Università degli Studi di Palermo

AREA QUALITÀ, PROGRAMMAZIONE E SUPPORTO STRATEGICO
SETTORE STRATEGIA PER LA RICERCA
U. O. DOTTORATI

Dottorato in Information and Communications Technologies
Ciclo XXXVII
Dipartimento di Ingegneria
Laboratorio di Interazione Uomo-macchina

Developing and integrating computationally efficient deep neural networks for medical image segmentation

Thesis written with financial support from the FSE-REACT-EU programme,
PON "Ricerca e Innovazione" 2014-2020 (D.M. 1061/2021) Azione IV.5
"Dottorati su tematiche Green/Innovation"

Ph.D. Candidate:
Luca Cruciata

Ph.D. Coordinator:
Ch.mo Prof. **Ilenia Tinnirello**

Supervisor:
Ch.mo Prof. **Roberto Pirrone**

Co-Supervisor:
Dr. Albert Comelli

CICLO XXXVII
ANNO CONSEGUIMENTO TITOLO 2025

UNIONE EUROPEA
Fondo Sociale Europeo
Ministero dell'Università e della Ricerca
PON RICERCA E INNOVAZIONE
REACT EU
Università degli Studi di Palermo

Contents

Abstract

In recent years, the increasingly widespread use of artificial intelligence systems has led to a significant increase in the demand for computational resources, raising questions regarding the energy sustainability and efficiency of the models. This research work fits into this context with the aim of developing and integrating deep neural network architectures that are computationally efficient and suitable for real-world applications in the medical domain, where the scarcity of labelled data and stringent privacy restrictions represent major constraints. Through the project, several directions of research were explored: in a first step, a training process optimisation strategy called Learn&Drop was proposed, which allows to dynamically identify and remove less relevant layers during training, reducing the number of parameters and improving computational efficiency. On this basis, a lightweight architecture was designed, the Few-Parameter Architecture (FPA), capable of maintaining competitive performance in classification and segmentation while operating with reduced computational resources. The experimental validation of these models was conducted on real biomedical data, characterised by high variability and the presence of artefacts, in order to meet realistic and challenging requirements aligned with the state of the art. A primary issue that emerged lies in the integration of AI into clinical flows, which led to the development of IODeep, a DICOM-compliant data structure that enables complaint entry into hospital PACS, ensuring patient privacy and compliance with field protocols. The domain considered presents an additional challenge, namely the scarcity of annotated data, which in the first instance was addressed using generative AI approach to generate cross-domain synthetic images useful to fill this gap. In addition, a meta-learning-based approach based on the few-shot segmentation paradigm was developed. In this way, even where synthetic generation is not sufficiently reliable, it is possible to provide clinical practice with tools capable of operating under conditions of low data availability. Overall, the results obtained highlight how it is possible to reconcile predictive accuracy, computational efficiency and environmental sustainability through the adoption of innovative architectural design techniques and adaptive learning strategies, thus contributing to the development of viable and safe artificial intelligence solutions in the health sector.

List of Figures

List of Tables

List of Acronysms

- AI - Artificial Intelligence
- DL - Deep Learning
- EHDS - European Health Data Space
- NLP - Natural Language Processing
- ML - Machine Learning
- LLM - Large Language Model
- DNN - Deep Neural Network
- DICOM - Digital Imaging and COmmunications in Medicine
- IOD - Information Object Definition
- PACS - Picture archiving and communication system
- MRI - Magnetic Resonance Imaging
- CT - Computed Tomography
- ROI - Region Of Interest
- GMM - Gaussian Mixture Model
- CNN - Convolutional Neural Network
- FCN - Fully Convolutional Network
- MLP - Multy Layer Perceptron
- NAS - Neural Architecture Search
- PET - Positron Emission Tomography
- RCNN - Region-based Convolutional Neural Networks
- ViT - Vision Trasformer
- FLOP - Floating Point Operations per second
- GAN - Generative Adversarial Network

- cGAN - Conditional GANs
- ART - Aggregated Residual Transformers
- DDPM - Denoising Diffusion Probabilistic Model
- LVB - lower variational bound
- FID - Fréchet Inception Distance
- FSS - Few-shot semantic segmentation
- MAML - Model-agnostic meta-learning
- FPA - Few-Parameter Architecture
- AAPD - Average Absolute Partial Derivative
- SGD - Stochastic Gradient Descent
- MMAC - Mega Multiply-Accumulate
- DSCNN - Depth Separable Convolutional Neural Architecture
- NIN - Network In Network
- GAP - Global Average Pooling
- DSC - Deep Separable Convolution
- ACR - American College of Radiology
- NEMA - National Electrical Manufacturers Association
- SOP - Service-Object Pair
- UID - Unique IDentifier
- HIS - Health Information Management Systems
- RIS - Radiology Information Systems
- XAI - eXplainable Artificial Intelligence
- WSI - Whole Slide Images
- RTSS - RT Structure Sets

- IOM - Information Object Module
- MAE - Mean Absolute Error
- MSE - Mean Squared Error
- RMSE - Root Mean Squared Error
- PSNR - Peak Signal-to-Noise Ratio
- SSIM - Structural Similarity Index

Chapter 1

Introduction

Artificial intelligence (AI) has become an established factor in everyday life. Apps that recommend transportation routes, voice assistants like Siri and Alexa, and algorithms that tailor social media material are all examples of imperceptible yet successful technologies. In healthcare, AI enables earlier and more accurate diagnosis, improve access to care. Even at work, technology automates tedious chores, leaving up more time for creativity. The benefits are significant: it may improve productivity, eliminate waste, and democratize access to technology. Furthermore, when utilized appropriately, AI has the potential to help tackle global problems such as climate change and expanding educational possibilities. The real issue is to develop it ethically and inclusively, so that the benefits are shared by all people. The demonstrated ability to process vast amounts of data and succeed in difficult tasks has made it a tool to support various operations in order to boost productivity and efficiency by reducing errors and biases caused by subjective decisions or evaluations of operators.

In my research project, I conducted a systematic analysis of state-of-the-art neural networks with the goal of gaining a thorough understanding of their architectures, training techniques, and the major challenges that remain unsolved, in order to assess the impact of the number of training parameters on semantic segmentation performance. This work addressed particular essential difficulties in model optimization methods, aiming at finding a balance between prediction accuracy and computing efficiency. The analysis emphasized the need of advanced tactics for improving model convergence and mitigating concerns like overfitting and sensitivity to beginning conditions. In fact, while AI has many benefits, it also has a substantial environmental impact, owing to the enormous energy consumption required by data centers for training and model inference. These energy-intensive processes are mostly powered by coal-

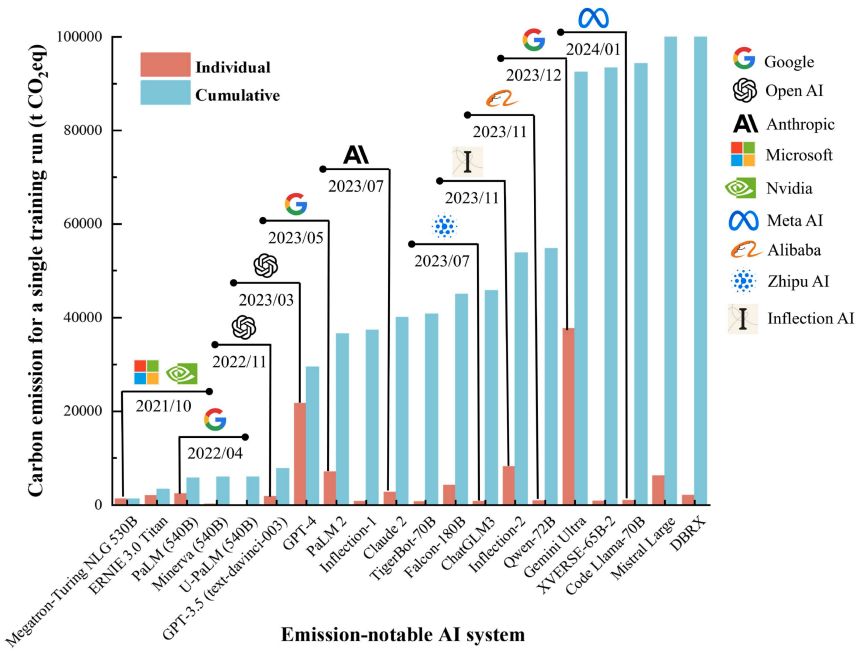


Figure 1.1: Top 20 AI systems in terms of carbon emission for a single training run [1].

and gas-fired power plants, whose intensive output contributes significantly to the rise in greenhouse gas emissions, with irreversible repercussions for climate change. Furthermore, resource consumption reveals itself in the increased demand for water resources and heat transfer systems within data centers.

In their 2024 paper [1], Yu et al. analyzed the carbon emissions of 79 prominent AI systems released between 2020 and 2024. They reported that the total emissions of the top 20 systems could reach up to 102.6 million tonnes of CO_2 equivalent per year (see Figure 1.1).

This might have a substantial impact on environmental economics, creating a market with costs that could surpass $10 billion per year, given expected to carbon penalties in the near future. As a result, the study emphasizes the significance of adopting standardized quantitative analysis procedures in order to produce adequate measurements for quantifying the carbon emissions connected with AI systems. This would encourage the sector to invest in green practices and technology within these restrictions, propelling the increasingly collaborative AI world toward a more sustainable future.

Fortunately, the research community is beginning to develop a growing awareness of these issues. De Vries [2] in his 2023 paper, analyses the impact of massive and globally used applications such as web browsers. As can be seen from the Figure 1.2, energy consumption will grow over the years to

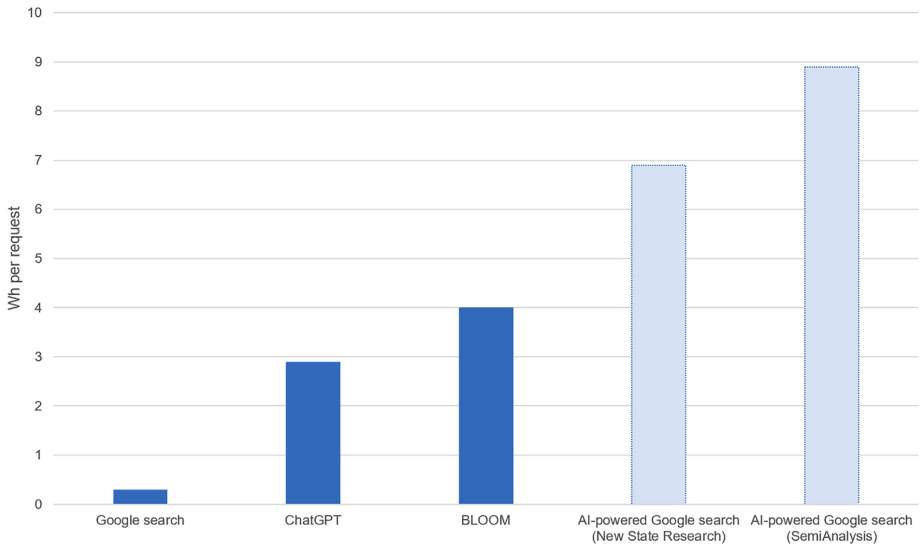


Figure 1.2: Estimated energy consumption per request for various AI-powered systems compared to a standard Google search [2].

approximately 9Wh per search request on browser-accessible search engines.

In fact, the introduction of AI in Google Chrome would result in an annual electricity consumption similar to that of the whole country of Ireland for each search. De Vries himself emphasizes the significance of developing methods to monitor and regulate these platforms, preventing their indiscriminate usage.

To deal with this issue, different ways for lowering computing costs or providing greener solutions have been offered. In 2024, Schwartz [13] introduces a paradigm for evaluating energy efficiency in NLP models, including computational metrics alongside traditional accuracy criteria. Canedo et al. [14] empirical investigation demonstrates that using optimized training approaches and renewable energy-powered cloud infrastructures can greatly lower the carbon footprint of machine learning models. Patterson et al. [15] present another example that outlines the primary criteria known as best practices for containing and reducing ML-related emissions. This work shows that using more efficient models, specialised hardware, optimised data centres (in the cloud) and sources with reduced emissions can halt, and even reverse, the trend of growing carbon footprints over the next few years. A recent example was Deepseek [16] which uses techniques inspired by the work of 'Efficiency Misnomer' [17], demonstrating that intelligent parameter selection, as well as the use of FlashAttention presented by Dao et al. [18] reduce the load on the memory during training, halving the GPU usage and thus reducing the overall cost by $30 - 50\%$. This avoids unnecessary computational processes

and optimizes resource use, mitigating the environmental impact of artificial intelligence and increasing its sustainability. However, most efforts in this direction have focused on general-purpose models or large-scale applications in fields such as natural language processing or computer vision. The biomedical field, despite its critical societal importance and the growing adoption of AI technologies, has received relatively less attention in this context. For this reason, and to define a clear and coherent research area for my PhD, I focused on analyzing the challenges of sustainability in the biomedical sector. AI systems are increasingly integrated into clinical workflows and diagnostic pipelines, where ensuring high performance while reducing excessive energy consumption is both a technical and ethical imperative.

One of the first significant critical issues was the limited accessibility of clinical data, mainly due to regulatory constraints on privacy and the need to keep such data within the hospitals that manage them. This issue impedes the sharing and distribution of datasets, limiting the possibilities of large-scale training of artificial intelligence models. In this sense, the European Commission's decision to promote the establishment of the European Health Data Space (EHDS)[1] was important. The EHDS represents an integrated and regulated ecosystem for health data management, consisting of shared norms, technical standards, digital infrastructures and a common governance framework. The main objective is to give citizens back control over their electronic health data, while ensuring secure and regulated access by healthcare professionals and researchers. In particular, the EHDS aims to create a single European market for electronic health record (EHR) systems (see Figure 1.3), advanced medical technologies and high-risk artificial intelligence solutions. In addition, it aims to provide the scientific community with a reliable regulatory environment that facilitates the secondary use of data for research purposes through a transparent and supervised process [19, 20].

The problem of annotated data scarcity is accentuated in complex tasks like segmentation. To address this issue, in addition to traditional data augmentation techniques, the industry community is beginning to use generative AI approaches to increase data availability and facilitate the training of classification and segmentation models through the use of synthetic data that can replicate real data's information and characteristics. In reality, unlike classification, which associates one or more labels with the entire image, segmentation requires pixel-wise annotation, i.e. the creation of high-precision ground truth

[1] urlhttps://www.european-health-data-space.com/

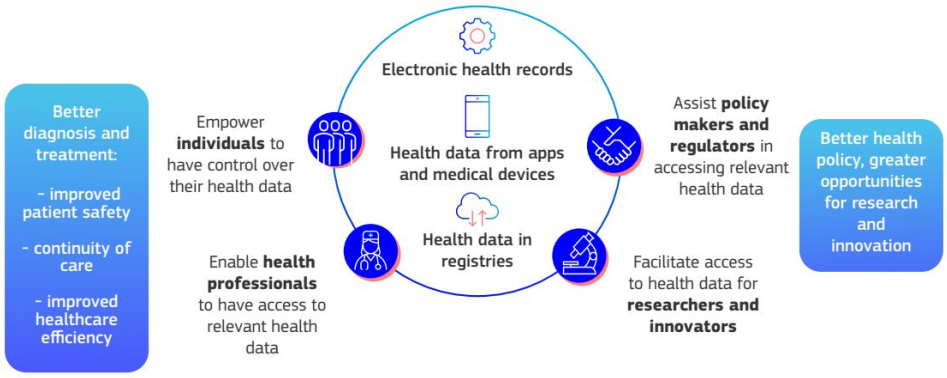


Figure 1.3: EHDS structure and proposed improvements to the healthcare system. https://www.european-health-data-space.com

masks. This process is significantly more costly in terms of time, human resources, and expertise, making it extremely challenging to create large and well-balanced datasets.

The 2.1 section discusses the task of segmentation, providing a general overview of the meaning of the term and the different application declinations in which it can manifest itself. The main challenges associated with each type of segmentation are also discussed, highlighting the practical and methodological implications of the availability (or lack thereof) of detailed annotations.

New methods have emerged to address these limitations, of which the few-shot is undoubtedly one of the most promising. In fact, it is particularly suitable in situations of data scarcity because it allows one to adapt quickly and with few samples to the task at hand.

In fact, during the research period abroad at Keele University, it was decided to further study these methods on the segmentation task. The choice fully adheres to the planned line of research and focuses on the construction of models capable of generalisation using the meta-learning paradigm of learning from experience.

In addition to overcoming the issue of data scarcity, these techniques have the benefit of considerably lowering computational costs during the training phase. Unlike few-shot language models, which generalise by using prompts other than those in the training set, segmentation scenarios require the adaptation of model weights or the use of specific architectures such as Matching Networks. The latter use a prototypical representation of classes: each class is described by a vector (prototype) created by averaging the embeddings of the examples in that class. During the learning phase, the network processes

a small set of annotated images (the support set) to build the relevant representations and prototypes. When a new image (query) is provided, the model estimates its embedding and compares it to the prototypes using a distance metric, assigning the class to the closest prototype. This non-parametric classification makes no assumptions about predefined classes and instead relies on the model's capacity to generalize dynamically based on the support set provided. The training follows a meta-learning protocol, consisting of a meta-training phase and a meta-testing phase. This framework makes it possible to assess the model's adaptability to new tasks, while guaranteeing accurate performance. It is evident how a few-shot learning solution can significantly broaden the applications of AI in the medical field, overcoming data scarcity limitations and promoting a more sustainable and adaptive artificial intelligence paradigm. A detailed discussion of the few-shot learning approach and its implementation in the presented study will be provided in Chapter 8.

1.1 Motivation and goals

Motivated by the intention to search for an approach aimed at optimising neural networks, one phase of my research activity was based on the definition of innovative methodologies aimed at increasing the robustness and computational efficiency of models. Indeed, one of the main emerging problems in the use of AI algorithms is related to the excessive consumption of computational resources. The advent of LLMs has favoured their expansion and uncontrolled use. In fact, a common trend emerges from the state of the art, based on the use of models with more and more parameters and a consequent increase in the computational resources required; this trend in previous years led to an improving performance that has reached a plateau in recent years. In this regard, the first research question addressed within this thesis is related to a changing approach to tasks in contrast to the general trend.

To answer this question, the first phase of the activity focused on an optimisation strategy based on parametric efficiency, aimed at improving the training process by selectively removing layers that had no longer contributed to learning. This approach proposes to preserve the final performance of the model without compromising its accuracy, while achieving a significant acceleration of the training phase compared to traditional training methodologies. These results are described in detail in Chapter 4 and were published in a first quartile journal and directed research towards the construction of a low-parameter

network capable of performing correctly. In fact, in Chapter 5, the problem of segmentation in the absence of annotations is explicitly addressed, proposing the development of a network capable of automatically generating segmentation masks from the information apprehended throughout a classification task.

The idea behind this approach is to exploit the model's ability to identify and locate discriminative patterns in images, even in the absence of explicit supervision in the form of masks. More specifically, the method described in this chapter is based on the development of an extremely lightweight architecture, designed to operate efficiently even in environments with limited computational resources. This Deep Neural Network (DNN) showed very competitive performance in supervised classification and segmentation tasks, confirming the validity of the proposed architectural design. However, the more complex agnostic segmentation tasks, i.e. in scenarios without mask annotations, did not perform as well as expected. The absence of masks compromises the accuracy of the segmentation, highlighting the need for further methodological refinements to make the model capable of generalising effectively even in completely unsupervised contexts. These two difficulties are fairly universal, and they were solved with general benchmark datasets. In the second phase of my work, the knowledge gained was applied to the medical area.

The study of images in the medical field highlighted the problem of lack of and difficult access to data due to privacy issues, hence the second research question: is it possible to integrate AI directly into the DICOM standard in order to avoid moving data from their storage locations?

From this perspective, the section 6.1 describes the process of storing and managing clinical data within hospital institutions, highlighting the implications of the DICOM standard. The analysis of this issue led to the development of an innovative methodology, described in Chapter 6, which allows the training of models directly into the clients of the hospital network, guaranteeing compliance with security regulations and the standard. This contribution resulted in a publication in a first quartile journal.

Considering that the development of an IOD that integrates AI within Server PACS is an inefficient measure to completely overcome the problem of the shortage of labelled data, the research activity addressed a new research question, based on identifying alternative methodologies to compensate for these deficits.

In order to overcome this critical issue, during my research period at the hosting company, I worked on developing a generative approach for the creation

of synthetic images, with particular reference to the conversion from the MRI to the CT domain. In Chapter 7, a comparative study between different models is presented, with a detailed analysis of their performance in terms of the quality of the synthetic images obtained.

Finally, consistent with the work done at the company, during the period abroad the problem of image shortage was addressed by changing the point of view. In fact, while the generation of synthetic images can increase the sample population, finding a new paradigm to solve the problem directly during training can be an alternative solution. The few-shot, as described above, structures learning by recreating real situations, with few samples for a new class provided the model must be able to extrapolate the information to create segmentation masks. This will be described in detail in the Chapter 8.

1.2 Dissertation outline

The remainder of the dissertation is organized as follows.

In **Chapter 2** and **Chapter 3**, the state-of-the-art in the tasks mentioned in the preceding sections is analyzed. A description of the datasets used in the experimental portions of this thesis is also provided.

In **Chapter 4**, provided a strategy that has been to optimizing neural network training. This approach employs metrics that analyze the gradient to remove layers that are no longer helpful to the training, resulting in fewer parameters and less time and resources consumption.

In **Chapter 5**, a lightweight convolutional DNN for classification and segmentation is described, based on the same principle of optimization and parameter reduction. As an additional task, classification performance has been utilized to generate ROI proposals.

IODeep is discussed in **Chapter 6** to solve the difficulties and complexities of dealing with clinical data, such as privacy limits and sensitive data handling. The latter is a DICOM-compliant module that enables neural models to be incorporated directly into hospital environments, removing the need to transfer data from its storage location.

In **Chapter 7** the subject of image generation is addressed, using generative networks for the creation of new synthetic samples, which retain the characteristics of the originals. In our use case, it allowed us to generate CTs, having only MRIs available.

To address the problem of training in the presence of few samples, the prob-

lem of few-shot segmentation is addressed in **Chapter 8**, where a proposed experimental approach is also illustrated.

In the Figure 1.4, the topics addressed and the flow followed in the course of this thesis work are presented graphically.

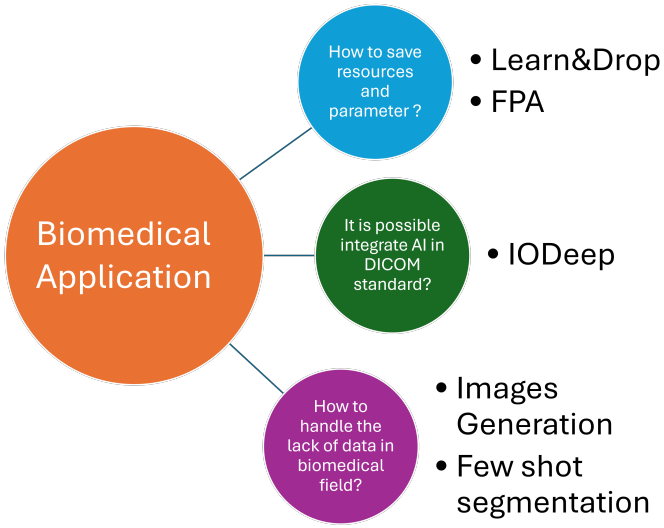


Figure 1.4: Summary of topics covered in the thesis

Chapter 2

State of the art

In this chapter, the state of the art of approaches in line with the research topics addressed in this thesis are described. First of all, an introduction will be made to segmentation, its applications and the solutions adopted to solve this task; subsequently, a description of approaches using networks with a few parameters is given, moving on to generative models and few-shot learning approaches.

2.1 Image Segmentation

The task of subdividing an image into meaningful parts is known as image segmentation, a computer vision approach that tries to separate an image into discrete objects or regions for simpler comprehension and analysis. Segmentation has applications in many fields, such as medicine, where it helps physicians diagnose illnesses by analyzing radiological images; autonomous driving, where it enables cars to differentiate between roads, pedestrians, and obstacles; and, more importantly, object recognition in challenging situations, which enhances environmental awareness for security and surveillance applications.

The problem is that given an image I of dimension $H \times W \times C$, the objective is to assign each pixel (i, j) a label s_{ij} to indicate its belonging to a specific class. Formally, the creation of a segmentation mask S of dimension $H \times W$ can be stated as:

$$S = \{s_{ij} | 1 \leq i \leq H; 1 \leq j \leq W\} \tag{2.1}$$

with $s_{ij} \in \{1, 2, ..., K\}$ and K is the number of classes.

2.1.1 Segmentation techniques

Image segmentation is distinguished into different types, each characterised by specific operational features and areas of application. A distinction is made between these types according to the objectives pursued and the degree of precision required, which can vary from a simple subdivision by regions to a detailed classification at the level of individual pixels.

Semantic segmentation is the most popular type, wherein each pixel is assigned a label that corresponds to a broader category, like "street," "tree," or "person." However, this mode does not distinguish between distinct objects belonging to the same class: for example, two neighbouring persons are considered as one area. Such methods and uses have been thoroughly examined by Csurka et al. [21] and investigated in a number of real-world settings by Mazhar et al. [22], referring to industries such remote sensing, healthcare, and automobiles.

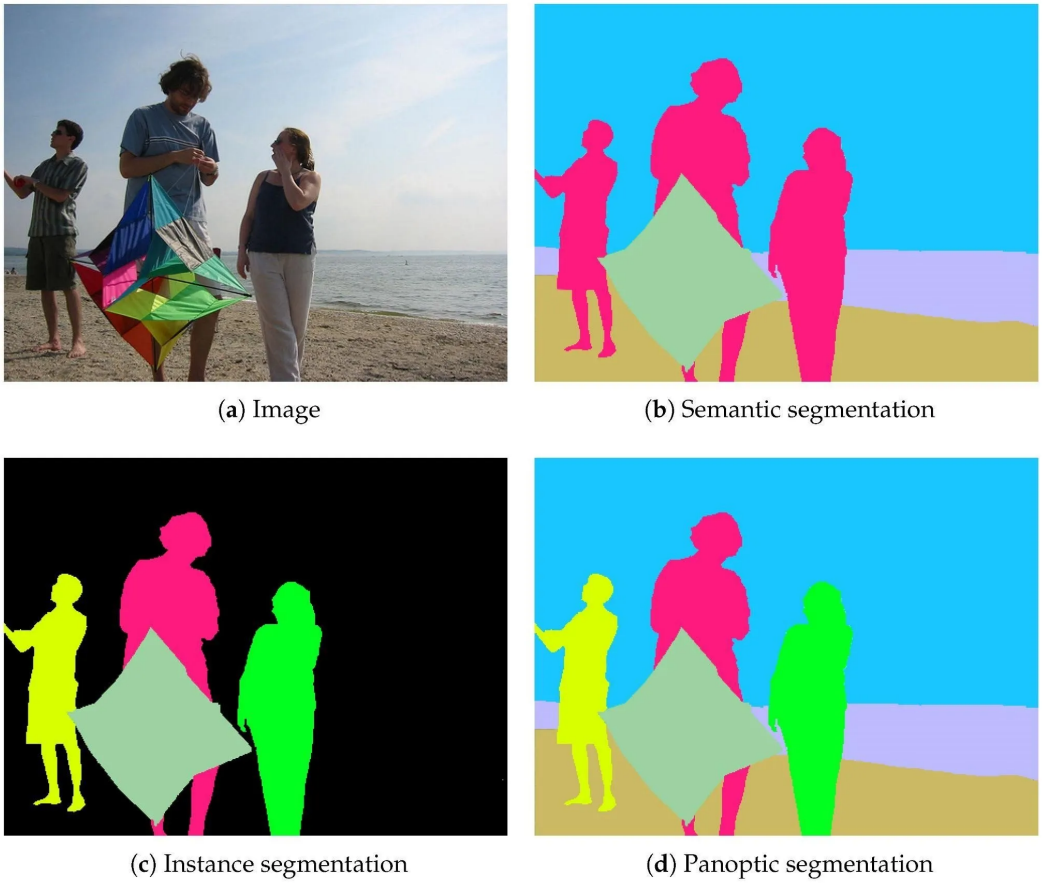
In order overcome around this restriction, they use *segmentation by instance*, which distinguishes each individual object furthermore to distinguishing the class, making it possible to identify and separate elements from the same category. Hafiz and Bhat [23] give a thorough rundown of the key architectures and technical challenges associated with this method, which has become relevant in situations like robotics and urban tracking. Last but not least is *panoptic segmentation* Zhang et al.[24], which combines the benefits of both of the previous two and gives each pixel both instance identity and semantic class information. This makes it possible to depict the scene in great detail, which is especially helpful in complicated settings like cities or manufacturing. The concept was formalised by Kirillov et al. [25], while more recent architectural solutions, such as UPSNet [26], demonstrate how semantic and per-instance segmentation can be effectively integrated in a single deep network.

An example of how the three different segmentations identify image information is shown in Figure 2.1

2.1.2 State of the Art

The concept of segmentation originated with the development of digital image processing and initially relied on "classical" methods. These approaches included:

- **Thresholding Techniques**, such as Otsu's Method [27], are utilized to distinguish the foreground from the background through image analysis.



(**a**) Image (**b**) Semantic segmentation

(**c**) Instance segmentation (**d**) Panoptic segmentation

Figure 2.1: Visual comparison of b) semantic, c) instance, and d) panoptic segmentation applied to the same image [3].

Contour detection algorithms, such as Canny and Sobel operators [28], [29], identify the edges of objects based on intensity gradients.

- **Methods based on pixel similarity**, such as Region-Growing and Watershed [30], [31], that segment an image by identifying groups of pixels with similar characteristics.

- **Non-supervised clustering techniques**, such as K-Means and Gaussian Mixture Model (GMM), divide images into clusters based on pixel statistics.

These approaches produced adequate outcomes, but their effectiveness was significantly reliant on image preprocessing quality and manual parameter selection, making them very dependent and limited in their usability. With the introduction of Deep Learning (DL), segmentation witnessed a dramatic transformation. Convolutional neural network (CNN) models enabled the automatic learning of pertinent features, overcoming the limits of previous approaches. The initial attempt was the Fully Convolutional Network (FCN) [32], which aimed to change networks created for classification by replacing final Multi Layer Perceptron (MLP) with convolutional layers. Upsampling operations restore the image to its original size, which allows pixel-by-pixel classification.

After the introduction of U-Net [33], a major step forward was taken in semantic segmentation, particularly in the medical field. Its architecture is derived from the AutoEncoder structure, but introduces a distinctive feature: skip-connections between symmetrical encoder and decoder layers. These connections allow the direct transfer of spatial features learnt during the encoding phase, preserving details that are essential for precise localisation and helping to avoid the disappearance of the gradient in the deepest phases of training. Due to its effectiveness and simplicity, U-Net has become a standard for several subsequent variants, such as U-Net++ [34], which further refines the connections between encoders and decoders through dense paths and variable depth, and V-Net [35], which extends the concept to 3D volumetric images. In parallel, in the field of instance-based segmentation, an important contribution has been made by Mask R-CNN [36], an extension of the Faster R-CNN [37] framework that adds a third branch to the network, dedicated to the prediction of binary masks for each detected object. This made it possible to effectively combine object recognition with pixel-wise segmentation, maintaining high performance even in the presence of overlapping objects. In more recent

years, the focus has shifted to transformers, originally designed for natural language processing. The Vision Transformer (ViT) [38] model also represented a paradigm shift in computer vision, replacing convolutions with a global attention mechanism on non-overlapping patches of the image. Although powerful, ViT requires large amounts of data to be effective, due to the lack of spatial inductive biases specific to CNNs. To address this limitation, Swin Transformer [39] was introduced, combining the efficiency of the transformer model with the locality of convolutions. By using shifted windows to compute attention in local regions, Swin preserves the hierarchical structure of features, making it more suitable for segmentation and classification tasks. Finally, SegFormer [40] proposed a fully transformer-based segmentation architecture, but with a lightweight and modular design. The transformer backbone extracts multi-scale features efficiently, while the decoder is designed to be simple yet effective, producing high-resolution segmentation maps. SegFormer performed excellently on several segmentation benchmarks, demonstrating that transformers can not only match, but in some cases exceed, CNNs even in dense tasks such as semantic segmentation.

2.2 Few parameter network

In recent years, research by the scientific community has been moving towards networks with a low number of parameters and reduced computational cost to make deep learning accessible even on edge and mobile devices, where resources such as memory, computing power and energy are limited. This approach allows for real-time inferencing, improves data privacy (by avoiding the use of the cloud) and reduces distribution costs. Furthermore, lightweight models favour greater energy sustainability, which is particularly relevant in large-scale scenarios. Efficiency is no longer an option, but a true necessity. In this context, lightweight architectures such as the MobileNet family are a benchmark. The first version, MobileNetV1 [41], introduced separable depthwise convolutions as a structural strategy to drastically reduce (up to 90%) the number of parameters, compared to traditional models such as VGG16. This innovation has shown that it is possible to significantly compress a network without excessively compromising its performance.

Afterwards, MobileNetV2 improved the concept by introducing the inverted residual block with bottleneck linear, obtaining an improvement in performance (about +1% on ImageNet) while keeping the computational complex-

ity invariant [42]. The evolution was consolidated with MobileNetV3, which integrated Neural Architecture Search (NAS) techniques and manual optimisations, inserting modules such as Squeeze-and-Excitation and more efficient activation functions such as hard-swish [43]. The 'Large' and 'Small' versions of MobileNetV3 have shown significant improvements in accuracy (+3.2% and +4.6% respectively) compared to the previous generation, while reducing inference latency on mobile CPUs.

At the same time, networks have been specifically designed to maximise efficiency on specific hardware through platform-aware NAS. This is the case of MnasNet, designed taking into account the computing characteristics of smartphones, which has achieved competitive results with just 4–5 million parameters [44]. Similar approaches have been adopted by FBNet [45] and ProxylessNAS [46], both aimed at customising the architecture according to the performance of the target device. A more recent strategy is the once-for-all model, which allows the generation of specialised lightweight variants for each execution environment starting from a generalist starting network.

In parallel with the development of lightweight architectures for smartphones, EfficientNet proposed a new paradigm of balanced architectural scaling, compound scaling, which allows the depth, width and resolution of the network to be increased smoothly [47]. This family of models, also originated via NAS, has shown that excellent results can be obtained even with the most compact versions (EfficientNet-B0), while the B7 version has achieved peak performance with a significantly smaller size compared to previous state-of-the-art architectures.

Following the trend towards parameter reduction, GhostNet explored the cost-effective generation of feature maps through linear operations, maintaining accuracy and reducing computational costs (Han2020GhostNet). With the same latency, GhostNet achieved slightly better performance than MobileNetV3. This direction was further taken to the extreme by MicroNet, which introduced factored micro-convolutions and innovative activation functions, achieving good results (up to 61% top-1 on ImageNet) with a minimal computational budget (only 12 MFLOPs), demonstrating the effectiveness of ultra-compact CNNs in less complex tasks [48].

Another emerging research line is the integration of attention mechanisms and Transformers in compact architectures. The MobileViT model is a prime example, merging the convolutional blocks of MobileNet with the global attention of Vision Transformers. The result is a model that, while maintaining

a small size (6M parameters), outperforms MobileNetV3 and compact Transformer models [49], highlighting how the combination of technologies can extend the capabilities of lightweight models.

Finally, the use of advanced architectural techniques, such as depthwise separable convolutions, SE blocks, inverted bottlenecks and balanced scaling, has shown that parametric efficiency is a viable way to obtain high-performance networks, especially in areas where computational resources are limited. The continuous evolution of models such as MobileNet, EfficientNet, GhostNet and MobileViT confirms that research is increasingly moving towards optimised, versatile and scalable architectures, capable of extending artificial vision to embedded and real-time contexts, without significantly compromising recognition quality.

2.3 Generative Models

Medical imaging represents a fundamental aspect of modern diagnostics, offering detailed, noninvasive visualization of human anatomy and physiology. Among the most widely used techniques, magnetic resonance imaging (MRI) and computed tomography (CT) are distinguished by their ability to generate high-resolution representations of tissue structures, playing a crucial role in therapeutic planning and disease follow-up. MRI, based on the principles of nuclear magnetic resonance, allows the acquisition of multiplanar images with optimal contrast between soft tissues, while CT, based on the use of X-rays, allows rapid and precise three-dimensional reconstructions, which are particularly valuable in urgent clinical settings.

Technological advances in this area have greatly expanded the applications of these methods, fostering integration with artificial intelligence algorithms [50] and computational analysis techniques, resulting in increased diagnostic accuracy. In particular, recent developments in the area of deep learning have enabled new capabilities, such as the prediction of diagnostic information from heterogeneous examinations [51], paving the way for significant advantages in the estimation of radiant dose during therapeutic planning and optimization of treatment placement by avoiding overexposure of healthy tissues.

Such MRI-based CT-equivalent images are often referred to as pseudo-CT or synthetic CT (sCT), an example of which can be seen in Figure 2.3 [52, 53, 54]. In Figure 2.2, on the other hand, summarises the training, validation and testing steps of a generic deep learning algorithm.

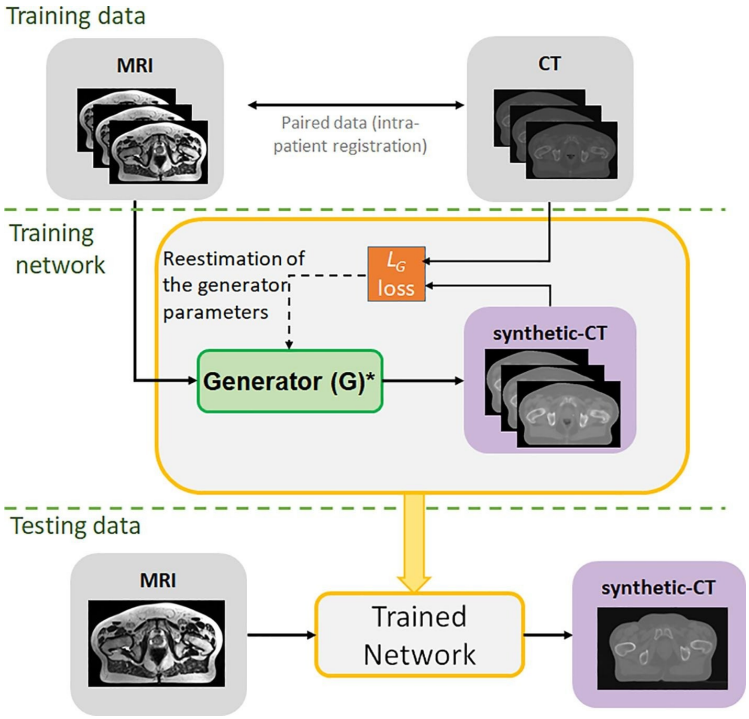


Figure 2.2: A schematic representation illustrating a typical procedure for training for a generative algorithm. [4]

The 2020 publication by Wong et al. [55] provides a practical application of the MRI-derived sCT concept in the context of intensity-modulated radiotherapy (IMRT) for lymphoma treatments, representing a promising alternative approach. CT scans are usually used for total bone marrow and lymphoid irradiation, however, this process can take a long time for target delineation and treatment planning; MRI scans are generally easier and faster to segment than CT scans thanks to the possibility of highlighting body parts such as soft tissues with greater contrast. The generation of accurate sCT from MRI images is also fundamental for PET attenuation correction in hybrid PET/MRI systems [56] [57] of which a functional diagram is shown in 2.4.

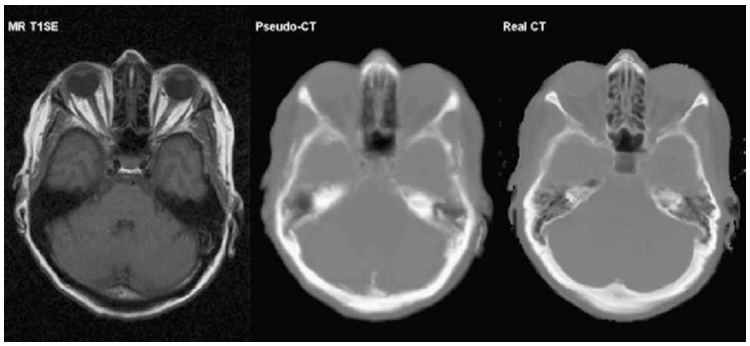


Figure 2.3: The source MRI image, the derived sCT and the original CT.

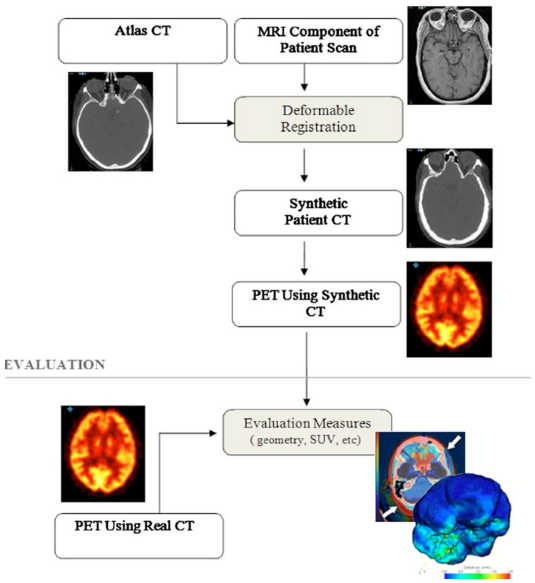


Figure 2.4: Representation of the procedure for creating a PET from a sCT. [5]

Many different methods for the automatic generation of sCT from MRI images have been proposed in the scientific literature. They can be divided into three main categories:

- **Approaches based on tissue segmentation** that consist of segmenting or classifying the voxels of the MRI images into different classes of tissue, such as air, fat, soft tissue and bone, and then refining them manually. Mass density assignment is then used to obtain a different CT scan number for each tissue type. Tissue segmentation is not a simple problem and an MRI image volume is usually insufficient to separate all the main tissue types. Furthermore, conventional MRI sequences are not able to accurately differentiate bone tissue from air [58]. Consequently, most tissue segmentation methods require the use of multiple MRI sequences, especially those specialised in ultrafast echo time (UTE), which involves longer image acquisition times and a more complicated workflow [54] [59].

- **Atlas-based approaches** that apply the image registration technique to align a target MRI image with one belonging to the medical atlas; an *atlas* [60] is a collection of medical images of many subjects belonging to a heterogeneous class, both healthy and affected by pathologies, which form a database containing anatomical information useful as a reference for diagnosis, clinical applications and for finding correspondences be-

tween images of different individuals. The correspondence information can be used to deform the image to the target MRI and generate an sCT. These approaches are valued for their ability to produce reliable estimates; however, they require accurate deformable registration of the atlas and patient MRIs. This information can be difficult to obtain, especially in the presence of wide anatomical variations or pathological variations. To mitigate this negative effect, multiple atlases and hybrid methods that combine atlas-based methods with prediction-based methods are used [54] [59].

- **Learning-based approaches** that use statistical learning or model fitting techniques to construct a non-linear mapping function that attempts to associate MRI voxel intensities with the corresponding CT ones. However, this approach requires additional information to that provided by conventional MRI alone; in fact, it is often hard to clearly identify the different areas within the volume. The supplementary information includes, for example, bone volume, and the use of multiple sequences such as UTE to increase the discrimination between bone voxels and air.

  To improve the accuracy of sCT prediction, models have also been designed and used that analyse intensity variations within a restricted (local) area of the MRI image and coordinate-related characteristics, such as the position of a voxel in relation to a specific anatomical structure [54] [59].

The latest category includes both machine learning and deep learning techniques, whose progress in recent years has allowed diagnostic systems to plan radiotherapy treatments based on MRI images alone.

The task of generating synthetic medical images became relevant between 2016 and 2017, due to the first approaches proposed by Nie et al. [61] and Xiang et al. [62]. Both authors exploited state-of-the-art models based on FCN and autoencoders, demonstrating the feasibility of such a complex task. The escalation of increasingly complex architectures, such as U-Nets and GANs, allowed for great strides forward in the domain. These had demonstrated their improved capacity in feature extraction, allowing generative algorithms to improve images also in terms of detail.

The pioneer in using U-Net for generation has been Han et al. [54] who published a study on the generation of sCT using a U-Net, similar to the model proposed by Ronneberger [63] for image segmentation but without the

three fully connected layers. This variation led to a 90% reduction in the number of parameters. In Figure 2.5, the model proposed in Wang et al. [6] is shown, where we can see the use of batch normalization and LeakyReLu as the activation function.

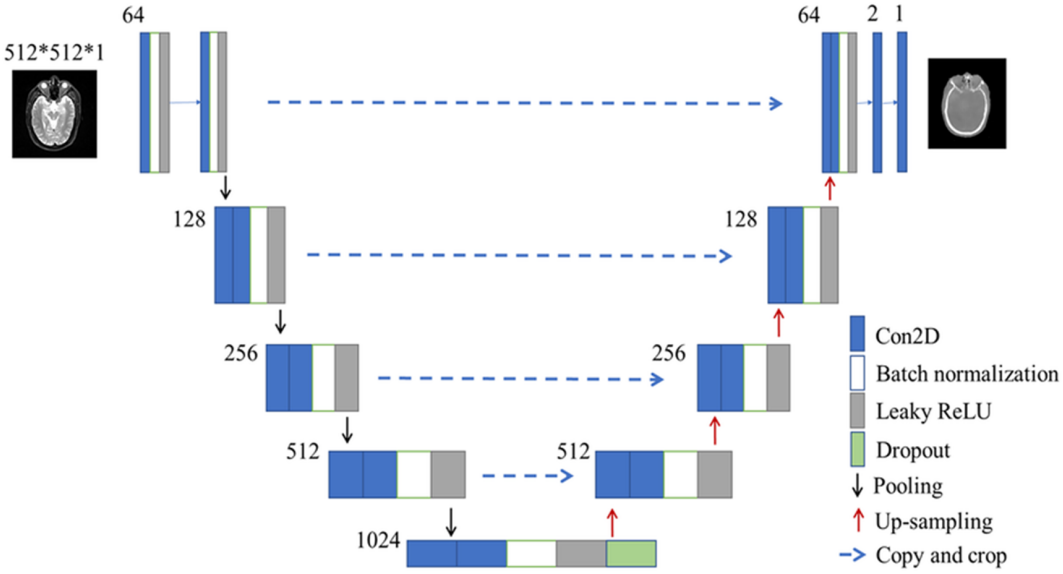


Figure 2.5: Architecture proposed by Wang et al. [6].

In Jang et al. [64] a classic encoder-decoder structure has been used, i.e. without skip connections.

In the Dong et al. paper [57], it could be observed that skip connections in U-Nets could carry irrelevant noise from the input image. To try to solve the problem, they developed a self-attention strategy that, starting from the encoder's feature maps, is able to identify only the most relevant ones, reducing the noise propagated in the decoder.

The special feature of this work is the inclusion of a fully connected conditional random layer [65]; this probability-based statistical model also takes neighbouring voxels into account when predicting labels, improving overall performance.

As can be seen from the earliest approaches, the autoencoder model has been the starting point for the development of various architectures, such as Deep Embedding CNN (DECNN) or Embedded Net [66], Efficient CNN (eCNN) models [67] and ResNet [68]. In fact, all these architectures can be classified as variants of the U-Net: DECNNs involve the insertion of multiple embedding blocks to improve gradient propagation during training, resulting in faster convergence and easier training of synthetic images, as demonstrated by the study by Xiang et al [66]; in eCNN the standard convolutional layers are replaced

with specialised structures to extract features from the input images [67]; finally, ResNets are architectures composed of three convolutional layers (with convolution, batch normalisation and ReLU activation function), followed by nine residual blocks (with convolutional layers, batch normalisation and ReLU activation function) and fully connected layers.

The innovative approach suggested by Nie et al. [69] has transformed the way synthetic images are created, allowing for results that are closer to reality and with a reduced presence of artefacts. Two fundamental components, the *discriminator* and the *generator*, were presented; they derive from game theory and apply a *Min-Max* approach. In fact, this technique aims to maximise the objective function, minimising that of the 'opponent' (adversarial). These models are called Generative Adversarial Networks (GAN) after this contest. The objective function is shown in equation 2.2:

$$L_{Gan} \;=\; E_{x\ p(x)}\ [log\ D(x)] + Ez\ p(z)[log(1 - D(G(z)))] \tag{2.2}$$

In equation 2.2, G and D refer to the **G**enerator and the **D**iscriminator, respectively; x is the real sample extracted from the distribution of real data and z is the random noise used by the generator to create fake data. The discriminator D will try to understand if the samples it is receiving are part of the real distribution or the generated one, while G aims to create synthetic samples with a distribution similar to the original one.

The breakthrough introduced by GANs has been very successful and for this reason, several variants of them have been proposed over the years for the synthesis of medical images, including Conditional GANs (cGANs) [70]. Unlike traditional architectures, cGANs incorporate a constraint in the objective function (e.g. labelled data or an image) to improve model control by generating more realistic images that are consistent with the conditioning input. For example, to generate sCT images, labelled MRI data can be used as a constraint. The objective function will then be represented by the formula:

$$L_{cGan} \;=\; E_{x,y}\ [log\ D(x,y)] + Ex,z[log(1 - D(x, G(x,z)))] \tag{2.3}$$

A further variant is CycleGAN, introduced by Zhu et al. in 2017 [71]. They use two pairs of generator and discriminator for image synthesis between different domains (A and B) without the need for paired images. The two generators are G_A, which translates images from domain B to domain A, and G_B, which translates images from domain A to domain B. The objective function is

composed of that of the classic CGANs L_{GAN} (G_A D_A) and LGAN (G_B D_B), in addition to an additional cyclic consistency loss (L_{cycle}) that calculates the pixel-wise or L_1 loss between real and cyclically generated data. The objective function shown is in 2.4.

$$L_{cyc}\,(G, D) = \; E_{\;A\;\,p(A)}\,[||D(G(A) \; - A||_1] + EB\,p(B)[||G(D(B)) - B||1] \quad (2.4)$$

In recent years, new transformer-based approaches have been implemented for the image generation task. Zhao et al. [72], in their paper published in 2023, used a deeper transformer-based feature extractor as the bottleneck layer of an encoder-decoder generative network. Specifically, this block consists of multiple transformer layers and a convolution layer to which a residual concatenation has been introduced before calculating the self-attention, further improving the network's stability. In the ResViT conditional network, proposed by Dalmaz et al. in 2022 [73], the generator bottleneck is modelled using a series of aggregated residual transformers (ART) consisting of a combination of residual convolution layers and transformer layers. To reduce the memory requirements generated due the use of multiple ART blocks, a method of parameter-sharing has been adopted in the bottleneck, which optimises the computational efficiency of the model during the training process.

The results obtained from these hybrid architectures have led to a growing trend in development towards combining CNNs and Transformers with the purpose of exploiting the potential of both types of layer: the former to extract local details and the latter to provide a global understanding of the image structure. As an example, Zeng et al. [74] have proposed an innovative method to effectively combine CNN and transformer by replacing the linear embedding and the linear projection layers in the transformer with convolutional embedding and projection.

Another type of generative model widely used in the medical domain is the Denoising Diffusion Probabilistic Model (DDPM). They have been introduced for the first time by Ho et al. in 2020 [75] and as shown in Figure 2.6 they are a parameterised Markov chain that is trained to map the pixels from noise through a gradual process over a finite number of time steps T.

Markov chains are models that describe a sequence of events in which the probability of transition from one state to another depends only on the current state and not on the sequence of previous events. The parameterisation, i.e. the possibility that the parameters vary as time or other factors change, allows

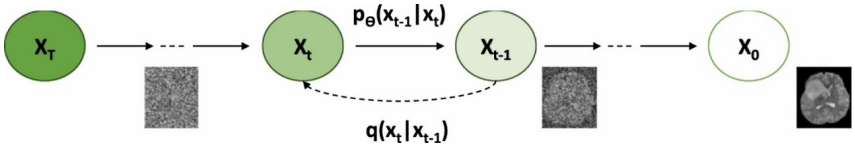


Figure 2.6: Illustration of a Markov chain modulating noise distribution in DDR approaches.

the model to adapt and learn the distribution of data over time. The learning process consists mainly of two phases, the forward and the inverse. In the forward process, random Gaussian noise is added to the input image x_t in a series of time steps large enough to obtain a noisy image from an isotropic Gaussian distribution (N). This is expressed by the equation:

$$q(x_t|x_{t-1}) = N(x_t;\ \sqrt{1\ -\ \beta_t}\ x_{t-1},\ \beta_t I) \tag{2.5}$$

Where β_t is the parameter that represents the variance of the noise added at time step t. When multiplied by the identity matrix I, it indicates that the added noise is isotropic, therefore equal in all directions. This process forms a Markov chain through which the average distribution of the current step is conditioned to the sample of the previous step following a noise variance schedule as in the equation 2.6:

$$x_t = \sqrt{1\ -\ \beta_t}\ x_{t-1} +\ \varepsilon\ \sqrt{\beta_t}\ ;\ \varepsilon \quad N\ (0,\ I) \tag{2.6}$$

where ϵ is the added noise, N is the Gaussian distribution and I is the identity covariance matrix. Thanks to the Markov property of this process, the marginal distribution can be obtained directly with a given input sample x.

The reverse diffusion process:

$$p_\theta(x_{t-1}|x_t) \tag{2.7}$$

begins with the pure noise distribution, random or unstructured data that does not carry significant information. At each time step, the noise is gradually removed from the sample, thus building a Markov chain from x_T to x_0 as shown in Figure 2.7.

With a smaller β_T the transition between x_t and x_{t-1} can be approximated as a Gaussian distribution since both the forward and reverse processes have the same functional form:

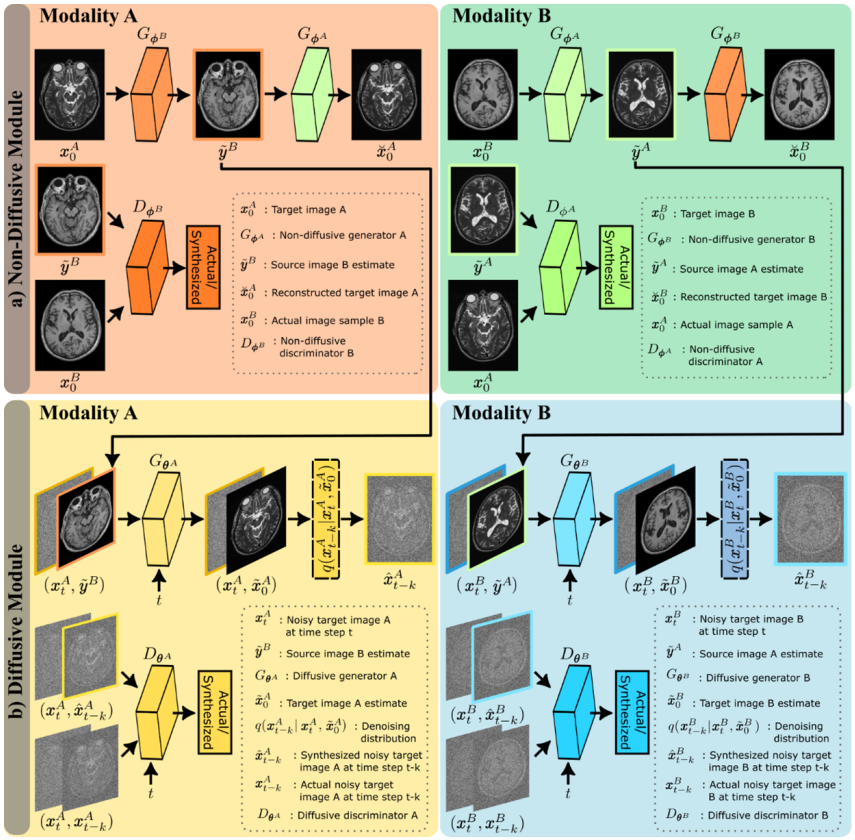


Figure 2.7: SynDiff architecture combining a dual diffusion and nondiffusion approach. [7]

$$p_\theta\left(x_{t-1}|x_t\right) = N(x_{t-1}; \mu\theta(x_t; t, \sum \theta(x_t; t)) \tag{2.8}$$

Each inverse phase of the diffusion models is mapped through a neural network p_θ, which is trained by optimising the lower variational bound (LVB) on the simplified log-likelihood, a statistical measure that evaluates the effectiveness with which the model represents the observed data. The LVB allows us to approximate the log-likelihood of the real data with an objective function that can be optimised during training. The network p_θ estimates the added noise ϵ by minimising the loss between the actual noise and the network estimate ε_θ as follows:

$$L_{error} = x_{t,x_0,\varepsilon}[||\varepsilon - \varepsilon_\theta\left(x_t\right)||_2^2] \tag{2.9}$$

At each inverse step $t \in 1, T$, the mean distribution $\mu\theta$ is derived using ε_θ and the sample x_{t-1} as in the previous equation [57].

Although diffusion-based medical image synthesis has shown promising results, its effectiveness is limited due to the high computational load required

for image sampling, which is characteristic of likelihood-based models. Consequently, most recent studies, such as that of Rombach et al. [76], have focused on reducing the computational resources required for modelling.

The architectural innovations introduced focus on conditioning the inverse process. In particular, two approaches can be distinguished: conditioning by means of a classifier and conditioning without a classifier. The first [77] involves using a classifier to guide the image generation process, influencing the distribution of noise during the inverse denoising phase. However, this method has limitations due to its strong dependence on the performance of the classifier, which may vary depending on the dataset or context. In the classifier-free approach [78], the conditioning takes place in a supervised manner; the conditioning information is incorporated directly into the inverse process.

Lyu and Wang [79] have developed a CT image synthesis model that is conditioned to magnetic resonance images through a diffusion process guided by an approximate log-likelihood score. They used pairs of co-registered images, i.e. CT and MRI images acquired in the same subject and registered in the same coordinate space to allow the model to generate realistic images consistent with the originals.

U-Nets have been widely used in diffusion models for supervised conditioning, where the model learns to remove noise guided by examples of clean images. However, there are also adversarial diffusion models, such as the ones introduced by Özbey et al. [80], in which the model learns to generate realistic images without the use of labelled training examples.

Meng et al. [81] have expanded the concept of conditioning through a multimodal approach, allowing the network to synthesise images by simultaneously considering all available imaging modalities to maximise the use of information and improve the accuracy of the generated images. For example, when generating an MRI image from a CT and a PET scan, the model considers all three modalities to produce a coherent, high-quality MRI image.

Recently, Zhu et al. [82] extended diffusion models from 2D to 3D, enabling more realistic synthesis of three-dimensional images with lower computational requirements. Finally, Pan et al. [83] proposed a Transformer-based diffusion model with Swin-ViT that outperforms previous diffusion models in terms of both image quality and Fréchet Inception Distance (FID) scores, a metric that quantifies how similar the generated images are to the real ones. This innovative architecture significantly improves the efficiency and scalability of the diffusion model, allowing it to manage high-resolution images with limited

computational resources.

2.4 Few Shot Segmentation

Few-shot semantic segmentation (FSS) aims to overcome the difficulty of obtaining large datasets with pixel-level annotations. The objective of FSS is therefore to generalise to new categories with minimal annotation effort, partially emulating the human ability to learn visual concepts from a few examples.

To address learning problems with few examples (few-shot learning), a commonly used strategy is episodic learning, a meta-learning approach that simulates, already during the training phase, typical conditions that the model will encounter in testing. Instead of performing training conventionally on all classes in the dataset, the model is exposed to a sequence of episodes. Each episode is constructed by randomly picking a subset of classes (called the support set), from which a few annotated examples are extracted for each class. Contextually, a query set is generated containing images belonging to the same classes but not included in the support set. The model is then trained to correctly segment the images in the query set based solely on the information provided by the examples in the support set. This episodic pattern allows the model to learn to quickly generalize to new classes in the presence of an extremely small number of annotations. This is particularly relevant in the context of medical segmentation, where the availability of labeled data is often limited and the cost of clinical supervision is high. Over the last few years, research has produced a wide range of approaches, in particular two main strands: approaches based on matching networks/metric learning and approaches based on optimised meta-learning (e.g. Model-Agnostic Meta-Learning and variants). Both paradigms aim to solve the problem with few examples, but with different strategies:

- the former focus on similarity metrics in the feature space between support images and queries,
- the second try to learn how to quickly adapt the parameters of a segmenter to new data.

The **metric-based approaches** tackle few-shot segmentation by building a model of similarity between the representations of the support image (with its mask) and those of the query image. Basically, instead of updating the

weights of the architecture, they extract visual characteristics from the images and directly compare the features to decide whether a query pixel belongs to the given (support) class or otherwise. This paradigm descends from Matching Networks, introduced in the context of few-shot classification [84] and from Prototypical Networks [85], adapted to dense segmentation. In practice, a few-shot segmentation model typically has a double branch: a feature encoder for the support images (which also uses the support mask to extract an embedding representative of the target class) and an encoder for the query image. A matching module then compares the features of the two branches for each position of the query.

For example, CANet [86] (Zhang et al. 2019) uses a Dense Comparison module to compare the features extracted from support and query at multiple levels, thus creating a similarity map. This map is iteratively processed to refine its quality until the final prediction is obtained. Another approach, which has proved to be one of the most popular in the community, is PANet [87] which introduces a mechanism based on the alignment of bidirectional prototypes. PANet calculates a prototype for the support class, which is used to predict the query mask, and this is then used to generate a prototype useful for predicting the support mask. This bidirectionality allows consistency between the prototypes of both sets, increasing the segmentation capacity of the architecture.

Clearly, PANet has been the founding architecture of today's Prototype-based approaches to FSS, as demonstrated by the variants that have been published since it was first published. For example, Prototype Matching Nets such as SG-One [88] simplified the architecture by guiding query segmentation through a single prototype vector extracted from the support, to which masked average pooling operations are applied to preserve spatial information.

An additional example is provided by ASGNet [89], which introduces multiple prototypes to represent different components of the same class: instead of a single average vector, the support is subdivided through a Superpixel-guided Clustering approach into subclusters of features (e.g. parts of the object), that allow the computing of a set of adaptive prototypes that will be associated with the query features. This strategy is successful in cases where the classes have great internal variability (e.g. different poses of an object).

Multi-prototypes have proven to be effective, so much so that in 2020 Gairola et al. [90] presented SimPropNet. It simultaneously predicts support and query masks using a shared decoder. This approach forces semantic alignment

between support and query features, improving similarity propagation. It also introduces the Foreground-Background Attentive Fusion (FBAF) mechanism that exploits similarities in both foreground and background regions between support and query images.

Multiple-prototype approaches allow for more effective handling of different classes that nevertheless remain relationally separate. To address this limitation, PGNet proposes an approach based on Pyramid Graph Neural Networks (PGNet) [91] that explicitly models the relationships between multiple support and query images, where nodes represent object prototypes and messages in the graph transfer context and refine the match. This allows local semantic information to be captured at various levels of granularity. Unlike previous methods that use global features, PGNet searches for matches between specific regions of the support and query images.

More recently, the use of Transformers has made it possible to implement matching modules such as cross-attention: for example CyCTR [92] (Zhang et al.) uses a Cycle-Consistent Transformer that alternates attention between support and query features, matching relevant regions of the two images and producing more accurate masks, especially for small or occluded objects. Similarly, HSNet [93] (Min et al.) introduces the so-called Hypercorrelation Squeeze, building a high-dimensional representation of all local support-query correlations and using a network block to 'compress' these correlations into a segmented mask. In general, metric matching approaches have the main advantage of not needing further fine-tuning on the new class: the model, trained on episodes of different classes, performs a forward pass at test time to extract prototypes or affinities and segments the query directly.

One of the most common problems encountered is lack of performance when the support set is not well reflective of the sample set, which results in low variability and leads to segmentation errors. Furthermore, the problem of support set dilution has more recently been observed; in fact, increasing the number of support examples (going from 1-shot to 5-shot) does not always result in a monotonic increase in performance; on the other hand, some methods degrade due to redundant and conflicting information between the supports. Some proposed solutions such as [94] use adaptive fusion techniques of support information to prevent the addition of new examples from 'diluting' the discriminative characteristics, maintaining the benefit of more data without incurring local overfitting.

Approaches based on optimised Meta-Learning are the second large

family of solutions specifically employs meta-learning concepts to the problem, attempting to learn a segmentation model that can be quickly adapted to new classes with few gradient steps, or can produce the parameters for new classes directly. These methods often share the use of episodic training like the previous ones, but emphasise the internal optimiser or the conditional generation of weights on the support data. A ground-breaking approach was One-Shot Segmentation (OSLSM) [95] by Shaban et al. (2017), which introduced a meta-model called Weight Generator: given the support image and its mask, a network that adapts the weights of the final classifier of a segmentation network, to predict the masks of the new class. Several works have since focused on the prediction of conditional parameters: for example, the Adaptive Masked Proxies method [96] (AMP) method (Siam et al., ICCV 2019) directly constructs the weights of the last segmentation layer from the few examples, by using multi-resolution pooling on the features masked by the support to obtain positive class proxies and merging them with the 'signatures' of the previously seen classes. Interestingly, AMP does not require a second dedicated branch to compute prototypes or parameters – unlike many metric architectures – making it flexible for integration with other segmentation networks (e.g. combining appearance and motion streams for video). Model-agnostic meta-learning (MAML) is another category of meta-learning that consists of rapid optimisation. In this scheme, the model is trained to obtain a set of initial parameters, which with a fine-tuning on the k samples of a new class lead to good performance. Applying MAML directly to segmentation is computationally expensive (given the high dimensionality of the parameters), but variants have been proposed. For example, Hendryx et al. [97] explore 'Meta-Learning Initialisations for Image Segmentation': they propose using meta-learning to find an optimal initialisation for a segmentation model, so that it is able to adapt quickly to new tasks with little data. The results show that the application of this approach achieves better performance than when a randomly initialised or pre-trained model is used. Russwurm et al. [98] apply MAML in Meta-Learning for Few-Shot Land Cover Classification of unseen geographical regions, where the approach proves effective when there is a shift between the source and target domain distributions. Nevertheless, in ideal scenarios (identical distributions), traditional pre-training performs better.

Zhu et al. [99] in their paper entitled Self-Supervised Tuning for Few-Shot Segmentation report how a simple meta-learner can fail to produce discrimi-

native descriptors if the features extracted from the support are compressed in the embedding space. They therefore propose a framework with a double adaptation loop: a self-supervised inner loop in which the network (called the base learner) solves a self-supervised task on the same support image (for example, recomposition of a 3D puzzle of the features or an unlabelled self-segmentation). This branch allows stronger latent features to be extracted, while a meta-learning outer loop episodically updates the network parameters so that auto-supervised features improve few-shot segmentation; this structure emphasises self-learning and enhances traditional meta-learning in segmentation.

While metric-based methods avoid fine-tuning on the support, some meta-learning approaches combine matching with small gradient steps. For example, 'RePRI: Few-Shot Segmentation via Rich Prototype Refinement and Interpolation' [100] presents an algorithm without explicit fine-tuning which iteratively updates the query mask prediction using its own previous refined estimation (self-refinement) as a guide. In practice, RePRI uses the mask predicted in the previous iteration as an additional signal for the next step. This approach is very similar to the process of optimising architectures, but acts on the features of the predicted image rather than on the network weights. This strategy allows to avoid the costly optimisation for each new class, achieving optimal results comparable to state-of-the-art architectures.

At the same time, more complex tasks are being addressed, such as the segmentation of multiple unseen classes simultaneously (instead of one class at a time) and weak few-shot segmentation (where the supporting annotations are partial or image-level instead of complete masks). For example, a recent variant is incremental few-shot segmentation [101] [102], in which classes with few examples are progressively added, pushing the model to enlarge its cohort without forgetting the previous classes; this strategy combines meta-learning and lifelong learning. Some works have introduced continuous meta-learning modules, for example a meta-module whose parameterisations are shared between the incremental phases to transfer knowledge from old classes to new ones

2.5 Metrics

This section presents the metrics used within this thesis work, highlighting the choice guided by the nature of the task and the type of data processed.

The overview aims to provide knowledge to correctly interpret the results presented in the following chapters, as well as to understand the significance of the comparisons made during the experimental validation phase.

In relation to the Classification task, to evaluate the ability to correctly predict the classes for each data, different metrics were used, such as:

Accuracy (ACC), i.e. the ratio of correct predictions to the total number of observations (eq. 2.10).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

and the *ROC-AUC* (eq. 2.11), a graphical tool that offers a visual representation of the ability of a classifier to distinguish between classes, showing the relationship between sensitivity, or True Positive Rate (TPR), and the False Positive Rate (FPR).

$$\text{AUC} = \int_0^1 TPR(FPR^{-1}(x))\, dx \tag{2.11}$$

In the absence of classical evaluation metrics such as those proposed and used in the Grad-CAM paper [103] (e.g. *Rank Correlation w/ occlusion*) due to the limited amount of information provided, it was essential to introduce the coefficient C, which evaluates the degree of coverage of the predicted ROI with respect to the ground-truth. Specifically, C is calculated according to the equation 2.12, where T stands for the ground-truth and P for the features that have been predicted.

$$C = \frac{T \cap P}{T} \tag{2.12}$$

As can be seen in the Figure 2.8, when the value of P partially covers T, we will have a value of $C < 1$ (a); when P totally covers T we will have a value of $C = 1$ (b and c).

Furthermore, the semantic segmentation is evaluated using the Dice Score, also known as Dice Similarity Coefficient. This measure compute the similarity between two images, typically represented as binary arrays. The Dice score is calculated as described in the equation 2.13.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{2.13}$$

Where |X| and |Y| are the cardinalities of the two images.A value of 1 indicates perfect overlap, while 0 indicates no overlap

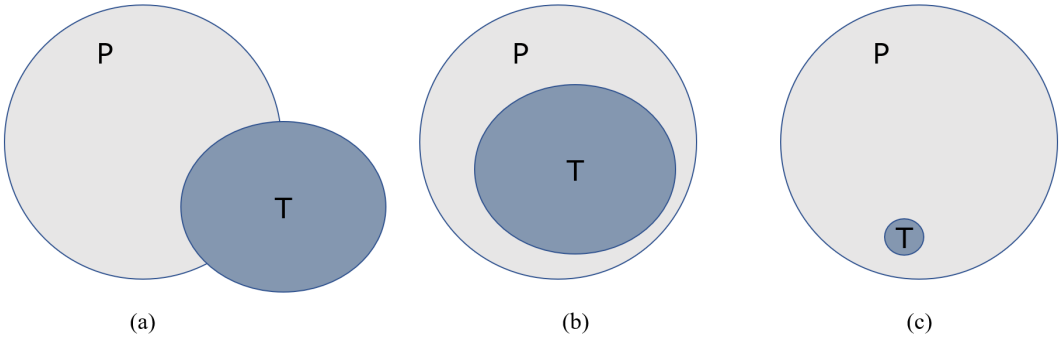


Figure 2.8: Diffent cases where $C > 0$.

For the generation the model were evaluated using metrics commonly used in the literature. The metrics selected can be divided into two categories: i) metrics for pixel value assessment (e.g. ME, MAE) ii) metrics for overall assessment of the generated image (PSNR, SSIM). These metrics will be described in detail in the following sections. The pixel-wise metrics used are: The ME (Mean Error) is a measure of the average error between the original and generated images. Depending on the sign (positive or negative), it indicates an overestimation or underestimation with respect to the original image. It was calculated using the formula:

$$ME = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i) \tag{2.14}$$

where N is the number of pixels, x_i is the value of the original pixel and $\hat{x}_i$ is the value of the generated pixel. The unit of measurement is the Hounsfield unit (HU).

The Mean Absolute Error (MAE) is similar to the previous one but calculates the difference in absolute form:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i| \tag{2.15}$$

Mean Squared Error (MSE) measures the average of the squares of the errors between the original images and the generated images, penalising the larger errors more:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{2.16}$$

The measure unit is the HU^2 unit.

Lastly, the Root Mean Squared Error (RMSE) is an absolute measure of the

discrepancy between the generated output and the reference. It is obtained by squaring the differences between the corresponding values, an operation that avoids the cancellation of deviations of opposite sign and emphasises the larger errors. The RMSE is calculated as the square root of the Mean Squared Error (MSE), and is therefore expressed in the same unit of measurement as the original data, in this case Hounsfield units (HU).

The global evaluation metrics used are:

The Peak Signal-to-Noise Ratio (PSNR), is a measurement that quantifies the ratio between the maximum possible value of the image pixel intensity and the noise level, the latter represented by the RMSE. PSNR is expressed on a logarithmic scale, with units of measurement in decibels (dB), and is particularly useful for evaluating the quality of image reconstruction or synthesis. The higher the value, the greater the visual similarity between the two. It is computed as follows

$$PSNR = 20\ log_{10}\ \frac{{MAX_I}^2}{\sqrt{MSE}} \tag{2.17}$$

where MAX_I is the maximum possible value of a pixel in the image.

The SSIM (Structural Similarity Index) evaluates the similarity between two images considering luminance, contrast and structure. The value, which is dimensionless, varies between -1 and 1, where 1 indicates that the images are identical. It can be calculated using the formula:

$$SSIM\,(x,\ y) = \frac{(2\mu_x\mu_y\ +\ c1)(2\sigma_{xy} +\ c2)}{(\mu_x{}^2\ +\mu_y{}^2\ \ +\ c1)(\ \sigma_x{}^2 +\ \sigma_y{}^2\ +\ c2)} \tag{2.18}$$

where μ_x and μ_y represent the average of the samples of x and y, ${\sigma_x}^2$ and ${\sigma_y}^2$ the variance, σ_{xy} the covariance between x and y. $c1$ and $c2$ instead represent two variables that have the function of stabilising the denominator so that it is not too weak.

Chapter 3

Datasets

As discussed in section 1, the results obtained with new DL technologies have consolidated its use in everyday life. Its extremely diverse applications require specific data to address heterogeneous computational tasks. However, these abilities are intrinsically linked to the quantity and quality of the training data, as they represent the cognitive basis through which the model obtains its understanding. Therefore, the link between data and models is fundamental and inseparable. The use of limited, unrepresentative or erroneous data can lead to models with a poor generalisation capacity, ineffective in producing accurate forecasts on data never observed or even to the learning of spurious correlations due to artefacts in the data. In order to standardise the evaluation of model performance, the scientific community has identified the need for shared datasets, known as benchmarks, which define the reference standards for comparing different approaches. However, these datasets are typically generic and not specific to a particular application domain, as they are designed to evaluate models on well-defined tasks. Only after having validated their performance on these benchmarks, are the models evaluated on domain datasets, which determine their actual effectiveness in the target application context.

In the early stages of experimentation, particularly with the Learn&Drop application, general-purpose datasets such as MNIST, Imagenette and CIFAR-10 were used. The objective was to validate the effectiveness of the proposed methodology in a controlled and established context, exploiting benchmarks widely used in the community to highlight the general potential of the approach. Subsequently, the development of the Few-Parameter Architecture (FPA) marked a gradual transition to the biomedical domain. In this transition phase, datasets such as MedMNIST, UW-Madison GI Tract and Brain Tumor were adopted, with the aim of testing the framework on real classification and segmentation tasks in the clinical domain. These datasets represent hetero-

geneous scenarios in terms of complexity and type of medical image, making them suitable for evaluating the generalisation capabilities of the models. Regarding the integration of artificial intelligence in clinical workflows, explored through the IODeep application, the models were integrated directly within a Server PACS and tested in DICOM-compatible environments. Again, the models used for segmentation were trained on UW-Madison and Brain Tumor, in order to ensure methodological continuity with the previous phases. In the context of generative applications, the CERMEP dataset was used, which is particularly suitable for domain shift studies due to the availability of images acquired with different modalities (CT and MRI) for each subject. This made it possible to develop and test controlled generation models while maintaining consistency between domains. Finally, the ISIC, DeepGlobe and Chest X-Ray datasets were selected for the few-shot segmentation (FSS) task. These sets are commonly used in the literature for comparative evaluation and allow direct comparison of the performance of the proposed models with those of the main state-of-the-art approaches.

In the following sections, the datasets used in this thesis will be discussed and described.

3.1 MNIST

Certainly, the most famous and widely used toy-dataset is that of the MNIST(Modified National Institute of Standards and Technology database) [104], formed by hand-written numerical digits was invented by the U.S. Postal Service to identify home addresses automatically.

Figure 3.1: Few samples from MNIST dataset

The images are gray scale, with the size of $28x28$, the whole data collection consists of 60000 images for training and 10000 for testing. Being composed of the digits 0 to 9, it is often used for multiclass classification. MNIST is the

evolution of NIST, receptively a new extended version was released EMNIST [105] with more samples. Over time, this dataset has become so well established that it is considered "hello world" for those new to machine learning. To determine the functioning of one of the approaches to be proposed, which will be described in Chapter 5, a modified version of MNIST was created. This new version, starting from the original data, randomly groups 4 digits with the only condition that at least two of them are the same; to further complicate the structure of the information, 4 different geometric transformations will be applied to the 4 digits (e.g. translation, rotation, shearing). An example of the structure of the mod-MNIST specifically built to stress the network's localisation capacity is shown in Figure 3.2.

Figure 3.2: Example of our mod-MNIST

3.2 CIFAR10

CIFAR10 (Canadian Institute For Advanced Research) [8] is a widely utilized benchmark dataset for machine learning and computer vision tasks. Composed of 60000 photos, with 50000 for training and 10,000 for testing. The samples are in RGB format, 32×32, and divided into ten groups (airplanes, vehicles, birds, cats, deer, dogs, frogs, horses, ships, and trucks), each with 6000 samples. An example of the images present in CIFAR10 is shown in Figure 3.3.

Datasets of this type, which contain a wide range of information, are extremely valuable since they can highlight the models' expressive capabilities. The enlarged version of CIFAR10, called **CIFAR100**, includes 100 classes,

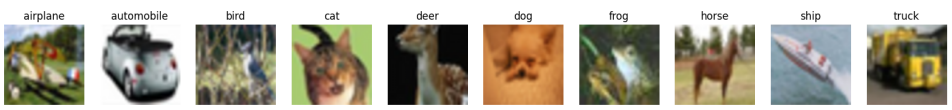


Figure 3.3: One sample for each class from CIFAR10 [8]

600 samples per class, and can be grouped into 20 macro-categories as can be seen from table 3.1.

Superclass	**Classes**
aquatic mammals	beaver, dolphin, otter,seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

Table 3.1: Macro classes and generic classes for CIFAR100 dataset.

3.3 ImageNET-1k

ImageNet is a huge dataset that is widely used in computer vision research, notably for deep learning applications. Deng et al. [106] published it in 2009, and it already consists of over 14 million annotated images from 20,000 categories grouped according to the WordNet hierarchy. The dataset gained popularity during the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [107], which ran from 2010 to 2017 and contributed significantly to the development of convolutional neural networks (CNNs). The most commonly used subset of ImageNet is 1,000 item categories, which include around 1.2 million training shots, 50,000 validation images, and 100,000 test images. ImageNet has been critical in the development of deep learning models, acting as a baseline for assessing the performance of classification, detection, and segmentation techniques. Many models used for transfer learning have been intensively trained on ImageNet. There are various smaller versions of ImageNet, such as *Imagenette* [108], which only includes a subset of 10 classes from the original 1000. The classes are *tench, English springer, cassette player,*

chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute. Overall, the adopted dataset includes about 1,000 color images per class with a resolution of 160 x 160 pixels. The images are obtained from the ones in the original Imagenet dataset by performing a resizing that preserves the original aspect ratio.

3.4 Pascal-VOC

The Pascal Visual Object Classes (VOC) 2012 dataset is a popular benchmark in computer vision, it was designed for different tasks like object recognition, segmentation, and classification. It was introduced as part of the Pascal VOC challenges [109] and offers annotated photos of 20 different classes as shown in table 3.2.

Macro-class	**Instances**
Person	person
Animal	bird, cat, cow, dog, horse, sheep
Vehicle	aeroplane, bicycle, boat, bus, car, motorbike, train
Indoor	bottle, chair, dining table, potted plant, sofa, tv/monitor

Table 3.2: Classes and instances for Pascal-VOC dataset.

The dataset contains 11,530 photos with 27,450 region-based annotations, making it an excellent resource for testing machine learning models in real-world applications. The annotations contain object bounding boxes, pixel-wise segmentation masks, and object class labels, which allow for the creation and testing of various computer vision techniques. The Pascal VOC 2012 dataset has contributed significantly to the advancement of deep learning-based object recognition and continues to act as a core benchmark for current vision models. Figure 3.4 provides an example of two images and a mask.

Figure 3.4: Examples of image-mask pairs that highlight the presence of heterogeneous classes within the dataset

3.5 MedMNIST

MedMNIST [110] is a dataset conceptually comparable to MNIST. It includes standardized biological data in 2D and 3D images scaled to 28×28 (2D) and $28 \times 28 \times 28$ (3D). The data supplied is wide-ranging, making it ideal for assessing the efficiency and reliability of models built using data from multiple sources but all belonging to the same domain. It is suitable for classification tasks including binary/multi-class, multi-label, and ordinal regression.For this reason, medMNIST includes various types of sub-datasets with a different number of training, validation and test samples for each of the above tasks. A summary of the data amounts and the task for each of them is shown in table 3.3.

MedMNIST2D	Data Modality	Tasks (Classes/Labels)	Samples	Training	Validation	Test
2D applications						
PathMNIST	Colon Pathology	Multi-Class (9)	107,180	89,996	10,004	7,180
ChestMNIST	Chest X-Ray	Multi-Label (14) Binary-Class (2)	112,120	78,468	11,219	22,433
DermaMNIST	Dermatoscope	Multi-Class (7)	10,015	7,007	1,003	2,005
OCTMNIST	Retinal OCT	Multi-Class (4)	109,309	97,477	10,832	1,000
PneumoniaMNIST	Chest X-Ray	Binary-Class (2)	5,856	4,708	524	624
RetinaMNIST	Fundus Camera	Ordinal Regression (5)	1,600	1,080	120	400
BreastMNIST	Breast Ultrasound	Binary-Class (2)	780	546	78	156
BloodMNIST	Blood Cell Microscope	Multi-Class (8)	17,092	11,959	1,712	3,421
TissueMNIST	Kidney Cortex Microscope	Multi-Class (8)	236,386	165,466	23,640	47,280
OrganAMNIST	Abdominal CT	Multi-Class (11)	58,830	34,561	6,491	17,778
OrganCMNIST	Abdominal CT	Multi-Class (11)	23,583	12,975	2,392	8,216
OrganSMNIST	Abdominal CT	Multi-Class (11)	25,211	13,932	2,452	8,827
3D applications						
OrganMNIST3D	Abdominal CT	Multi-Class (11)	1,742	971	161	610
NoduleMNIST3D	Chest CT	Binary-Class (2)	1,633	1,158	165	310
AdrenalMNIST3D	Shape from Abdominal CT	Binary-Class (2)	1,584	1,188	98	298
FractureMNIST3D	Chest CT	Multi-Class (3)	1,370	1,027	103	240
VesselMNIST3D	Shape from Brain MRA	Binary-Class (2)	1,908	1,335	191	382
SynapseMNIST3D	Electron Microscope	Binary-Class (2)	1,759	1,230	177	352

Table 3.3: Schematic representation of all the medMNIST variants available for 2D and 3D applications.

3.6 ISIC

The ISIC 2018 dataset is another well-known benchmark dataset for skin lesion analysis developed as part of the International Skin Imaging Collaboration (ISIC) Challenge [111, 112].The collection includes dermoscopic images annotated by experts for tasks such as lesion classification, segmentation, and feature recognition. This set contains 10,015 high-resolution pictures divided into seven categories, including melanoma, nevus, and basal cell carcinoma. Each image has pixel-wise segmentation masks and diagnostic annotations, making it easier to develop and test deep learning models for automated skin cancer diagnosis. The ISIC 2018 dataset has significantly benefited the progress of

deep learning applications in dermatology by providing a consistent benchmark for assessing the efficacy of computer-aided diagnosis systems. The ISIC 2018 dataset is regarded as the most challenging because the model has to understand specific features of each skin lesion, such as its border and pigmentation. An example of the images contains in ISIC 2018 are reported in Figure 3.5.

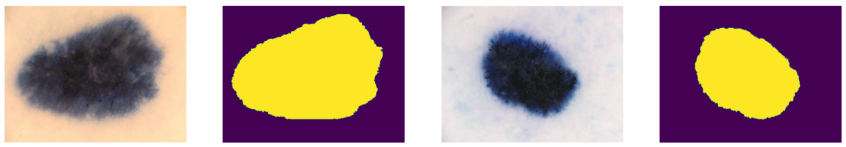


Figure 3.5: An image and segmentation mask from the ISIC18 dataset.

3.7 CERMEP-IDB-MRXFDG

The dataset *CERMEP-IDB-MRXFDG* [9] is a collection of FDG PET, T1 MRI, FLAIR MRI brain scans and CT images, acquired from 37 healthy subjects, with mild cerebral deterioration or in more severe cases also suffering from Alzheimer's. In full respect of privacy policies and sensitive data, the data has been anonymised. The 3D volumes are available in 3 formats: DICOM, co-registered NIFTI and NIFTI normalised in MNI space; in addition, the patients' clinical metadata such as age, gender, diagnosis and neuropsychological scores are provided. Co-registration ensures that the same anatomical regions of the brain are co-located for each individual. In addition to classic co-registration, the images are processed within the MNI space, through an encoding in probability maps of the grey and white matter; in this way, each voxel is assigned a certain probability of belonging to a specific type of brain tissue. The volumes within the dataset were processed through a rigorous pre-processing pipeline to obtain normalised and standardised data, without background objects (e.g. bed, pillow), making them fully usable and ensuring the highest quality, as can be seen from the slices in Figure 3.6.

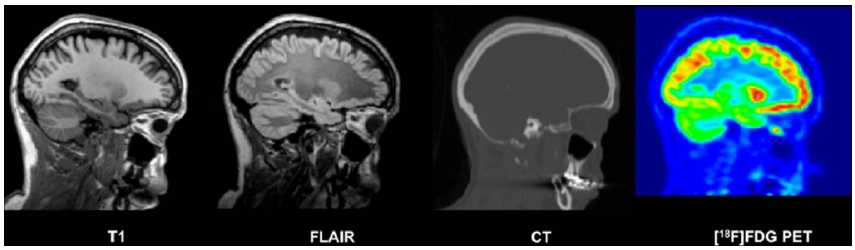


Figure 3.6: Slice t1-w, Flair, CT and PET obtained from the same patient. [9]

3.8 UW-Madison

The UW-Madison dataset (or UW-Madison GI Tract Image Segmentation Dataset) [113] is a public collection of medical images designed for research on image segmentation of the gastrointestinal (GI) tract. It was created by the University of Wisconsin-Madison and released on platforms such as Kaggle for medical segmentation competitions. It contains 38496 MRI images from 85 case study of the gastrointestinal tract and masks to segment the stomach, large intestine and small intestine. The dataset is designed to tackle the task in 2D or 3D. Compared to CERMEP, the quality and standardisation of the images is not the optimal, which makes it a challenging dataset, due the presence of artefacts caused by patient movement during acquisition; furthermore, the distribution of the classes is not homogeneous. It has been used as a benchmark dataset for several segmentation studies [114, 115]. An example of the images it contains is shown in the Figure 3.7.

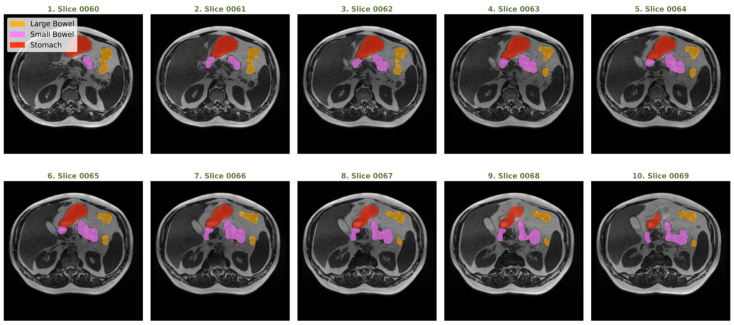


Figure 3.7: An example from the UW-MADISON dataset that shows the MRI image and the masks for stomach, large bowel and small bowel

3.9 Brain Tumor

The Kaggle Brain Tumor consisting of 3064 T1-weighted contrast-inhanced images from 233 patients with three kind of brain tumor:

- meningioma (708 slices)
- glioma (1426 slices)
- pituitary tumor (930 slices)

This dataset is ideal for classication or segmentation and provides three different brain views depending on the illness, as seen in Figure 3.8.

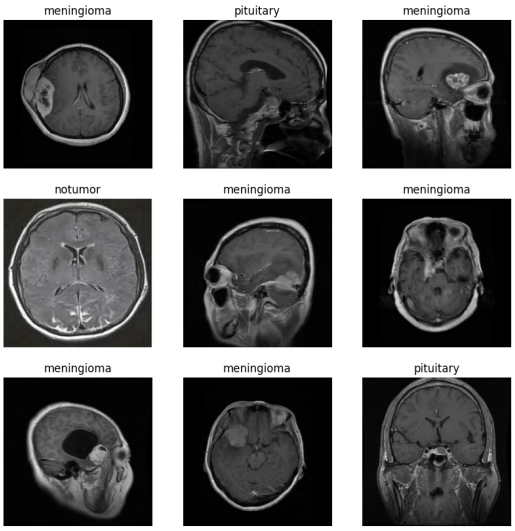


Figure 3.8: Several examples from the MRI Brain Tumor dataset that highlight the various diseases

3.10 Few-Shot Segmentation Dataset

The Few-Shot Segmentation Dataset [116], frequently referred to as FSS-100, was specifically defined for the purpose of few-shot segmentation. Because it comprises of 100 classes with just ten samples for each, it is structured in such a way that it emphasizes the network's generalization skills. The end result is a collection of annotated photographs with a variety of subjects; in fact, the classes vary from common things (e.g., 'dog', 'car') to unusual categories (e.g.,'sunset on Mars', 'electron microscope'), putting the generalization capacity to the test across diverse domains. It is separated into 760 training courses and 240 validation classes; as needed by the few shot, the two sets of classes in each set must be dissociated.

3.11 DeepGlobe

DeepGlobe [117] is a dataset designed for semantic segmentation of high-resolution satellite images. It initially showed up at the CVPR Conference as the DeepGlobe Challenge.

It includes pixel-by-pixel annotated masks for seven different classes. Specifically, the classes are:

- built-up
- agriculture

- rangeland/forest
- barren land
- water
- cloud/cloud shadow
- ignore label

The train, val, and test datasets contain 1,146 satellite photos with a resolution of 2448×2448 pixels.

Chapter 4

Learn&Drop

Although research is improving, the aim is still to seek ever-better performance with the objective of equalling and surpassing human capabilities. In this environment, when models are constantly pushed to the limit, the resources necessary for the training and/or inference of these models, which require enormous computing and energy resources, are often neglected or marginalised. The underlying issue now seems to be a lack of resources required by the models, rather than a lack of performance per se. Research is moving away from the development of increasingly complex and large models with almost prohibitive computational costs, and is moving towards more efficient models that reduce time, computational costs and energy without compromising the accuracy of the results. [118]. In this chapter, the first study will be conducted to understand how classification performance depends on the number of parameters present in the neural architecture. In this context, a new technique is developed to detect and remove network components that are no longer required during training, resulting in increasing efficiency while maintaining performance. The "Learn&Drop" method proposes to evaluate the gradient magnitude of each layer to identify which of these have finished the learning phase and can therefore be removed.

Typically, optimisation can be performed at different stages of the life of a neural network and according to different approaches:

- **methods for inference efficiency**: this category includes methods that compress the network during the training phase to use a more compact model during inference [119, 120, 121, 122, 123];
- **methods for training efficiency**: fall into this category methods that improve the efficiency of model training only, for example by freezing some layers during training but continuing to use the entire model during

the inference phase [124, 125, 126].

Intuitively, one would expect that focusing on the efficiency of the inference step would be far more advantageous in the long term because a model is trained once and then utilized many times. However, this is not always the case because in some applications, such as tracking, the model must constantly train itself to quickly respond to changes in the target over time. In addition, recommendation systems and large language models must be retrained on a regular basis, which can take several weeks. The presented work [10] aims to increase the efficiency of CNN training by gradually compressing the model by removing convolutional layers based on a criteria. The maps deleted during each iteration are retained and used as new input to the remaining layers, resulting in a reduction in training time.

In fact, training neural networks consists of two fundamental phases: forward pass and backpropagation. In the first, the data is processed sequentially through the various layers, in which a linear transformation is applied (using weights) and a non-linear activation function is calculated at the output. The second phase, described in the literature [127], calculates gradient of the loss function in relation to the network weights, allowing it to be updated through optimisation methods.

A notable pattern observed during the training of convolutional neural networks (CNNs) is the tendency of earlier layers to converge before deeper ones. This sequential convergence behaviour motivated the procedure of a progressive pruning, in which layers that have already stabilised are removed incrementally. To preserve the consistency of the learned representations, the feature maps produced by the removed layers are cached and reused as fixed inputs for the remaining architecture. This method offers two main advantages: (1) a reduction in computational complexity during training, as the weights of the pruned layers are no longer subject to updates; and (2) consistency in the data representation, guaranteed by the reuse of the pre-computed feature maps, rather than their re-processing at each step.

4.1 Dropping Layers for Training Efficiency

In this section “Learn&Drop" method is described in Algorithm 1 and its main steps are:

- Compute the “Layer Importance metric" based on the gradient of the layer’s weights;

- Apply the “Fast Learning" algorithm, the core of our method. The algorithm consists of steps to: (1) select the layers to be dropped, (2) split the network into a “tail", composed of the dropped layers, and a “head", composed of the layers to still train, (3) compute and store output feature maps from the tail, (4) train the head by using output feature maps from the tail.

Algorithm 1 Fast Learning by layer dropping

Require:

$model$, a model of L layers with randomly initialized parameters;
$e_1 > 0$, number of warm-up epochs;
$e_2 > 0$, number of training epochs;
L_0, number of dense layers + 1;

Ensure:

Trained $model$;

Initialize models $head$ and $tail$
Initialize $Data$ with the training images
$save\_features$ = False.
Train $model$ for e_1 epochs on $Data$
$L = model$.layers.length()
$head$.layers = $model$.layers$[0 : L - 1]$
$tail$.layers = []
$features\_maps$ = []
Initialize P'_l with l in $head$
for $k = 0$ **to** e_2 **do**
    if $save\_features$ **then**
        $save\_features$ = False
        $features\_maps = tail(Data)$
        Train $head$ on $features\_maps$
        Update $model$ weights based on $head$
        store $features\_maps$ to memory
    else
        if $features\_maps! = []$ **then**
            Initialize $Data$ with $features\_map$
            $features\_maps$ = []
        end if
        Train $head$ for 1 epoch on $Data$
        Update $model$ weights based on $head$
        if $L > L_0$ **then**
            for $\forall$ conv. layer l in $head$ **do**
                update P'_l (Eqs. 4.2& 4.3)
            end for
            Find n^* (Eq. 1) for layer dropping
            Estimate median values M_c and M_d
            if $n^* > 1 \wedge M_c \geq M_d$ **then**
                $save\_features$ = True
                $tail$.layers = $head$.layers$[0 : n^*]$
                $head$.layers.pop$(0 : n^*)$
                $L = head$.layers.length()
            end if
        end if
    end if
    validate $model$ on validation set
end for

4.2 Layer importance

The gradients of the loss function ∇g become the foundation for our layer removal strategy. In general, gradient values can be regarded as the rate at which the weights change. On the other hand, the sign of the partial derivatives reflects the inverse direction in which the weights must change to reach a minimum. It is well understood in theory that the gradient, as a vector, has a magnitude expressing the amplitude of the shift and a direction indicating the direction of the shift. The gradient of the loss function can be thought of as the pace at which the weights in the model change, with the sign representing the shift required for each signular parameter to reach the minimum.

Given a neural network with layers $L = \{l_0, l_1, ..., l_L\}$, the average absolute partial derivative value $\overline{g_l^{(k)}}$ corresponding to the weights of the l-th layer is calculated as:

$$\overline{g_l^{(k)}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \mid g_{lij}^{(k)} \mid \tag{4.1}$$

where N is the number of weights in layer l, M is the number of iterations in epoch k, and $g_{lij}^{(k)}$ is the partial derivative of the loss function with respect to the i-th weight in layer l at the j-th iteration in epoch k.

Figure 4.1 shows two graphs representing the average absolute partial derivative value $\overline{g_l^{(k)}}$ of each convolutional layer in the VGG11 and ResNet18 models. Interestingly, the beginning layers of both networks had larger average partial derivative values than the final part of the models. This graph indicates that during training, the network does not train equally, but rather the early layers change their weights faster and with higher gradient values to adapt to the task at hand. The described behavior is much more noticeable in the initial phases of training.

Thus, partial derivative values can help understanding if the weights of a layer are still changing or not. However, using directly the average absolute partial derivative values could be misleading since the weights may have small magnitude but still change.

Hence, we decided to adopt the metric proposed in [124], and define for the l-th layer the score:

$$P_l^{(k)} = 1 - \frac{\sum_{i=1}^{N} \mid \sum_{j=1}^{M} g_{lij}^{(k)} \mid}{\sum_{i=1}^{N} \sum_{j=1}^{M} \mid g_{lij}^{(k)} \mid} \tag{4.2}$$

with $0 \leq P_l^{(k)} \leq 1$, where $P_l^{(k)}$ measures the degree of changes of the weights in layer l at the k-th epoch. $P_l^{(k)}$ will be 1 if the partial derivatives cancel each

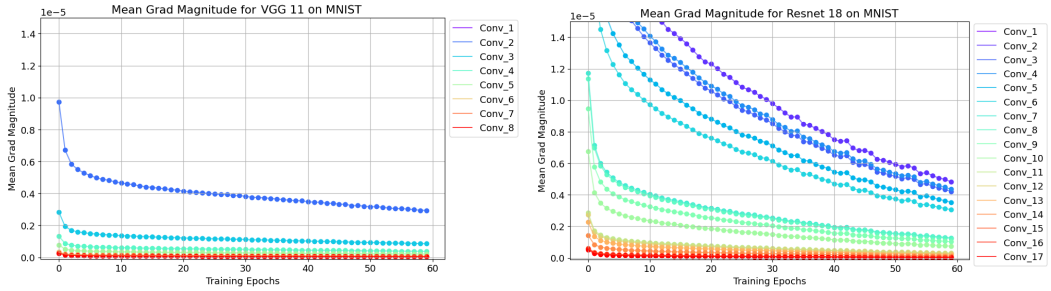


Figure 4.1: The graphs represent the average absolute partial derivative (AAPD) value of each convolutional layer in the VGG-11 (left) and ResNet-18 (right) on the MNIST dataset (AAPD on the y-axis, epochs on the x-axis). Each curve represents a different layer (purple for layers close to the input, red for those close to the output). Weights in the first layers undergo higher changes than those in the layers closest to the output. AAPD help measuring if a layer is still learning or not. [10]

other across the M iterations. In such a case, within the epoch the layer weights do not change much and, intuitively, the layer has stopped to learn. $P_l^{(k)}$ will tend to 0 if most of the partial derivatives are in the same direction across iterations. In this case, layer weights are changing during the epoch. Thus, the layer is learning something about the problem to solve. We note here that the normalization factors make the scores comparable across the layers despite the different magnitude of the weight's partial derivatives. The score $P_l^{(k)}$ indicates the relevance of the layer during training. Layers with a score approaching 0 must be trained. Layers with a score close to 1 are unlikely to learn much and should be deleted to speed up model training. In contrast to [124], which utilizes this score to lock the layer and cease back-propagation computation until the l-th convolutional layer, our technique uses P_l to drop the l-th convolutional layer. The feature maps generated by the final abandoned layer serve as input for the remaining model.

4.3 Improving training efficiency

In this approach, the removal of layers from the model to improve training efficiency must be done in sequential order. At the k-th epoch, the layers to be removed are selected according to the importance score $P_l^{(k)}$.

This section describes the stages that constitute into the quick training algorithm:

1. At the end of epoch k, the metric $P_l^{(k)}$ is calculated for each layer l. The

score values are then standardized:

$$P'_l = \frac{P_l - \overline{P}}{\sigma_p} \tag{4.3}$$

where $\overline{P}$ and σ_p represent the average score over the layers and the standard deviation respectively. For simplicity, we omitted the apex k referring to the epoch.

At each iteration, the weights in *head* are used to update the equivalents in *model*, which remains in its initial configuration. The $save\_features$ flag specifies whether the *tail* model should be used to estimate feature maps with the removed layers. *Data* initially holds images for model training; after iterations and layer removal, they are replaced by the associated feature maps. Each iteration, the relevance of layer P'_l is computed for each layer of model *head*, as described in equations 4.2 and 4.3.

The problem of selecting the subsequent layers to drop starting from the first layer turns into the problem of finding the sub-vector of maximum sum starting from the first element of an array. In our case, the array represents the list of scores P'_l with $l \in L$.

Let us assume that the current layers in the model are $L = \{l_z, l_{z+1}, \ldots, l_L\}$. Candidate layers to drop are $l_z ... l_{n^*}$ with n^* computed as:

$$n^* = \min_t \{t \in [z, \ldots, L-1] : P'_{l_t} > 0 \wedge P'_{l_{t+1}} < 0\}. \tag{4.4}$$

2. To determine the best moment to drop layers, the candidates $l_z ... l_{n^*}$ and the median M_c of their scores were initially calculated. This value is compared to M_d, which is the median calculated by P'_{l_t} with $l_t \in l_0 ... l_{z-1}$, i.e., the drop values from the preceding iterations. If $M_c \geq M_d$, the removal operation will be executed. This prevents executing the action too early, which can affect the network's overall performance. After identifying the layers to be removed sequentially, the model could be divided into two parts: the "tail" (up to l_{n^*}) and the "head" (from l_{n^*+1} to the output).

3. At the next epoch $k+1$, the queue is used for feature map extraction,

which will be saved on a physical memory such as a disk. The latter will be provided as input to the head to continue its training.

4. In epoch $k + 2$, the stored feature maps are retrieved from the memory and used to train the head.

The proposed technique varies from that of [124]. The latter strategy does not delete high-scoring layers from the network, but it does not train their weights. However, during forward propagation, data from all levels, including those with no updated weights, must be analyzed at each iteration. This constraint is overcome through a strategy that removes the layers in order, starting from the first. The effectiveness of this strategy in drastically reducing the computational cost of the training process has been empirically demonstrated. Furthermore, in the suggested approach, the choice of which layer to cut is completely automatic and depends on the score $P_l^{(k)}$, as discussed in detail above.

4.4 Fast-Training Algorithm

Figure 4.2 depicts a comprehensive and compact representation of the proposed algorithm. To change the weight distribution of the *model*, a few warming e_1 epochs are done. At this point, the weights of *model* are reproduced in *head*, while *tail* is still empty. The model to be trained is *head*, and its complementary *tail* stores the dropped layers required to compute the feature maps that will be fed into *head*. At each iteration, the weights in *head* are used to update the equivalents in *model*, which remains in its initial configuration. The $save\_features$ flag specifies whether the *tail* model should be used to estimate feature maps with the removed layers. *Data* initially holds images for model training; after iterations and layer removal, they are replaced by the associated feature maps. Each iteration, the relevance of layer P'_l is computed for each layer of model *head*, as described in equations 4.2 and 4.3.

Then, n^* is calculated based on Eq. 1. The layer is dropped if a sub-sequence with maximum sum of scores P'_l is found starting from the first layer of the *head* that includes at least one layer, and the median value of the scores in the sub-sequence found is greater than the median score of the previously dropped layers. The scores of the discarded layer are not recomputed each time, but stored during the training process and kept updated until the layer is discarded. The n^* index is also used to further compress the *head* model. In

particular, the *tail* model stores the dropped layers, i.e. the first n^* layers of the *head* model. These same layers are taken from the *head* model, resulting in a reduced model.

The method is iterated until no more convolutional and dense layers exist, after which they are trained for the maximum number of epochs e_2 and/or patience. The whole model is evaluated on a validation set, demonstrating that the approach does not degrade performance. To demonstrate that learning rate does not effect training, all tests were performed with the SGD (Stochastic Gradient Descending) optimizer to keep the rate fixed. Furthermore, no Early Stopping techniques were utilized to compare the various tactics at the same era.

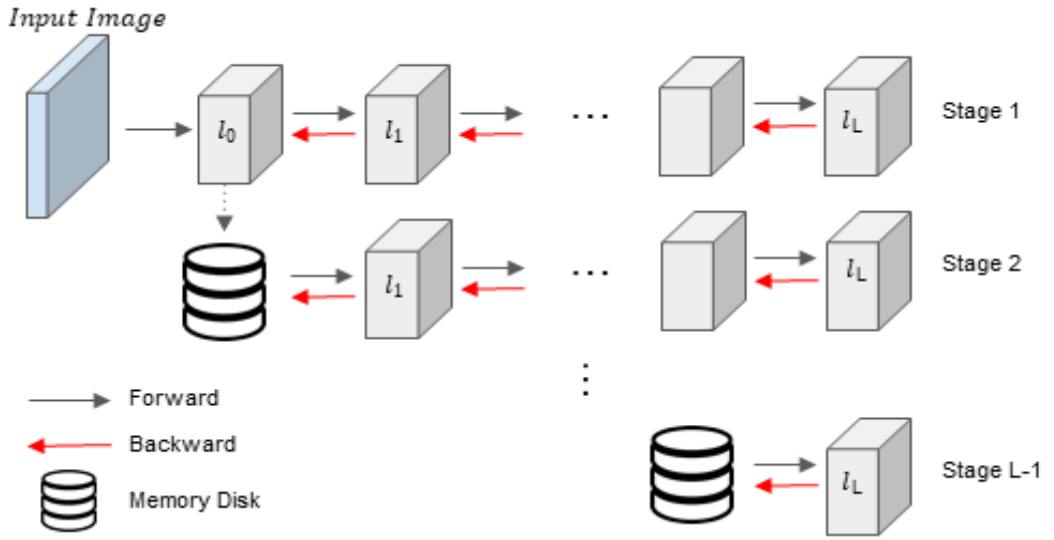


Figure 4.2: The image shows how the process flows through the sequence of stages. At the first stage the input is the original image, subsequently the features maps stored from dropped layers are used as input for the remaining layers. [10]

4.5 Experimental Results

The following section will present the results of the proposed method, going to evaluate different architectures and different benchmark datasets. The results of the experiments are shown in tables 4.1 and 4.2. The results are presented with:

- **Network**, indicates the neural architecture used.
- **Dataset**, specifies the dataset used for training and testing the *Network*.

- **SGD**, **Freezing**, **Dropping** indicates which strategy is used for training. *SGD* is classical training without any strategy. *Freezing* is the approach used in [124], while *Dropping* is the proposed method.

- **T**, refers to the total training duration and its value is expressed in minutes, including warm-up epochs.

- **A**, indicates the accuracy value on the test, it is reported as the percentage of correct outputs.

- ΔT, serves to highlight the difference of the specific method, compared with the basic SGD approach. The method equation used to calculate this difference is expressed as:

$$\Delta T = \frac{T_{SGD} - T}{T_{SGD}} \cdot 100. \tag{4.5}$$

In this implementation, feature maps produced by the dropped layers are stored on disk directly as PyTorch tensor using the "Pickle" Python package [128], that implements binary protocols for serializing and de-serializing a Python object. We experimentally noted that using Pickle is faster than writing and reading files on disk with the Numpy package [129] and PyTorch [130].

4.5.1 Neural Architectures

To assess this idea two neural network architectures widely adopted in the computer vision field are considered. VGG (Visual Geometry Group) is a convolutional neural network introduced in [131]. The VGG architecture is characterized by its depth and the use of small convolutional filters. It consists of a sequence of convolutional layers, followed by a sequence of fully-connected layers. There are several configurations of the VGG architecture. The smaller version is the VGG-11 with only 11 layers. The VGG-16 has 16 layers and the VGG-19 has 19 layers. As the number of layers in VGG increases, so do the training time and memory requirements.

The presence of the batch normalization layer in the model determines the order of the curves reflecting the layer scores P'_l over the epochs, according to the tests carried out using VGG. In fact, as can be seen in the Figure 4.3, the order of the scores of the VGG+BN layers is the inverse of the order of the scores of the VGG without BN. This is because batch normalisation speeds up learning in neural networks by normalising the inputs of each layer, which reduces the shift of the internal covariate. This makes the optimisation more

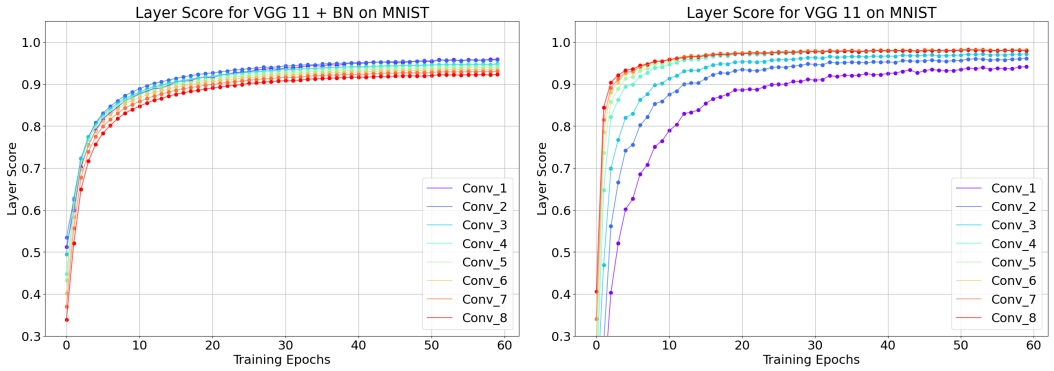


Figure 4.3: The plots show the scores $P_l^{(k)}$ on the MNIST dataset for a VGG-11 trained with (on the left) and without batch normalization (on the right). Bath normalization reverses the order of the score curves and reduces the internal covariate shift making the optimization more stable and quick. As an effect, layers are learned sequentially from input to output. [10]

stable and allows the network to learn more quickly and with greater accuracy. So, based on experiments, to use this technique with a VGG it is advisable to include the batch normalisation layer (one after each convolutional layer).

ResNet (Residual Network) is a convolutional neural network introduced in [132]. ResNet is characterized by the use of residual blocks, which help to alleviate the vanishing gradient problem and allow for the creation of much deeper neural networks. The original ResNet architecture has several configurations, including ResNet-18, a relatively small version of the ResNet architecture with only 18 layers. ResNet-50, ResNet-101, and ResNet-152 are much deeper versions of the ResNet architecture with 50, 101, and 152 layers, respectively.

Note that in the case of ResNet, in order to maintain the original behaviour as shown in the Figure 4.4, it is necessary to save not only the feature maps, but also the output of the skip connections.

4.5.2 Dataset and Hyper-parameters

To evaluate our algorithm we use three popular classification datasets: MNIST, CIFAR-10, and Imagenette. The MNIST and CIFAR-10 datasets were selected for optimal overlap and comparison with [124]. The Imagenette dataset, was chosen since it is known to be stressful and thus difficult for the algorithm.

The learning rates are set to common values used in the literature, while the number of epochs and warm-up are selected empirically. On the MNIST and CIFAR-10 datasets, the total number of epochs (including model warm-up) is set to 60. On the MNIST dataset, the learning rate is fixed to 0.001. The

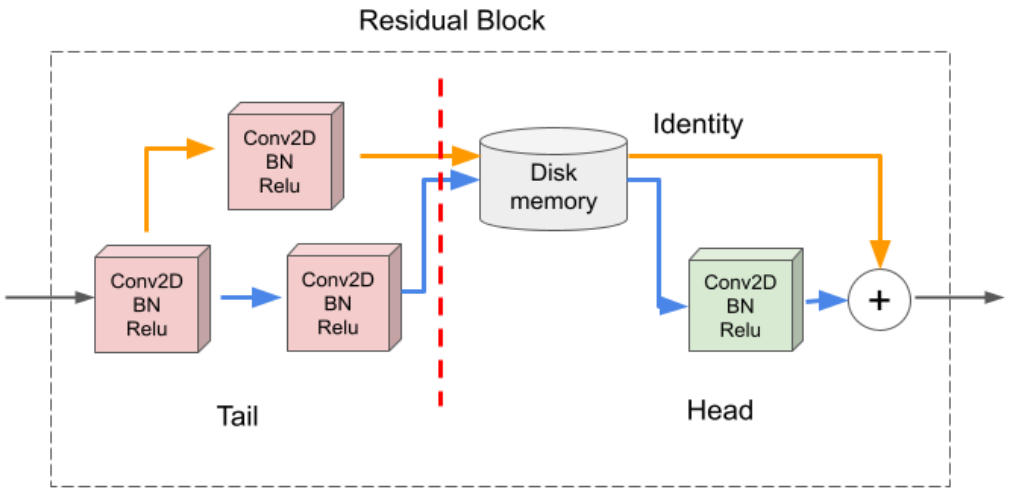


Figure 4.4: The image shows how dropping takes place in the ResNet at any residual block. The feature maps saved to the memory come from the layers in red inside the residual block and on the skip connection. The layers on the left of the vertical dotted line are dropped and belong to the tail, the ones on the right belong to the head model and are trained based on the stored feature maps. [10]

warm-up epochs are 5. On the CIFAR-10 dataset, the learning rate is fixed to 0.1 and scaled x10 after 20 epochs. The warm-up epochs are 10. Finally, on the Imagenette, the number of epochs is set to 150 and the learning rate is fixed to 0.01 and scaled x10 after 50 epochs. The warm-up epochs are 25.

Table 4.1: Fast Training of VGG architectures. SGD refers to the standard training strategy of the entire model. Freezing refers to excluding the parameters of some layers from the training without removing the layers from the model. Dropping is our method where layers are deleted from the trained model. T is the training time in minutes. A is the test accuracy value. ΔT is the percentage of reduced training time with respect to the time of SGD.

		SGD		Freezing			Dropping (Ours)		
Network	**Dataset**	**T (min)**	**A (%)**	**T (min)**	**A (%)**	ΔT (%)	**T (min)**	**A (%)**	ΔT (%)
VGG-11	MNIST	20.83	**98.64**	19.58	98.25	6.00	**8.74**	98.25	**58.04**
VGG-11	CIFAR-10	23.83	**92.02**	23.54	91.72	1.21	**8.21**	91.72	**65.54**
VGG-11	Imagenette	61.01	**75.33**	59.33	74.08	2.75	**18.32**	74.08	**69.97**
VGG-16	MNIST	22.54	**98.85**	21.45	98.26	4,24	**9.01**	98.26	**60.03**
VGG-16	CIFAR-10	26.54	**93.12**	24.94	92.84	6.03	**9.56**	92.84	**63.98**
VGG-16	Imagenette	74.73	**78.76**	71.21	77.83	4.71	**25.23**	77.83	**66.24**
VGG-19	MNIST	23.02	**98.52**	22.68	96.22	1.48	**9.45**	96.22	**58.95**
VGG-19	CIFAR-10	27.02	**93.10**	25.78	91.71	4.59	**11.53**	91.71	**57.33**
VGG-19	Imagenette	110.35	**80.32**	105.34	78.13	4.54	**37.76**	78.13	**65.78**

4.5.3 Results and Comparison

The results obtained with the VGG family, then with the various depths (11, 16, 19) and batch normalization, are shown in Table 4.1. Clearly, the effect of freezing or removing layers to increase training efficiency is marginal, with minimal differences in the range from $0.26(for VGG-16 trained on CIFAR-10) to 2.38$ (for VGG-19 trained on MNIST). The differences increase with different depths, however a minimal variation in accuracy is associated with a reduction in training time.

According to the table, for the VGG models, the training time reduction with the freeze layer technique ranges between 0.40% (for VGG-16 on the MNIST dataset) and 6.03% (for VGG-16 on the CIFAR-10 dataset), whereas the training time reduction with our layer reduction technique ranges between 58.04% (for VGG-11 on the MNIST dataset) and 69.97% (for VGG-11 on the Imagenette dataset). Compared to the average of 3.52% for the frozen layer method,this strategy produces yields of 62.87%. The ResNet family was treated using the same technique and concerns (18, 50, 101, 152). Table 4.2 shows that accuracy values vary between techniques and range from 0.4% (for ResNet-18 on the MNIST dataset) to 3.2% (for ResNet-152 on the Imagenette dataset). As expected, extending the network's complexity to a higher depth improves performance. The layer freezing strategy reduces training time by between $1.16\%(for ResNet-50 on the MNIST dataset) and 7.8\%$ (for ResNet-

Table 4.2: Fast Training of ResNet architectures. SGD refers to the standard training strategy of the entire model. Freezing refers to excluding the parameters of some layers from the training without removing the layers from the model. Dropping is our method where layers are deleted from the trained model. T is the training time in minutes. A is the test accuracy value. ΔT is the percentage of reduced training time with respect to the time of SGD.

		SGD		Freezing			Dropping (Ours)		
Network	**Dataset**	**T (min)**	**A (%)**	**T (min)**	**A (%)**	ΔT **(%)**	**T (min)**	**A (%)**	ΔT **(%)**
ResNet-18	MNIST	23.67	**98.2**	23.10	97.78	2.41	**8.64**	97.78	**63.50**
ResNet-18	CIFAR-10	27.67	**92.25**	25.97	91.82	6.14	**11.90**	91.82	**56.99**
ResNet-18	Imagenette	253.07	**80.12**	242.32	79.07	4.25	**83.78**	79.07	**66.89**
ResNet-50	MNIST	35.43	**98.75**	35.02	96.85	1.16	**11.23**	96.85	**68.30**
ResNet-50	CIFAR-10	38.43	**94.40**	35.40	92.05	7.88	**13.05**	92.05	**66.04**
ResNet-50	Imagenette	336.00	**82.78**	315.34	80.34	6.15	**86.28**	80.34	**74.32**
ResNet-101	MNIST	53.12	**97.81**	51.64	95.45	2.79	**18.56**	95.45	**65.06**
ResNet-101	CIFAR-10	56.12	**93.98**	52.03	91.26	7.29	**19.53**	91.26	**65.20**
ResNet-101	Imagenette	402.34	**82.23**	380.23	80.75	5.29	**120.44**	80.75	**70.06**
ResNet-152	MNIST	70.76	**97.43**	65.30	95.12	7.72	**23.34**	95.12	**67.01**
ResNet-152	CIFAR-10	74.76	**93.45**	68.93	91.03	7.80	**25.75**	91.03	**65.56**
ResNet-152	Imagenette	540.76	**82.65**	504.72	79.45	6.66	**180.34**	79.45	**66.65**

152 on the CIFAR-10 dataset). Training time is reduced by about 56.99% for ResNet-18 on the CIFAR-10 dataset and 74.32% for ResNet-50 on the Imagenette dataset On average, freezing the layers reduces training time by 5.46%. However, the strategy reduces training time by approximately 66.30%. The results support the hypothesis, since the loss of accuracy is equivalent to that obtained with other approaches, despite a time saving of more than half.

4.6 Training Time and Parameter Reduction

Section 4.5 discusses how the proposed strategy improved model accuracy and reduced training time on our machine. In order to analyze and explain

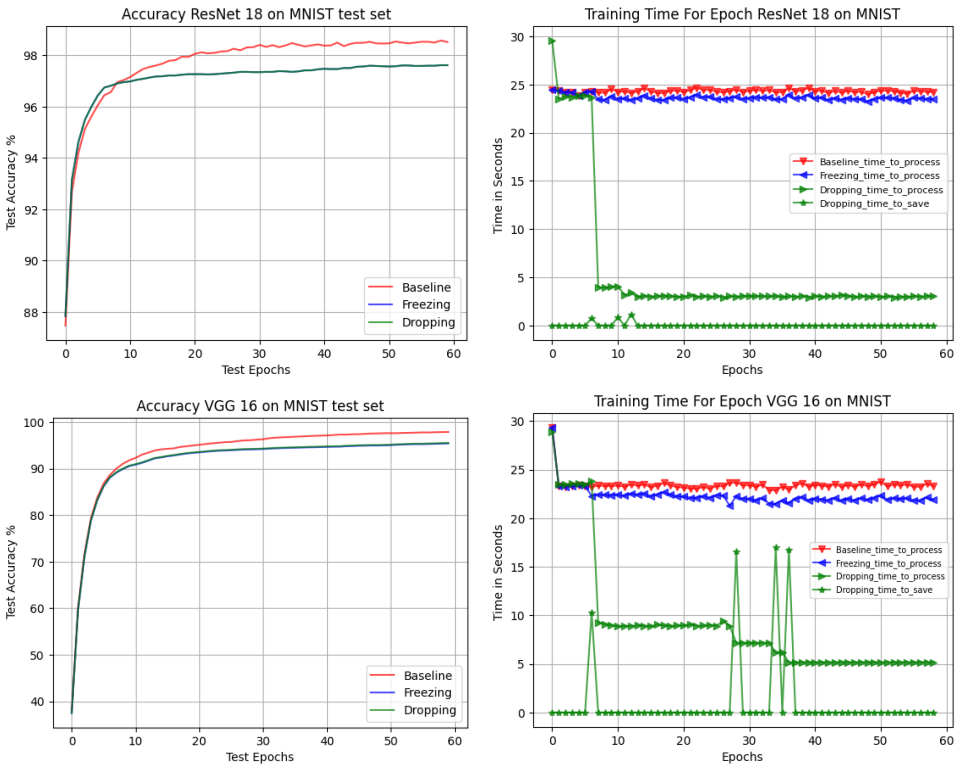


Figure 4.5: The plots on the left show the test accuracy values of a ResNet-18 (top) and VGG-16 (bottom) trained on the MNIST dataset with different strategies: SGD (red curves), layer freezing (blue curves), and layer dropping (green curves). The experiments were repeated 10 times with different starting weights and data randomization. Freezing and dropping layers achieve nearly equivalent test accuracy values, and the values are slightly lower than those achieved by training the entire model. On the right, the plots show training time per epoch. Starred curves show the time required to store the feature maps to disk, while the other curves show the training time which decreases over the epochs due to the lower cost of forward propagation in our method. [10]

the usefulness of this method, this section focuses on the impact that it has on the number of parameters and operations performed during forward propagation during training. In reality, this methodology reduces both the number of weights for which partial derivatives must be calculated during gradient propagation and the number of operations executed during forward propagation. One potentially critical aspect of the proposed method is the phase in which the features map are stored on the physical storage medium. Given that certain models generate embeddings with significantly more dimensionality than the original input during the characteristic extraction phase. These considerations could indicate that these disk I/O operations introduce an overload, reducing the temporal advantage derived from removal. However, actual evidence suggests that layer removal never occurs layer by layer, but rather that numerous sequential layers are typically removed concurrently. Figure 4.5 displays (left) the accuracy scores of the ResNet-18 (top row) and VGG-16 (bottom row) tests on the MNIST dataset over epochs using SGD optimiser, comparing the layer freezing method with the same strategy.

The graphic 4.5 outlines all the considerations mentioned thus far. In fact, the left side displays a comparative graph of the accuracies obtained using the various techniques, while the right side displays the times required to complete the different trainings. The overhead values outlined above are shown individually to emphasize the importance of these processes during the epochs in which dropping occurs. The duration of this procedure is significantly dependent on the data, specifically the dimensionality of the maps to be saved. It should be noted that the training time includes computing gradients, changing layer weights, loading training data, and executing forward propagation. To summarize, the graph on the right enhances the effectiveness of our method, which becomes faster and faster as the epochs progress, as opposed to the other approaches that obviously remain constant over time.

The left side of Figure 4.6 shows the number of parameters for each epoch for ResNet-101 (top) and VGG-16 (bottom). The red curves reflect the number of model parameters when using the basic and layer freezing procedures, while the green curves represent the number of parameters when using dropping. Due to the fact that the model is compressed during training, the number of parameters related to the layers removal epochs is considerably reduced. This reduction in parameters is consistent with the reduction of the MMAC during direct propagation, as illustrated in the graphs on the right. While MMAC remains constant in the basic and layer freezing procedures, it continues to

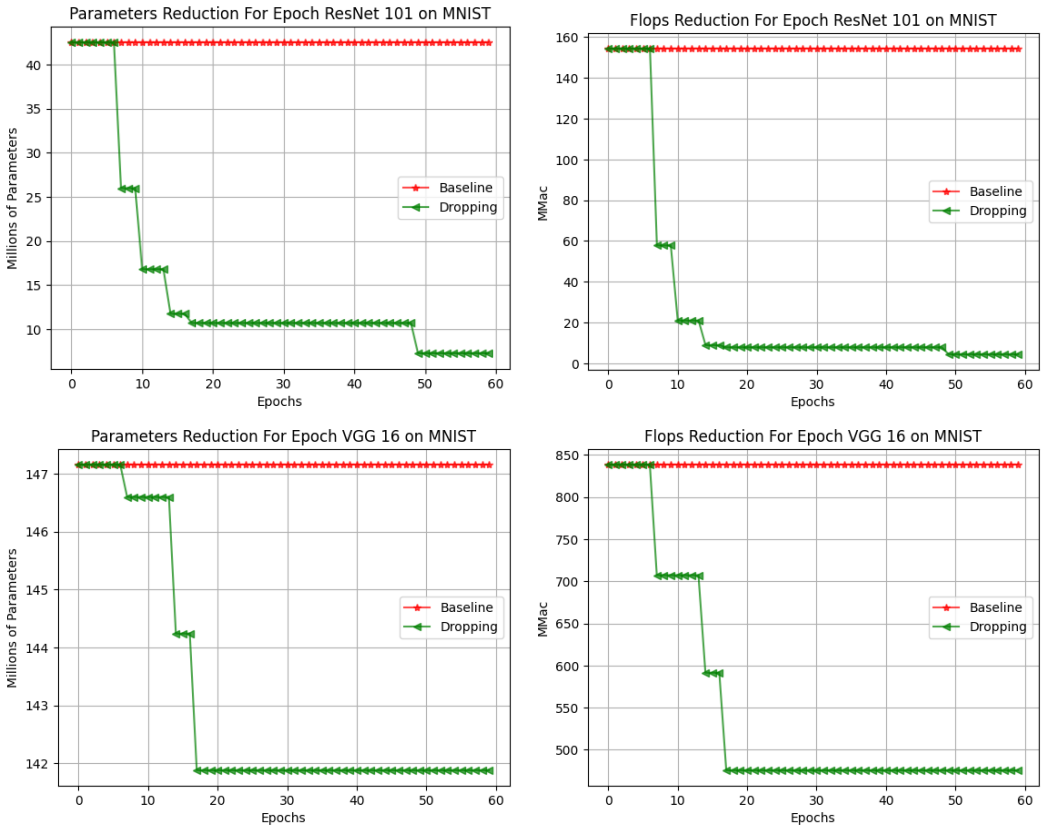


Figure 4.6: Left plots show the number of network parameters in each epoch for the ResNet-101 (top) and VGG-16 (bottom). The number of parameters remains constant when training or freezing the layers (red curves); it decreases with our approach (green curves). This parameter reduction is correlated with the MMAC (Mega Multiply-Accumulate) reduction, shown in the plots on the right, because it results in the less number of operations during forward propagation. [10]

decrease due to the reduction in the number of factors. To make an objective and measurable assessment, the FLOPs (floating point operations per second) required to complete the different approaches being compared were calculated. In particular, the MMAC was calculated, a FLOPs metric that counts the number of matrix multiplications and accumulations (MAC) that a neural network performs in one second. The metric is expressed in millions (mega) of operations and is useful for evaluating the computational complexity of a neural network and comparing the performance of different architectures. Figure 4.6 shows that the baseline (red curve) remains constant over time both in terms of FLOPs and parameters, while *Learn&Drop* (green curve) benefits from the elimination effect that reduces the size of the model and consequently results in a respective decrease in computational cost. Table 4.3 shows the FLOP values compared with the classic SGD approach. The last column calculates

the difference ΔFLOP between the two approaches. The gain is even more evident with deeper models such as ResNet-101 and ResNet-152 with greater variations of 80%.

Table 4.3: FLOPs reduction across architectures. SGD refers to the standard training strategy of the entire model. Dropping is our method where layers are deleted from the trained model. FLOPs are measured during the forward propagation. ΔFLOPs is the percentage of reduced FLOPs with respect to SGD. Our approach reduces the FLOPs of all architectures, especially of the largest ones.

	SGD	Dropping (Ours)	
Network	**FLOPs**	**FLOPs**	**ΔFLOPs (%)**
VGG-11	31,203.60	**25,847.02**	17.17
VGG-16	50,311.19	**33,049.44**	34.31
VGG-19	66,000.00	**44,079.31**	33.21
ResNet-18	1,987.80	**640.53**	67.78
ResNet-50	4,704.59	**2,193.45**	53.38
ResNet-101	9,262.2	**1,667.27**	82.00
ResNet-152	13,823.39	**2,247.46**	83.74

Chapter 5

Few-Parameters Architectures

According to the evidence obtained and discussed in Chapter 4, the research activity has been oriented towards the development of architectures with a low number of parameters capable of performing in a way comparable to the state of the art in classification and segmentation tasks. Even more ambitious was the desire to try to build an agnostic segmenter whose few-parameter segmentation was conditioned only by the features extracted to perform a classification.

In this chapter we will present and discuss in detail the Few-Parameter architecture, FPA. This is a Depth Separable Convolutional Neural Architecture, DSCNN, which uses 9 convolutional layers followed by a Global Average Pooling (GAP) after which an output activation function is applied. This architecture was designed to maximise the extraction of classification features, the main task for which it is trained, and also to fulfil a region proposal task. The use of GAP is borrowed from the method known as Network In Network (NIN) architecture and can be used instead of the typical dense classification layer in a similar way to NIN block, with a drastic reduction in the overall parameters. The Region Of Interest (ROI) proposal is obtained using the winning class activation map extrapolated from the last convolutional layer. FPA was trained on different reference data sets in the classification domain; in addition, datasets created for the segmentation of medical images were added. The experiments performed are very promising, reaching state-of-the-art performance in classification. To evaluate the ability to identify ROIs, a metric was created for the segmentation task.

5.1 Theoretical Background

As discussed in Chapter 3, the difficulty in retrieving and sharing data is one of the major problems of the lack of data, a problem amplified in the medical field by privacy issues due to the personal information of patients that prevents its free diffusion.

This, as already mentioned in the previous sections, is reflected in the scientific community in a severe difficulty in training neural architectures for classification and segmentation tasks in the biomedical sector. To overcome this problem, the idea behind the development of this architecture is dual. Firstly, the aim is to build a DNN with a reduced number of parameters, to optimise the computational costs and the resources required for its training, and in addition to obtain a model that is able to perform several tasks simultaneously. This second aim is motivated precisely by the lack of annotated data in the biomedical domain. The FPA architecture proposes to accomplish this task by making two substantial changes: i) replacing the “mlpconv" that Lin et al. propose in their NiN-block with classic Depth Separable Convolution (in Figure 5.1).

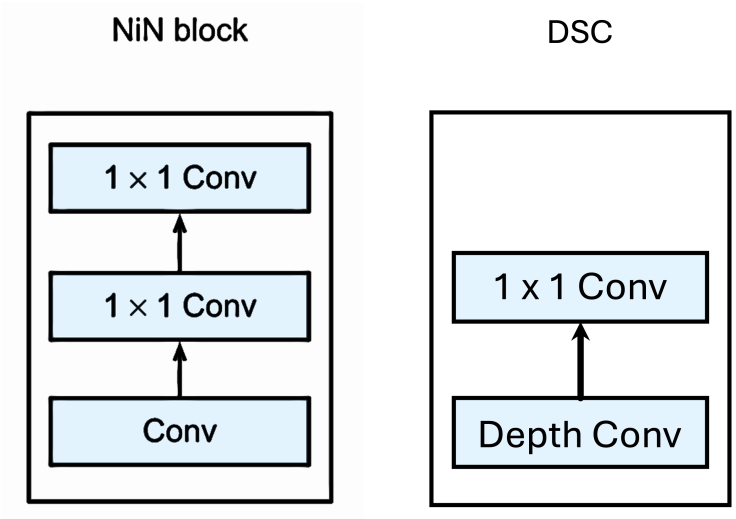


Figure 5.1: Differences in structure between the NiN [11] and DSC blocks.

ii) replacing the output of the MLP, commonly used to define the classification block, with a Global Average Pooling (GAP) layer, as described in the NIN architecture [11].

5.2 FPA architecture

The FPA architecture proposed in this work is composed of nine blocks for feature extraction. Each block begins with a Deep Separable Convolution (DSC) followed by Parametric Rectified Linear Units (PReLU) that allow the cut-off values to be handled according to trainable parameters in line with equation 5.1.

$$\begin{aligned} \mathrm{PReLU}_i(x) &= \begin{cases} x & \text{if } x > 0 \\ \gamma_i x & \text{if } x \leq 0 \end{cases} \\ &= \max(0, x) + \gamma_i \min(0, x) \end{aligned} \tag{5.1}$$

In addition, PReLU preserves a portion of the negative values each time the activation function is computed, in order to maintain them and regulate their flow to the next layer.

Each block ends with a Batch Normalisation layer to normalise and reduce the internal covariant shift [133], a typical side effect when training networks where the weights at the i-th level tend to generate activation values that shift towards regions where the activation derivatives are zero.

To reduce the spatial dimension at the input of each sequential block, the feature maps are reduced by mean pooling with a kernel$=2 \times 2$. Mean pooling was selected to maximise the information transmitted from each feature map to the lower levels, in order to propagate as much complete information as possible to the last block.

The choice of DSC layers allows us to reduce the number of trainable parameters. In a classic convolution layer defined as $(d, w, h) \rightarrow (d_o, w_o, h_o)$ with kernel $s \times s$, $d_o(s \times s \times d)w_o h_o$ multiplications are necessary.

In DSC, d *spatial* kernels $\mathsf{K}^S_{(h)}$ with $s \times s$ size compute 1-depth convolutions, and a $1 \times 1 \times d$ *depth* kernel K^D gives the final convolution output, as reported in equation 5.2.

$$\begin{aligned} \mathsf{Y}^{(h)}_{i,j} &= \sum_{l=1}^{s} \sum_{m=1}^{s} \mathsf{X}_{i-l,j-m,h} \mathsf{K}^S_{(h)\ l,m}, \ \ h = 1 \ldots d \\ \mathsf{Y}_{i,j,k} &= \sum_{n=1}^{d} \mathsf{Y}^{(h-n)}_{i,j} \mathsf{K}^D_n \end{aligned} \tag{5.2}$$

With this arrangement, only $d_i(s \times s \times 1)w_o h_o + d_o(1 \times 1 \times d_i)w_o h_o$ multiplications are needed, which are of the order of $\frac{1}{s^2}$ compared to the standard

case.

Once feature extraction was completed thanks to the DSC blocks, GAP (see equation 5.3) as proposed in the document by Lin et al. [11] was used in replacement of the MLP classification block.

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_{ijc} \tag{5.3}$$

A structural scheme of the proposed architecture is shown in Figure 5.2.

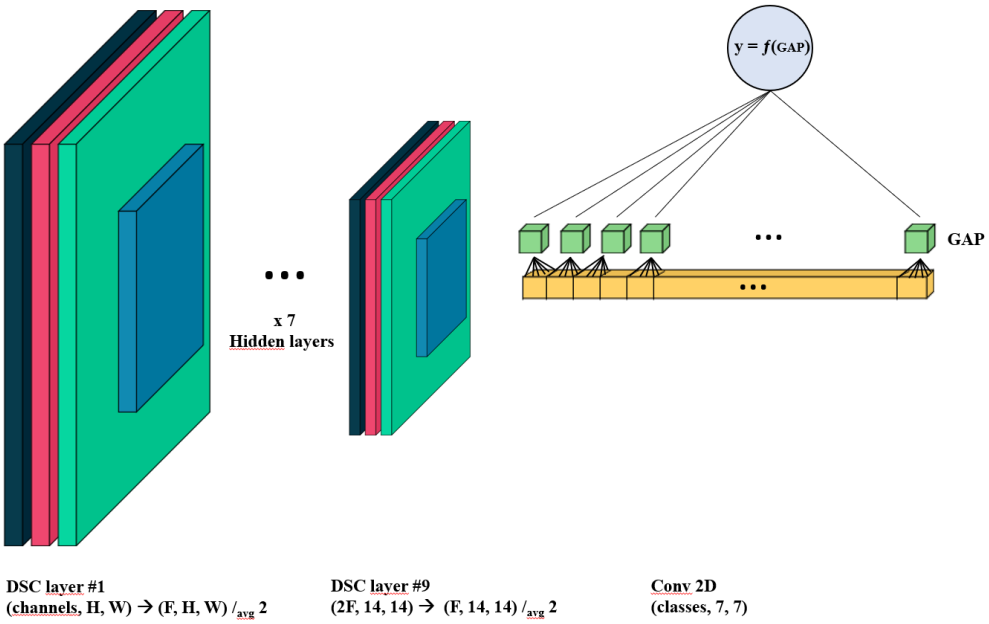


Figure 5.2: Schematic representation of Few-Parameter Architecture.

5.3 Experimental procedure

In this section, the results obtained from the FPA experimental phases will be presented and discussed.

The experiments conducted are grouped into three tasks:

1. **Classification task**, on binary sets, multi-class sets and multi-target multi-class sets.

2. **Region Proposal Task**, to stress the agnostic detection capability of FPA.

3. **Segmentation Task**, using labelled data sets for segmentation.

The experiments in tasks 1 and 2 were performed using several benchmark datasets present in the literature and described in Chapter 3. The first one used is a modified version of the well-known *MNIST handwritten digits* database, called mod-MNIST, and was chosen to highlight the model's ability to identify a specific area for a given class. In addition, to validate the proposal in the biomedical domain, three other data sets were used, two of which belong to MEDMNIST v2 (see 3.5), to *UW-Madison* and *Brain Tumor*. The experiments for task 2 were conducted using the *UW-Madison* and *Brain Tumor* data sets described in Chapter 3. These two data sets were created as segmentation data sets, but in the proposed research activity they were used for classification and feature identification activities. The presence of segmentation masks makes them the most suitable for testing the region proposal task that will allow us to calculate the metrics useful for evaluating the approach. It is important to remember that the neural network was trained agnostically with respect to the ground-truth masks and the training phase was guided only in terms of classification performance. Finally, the same datasets were also used to train the architecture in task 3, i.e. segmentation.

5.4 Results

In this section, the results obtained for the tasks described above are shown, with the addition of ablation study results.

5.4.1 Task 1. Classification

Table 5.1 shows a summary of the results obtained by the FPA architecture, on all the data sets (the Image dim. column shows the size of the images) used in task 1. As can be seen, the performance in terms of AUC is very good for each set, with particular reference to the overall accuracy performance (ACC) for the *mod-MNIST* and *UW Madison* data sets. This last result is significant because it shows the reliability of the network that uses MRI images for the classification of biomedical images.

The performances obtained with the *TissueMNIST* and *DermaMNIST* data sets in terms of accuracy are lower than those obtained with the other data sets. Based on this observation, a comparative study was conducted on the reference architectures used in the article [134]. Table 5.2 shows the results in terms of accuracy, AUC and number of parameters of the architecture. As can be seen, FPA in *DermaMNIST* has the best performance in terms of accuracy,

Table 5.1: Classification results on all classification data sets.

Dataset	**AUC**	**ACC**	**Image dim.**
modMNIST	0.9997	0.9974	(224,224)
tissueMNIST	0.9319	0.6857	(224,224)
tissueMNIST	0.9375	0.6992	(28,28)
DermaMNIST	0.9338	0.7646	(224,224)
DermaMNIST	0.9338	0.7855	(28,28)
UW Madison	0,9989	0,9866	(224,224)
Brain Tumor	0,9634	0,9961	(224,224)

AUC and, above all, the number of parameters used by the architecture. For TissueMNIST, despite AutoKeras slightly outperforming our architecture, it uses $\sim \frac{1}{3}$ of the parameters.

Table 5.2: Comparison between FPA and the reference architectures proposed on DermaMNIST and TissueMNIST.

Architecture	DermaMNIST		TissueMNIST		Paramater
	AUC	ACC	AUC	ACC	
ResNet-18 (28)	0.9170	0.7350	0.9300	0.6760	11,689,512
ResNet-18 (224)	0.9200	0.7540	0.9330	0.6810	11,689,512
ResNet-50 (28)	0.9130	0.7350	0.9310	0.6800	25,557,032
ResNet-50 (224)	0.9120	0.7310	0.9320	0.6800	25,557,032
auto-sklearn	0.9020	0.7190	0.8280	0.5320	NaN*
AutoKeras	0.9150	0.7490	**0.9410**	**0.7030**	147,713
Google AutoML V.	0.9140	0.7680	0.9240	0.6730	NaN*
FPA (28)	**0.9338**	**0.7855**	0.9375	0.6992	**49,995** / **50,086**
FPA (224)	**0.9338**	0.7646	0.9319	0.6857	94,125 / 97,623

* the parameters of these two architectures are not available because they are only accessible via web API

5.4.2 Task 2. Region proposal

In this section we will report the results obtained by the architecture in the region proposal agnostic task. In fact, as reported, the objective of this approach was to identify the most important features map obtained through Global Average Pooling. This assumption aims to extract the areas of interest for a single class within the entire image only thanks to the information obtained from the classification.

Specifically, the aim was to demonstrate how each of the network outputs,

activated on the characteristics that characterise the reference class, focused on the region relevant to the identification of the class. An example is shown in Figure 5.3 calculated on the 10 outputs for the mod-MNIST test images.

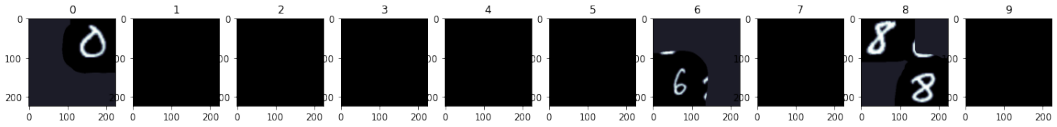


Figure 5.3: A graphical representation of the architecture's feature selection using modMNIST training. It demonstrates that the model is able to identify zones and assign them to the corresponding classes.

As can be seen from the figure, in an image containing the classes $0, 6, 8, 8$ the regions that are detected as relevant are those corresponding to the correctly predicted class. The results obtained show that the theory underlying this representation is correct; nevertheless mod-MNIST is still an oversimplified example, which is why it was deemed necessary to test the approach also on datasets with more complex feature structures. The *UW Madison* and *MRI Brain* datasets were selected for this second experimental phase of task 2. Also in this phase it was observed how the feature maps of the correctly predicted classes highlight the most relevant pixel regions, as can be seen in Figure 5.4 where the initial image, the ground truth, the prediction of our architecture and the intersection between the latter two are shown.

Although FAP provides for ROIs, it does not achieve performance comparable with architectures designed specifically for semantic segmentation, which is the reason why we considered it necessary to define the intensities of the expected characteristics through a trade-off described by the equation 5.4

$$M_i = p_i \times F_i \tag{5.4}$$

where M_i is the ROI mask predicted for the i-th class, p_i is the classification prediction for the same class and F_i is the feature map of the i-th channel of the last layer.

5.4.3 Task 3. Semantic segmentation

The final task for whom experiments have been carried out is segmentation. In contrast to the region proposal task, where the training loss only used a cross-entropy (or a binary cross-entropy), in the experiments carried out for this task a different loss was used that took into account the error calculated

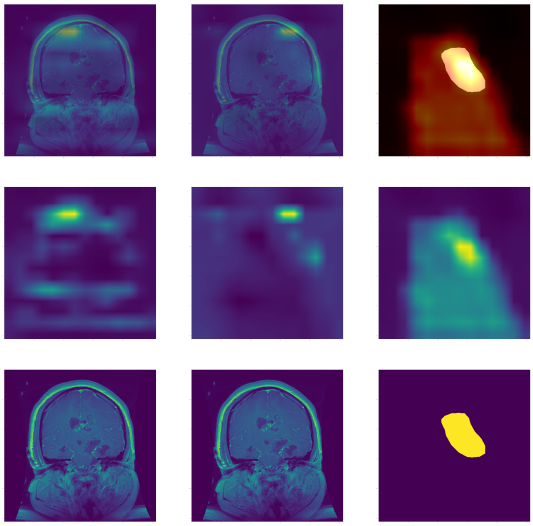


Figure 5.4: The first row displays the predictions overlaid on the original image, allowing a direct observation of how the model interprets the scene. The second row highlights only the predicted areas, facilitating the analysis of the segmentation obtained. Finally, the last row shows the original image and the corresponding prediction mask side by side, offering a direct comparison between the real content and the model's output.

on the ground-truth masks. The results obtained from this phase are shown in the table 5.3.

Dataset	**AVG_DICE**	**Parameters**
UW-Madison	0.6072	165,484
Brain Tumor	0.7528	165,484

Table 5.3: Results obtained in the segmentation task on the UW-Madison and Brain tumour datasets

As can be seen in the table, the Average Dice score values are acceptable considering the very low number of parameters. In fact, at the state of the art, the models that perform best on these datasets have a number of parameters ranging from 7M to 100M. An example of the results obtained by FPA in task 3 are shown in the Figure 5.5

5.4.4 Ablation study

This section shows the results achieved from the ablation study to compare the results gained from FPA with and without an MLP layer for classification. Table 5.4 shows the results obtained from the first phase of the ablation study.

A further set of experiments was conducted varying the activation function

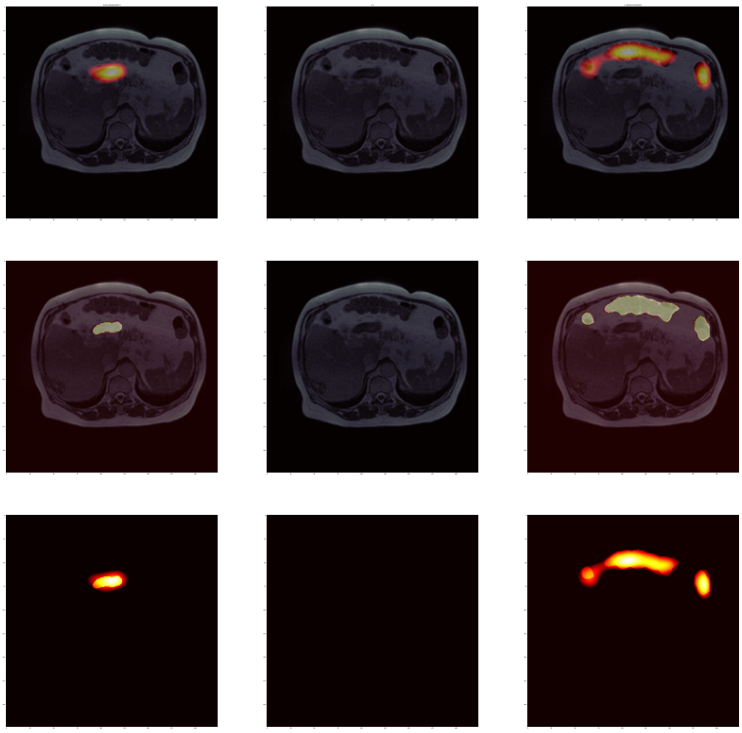


Figure 5.5: Example of the results obtained on the UW-Madison dataset. The three columns present, from top to bottom, the intersection between the prediction and the ground-truth mask, the original mask and the prediction of the proposed architecture for three different samples.

Table 5.4: A comparison of FPA and MLP-out architectures on DermaMNIST and modMNIST.

Architecture	Dataset	AUC	ACC	Parameters
FPA	DermaMNIST	**0.9338**	**0.7855**	**49,995**
Dense	DermaMNIST	0.9312	0.7601	274,181
FPA	modMNIST	**0,9997**	**0,9974**	**50.376**
Dense	modMNIST	0,9994	**0,9974**	82.450

PReLU, selected for its ability to preserve negative contributions in feature maps, with a ReLU. The results of this ablation study are reported in table 5.5

As can be seen from the results obtained from both ablation study experiments, the choice to use only GAP and PReLU has proven to be a winning

Table 5.5: Comparison between FPA with PReLU and ReLU as activation function on DermaMNIST.

Dataset	**Architecture**	**AUC**	**ACC**
DermaMNIST (224)	FPA (224)(PReLU)	**0.9338**	**0.7646**
	FPA (224) (ReLU)	0.6599	0.6688
DermaMNIST (28)	FPA (28) (PReLU)	**0.9338**	**0.7855**
	FPA (28) (ReLU)	0.9282	0.7680

choice. In fact, although the proposed architecture performs very well in task 1, the preliminary results obtained in tasks 2 and 3 are still very promising but not fully satisfactory. The research is therefore still ongoing with the aim of maximising the potential of the model. The obstacle to overcome still lies in the hybrid task, where the absence of masks makes it hard to guide the training in a consistent way.

Chapter 6

AI integrating in medical standard: IODeep

The next section, 6.1, provides an in-depth description of the DICOM (Digital Imaging and Communications in Medicine) standard, with a focus on its architecture, communication mechanisms and information models. This overview is fundamental in order to rigorously contextualise the design choices that led to the definition of IODeep. In particular, an understanding of the DICOM specifications makes it possible to interpret the motivations behind the solutions adopted in terms of interoperability, metadata management, data organisation and compliance with regulatory and standardisation requirements. The analysis of the standard is the foundation of the theoretical and technical premise on which IODeep is designed and developed, which, remaining consistent with the native DICOM constructs, proposes to integrate AI models in the medical field.

6.1 DICOM

In the 1970s, With the advent of Computed Tomography (CT) and other digital imaging modalities on the one hand and with the increasing use of computers in clinical applications on the other, the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) highlighted the need for a standardized method of transmitting medical images between different devices. To meet this need, the solution was to create a standard to improve communication between imaging devices, support image archiving and communication systems (PACS), and enable the development of diagnostic image databases. The standard underwent substantial alteration and was renamed Digital Imaging and Communications in Medicine (DICOM)

in 1993, after the initial versions were deployed. DICOM, unlike its predecessors, was built for TCP/IP-enabled network systems, facilitated offline media interchange, and featured a service-oriented architecture for imaging workflow management. As time passed and technology advanced, a greater number of people were interested in the world of medical imaging, therefore the ACR-NEMA committee became the DICOM Standards Committee. The latter was expressly intended to accommodate all of these new figures in a single, globally consistent format.

The DICOM standard is the primary communication protocol and file format for digital medical imaging (Figure 6.1). It establishes a set of principles and criteria for the collecting, storage, transportation, and presentation of diagnostic images.

The DICOM standard aims to support as many *imaging modalities* as possible, including radiography (RX), magnetic resonance imaging (MRI), ultrasound, and CT. However, making use of new diagnostic processes and the development of new technology necessitates ongoing updates to keep up with advancements.

The DICOM format is defined by a framework that could include not just multimedia data (images/videos), but also clinical information about the patient, the reference examination, the acquisition equipment, technical specifications, and further information. This information, known as *metadata*, is essential for maintaining the integrity, traceability, and compliance of medical images.

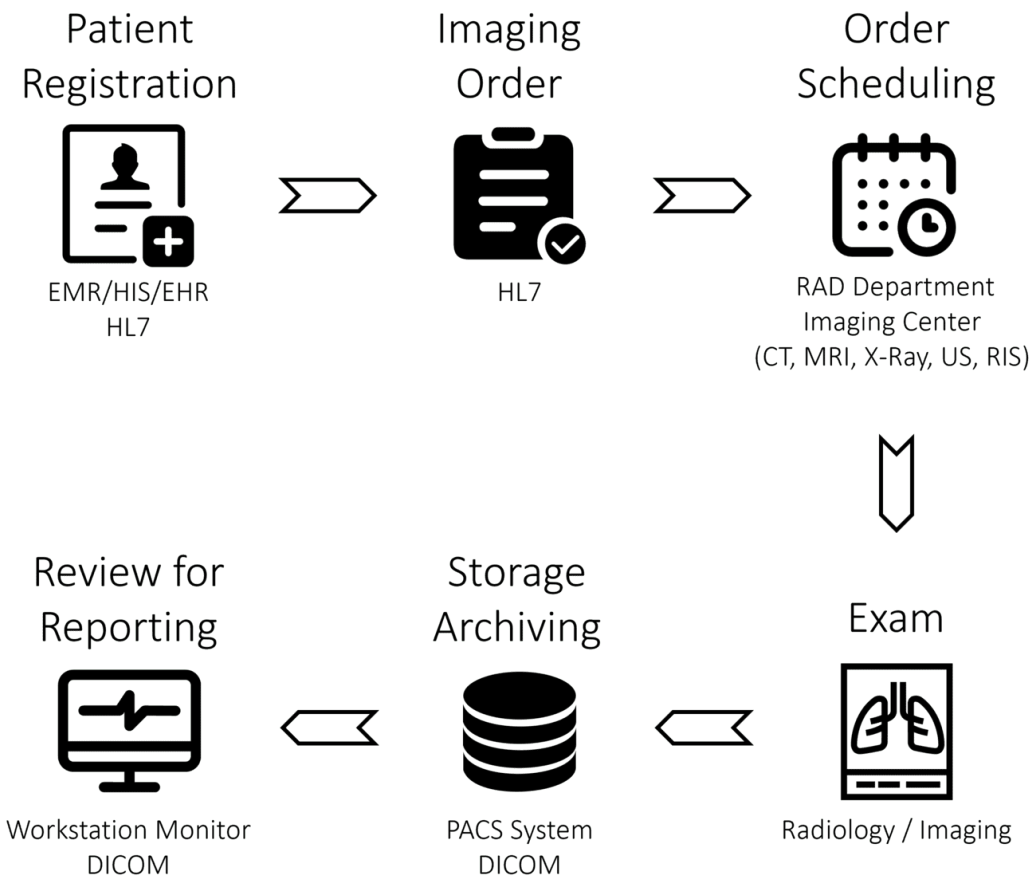


Figure 6.1: The diagram outlines the standard imaging workflow, which begins with patient registration and imaging order entry. The latter, once acquired, are stored in DICOM format through PACS systems and analysed by radiologists to generate diagnostic reports. This clinical pipeline is the infrastructure into which artificial intelligence-based systems should be integrated.

6.1.1 DICOM Standard Hierarchy

The DICOM format is characterized by its hierarchical data organization; this architecture allows information to be naturally classified, resulting in ease of insertion and retrieval for common management tasks such as modification, update, and/or deletion. Figure 6.2 clarifies the hierarchical framework's four levels and key participants.

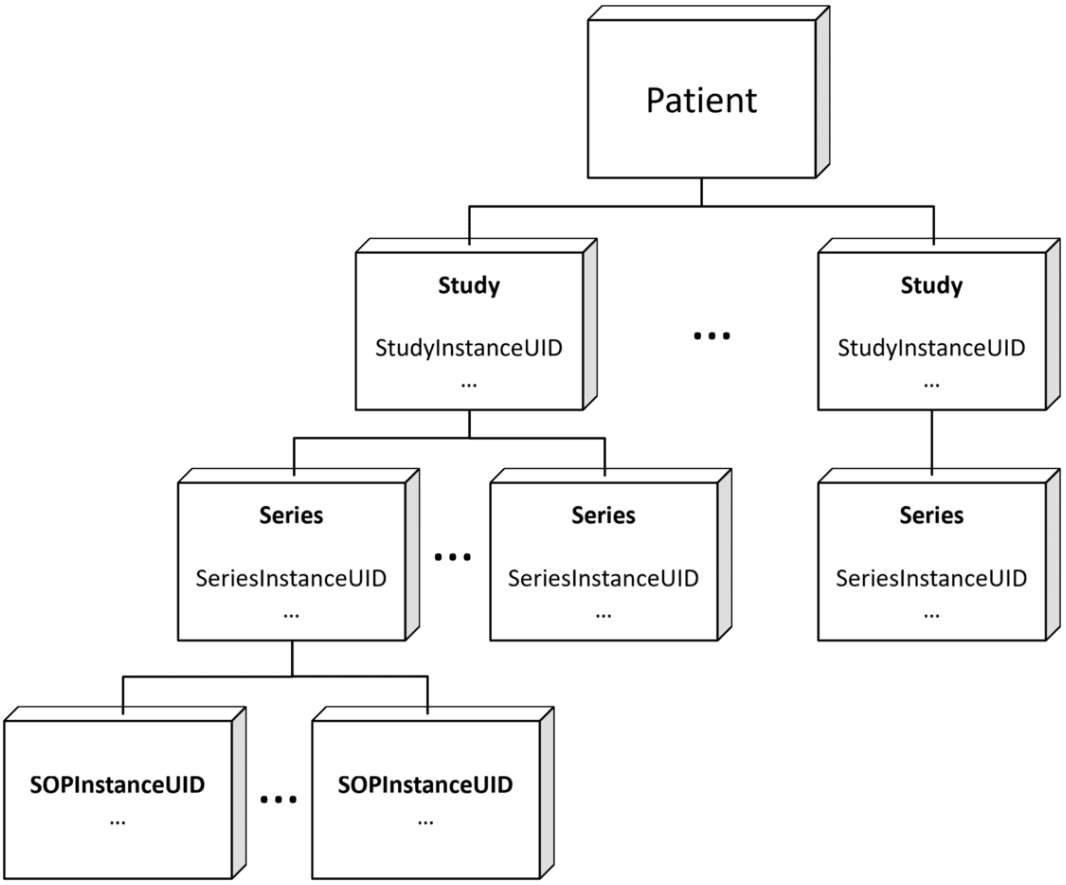


Figure 6.2: Standard DICOM's hierarchical structure. [12]

As can be seen from the figure, the standard is therefore organised into:

- **Patient**: The DICOM hierarchy starts from the patient. The *Patient ID* is the unique identifier assigned to each patient. This identifier offers all patient information, including diagnostic images, to be linked to the subject's personal information.

- **Study**: refers to a set of tests performed on a patient on a given occasion or for a particular medical problem. Each study is identified by a study number, the *Study Instance UID* (0020,000D).

- **Series**: is a subset of images acquired in the same context during a study. Each series has a unique identifier called *Series Instance UID* (0020,000E). The series are organised according to the imaging modalities used and can represent different perspectives or acquisition modalities within the same study.

- **Instance**: represents a single image acquired within a series. They are identified by an instance number called *SOP Instance UID* (0008,0018)

of the series and can differ according to the imaging modality used (e.g. an instance can represent a single cross-section of the patient's body).

The widespread adoption of this standard has influenced how healthcare professionals share and access medical information, providing seamless integration across various systems and imaging devices. Historically, differences in proprietary formats and system requirements impeded efficient communication, slowing workflows and complicating patient care. This system has addressed many of these challenges by standardizing data formats and communication protocols, which allows for a more fluid flow of medical data regardless of software or hardware.

This allows clinicians to access and transmit crucial patient data more quickly, leading to faster and more accurate diagnoses. Better data accessibility has also promoted more effective collaboration among healthcare practitioners, encouraging specialists from disparate locations to work together more efficiently. This has caused a particularly significant impact in sectors such as radiology, cancer, and cardiology, where timely access to imaging tests is critical for decision-making. Furthermore, the ability to connect several imaging modalities and patient records into a unified digital ecosystem facilitates a more comprehensive approach to patient care. With a uniform framework in place, healthcare organizations may better evaluate medical images using cutting-edge technologies such as artificial intelligence and machine learning. This can result in more accurate diagnosis and predictive analytics for the course of diseases. Because of its wide adoption, the DICOM standard is now an integral component of modern medical imaging and the digital infrastructure that supports diagnostic operations. It is a crucial component of modern medical practice since it not only enhances the quality and effectiveness of healthcare services, but also ensures data security and accessibility in the long term.

6.1.2 DICOM Tag's

A structured collection of tags, or attributes, make up each DICOM file, collecting specific information about the medical image and the acquisition environments. These tags are in the "tag-value" format, meaning that each tag is associated with a unique code (tag ID) and can hold text, numbers, dates, or other pertinent information. By ensuring that medical images include the necessary metadata, this methodical approach helps to make them easier to understand, store, and transmit across many systems. DICOM tags are cate-

gorized into modules that classify similar attributes according to their function or application in order to preserve organization and consistency. For example, some modules might have information on the imaging instrument, acquisition parameters, or study-specific metadata, while others might contain patient-related information.

Tables 6.1 and 6.2 show key DICOM tags organized by module. Each row in these tables contains significant details about the tag, such as its name, unique numerical identification, Value Representation (VR), which identifies the data type associated with the tag, and an example of a possible value it can hold. This standardized representation enables medical professionals and software systems to efficiently interpret and handle DICOM data, ensuring compatibility with a variety of medical imaging platforms.

Table 6.1: Main tags in Patient module, General Study module e General Series module

Name	TAG	VR	Values
Patient module			
Patient's Name	(0010,0010)	Person Name	e.g. "Mario Rossi"
Patient ID	(0010,0020)	Long String	e.g. "HF1316"
Patient's Birth Date	(0010,0030)	Time	e.g. 19700101
Patient's Sex	(0010,0040)	Code String	e.g. "M"
General Study module			
Study Description	(0008,1030)	Long String	e.g. "MRI HEAD"
Study Instance UID	(0020,000D)	Unique Identifier	UID
General Series module			
Modality	(0008,0060)	Code String	e.g. "MRI", "CT"
Body Part Examinated	(0018,0015)	Code String	e.g. "HEAD"
Series Instance UID	(0020,000E)	Unique Identifier	UID

Table 6.2: Below a brief overview about Patient, General Study and General Series modules

Name	TAG	VR	Values
Image Plane module			
Image Position (Patient)	(0020,0032)	Decimal String	e.g. "[0, 265, 0]"
Image Orientation (Patient)	(0020,0037)	Decimal String	e.g. "[1,0,0,0,1,0]"
Pixel Spacing	(0028,0030)	Decimal String	e.g. "[1.367, 1.367]
Image Pixel module			
Sample per Pixel	(0028,0002)	Unsigned Short	1 or 3
Photometric Interpretation	(0028,0004)	Code String	e.g. "RGB"
Rows	(0028,0010)	Unsigned Short	e.g. "128"
Columns	(0028,0011)	Unsigned Short	e.g. "128"
Pixel Data	(7FE0,0010)	Other Byte/Word String	Pixel matrix
SOP Common module			
SOP Instance UID	(0008,0018)	Unique Identifier	UID

6.1.3 Modality RTSTRUCT

The RTSTRUCT (Radiotherapy Structure Set) modality is one of the most important uses of the DICOM standard in the field of radiotherapy. Due to it makes it possible to accurately identify and archive anatomical structures and target regions subject to radiation therapy, this method is essential to treatment planning.

An accurate characterization of regions of interest is necessary for successful radiotherapy because it ensures that radiation is supplied to the intended places with the least amount of exposure to neighboring healthy tissues. The RTSTRUCT modality simplifies this process by allowing users to store contour data for these places into designated DICOM tags. As shown in Table6.3, these tags are arranged within the ROI Contour Module and include crucial details about each region, such as its spatial relationships, geometric borders, and related labels. RSTRUCT improves interoperability between

imaging and treatment planning systems by standardizing the representation of structures in radiation. This guarantees that various software and equipment can correctly read and use the same data. This is especially crucial in multi-institutional partnerships, adaptive radiotherapy, and the incorporation of cutting-edge imaging methods, where accurate anatomical mapping is necessary to maximize patient outcomes and treatment efficacy.

Table 6.3: Main Tags in the ROI Contour Module

Name	TAG	VR	Values
ROI Contour module			
ROI Contour Sequence	(3006,0039)	Sequence	
▷ Contour Sequence	(3006,0040)	Sequence	
▷ Contour Image Sequence	(3006,0016)	Sequence	
▷ Referenced SOP Instance UID	(0008,1155)	Unique Identifier	UID
▷ Contour Data	(3006,0050)	Decimal String	Coords

DICOM tags with the Value Representation (VR) "Sequence" work as containers for structured collections of other tags, allowing complex hierarchical data to be stored and arranged efficiently. Two crucial tags in radiotherapy applications are *Contour Data* and *Referenced SOP Instance UID*.

- **Contour Data** stores the three-dimensional coordinates of the points that define the boundaries of a Region of Interest (ROI). These coordinates play a crucial role in precisely mapping anatomical structures, ensuring accurate delineation for treatment planning.
- **Referenced SOP Instance UID** contains the unique identifier (UID) of the SOP Instance to which the ROI corresponds. This ensures that each contoured region is correctly linked to its associated medical image, preserving the integrity and consistency of the data.

The use of these tags is particularly significant for training machine learning and deep learning models for medical image analysis. By associating contour data with a specific SOP Instance UID, it is easy to create a supervised training dataset in which images are associated with their identified anatomical features. This structured data set allows for the creation of powerful AI

models capable of conducting autonomous segmentation, tumor detection, and optimization of treatment planning in radiotherapy. Furthermore, incorporating DICOM RTSTRUCT data into AI-driven workflows increases the potential for better clinical decision-making, reduces manual contouring efforts, and improves the consistency and accuracy of delineated regions between institutions and practitioners. This standardization is crucial for the advancement of personalized and adaptive radiotherapy, where AI-assisted segmentation can dynamically refine treatment plans based on patient-specific anatomical variations.

6.1.4 Server PACS

A Server PACS (Picture Archiving and Communication System) [135] is a information system that is mainly used in the medical field to store, manage, and distribute diagnostic images in a centralized way. Widely used in the healthcare sector, Server PACS eliminate the need for physical image storage, enabling secure digital archiving and seamless access to medical images for healthcare professionals, both locally and remotely. As previously explained, DICOM files contain not only the image itself, but also a number of metadata, such as patient information, scan parameters, and acquisition device information. These metadata are used by Server PACS, which are designed to efficiently manage DICOM images (Figure 6.3), with the following functionality:

- **Medical image archive**: A Server PACS allows physician to safely archive large volumes of DICOM data. This digitization has enabled the removal of barriers and physical supports, facilitating access and distribution by healthcare professionals, even remotely. This transition has significantly improved workflow efficiency, data preservation, and remote collaboration in the medical field.

- **Image Distribution**: One of the key functions of a Server PACS is the distribution of medical images to authorized healthcare professionals, allowing them to quickly access the images required for diagnostic and therapeutic decisions. This distribution can occur within a healthcare facility or between different structures, allowing remote image consultation. Beyond a question, the most important characteristic of a Server PACS is the flexibility with which images can be distributed. Authorized clinicians can immediately access the information they need to make diagnostic and treatment decisions. A system that can scale at different

levels and diffent scenarios:

  - Local Server PACS for a single department where all users belong to a specific clinical unit.
  - Local Server PACS shared across multiple departments that consolidates imaging data from different sources within the same institution.
  - Centralized Server PACS for multiple facilities acting as a unified hub to store and distribute imaging studies throughout an entire healthcare network.

  This flexible architecture supports distributed access to medical images, improving collaboration among specialists and enabling remote diagnostics, second opinions, and multidisciplinary case discussions.

- **Image management**: A Server PACS organizes medical images according to the hierarchical framework outlined in 6.2. A management system that allows health care providers to quickly navigate the tree or search for specific values such patient name, study date, and exam type. Numerous PACS systems support instance viewing with specific plugins, allowing for direct inspection of information without the need for an external application.

- **Integration with other platforms**: Server PACS are part of a larger ecosystem of healthcare information systems, including health information management systems (HIS) and radiology information systems (RIS), to create an integrated workflow and improve operational efficiency [136]. The exchange of data between HIS, RIS, and PACS follows a specific Integration Profile (i.e., a standardized scenario) known as Scheduled Workflow, as outlined by IHE (Integrating the Healthcare Enterprise). The latter is a non-profit organization that collaborates with several worldwide healthcare organizations to encourage the use of common information technology standards for data sharing.

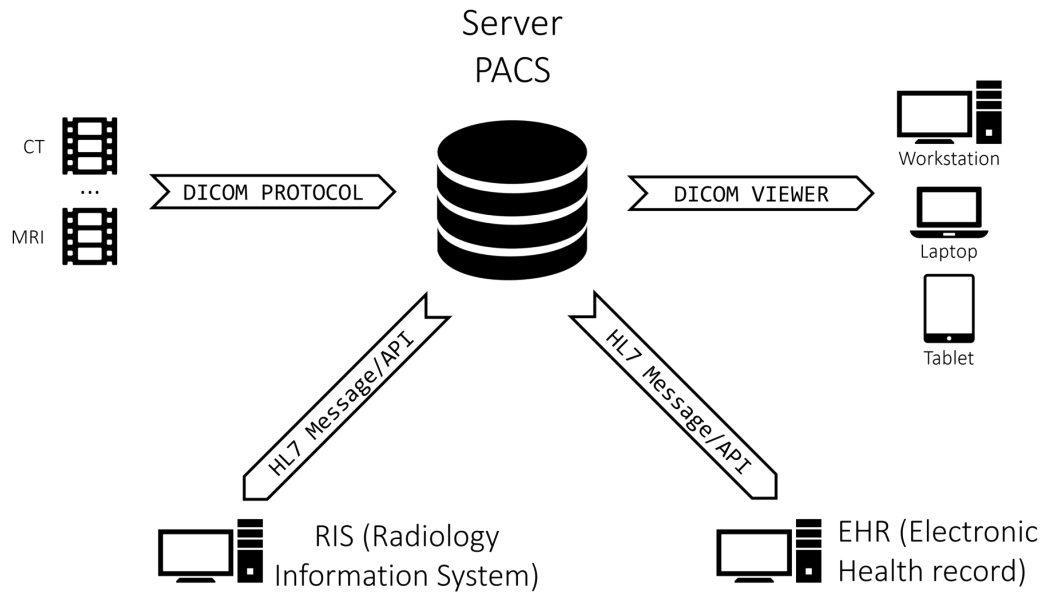


Figure 6.3: Server PACS architecture. [12]

6.1.5 DICOM Viewer

A **DICOM Viewer** is a specialized software application used to view, analyze and process medical images stored in the DICOM format. It is an essential tool in the medical field, where imaging plays a crucial role in diagnosis and treatment planning. A DICOM viewer supports images from various different sources (MRI, CT), ensuring compliance with the DICOM standard for interoperability with image archiving and communication systems (PACS). It allows users to view images from multiple medical imaging devices on a single platform. It offers a set of advanced image manipulation features, including *zoom, pan, rotate and flip* for better viewing and analyzing. Users can also adjust window leveling and contrast to improve image quality. In addition, features such as 3D reconstruction and multiplanar reconstruction (MPR) help improve spatial understanding of structures. Measurement and annotation tools allow users to measure distances, angles and regions of interest (ROIs) and add text annotations, markers and notes to images to improve communication between specialists. Integration with PACS, HIS (Hospital Information Systems) and RIS (Radiology Information Systems) ensures image retrieval. Security and compliance are integral aspects of a DICOM viewer, which implements encryption, user authentication and access control to protect patient data, ensuring compliance with defined standards for medical data. There are different types of DICOM viewers. Standalone viewers are installed on

local computers for offline access and analysis, while Web-based viewers are cloud-based solutions accessible through Web browsers without the need to install software. Open source viewers, such as RadiAnt [137], Horos [1], OsiriX[2], and 3D Slicer[3], offers community-supported alternatives, whilst commercial viewers are feature-rich software solutions that are integrated with PACS and distributed by medical imaging companies. Furthermore, DICOM viewers are useful in telemedicine since they enable for remote consultation and diagnosis. For these reasons, they are essential tools for modern medical imaging, allowing healthcare personnel to efficiently interpret and handle medical images. DICOM viewers are evolving to include advanced capabilities for better patient care and medical research, thanks to improvements in cloud computing and artificial intelligence.

6.2 AI in DICOM

In recent years, artificial intelligence (AI) has made significant advances in a variety of areas, resulting in increasingly advanced performance in complicated tasks. The medical field has also attracted the attention of the scientific community, with an increasing number of studies aimed at incorporating AI models into diagnostic and therapeutic processes. Despite the remarkable outcomes that were obtained, the community struggles to accept and opposes the widespread application of AI in clinical practice. The lack of trust in healthcare professionals in the face of AI-powered predictions is a significant obstacle. Although such models can be helpful for diagnosis, their application in healthcare facilities is growing with caution. The effective usage of AI could reduce the strain of healthcare staff by streamlining analysis time and lowering the danger of human errors [138]. In addition, patients would benefit from more objective assessments that are devoid of prejudice caused by the individual specialist's subjective interpretation. A well-trained AI model has the potential to identify diagnostic patterns that are invisible to the naked sight, thereby contributing not only to the diagnostic phase but also to the preventive phase by recognizing early indicators of disease that would otherwise go unnoticed. Despite their obvious potential, these novel tools are rarely integrated into clinical workflows. Currently, the use of AI in clinical practice is limited to specific experimental applications, although the field is rapidly evolving. The goal of

[1]https://horosproject.org/
[2]https://www.osirix-viewer.com/
[3]https://www.slicer.org/

this researches is to construct transparent, reliable, and regulatory-compliant models that can operate alongside physicians without replacing them, increasing the quality of diagnosis and healthcare [139]. A seamless integration of DL in this type of system could lead to the recognition of different types of diseases and disorders, with real data that comes from various sources (MR, CT) but also from multiple facilities, using different tools to acquire the same type of image with a variety, making training more robust and less data-dependent. A configuration as described fits neatly with the idea of federated learning. The problem was solved in [140] and [141], using a local distribution of models that go to update the weights of a central model. If the models achieve acceptable performance, they may utilize XAI techniques to demonstrate robustness and increase confidence in their predictions [142]. Recently, the first approaches of framework integration within the DICOM standard, as done in [143], suggests a solution for slides and medical reports management, as well as an image analysis compatible with open-source applications such as Dicoogle PACS, with the possibility of integrating ad hoc solutions developed by users. In [144], the work develops Visilab Viewer, a web-based platform that allows for the viewing of a particular type of image: Whole Slide Images (WSI) from a multi-frame DICOM. AI algorithms are being integrated into clinical workflows rather than being included in standards, as shown in [145]. Images are stored in a PACS system and accessed through external services through a typical web application configuration, resulting in a RESTful web services (DICOMweb™)3 system. In this case, the external application must implement DICOMWeb Client APIs that invoke the DICOMWeb Server via HTTPS Protocol. The authors then chose a subset of DICOM tags required to investigate the microscope images.

Similar applications, as described in [146], involve the development of a platform by the EMPAIA consortium that permits direct connectivity between a web browser and a medical or computer infrastructure. This enables users to manage data while running third-party artificial intelligence applications on the same data. A significant advantage is being independent from the PACS system or WSI image management tools. In [147], a roadmap for integrating image analysis algorithms based on artificial intelligence in existing radiologic workflows is proposed, including a study on cerebral magnetic resonance. In the work [148], a strategy for integrating AI-based image analysis algorithms into traditional radiologic processes is proposed, illustrating a real-world case of cerebral magnetic resonance. In [149], the framework Niffler is presented,

which is a machine learning system capable of acquiring images from PACS via DICOM network tools. Niffler extracts and generates metadata from stored images, which allows real-time analysis of both radiological and textual data. Finally, in [150], an automatic learning model for automated medical image analysis is demonstrated, which is integrated in a PACS system using DICOM services based on open source tools. Other examples of AI applications can be found in [151] the authors proposed a DICOM Imaging Router which incorporates CNNs for categorizing unknown DICOM X-ray images into five anatomical groups: "abdominal", "adult chest", "pediatric chest", "spine", and "others". Although the frameworks mentioned above achieve a DICOM infrastructure to perform their job, it requires the usage of a range of accouterments that burden specialized workers who must learn how to handle them [152].This highlights that while the DICOM standard and AI applications remain two separate worlds, they may operate together synergistically when necessary, albeit without true integration. In [153] the PyRaDiSe package has been developed, which goes in the direction of a tight integration. PyRaDiSe is an open-source Python package which is independent of DL frameworks, and addresses the issue of artifacts caused by 2D reconstruction as it provides a framework for developing auto-segmentation solutions that can directly operate on DICOM data. Authors claim that PyRaDiSe helps to bridge the gap between data science and clinical radiotherapy by facilitating the implementation of deep learning segmentation models in clinical research practice. Actually, PyRaDiSe has the same research objective as IODeep, and the authors leverage DICOM RT Structure Sets (RTSS) to allow data conversion from DICOM to other image formats thus enabling easy auto-segmentation routines. It is well known in the DICOM related literature [154] that DICOM RTSS are a suitable place to store AI related information like the labels for models training, and also the framework we devised for IODeep makes use of DICOM RTSS to store ROI contours. Differently from PyRaDiSe, in IODeep there is no need to code the segmentation solution. IODeep provides the physician with predictions about relevant ROIs in a completely transparent way. Right now, the information architecture of the DICOM standard does not foresee any type of inclusion of AI, neither as a proper IOD nor in terms of defining a suitable Information Object Module (IOM) to be included in a more general structure. Currently, DICOM is moving towards the integration of AI applications, and a Work Group has been created purposely that is the "WG-23: Artificial Intelligence/Application

Hosting"[4]. The main activity of the WG-23 has been oriented in defining mechanisms for discovering heterogeneous AI services that can expose a suitable manifest for declaring the DICOM services provided to the imaging platform. In the present work we adopt an approach that relies on a direct extension of the DICOM information architecture. This architectural choice derives from the previous experience by some of the authors in developing a framework for adaptive configuration of the PACS viewers' GUIs based either on the content (i.e. reason for study, modality and body part) of the images to be displayed or on explicit preferences issued by the radiologist. Configuration information is stored in a dedicated IOM which extends the DICOMDIR IOD [155]. We think that the design of new components of the DICOM information architecture, as in the case of IODeep, makes the extension to the standard simpler than defining the interfaces to interact with an external software ecosystem. Moreover, the architectural solution devised by the WG-23 could be prone to problems in a training and/or fine-tuning scenario, when a huge amount of data would have to be moved across the interface to feed the DL model. We developed both a "monolithic" PACS client that implements all the workflow related to the use of IODeep and the service architecture that makes use of IODeep according to the indications of the WG-23. In the next section we detail our implementation, and we compare the two solutions.

In this scenario, during my research for my doctorate I conducted an experiment to fill this gap and finally succeed in integrating a Deep Neural Network into the DICOM standard, presented in the paper 'IODeep: an IOD for the introduction of deep learning in the DICOM standard' by Contino et al. [12].

6.3 IODeep Architecture

This section will describe the architecture and data model, schematized with an E-R diagram Figure 6.4, this represents how IODeep is connected to the real-world DICOM model[5]. The diagram highlights that the information contained in IODeep is needed to instantiate a specific DNN, which will go to predict ROIs on different images. The references of the images and their respective ROIs, will be stored in the DICOM architecture.

The entire IODeep structure is shown in table 6.4, the informations stored in "DNN", "Image Pixel", "General Study", and "General Series" IOMs are essential for the right choice of the neural network, that will be used to identify ROIs

[4] `https://www.dicomstandard.org/activity/wgs/wg-23`

[5] `https://dicom.nema.org/medical/dicom/current/output/html/part03.html`

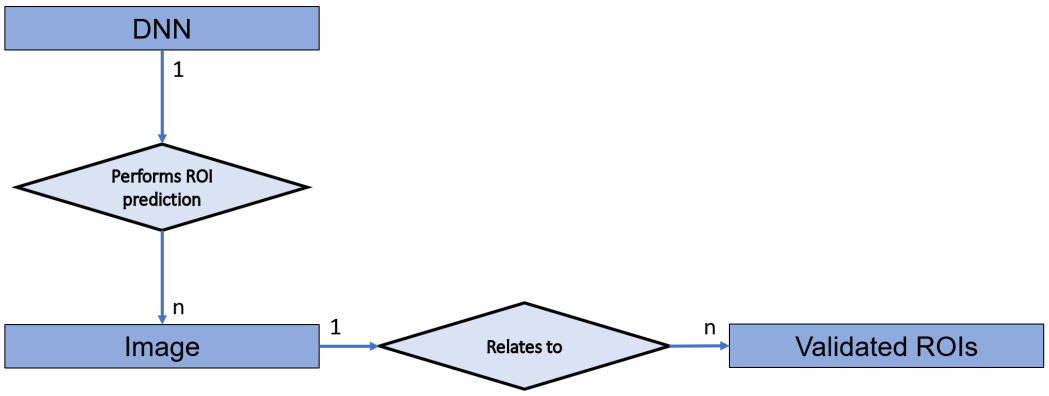


Figure 6.4: E-R diagram representing the connection between IODeep and the Dicom Model of the Real World [12]

over the examined image. The suggested IOM, *DNN*, comprises model-related data. It contains information about the architecture and weights, as well as indicators about the training hyperparameters, whether the network was trained from scratch or through fine tuning. The *Image Pixel* IOM maintains information about the image's structural properties, which are required for arranging the input for the DNN instance. The *General Series* IOM contains information on the image modality and which anatomical part is referred to during the capture. The *General Study* is a mandatory module, because it contains the *StudyInstanceUID*, this tag its fundamental to create the Service Object Pair (SOP) Class aimed at storing/retrieving IODeep instances from the Server PACS.

As defined by WG-23 DICOM, the proposed IOD is a non-patient IOD and contains no patient information. In full compliance with the standard and its VMs, the Patient Module fields (not shown in the 6.4 table) are not populated with any information. The architecture of IODeep, is to set the value of the tag *DnnUID* to populate the fields The *StudyInstanceUID* in the "General Study" module and the *SeriesInstanceUID* in the "General Series" module, which however remain mandatory but not used in this work.

Assuming that our PACS runs in an IT infrastructure where it is possible to access a DNN back-end in Pytorch or Tensorflow, the DNN Module structure contains information about the network. The defined tags are:

- *DnnUID* is the unique identification code.
- *DnnName* is a name that summarizes data on which it was trained and model type.

Table 6.4: The general *IODeep* structure reporting both the modules and the tags required for selecting and instancing a DNN architecture.

Name	**TAG**	**VR**	**Values**
DNN module			
DnnArchitecture	(0017, 00XX)	UT	DNN architecture
DnnWeitghts	(0017, 00X1)	UT	DNN weight
DnnName	(0017, 00X2)	PN	e.g. "Brain Tumor segmentaion Unet"
DnnUID	(0017, 00X3)	UI	UID*
DnnTrainingInfo	(0017, 0024)	UT	Dictionary
DnnFineTuning	(0017, 0026)	CS	True or False
Image Pixel module			
PhotometricInterpretation	(0028, 0004)	CS	RGB, MONOCHROME1... (DICOM standard Section C.7.6.3.1.2)
SamplesPerPixel	(0028, 0002)	US	3 or 1
PatientOrientation	(0020, 0020)	CS	["P", "F"]; ["L", "P"];["L", "F"] ... (DICOM standard Section C.7.6.1.1.1)
PlanarConfiguration	(0028, 0006)	US	0 or 1
General Study module			
StudyInstanceUID	(0020, 000D)	UI	UID*
General Series module			
SeriesInstanceUID	(0020, 000E)	UI	UID*
Modality	(0008, 0060)	CS	CT, MR, PT ... (DICOM standard Section C.7.3.1.1.1)
BodyPartExamined	(0018, 0015)	CS	BREAST, ABDOMEN, CHEST ... (DICOM Part 16: Content Mapping Resource)

* These tags share the same Unique IDentifier that is the one defined for the *DnnUID* tag

- *DnnArchitecture* contains the architecture of the DNN.
- *DnnWeights* contains the weights of the DNN model.

It should be highlighted, that the DnnArchitecture and DnnWeights fields are defined with VR equal to unlimited text (UT), this is an intentional choice motivated by the fact that the authors claim to be able to accept structured files such as JSON and XML or URIs to local binary files.

In 6.5, an example JSON file is provided, where the `input_shape` keys stand out, in which we acquire information about the input tensor and we might change the picture to match the considered slide. The field `architecture` contains a sequential list of layers that compose up the model. To expand the potential of network representation in this structure, the key `skip_connection` has been included. It allows you to generate not only "linear" models like the traditional VGG, but also more sophisticated networks like the ResNet family or a U-Net. In addition, thanks to the `backend` field, the parser will be able to instantiate the model using the selected DNN back-end; this process is quite transparent to the user. The system will automatically instantiate the model provided in the JSON file and, if available, load pre-trained weights for a certain image type. The defined pipeline is fully complaint with the standard and complies with privacy and security terms, in fact the back-end goes to operate on the image displayed in the PACS. This ensures that no privacy issues occur because the radiologist is authorized to see the patient's data and the network uses only the values stored in the "Frame Of Reference" and "Image Pixel" modules of the slice.

6.4 The ROI prediction workflow

As previously noted, the most interesting aspect of IODeep is the idea of incorporating AI into the DICOM standard. However, it is important to remember the ROI prediction tool, which shows regions that the specialist may would like to monitor. In the Figure 6.6, the UML diagram explains the pipeline of ROI prediction. To show the efficiency of the suggested method, a very basic and lightweight PACS viewer was constructed.

The doctor starts the process by selecting a patient to observe. Next, select the reference study and, within it, choose one of the series available from the list. After identifying the series, visualize the arrangement of the slides for a more in-depth analysis. After identifying that particular slide to work on, the doctor can interact with the Viewer PACS and click the "AI ROI" button. This command activates an automatic mechanism tasked with identifying and predicting the regions of interest (ROI) inside the slide.

The implemented PACS Client starts the process by calling the Server PACS and requesting a list of IODeep instances. This is accomplished by reading the tags in "General Series" and "Image Pixel" from the currently active slice. After that, certain tags are analyzed to identify imaging properties. The Modality

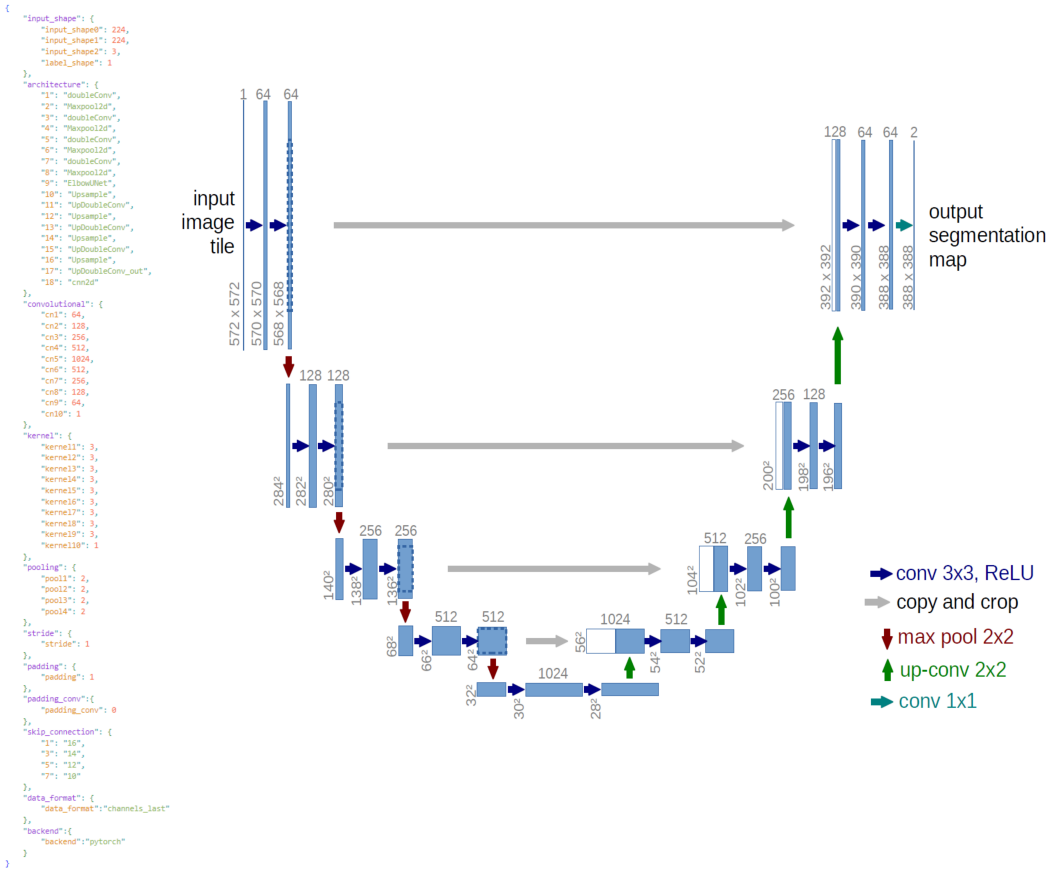


Figure 6.5: Representation of the U-net neural network used in our example, and its JSON description.

tag (0008, 0060) specifies the modality of image acquisition, whereas the BodyPartExamined tag (0018, 0015) identifies the anatomical region under study. If the latter is empty, an attempt is made to derive it from the values of the StudyDescription tag (0008, 1030), whilst the SamplePerPixel tag (0028, 0002) offers information on the number of channels in the recorded data. The combined analysis of these factors allows us to determine both the type of examination performed on the patient and the body part involved. Finally, the method 2 is used to identify the most suited neural network. Despite the fact that all of the information required for the construction of the model is available, it is critical to perform a preliminary check on the coherence of the shape of the slices to ensure that they can be correctly elaborated from the neural network. This step is critical for avoiding compatibility errors during the inference phase and ensuring accurate medical image preparation. To conduct this evaluation, the tags SamplePerPixel (0028, 0002), Rows (0028, 0010), and Columns (0028, 0011) are examined, which provide information on the number of channels in the image, the number of rows, and the number of columns

in the pixel matrix. If the values extracted by these tags do not correspond to the shapes required for neural network input, the appropriate transformations, such as resizing, format conversion, or normalization, are applied to ensure compatibility. Aside from the image format, a control over the pixel interpretation mode is performed using the value of the tag PhotometricInterpretation (0028,0004). This parameter defines how pixel data should be read and interpreted by the system, distinguishing between several modes such as MONOCHROME2, MONOCHROME1, RGB, and others. This verification is critical for avoiding distortions during image analysis and ensuring that the deep learning model receives data in the correct format. An additional analysis is performed on the tag PixelRepresentation (0028,0103), which indicates the type of numerical representation of pixels, specifying if they must be read as signed or unsigned. This validation and trasformation process are performed in *check_tensor_shape()*, its implementation is shown in the Algorithm 3. This functions make sure about images shape and their fully compatibility with the model input requirements, avoiding the risk of errors.

To evaluate the performance of our implementation, we trained some example U-Net models using two specific datasets in the field of medical imaging. The first dataset employed is "Brain Tumor Classification (MRI), " a dataset containing brain magnetic resonance imaging (MRI) images used for brain tumor classification. The second dataset used is "UW-Madison GI Tract Image Segmentation ". This allowed us to recreate two real-world scenarios that a physician may face during his or her diagnosis and evaluation activities.

The process described so far will output a ROI that will be shown by the viewer. After a visual evaluation, the clinician can decide whether to validate or reject the proposed model. The Figure 6.7 shows an example of a proposed ROI, with the appropriate buttons for managing the prediction.

The DICOM viewer was implent used Python and its PyQT library, it was designed so that after selecting first study and then series, the instance with its slices is shown. The display shows two blocks of information at the top. The left block displays the patient information: *PatientName* (0010, 0010), *PatientID*(0010, 0020), *Patient Birth Date* (0010, 0030) , *Patient Gender* (0010, 0040). The second block displays clinical information: *AccessNumber*(0008, 0050), *InstitutionName*(0008, 0080), *PhysicalNameReferral* (0008, 0090), *StudyDate* (0008, 0020), *StudyDescription* (0008, 1030), *StudyID* (0020, 0010), *StudyInstanceUID* (0020, 000D) and *StudyTime* (0008, 0030). The central body of the viewer will display the image. In this part of the layout are

the “autoplay” and navigation buttons, in addition to the "AI ROI" button, already mentioned. The interaction between the viewer and the DICOM standard was developed using the Pydicom library, while the server is an instance of the DICOM Orthanc[6] server.

As previously explained, the GUI allows the physician to check the ROIs offered by the algorithm and decide which of them will be preserved in a

[6] `https://www.orthanc-server.com/`

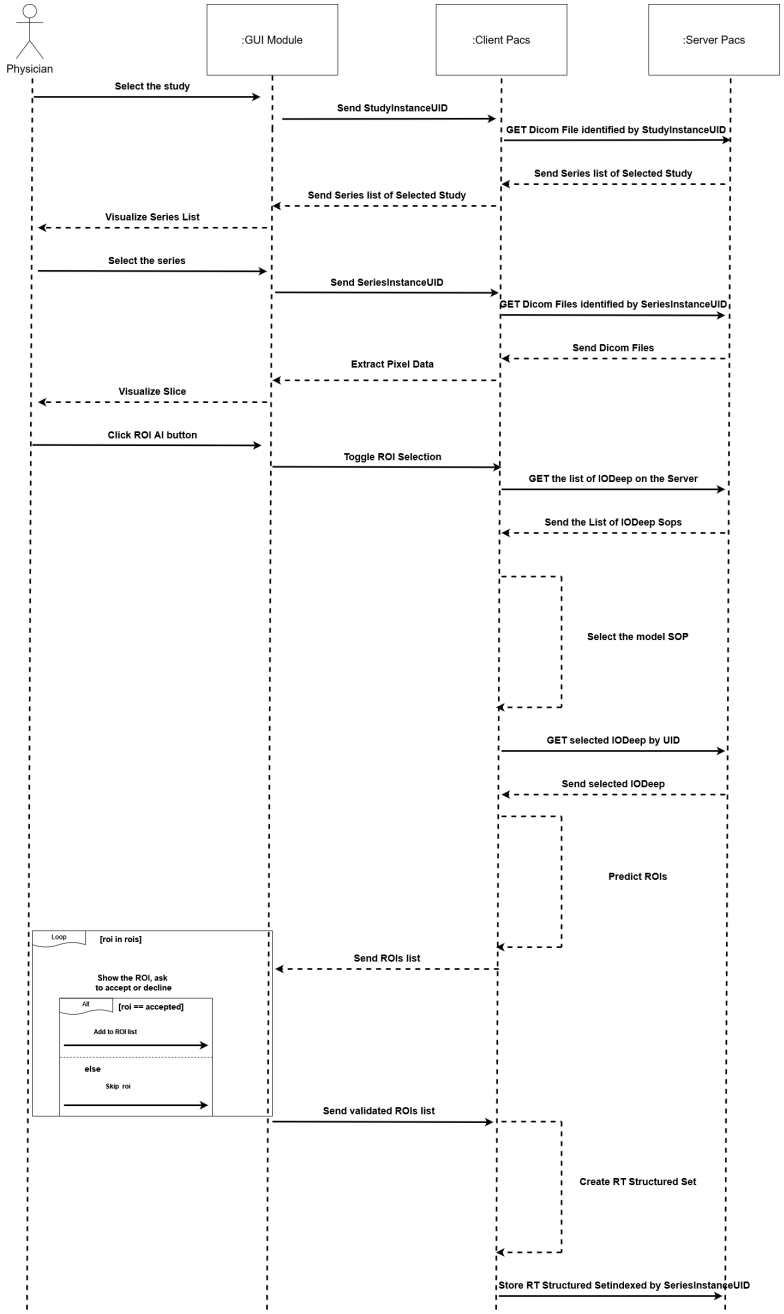


Figure 6.6: Sequence diagram of the ROI prediction scenario

Algorithm 2 Network selection algorithm

Require: *SliceTagList*: list(*Modality*, *SamplePerPixel*, *BodyPartExamined*, *StudyDescription*)
▷ The list of relevant tags from the ▷ current slice

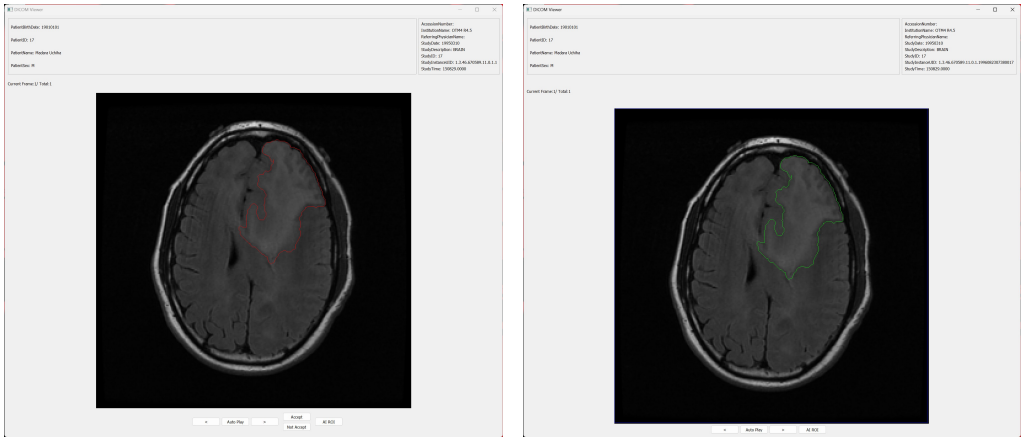
Require: *IODeepList*
▷ The list of all the IODeep instances ▷ retrieved from the Server PACS

if *SliceTagList.BodyPartExamined* == ∅ **then**
    SliceTagList.remove(*BodyPartExamined*)
else
    SliceTagList.remove(*StudyDescription*)
end if
for *net* in *IODeepList* **do**
    IODeepTagList ← list(*net.Modality*, *net.SamplePerPixel*, *net.BodyPartExamined*)
    control ← 0
    for *slice_tag*, *net_tag* in (*SliceTagList*, *IODeepTagList*) **do**
        if *slice_tag* == *net_tag* and not *SliceTagList*.isLast(*slice_tag*) **then**
            control ← *control* + 1
        end if
        if *SliceTagList*.isLast(*slice_tag*) **and** (*slice_tag* == *net_tag* **or** *slice_tag*.substring(*net_tag*) == True) **then**
            control ← *control* + 1
▷ When using *StudyDescription*, a pattern search ▷ is used in place of strict equality on the last tag
        end if
    end for
    if control == 3 **then**
        net_sop_uid ← *net.DnnUID* ▷ The SOP UID of the selected IODeep instance
        return *net_sop_uid*
    else
        net_sop_uid ← *None*
    end if
end for
return *net_sop_uid*

Algorithm 3 ROI prediction algorithm

Require: *IPModule* ▷ The ImagePixel module of the current slice
Require: *FRModule* ▷ The FrameOfReference module of the current slice
Require: *IODeep* ▷ The selected DNN
Require: *Backend* ▷ The reference to the DNN back-end
rois ← list()
network ← parser(*IODeep.DnnArchitecture*)
check_tensor_shape(*IPModule*,*network.input_shape*) ▷ tensor shape analysis
model ← *Backend*.createNetwork(*network*) ▷ instance of the network in the DNN back-end
model.load(*IODeep.DnnWeights*) ▷ loading method of the DNN back-end
preds ← *model*.predict(*IPModule.pixelData*) ▷ prediction method of the DNN back-end
for *roi* in *preds* **do**:
    slice_id ← *FRModule.FrameOfReferenceUID*
    rois.add(dict("*sliceID*" : *slice_id*, "*polyline*" : *roi*)
▷ These value will populate the RT Structure Set containing the ROIs
end for
return *rois*

DICOM RT Structure Set, a specific construct of the standard that permits data from virtual instruments/positions to be stored. In addition to the classical patient information obtained from the IOD, the RT Structure Set will store frame references and their related ROIs. The values of the "Frame of Reference" and "ROI Contour" modules will be added to the instance of the *FrameReferenceUID* tag (0020, 0052) and the *ROIContourSequence* tag (3006, 0039).



(a) (b)

Figure 6.7: The viewer interface for ROI validation. (a) Predicted ROIs are displayed and outlined in red. (b) Validated ROIs are outlined in green. [12]

Finally, the validation information will be kept via the ”Approval” form, which will include the tags:

- *ApprovalStatus* (300E, 0002)
- *ReviewDate* (300E, 0004)
- *ReviewTime* (300E, 0005)
- *ReviewerName* (300E, 0008)

Saving DICOM RT Structure Sets in the Server PACS has the significant advantage of allowing you to access cases of interest with ROI references without having to recalculate them. The ROI data is stored in DICOM as a list of polygon points.

6.5 Comparison with the DICOM WG-23 proposal

The DICOM community is actively working on integrating artificial intelligence algorithms into the standard, and a separate working group has been formed: the WG-23 ”Artificial Intelligence and Application Hosting. The present WG-23 proposal was submitted in the form of *work item*[7], as is the standard procedure for working groups to publish their results.

[7] `https://www.dicomstandard.org/workitems`

WG-23 aims to integrate AI utilizing a (micro-)service discovery method based on the Open Application Model (OAM) [8]. In this paradigm, AI applications are deployed as services that run on a variety of computing infrastructures, which may be on the same network as the PACS or in the cloud. Each application has a "Manifest" in the form of a YAML file that contains the definition of a task and the information required to execute it. OAM also defines a "application operator," which is the infrastructure that actually deploys the application, as well as additional operational information to enable the so-called "platform" (the PACS infrastructure that uses the services, in our case to explicitly integrate them into its workflow).

The concept shifts from a monolithic implementation perspective in which information is completely described according to the DICOM information architecture and just a TCP/IP socket connection is required to connect to the DNN back-end running on the same network as the PACS. In addition to the privacy and security concerns discussed above, we chose this option for efficiency reasons. The service ecosystem described by WG-23 is a scalable solution that requires little implementation effort on the part of the PACS; nevertheless, it is designed for an inference situation and is not ideal for training or tuning.

Radiology departments are generally focused on specific diseases due to their location and the presence of specific medical specialties. As a result, putting up a network trained using the unique data generated in a specific PACS infrastructure looks to be a reasonable extension of the usage of DNNs in Medical Imaging. Making a dataset available to the DNN's backend for model training requires a significant quantity of bandwidth. A service implementation is not suitable for training and/or tweaking because, in general, a service architecture is designed to be hosted in a cloud location apart from the applications that utilize the service. In the model training situation, this causes considerable latency and privacy difficulties. On the other hand, our system requires only one data transfer across the department's internal network, which consumes very few resources.

IODeep can also be utilized explicitly in this context: a PACS client specialized to back-end administration could select a dataset consisting of various sets of slices and launch the above-mentioned network selection phase, followed by backend training. Nonetheless, we adapted our implementation in accordance with the WG-23 work strategy. In this case, the algorithm 2 is remade to

[8] `https://oam.dev/`

give a service discovery REST (Application Programming Interface) API at a known endpoint of the service design. The PACS platform simply has to know the URI, and IODeep stores the API call metadata in the service discovery endpoint. This interaction link the platform to the selected DNN service endpoint. At the service endpoint, the PACS platform contacts the DNN service API, passing the slice and tensor shape verification information stored in the indicated IODeep. Finally, the chosen DNN service uses the 3 algorithm. This service architecture does not need to save network information in the IODeep architecture since we can assume that all networks have already been established in the DNN backend. In our approach, the Server PACS, client, and REST services all operate in distinct Docker containers which communicate through a virtual LAN defined in the YAML configuration file. Containers can be deployed on a single system or across a cluster.

With a vision of real-world application, the developed dicom viewer was equipped with the ability to require physician validation of the model's predictions. These are the prerequisites for the application of a principle of "human in the loop" or better yet "expert in the loop" that allows only the most clinically meaningful ROIs to be stored in RT Structure Set. In an application scenario, in which a hospital facility manages to create its own IODeep instances for various departments and various types of images, one can imagine fine tuning over time as new images are acquired to create increasingly robust and high-performing instances. This is possible provided that the bandwidth problems of moving data from PACS to the back-end of the DNN are easily solved. In this regard, a solution that runs entirely in the same LAN is definitely preferable to a cloud service ecosystem.

Chapter 7

Image Generation

In Chapter 3, one of the main issues negatively affecting the training and performance of Deep Learning models is the scarcity of data. To overcome this limitation, data augmentation techniques or generative methodologies are used that utilise models to create new synthetic data that faithfully reproduces the statistical and structural characteristics of the real data. The use of Artificial Intelligence models in the medical field is increasingly moving towards the non-invasive generation of images, starting from previously acquired images. These technologies allow us not only to simulate anatomical structures, organs and physiological processes with a high degree of detail and accuracy, but also to expand the diagnostic and theranostic potential of traditional medical imaging, supporting the evolution of modern medicine.

Acquisition techniques such as computerised tomography (CT), magnetic resonance imaging (MRI) and ultrasound have profoundly transformed the diagnostic and therapeutic approach to numerous diseases. However, these methods also have their limitations: CT, for example, exposes the patient to ionising radiation, with a consequent increase in the long-term risk of cancer. Furthermore, the need to perform multiple scans to obtain high-resolution images can generate high costs and increase the psychological and physical stress on the patient, adversely affecting his/her general state of health.

Diagnostic imaging allows doctors and healthcare professionals to visualise specific anatomical structures, identify anomalies and pathologies, make early diagnoses, monitor clinical evolution and plan personalised treatments. All this happens in a non-invasive way, offering essential support for therapeutic follow-up, in contexts where clinical observation alone would not be sufficient.

The introduction of advanced AI-based technologies has opened up innovative scenarios, making it possible to generate high-resolution images from low-resolution scans. This makes it possible to significantly improve diag-

nostic quality, while reducing the invasiveness of the process and the overall exposure of the patient, with important benefits in terms of sustainability and safety.

In light of the clinical and scientific relevance of these developments, we decided to study in depth the Deep Learning models used to generate synthetic images from pairs of CT and MRI scans. The aim of this analysis is to contribute to the development of tools that can support radiotherapy dose planning with greater accuracy, improving the precision of the treatment and at the same time guaranteeing high safety standards for the patient. The use of synthetic CT scans (sCT) aims to reduce the dose of patient exposure to ionising radiation, offering further advantages in terms of safety. In addition, MRI can provide more detailed images of certain soft tissues than CT, potentially improving treatment planning. Finally, the aim is to reduce the time, costs and physical difficulties associated with performing two different scans, both for patients and healthcare personnel. Another field that benefits greatly from medical imaging is radiotherapy. This is a therapeutic technique based on the use of high-energy radiation, such as X-rays, aimed at destroying or slowing down the proliferation of tumour cells. It can be applied as a stand-alone treatment or combined with other therapies, such as chemotherapy or surgery. A careful planning of the dose, accompanied by constant monitoring during the treatment, is essential to maximise its therapeutic effectiveness and minimise damage to the surrounding healthy tissue. In fact, despite its usefulness, it can cause some side effects that vary depending on the dose of radiation supplied. Furthermore, several sessions are often required over the course of days or weeks, depending on the type and size of the tumour, requiring repeated CT scans.

However, the lack of data is still a limitation, especially in the field of biomedical data. The search for open-source datasets has shown that the area most subject to applications is the brain, and consequently the amount of data is greater than for other anatomical areas.

A state-of-the-art analysis has shown that many of the medical datasets used in the literature are private, making it more difficult to compare different approaches. However, what emerges is a prevalence of generative approaches that favour the cerebral region for training and prediction, as shown in Figure 7.1

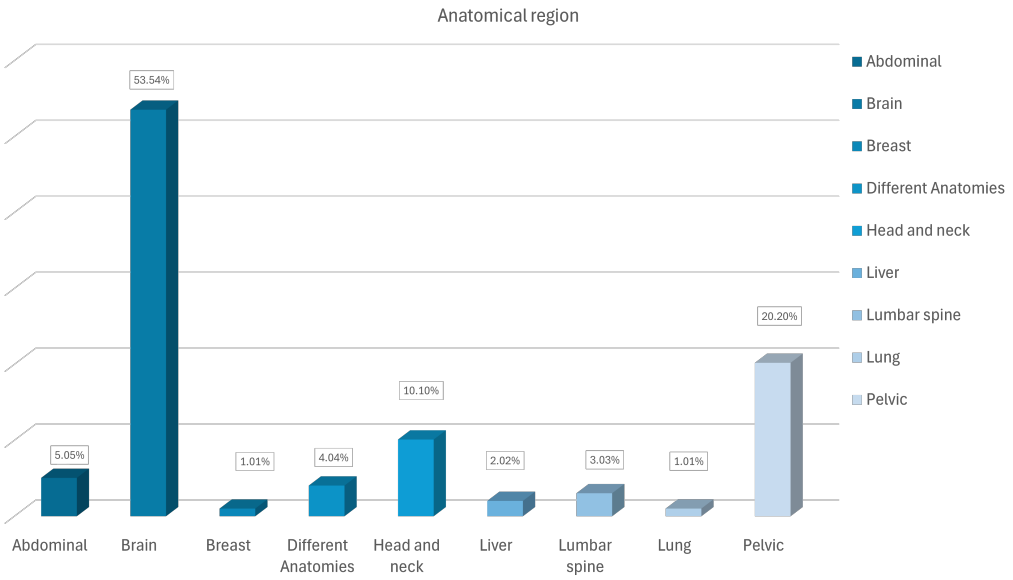


Figure 7.1: barplot chart showing the anatomical regions used in literature.

7.1 Adopted Models

The dataset used in this study is CERMEP-IDB-MRXFDG, described in detail in Chapter 3. In order to reduce the overall number of samples and allow the model to focus exclusively on slices containing relevant information, slices devoid of informative content, typically present at the beginning and end of three-dimensional acquisitions, were eliminated. The 3D volumes were then divided into slices, treating the problem from a two-dimensional (2D) perspective. Data leakage [156], i.e. contamination between the training and test sets with images that have temporal or spatial correlations (as in the case of video sequences), was prevented by avoiding a random split. For this purpose, the dataset was divided by extracting a test set composed of 4 samples, equal to about 10% of the total of 37 patients, to be used exclusively outside the K-Fold validation procedure. The first model used is Pix2Pix [157], a neural network belonging to the GAN family. This is characterised by a generator based on the U-Net architecture, exploiting its ability to maintain consistent information. The discriminator, on the other hand, exploits the PatchGAN network that divides the image, real or synthetic, into sub-regions (patches) to which it assigns a probability of authenticity; the scores obtained in this phase are then averaged to obtain an overall evaluation of the image. To speed up the process and reduce the number of parameters, the discriminator performs the convolution only with the patches; this local approach guarantees a high level of detail. Figure 7.2 shows the architecture.

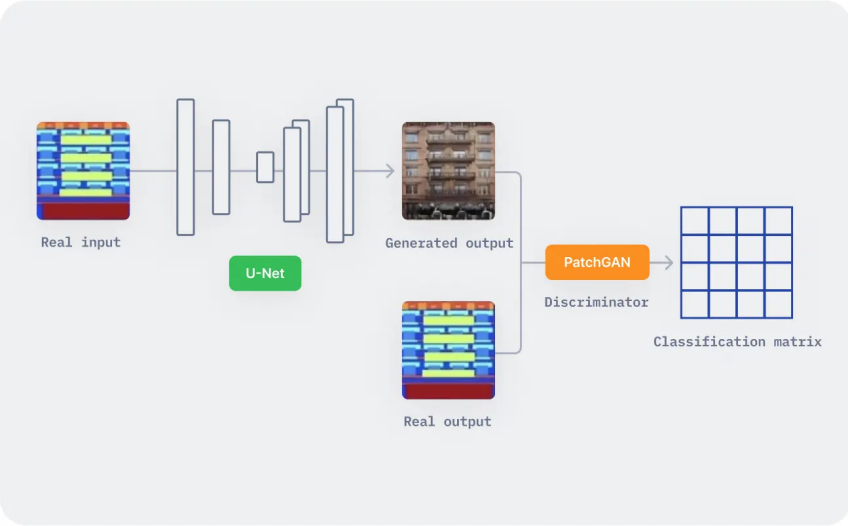


Figure 7.2: Pix2Pix architecture https://www.v7labs.com/blog/generative-adversarial-networks-guide

The second model utilized is defined in [158], which provides an innovative approach for generating computed tomography (CT) images from magnetic resonance imaging (MRI) without the necessity for paired data. This is especially useful in clinical contexts where obtaining aligned examinations is difficult or costly; the method under consideration overcomes this constraint, and it also takes advantage of the cyclic consistency characteristic of CycleGANs to force the network to learn a mapping between domains rather than over images.The model is trained using a loss function, which includes a term that quantifies the difference between the anatomical contours in the input MR and the synthetic CT, ensuring the generation of anatomically coherent structures. Furthermore, it uses segmentation masks of anatomical structures, if readily available to bind the generator and maintain them. The Figure 7.3, shows and highlights the distinct parts specified above.

Finally, the third model employed is SynDiff, a hybrid architecture that integrates diffusion-based generative mechanisms with nondiffusion deep learning components. Diffusion models, recently introduced in the field of medical image synthesis, are known for their ability to generate realistic, high-quality samples by iteratively refining noise into structured content. However, their high computational cost and the difficulty of explicitly controlling the generated structures often limit their stand-alone applicability.

SynDiff addresses this problem by combining the expressive generation capabilities of diffusion models with the domain-specific structure preservation

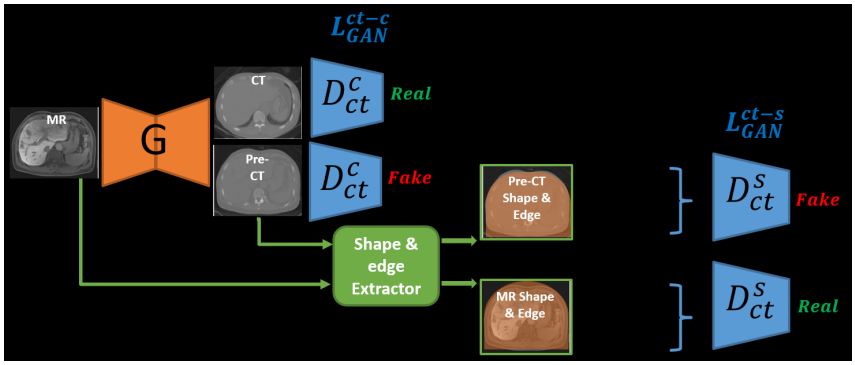


Figure 7.3: CycleGAN+UNet architecture, the generative part produces new images that are refined in morphological aspects by the convolutional network.

provided by nondiffusion models (e.g., U-Net modules). In the context of this thesis, SynDiff has been used to synthesize cross-modal medical images (e.g., PET from MRI) in an unsupervised context where paired data from different examinations are not always available.

7.2 Results

In this section, the results obtained on the test set for each fold are reported in table 7.1, comparing the various architectures.

The comparative analysis shown in the table reveals a clear distinction between the performance of the three models under consideration. Specifically, SynDiff demonstrates the best overall performance. The Pix2Pix and CycleGAN-UNet models show almost identical performance in terms of numerical error metrics, such as RMSE and MSE, suggesting a strong similarity in the results generated. However, it is important to note that Pix2Pix has a slight higher SSIM score, showing better local image structure preservation, even though the overall result is really low. SynDiff has significantly lower MSE and RMSE values than the other two models, suggesting a lower average deviation from the target image, even though ME has strongly negative values. The PSNR is also significantly higher, indicating higher reconstruction quality in terms of signal-to-noise ratio. An example of the quality of the images generated by the three architectures is shown in Figure 7.4.

As can be seen from the figure, the CTs generated by SynDiff and Pix2Pix appear qualitatively acceptable, while the architecture based on CycleGan + Unet appears hallucinatory during the generation phase. The proposed results are not satisfactory in terms of purely numerical evaluation. This could be motivated by the high difficulty of the task, mainly due to cross-domain

Fold	Architecture	ME	MAE	MSE	RMSE	PSNR	SSIM
fold1	Cycle GAN + Unet	1066.9203	1066.9203	1474882.4281	1212.8088	34.6662	0.0008
fold1	SynDiff	-345.6443	496.7377	427681.403	650.5487	40.1068	0.1242
fold1	Pix2Pix	1058.2399533	1058.239953	1460600.20697204	1206.6720	34.7121	0.0007
fold2	Cycle GAN + Unet	1066.9203	1066.9203	1474882.4296	1212.8088	34.6662	0.0008
fold2	SynDiff	-345.8916	497.5934	427501.5513	650.4576	40.1074	0.1195
fold2	Pix2Pix	1058.2237	1058.2237	1460557.8586	1206.6550	34.7122	0.0007
fold3	Cycle GAN + Unet	1066.9203	1066.9203	1474882.3776	1212.8088	34.6662	0.0008
fold3	SynDiff	-345.6407	494.7006	425473.5586	648.9066	40.1283	0.1247
fold3	Pix2Pix	1058.2191	1058.2191	1460560.943	1206.6562	34.7122	0.0007
fold4	Cycle GAN + Unet	1066.9203	1066.9203	1474882.4278	1212.8088	34.6662	0.0008
fold4	SynDiff	-345.6423	495.7491	426315.8	649.5997	40.1183	0.1244
fold4	Pix2Pix	1058.2397	1058.2397	1460640.368	1206.6892	34.7120	0.0007
fold5	Cycle GAN + Unet	1066.9203	1066.9203	1474882.4293	1212.8088	34.6662	0.0008
fold5	SynDiff	-345.6454	496.6046	427575.0488	650.533	40.1061	0.1366
fold5	Pix2Pix	1058.2269	1058.2269	1460565.691	1206.6583	34.7122	0.0007

Table 7.1: Quantitative results reporting the values for the chosen metrics obtained from the 3 architectures at each fold.

generation.

Furthermore, although the models manage to replicate the anatomical structures and are consistent, the real problem lies in the difficulty in perfectly learning the distribution of grey values within the image. The latter are often far from the true value, hence the high value in the metrics evaluating the difference at the pixel level. However, in evaluations of clinically generated images, they are essential in order to be able to accurately determine tissue density.

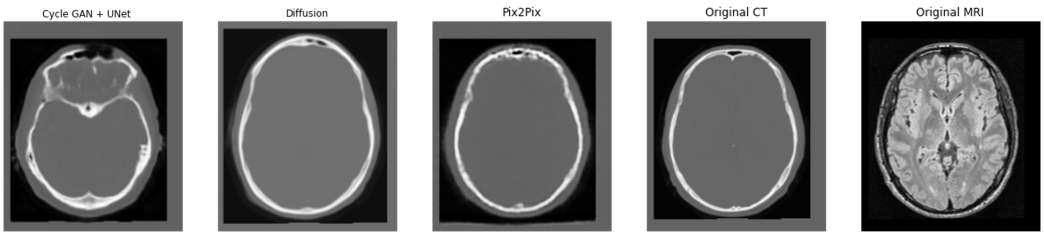


Figure 7.4: Example of images generated by the three tested models. From left to right are the MRI provided as input, the ground truth CT, and the results obtained by Pix2Pix, SynDiff, and CycleGan-Unet.

Chapter 8

Few Shot Segmentation

As already introduced in the Chapter 1, the performance of deep learning models is highly dependent on the quantity and quality of the available data. This is particularly relevant in semantic segmentation tasks, where the model has to make decisions at the level of a single pixel and, thus, requires a very strong discriminative capacity. However, in fields such as the biomedical sector, the availability of annotated data is often extremely limited. The main causes are the shortage of experts for annotation, the variability in acquisition protocols, and the difficulty in accessing data on rare diseases. To address this challenge, the Few-Shot Segmentation (FSS) paradigm has been proposed in recent years. This approach makes it possible to train models capable of generalising to new, unseen classes using only a few annotated examples. The goal is no longer to learn a segmentation for every possible class, but to learn how to segment: that is, to learn a rapid adaptation mechanism that uses a few images as support to generate the mask on a new query image.

Formally, the problem can be defined as the prediction of a mask $\hat{M}_q$ for a subject in class c in a query image I_q given a support set S that contains k image-mask pairs for that class. It can be summarised by equation 8.1.

$$S_c = \{(I_c^i, M_c^i)\}_{i=1}^k \tag{8.1}$$

The final objective of FSS training according to the above paradigm can be formalised as in the equation 8.2.

$$\hat{M}_q = f_\theta(S_c, I_q) \tag{8.2}$$

In this scenario, the number of k is fundamental because it indicates the number of samples available for a class in an iteration. To compensate for the lack of data, techniques such as *episodic learning* or *transfer learning* can be

used to optimise the models' ability to satisfy new segmentation requests.

In this chapter, we will describe in detail the implemented few-shot learning method.

8.1 Proposed Architecture

In the proposed implementation, benchmark datasets for few shots were used. Among these is FSS-1000, used for the creation of tasks in an episodic learning approach based on meta-training; as already described in section 2.4, this type of FSS proposes to use a pre-trained model to segment new classes using a limited number of samples. For the meta-testing phase, *ISIC-2018*, *DeepGlobe*, *Chest X-Ray* were used; these datasets were selected to intensively evaluate the cross-domain performance of the proposed algorithm.

The choice of the Seq2Seq architecturem shown in Figure 8.1, for segmentation arises from the desire to treat the task as a structural translation problem, where the input (query image) is interpreted on the basis of a context (support image) in a similar way as in translation between languages. This approach allows the use of attention forms to explicitly relate salient regions between support and query.

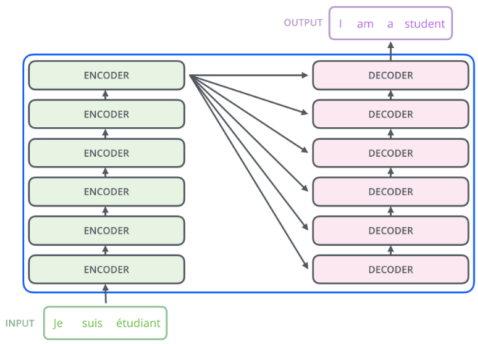


Figure 8.1: Sequence to Sequence architecture

The starting architecture includes an encoder based on a ResNet-50 backbone pre-trained on ImageNet1K, which is used for feature extraction from support and query images. Furthermore, the choice of ResNet-50 as a backbone is motivated by its proven ability to generalise to different domains and its compatibility with the few-shot training paradigm via multi-level feature extraction. The encoder, according to the FSS paradigm, will be used to process first the Support image and then the Query image. This phase allows information to be obtained from the two different images. Each feature map

obtained as output for the support and query images during the feed-forward phase is stored in two separate lists: F_l for the support image and Q_l for the query image, in order to keep this information extracted from the different layers. In this way, only the last element of the query is selected, denoted as $Q_N = Q_l[-1]$, which will be used to iteratively calculate the similarity with the other feature maps (stored in F_l). During each iteration, the masked average pooling is calculated as follows

$$masked\_features_i = \frac{F_i * Mask}{k} \tag{8.3}$$

Once the encoding phase is completed, the second region of the architecture, used for decoding, applies a self-attention operation to the Q_N element identified at the i-th iteration, according to the formula 8.4.

$$Attention(q, k, v) = softmax(\frac{q * k^T}{\sqrt{d_k}}) \tag{8.4}$$

The output produced using this method is normalised and concatenated with the original vector, to produce an enriched representation. This vector is then used as a query within a cross-attention mechanism, together with the features

$$\{F_i\}_{i=1}^{L}$$

of the support. The use of attention-based strategies to fuse information between support and query was preferred to simple chaining, as it allows the model to assign different weights to the various regions of the image, facilitating the identification and improved learning of the most relevant areas for the target class. The aim of these operations is to emphasise the most relevant components of the query, in order to identify significant correspondences with the support, isolating those that are most distinctive for the target class.

The process is then repeated iteratively on all the various levels of the support, from the output of the last level to the input of the first: in this way, the information is navigated, starting from more abstract semantic representations and progressively moving towards more detailed spatial information. This strategy allows for more precise segmentation masks, combining both semantics and low-level information. The decoding operations described above can be formalised as follows and can be represented as described in Figure 8.2.

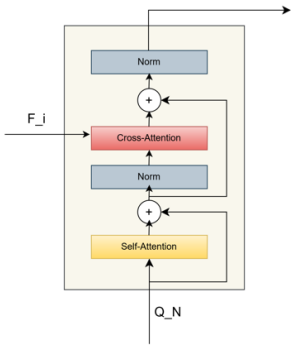


Figure 8.2: Decoder Block used in the architecture

$$
\begin{aligned}
Q_{AN} &= Norm(\text{Self-Attention}(Q_N)) \\
Q_{AN} &= Q_N + Q_{AN} \\
Output &= Norm(\text{Cross-Attention}(Q_{AN}, F_i) \\
Output &= Q_{AN} + Output
\end{aligned}
\tag{8.5}
$$

Furthermore, convolutions or upsampling operations are carried out inside the decoders to maintain consistency in dimensions. An example of the decoding block is shown in Figure 8.2.

At the end of the decoder chain, a classic convolutional block is used to obtain the desired size and number of channels.

Figure 8.3 shows the entire FSS architecture developed and used in the proposed research activity.

8.2 Results

The results obtained from the model are reported in this section. The architecture has proven to perform well on tasks where the objects to be segmented fall within well-defined and consistent image regions. Specifically, table 8.1 shows the Dice Score calculated on the ISIC, Chest X-ray and DeepGlobe datasets using a support sample number of $k = 5$.

As can be seen, the results do not reach values comparable to the classic segmentation approaches but despite this evidence, the qualitative analysis of

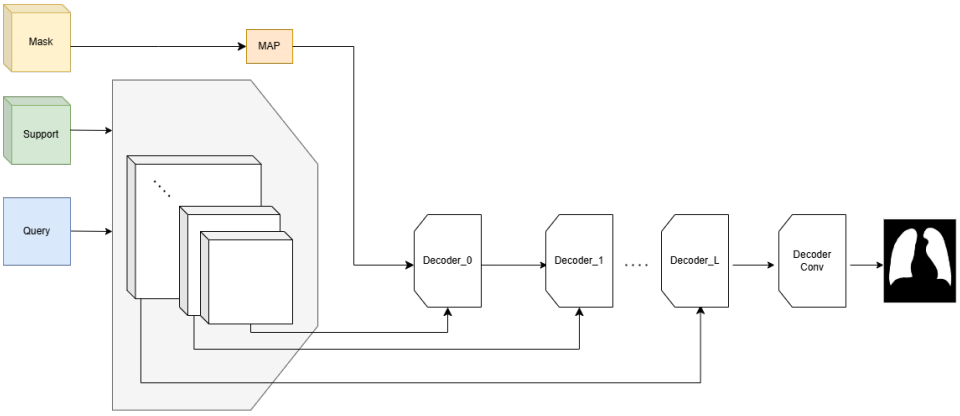


Figure 8.3: overview of the proposed architecture

Dataset	Avg_DICE	k
Chest-Xray	0.5753	5
ISIC18	0.4072	5
DeepGlobe	0.07283	5

Table 8.1: Table shows the results obtained on the various datasets, considering k=5 and meta-training on FSS-100 (760 classes).

the generated segmentation masks is still promising. Actually, as previously mentioned, if the region to be segmented remains consistent between the various supporting images, as in the case of the Chest x-ray dataset (see Figure 8.4), the segmentation is correct.

In contrast, as observed on DeepGlobe, when feature-rich images are contained in the dataset where the distribution of objects is not consistent between the various images, the limits of the proposed approach appear. Figure 8.5 shows an example of the mask predicted on DeepGlobe.

The results obtained reveal the preliminary nature of the presented architecture, however not yet comparable to the state of the art, while offering optimism for its future evolution. One of the potential structural improvements might be the use of FSS paradigms based on many prototypes. These would enable more accurate local discrimination of the various attributes for each instance inside structurally harder images. This focus on selectivity would favor a more granular study, simplifying self-attention and cross-attention procedures while also overcoming previous limitations. The evolution will undoubtedly focus on cross-feature extraction in order to create an embedding to identify

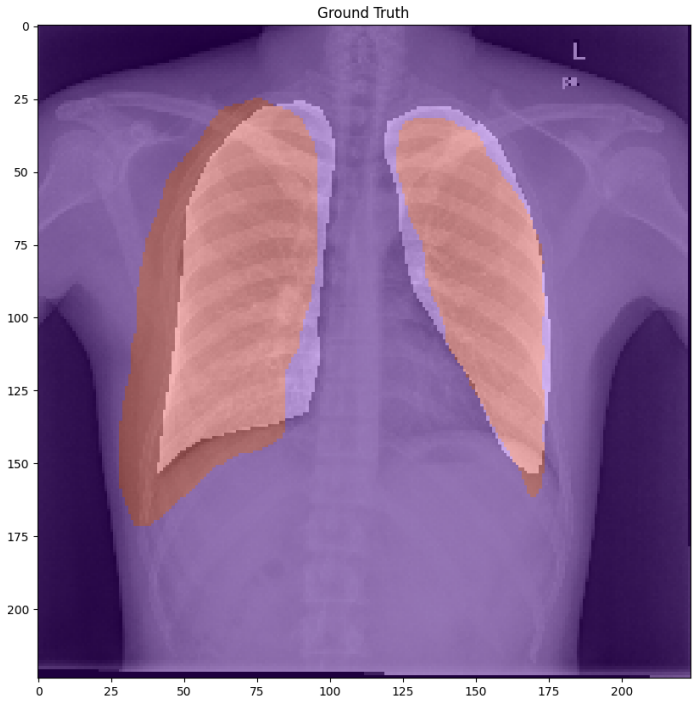


Figure 8.4: An example of prediction over Chest X-Ray dataset.

matches between the support and query sets. The problem remains under investigation, and researchers are testing new backbone architectures to see if approaches based on backbone Transformers or foundation models are able to produce deeper relationships.

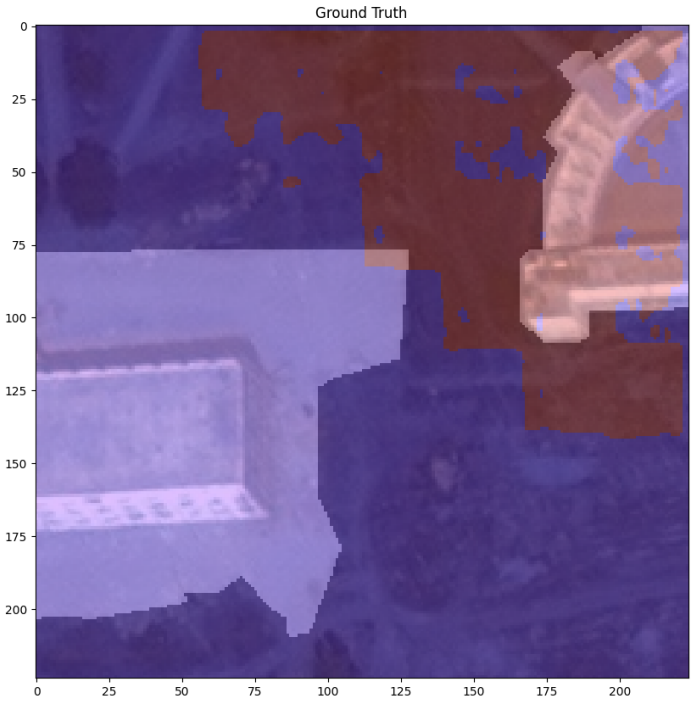


Figure 8.5: An example of prediction over DeepGlobe dataset

Chapter 9

Conclusion

Artificial intelligence applications are becoming progressively more relevant today and, thanks to the emergent LLMs, they are increasingly being used. This accessibility and availability to everyone is causing an uncontrolled growth in the resources needed to sustain their functionality. This issue is becoming increasingly concrete and has motivated my research activity during this three-year period. In fact, I have focused my work on designing and implementing lightweight models that combine predictive effectiveness, computational efficiency and energy sustainability, with a particular interest in practical applications within the biomedical domain. The research has shown that classic training techniques can be replaced or reformulated to maintain the desired performance. In fact, by using optimisation techniques, the training or inference phases can be lightened, leading to savings in terms of resources used. This results in lower energy consumption and reduced emissions into the atmosphere.

In fact, as discussed in the chapter on Learn&Drop and the development of Few-parameter Architecture (FPA), it has been demonstrated that in classification and segmentation tasks, the results obtained on the datasets tested respect the minimum performance requirements, despite the reduction in trainable parameters. This result is promising, especially considering that the FPA architecture has fewer than $150k$ parameters. In addition, lighter models, when properly optimised, can achieve competitive performance compared to more complex solutions, while maintaining greater computational efficiency. This makes such models particularly suitable for integration in real clinical contexts, including those with limited resources, or in edge/point-of-care scenarios, where fast processing and low energy consumption are key elements.

As already mentioned, the research activity has been focused on the medical domain, addressing the issues that such a "sensitive" field presents. In

this context, the development of Artificial Intelligence systems that integrate directly within the DICOM Standard was the first topic addressed. IODeep is the data structure that integrates Deep Learning algorithms into Clients PACS in a compliant way, enabling the development of automated diagnostic solutions that support doctors in the patient analysis and evaluation phase. This result is a major step forward for secure data processing, ensuring privacy for cutting-edge precision medicine applications. Continuing in the medical field, during my research at the Ri.MED foundation, my work addressed the issue of data scarcity. In fact, in the medical field, due to the privacy issues mentioned above, labelled datasets (both for classification and segmentation) are lacking. To overcome this problem, the issue of cross-domain image generation was addressed. Specifically, three state-of-the-art deep neural architectures (Pix2Pix, CycleGan-Unet and SynDiff) were implemented with the aim of generating sCT from MRI scans. Cross-domain generation is one of the most complex tasks, especially in a domain where the fidelity of the generated images must be absolute. The results obtained are not optimal and certainly do not achieve the desired outcome. Despite this, the use of diffusion models, such as SynDiff, is the best and certainly the most viable way to generate sCT effectively and reliably, reducing patients' exposure to ionising radiation. Finally, during the research period at Keele University, the problem of labelled data scarcity was addressed by changing point of view. During this period, the development of Few-shot Segmentation (FSS) architecture was the main focus. Using meta-learning paradigms, an encoder-decoder architecture was created that combined the characteristics of this type of learning to correctly solve segmentation tasks. The results obtained from the experimental phase show that the proposed architecture, still in its preliminary version, and on 5-shot segmentation tasks on feature-rich image datasets, does not perform very well, as in the case of DeepGlobe.

Main Contributions and Results The thesis addressed various issues in the field of artificial intelligence applied to the biomedical context, with a focus on computational efficiency and sustainability. The main contributions are summarised below:

- **Learn&Drop and Few-Parameter Architecture (FPA)**, Optimisation techniques and architectural solutions have been proposed that allow a significant reduction in the number of parameters in deep learning models. In particular, the FPA (Few-Parameter Architecture) is de-

signed to maintain competitive performance while having less than 150k parameters. This allows these models to be used in resource-limited environments, such as edge devices or peripheral clinical settings, while reducing energy consumption and environmental impact.

- **IODeep and Integration into the DICOM Standard**, IODeep, a DICOM-compliant data structure designed to integrate deep learning algorithms within hospital PACS servers, was developed. This solution enables the development of automated diagnostic support systems that operate locally, preserving patient privacy and respecting the security and compliance constraints of the hospital infrastructure.

- **Cross-domain Image Generation (sCT from MRI)**, comparing state-of-the-art generative models, despite the produced images seems good but not enough to be used in real medical application, but will contributing to reduced to the patients ionic radiation.

- **Few-shot Segmentation (FSS)**, A segmentation framework based on meta-learning was developed, designed to operate in contexts with scarce labelled data. The proposed encoder-decoder architecture is still at a preliminary stage, but results on complex datasets (such as DeepGlobe) indicate that, while requiring improvement, the approach is valid and promising for low-data scenarios.

Future Directions Based on the obtained data, numerous prospective enhancements that could reinforce and expand the proposed work have arisen. For example:

- *Improvement of the FPA architecture*: with the aim of obtaining more precise ROI masks that can also be used as a backbone for more complex or multitask models.

- *Expansion of the functionality of IODeep*: including the possibility of fine-tuning or training directly within the hospital infrastructure, thus ensuring greater adaptability to local data and autonomous model management.

- *Optimisation of the FSS architecture*: in order to resolve the current limitations that emerged in experimental tests, improving generalisation to complex datasets through the introduction of attention mechanisms or transformer-based components.

- *Development of hybrid pipelines for cross-domain generation*: combining supervised and unsupervised approaches with diffusion models, to increase the morphological fidelity and semantic coherence of synthetic images.
- *Quantitative analysis of the energy impact*: evaluating the computational effort of the various proposed architectures, in order to guide future design choices in a sustainable AI perspective.

Bibliography

[1] Y. Yu, J. Wang, Y. Liu, P. Yu, D. Wang, P. Zheng, and M. Zhang, "Revisit the environmental impact of artificial intelligence: the overlooked carbon emission source?" *Frontiers of Environmental Science & Engineering*, vol. 18, no. 12, pp. 1–5, 2024.

[2] A. De Vries, "The growing energy footprint of artificial intelligence," *Joule*, vol. 7, no. 10, pp. 2191–2194, 2023.

[3] S. Jung, H. Heo, S. Park, S.-U. Jung, and K. Lee, "Benchmarking deep learning models for instance segmentation," *Applied Sciences*, vol. 12, no. 1717, p. 8856, Jan. 2022.

[4] M. Boulanger *et al.*, "Deep learning methods to generate synthetic ct from mri in radiotherapy: A literature review," *Physica Medica*, vol. 89, pp. 265–281, September 2021.

[5] E. Schreibmann, J. A. Nye, D. M. Schuster, D. R. Martin, J. Votaw, and T. Fox, "Mr-based attenuation correction for hybrid pet-mr brain imaging systems using deformable image registration," *Medical physics*, vol. 37, no. 5, pp. 2101–2109, 2010.

[6] Y. Wang, C. Liu, X. Zhang, and W. Deng, "Synthetic ct generation based on t2 weighted mri of nasopharyngeal carcinoma (npc) using a deep convolutional neural network (dcnn)," *Frontiers in Oncology*, vol. 9, November 2019.

[7] I. Lab, "icon-lab/syndiff," GitHub, 2024, accessed: 11 June 2024. [Online]. Available: https://github.com/icon-lab/SynDiff

[8] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Technical Report*, pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[9] I. Mérida *et al.*, "Cermep-idb-mrxfdg: a database of 37 normal adult human brain [18f]fdg pet, t1 and flair mri, and ct images available for research," *EJNMMI Research*, vol. 11, no. 1, p. 91, September 2021.

[10] G. Cruciata, L. Cruciata, L. Lo Presti, J. van Gemert, and M. La Cascia, "Learn & drop: fast learning of cnns based on layer dropping," *Neural Computing and Applications*, vol. 36, no. 18, pp. 10 839–10 851, 2024.

[11] M. Lin, Q. Chen, and S. Yan, "Network in network," no. arXiv:1312.4400, Mar 2014, arXiv:1312.4400 [cs]. [Online]. Available: http://arxiv.org/abs/1312.4400

[12] S. Contino, L. Cruciata, O. Gambino, and R. Pirrone, "Iodeep: An iod for the introduction of deep learning in the dicom standard," *Computer Methods and Programs in Biomedicine*, vol. 248, p. 108113, May 2024.

[13] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Commun. ACM*, vol. 63, no. 12, p. 54–63, Nov. 2020. [Online]. Available: https://doi.org/10.1145/3381831

[14] V. Bolón-Canedo, L. Morán-Fernández, B. Cancela, and A. Alonso-Betanzos, "A review of green artificial intelligence: Towards a more sustainable future," *Neurocomputing*, p. 128096, 2024.

[15] D. Patterson, J. Gonzalez, U. Hölzle, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. R. So, M. Texier, and J. Dean, "The carbon footprint of machine learning training will plateau, then shrink," *Computer*, vol. 55, no. 7, pp. 18–28, 2022.

[16] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou,

S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," no. arXiv:2501.12948, Jan. 2025, arXiv:2501.12948 [cs]. [Online]. Available: http://arxiv.org/abs/2501.12948

[17] M. Dehghani, A. Arnab, L. Beyer, A. Vaswani, and Y. Tay, "The efficiency misnomer," no. arXiv:2110.12894, Mar. 2022, arXiv:2110.12894 [cs]. [Online]. Available: http://arxiv.org/abs/2110.12894

[18] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," no. arXiv:2205.14135, Jun. 2022, arXiv:2205.14135 [cs]. [Online]. Available: http://arxiv.org/abs/2205.14135

[19] A. C. Penedo, "The Regulation of Data Spaces under the EU Data Strategy: Towards the 'Act-ification' of the Fifth European Freedom for Data?" *European Journal of Law and Technology*, vol. 15, no. 1, May 2024. [Online]. Available: https://www.ejlt.org/index.php/ejlt/article/view/995

[20] P. Terzis and E. O. S. Echeverria, "Interoperability and governance in the European Health Data Space regulation," *Medical Law International*, Apr. 2023. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/09685332231165692

[21] G. Csurka, R. Volpi, and B. Chidlovskii, "Semantic image segmentation: Two decades of research," *arXiv preprint arXiv:2302.06378*, 2023. [Online]. Available: https://arxiv.org/abs/2302.06378

[22] M. Mazhar, S. Fakhar, and Y. Rehman, "Semantic segmentation for various applications: Research contribution and comprehensive review," *Computer Sciences and Mathematics Forum*, vol. 2, no. 1, p. 21, 2022. [Online]. Available: https://www.mdpi.com/2673-4591/32/1/21

[23] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: state of the art," *International journal of multimedia information retrieval*, vol. 9, no. 3, pp. 171–189, 2020.

[24] D. Zhang, Y. Song, D. Liu, H. Jia, S. Liu, Y. Xia, H. Huang, and W. Cai, "Panoptic segmentation with an end-to-end cell r-cnn for pathology image analysis," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*. Springer, 2018, pp. 237–244.

[25] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," *arXiv preprint arXiv:1801.00868*, 2019. [Online]. Available: https://arxiv.org/abs/1801.00868

[26] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, "Upsnet: A unified panoptic segmentation network," *arXiv preprint arXiv:1901.03784*, 2019. [Online]. Available: https://arxiv.org/abs/1901.03784

[27] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[28] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[29] I. Sobel, "An isotropic 3x3 image gradient operator," *Machine Vision for Three-Dimensional Scenes*, pp. 376–379, 1990.

[30] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[31] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection," *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation*, pp. 2.1–2.12, 1979.

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[34] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "U-net++: A nested u-net architecture for medical image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 839–848.

[35] N. Hirose, A. Sadeghian, F. Xia, R. Martín-Martín, and S. Savarese, "Vunet: Dynamic scene view synthesis for traversability estimation using an rgb camera," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2062–2069, 2019.

[36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," Jun 2021, arXiv:2010.11929 [cs]. [Online]. Available: http://arxiv.org/abs/2010.11929

[39] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[40] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in neural information processing systems*, vol. 34, pp. 12 077–12 090, 2021.

[41] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[42] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[43] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324.

[44] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2820–2828.

[45] B. Wu, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, and K. Keutzer, "FBNet: Hardware-aware efficient convnet design via differentiable neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 734–10 742.

[46] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," in *International Conference on Learning Representations (ICLR)*, 2019.

[47] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, vol. 97, 2019, pp. 6105–6114.

[48] M. Li, M. Shen, K. Han, Y. Wang, and C. Xu, "MicroNet: Towards image recognition with extremely low flops," *arXiv preprint arXiv:2011.12289*, 2020.

[49] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2022, iCLR 2022.

[50] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[51] T. Zhou, S. Canu, P. Vera, and S. Ruan, "Latent correlation representation learning for brain tumor segmentation with missing mri modalities," *IEEE Transactions on Image Processing*, vol. 30, pp. 4263–4274, 2021.

[52] H. Emami, M. Dong, S. Nejad-Davarani, and C. Glide-Hurst, "Generating synthetic cts from magnetic resonance images using generative adversarial networks," *Medical Physics*, June 2018.

[53] J. Jiang *et al.*, "Cross-modality (ct-mri) prior augmented deep learning for robust lung tumor segmentation from small mr datasets," *Medical Physics*, vol. 46, no. 10, pp. 4392–4404, 2019.

[54] X. Han, "Mr-based synthetic ct generation using a deep convolutional neural network method," *Medical Physics*, vol. 44, no. 4, pp. 1408–1419, April 2017.

[55] J. Wong, A. Filippi, M. Scorsetti, S. Hui, L. Muren, and P. Mancosu, "Total marrow and total lymphoid irradiation in bone marrow transplantation for acute leukaemia," *The Lancet Oncology*, vol. 21, no. 10, pp. e477–e487, October 2020.

[56] A. Ben-Cohen, E. Klang, S. Raskin, M. Amitai, and H. Greenspan, "Virtual pet images from ct data using deep convolutional networks: Initial results," July 2017.

[57] X. Dong *et al.*, "Synthetic ct generation from non-attenuation corrected pet images for whole-body pet imaging," *Physics in Medicine and Biology*, vol. 64, November 2019.

[58] S.-H. Hsu, Y. Cao, K. Huang, M. Feng, and J. Balter, "Investigation of a method for generating synthetic ct models from mri scans of the head and neck for radiation therapy," *Physics in Medicine and Biology*, vol. 58, no. 23, pp. 8419–8435, December 2013.

[59] C.-B. Jin *et al.*, "Deep ct to mr synthesis using paired and unpaired data," *Sensors*, vol. 19, no. 10, January 2019.

[60] A. C. Evans, D. L. Collins, S. Mills, E. D. Brown, R. L. Kelly, and T. M. Peters, "3d statistical neuroanatomical models from 305 mri volumes," in *1993 IEEE conference record nuclear science symposium and medical imaging conference*. IEEE, 1993, pp. 1813–1817.

[61] D. Nie, X. Cao, Y. Gao, L. Wang, and D. Shen, "Estimating ct image from mri data using 3d fully convolutional networks," in *Deep Learning and Data Labeling for Medical Applications*, 2016, pp. 170–178.

[62] L. Xiang, Y. Qiao, D. Nie, L. An, Q. Wang, and D. Shen, "Deep auto-context convolutional neural networks for standard-dose pet image estimation from low-dose pet/mri," *Neurocomputing*, vol. 267, pp. 406–416, December 2017.

[63] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241.

[64] H. Jang, F. Liu, G. Zhao, T. Bradshaw, and A. McMillan, "Technical note: Deep learning based mrac using rapid ultrashort echo time imaging," *Medical Physics*, vol. 45, no. 8, pp. 3697–3704, August 2018.

[65] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1529–1537.

[66] L. Xiang *et al.*, "Deep embedding convolutional neural network for synthesizing ct image from t1-weighted mr image," *Medical Image Analysis*, vol. 47, pp. 31–44, July 2018.

[67] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, December 2017.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

[69] D. Nie *et al.*, "Medical image synthesis with context-aware generative adversarial networks," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2017*, 2017, pp. 417–425.

[70] X. Li, Y. Jiang, J. Rodriguez-Andina, H. Luo, S. Yin, and O. Kaynak, "When medical images meet generative adversarial network: recent development and research opportunities," *Discover Artificial Intelligence*, vol. 1, no. 1, p. 5, September 2021.

[71] J.-Y. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.html

[72] B. Zhao *et al.*, "Ct synthesis from mr in the pelvic area using residual transformer conditional gan," *Computerized Medical Imaging and Graphics*, vol. 103, p. 102150, January 2023.

[73] O. Dalmaz, M. Yurt, and T. Çukur, "Resvit: Residual vision transformers for multimodal medical image synthesis," *IEEE Transactions on Medical Imaging*, vol. 41, no. 10, pp. 2598–2614, October 2022.

[74] P. Zeng *et al.*, "3d cvt-gan: A 3d convolutional vision transformer-gan for pet reconstruction," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, 2022, pp. 516–526.

[75] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv*, December 2020.

[76] R. Rombach, D. Blattmann, P. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," *arXiv*, April 2022.

[77] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv*, July 2022.

[78] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Advances in Neural Information Processing Systems*, 2021, pp. 8780–8794. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html

[79] Q. Lyu and G. Wang, "Conversion between ct and mri images using diffusion and score-matching models," *arXiv*, September 2022.

[80] M. Özbey *et al.*, "Unsupervised medical image translation with adversarial diffusion models," *IEEE Transactions on Medical Imaging*, vol. 42, no. 12, pp. 3524–3539, December 2023.

[81] X. Meng *et al.*, "A novel unified conditional score-based generative framework for multi-modal medical image completion," *arXiv*, July 2022.

[82] L. Zhu *et al.*, "Make-a-volume: Leveraging latent diffusion models for cross-modality 3d brain mri synthesis," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, 2023, pp. 592–601.

[83] S. Pan *et al.*, "Synthetic ct generation from mri using 3d transformer-based denoising diffusion model," *Medical Physics*, vol. 51, no. 4, pp. 2538–2548, 2024.

[84] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.

[85] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *CoRR*, vol. abs/1703.05175, 2017. [Online]. Available: http://arxiv.org/abs/1703.05175

[86] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5217–5226.

[87] X. Wang, S. You, X. Li, and H. Xu, "Panet: Few-shot image segmentation with prototype alignment," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

[88] X. Zhang, Y. Wei, Y. Yang, and C. Huang, "Sg-one: Similarity guidance network for one-shot semantic segmentation," in *IEEE Transactions on Cybernetics*, vol. 50, no. 9, 2020, pp. 3855–3865.

[89] X. Li, H. Zhang, K. Zhang, S. He, L. Wang, M. R. Lyu, and I. King, "Adaptive prototype learning and allocation for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8334–8343.

[90] S. Gairola, M. Hemani, A. Chopra, and B. Krishnamurthy, "Simpropnet: Improved similarity propagation for few-shot image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

[91] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9587–9595.

[92] C. Zhang, X. Wei, Y. Yang, and C. Huang, "Cycle-consistent transformer for few-shot segmentation," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2021.

[93] J.-S. Min, S. Hong, M. Cho, and B. Han, "Hypercorrelation squeeze for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6941–6952.

[94] W. Tang, B. Yang, P.-A. Heng, and C.-W. Fu, "Overcoming support dilution for robust few-shot semantic segmentation," *arXiv preprint arXiv:2303.15583*, 2023.

[95] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-Shot Learning for Semantic Segmentation," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[96] M. Siam, B. Oreshkin, and M. Jagersand, "Adaptive masked proxies for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

[97] S. M. Hendryx, A. B. Leach, P. D. Hein, and C. T. Morrison, "Meta-learning initializations for image segmentation," *arXiv preprint arXiv:2010.08166*, 2020.

[98] M. Rußwurm, S. Wang, M. Korner, and D. Lobell, "Meta-learning for few-shot land cover classification," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops*, 2020, pp. 200–201.

[99] K. Zhu, W. Zhai, Z.-J. Zha, and Y. Cao, "Self-supervised tuning for few-shot segmentation," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, 2020, pp. 1072–1078.

[100] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. B. Ayed, and J. Dolz, "Repri: Few-shot segmentation via rich prototype refinement and interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2593–2602. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Boudiaf_RePRI_Few-Shot_Segmentation_via_Rich_Prototype_Refinement_and_Interpolation_CVPR_2021_paper.html

[101] M. Zięba, P. Semberecki, and T. Trzciński, "Continual few-shot learning for semantic segmentation," *ICANN*, 2021.

[102] Z. Tian, X. Lai, L. Jiang, S. Liu, and J. Jia, "Incremental few-shot semantic segmentation via embedding adaptive-updating and hyperbolic geometry," in *CVPR*, 2022.

[103] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, p. 336–359, Feb 2020, arXiv:1610.02391 [cs].

[104] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[105] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," *CoRR*, vol. abs/1702.05373, 2017. [Online]. Available: http://arxiv.org/abs/1702.05373

[106] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, p. 248–255. [Online]. Available: https://ieeexplore.ieee.org/document/5206848

[107] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[108] J. Howard, "Imagenette," https://github.com/fastai/imagenette/.

[109] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98 – 136, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:207252270

[110] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, p. 41, 2023.

[111] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti *et al.*, "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1902.03368*, 2019.

[112] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018.

[113] H. H. CN, Maggie, P. Culliton, P. Yadav, and S. L. Lee, "UW-Madison GI Tract Image Segmentation," https://kaggle.com/competitions/uw-madison-gi-tract-image-segmentation, 2022, accessed April 2025.

[114] A. Chou, W. Li, and E. Roman, "Gi tract image segmentation with u-net and mask r-cnn."

[115] M. Sharma, "Automated gi tract segmentation using deep learning," *arXiv preprint arXiv:2206.11048*, 2022.

[116] X. Li, T. Wei, Y. P. Chen, Y.-W. Tai, and C.-K. Tang, "FSS-1000: A 1000-class dataset for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

[117] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2018, pp. 172–17 209.

[118] G. Menghani, "Efficient deep learning: A survey on making deep learning models smaller, faster, and better," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–37, 2023.

[119] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 264–11 272.

[120] T. Choudhary, V. Mishra, A. Goswami, and J. Sarangapani, "Heuristic-based automatic pruning of deep neural networks," *Neural Computing and Applications*, vol. 34, no. 6, pp. 4889–4903, 2022.

[121] R. Zemouri, N. Omri, F. Fnaiech, N. Zerhouni, and N. Fnaiech, "A new growing pruning deep learning neural network algorithm (gp-dlnn)," *Neural Computing and Applications*, vol. 32, pp. 18 143–18 159, 2020.

[122] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1389–1397.

[123] S. Xu, H. Chen, X. Gong, K. Liu, J. Lü, and B. Zhang, "Efficient structured pruning based on deep feature stabilization," *Neural Computing and Applications*, vol. 33, no. 13, pp. 7409–7420, 2021.

[124] X. Xiao, T. B. Mudiyanselage, C. Ji, J. Hu, and Y. Pan, "Fast deep learning training through intelligently freezing layers," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing*

and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, 2019, pp. 1225–1232.

[125] S. Liu, I. Ni'mah, V. Menkovski, D. C. Mocanu, and M. Pechenizkiy, "Efficient and effective training of sparse recurrent neural networks," *Neural Computing and Applications*, vol. 33, pp. 9625–9636, 2021.

[126] J. Zhang, X. Chen, M. Song, and T. Li, "Eager pruning: Algorithm and architecture support for fast training of deep neural networks," in *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2019, pp. 292–303.

[127] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[128] G. Van Rossum, *The Python Library Reference, release 3.8.2*. Python Software Foundation, 2020, https://github.com/python/cpython/blob/3.11/Lib/pickle.py.

[129] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, https://numpy.org/. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[130] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035, https://pytorch.org/.

[131] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015.

[132] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[133] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," no. arXiv:1502.03167, Mar 2015, arXiv:1502.03167 [cs]. [Online]. Available: http://arxiv.org/abs/1502.03167

[134] J. Yang, R. Shi, and B. Ni, "Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis," in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, Apr 2021, p. 191–195, arXiv:2010.14925 [cs]. [Online]. Available: http://arxiv.org/abs/2010.14925

[135] H. Huang, *Pacs-based multimedia imaging informatics: Basic principles and applications*. John Wiley & Sons, 2019.

[136] T. Benson and G. Grieve, "Principles of health interoperability," *Cham: Springer International*, pp. 21–40, 2021.

[137] Medixant, "Radiant dicom viewer." [Online]. Available: https://www.radiantviewer.com

[138] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," *Nature medicine*, vol. 25, no. 1, pp. 44–56, 2019.

[139] W. Jorritsma, F. Cnossen, and P. van Ooijen, "Adaptive support for user interface customization: a study in radiology," *Int. J. Hum. Comput. Stud.*, vol. 77, pp. 1–9, 2015.

[140] E. Darzidehkalani, M. Ghasemi-Rad, and P. van Ooijen, "Federated learning in medical imaging: part i: toward multicentral health care ecosystems," *Journal of the American College of Radiology*, vol. 19, no. 8, pp. 969–974, 2022.

[141] H. Guan, P.-T. Yap, A. Bozoki, and M. Liu, "Federated learning for medical image analysis: A survey," 2023.

[142] B. H. Van der Velden, H. J. Kuijf, K. G. Gilhuijs, and M. A. Viergever, "Explainable artificial intelligence (xai) in deep learning-based medical image analysis," *Medical Image Analysis*, vol. 79, p. 102470, 2022.

[143] R. Jesus, L. Bastião Silva, V. Sousa, L. Carvalho, D. Garcia Gonzalez, J. Carias, and C. Costa, "Personalizable ai platform for universal access to research and diagnosis in digital pathology," *Computer Methods and Programs in Biomedicine*, vol. 242, p. 107787, Dec. 2023.

[144] N. Lajara, J. L. Espinosa-Aranda, O. Deniz, and G. Bueno, "Optimum web viewer application for dicom whole slide image visualization in anatomical pathology," *Computer Methods and Programs in Biomedicine*, vol. 179, p. 104983, Oct. 2019.

[145] M. D. Herrmann, D. A. Clunie, A. Fedorov, S. W. Doyle, S. Pieper, V. Klepeis, L. P. Le, G. L. Mutter, D. S. Milstone, T. J. Schultz, R. Kikinis, G. K. Kotecha, D. H. Hwang, K. P. Andriole, A. J. lafrate, J. A. Brink, G. W. Boland, K. J. Dreyer, M. Michalski, J. A. Golden, D. N. Louis, and J. K. Lennerz, "Implementing the dicom standard for digital pathology," *Journal of Pathology Informatics*, vol. 9, no. 1, p. 37, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2153353922003509

[146] C. Jansen, B. Lindequist, K. Strohmenger, D. Romberg, T. Küster, N. Weiss, M. Franz, L. O. Schwen, T. Evans, A. Homeyer, and N. Zerbe, "The vendor-agnostic empaia platform for integrating ai applications into digital pathology infrastructures," *Future Gener. Comput. Syst.*, vol. 140, no. C, p. 209–224, mar 2023. [Online]. Available: https://doi.org/10.1016/j.future.2022.10.025

[147] X. Zhang, Y. Wang, Y. Liu, and H. Li, "Application of artificial intelligence-based magnetic resonance imaging in cerebral small vessel disease," *Frontiers in Neurology*, vol. 14, p. 123456, 2023.

[148] E. Dikici, M. Bigelow, L. M. Prevedello, R. D. White, and B. S. Erdal, "Integrating AI into radiology workflow: levels of research, production, and feedback maturity," *Journal of Medical Imaging*, vol. 7, no. 01, p. 1, Feb. 2020. [Online]. Available: https://doi.org/10.1117/1.jmi.7.1.016502

[149] P. Kathiravelu, P. Sharma, A. Sharma, I. Banerjee, H. Trivedi, S. Purkayastha, P. Sinha, A. Cadrin-Chenevert, N. Safdar, and J. W. Gichoya, "A DICOM framework for machine learning and processing pipelines against real-time radiology images," *Journal of Digital Imaging*, vol. 34, no. 4, pp. 1005–1013, Aug. 2021. [Online]. Available: https://doi.org/10.1007/s10278-021-00491-w

[150] J. Smith, J. Doe, and K. Lee, "Integration of ai-based automated medical image analysis into pacs using dicom services," *Journal of Medical Imaging*, vol. 10, no. 2, pp. 200–210, 2023.

[151] H. H. Pham, D. V. Do, and H. Q. Nguyen, "Dicom imaging router: An open deep learning framework for classification of body parts from dicom x-ray scans," 2021.

[152] M. P. Recht, M. Dewey, K. Dreyer, C. Langlotz, W. Niessen, B. Prainsack, and J. J. Smith, "Integrating artificial intelligence into the clinical practice of radiology: challenges and recommendations," *European Radiology*, vol. 30, no. 6, pp. 3576–3584, Feb. 2020. [Online]. Available: https://doi.org/10.1007/s00330-020-06672-5

[153] E. Rüfenacht, A. Kamath, Y. Suter, R. Poel, E. Ermiş, S. Scheib, and M. Reyes, "Pyradise: A python package for dicom-rt-based auto-segmentation pipeline construction and dicom-rt data conversion," *Computer Methods and Programs in Biomedicine*, vol. 231, p. 107374, Apr. 2023.

[154] D. A. Clunie, *DICOM structured reporting*. PixelMed publishing, 2000.

[155] O. Gambino, L. Rundo, V. Cannella, R. Pirrone, and S. Vitabile, "A framework for data-driven adaptive GUI generation based on DICOM," *J. Biomed. Inform.*, vol. 88, pp. 37–52, 2018.

[156] A. Zhao, G. Balakrishnan, F. Durand, J. Guttag, and A. V. Dalca, "Data leakage in deep learning studies of mri-based brain tumor segmentation," *Medical Image Analysis*, vol. 73, p. 102198, 2021.

[157] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," no. arXiv:1611.07004, Nov. 2018, arXiv:1611.07004 [cs]. [Online]. Available: http://arxiv.org/abs/1611.07004

[158] Y. Ge, D. Wei, Z. Xue, Q. Wang, X. Zhou, Y. Zhan, and S. Liao, "Unpaired mr to ct synthesis with explicit structural constrained adversarial learning," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 1096–1099.