | FID |
|---|---|---|---|---|
| | Apical | Medial | Basal | |
| B50f | 0.32 | 0.40 | 0.31 | 25.12 |
| B30f | **0.58** | **0.44** | **0.37** | **16.7** |

By utilizing the model trained with a learning rate of 0.0001 for both B50f to B30f and vice versa, and directly comparing their performance, it becomes evident that the model struggles more to accurately generate high-resolution images (B50f) in contrast to low-resolution images (B30f).

## 6.3 Qualitative Analysis

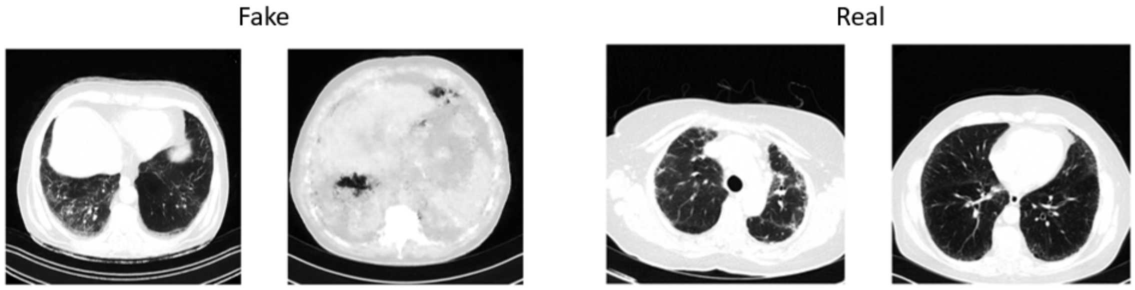
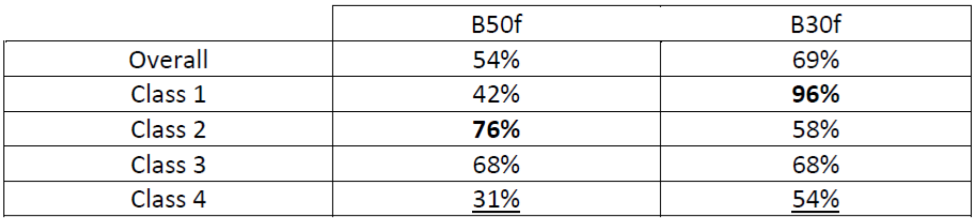


Figure 6.10: Sample of the test used for assessing the quality of the generated B50f images. Left: Fake pairs(class 2). Right: Real pairs(class 1)

In order to evaluate the generated images for medical professionals, a qualitative analysis was conducted with an experienced pneumologist. The analysis involved categorizing pairs of images into four classes: Class 1 represented real images, Class 2 represented fake images, Class 3 and Class 4 represent respectively fake image followed by real image and real image followed by fake image (Figure 6.10). A total of 100 pairs of B50f and B30f CT generated images and real images were used for the analysis.

Table 6.5: The accuracy metrics for the true labels and the predicted labels.Bold font highlights the best value among the listed options, while the underlined font denotes the worst value



| | B50f | B30f |
|---|---|---|
| Overall | 54% | 69% |
| Class 1 | 42% | **96%** |
| Class 2 | **76%** | 58% |
| Class 3 | 68% | 68% |
| Class 4 | 31% | 54% |

The table 6.5, shows that the best overall accuracy is within the B30f images. The worst predicted class in the B50f and B30f images is the class 4. It is noteworthy that the accuracy of class 1 in B50f images is 42%, while class 2 demonstrates an accuracy of 76%. Upon examining the provided Figure 6.11, it becomes apparent that for class 4 images, the incorrect predictions were often categorized as class 2 (fake pair). This suggests that our rater exhibits a bias towards selecting fake images, thereby creating an imbalance and potentially inflating the accuracy of class 2. The rater's ability to distinguish between real and fake images was influenced by a potential change in the image format used for the test. This change created some confusion for the rater, making it difficult to determine if the images were genuine or if they contained artifacts. The accuracy in class 1 and class 4 from the B50f images, is a good indicator that there is some proximity to the real images, since the accuracy is less than 50%.

The overall accuracy of B30f images can be attributed to the artifacts present in the generated images. Due to the lower definition and limited resolution of B30f, the quantitative evaluation values may be inflated as a significant portion of the image lacks clarity, particularly in the medial cuts. Consequently, the blurry and noisy nature of the generated images disturbs the ability to discern the definition of certain regions( Figure 6.11/6.12). A more detailed analysis can be observed in Figure 6.13, which illustrates two confusion matrices: one for the B50f images and another for the B30f images.

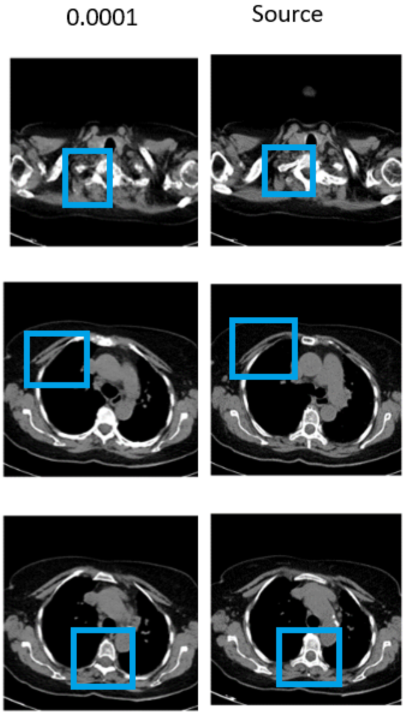


Figure 6.11: Comparison between B30f medial and apical images (Apical and Medial cuts). Right: B30f Real images. Left : Generated B30f images. Blue rectangle, used to compare the region of interest and their differences

Figure 6.11 illustrates challenges in accurately reproducing the bone structure, particularly in the first and third pairs. The effectiveness of the apical and medial cuts is relatively better compared to the basal cuts overall. In Figure 6.12, there is an observable increase in the lack of clear boundaries for the organs. The texture and appearance of the organs appear to be noisy in the first and third pairs.

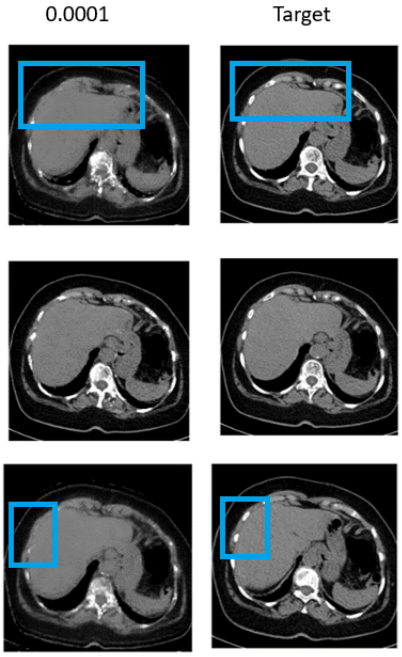


Figure 6.12: Comparison between B30f medial and apical images (Basal cuts). Right: B30f Real images. Left : generated B30f images. Blue rectangle, used to compare the region of interest and their differences

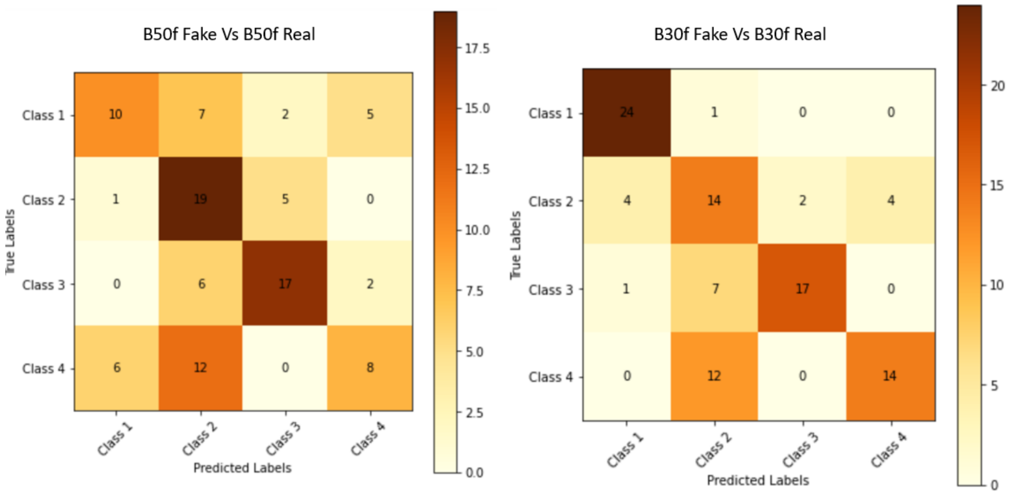
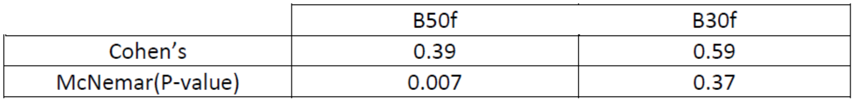


Figure 6.13: the figure on the left illustrates the confusion matrix, which shows the relationship between the true labels and the predicted labels assigned by the pneumologist for the B50f images. Similarly, the figure on the right represents the confusion matrix for the B30f images, depicting the correspondence between the true labels and the predicted labels assigned by the pneumologist

Cohen's kappa coefficient is a statistical measure employed to evaluate the agreement between two classifiers, specifically the true and predicted labels. It takes into account both the observed agreement and the agreement that would be expected by chance. The coefficient ranges from -1 to 1, where values closer to 1 indicate a higher level of agreement beyond what would be expected by chance [81]. McNemar's test, is used to compare the proportions of discordant pairs in a paired data set. It is commonly employed when examining the performance of two classifiers or treatments on the same subjects. A p-value below 0.005 is considered statistically significant, indicating a substantial difference between the ratings of the true labels and the predicted labels [82].

Table 6.6: The Cohen's and McNemar p-value test , for the true labels and predicted labels of the B50f and B30f.



| | B50f | B30f |
|---|---|---|
| Cohen's | 0.39 | 0.59 |
| McNemar(P-value) | 0.007 | 0.37 |

Both the B50f and B30f have a McNemar p-value above 0.005 (Table 6.6), indicating no major statistical differences between the ratings, but we have to consider the fact that using only 100 pairs, could limit at some extent this conclusion. The Cohen's value indicates a fair agreement for the B50f images and a moderate agreement for the B30f images.

Based on the available information, it can be inferred that the specialist's opinion regarding the B50f images demonstrates moderate agreement with the true labels, although there are some discrepancies among different classes.More important with class 1, with less 50% of the real images being rated as real. The results of the McNemar's test indicate that there is no statistically significant difference between the real and predicted labels. It is important to acknowledge the potential presence of bias in the specialist's ratings.

In the case of B30f images, the specialist's opinion exhibits a high level of agreement with the true labels, particularly for class 1 and class 3, where higher accuracy's are observed. The results of the McNemar's test indicate that there are no significant differences between the specialist's predictions and the true labels. This can be attributed to the presence of artifacts in the B30f images, which are caused by the lack of resolution in the input images.

# Chapter 7

# Conclusions

This thesis highlights the significant role of Machine Learning (ML) and Deep Learning (DL) techniques in the medical field, with a specific focus on the importance of CT images like B50f and B30f in lung diagnosis. The study emphasizes the potential of DL algorithms, particularly Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GANs), in the medical imaging field. It demonstrates how these algorithms can be utilized to enhance the accuracy and efficiency of lung diagnosis, contributing to more precise medical decisions and improved patient outcomes. Furthermore, the thesis addresses the domain transfer problem associated with CT images, specifically the challenges in transferring information from B30f to B50f and vice versa. It uses a model called PIX2PIX, to overcome this problem, highlighting the importance of developing robust models capable of adapting to different imaging conditions and maintaining diagnostic accuracy. The evaluation methods discussed in the thesis play a crucial role in assessing the performance and reliability of the developed models. Both qualitative and quantitative evaluation techniques are explored, enabling a comprehensive analysis of the algorithms' effectiveness.

In summary, the use of PIX2PIX demonstrates its effectiveness in performing domain transfer between a B50f kernel and a B30f kernel. However, there are a few limitations that need to be addressed. Firstly, the availability of an adequate amount of training data and the computational resources required for such tasks are crucial. Similar to other deep learning methods, increasing the quantity of data leads to improved results, for that is important to use powerful GPU and CPU machines. Our findings indicate that employing a learning rate of 0.0001 yields satisfactory performance in the domain transfer of CT. It is important to visually examine the issues associated with B30f and determine whether the same fine-tuning techniques utilized for B50f are applicable. To the best of our knowledge, the application of the PIX2PIX architecture for this specific task has not been previously explored.

This advancement represents a major breakthrough in the vast world of medical imaging. We hold a strong conviction that it marks a critical milestone in the development of innovative methodologies. These approaches transcend the boundaries of domain transfer problem and can be used for the augmentation of CT scan quality, enabling more accurate and objective disease prognostic, as well as the generation of synthetic medical images.

# Chapter 8

# Blibliography

[1] H. Arabi, A. AkhavanAllaf, A. Sanaat, I. Shiri, and H. Zaidi, “The promise of artificial intelligence and deep learning in PET and SPECT imaging,” Physica Medica, vol. 83, pp. 122–137, Mar. 2021, doi: 10.1016/j.ejmp.2021.03.008.

[2] A. Barragán-Montero et al., “Artificial intelligence and machine learning for medical imaging: A technology review,” Physica Medica, vol. 83. Associazione Italiana di Fisica Medica, pp. 242–256, Mar. 01, 2021. doi: 10.1016/j.ejmp.2021.04.016.

[3] Pamela McCorduck, Machines Who Think , Second edition.

[4] Daniel Crevier, AI: the tumultous Search for Artificial Intelligence. 1993.

[5] V. Kaul, S. Enslin, and S. A. Gross, “History of artificial intelligence in medicine,” Gastrointestinal Endoscopy, vol. 92, no. 4. Mosby Inc., pp. 807–812, Oct. 01, 2020. doi: 10.1016/j.gie.2020.06.040.

[6] Zhang, “On Definition of Deep learning,” 2018.

[7] Berbar Marr, “Big data: the 5 Vs Everyone Must know,” 2014.

[8] S. A. Diwani and Z. O. Yonah, “Holistic diagnosis tool for early detection of breast cancer,” International Journal of Computing and Digital Systems, vol. 10, no. 1, pp. 417–432, 2021, doi: 10.12785/IJCDS/100141.

[9] H. Arabi and H. Zaidi, “Applications of artificial intelligence and deep learning in molecular imaging and radiotherapy,” European Journal of Hybrid Imaging, vol. 4, no. 1. Springer Science and Business Media B.V., Dec. 01, 2020. doi: 10.1186/s41824-020-00086-8.

[10] Gong, “Machine learning in PET: from photon detection to quantitative image reconstruction,” 2020.

[11] L. Gallardo-Estrella et al., “Normalizing computed tomography data reconstructed with different filter kernels: effect on emphysema quantification,” Eur Radiol, vol. 26, no. 2, pp.

478–486, Feb. 2016, doi: 10.1007/s00330-015-3824-y.

[12] N. Sharma, R. Sharma, and N. Jindal, “Machine Learning and Deep Learning Applications-A Vision,” Global Transitions Proceedings, vol. 2, no. 1, pp. 24–28, Jun. 2021, doi: 10.1016/j.gltp.2021.01.004.

[13] Bishop, Pattern recognition and machine learning (Information science and statistics). 2006.

[14] Shrestha, “Augmenting organizational decision-making with deep learning algorithms: Principles, promises, and challenges,” Journal of Business, pp. 588–603.

[15] S.B KOTSIANTIS, “Supervised machine learning: a review of classification techniques”.

[16] R.S. Sutton, “Introduction: the challenge of reinforcement learning in: Machine learning”.

[17] Lotter W, “Robust breast cancer detection in mammography and digital breast tomosynthesis using an annotation-efficient deep learning approach”.

[18] Shen L., “Deep Learning to Improve Breast Cancer Detection on Screening Mammography”.

[19] Hesamian, “Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges”.

[20] Ibragimov, “Segmentation of organs-at-risks in head and neck CT images using convolutional neural networks”.

[21] James, “An Introduction to Statistical Learning: with Applications in R”.

[22] Chen C, “Unsupervised Bidirectional Cross-Modality Adaptation via Deeply Synergistic Image and Feature Alignment for Medical Image Segmentation”.

[23] Dou Q, “AdaNet: Plug-and-Play Adversarial Domain Adaptation Network at Unpaired Cross-Modality Cardiac Segmentation”.

[24] Liu Y, “MRI-based treatment planning for proton radiotherapy: dosimetric validation of a deep learning-based liver synthetic CT generation method”.

[25] Lei Y, “MRI-only based synthetic CT generation using dense cycle consistent generative adversarial networks”.

[26] Aganj, “Unsupervised Medical Image Segmentation Based on the Local Center of Mass”.

[27] Jin, “Generating Lumbar Spine MR Images from CT Scan Data Based on Semi-Supervised Learning”.

[28] Wang Z, “Semi-supervised mp-MRI data synthesis with StitchLayer and auxiliary distance maximization”.

[29] Burton W, “Semi-supervised learning for automatic segmentation of the knee from MRI with convolutional neural networks”.

[30] Cheplygina V, “ Not-so-supervised: A survey of semi-supervised, multiinstance, and transfer learning in medical image analysis”.

[31] Chen T, “A Simple Framework for Contrastive Learning of Visual Representations”.

[32] Shen, “Intelligent inverse treatment planning via deep reinforcement learning, a proof-of-principle study in high dose-rate brachytherapy for cervical cancer”.

[33] Christian Janieshch, “Machine learning and deep learning”.

[34] I.Goodfellow, “Deep learning,” in The MIT Press, 2016.

[35] “Deep learning for sentiment analysis:A survey,” Zhang Lei.

[36] Zhang Y, “Application of machine learning methods in the field of bioinformatics. Briefings in Bioinformatics”.

[37] M LI, “Predicting the epidemic trend of COVID-19 in China and across the world using the machine learning ap proach, medRxi”.

[38] P. Kumar, “Forecasting the dynamics of COVID-19 pandemic in top 15 countries in April 2020 through ARIMA model with machine learning approach”.

[39] Masci J, “Stacked convolutional auto-encoders for hierarchichal feature extraction”.

[40] Altaf F, “Going deep in medical image analysis: concepts, methods, challenges and future directions”.

[41] Zhao, “ Variational autoencoder for medical image analysis,” Journal of Healthcare Engineering, 2019.

[42] Ronneberger O, “U-net BT. Convolutional networks for biomedical image segmentation in: Internacional Conference on Medical Image Computing and computer-Assisted intervention”.

[43] Oktay O, “Attention unet: Learning where to look for the pancreas”.

[44] Diakogiannis FI, Resunet-a: a deep learning framework for semantic segmentation of remotely sensed data. 2020.

[45] Isola P, “ Image-to-image translation with conditional adversarial networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition,

[46] Zhu J-Y, “ Unpaired image-to-image translation using cycle-consistent adversarial networks,” in n Proceedings of the IEEE international conference on computer vision,

[47] Narin, “Automatic Detection of Coronavirus Disease (COVID19) Using X-ray Images and Deep Convolutional Neural Networks”.

[48] C. Rachna, “ Difference Between X-ray and CT Scan,” 2020.

[49] Waheed, “ CovidGAN: Data augmentation using auxiliary classifier GAN for improved Covid-19 detection,” in IEEE, 2020.

[50] Lambin P, “Radiomics: the bridge between medical imaging and personalized medicine”.

[51] Lambin P, “Radiomics: extracting more information from medical images using advanced feature analysis”.

[52] Morra L, “Artificial Intelligence in Medical Imaging”.

[53] Shmueli, Predictive analytics in information systems research. 2011.

[54] Heinrich, “Is Bigger Always Better? Lessons Learnt from the Evolution of Deep Learning Architectures for Image Classification,” 2019.

[55] Haselton, “The evolution of cognitive Bias,” 2015.

[56] Gama, “A survey on concept drift adaptation,” 2014.

[57] Widmer, “Learning in the presence of concept drift and hidden contexts,” 1996.

[58] Pentland, “The dynamics of drift in digitized processes ,” 2020.

[59] Heinrich, “Fool me once, shame on you, fool me twice, shame on me: A taxonomy of attack and defense patterns for AI security.,” in Proceedings of the 28th European Conference on Information Systems (ECIS), 2020.

[60] Eykholt, “Robust physical-world attacks on deep learning visual classification,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[61] Adadi, “ Peeking inside the black-box: A survey on explainable artificial intelligence (XAI),” in IEEE, 2018.

[62] Assaf, “Explainable deep neural networks for multivariate time series predictions,” in Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019.

[63] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. 2019.

[64] Amey Thakur, “Fundamentals of Neural Networks”.

[65] M. Nielsen, “Neural Networks and Deep Learning: Perceptron”.

[66] Hewamalage H, Recurrent neural networks for time series forecasting: current status and future directions. 2020.

[67] Laith Alzubaidi, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”.

[68] Xiao-Xia, “U-Net-Based Medical Image Segmentation”.

[69] I. J. Goodfellow et al., “Generative Adversarial Networks,” Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.2661

[70] Buckler, A collaborative enterprise for multi-stakeholder participation in the advancement of quantitative imaging. 2011.

[71] Berenguer, “Radiomics of ct features may be nonreproducible and redundant: Influence of ct acquisition parameters,” 2018.

[72] Kemerink GJ, CT lung densitometry: dependence of CT number histograms on sample volume and consequences for scan protocol comparability. 1997.

[73] Choe, “eep Learning–based Image Conversion of CT Reconstruction Kernels Improves Radiomics Reproducibility for Pulmonary Nodules or Masses”.

[74] Andrew D. Missert, “ Synthesizing images from multiple kernels using a deep convolutional neural network”.

[75] Kim J, “Accurate image super-resolution using very deep convolutional networks,” in omputer Vision and Pattern Recognition., 2016.

[76] Liang G, “GANai: standardizing CT images using generative adversarial network with alternative improvement,” 2018.

[77] hu J-Y, “Unpaired Image-to-Image Translation Using CycleConsistent Adversarial Networks,” in 2017 IEEE International Conference on Computer Vision, 2017.

[78] S. K. Shizuo Kaji, Overview of image to image translation by use of deep neural networks: denoising, super resolution, modality conversion, and reconstruction in medical imaging.

[79] Kazemifar S, “MRI-only brain radiotherapy: Assessing the dosimetric accuracy of synthetic CT images generated using a deep learning approach”.

[80] Ali Borji, “Pros and Cons of GAN Evaluation Measures”.

[81] B. Więckowska, K. B. Kubiak, P. Jóźwiak, W. Moryson, and B. Stawińska-Witoszyńska, “Cohen’s Kappa Coefficient as a Measure to Assess Classification Improvement following the Addition of a New Marker to a Regression Model,” Int J Environ Res Public Health, vol. 19, no. 16, Aug. 2022, doi: 10.3390/ijerph191610213.

[82] M. Q. R. Pembury Smith and G. D. Ruxton, “Effective use of the McNemar test,” Behav Ecol Sociobiol, vol. 74, no. 11, Nov. 2020, doi: 10.1007/s00265-020-02916-y.