

Kidney segmentation in MRI images using deep learning: A comparison between U-NET and Attention U-NET

Francisco Rabaça Moller Freiria

Thesis to obtain the Master of Science Degree in
Electrical and Computer Engineering

Supervisor(s): Prof. António Manuel Gonçalves Pinheiro
Prof. Plinio Moreno López
Ph.D. Researcher Jorge Rafael Mendes Rodrigues

Examination Committee

Chairperson: Prof. Joao Manuel de Freitas Xavier
Supervisor: Prof. Jacinto Carlos Marques Peixoto do Nascimento
Member of the Committee: Prof. Plinio Moreno López

June 2023

Do amor por ti, parti
Perdi (-me), por mares ondulados
Se não morri, afogado vi
Que do nada, nado

Aos que têm no seu coração
a sua parte do que sou,

À minha família.

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First and foremost, I want to thank my Mom and Dad for all the love and support they've always given me. All the good and love I have within me, I owe them. Thanks to my grandmothers, aunts, and sister for supporting the family with their strong bonds. I have no words for them to do justice to their light.

A special thank you to Professor António Pinheiro for trusting my abilities to carry out this work. Also, for all the willingness, attention, and support given that many times it was a source of motivation and inspiration to continue creating. He was always available to discuss solutions and share his profound insights and expertise. His knowledge and critical feedback in image processing are undoubted and helped drive the presented results.

I sincerely thank Professor Plinio Moreno for trusting in my abilities and the work carried out. He was always very understanding throughout the work. Very grateful for your continuous support and constructive feedback, which significantly improved the quality and rigor of this work.

A big thank you to Rafael Rodrigues, who was always very attentive during the study. He always provided his theoretical and technical knowledge and constructive feedback to give new solutions, which significantly improved the quality of this work.

I couldn't stop thanking my friends. Happiness is only real when shared. Fortunately, I am blessed to have people by my side who truly care and love me. I would like to especially thank Renato Farinha and David Brito. To Diogo Pereira, Diogo Prata, João Vaz, Carlos Bidarra, and Ricardo Oliveira, for a lifetime friendship. Ricardo Carvalho and Virgílio Cruz that never forget me. And to all those who have their part of what I am in their heart. A *sin-cero*, thank you.

Resumo

A crescente prevalência da Doença Renal Crónica representa um desafio significativo para a saúde pública, afetando mais de 10% da população. Os biomarcadores de Ressonância Magnética demonstraram ser sensíveis a alterações patofisiológicas podem ajudar a reduzir a necessidade de biópsia, por serem não-invasivos, bem como o risco de complicações em pacientes com Doença Renal Crónica. O volume total do rim é o parâmetro mais avaliado em pacientes com Doença Renal Policística Autossómica Dominante e auxilia o acompanhamento da evolução da Doença Renal Crónica. A segmentação renal é um passo essencial para avaliar o volume do rim. No entanto, depende, frequentemente, da segmentação manual, que é uma tarefa demorada e altamente subjetiva. As metodologias de *deep learning* contribuíram para o desenvolvimento de algoritmos que fornecem resultados precisos, baratos e independentes do utilizador. Esta dissertação explora abordagens baseadas em *deep learning* para segmentação renal, em particular a U-Net e a Attention U-Net. Ambas as arquiteturas foram testadas usando, como funções de custo, a entropia cruzada padrão e uma função de entropia cruzada focal. Finalmente, as soluções propostas foram usadas para realizar segmentação em 2 e 3 classes, como uma tentativa de melhorar a segmentação em regiões fronteiriças do rim. Dados fornecido pela Ação COST PARENCHIMA foram usado tanto para o treino dos modelos, como para a sua validação, através de validação cruzada *leave-one-out*. O melhor coeficiente de Dice obtido com o conjunto de teste foi de 0.966%.

Palavras-chave: Aprendizagem profunda, Doença Renal Crónica, Imagem por Ressonancia Magnética, Segmentação Renal.

Abstract

The increasing prevalence of Chronic Kidney Disease represents a significant public health challenge, affecting more than 10% of the population. Magnetic Resonance Imaging biomarkers have been shown to be sensitive to pathophysiological changes, are non-invasive, and may help reduce the need for biopsies and the risk of complications in patients with Chronic Kidney Disease. Total kidney volume is the most assessed parameter in patients with Autosomal Dominant Polycystic Kidney Disease and helps monitor the progression of Chronic Kidney Disease. Kidney segmentation is an essential step for assessing kidney volume. However, it often relies on manual segmentation, a time-consuming and highly subjective task. Deep learning methodologies have contributed to developing algorithms that provide accurate, inexpensive, and user-independent results. This dissertation explores deep learning-based approaches for kidney segmentation, notably U-Net and Attention U-Net. Both architectures were tested with the standard cross-entropy loss function and a focal cross-entropy loss function. Finally, the proposed solutions were used to perform both 2-class and 3-class segmentation as an attempt to improve the segmentation of border regions. A dataset provided by the COST action PARENCHIMA was used for both model training and validation, with a leave-one-out cross-validation scheme. The best Dice coefficient obtained on the testing set was 0.966%.

Keywords: Chronic Kidney Disease, Deep Learning, Kidney Segmentation, Magnetic Resonance Imaging.

Contents

Acknowledgments	vii
Resumo	ix
Abstract	xi
List of Tables	xvii
List of Figures	xix
Nomenclature	xxi
Glossary	1
1 Introduction	1
1.1 Motivation	1
1.2 Objectives and Deliverables	2
1.3 Thesis Outline	3
2 Background and theoretical context	5
2.1 From Artificial Intelligence to Deep Learning	6
2.1.1 Overview	6
2.1.2 Machine Learning	7
2.2 Artificial Neural Network	10
2.2.1 Overview	10
2.2.2 Multi-layer Network	11
2.2.3 Backpropagation	12
2.2.4 Activation functions	13
2.2.5 Loss functions	16
2.2.6 Metrics	18
2.3 Convolutional Neural Network	22
2.3.1 Overview	22
2.3.2 Convolutional layers	23
2.3.3 Pooling layers	25
2.3.4 U-Net	27
2.3.5 Attention U-Net	28

3	State-of-the-art	31
3.1	Image Segmentation	32
3.1.1	Overview	32
3.1.2	Coarse-to-fine segmentation	32
3.2	Kidney Imaging Modalities	33
3.2.1	Ultrasound	33
3.2.2	Magnetic resonance imaging	35
3.2.3	Computed tomography	36
3.3	Kidney Imaging Segmentation	37
3.3.1	Traditional Methods	37
3.3.2	Kidney image segmentation using deep learning	38
4	Methodology and Dataset	43
4.1	Dataset	44
4.1.1	Dataset description	44
4.1.2	Ground-truth	46
4.1.3	Pre-processing	46
4.1.4	Coarse-to-fine segmentation	46
4.1.5	Data Augmentation	47
4.2	Architectures	51
4.2.1	U-Net	51
4.2.2	Attention U-Net	53
4.3	Training	55
4.4	Test cross-validation	56
5	Results	57
5.1	Segmentation Results	58
5.2	Comparison with the state-of-the-art	61
6	Conclusions	63
6.1	Achievements	63
6.2	Future Work	63
	Bibliography	65
A	Tables, figures, and schemes	77
A.1	Algorithm pseudo-code to obtain contour class from manually annotated ground truth . .	77
A.2	Pipeline illustration of the processes that the original dataset was subjected to the training process.	77
A.3	Train results	79
A.3.1	2 class	79

A.3.2	3 class	79
A.4	Test results	80
A.4.1	2 class	80
A.4.2	3 class	80
A.5	Loss funtion results	81
A.5.1	2 class	81
A.5.2	3 class	81

List of Tables

4.1	Data augmentation transformations.	47
4.2	Hyperparameters used in the proposed implementation.	56
5.1	Evaluation results of Dice coefficient, Jaccard Index and Loss function of binary segmentation models	58
5.2	Evaluation results of Dice coefficient, Jaccard Index and Loss function of three classes segmentation models	58
5.3	Comparison of the DSC and IoU with the state-of-the-art methods	62
A.1	Evaluation results of Dice coefficient, Jaccard Index, Accuracy, Precision, Recall, Loss function and training time of the two classes segmentation models	79
A.2	Evaluation results of Accuracy, Loss function, and training time of the multi-class segmentation models	79
A.3	Evaluation results of Dice coefficient and Jaccard Index of the multi-class segmentation models for each class in the training process	79
A.4	Evaluation results of Precision and Recall of the multi-class segmentation models for each class in the training process	79
A.5	Evaluation results of Dice coefficient, Jaccard Index, Accuracy, Precision, Recall, Loss function and classification time of the two classes segmentation models	80
A.6	Evaluation results of Accuracy, Loss function, and classification time of the multi-class segmentation models	80
A.7	Evaluation results of Dice coefficient and Jaccard Index of the multi-class segmentation models for each class	80
A.8	Evaluation results of Precision and Recall of the multi-class segmentation models for each class	80
A.9	Loss function evolution from the highest dice coefficient result for each of two class methods	81
A.10	Loss function evolution from the highest dice coefficient result for each of three class methods	81

List of Figures

2.1	Venn diagram of the context of Artificial Intelligence, Machine Learning, and Deep Learning methods.	7
2.2	Generic representation of ML algorithms types. a) Supervised Learning. b) Unsupervised Learning. c) Reinforcement Learning. d) Semi-supervised Learning.	9
2.3	From neuron inspiration to ANNs. a) Example of a biological neuron b) Mathematical model of a neuron.	11
2.4	Generic illustration of Multi-Layer Neural Network.	12
2.5	Graphical representation of ReLU activation function.	14
2.6	Graphical representation of Sigmoid activation function	15
2.7	Graphical representation of Soft-max activation function.	16
2.8	Visual representation of confusion matrix.	18
2.9	Visual representation Intersection Over Union metric.	19
2.10	Visual representation Dice coefficient metric.	20
2.11	Illustration of the difference in the number of weights associated with each neuron between an NN and a CNN in the first layer of a $256 \times 256 \times 3$ image.	22
2.12	Illustration of the convolution operation of 4×4 input image patch by a 3×3 kernel obtaining a 2×2 feature mapping.	24
2.13	Illustration of the convolution operation after zero padding.	24
2.14	Image illustration of decision process difference between generic FCN and CNN architectures.	25
2.15	Max-pooling operation applied to a feature map of size 4×4 and the it output of size 2×2 . .	26
2.16	Workflow of CNN main block.	26
2.17	Illustration of the transposed convolution operation of 2×2 feature map by a 2×2 kernel obtaining a 3×3 output image.	28
2.18	Schematic representation of the proposed originally U-Net architecture.	28
2.19	Schematic representation of Attention mechanism architecture.	30
3.1	Example of medical image segmentation.	32

3.2	Example of changes applied to the ground truth on the coarse-to-fine approach. a) Example of an original image from the dataset used in this work; b) Example of ground truth for binary segmentation; c) Example of ground truth with contour class used in the coarse-to-fine approach.	33
3.3	Example image from kidney imaging techniques: a) Ultrasonography, b) Computed tomography and c) Magnetic resonance imaging.	34
4.1	Axis orientation of MRI image.	44
4.2	Original abdominal MRI image.	45
4.3	Ground-truth image of kidney manual segmentation.	45
4.4	Data augmentation inversion example	48
4.5	Data augmentation rotation example	49
4.6	Data augmentation grayscale level example	50
4.7	Illustration of U-Net architecture used in the proposed model	52
4.8	Illustration of attention mechanism architecture used in the proposed model	53
4.9	Illustration of Attention U-Net architecture used in the proposed model	54
A.1	Pipeline illustration of the processes that the original dataset was subjected to the training process.	78
A.2	Loss function evolution from the highest dice coefficient result for each of two class methods	81
A.3	Loss function evolution from the highest dice coefficient result for each of three class methods	82

Nomenclature

ADPKD Autosomal Dominant Polycystic Kidney Disease

AG Attention Gate

AI Artificial Intelligence

ANN Artificial Neural Network

BP Backpropagation

CE Cross-Entropy

CKD Chronic Kidney Disease

CNN Convolutional Neural Network

CT Computed Tomography

DL Deep Learning

DSC Dice similarity coefficient

FCN Fully Convolutional Network

FC Fully Connected network

FL Focal Loss

FN False Negative

FPN Feature Pyramid Network

FP False Positive

GAN Model Generators and Adversary Training

IoU Intersection over Union

LR Learning Rate

MLNNs Multi-Layer Network

ML Machine Learning

MRI	Magnetic Resonance Imaging
NN	Neural Network
PKD	Polycystic Kidney Disease
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RF	Radio Frequency
ROI	Region of Interest
SVM	Support Vector Machine
TE	Echo Time
TKV	Total Kidney Volume
TN	True Negative
TP	True Positive
TR	Repetition Time
US	Ultrasonography

Chapter 1

Introduction

1.1 Motivation

The increasing prevalence of Chronic Kidney Disease (CKD) represents a significant public health challenge, affecting more than 10% of the population. No new therapies have appeared in the last fifteen years, and many recent large trials of CKD progression have failed. Magnetic resonance imaging (MRI) biomarkers have shown high potential to help fill this gap, as they are non-invasive and sensitive to the pathophysiology of CKD due to the need for dedicated in-house expertise and development [1]. Renal volume, cortical/medullary volume, cortical thickness, and volume of cysts or cancers are some biomarkers that can be produced from Abdominal MRI images for the evaluation of CKD, as they are sensitive to pathophysiological changes associated with inflammation, fibrosis, oxygenation, and microstructure. These biomarkers are obtained directly from the kidney as opposed to those obtained from fluids and help to reduce the need for repeat biopsies, reducing the risk of complications and the need for invasive measurements.

To make clinical acceptance wider through the standardization and availability of biomarkers in MRI images, the PARENCHIMA action was accepted by COST, which made available the database used in this study.

Total kidney volume (TKV) measurement is the most accessed parameter in patients with ADPKD [2] and helps in monitoring the progression of CKD, among others. From the kidney's segmentation, the kidneys' volumetry can be obtained. A manual operator can do this segmentation, which makes the process tedious and operator-dependent, but automatic algorithms can also do it. Several problems are addressed in segmenting the renal image; among them is the similarity of the intensity of the renal tissues to the adjacent organs. Thus, intensity threshold-based methodologies do not prove robust segmentation. More complex approaches are required. Therefore, deep learning techniques are applied to medical imaging tasks and have become the leading approach to solving segmentation problems in medical images [1]. Deep learning approaches can potentially analyze an image and provide reproducible and robust volume and segmentation calculations, as they can "learn" important image features to perform pixel-wise classification. By training deep learning models, they can detect patterns, pixel

intensities, and information in ways not easily detected by the naked eye [3]. Volumetry measurements can be accessed through 2D or 3D images since the width resolution is greater than the spatial resolution; segmenting the kidney in 3D images is not justified, which takes more time and requires more computational power.

Currently, the most used approaches in the treatment of medical images are Convolutional Neural networks (CNN) based. These are architectures composed of convolution layers capable of obtaining more important features of the image by reducing its image complexity. This information is combined for image reconstruction. An example of a CNN is the U-Net, the architecture used the most in medical image analysis and to solve segmentation tasks.

The biggest problem for a successful implementation is having large datasets. A particular benefit of this architecture is that it does not require a large dataset compared to other architectures. Another common problem in kidney segmentation comes from class imbalance. Automated kidney segmentation is still challenging since the kidney occupies a small fraction of the image (e.g. $<3\%$). Adding an attention mechanism to focus learning on the image's most important features can help overcome this problem. The literature also suggests implementing Coarse-to-Fine segmentation approaches [4], addressing itself as a solution to the problem of class imbalance or improving the border region between classes.

This dissertation presents an approach using Deep Learning methods for kidney segmentation in abdominal MRI images. A dataset of 21 manually annotated images is used to obtain the ground truth. An automated approach using a modified U-Net and Attention U-Net for kidney targeting has been developed, which may help further evaluate kidney diseases.

1.2 Objectives and Deliverables

A preliminary study was carried out in deep learning, biomarkers in MRI images, and image segmentation in the medical context. Through the analysis of State-of-the-art in deep learning methods, it was concluded that architectures like U-Net are the ones that lead to better segmentation results. Thus, the main idea of this study is to make a comparative study between the results obtained for the segmentation of the U-Net and Attention U-Net kidneys and to prove whether the attention mechanisms will correct the class imbalance problem that is common to several studied approaches in kidney segmentation and whether coarse-to-fine approaches with the introduction of an edge class would improve the segmentation results.

To this end, three main steps are presented in this dissertation: a) the development and study of the loss function that best fits the results obtained for U-Net and Attention U-Net, b) a comparison of results obtained between binary or multi-class segmentation, and c) a comparison of results obtained between U-Net and Attention U-Net. In short, eight segmentation algorithms were implemented and compared:

- U-Net and cross-entropy function as the loss function for binary or multi-class segmentation of kidneys;

- U-Net and focal loss function as the loss function for binary or multi-class segmentation of kidneys;
- Attention U-Net and cross-entropy function as the loss function for binary or multi-class segmentation of kidneys;
- Attention U-Net and focal loss function as the loss function for binary or multi-class segmentation of kidneys.

1.3 Thesis Outline

In this dissertation, the following chapters are presented:

- Chapter 1: The motivation, objectives, and main difficulties of this dissertation;
- Chapter 2: The fundamental concepts related to deep learning so that an arbitrary reader can have the context and conceptual idea of how the methodologies presented by this study work;
- Chapter 3: Critical concepts about image segmentation techniques and kidney imaging modalities. First, an overview of kidney segmentation and its motivation in a medical context is given. Next, kidney imaging techniques are described. Finally, state-of-the-art methods for segmentation of the kidneys in medical images are presented, from traditional methods to deep learning methodologies.
- Chapter 4: Dataset and implementation details. The architectures used are presented, and the training process is described, from pre-processing to classification and validation methodologies.
- Chapter 5: Results obtained for the kidney segmentation on abdominal MRI images are listed and discussed. A comparison of the results obtained with state-of-the-art methods is also made.
- Chapter 6: The conclusions of the work are summarized, as well as possible future works.

Chapter 2

Background and theoretical context

In this chapter, the background regarding deep learning is presented. Understanding the concept of deep learning since its formulation, as the context in which it was conceived and the motivation for its appearance is important so that we can then understand the evolution of its functioning and, in turn, how it obtained the importance it has when presenting representative methodologies of State-of-the-art for kidney segmentation in Magnetic resonance imaging.

Thus, three sections are in the following: Section 2.1 provides the context and motivation for the emergence of deep learning since the formulation of the concept of Artificial Intelligence, as well as its main tasks and types. In section 2.2, theoretical considerations on the functioning and composition of neural networks are presented. Finally, in section 2.3, theoretical conceptualization about CNN functionality is raised, and why these are the neural networks used in image segmentation.

2.1 From Artificial Intelligence to Deep Learning

2.1.1 Overview

To understand complex algorithms of deep learning (DL) as a key instrument to image segmentation, we must briefly perceive the concept of Artificial Intelligence (AI) and Machine learning (ML).

AI is defined as a large range of techniques that are designed to simulate human intelligence, including thinking, behaviors, and perception in computers [5]. AI concept urged in the 1950s.

In general, the concept was initially defined as being the proceed based on the conjecture that every aspect of learning or any other intelligence feature could, in principle, be so precisely described that a machine can be made to simulate it [6]. Soon after the formulation of the AI concept, which was defined in a post-war period, combined with the long history of the logical-philosophical debate of what human intelligence would be, it was concluded that it would be possible to abstract the cognitive faculties of the brain from its physical operations. So, researchers began studying the formal processes that formed intelligent human behavior in medical diagnosis, mathematics, linguistic processing, etc., and trying to reproduce their behavior by computational means [6]. Many of those interested in building automated pattern recognition focused their efforts on artificially replicating the brain's synapses in Artificial Neural Networks (ANNs).

The history of AI is therefore not just the history of mechanical attempts to replicate or replace some static notion of human intelligence, but also a paradigm shift on what intelligence itself is about [6].

Nowadays, most researchers intend to design automated systems to solve complex problems by any means, focusing more on their outcome rather than human-like means, arising ML concept.

ML techniques are a core subset of AI and express the capability of software applications to learn and improve without being explicitly programmed [7]. They have been applied to various domains such as manufacturing, finance, and biomedical problems, and their main tasks include classification, regression, and clustering [7]. Its types and tasks will be discussed in the following subsection 2.1.2.

Although ML techniques are able to categorize and classify data, these techniques are limited in their ability to process data from its raw form. Considerable domain expertise would be needed to design a feature extractor to transform the raw data into a suitable representation for these systems to produce adequate results [8]. ML algorithms have the property of generalization. It is the performance measure of the system to new data; in other words, it is the model's ability to adapt appropriately to unseen data through the same distribution of the one that originated the model, presenting itself as a crucial key in machine learning systems.

As new data cannot contain the same information as the presented on training set, algorithms must be able to generalize what was learned to predict new information. Thus, the generalization mainly serves to identify, *a posteriori*, two types of problems that the algorithm can present: overfitting and underfitting. A model is said to be underfitted when it reaches a state of too much simplicity, considering that it is not capable or flexible enough to understand the standardizable aspects of the data. The model is said to overfit when it learns non-relevant features from the dataset, leading to a noisy generalization, meaning it is too specialized in representing the data from the training dataset because the model has

learned non-relevant features from training data. Thus, the network hyperparameters must be balanced to prevent these problems.

Deep learning architectures created a great breakthrough based on the idea of learning from raw input, avoiding feature engineering. They were achieved by composing simple but non-linear modules from raw input into a higher representation but slightly abstract. Very complex functions can be learned with an optimized number of compositing transformations. They are a subfield of neural network models that have the ability to represent features by having a hierarchical structure with more layers, which allows multidimensional data processing, deepening the learning ability.

DL algorithms are succeeding in devising solutions to several problems that have resisted being solved for several years by the AI community. It has become very good at discovering intricate structures in high-dimensional data [9]. These methods have dramatically improved State-of-the-art in speech recognition, visual object recognition, object detection, and many other domains such as drug discovery and genomics [9].

In short, DL methods refined ML methods and are capable of presenting solutions to problems for which more extensive and complex data structures are needed. It should be noted that this type of method was only possible to build from ideals, propositions, and investigations that started many years ago and proved to be very important in what today constitutes the state-of-the-art of several scientific, social, or economic areas, thus contributing to its evolution.

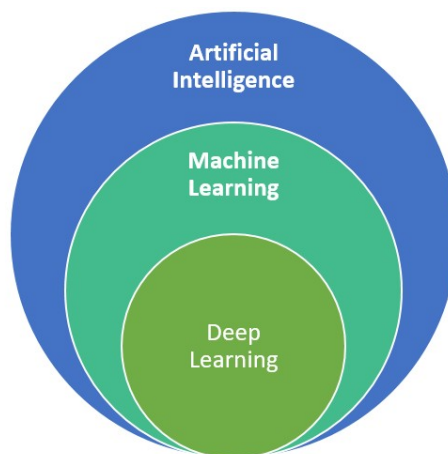


Figure 2.1: Venn diagram of the context of Artificial Intelligence, Machine Learning, and Deep Learning methods. Image adapted from [10].

2.1.2 Machine Learning

The general concept of machine learning (ML) was addressed in the previous subsection 2.1.1 to contextualize them within AI and DL methods. Thus, this subsection will address the different types of methods, tasks, and the concept of generalization associated with ML.

Types

Supervised Learning: ML supervised learning algorithms require a pair of mapped original data and the corresponding ground truth as inputs to the model. This pair of inputs corresponds to training data and the desired outputs, and it is defined as a training dataset. During the training process, the training dataset adjusts model weights until it fits correctly [11].

Model training aims at minimizing a certain loss function. Subsequently, the model can map output from an unseen input. The algorithm proposed in this study fits in this type of ML algorithm, where the training dataset, composed of the original images and the corresponding ground truth, is provided to create a prediction map to the ground truth associated with a random test image. Figure 2.2 (a) shows a generic illustration of the supervised learning process.

Unsupervised Learning: This type of ML algorithm receives a set of inputs but no desired outputs; that is, the model is trained with unlabeled data, as opposed to supervised learning, and tries to search hidden patterns to group data points without human intervention [12], performing clustering methods.

An example of an unsupervised ML algorithm is the work done in the study “An unsupervised learning approach to content-based image retrieval” [13]. Figure 2.2 (b) shows a generic illustration of an unsupervised learning process.

Semi-Supervised Learning: Semi-supervised learning falls between the two types described above. This ML approach is mainly presented as a solution when the results for a specific problem are not good enough with supervised or unsupervised learning. The idea is to use a smaller amount of labeled data once this type of data is more scarce and expensive but has greater ease of convergence. Unlabeled data is also provided, exploring the idea that unlabeled data contain important information for the decision-making process [14].

An example of a semi-supervised ML algorithm is the work done in the study “A Novel Semi-Supervised Learning Approach to Pedestrian Reidentification” [15]. Figure 2.2 (c) shows a generic illustration of the unsupervised learning process.

Reinforcement Learning: This type of algorithm aims to learn the behavior of software agents based on the environment’s response. The algorithm uses observations from the environment and decides to maximize the reward or minimize the cost. Iteratively, the learning agent interprets its environment acting and learning through a trial and error process, usually through a Markov decision process [16], using unlabeled data.

An example of reinforcement learning is shown in work “Reinforcement Learning for Relation Classification From Noisy Data” [17]. Figure 2.2 (d) shows a generic illustration of the semi-supervised learning process.

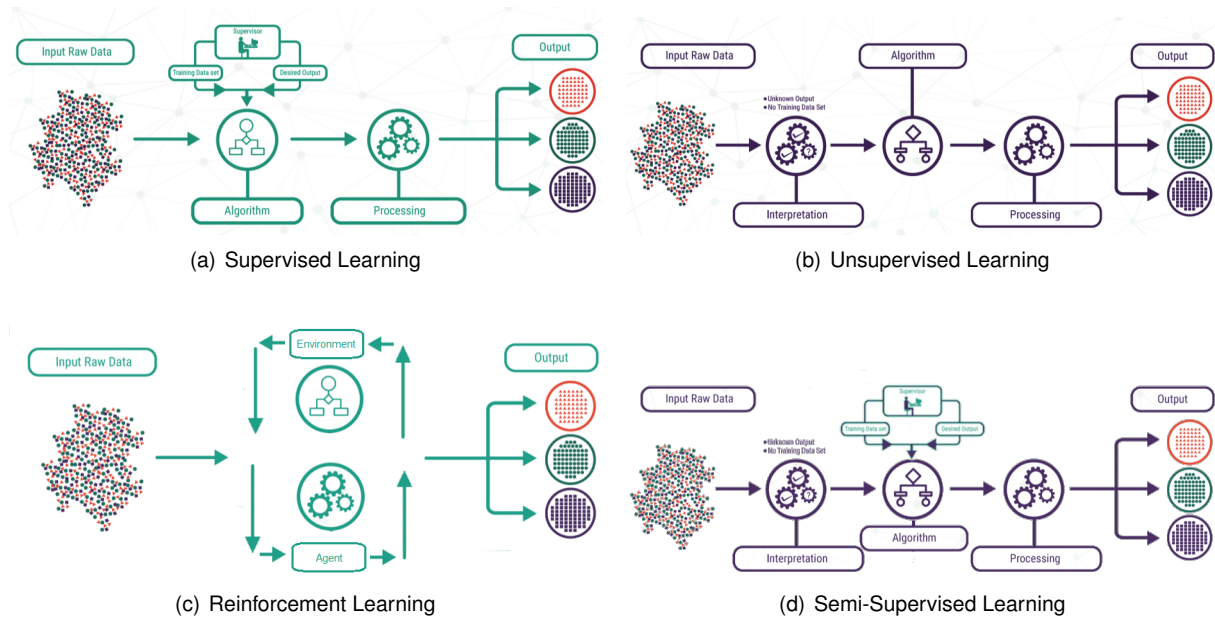


Figure 2.2: Generic representation of ML algorithms types. a) Supervised Learning. Image retrieved from [18]. b) Unsupervised Learning. Image retrieved from [18]. c) Reinforcement Learning. Image adapted from [18]. d) Semi-supervised Learning. Image adapted from [18].

Tasks

Different tasks can be performed with ML algorithms. Since the method proposed in this study fits into supervised learning methods, in this section, we will address the tasks in which supervised learning can perform classification, regression, and segmentation as classification tasks. There are algorithms capable of performing several different tasks since classification problems can be solved through regression problems by fitting the values to the referred classes through approximations.

Classification: Classification tasks are performed once the algorithm learns which features belong to each class from training data. It attempts to classify new data into the corresponding classes, labeling it appropriately. Neural Networks (NNs), Support Vector Machine (SVM), k-nearest neighbor and decision trees are the classification algorithms most commonly used. An example of a classification algorithm was proposed in the study “An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network” [19] where a method of diagnosing diabetic retinopathy for retinal fundus images was proposed, classifying them as a form of diagnosis from a convolutional neural network.

Regression: Regression problems are solved by understanding the relationship between dependent and independent variables and mapping the inputs to a continuous output variable. Classification and regression problems only differ in how output variables are assigned to continuous or discrete outputs. Linear, logistic, and polynomial regression algorithms are the most commonly used. An example of a regression algorithm was proposed in the study “A Regression Approach to Speech Enhancement Based on Deep Neural Networks” [20] where a supervised method to enhance speech by means of

finding a mapping function between noisy and clean speech signals based on DL algorithm is proposed using a non-linear regression model to ensure powerful modeling capability.

Segmentation: Image segmentation is the process of dividing an image into multiple non-overlapping regions according to certain criteria. The division of the image into a certain region or set of pixels reduces the analysis area. The main goal of semantic segmentation algorithms is to make a pixel-wise prediction. Each pixel is classified with a class, and in this way, it becomes a classification task [21]. Often, it is used to find a pattern, as objects or forms like lines or curves, in images.

To perform segmentation, it is necessary to consider two crucial factors for this task: The division of groups of pixels along an image and the number of images to compare common features. It is, therefore, essential to assess a balance point in each of these parameters.

The idea of dividing the images into small groups of pixels is to facilitate the process of standardizing features present in the images of the training set. When analyzing all the small groups of pixels in the images and considering that each set of pixels is supposed to contain very similar features throughout the entire dataset, it is possible to make a more significant analysis of the features present in the whole image. The method presented by this study is an example of image segmentation using a ML algorithm.

2.2 Artificial Neural Network

2.2.1 Overview

Neural network (NN) or Artificial Neural Network (ANN) algorithms were inspired by the sophisticated behavior of brain functions. Since many real-life problems can be solved through engineering and many of them through models, researchers built this type of algorithm based on the idea of creating a model that simulates the behavior of the brain. This is due to its high processing capacity since it can perform complex tasks, such as image and sound processing, among many others, through many processing units, neurons, at a low speed, due to being highly connected. Researchers reached certain levels of intelligence through this type of algorithm since several software for language translation, sound processing, and pattern recognition, among many others, were successfully conceived. It is important to consider that the simulation of human consciousness or its emotions is far from being implemented by computer models [22, 23].

ANNs is a network of connected inputs and outputs in which a weight is associated with each processing unit, neurons. It consists of one input layer, a hidden layer, and an output layer, each with one or more processing units. The weighted sum of the inputs activates processing units, and the signal is activated by transfer functions so that each neuron produces only one output. These transfer functions introduce the non-linearity ability to the training process since the data has an inherently non-linear distribution.

Figure 2.3 (a) and (b) show a biological neuron and an illustration of a mathematical model of a neuron used in ANNs, respectively.

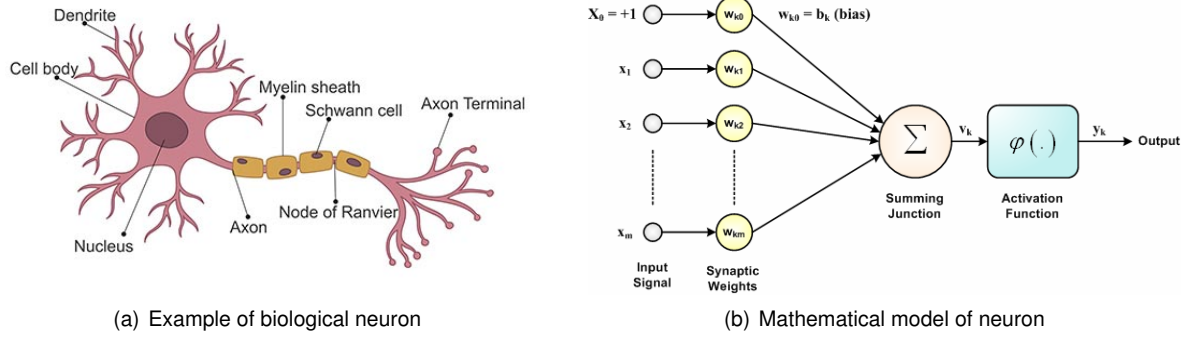


Figure 2.3: From neuron inspiration to ANNs. a) Example of a biological neuron, image retrieved from [24]. b) Mathematical model of a neuron. Image retrieved from [25].

The function that presents the mathematical model of neuron (figure 2.3 (b)) is given by:

$$y = f\left(\sum_i (w_i x_i + b)\right) \quad (2.1)$$

where the inputs (x_i) are weighted from a set of weights (w_i) and the processing unit will sum weighted inputs and pass them through a non-linear activation function $f(\cdot)$. b is a bias term that works as a threshold.

It should be noted that the weights associated with these processing units are updated and optimized throughout the training process, making these algorithms less affected by noise and improving learning ability. Parallelism, which increases the network's speed, and the result's comprehensibility are considered advantages of neurons. Its main disadvantages are cost, time consumption, and interpretability since the parameters associated with hidden units are often difficult to understand [26].

2.2.2 Multi-layer Network

(NNs) are organized in layers that are a combination of neurons or perceptrons. Multi-layer Networks (MLNNs) are NNs structures that combine as many as needed layers. Information is carried along the network from layer to layer. Each neuron has important information distributed to each neuron in the next layer.

It is also important to note that the development of MLNNs is based on concepts such as hidden layers, backpropagation algorithms, activation functions, loss functions, and metrics which are addressed throughout this section. In this subsection, MLNNs input, hidden, and output layers concepts are described. Generic illustration of MLNNs and its layers are represented in figure 2.4.

Input layer: It is the layer that will accept the input data. It is the most superficial layer once does not make essential operations but has the role of restructuring data.

Hidden layer: These layers are called hidden layers because it is impossible to know their values, only their inputs and outputs. They are all layers that lie between the input and output layers and perform the most complex tasks requiring the most computational effort on MLNNs since they calculate many

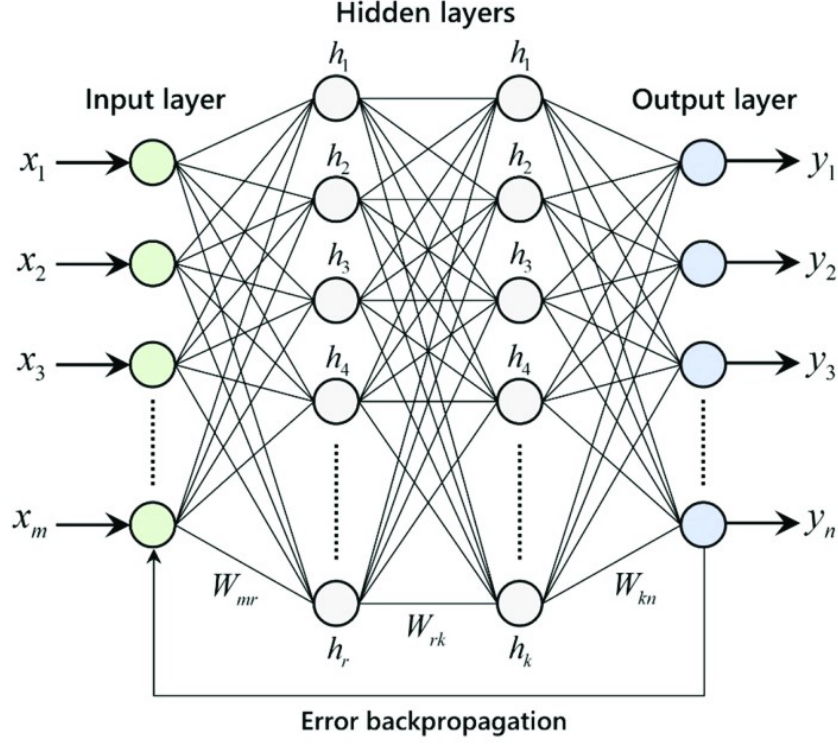


Figure 2.4: Generic illustration of Multi-Layer Neural Network. Image Retrieved from [27].

functions at the same time, allowing the extraction of features from the data through activation functions implemented in neurons (see subsection 2.2.4). They are, therefore, responsible for the excellence of the performance and complexity of the network.

Output layer: Is responsible for producing a meaningful output to solve the proposed problem. Usually uses a different activation function than those used in hidden and input layers.

2.2.3 Backpropagation

Backpropagation (BP) algorithms aim to provide boundaries to the behavior of the data provided as inputs and the desired output, mapping to optimize a function for this relationship.

In general, BP algorithms set values to the network parameters to minimize the error in the output, based on the training cycle, defined by:

$$\hat{w} = \arg L(w) \quad (2.2)$$

w denotes the model's parameters and $L(w)$ is the loss function.

This type of algorithm was introduced in the 1980s through a proposed method called the stochastic gradient descent method. As stochastic gradient descent method, BP methods aim to minimize loss function using the gradient algorithm as it is shown in the following equation:

$$w_i(t+1) = w_i(t) + \Delta w_i(t), \quad \Delta w_i(t) = -\eta \frac{\partial L}{\partial w_i} \quad (2.3)$$

The first step of BP algorithm is to initialize the weights and biases of the network, usually with random values. Once input data propagates through the network with initial weights and biases, output data is compared with ground truth using the loss function. It is possible to calculate the gradient to minimize loss function as it is described by $(\partial L / \partial w)$ of equation (2.3). This method starts by computing the gradient for the last layers and goes backward, applying the chain rule for each parameter through the derivative of the cost function of the loss function. Thus, the backpropagation algorithm calculates the gradient, and the weights and biases can be updated at each iteration according to the optimization function used.

It is also worth mentioning another hyperparameter (η), LR, which minimizes the cost function impacts to the updating of weights and biases. This hyperparameter must be balanced to prevent overfitting and to obtain accurate results.

Summarizing, the training process corresponds to an optimization problem using the minimization of the loss function to tune model parameters. The model's parameters are adjusted according to the optimizer formula and usually require the calculation of the gradient of loss function through BP algorithm. As BP algorithm is an iterative method, it should be computed until any evaluation metric converges or some stop criteria are verified.

2.2.4 Activation functions

As mentioned before, activation functions are primary functions of the network since it defines the output at each neuron. Without it, the NNs would have a linear behavior between the input and the output [28]. They allow the network to learn non-linear decision boundaries once data usually do not have a linear distribution. The non-linear activation functions can be interpreted as the mechanism that decides whether the neuron should be active or not, for a given input. Its use in deep learning problems resorts to differential functions so that the algorithm can optimize by the backpropagation algorithm. Choosing the optimal activation functions is also a hyperparameter to take into account when designing a neural network architecture.

Below is a list of activation functions commonly used in deep learning problems, however, the proposed method only uses the Rectified Linear Unit (ReLU) activation function in the input and hidden layers and the Soft-max or sigmoid activation function in the output layer depending on the number of classes considered.

ReLU function: It was first introduced in the study “Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements” [29] and therefore used in the NNs first works. With the development of new techniques and activation functions, it was believed not to be the function that would bring the best performance to networks in a generalized way. However, from 2009 onwards, it became recommended for DL problems, pointing to better performances and learning ability [30].

Equation 2.4 describes the ReLU activation function and is graphically represented in figure 2.5.

$$f(x) = \max(0, x) \quad (2.4)$$

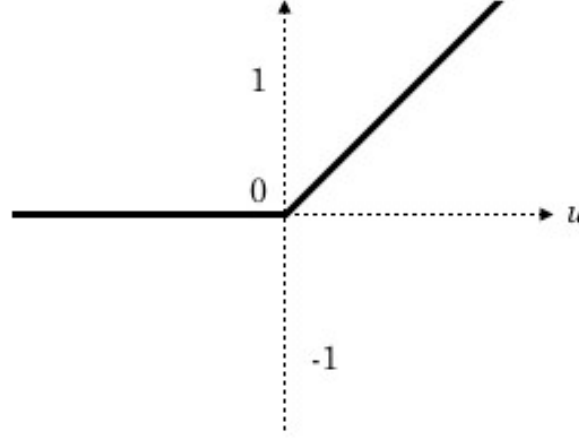


Figure 2.5: Graphical representation of ReLU activation function. Image retrieved from [31].

This function has the advantages of not suffering from vanishing gradient, in the positive region ($x \geq 0$) it does not saturate, it is the most computationally efficient non-linear activation function, converges faster than other functions also commonly used as sigmoid or Hyperbolic Tangent [32].

The disadvantage is that its output is not zero-centered; when the gradient is 0, ($x < 0$), that neuron will be inactive in the network, making it a dead neuron, which means that it will never be updated or activated by the network [32].

This activation function is used in the proposed method's input and hidden layers.

Sigmoid function: It is a very commonly used function. This function produces an output between two values, generally between 0 and 1. When defining a threshold, its result is the probability that a given data belongs to a class A or B . Thus, this function is used in output layers of binary classification problems.

Equation 2.5 describes the Sigmoid activation function and is graphically represented in figure 2.6.

$$f(x) = \frac{1}{1 + e^x} \quad (2.5)$$

This function suffers from a vanishing gradient problem, which can kill neurons. That is, the gradient will be so slight that there will be no signal flowing through the saturated neuron as it is propagated in the backward hidden layers, making slow learning, which makes it discouraged by modern literature. Other disadvantages are that the output is not zero-centered and becomes computationally expensive, as it is an exponential function [32].

As an advantage, it can produce a result between 0 and 1, translating its result into a probability and interpreted as a saturating “fire rate” of a neuron [32].

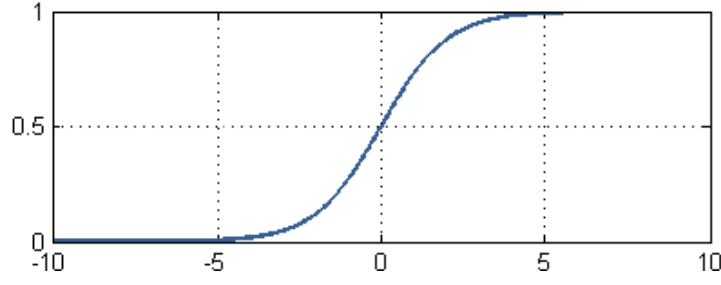


Figure 2.6: Graphical representation of Sigmoid activation function. Image retrieved from [33].

In summary, since this function is an activation function that is a cumulative distributive function, it produces a very useful probabilistic output in classification problems.

Soft-max function: The soft-max function normalizes a vector of numbers to probabilities. Thus, when the network produces N output values in a classification problem, these values must be normalized to assign the most likely class to the predicted output. This function converts the weighted sum of the output into a probability that, when all added together, will give unity.

Bishop [34] et al. says, "The term soft-max is used because this activation function represents a smooth version of the winner-takes-all activation model in which the unit with the largest input has output +1 while all other units have output 0", unlike sigmoid which assigns a probability other than 0 to each class.

This activation function is usually used in the output layer to predict a multinomial probability distribution, making it a recurrent tool in multi-class classification problems. Goodfellow et al. [35] in the book "Deep Learning" which presents a wide range of topics in deep learning says: "Any time we wish to represent a probability distribution over a discrete variable with n possible values, we may use the soft-max function. This can be seen as a generalization of the sigmoid function, which was used to represent a probability distribution over a binary variable.", reinforcing the above.

Although it can be used as an activation function in hidden layers, this is less common; it should only be used when the model internally needs to choose or weigh several different inputs in a concatenation layer.

Equation 2.6 describes the soft-max activation function where \vec{z} is the input vector with z_i values and K classes and can take any real value, e^{z_i} is the exponential function applied to each element of the input vector and sum (\sum) is the normalization term, that ensures that all output values sum to 1 and are in the range (0,1). The sigmoid function is a soft-max function with two input classes. Figure 2.7 illustrates graphically the soft-max function. As can be seen in the figure, the sigmoid and soft-max functions have the same shape.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.6)$$

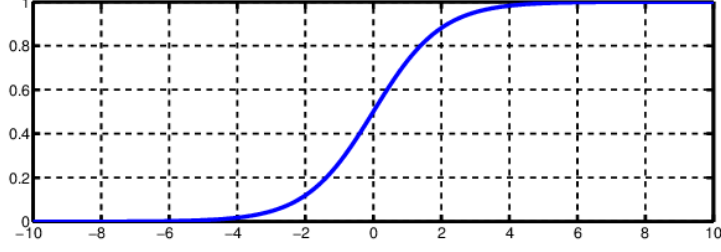


Figure 2.7: Graphical representation of Soft-max activation function. Image adapted from [36].

2.2.5 Loss functions

The loss function quantifies the difference between the expected result and the result produced by the model. It is one of the most critical parameters in neural network models. Any ML problem aims to minimize the loss function since it calculates the gradient to minimize the cost function of the loss function to update the weights and bias values through backpropagation algorithms, as referred to in subsection 2.2.3, thus becoming a model evaluation metric. The adjustment of the hyperparameters has the purpose of reducing the loss value. This subsection discusses the loss functions used in this study.

Categorical Cross Entropy: In information theory, cross-entropy between two probability distributions over the same set of events, measures the average number of bits needed to identify a given event if the encoding used is optimized for the predicted probability distribution. The equation 2.7 describes the calculation of this entropy where p is the true probability distribution, and q is the estimated probability distribution.

$$H(p, q) = \sum_i p_i \log(q_i) \quad (2.7)$$

In machine learning, cross-entropy defines a loss function that compares the predicted probability of a given pixel belonging to a given class with the class it belongs to and scores by penalizing the probabilities based on the expected value. This statistical distribution of labels plays an important role in evaluating training accuracy.

Thus, the cross entropy loss function gives a measure of dissimilarity between the true probability and the probability estimated by the model, which is given by the following equation:

$$L_{CE}(y, \hat{y}) = - \sum_{c=1}^c \sum_{i=1}^N (y_{i,c} \log \hat{y}_{i,c}) \quad (2.8)$$

The predicted probability for each pixel belonging to a determined class ($\hat{y}_{i,c}$) is given by the soft-max equation 2.6 defined in previous subsection 2.2.4, where $z_i(x)$ denotes the activation function in feature channel i at pixel position $x \in \Omega$ with $\Omega \subset \mathbb{Z}^2$, then penalizes it at each position the deviation of true value ($\hat{y}_{i,c}$) using:

$$E = \sum_{x \in \Omega} w(x) \log(\hat{y}_{i,c}(x)) \quad (2.9)$$

Where $C : \Omega \rightarrow \{1, \dots, K\}$ is the true label of each pixel and $w : \Omega \rightarrow \mathbb{R}$ is a weight map that gives to some pixels more importance in training [37]. The soft-max function calculates each pixel's energy function (equation 2.9). Combined with the cross-entropy loss function, we obtain the approximate maximum function of the probability of a pixel belonging to the correct class.

In short, categorical cross-entropy is the loss function that uses the soft-max activation function to choose the one that produces the least loss among several classes.

Dice loss function: Dice loss is commonly used in medical image segmentation tasks and is recommended in problems with imbalanced datasets. When training an imbalanced dataset, the training tends to favor the majority class and produces biased results. The dice function tends to penalize false negatives more heavily than false positives. False negatives are the most costly errors, corresponding to the misclassified minority class. By penalizing false negatives more heavily, the loss function can help to balance the cost of errors. It is also well-suited to problems where the objective is to optimize the overlap region between predicted results and ground truth, as in image segmentation problems.

However, it only addresses the problem of foreground and background imbalance. Still, it ignores another imbalance between easy and hard data examples that also severely affects the process of training a learning model.

Dice loss is calculated through the index coefficient and is used to calculate the dissimilarity between two images and is given by the following equation:

$$L_{Dice}(y, \hat{y}) = 1 - \frac{2 \sum_{c=1}^C \sum_{i=1}^N (y_{i,c} \hat{y}_{i,c})}{\sum_{c=1}^C \sum_{i=1}^N (y_{i,c} + \hat{y}_{i,c}) + \epsilon} \quad (2.10)$$

C represents the number of classes and N is the number of output pixels. $(y_{i,c})$ and $(\hat{y}_{i,c})$ the true and predicted probabilities, respectively, of pixel i of sample t being on class c . ϵ is the smooth value that protects the denominator of taking the value 0.

Focal loss function: Focal loss is used to deal with the front-background imbalance problems in object detection, making it very useful in problems where the background class is imbalanced compared to the object class. This loss function is similar to the cross-entropy loss function. Still, a modeling term is applied to focus learning on hard-to-classify examples, decreasing the weight on easy-to-classify examples and increasing the weights on hard-to-classify examples, depending on the training process confidence of predictions. Consequently, it improves performance on imbalanced datasets, as in kidney segmentation problems.

The modeling term is dynamically dimensioned, where the factor decays to zero as confidence in the correct class increases. This modeling term $(1 - p)^\gamma$ is added to the standard cross-entropy criteria. Setting $(\gamma > 0)$ reduces the relative loss for well-classified examples $(p > 0.5)$ and increases focus on misclassified examples. γ is the focus parameter. As a Cross-entropy loss function, the focal loss uses

the soft-max equation to obtain the estimated probability (p), which attributes great numerical stability [38]. The parameter defines the importance of each class.

The following equation gives the focal loss:

$$L_{Focal}(p) = -\alpha(1 - p)^\gamma \log(p) \quad (2.11)$$

2.2.6 Metrics

Any machine learning model needs evaluation metrics to measure the results' quality. Usually, the training process is computed until an evaluation metric converges. It is also an important tool for model optimization.

Since there are several metrics, such as precision, accuracy, confusion matrix, cross-entropy, etc., it is necessary to study the most appropriate metrics for each problem, as the model can produce excellent results for a given metric and an invalid output. This happens when the metric considered is no longer correct for the problem. Thus, the evaluation metrics used in the proposed model will be presented in this subsection.

Confusion matrix: The confusion matrix, although not a metric, is a table that summarizes the results of the classification predictions and produces the parameters used to calculate the evaluation metrics. This matrix shows how many predictions are correct or incorrect per class, allowing you to understand which classes are confused with which other classes. Each column of the matrix represents instances in the actual class, and each row represents the instances in the predicted class or vice versa. Figure 2.8 shows an example of a binary confusion matrix.

		Actual class	
		P	N
Predicted class	P	TP	FP
	N	FN	TN

Figure 2.8: Visual representation of confusion matrix. Image retrieved from [39].

In the case of binary classification, the confusion matrix is a 2×2 matrix where **TP** denotes true positives (predicted correctly as positive), **TN** denotes true negatives (predicted correctly as negative), **FP** denotes false positives (mispredicted as positive) and **FN** denotes false negatives (mispredicted as negative) in a sense that predicted values are described as positives or negatives and actual values described as true or false.

Accuracy: In image segmentation, accuracy is the metric that informs the percentage of correct pixels between the model output for a given image and the corresponding ground truth. The equation 2.12 corresponds to the accuracy calculation equation.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (2.12)$$

This is not a very efficient metric for kidney segmentation due to the class imbalance problem. In case the ground truth of the images presents a much higher percentage of background class than the kidney class, it is easy to obtain a high accuracy value. For example, if a given ground truth of a kidney image has as background 96% of image pixels, is Class A and 4% of kidney pixels is Class B, if you classify the entire image as Class A, the accuracy will be 96%, which is a high value. In conclusion, this metric is inappropriate for evaluating the results produced by the model. However, this metric becomes important to evaluate the model's performance since high values are already expected, thus allowing it to monitor its evolution throughout the training process easier, but not the validity of the results.

Jaccard index: Jaccard index or Intersection Over Union (IoU) urges from the necessity of having a metric that quantifies the results in terms of the spatial similarity. Provides information on how well the algorithm could identify the true region of interest (ROI), measuring the overlap between the segmented image and the ground truth.

IoU is the most commonly used for comparing the similarity between two arbitrary shapes [40]. Encodes the shape properties on the same region of two images and then calculates a normalized measure focusing on their similarities. It is to be noted that this metric is even more relevant in multi-classification problems [41]. However, no strong correlation exists between minimizing the losses and an IoU improvement.

Equation 2.13 present the formula for the Jaccard index, and figure 2.9 presents the visual representation of IoU.

$$JaccardIndex = \frac{TP}{TP + FP + FN} \quad (2.13)$$

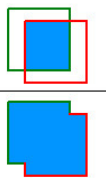
$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$


Figure 2.9: Visual representation Intersection Over Union metric. Image retrieved from [42].

Dice coefficient: Dice coefficient (DSC) measures the overlap between the segmented image and the ground truth, thus taking into account the spatial arrangement and therefore is a metric that resembles the Jaccard index. Its value varies from 0, when there is no spatial overlap between the training set and the result produced, and 1, when the overlap is complete.

The differences between the two metrics are that DSC tends to penalize errors closer to the average, and the IoU tends to penalize them closer to the worst case possible. DSC tends to emphasize the smaller regions of overlap.

It is also an important metric to assess the model's reproducibility and whether the results obtained in the training process are fighting the imbalance between the foreground and the background [43].

In equation 2.14 is represented DSC equation and figure 2.10 it is the visual illustration.

$$DCS = \frac{2TP}{2TP + FN + FP} \quad (2.14)$$

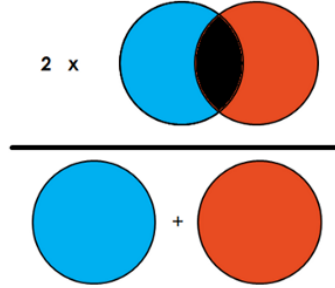


Figure 2.10: Visual representation Dice coefficient metric. Image retrieved from [44].

Recall: The recall is the metric that presents the ratio between correctly classified pixels and all pixels that belong to the object of interest, detecting false positives. Therefore, useful to understand how well the algorithm explicitly identifies the ROI. In a medical context, this metric is important to avoid unnecessary treatment.

The equation that defines the recall using the elements of the confusion matrix is presented in 2.15.

$$Recall = \frac{TP}{TP + FN} \quad (2.15)$$

Precision: Precision is the metric that presents the ratio between true positives and all positives; it is, therefore, useful to understand how well the algorithm correctly identifies the pixels that belong to the object of interest, which in a medical context is important so that the evaluation of the segmented object can help in the diagnosis and medical evaluation.

The equation that defines the precision using the elements of the confusion matrix is presented in 2.16.

$$Precision = \frac{TP}{TP + FP} \quad (2.16)$$

All the presented metrics are commonly used in image segmentation, and each provides a different understanding of the quality of the segmentation.

The Jaccard index and the Dice coefficient provide a better understanding of the assessment because they measure the overlap between the segmented image and the ground truth, considering the spatial arrangement, size, and shape of the segmented object [45]. They can also help to identify regions where the algorithm has difficulties segmenting, helping to improve the algorithm.

Accuracy does not consider the spatial arrangement of pixels, which in the case of class imbalance images, may not provide information of relevant interest. However, it is an excellent metric that helps monitor the evolution of the training process, allowing to improve of the algorithm over time since high values are expected.

Recall and precision are more spatially aware than accuracy. Still, they do not consider the size and shape of the segmented object, which is very important in medical image segmentation. They are interested in being used as metrics because, in the case of precision, it allows the detection of false positives to be evaluated, which in a medical context can lead to unnecessary treatment. The recall is the metric that best identifies whether the pixels of the object of interest are detected, which is important for medical evaluation in a medical context. Both metrics are based on the understanding and measure of relevance, hence are helpful in the quality performance description of the segmentation techniques [46].

2.3 Convolutional Neural Network

2.3.1 Overview

Convolutional Neural Networks (CNNs) were introduced in the 1980s and were based on studies of brain sensory response to visual features such as shapes, colors, or textures, and with their development they proved to be a very intrinsic tool for many computer vision tasks.

As previously mentioned, neurons are processing units of a neural network. These units capture the image features by assigning a weight to each pixel. The weights define the strength of the connection between the neurons so that the activation function can process the information from the initial to the output layer in a balanced way.

The number of weights associated with each neuron in NN solutions is given by $h \times w \times c + 1$, that for an RGB image is $256 \times 256 \times 3 + 1 = 196\,608$. This high number of weights for a fully connected network (FC) leads to overfitting problems and has a high computational cost [47]. To prevent this problem, CNNs were used. The basic proposition behind the concept of CNNs is that the pixels in the images are not entirely independent since surrounding pixels have relevant information about the pixel to be evaluated. Thus, each neuron only observes a small region of the image. Figure 2.11 shows the difference in the number of weights associated with each neuron on a first layer of a NN vs CNN.

In the brain, each neuron responds to stimuli in a restricted region of the visual cortex, the receptive field of that neuron [48]. Thus, it is possible to make a parallelism between the brain and the CNNs considering that their receptive field is the area that each neuron is processing.

A CNN has several layers; in each layer, the neurons are organized in several feature maps. Neurons belonging to the same feature map share the same weight parameters. This weight sharing allows you to reduce the number of parameters that can be learned [28].

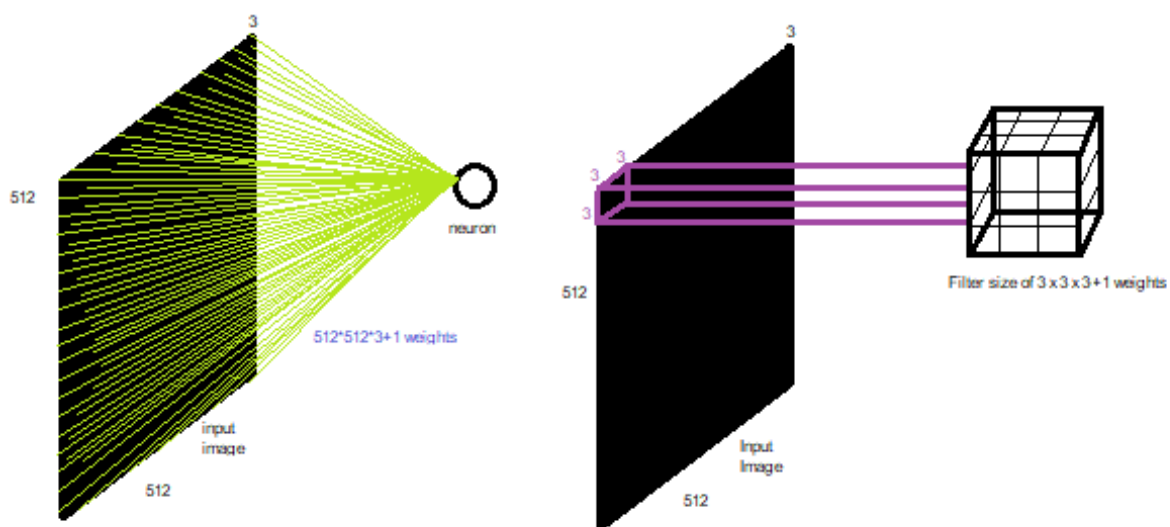


Figure 2.11: Illustration of the difference in the number of weights associated with each neuron between an NN and a CNN in the first layer of a 256 x 256 x 3 image.

Thus, disregarding image size, neurons from the same feature map that observe the same region (n, m, d) and share the same weights only need to learn $n \times m \times d + b$ parameters. Most of the proposed algorithms use $(3 \times 3 \times d + b)$ or $(5 \times 5 \times d + b)$ where d represents the depth of the input image or the number of features mapped by the last layer and b , usually takes the value of 1, represents the bias term.

The weight parameters are called the filter or kernel of the convolution operation and will be explained in more detail in the following subsection.

2.3.2 Convolutional layers

The main building block of CNN is the convolutional layer, and each layer has one or more filters that are the kernel of the convolution operations.

In convolutional layers of CNNs, convolution operations replace multiplication operations typical of FCs. These allow for a reduction of the number of parameters compared to FCs (see image 2.11) and the preservation of the spatial input structure. CNNs are ideal for data with a grid-like topology (such as images) as spatial relations between particular features are taken into account during convolution and pooling operations [49].

To compute convolution operation, a kernel slides along each image pixel and assign a value placed on the filtered image [28]. Each value of the filtered image is the result of taking a dot product between the filter and a small part of the input image multiplied by the weight of the filter and adding the bias term by the formula $(wx + b)$, where w denotes the weights of filter, b bias term and x the values of the image in a small window. In image 2.12, an example of a convolution operation is illustrated for an input size image of 4×4 and filtered by a 3×3 kernel, obtaining a 2×2 feature map. The filtered images or the feature maps are then used and filtered again on subsequent layers.

As one advances through the convolutional layers and the feature extraction is computed, the levels of abstract representation decrease. In the first layers, these can detect bubbles and edges, and in the lower layers, more discriminative artifacts.

The convolution operation naturally removes dimensions to form the filtered image from the original image because the kernel cannot be centered on the edges of the images. To solve this problem, there is the padding method that can be applied with several techniques, such as filling with zeros, used in the proposed model, which expands the edges of images with pixels that take the value of zero, allowing to obtain a filtered output with the same dimension as the original image. The number of pixels to fill in each direction is calculated by:

$$p = \frac{f - 1}{2} \quad (2.17)$$

where f denotes the kernel size and p is the number of pixels to fill in each direction, and the filled image passes from (h, w) size to $(h + 2p, w + 2p)$ size [28]. An illustration of the padding operation is presented in figure 2.12.

The kernel stride must be considered. This represents the number of pixels/units the filter moves

over time. In image 2.13, an example of zero padding with a stride of 1 is illustrated, showing that the size of the filtered image remains unchanged.

Since the convolution is a linear operation and the mapping must have a non-linear behavior, activation functions must be applied to the filters to allow the network to learn nonlinear decision limits and obtain non-linear results, as said in subsection 2.2.4.

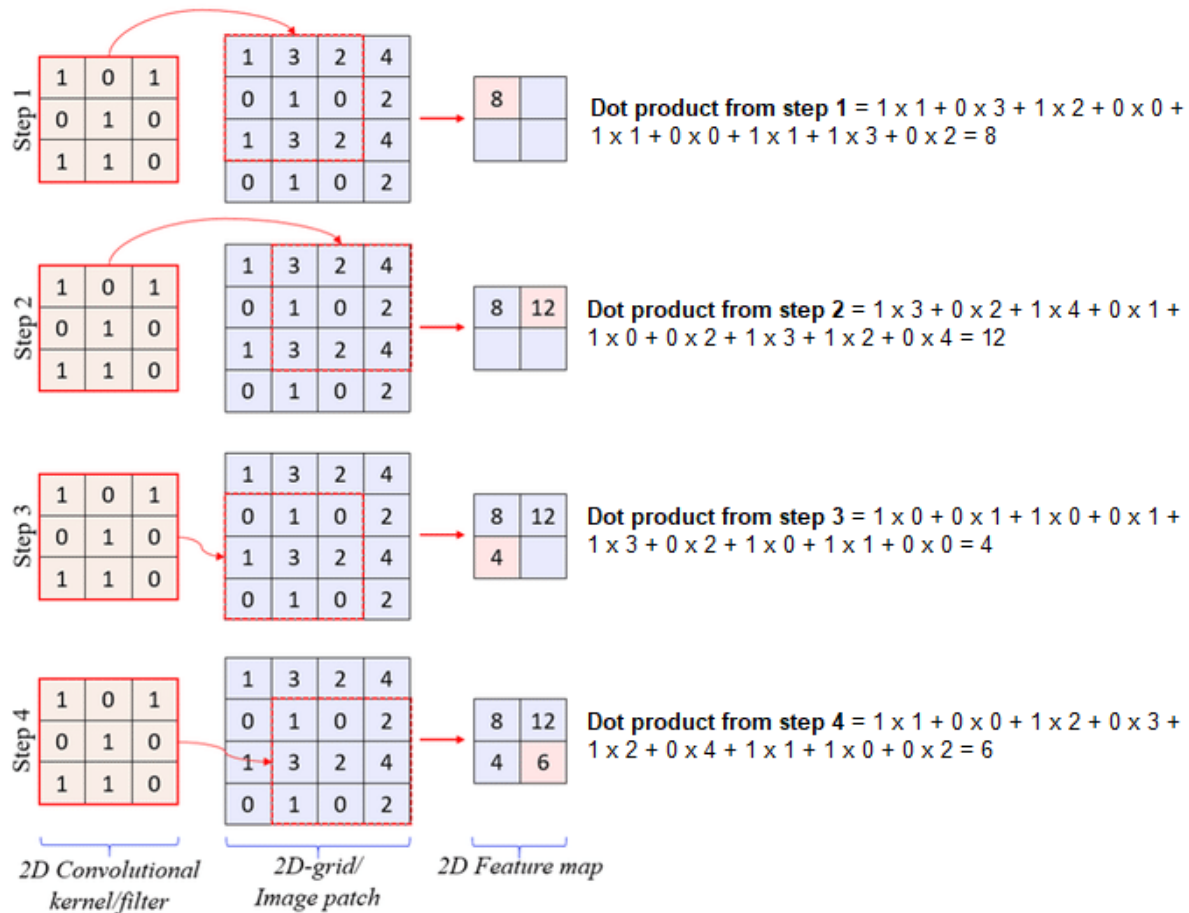


Figure 2.12: Illustration of the convolution operation of 4 x4 input image patch by a 3 x 3 kernel obtaining a 2 x 2 feature mapping. Image adapted from [50].

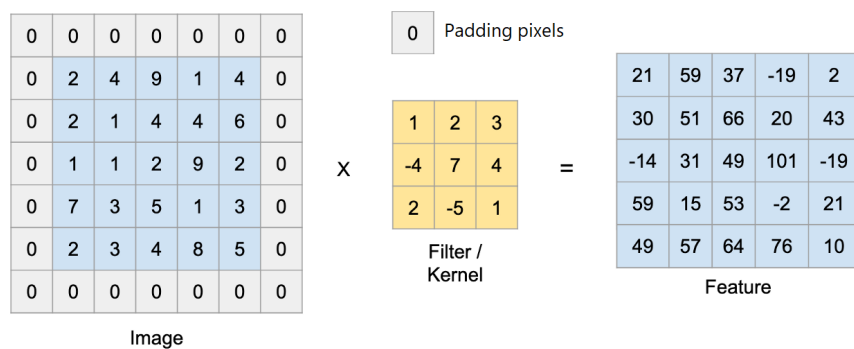


Figure 2.13: Illustration of the convolution operation after zero padding. Image retrieved from [51].

Fully convolutional network (FCN) is a CNN type of architecture in which the decision process,

through the output layer, is obtained by a convolutional operation allowing all decisions to depend on what is learned locally, through spatial information. In CNNs, an image goes through convolutional layers to obtain a feature mapping, and the decision process is done on fully connected layers (FC) assigning a category to the image which makes a good network for image classification. In FCN, the convolutional output layer performs pixel-wise classification as part of the segmentation process.

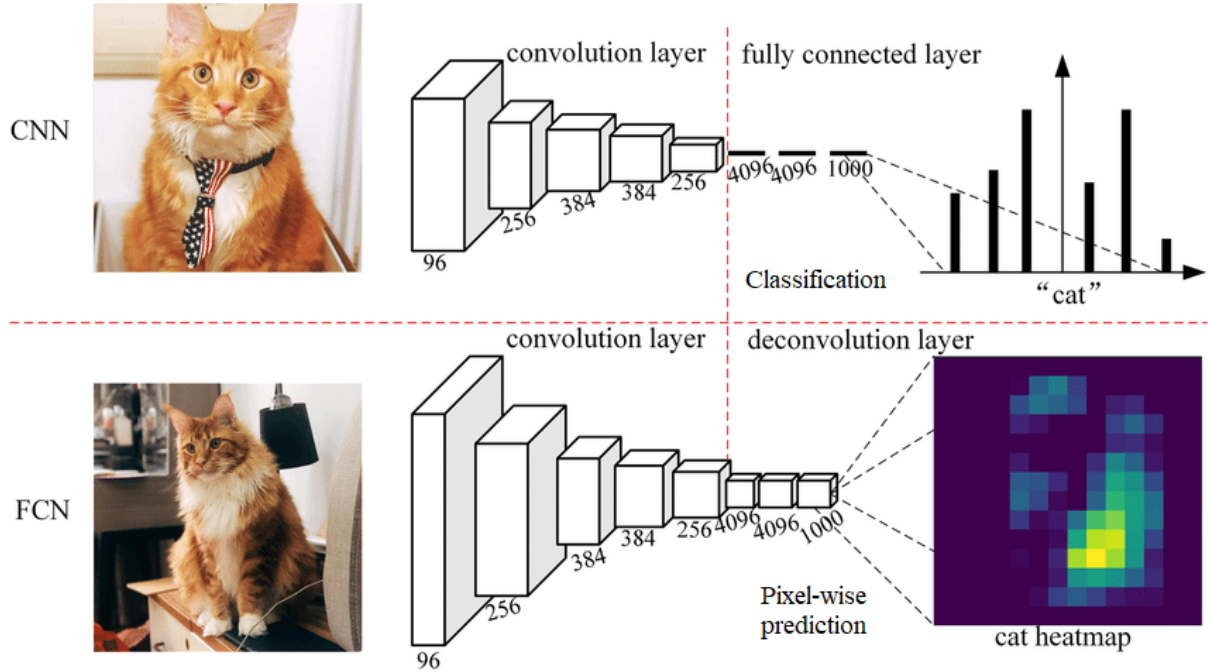


Figure 2.14: Image illustration of decision process difference between generic FCN and CNN architectures. Image Retrieved from [52].

2.3.3 Pooling layers

The pooling layers operate on each feature map by computing subsamples. This operation aims to reduce the feature map representation to a more manageable size, consequently reducing the number of trainable parameters, thus reducing the computational effort inherent to the network. These operations also allow obtaining a representation of features invariant to small translation in input [35]. Pooling operations allow extracting high-level features from the image, decreasing image size and increasing depth.

The relationship between the size of the output feature map (H, W) , and the size of the input feature map (h, w) given a pooled region of size $f \times f$ and stride s is given by the equation:

$$H = \frac{h - f + s}{s}, W = \frac{w - f + s}{s} \quad (2.18)$$

The most common pooling operation is max-pooling [53].

This operation slides a window along the feature map and assigns to each output position the maximum value corresponding to that window. Typically, this operation is applied to non-overlapping regions

of the feature map, and the window size is 2×2 with a stride of 2, yielding an output that is half the length and width.

In figure 2.15, the max-pooling operation is represented, where it is possible to understand, through the visual representation, the reduction of the feature window size and its mapping.

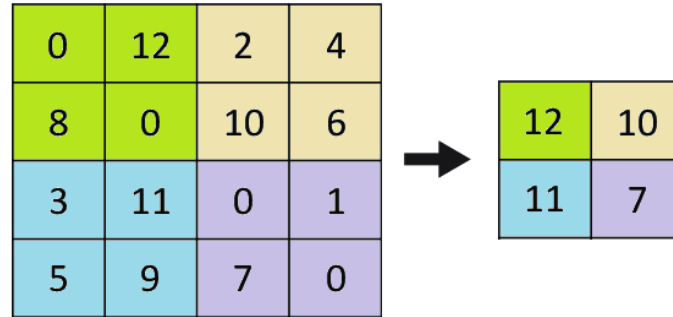


Figure 2.15: Max-pooling operation applied to a feature map of size 4×4 and the it output of size 2×2 .

To generically summarize the operation of the main block of a CNN in figure 2.16, its workflow is represented.

Since it is composed of convolution operations, N filters (w_1, \dots, w_N) are generated for a given image I , generating in turn $(I * w_1, \dots, I * w_N)$ filtered images. When the activation function is applied to these filtered images, $(f(I * w_1), \dots, f(I * w_N))$ feature maps are generated. Finally, a pooling operation can be applied to the feature maps generating $(pool(f(I * w_1)), \dots, pool(f(I * w_N)))$ reduced samples of the feature maps.

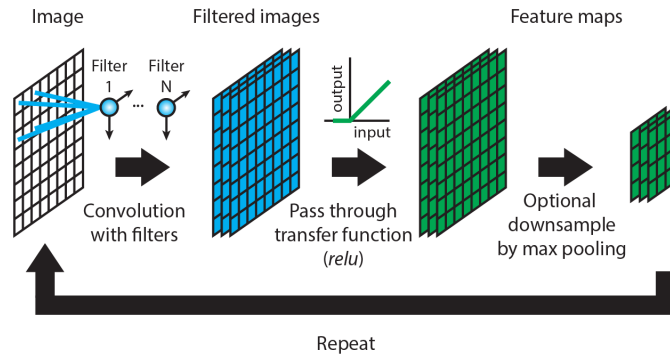


Figure 2.16: Workflow of CNN main block. Image retrieved from [54].

2.3.4 U-Net

U-Net is a deep learning architecture designed to solve the lack of data problems inherent in medical image segmentation. It is, therefore, an architecture designed to perform a pixel-wise classification.

In addition to the traditional methods already known for some decades, several CNN architectures were studied for this purpose. Still, due to the lack of training data, they did not answer the evolution of deep learning methods in the medical imaging context.

Other issues addressed to previously proposed architectures have been resolved. Until 2015, the method proposed by Ciresan et al. [55], a sliding-window convolution network, trained patch by patch, made the process very time-consuming and introduced a lot of redundancy due to overlapping patches. This feature leads to training models with larger patches, which leads to more max-pooling layers, which could reduce the location accuracy. On the other hand, too small patches would lead to the network not learning the context, so there is a trade-off between location accuracy and context usage. Thus, a classifier output was proposed in the U-Net implementation that considers the features obtained in the multiple layers, achieving a good classification accuracy and using the context simultaneously.

This architecture was inspired by a Fully Convolutional Network (FCN), and its main difference lies in the fact that it has more up-sampling layers. This architecture has as many up-sampling layers as down-sampling layers, making it symmetrical. Its name U-Net derives from the fact that it is a U-shaped architecture.

The U-Net architecture is divided into two paths, the first being called the down-sampling path. The down-sampling path is the contraction path, also called the encoder, composed of convolution and pooling layers. This type of operation allows extracting high-level feature structures from the input data. As you advance along the architecture, the size of the images is compressed while the depth increases. This makes it possible to increase the receptive field, allowing the filters to capture larger areas since the max-pooling operations remove less important pixels, but the spatial information decreases. Thus, the encoder generates feature maps which are low-resolution representations of the input image [56].

The second path of this architecture is the up-sampling path, also called the decoder. This converts low-resolution images into high-resolution images, representing the original image's pixel-by-pixel segmentation. The primary operations of the decoder are the transposed convolution operations, which are up-sampling operations with learnable parameters. The learnable parameters come from the kernel matrix. They are adjusted throughout the training process. Each image pixel produced in the down-sample path is multiplied by the kernel matrix, generating a matrix of a higher order for each pixel. Finally, all matrices are summed, obtaining the output of the transposed convolution. Figure 2.17 illustrates a transposed convolution operation.

Along the up-sample path, in each layer, the width and length of the image are doubled while the number of channels is halved.

Another characteristic of this U-shaped architecture is that spatial information from the down-sample path is concatenated into the up-sample path. This feature introduces an improvement in spatial location, represented by the grey arrows in figure 2.18, representing the schematic representation of the originally proposed U-Net architecture.

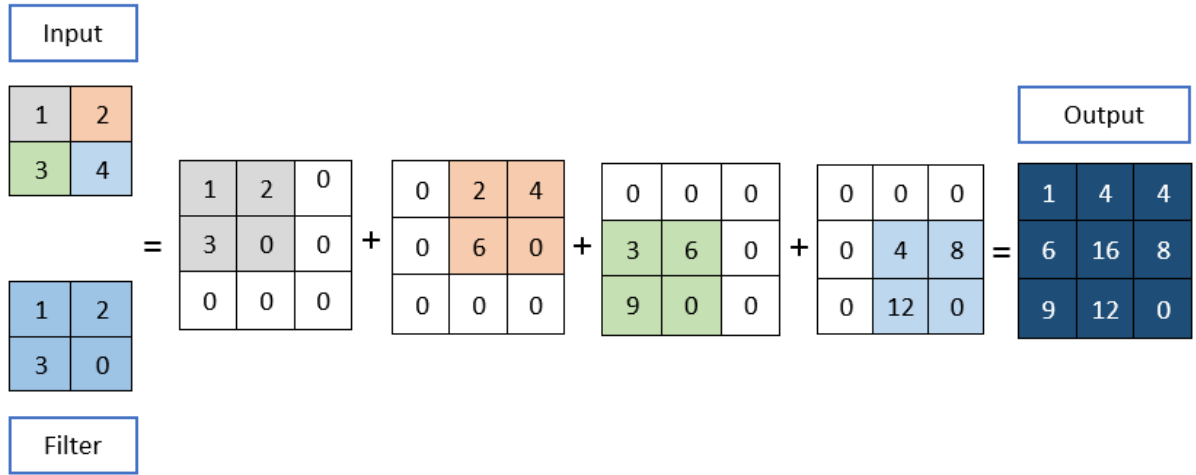


Figure 2.17: Illustration of the transposed convolution operation of 2×2 feature map by a 2×2 kernel obtaining a 3×3 output image. Image retrieved from [57].

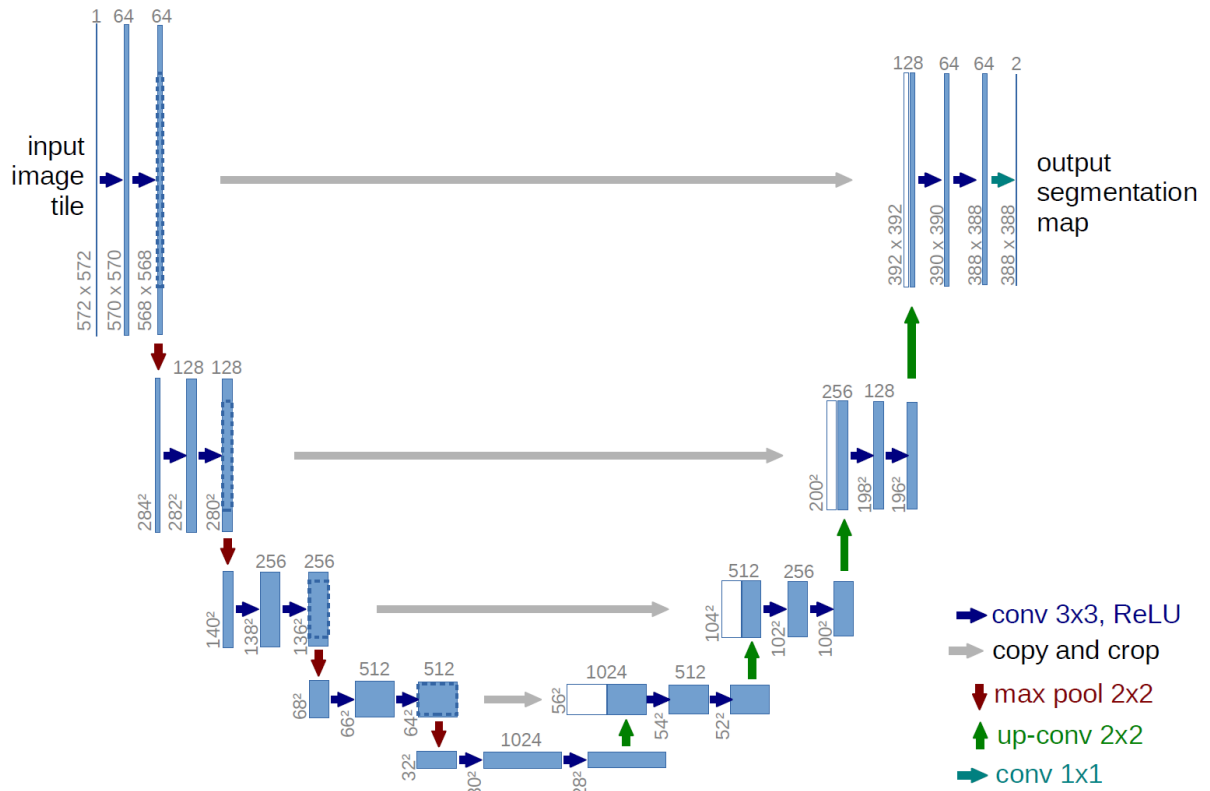


Figure 2.18: Schematic representation of the original U-Net architecture. Image retrieved from [37].

2.3.5 Attention U-Net

The Attention U-Net model is an architecture that applies attention methods to the U-Net architecture described in the previous subsection. It highlights only the relevant actions during training, reducing the

computational resources wasted on irrelevant activation and providing better network generalization.

To have a vision of the motivation and meaning of the implementation of this architecture in the segmentation of biomedical image context, the concept of this attention method will be indexed.

Human biological systems inspire attention mechanisms as they focus on learning relevant parts of the process when processing large amounts of information. Attention means focusing on what we care about and disregarding what is less important.

Although these mechanisms can be classified according to 4 different criteria [58], their implementation in this model concerns the smoothness of attention.

Thus, hard attention is a type of attention mechanism that highlights the relevant part of the image by cropping the relevant part. In other words, the training is done with cropped images of relevant parts of the original image or not so that the training can understand which parts to pay attention to. This implies that part of the training is not differentiable when the model is given part of the non-relevant image, forcing it to resort to reinforcement learning algorithms. Consequently, backpropagation mechanisms could not be used.

Soft attention is a type of attention that can be done in the training process itself, as opposed to hard attention. In the case of image segmentation, it assigns greater relevance throughout the training process to the relevant parts of the image, giving them greater weight, and less weight to less relevant parts, giving them less weight. In other words, the weights are also trained to pay more attention to the relevant parts of the image. Since this process occurs throughout the training process, backpropagation mechanisms can be used.

Attention U-Net uses a soft attention mechanism integrated into the U-Net architecture. They are integrated after skip connections along the up-sampling path, so the encoder has the same architecture as the original U-Net.

The assignment of weights by this mechanism results from the concatenation of two inputs, one that provides spatial information (x) that gives context and that comes from the down-sampling path (skip connections), with the feature map (g) that provides feature representation as rounded parts, sharp edges, texture, etc., coming from the previous layer (deeper layers). So, Attention blocks give up-convolution layers the best of two worlds: The spatial information from the down-sampling path and initial layers (Skip connection) and the feature representation from deeper parts of the network.

The figure below shows the attention gate used by the attention U-Net model.

Since the inputs do not have the exact dimensions, because the g signal always comes from a deeper layer than x , resizing them to apply the arithmetic sum identified in the figure above is necessary, the resizing is done through W_g and W_x convolution operations. In W_g , the convolution is done with a stride of (1, 1) to keep the image dimensions and double the number of filters. In W_x , the convolution is done with a stride of (2, 2) to reduce the image size to half and maintain the number of filters.

With the sum operation, the aligned weights get much larger assigning higher weights to more relevant pixels.

The ReLU activation function is applied to the result of the sum, in which the values of weights ≤ 0 take the value of 0, and for weights greater than zero, their maximum value. Then a convolution with only

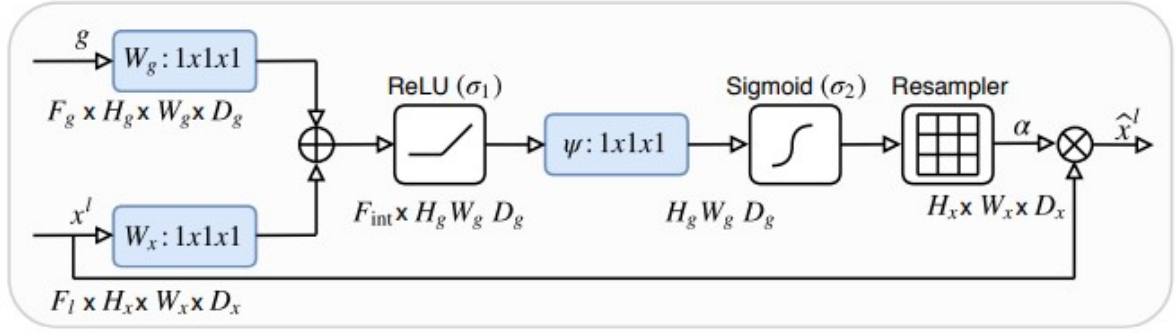


Figure 2.19: Schematic representation of Attention mechanism architecture. Image retrieved from [59].

one filter is applied to obtain a dimension tensor $(h, w, 1)$, and its values are the new weights assigned to each image pixel.

Since these weights are linearly mapped in \mathbb{R} , it is necessary to normalize these values to the scale of $[0, 1]$ to represent a probability. Therefore, the sigmoid function is applied.

Since the final result of the attention mechanism comes from the multiplication of this signal produced by the sigmoid activation function by the input x , both must have the same size. To do so, it will still be necessary to resample the signal produced by the sigmoid, which is done by the up-sample operation. Multiplication by vector x scales the weights based on their relevance.

Chapter 3

State-of-the-art

Research of deep learning techniques presents major advances in medical image segmentation applications nowadays [60].

To understand kidney segmentation in Magnetic Resonance Imaging (MRI) through deep learning methods, it is necessary to understand the representative methods of the state-of-the-art of kidney segmentation. It is an important analogy to understand the motivation for the development of more accurate methods and greater applicability in the practical world, as well as the foundation of knowledge applied in traditional methods that are transversal to deep learning applications.

Thus, this chapter is intended to highlight critical concepts about image segmentation techniques and kidney imaging modalities. Therefore, three sections are proposed: Section 3.1 provides an overview of image segmentation and motivation in a medical context. Section 3.2 contextualizes kidney imaging modalities. In section 3.3, representative state-of-the-art methods of kidney image segmentation are explored, resorting to traditional methods and some proposed works in the deep learning field applied to this task.

3.1 Image Segmentation

3.1.1 Overview

Image segmentation is the process of dividing an image into multiple non-overlapping regions according to certain criteria. The division of the image into a certain region or set of pixels reduces the analysis area. It facilitates and optimizes the process of searching for relevant features according to the imposed criteria [61]. This process results in a set of image segments that cover the original image when rejoined. Relevant image features must be highlighted to make image analysis easier and more meaningful.

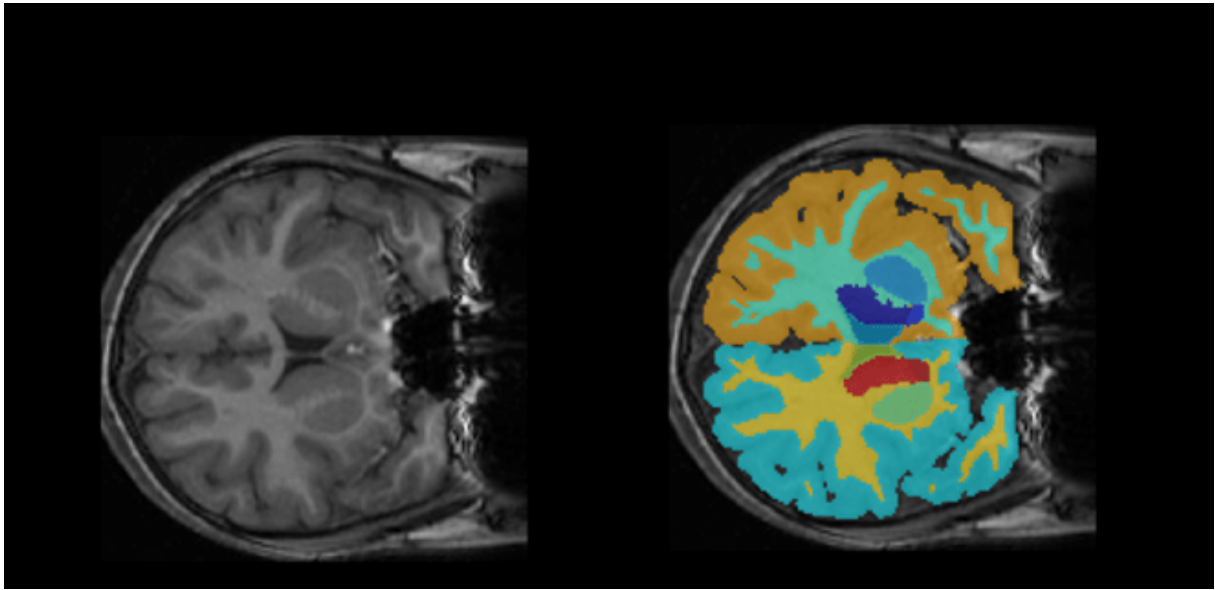


Figure 3.1: Example of medical image segmentation. Image adapted from [62].

The main goal of semantic segmentation algorithms is to make a pixel-wise prediction. Each pixel is classified with a class. Since it requires pixel-by-pixel semantic annotation, it makes the cost of the annotation process time-consuming and expensive [63].

Medical imaging segmentation in the context of deep learning is a task that faces several challenges. Each image segmentation problem requires the formulation of several hypotheses to which different solutions must be attributed, given the raised problem. A widespread problem in medical imaging is the lack of labeled training data [64]. This problem leads scientists to look for various solutions regarding network structure, data augmentation, or pre-processing techniques.

3.1.2 Coarse-to-fine segmentation

Coarse-to-fine segmentation is not yet a well-defined concept in computer vision [65]. Therefore, the mechanisms to be applied in medical image segmentation are not yet established. It is well known that there are several mechanisms to improve image segmentation, from pre-processing to post-processing techniques or recursive deep learning, among others. Therefore, coarse-to-fine approaches aim to

mitigate the limitations regarding pixel-wise annotations in weak semantic segmentation. Branching off this problem from this point on, there are coarse-to-fine approaches at the object label-level, such as bounding boxes, squiggles, or dots, or at the image-level, such as contours or zoom-in. In the method proposed by this dissertation, a coarse-to-fine segmentation at the label-level is studied by applying external contours of the region belonging to the kidney class to improve segmentation in the border region between the two classes. Figure 3.2 shows the differences in the ground truths of the classical approach to the coarse-to-fine segmentation approach of the proposed work. Coarse-to-fine approaches at the label-level have a lower cost to obtain [63] and require less human work [66] than at the image-level.

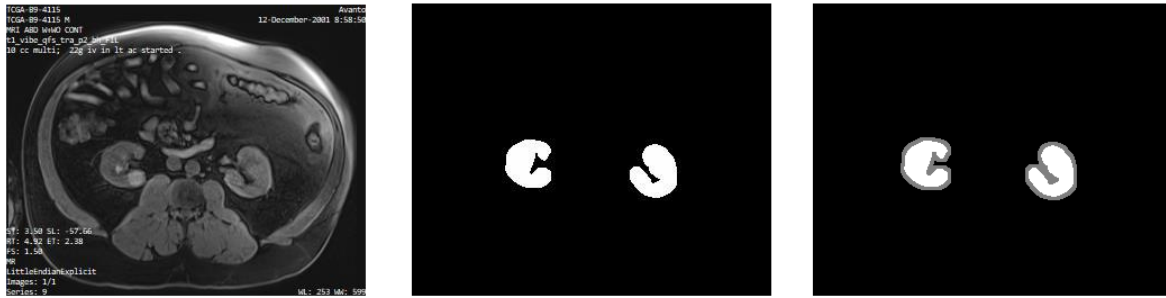


Figure 3.2: Example of changes applied to the ground truth on the coarse-to-fine approach. a) Example of an original image from the dataset used in this work; b) Example of ground truth for binary segmentation; c) Example of ground truth with contour class used in the coarse-to-fine approach.

3.2 Kidney Imaging Modalities

Through kidney imaging modalities, it is possible to visualize various tissues, such as the parenchyma, and compartments, such as the cortex, medulla, renal sinus, or pelvis, to evaluate a kidney functional analysis [67]. In this section, the three primary imaging modalities will be discussed, Ultrasound, MRI, and Computed Tomography (CT). Although they all produce images that allow a kidney generic functional analysis, each provides visualization of different regions or tissues in greater detail. The choice of imaging modality should depend on the evaluation stage and the clinical purpose. Despite this, all allow kidney segmentation [68].

3.2.1 Ultrasound

The image capture through ultrasound is done by the hospital Ultrasonography (US) examination. It is used to obtain the size of the kidney and its morphology. It is a low-cost imaging modality and does not rely on ionizing radiation. Images are produced by sending short bursts of acoustic energy to the patient through a transducer. By propagating this energy by waves through the tissues, they are reflected to the source when they pass through different tissues of different densities and acoustic impedance that when



Figure 3.3: Example image from kidney imaging techniques: a) Example of an Ultrasonography. Image retrieved from [69], b) Example of Computed tomography. Image retrieved from [70]. c) Example of Magnetic resonance imaging. Image adapted from [71].

a receiver detects the reflected waves, it is possible to generate an image based on the wave travel time and its amplitude [72].

As operating modes of US, the default mode is called B-mode or brightness mode, which produces grayscale images. It can enable a harmonic mode in which the transmitted signal has a lower frequency than the received signal, which aims to improve resolution and reduce image clutter [73].

There is a mode called S-mode or Stone mode, which aims to detect kidney stones more easily. The transmitted signal is optimized to excite microbubbles present in kidney stones, and the receiver signal is adjusted to detect these microbubbles. It has better sensitivity, specificity, and precision than the conventional B-mode [73].

Among the advantages of this imaging technique is the non-exposure to ionizing radiation, which accumulates throughout the patient's life and should be as low as possible, and is not recommended for children or pregnant women [74]. They have significantly better results in accuracy and sensitivity in children because the sonar distance to the region of interest is shorter. Grayscale ultrasound imaging is optimized for visualizing different soft tissues, making it a valuable tool for controlling diseases related to parenchymal deficiency. As previously mentioned, it is a low-cost exam and provides a real-time examination [75].

Among the disadvantages of this imaging technique is the significant variability of precision due to a wide range of sensitivities and specificities of ultrasound machines and the user application techniques of performing the exam, leading to varied reference standards. Produces low-quality images due to its low contrast, which creates acoustic shadows and drops that make kidney segmentation a more difficult task [75].

Thus, researchers have tried to study improvements in kidney visualization, increasing the sensitivity and specificity of the ultrasonography devices and decreasing user variability.

Depending on the clinical case, this imaging modality is recommended in a first kidney problems approach.

3.2.2 Magnetic resonance imaging

MRI provides complete kidney function and anatomy information. It allows for precisely visualizing the kidney's state and its constituent parts, such as the cortex, medulla, and pelvis. It will enable the identification of renal lesions, tumors, and small masses but is unsuitable for identifying calcification, including stones [73].

A magnetic resonance image is captured using a magnetic field that aligns the individual's free water protons along the magnetic field axis. A radio frequency (RF) antenna is placed over the area that is supposed to capture the image, which releases energy pulses. This RF pulses align protons to the angle of the magnetic field, causing the protons to spin in phase with each other creating resonance. Milliseconds after RF pulse burst, the nuclei return to resting alignment through various relaxation processes which release RF energy. MRI captures released energy to generate an image [76]. Fourier transform is used to convert the frequency information given by the signal from each location on the image plane to the corresponding intensity levels, displayed as shades of gray in a matrix of pixels.

Several images can be created by varying RF pulse sequence applied or collected. They can be changed by the repetition time (TR), which is the amount of time between successive pulse sequences applied on the same slice, and by the echo time (TE) which is the time between the delivery of the RF pulse and the reception of the echo signal [77].

Several types of MRI sequences exist. The most commonly used are T1-weighted and T2-weighted, and they vary in TR and echo time (TE). T1-weighted images are produced using short TE and TR as opposed to T2-weighted images, which are produced using long TE and TR times. T1, which is the longitudinal relaxation time, is the time constant that determines the rate at which the spinning protons (excited) return to equilibrium and realign with the external magnetic field. T2 (transverse relaxation time) is the time constant that determines the rate at which spinning (excited) protons reach equilibrium or go out of phase with each other, causing them to lose phase coherence between nuclei, spinning perpendicular to the main magnetic field [77]. T1 images usually highlight a type of adipose tissue within the body, while T2 images highlight adipose tissue and water within the body.

The most significant advantage of using MRI for disease control is to provide 3D images with high spatial resolution without radiation. MRI create images with higher contrast between soft tissues. Compared to CT in the medical setting, MRI is capable of detecting flowing blood and cryptic vascular malformations. It can also detect demyelinating diseases and has no beam-hardening artifacts, which allows better visualization of the posterior fossa. There are also some disadvantages. The cost relative to a CT scan is about three times higher. It has longer acquisition times and low temporal resolution and may be subject to motion artifact appearance. Thus, this imaging technique is recommended when 3D visualization is helpful, or the patient has already been exposed to excessively high radiation risks. It is, therefore, the recommended second-line modality after ultrasound [73].

3.2.3 Computed tomography

Images obtained by CT scan provide very similar information, regarding the anatomy of the kidney, to those provided by US. Depending on the clinical aim, it can be performed with different contrast values and imaging timings. By rotating the radiation source and the contralateral detector, it is possible to obtain multiple data points that, when processed by a computer, form 3D images [73].

CT is the procedure for generating computationally 2D x-ray images representing 3D physical objects. A narrow beam of X-ray photons is directed at a particular body area to obtain CT images. The X-ray photon beam is obtained by applying a high voltage to the X-ray tube, in which electrons are excited from a cathode, accelerating them toward the anode. X-rays pass through the body and are partially absorbed by body tissues. Then the photons that reach the detector are measured and converted back into electrons. The images are computationally obtained using the differences in measurements of successfully transmitted X-ray photons. The X-ray tube and array of detectors rotate around the body, capturing multiple X-ray images from different angles. A computer collects data from all detectors to reconstruct a series of cross-sectional images, or slices, of the body. They are obtained through mathematical processes such as back-projection, which uses the measured X-ray intensities to estimate the absorption of X-rays at each point of the body. The quality of the images depends on several factors, mainly the resolution and contrast of the image.

Kidney stones absorb far more radiation than parenchyma, making them more easily identifiable. This imaging modality is much more sensitive than the modalities described in this section for detecting kidney stones. It also allows to obtain information about the composition of kidney stones. Its evaluation involves the assessment of attenuation, obtaining the value of attenuation on the Hounsfield scale through the absorption of radiation by the stone.

It is possible to define the amount of radiation to be used and the area to be studied, depending on the patient's body habits, adapting to the clinical issue in question [78].

Produces more detailed anatomical images allowing better medical evaluations helping diagnose most diseases. CT is, therefore, a highly sensitive and specific technique for image calculations, and it is recommended in cases of clinical emergencies, in stone detection, and in surgical scenarios given the anatomical detailing [79].

The main disadvantages of this imaging modality are the exposure to ionizing radiation, which should be kept to a minimum, and the cost. The cost is about twice of US and one-third of MRI [80].

3.3 Kidney Imaging Segmentation

3.3.1 Traditional Methods

Techniques used before deep learning were established as the primary tool in medical image segmentation. Nowadays, they are called traditional methods, and they can be classified depending on how much human interaction is needed as manual, semi-automatic, or fully automatic segmentation techniques.

Manual methods are intrinsically human, labor-intensive, and time-consuming. They are used in different contexts. They serve to delineate limits of kidney regions in situations where automatic methods are not integrated into the clinical environment or as research tools to obtain a ground truth for automatic or semi-automatic methods [81]. It is, therefore, the technique recommended by the literature to have a high-quality label for the training phase of deep learning techniques when performed by specialists [82]. It can be done by simple image editing software, tools already developed to help delineate limits, or other methods such as Stereology [83].

In addition to its disposable, time-consuming, and humanly laborious nature, it is highly subjective once the inter/intra-observer effects are not easily quantified. However, it is important to refrain from refuting its usefulness when performed by specialists to evaluate the performance of automatic techniques.

Semi-automatic methods require manual initialization performed by an operator or the involvement of post-processing techniques. Typically the applied techniques are thresholding methods, region-based approaches, watersheds, active contours, etc. Each technique tries to overcome the adversities of the others and is often combined in many proposed works to improve segmentation results.

The simplest method is thresholding, which is the choice of an optimum threshold intensity value of the area to be segmented. As a rule, it is necessary to apply several thresholds due to non-homogeneity, low contrast-to-noise ratio values, and image variability. There are several approaches to the problem, but choosing an optimal threshold value is still a classic problem in image processing [81]. An algorithm proposed by Otsu [84] in 1979 is one of the most widely used, which uses the image region histogram to define the threshold that maximizes the inter-class intensity variance.

Region-based approaches group pixels according to the features of neighboring pixels [85]. The operator must define seed points according to the similarity criteria of the regions. This approach has low sensitivity to the signal-to-noise ratio, but the chosen regions must be optimized. However, it becomes an approach that introduces spatial information in the segmentation process. Hanson and Lundervold [86] combined the graph-cuts algorithm as a regularization algorithm with region-based segmentation and with edge information to cluster the pixels, resulting in a robust method that fits well for MRI kidney segmentation.

Watershed algorithms combine region growth with edge information [87]. The intensity of the image is evaluated through an elevated map to find regions where there is flooding (great contrast in intensity) to define two different watersheds. Where two catchment basins start merging, two different regions must be defined. This is an algorithm that essentially serves to optimize the definition of regions [88]. The

worst disadvantage of this algorithm is the over-segmentation and flooding of adjacent regions, when the object contour is not a closed curve. Belgherbi et al. [89] proposed a watershed algorithm for kidney segmentation based on mathematical morphology operations. In the first stage, it is combined with a thresholding method to eliminate the spine from the image and applies morphological reconstruction to detect both kidneys. The watershed algorithm performs kidney segmentation based on image gradient and markers extracted by the morphological operations. It should be noted that the over-segmentation problem can be mitigated through pre-processing techniques [68].

Active contour models usually require the initialization of the contour of the image by an operator. This contour is propagated by an algorithm based on local information of image properties combined with a parametrical representation of the contour. Active contours are considered optimization algorithms. In [90], an active contour framework was proposed where gray-level statistics of an initial ellipse are obtained to parameterize the ellipse. Afterward, a minimization scheme is applied to optimize the ellipse's parameters and segment the kidney using region-based statistics through convex relaxation of the energy functional to achieve a fast kidney segmentation.

Other traditional approaches generate a model of the kidney shape (registration) and combine image registration with measure data for segmentation. The image registration model can be built with a set of manually segmented images. Bokacheva et al. [91] proposed an algorithm that uses 3D dyadic wavelet expansion for MRI kidney segmentation and 3D rigid registration based on the Fourier transform.

The previously described methods can still be considered automated when human intervention is only required at their initialization.

It should be noted that some approaches also use iterative methods, such as clustering by k-means to minimize the square error of predictions, as optimization techniques.

The state-of-the-art of traditional methods evolved with the combination of several algorithms mentioned above, whether using them as pre-processing techniques or optimization problems on kidney segmentation.

3.3.2 Kidney image segmentation using deep learning

In recent years, DL models created a generation of segmentation models with notable performance improvements, causing a paradigm shift in the medical image segmentation field [92].

It was known that convolutional operations have applications in many different computer vision tasks due to their high representational power, and from that point of view CNNs, since they are composed of convolutional layers (see subsection 2.3.2), have become dominant in various computer vision tasks [93] resulting in being related to the state-of-the-art in medical image segmentation. Despite the various approaches to the problem based on segmentation networks, such as regional-CNN (R-CNN), Model Generators and Adversary Training (GANs) or Feature Pyramid Network (FPN), most approaches are U-Net or FCN based. U-Net is already the primary tool when it comes to medical image segmentation tasks [94].

U-Net is a U shape FCN architecture that uses descending convolutions to obtain spatial information

to compose a map of low-level features and ascending convolutions to make use of this map to get images back to their original size with detailed object boundaries. In section 2.3.4, U-Net architecture is described.

Several frameworks have used U-Net or variations of it for kidney segmentation, moreover combination of U-Net with FCN, VGG among other segmentation techniques were applied to build more complex algorithms. So, several methods have been proposed for kidney segmentation through deep learning.

Most recent approaches emerged with the KiTS19 and KiTS21 challenges that aimed to accelerate the development of reliable methodologies for semantic segmentation of kidneys and kidney tumors. In the 2019 challenge (KiTS19), a dataset was provided with the annotated ground truth of semantic segmentation for abdominal computed tomography (CT images) of the arterial phase of 300 patients with renal cancer who had undergone partial or partial nephrectomy radical, in which 210 images were destined for the training and validation process while the remaining 90 for testing. In the 2021 challenge (KiTS21), they expanded the set of images that were made available to 489 CT images that are used for the training set. Thus, many approaches present results for the semantic segmentation of the kidney in CT images. Several investigators suggest that algorithms that show successful segmentation on CT images may also show promising results on MRI images.

As far as MRI image segmentation is concerned, the state-of-the-art is much more limited. Only a few renal MRI datasets are available for investigation, making developing new works difficult.

In this section, some proposed methods of greater scientific relevance are presented. Firstly, one-stage and two-stage approaches show results with CT or SPECT images. Then, proposed works from 2015 onwards that represent the state-of-the-art of kidney segmentation in renal MRI by deep learning approaches are reported. The input and sample size of the datasets used in each work is also specified.

The development of new methodologies in MRI images should be further studied given the advantages that this image acquisition methodology has over CT images, preventing patients from being exposed to high levels of radiation.

One-stage methods: One-stage methods are implemented to segment images directly from the whole image [95].

An example of a fully CNN used for kidney segmentation is the work proposed by Andriy Myronenko [96], who, based on the KiTS19 dataset, presented an end-to-end framework for kidney segmentation and tumors.

Efremova et al. [97] compares the results obtained by U-Net and LinkNet-34 combined with Resnet-34 with pre-trained weights from ImageNet to reduce the time of convergence and overfitting, focusing a more careful approach on the learning process rather than on the complexity of the network. This approach has shown success in a wide range of applications in computer vision, with the main focus on kidney and liver segmentation.

Sharma et al. [2] presented a deep learning-based approach where using a basic encoder-decoder model adapting VGG-16 with batch normalization architecture for solving the internal covariance shift problem.

RAU-Net, which stands for Residual Attention U-Net, was proposed by Guo et al. [98] and specifically designed to target renal tumors. This approach presents results using the cross-entropy and the dice function as loss functions but shows a lack of generalization ability.

Zhou et al. [99] developed a Nested U-Net for medical image segmentation from CT images where the encoder and decoder are connected through a series of nested, dense skip pathways to reduce the semantic gap between the encoder-to-decoder feature map.

Xie et al. [100] developed, from the typical U-Net architecture, the Res NeXT U-Net (SERU), which takes advantage of SE-Net, ResNeXT and U-Net. The encoding path is not done by the typical convolution layers but by combining SE-Net and ResNeXT to optimize the feature extraction carried to the U-Net decoder by the skip connections.

Heo [101] presented results in the KiTS19 challenge using the U-Net as a segmentation network but using the sum of the dice loss function with the focal loss function as the loss function to solve the class misalignment problem by minimizing this function of loss.

Two-stage methods: The main objective of two-stage methods is to overcome the class imbalance problem [95].

Cruz et al. [102] presented a two-stage approach where a dataset scope reduction was first applied using AlexNet to remove the hard-to-classify examples and then performed a semantic segmentation using U-Net. Post-processing techniques were also applied to reduce false positives and delineate the kidneys, but the results were worse than expected.

Zhang et al. [103] presented a cascaded two-stage approach using a 3D-FCN in which, first locates the kidney and cut off the irrelevant background to reduce class imbalance and computation cost, in the second-stage segment, the kidney, and tumor on the cropped patch.

Hatamizadeh et al. [104] improved the edges representation in the learned feature maps with a module that can be combined into any encoder-decoder architecture, such as U-Net, to add the task of identifying the edge to the original network. Several investigations have already been made with this module introduced but without improving the results.

Zhao et al. [105] proposed a multi-scale supervised 3D U-Net designed for segmentation of kidneys and renal tumors from CT images in which deep supervision was combined with an exponential and logarithmic loss to increase the efficiency of the training process. A method of connected-component based post processing was also introduced in his approach.

Santini et al. [106] proposed an approach that combines Res-Net with Res-U-Net in a multi-stage approach. The approach, called EMS-DLA, shows promising results and may improve when benign cysts are better understood.

Chen and Liu [107] proposed a coarse-to-fine approach for the segmentation of kidneys, tumors, and cysts in the abdomen from CT images based on a multi-stage refinement strategy. In their work, the images were first submitted to a Rough ResSENormU-Net to be normalized, to identify the kidney's position, and to be cut and resized. Then a FineDenseTransU-Net was trained for semantic segmentation.

He et al. [108] proposed a two-stage cascaded method with multiple decoders. They used U-Net to locate and extract the kidney and a Multi-Scale Discriminative NET for segmentation.

Wei et al. [109] took a new and unique approach by introducing SeResUNet. This work presents results in abdominal CT images for segmenting kidney and renal tumors. Still, it presents a perspective of applicability in segmenting other organs. It also says that using an encoder-decoder architecture such as U-Net, ResNet can be used to deepen the encoder network to minimize the degradation of the training process and accelerate its convergence.

Cheng et al. [110] suggested an improvement on 3D SEAU-Net to perform a multi-class segmentation. The model assembles a Residual network, dilated convolutions, squeeze-and-excitation network, and attention mechanism to the U-Net. The multi-class segmentation is decomposed into two easier binary segmentations.

Proposed works on MRI images: Kline et al. [111] proposed an approach in which 11 different U-Net configurations were tested to segment the kidneys in images from patients with polycystic kidney disease (PKD). States that to run this approach, a large dataset is required to achieve successful segmentation, as the dataset was divided into 11 different and random datasets. Thus, this work presents results for the individual network and a multiple-observer network. The total kidney volume (TKV) is calculated from the segmentation of 2D images.

- **Input size:** 2D:256 x 256
- **Sample size:** 2000 images (Train and Validation) + 400 (Test)

In work presented by van Gastel et al. [112], Kline et al. [111] architecture configuration was enhanced to obtain the individual semantic segmentation of each kidney and then calculate the TKV with the propose of help in the diagnosis, treatment, and control of ADPKD. A 5-k cross-validation was included to validate the results.

- **Input size:** 2.5D:256 x 256 x 3
- **Sample size:** 352 images (Train) + 88 (Validation) + 100 (Test)

Bevilacqua et al. [113] and Brunetti et al. [114] proposed different approaches for kidney segmentation on patients with ADPKD. These compared two approaches based on CNNs in which they obtained a pixel-wise classification without requiring hand feature extraction. In the first approach, they obtained automatic segmentation without any pre-processing. In another approach, they opted for a two-stage approach in which, in the first stage, they trained an R-CNN to capture the region of interest (ROI) and then applied a mono-objective genetic algorithm to define the number of encoders, the structure and connected final layers of a CNN to segment the kidneys, but the accuracy of these methods did not exceed 90%.

- **Input size:** 2D:256 x 256 x 3
- **Sample size:** 526 images

Other methods were developed to obtain the semantic segmentation of the kidney and various organs captured through abdominal MRI, such as the liver, pancreas, or stomach. For example, Bobo et al. [115] presented an FCN that shows results for the segmentation of each kidney and presented better results with this CNN architecture than multiple approaches.

- **Input size:** 2D:512 x 512
- **Sample size:** 36 images (Train) + 9 images (Test)

Chen et al. [116] investigated the multi-organ segmentation problem. The segmentation of various organs presented by the abdominal MRI is very important to define the adaptive radiotherapy treatment to combat abdominal cancer. Thus, this method was called ALAMO (Automated deep learning-based Abdominal Multiorgan segmentation). They compared U-Net with Dense U-Net and concluded that Dense U-Net performed better.

- **Input size:** 2.5D:256 x 160 x 20
- **Sample size:** 66 images (Train) + 16 (Validation) + 20 (Test)

Summary: To date, what can be gauged from the dice coefficient results analysis of presented works concludes that:

- In CT images, the 3D models outperform the results obtained for 2D images;
- In MRI images, the 2.5D models outperform the results obtained for 2D images due to the differences in sample size, once 2D models are trained with larger slices;
- In MRI images, it has been found that 2D models tend to have lower segmentation dice;
- Better performances are obtained using large datasets.

The most widespread segmentation architecture for kidney segmentation is the U-Net shaped, and sample sizes are smaller compared to the other organs' segmentation [95]. Despite the multiple investigations, kidney segmentation is still a challenging problem since there are severe imbalanced data problems.

Chapter 4

Methodology and Dataset

In this chapter, we describe the dataset used and the methodology addressed to the problem of kidney segmentation in abdominal MRI images. First, we must recognize our dataset's typology, context, and limitations. This is so that one can understand the answers given by the proposed methodology to the problems inherent to the existing limitations for developing precise methods with scientific validation.

Therefore, we propose five sections in this chapter: The section 4.1 describes the dataset provided by the COST action in terms of its typology, acquisition methodology, and limitations. All modifications to which the dataset was subjected, from obtaining the ground truth to the training process, are also presented; the pre-processing applied and the detailed description of the data augmentation techniques used. The section 4.2 details the architectures studied by the proposed work. In section 4.3, the tuning parameters of the training process are presented, and in section 4.4, the methodology for validating the results obtained is exposed.

4.1 Dataset

4.1.1 Dataset description

MRI images were taken from a database made available in the COST action CA16103 – “PARENCHIMA – Magnetic Resonance Imaging Biomarkers for Chronic Kidney Disease”, aiming to obtain kidney volumetry in 2D MRI images. Accessing volumetry in 3D images is not justified since the longitudinal resolution of the kidneys is lower than the spatial resolution. The number of parameters associated with the training process of 3D image segmentation is much greater, which requires more time and greater computational power.

The dataset consists of 21 abdominal MRI images (.dcm files) and 21 ground truths obtained by manual kidney segmentation (.tif files) of each MRI image. All images were obtained from one patient.

MRI images were taken using the T1 VIBE acquisition methodology. Image acquisition through T1 means that magnetization has the same direction as the static magnetic field. They are taken with a Volumetric Interpolated Breath-hold Examination (VIBE) sequence that allows dynamic and high-resolution images in 30 seconds of apnea to minimize motion artifacts caused by respiratory movement but with high intrinsic contrast resolution for soft tissues. It has the advantage of improving the resolution of the Z-axis (see figure 4.1), which makes it possible to obtain high-quality multiplanar images and 3D reconstruction [117]. The VIBE acquisition methodology is a form of volumetric imaging using rapid 3D gradient-echo sequences [118]. All images have a slice thickness of 3.5 mm, a repetition time between 4.92 ms and 7.29 ms, and an echo time of 2.38 ms.

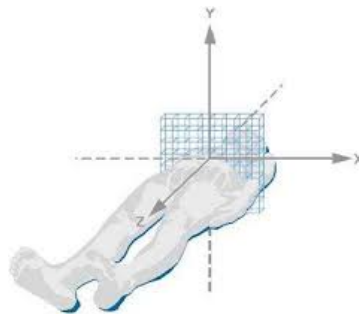


Figure 4.1: Axis orientation of MRI image.

Specialists in the clinical environment captured the images, and the COST action verified the quality. The original size of 11 of 21 images is 320x260, and the remaining 10 with an original dimension of 320x240.

Given the limitations inherent to the size of the dataset, data augmentation techniques were used so that the training process would produce results of scientific relevance, introducing greater robustness.

The present study also aims to compare the results of a segmentation with and without a contour line, the ground truth in the model without the contour line is annotated with two classes: the kidney, manually annotated, and the background. In the model with the contour line, three classes are presented: the kidney class, manually annotated; the contour class, obtained from the manual annotation of the kidney class and the background class with the imbalanced distribution.

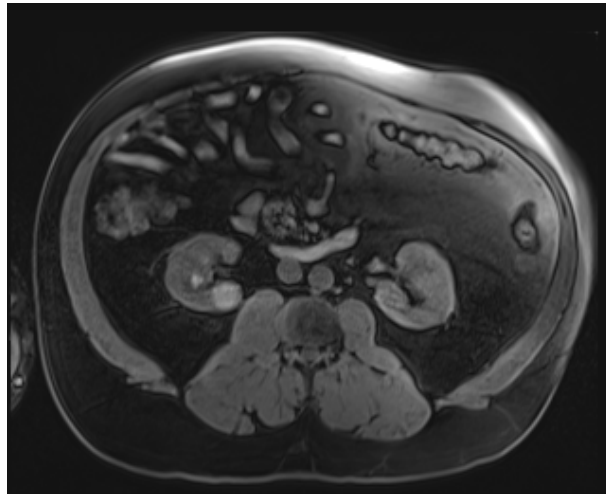


Figure 4.2: Original abdominal MRI image.



Figure 4.3: Ground-truth image of kidney manual segmentation.

4.1.2 Ground-truth

Obtaining the ground truths was performed by manual segmentation by an image processing researcher with experience in medical image segmentation. A ground truth example is illustrated in figure 4.3.

Obtaining the ground truth generates red binary images ([255,0,0]), so MATLAB was used to remove the red pixels to turn them into black binary images ([0,0,0]).

Firstly, the images from .tiff format were converted to .png to standardize a format between the dataset of the original images and masks. Then, the resizing was performed to the size defined for the training process, 256x256. To prevent the image transformations caused by the interpolation of the image's pixel values in the resizing scaling process, the masks were subject to a binarization by the threshold of 0.5 after resizing using the anti-aliasing filter to prevent the appearance of aliasing artifacts.

4.1.3 Pre-processing

The applied pre-processing is based on preparing the data to implement the deep learning model. That is, only image format conversion and resizing were applied. The images were converted from the .dicom format, a format typically used in medical imaging due to it being the standard format produced by most medical imaging devices that facilitates the storage and reading of metadata associated with the exam, to a .png format to maintain the same format as the masks dataset. It should be noted that the .jpg format was excluded because it compresses the images so that an exact RGB pixel can have different values, which would affect the comparison with the results predicted by the model. The images were resized to 256x256, the input data format defined for the training process.

Other pre-processing techniques, such as noise reduction, were excluded as they were inappropriate for the problem. If, on the one hand, they could make the model easier to learn the important features of the kidney, on the other hand, they would bring disadvantages for the generalization of results. Among the disadvantages would be, for example, the limitation of the application of data augmentation techniques. Grayscale variation techniques could be compromised because applying them to, for example, images filtered by a Gaussian filter could result in images not representative of the actual data [119]. There would be a loss of information by the removal or alteration of features of the segmentation region. It would also make the model less robust and more susceptible to overfitting because the training set would be normalized and, therefore, have less ability to generalize to new data. This is a problematic issue in medical image segmentation as there is significant variability in these types of images due to differences in anatomy, pathology, or imaging parameters [120].

4.1.4 Coarse-to-fine segmentation

In section 3.1.2, the segmentation refinement technique is explained, where it is defined that a contour class would be introduced to understand whether it would have a better performance in defining the border region between the kidney and the background.

The contour implementation was applied from the manual segmentation of two classes. From the mask obtained manually, a 3×3 kernel analyzed the image and eroded one pixel from the kidney class and two from the background class. Three contour pixels were defined so that the model had the flexibility to understand each class and that there was no loss of detail between the edges of the kidney.

The implementation is detailed by pseudo-code in section A.1.

4.1.5 Data Augmentation

Data augmentation techniques are recurrent and recommended by the literature on medical image segmentation problems. The primary purpose is overcoming the problem of a limited dataset and introducing robustness to the learning process. Considering the inherent limitations of the dataset, with only 21 images, inversions, rotations, translations, and variations in the grayscale were applied to the original images to create new data.

The transformations and parameterizations applied to the original images are shown in table 4.1.

Table 4.1: Data augmentation transformations.

Transformations		Parameters
Geometric	Inversion	Horizontal
	Rotation	$[-5^\circ, 5^\circ]$
	Translation	X axis $[-7\%, 7\%]$
		Y axis $[-3\%, 3\%]$
Gray Scale level	Brightness	$[-10\%, 40\%]$
	Contrast	$[-10\%, 40\%]$

Data augmentation was performed with the assisting and publicly available library Albumentations [121]. This library allows the creation of as much data as required from original images as long as parameters are established.

Considering the trade-off between available computational power and overfitting prevention, it was established that creating 1000 images would bring the capacity for feature extraction necessary for kidney segmentation to the training process. An empirical study was first performed with 500 data augmentation images in which the results are not presented because a conclusive value was not achieved. Creating images by data augmentation is random within the indicated parameters to validate the training process's robustness, with any image having any combination of the defined transformations.

Thus, the 20 images intended for the training process were transformed into a dataset composed of 1020 images.

The mirroring technique requires the geometric transformation of the original image. It has great interest since renal MRIs are centered on the abdomen, allowing the capture of both kidneys in a practically mirrored manner. By inverting an image, we obtain a new image in which the region of interest is

similar but has different image features. Figure 4.4 shows an example of mirroring where, although the images are the same, the image features differ in the regions of interest.

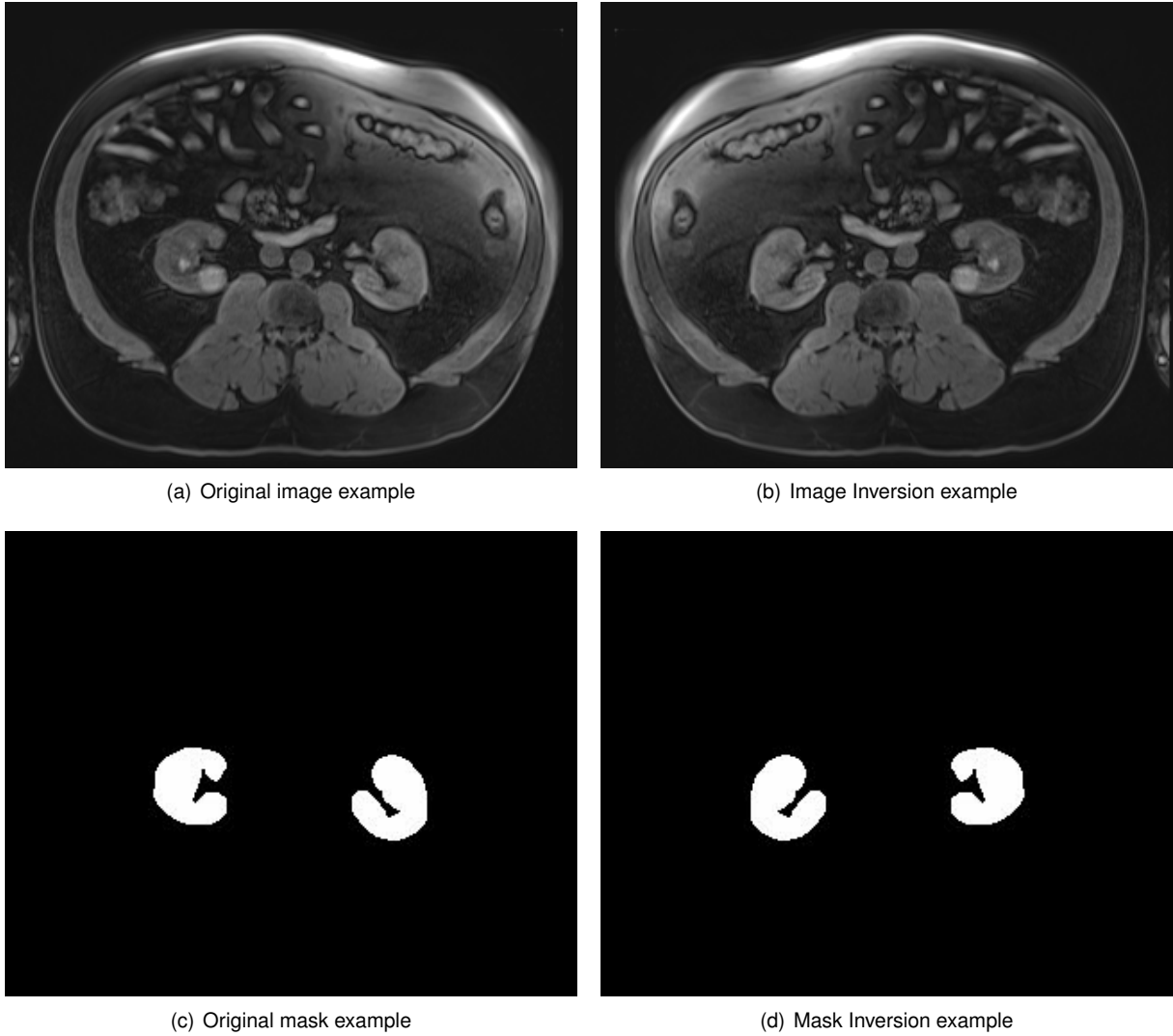
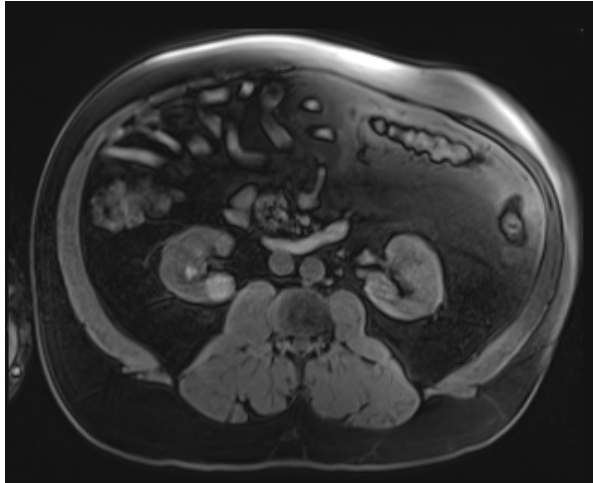


Figure 4.4: Data augmentation inversion example

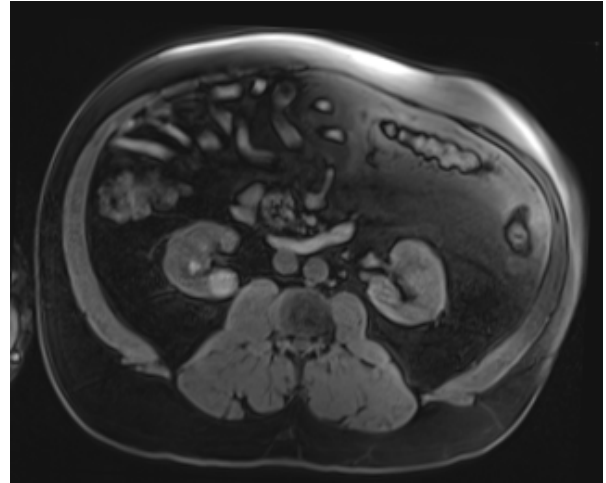
Rotations and translations are two applied geometric transformations that change the image's region of interest, which is manageable given the parameterization used. Figure 4.5 shows an example of mirroring where, although the images are the same, the image features differ in the regions of interest.

The literature also suggests creating images with variations in the grayscale, [37]; these transformations are only applied at the image level. Thus, images with different levels of brightness and contrast were created. Figure 4.6 shows an example of the brightness level transformation and an example of the contrast level transformation that contributed to the data augmentation process.

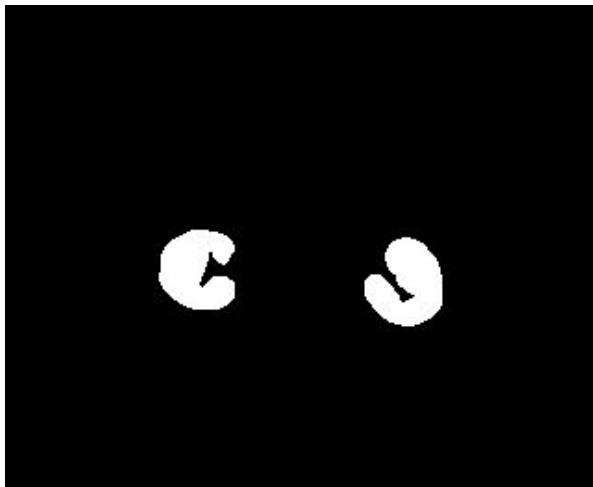
Although abdominal MRI techniques are already standardized, they always have an ambiguous character depending on the anatomical morphology of the patient, the technician, or the machine in which the examination is performed. Thus, data augmentation techniques are very important in obtaining new data for the training process, but they are also important in introducing robustness and greater tolerance to the learning process. The geometric transformations were intended to make the model more



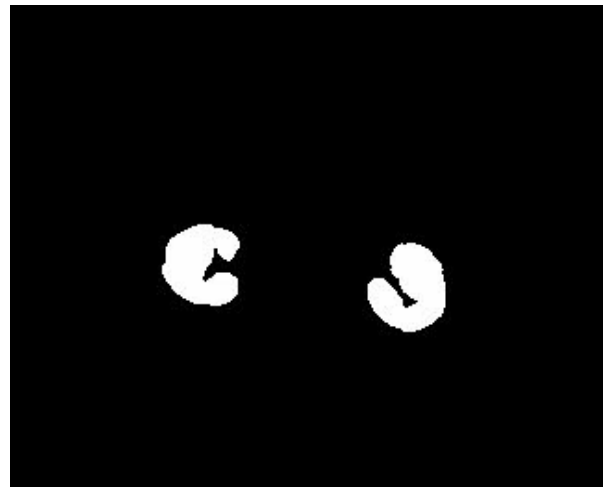
(a) Original image example



(b) Image -5° rotation example



(c) Original mask example

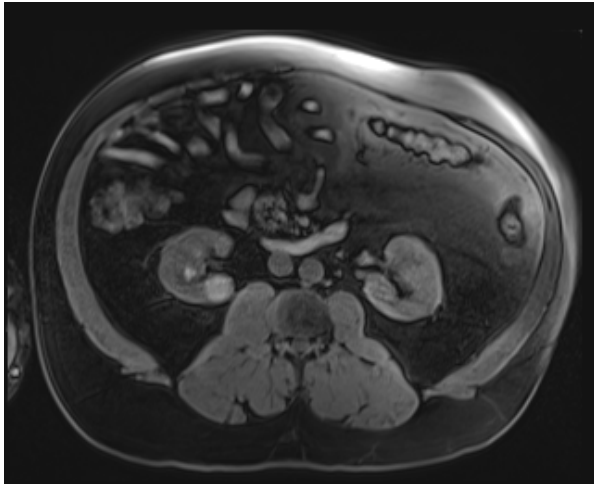


(d) Mask -5° rotation example

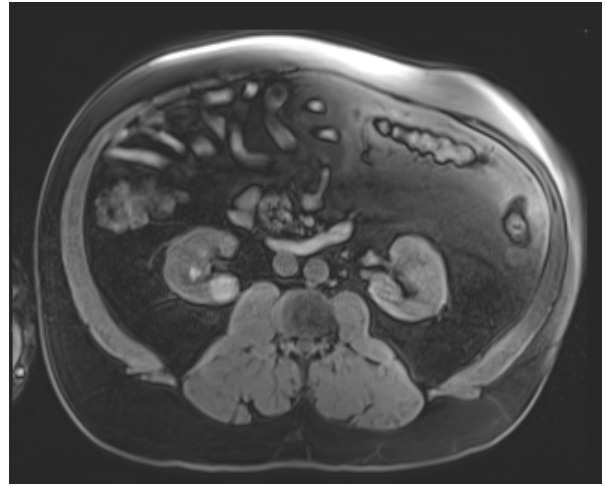
Figure 4.5: Data augmentation rotation example

tolerant to variations in the images' orientation and the shape of the kidney structure. The grayscale transformations helped the model to become more tolerant of different kidney tissue densities.

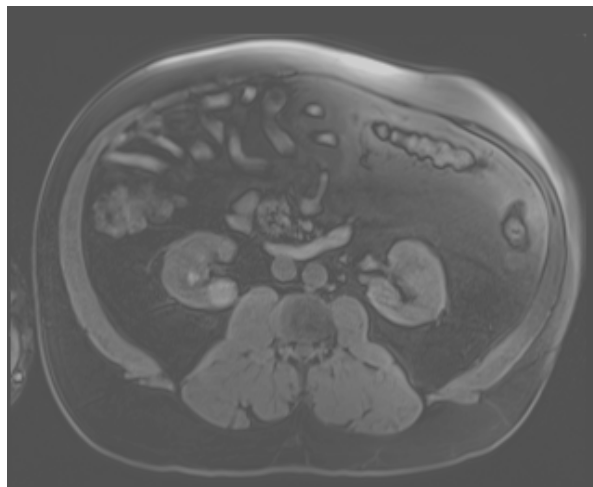
The pipeline shown in Annex A.2 illustrates all the processes to which the original dataset made available by the cost action was subjected to the training process.



(a) Original image example



(b) Image 3% increase in brightness level example



(c) Image 3% decrease in contrast level example

Figure 4.6: Data augmentation grayscale level example

4.2 Architectures

U-Net and Attention U-Net are the two architectures compared in this study. Currently, the biggest problem for a successful implementation of medical image segmentation algorithms lies in the need to have large datasets. A particular benefit of U-Shaped architectures is that they do not require a large data set compared to other architectures, essentially because it combines information from the feature map with the information obtained by reducing the spatial reduction. Another common problem in kidney segmentation comes from class imbalance. Aggregating an attention mechanism in U-Net to focus learning on the most important image feature can help overcome this problem.

4.2.1 U-Net

An adaptation of the original U-Net architecture as a segmentation model was established. This adaptation begins with an input of $256 \times 256 \times 1$, meaning it receives grayscale images (1 channel) of 256×256 . Furthermore, as a typical U-Net architecture, it has the U-shape, meaning it has two paths: the contraction and the expansion paths (see subsection 2.3.4). Like the original U-Net, this adaptation was built with four levels, meaning that each path has five layers, in which the two paths share the fifth layer, also called the middle layer.

For the contraction path, composed of five layers, including the input layer, two convolution operations are performed per layer with convolutional kernels of 3×3 and stride 1. A dropout rate and batch normalization as the regularization layer are applied between each convolution operation. This dropout rate takes the value of 0.1 in every layer. After convolutional operations, a 2×2 Max-Pooling operation is performed on each layer with a (2,2) stride, excluding the middle layer. Compared to the original U-Net, the number of initial filters in convolutional operations has been reduced from 64 to 16. It is doubled from layer to layer by the Max-pooling operation, meaning that we end up with 256 channels on the fifth layer. As an activation function of convolutional operations, the ReLU function was performed (see subsection 2.2.4).

In the expansion path, which is equally composed of 5 layers, including the output layer, each layer is dimensioned to be mirrored to the contraction path layers in terms of the number of filters, convolution operations, and dropout rates. Each layer has two convolutional operations with convolutional kernels of 3×3 and stride 1. Between each convolutional operation, a dropout rate of 0.1 and a batch normalization is performed in every layer. After convolution operations, an up-convolutional operation is performed with convolutional kernels of 3×3 and stride 2 to halve the number of channels. As an activation function on convolutional operations, ReLU functions were applied, taking into account that on the output layer, the sigmoid activation function was used (see subsection 2.2.4) to produce a classification by classes in each pixel and the soft-max was used to perform the multi-class segmentation. A concatenation of information obtained in the feature map of the contraction path through skip connections, with the result of the up-convolutional operation obtained on deeper layers, is performed, leading to four concatenation operations. In the output layer, a 1×1 convolution is made to map each component of the 16 feature vector components to each class. It is also worth mentioning that all convolutional operations maintain

the same padding.

The skip connections are represented in the illustration of the U-Net adaptation architecture of figure 4.7 as all the operations performed and the number of channels per layer.

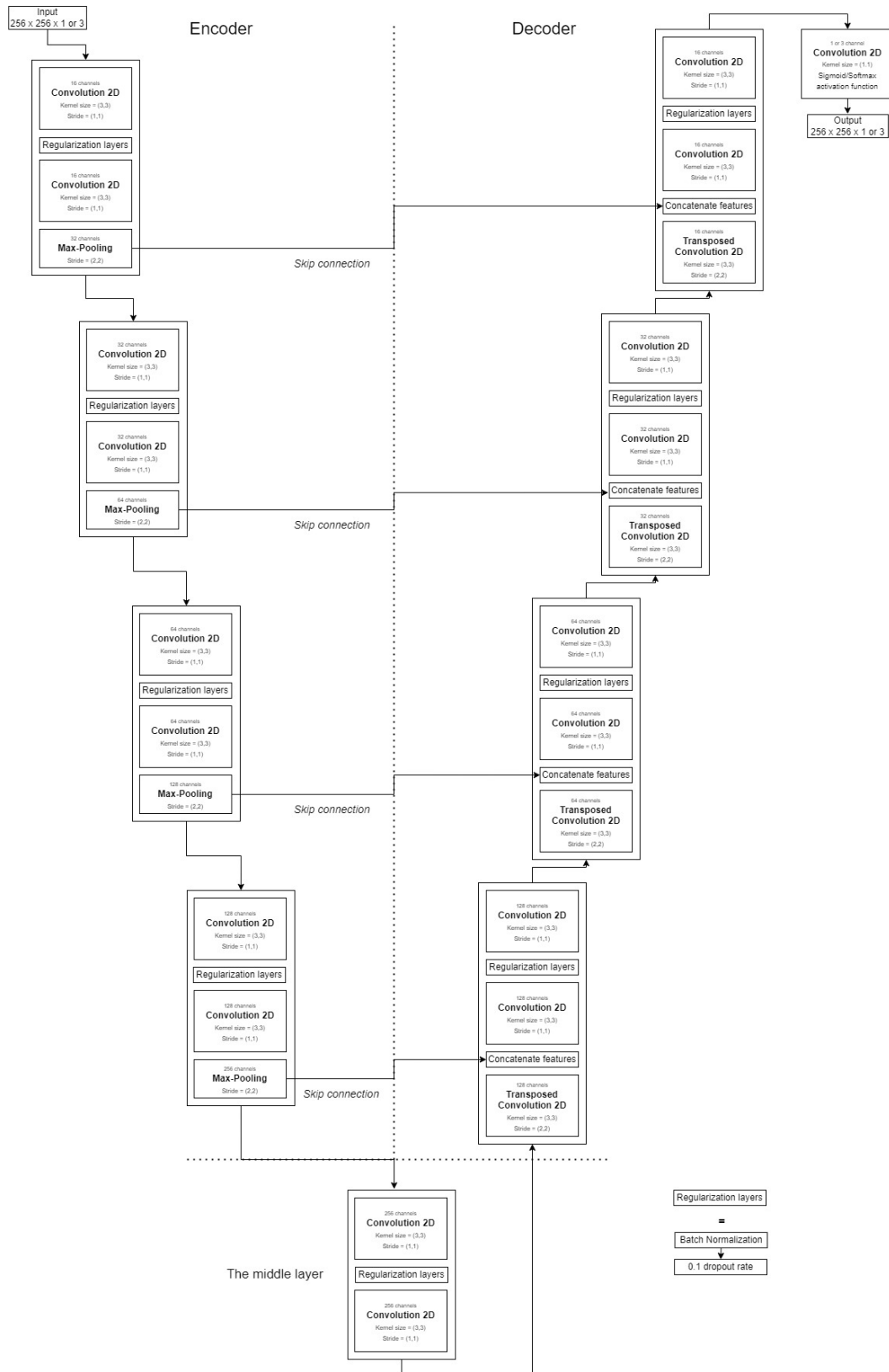


Figure 4.7: Illustration of U-Net architecture used in the proposed model.

4.2.2 Attention U-Net

An adaptation of the Attention U-Net model proposed by Oktay et al. [59] was chosen as a comparison model for the implementation developed for U-Net. This architecture is all similar to U-Net, only introducing an attention mechanism through attention gates (AGs). This AGs are integrated into skip connections along the expansion path, allowing the maintenance of the original U-Net architecture along the encoder path. AGs are intended to automatically learn how to focus learning on target structures, suppressing feature activation in irrelevant regions, and improving the sensitivity and accuracy of predictions (see subsection 2.3.5).

This adaptation of Attention U-Net begins with an input of $256 \times 256 \times 1$, which receives grayscale images (1 channel) of 256×256 size. To validate the comparison of the results produced by the U-Net architecture described in the previous section and Attention U-Net, both must have similar configurations. Thus, the implemented Attention U-Net was built with the same four levels, the same number of filters at the input, the same settings of convolution operations, max-pooling, up-convolutional, and the same dropout rates, batch normalization applied, and maintaining the same padding in convolutional operations. ReLU functions were used as activation functions of convolution operations, excluding the convolution operation of the output layer in which the soft-max or sigmoid function was performed, depending on whether it was used for binary or multi-class segmentation, respectively.

Once the implementation was built with four levels, attention operations were performed four times over the four skip connections received by the expansion path or decoder.

This attention mechanism is composed of an arithmetic sum operation between the x signal produced by the contraction path (skip connections) with the feature map g produced by an up-sample operation from previous layers (deeper layers). Since x and g have different dimensions, it is necessary to resize them. In g a convolution operation with $(1, 1)$ stride is applied to maintain the image dimension and double the number of filters. In x , a convolution operation with $(2, 2)$ stride is made to halve the image size and maintain the number of filters. A ReLU activation function is applied to the sum result to normalize the weights $w(0, \max)$. A convolution operation is also performed with only 1 filter to obtain a $(h, w, 1)$ dimension tensor in which its values are the new weights associated with each image pixel. Since the weights are intended to be mapped in \mathbb{R} , it is necessary to map them on the $[0, 1]$ scale so that they can represent a probability applying a sigmoid function (see subsection 2.2.4). Finally, it is necessary to resample the signal produced by the sigmoid to multiply with the x signal that scales the weights based on their relevance. Figure 4.8 depicts the architecture of the attention mechanism used, and figure 4.9 presents the illustration of Attention U-Net architecture, both with its operations.

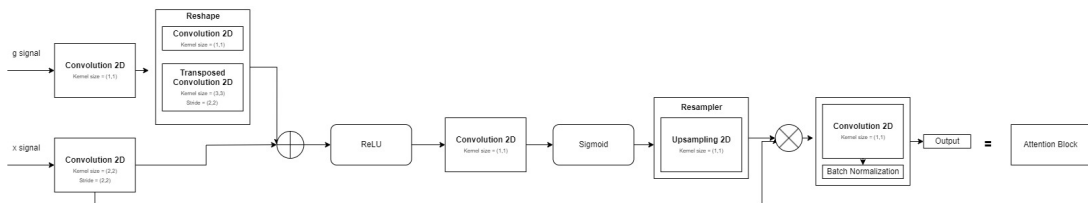


Figure 4.8: Illustration of attention mechanism architecture used in the proposed model.

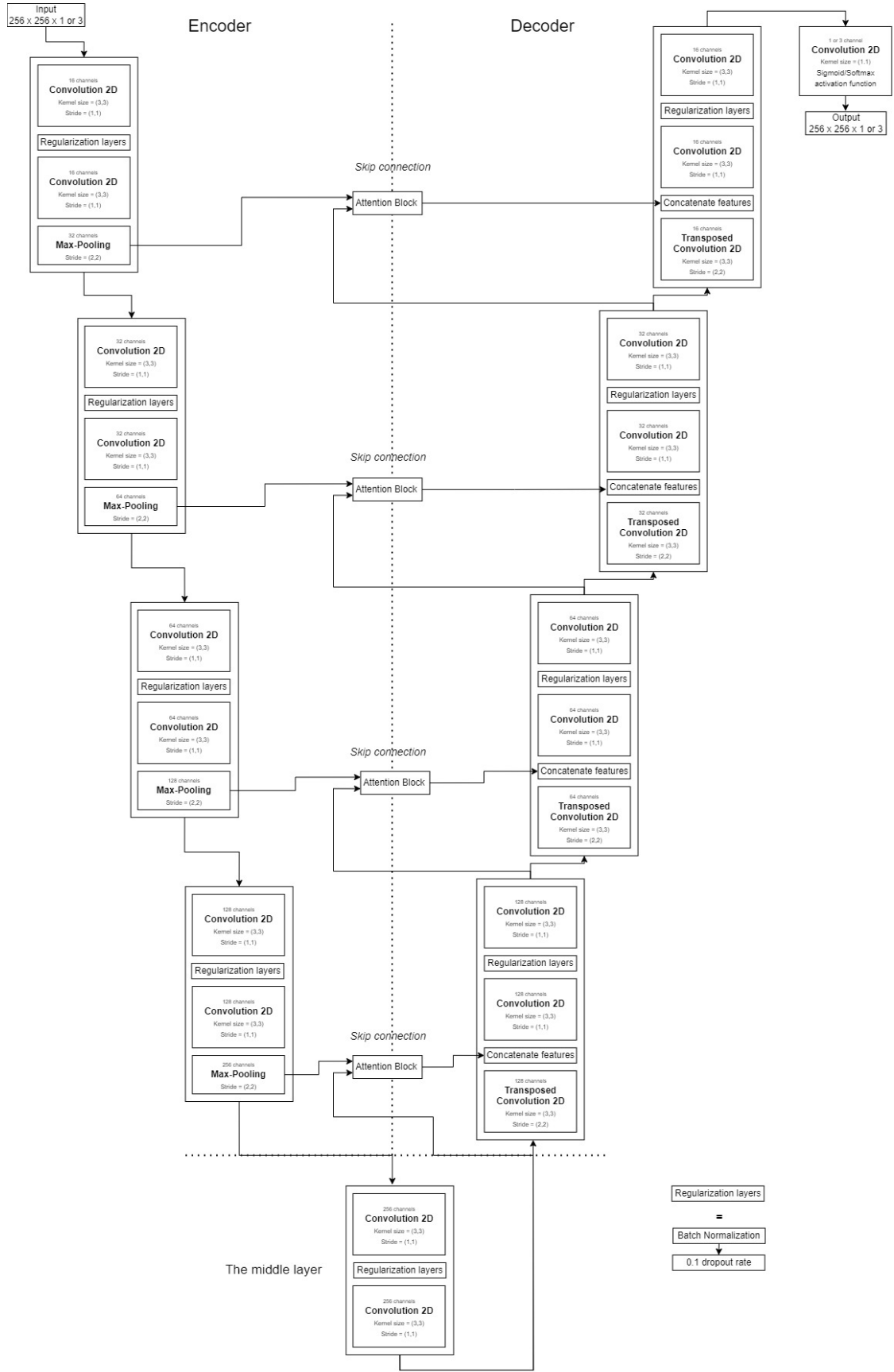


Figure 4.9: Illustration of Attention U-Net architecture used in the proposed model.

4.3 Training

This section discusses the U-Net and Attention U-Net training model configurations. The hyperparameters and evaluation metrics used will be clarified.

The training of the models was carried out during 75 epochs with a batch size of 8. The number of epochs was selected using empirical experiments, as the literature suggests [35, 37, 122, 123], to guarantee the convergence of the loss function without ever reaching overfitting. The training process lasted 75 epochs in order to balance the management of computational resources and time with the guarantee of model convergence. The batch size was chosen in order to prevent memory problems.

The Adam optimizer was used because image segmentation is a complex computer vision task and requires a large dataset. The literature suggests that this is an optimizer for large datasets and complex models compared to stochastic gradient descent with momentum (SGDM) [124]. It is a stochastic gradient descent optimization algorithm and dynamically adjusts timing parameters based on gradient descent updates. Considering the loss functions covered by the study, where pixel-wise differences between predicted and actual segmentation are calculated, adjusting the momentum parameters based on gradient descent updates helps to minimize loss functions. It improves the convergence rate even with noisy data with sparse gradients.

Cross-Entropy and Focal loss were the loss functions used in this study. The choice of the loss function is a crucial point in deep learning architectures as they determine how the model is optimized throughout the training process and measures its performance. They have implications for the model's ability to generalize. In subsection 2.2.5, the used loss functions are discussed, together with the reasons that make them the most suitable for evaluating kidney segmentation.

Data augmentation introduces robustness and new data tolerance to the model learning process by ensuring generalization ability.

A dropout rate was established as a regularization technique, reducing the interdependence of neurons to prevent memorization, standardization, and overfitting. It also makes the model more robust in the feature representation and removes the prominence between the training and test results.

Batch normalization was also used as a regularization technique, which helps to mitigate the displacement of internal covariance, adapting the model to different input distributions [125].

The initial learning rate is 10^{-2} ; we opted for an approach to the decay of the learning rate by step decay, in which it decays at a rate of 0.1 every 15 epochs, an approach that allowed us to avoid the model getting stuck at local minima, improve convergence by accelerating the training process and allowing unbiased training stability [126].

The table 4.2 summarizes the hyperparameters used in the proposed implementation.

Three evaluation metrics, Accuracy, Dice coefficient, and Jaccard index precision and recall, were used to evaluate the results produced and validate the learning process of the proposed algorithm. These have been as optimized as possible. In subsection 2.2.6, the metrics used are discussed as to why these are the most suitable for evaluating kidney segmentation.

Table 4.2: Hyperparameters used in the proposed implementation.

Model Parameters	U-Net	Attention U-Net
Starting LR	10^{-2}	
LR decay	drop = 0.1; Step decay = 15 epochs	
Optimization Strategy	Adam ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-7}$)	
Epochs	75	
Batch size	8	
Dropout rate	0.1	
Batch normalization	True	

4.4 Test cross-validation

To demonstrate the validity and robustness of the implemented approaches and considering that the dataset is quite limited, a leave-one-experiment-out 10-k cross-validation (see the figure in appendix A.2) was applied. This validation consists of leaving one dataset image for testing and the remaining 20 images for training. As such, 21 different training processes were performed. This avoids the bias introduced when testing a model with images with a high degree of similarity [127].

Cross-validation was performed ten times for each training set, resulting in a total number of 210 training processes. This is a very useful validation technique when we do not have a large data size, as happens in this study [128].

Chapter 5

Results

In this chapter, the results obtained by the implementation proposed in this dissertation are presented and compared from the benchmark using the U-Net architecture.

In section 5.1, a quantitative analysis of the results is made by evaluating the metrics obtained in each model. As experiments were carried out using two U-Net architectures and tested with two loss functions, eight results were obtained for the binary and multi-class kidney segmentation (contour class). Afterward, these results are commented, according to the number of classes, since the multi-class segmentation obtained, result in lower performance for all tested cases, than those produced with the binary segmentation. Finally, the results are summarized. Section 5.2 compares a comparative analysis with other representative state-of-the-art implementations of the addressed problem.

5.1 Segmentation Results

The segmentation performance for each model is given by evaluating the metrics obtained. Tables 5.1 and 5.2 present the results for the binary and multi-class segmentation, respectively, of Dice coefficient (DSC), Jaccard index(IoU) and loss function.

Quantitatively, the attention U-Net with the cross-entropy loss function is the model that presents the best results. The results obtained by U-Net using cross-entropy show very similar results in all metrics. It presents higher results in some metrics, but in order of magnitude, out of interest for a 256 x 256 resolution image. What distinguishes these two models is the evaluation of the Recall, indicating that the percentage of false negatives is lower using the Attention U-Net. The standard deviation value in all metrics is smaller in the Attention U-Net model, demonstrating better reliability.

It should be noted that the results obtained by adding a third contour class show worse results in all models.

To assess the quality of the segmentation, the most relevant metrics are the Dice coefficient and the Jaccard Index (see subsection 2.2.6) [45] because they measure the overlap between the segmented image and the ground truth, considering the spatial arrangement, size, and shape of the segmented object. When analyzing the results obtained by the metrics mentioned above in all experiments, the architecture and the loss function present, in most cases, the best result is the Attention U-Net with the cross-entropy loss function. It should be noted that U-Net and Attention U-Net were trained with the same configuration. Evaluating the loss function value in Attention U-Net architecture and considering the attention mechanism incorporated in Attention U-Net, it is possible to conclude that by optimizing parameters such as learning rate, learning rate decay, and the number of epochs, a better performance results as expected. Nevertheless, it is important to emphasize that the performance of U-Net was basically the same, presenting slightly lower performance.

Table 5.1: Evaluation results of Dice coefficient, Jaccard Index and Loss function of binary segmentation models.

Segmentation Model	<i>DCS</i>	<i>IoU</i>	<i>Loss</i>
U-NET + CE	0.96577±0.00871	0.93393±0.01608	0.00843±0.00461
U-NET + FL	0.95985±0.01119	0.92301±0.02035	0.00096±0.00055
ATT U-NET + CE	0.96557±0.00740	0.93354±0.01377	0.12585±0.00332
ATT U-NET + FL	0.94399±0.09431	0.90222±0.09393	0.00731±0.00175

Table 5.2: Evaluation results of Dice coefficient, Jaccard Index and Loss function of three classes segmentation models.

Segmentation Model	<i>DCS</i>	<i>IoU</i>	<i>Loss</i>
U-NET + CE	0.93667±0.01519	0.88755±0.02369	0.01642±0.00899
U-NET + FL	0.93627±0.01568	0.88696±0.02424	0.00042±0.00031
ATT U-NET + CE	0.93905±0.01277	0.89114±0.02018	0.07656±0.00797
ATT U-NET + FL	0.93747±0.01528	0.88878±0.02369	0.00090±0.00050

Binary classification results When looking at the experimental results for binary classification results from the annex tables A.1, A.5 and A.9, we can conclude:

- The use of the Cross-Entropy (CE) loss function is the loss function that presents the best segmentation performance. The learning process of the feature representation of the images and weights update is performed by minimizing the cost function. The models trained from minimizing the focal loss (FL) cost function converge quickly (during the first ten epochs), partially neglecting the rest of the training process. Figure A.2 shows the graphs of the evolution of the loss function throughout the training process for the case in which the best segmentation performance is achieved in each model. It should be noted that despite the focal loss reaching lower minimum values than the CE, these values are not synonymous with a better segmentation performance;
- The architecture that presents the highest absolute percentage values is U-Net. Observing the results obtained between U-Net and Attention U-Net, both using CE as a loss function, we conclude that U-Net obtained Accuracy 0.004% greater, Precision 0.574% greater, Dice coefficient 0.02% higher, Jaccard Index 0.04% higher while the recall was greater 0.536% in Attention U-Net. Objectively, it is concluded that U-Net produced fewer false positives, given its greater precision, and Attention U-Net fewer false negatives, given its greater recall, since the remaining metrics present objectively equal results;
- The Attention U-Net model with CE has the highest absolute percentage values of metrics during the training process. The architecture of Attention U-Net has more complexity than U-Net because it has a built-in attention mechanism. The increased complexity of the network optimized the learning process and improved the classification quality;
- The standard deviation obtained for the validation methodology used for the results of all metrics is smaller in the case of Attention U-Net. This architecture uses the attention mechanism that aims to assign greater weight to the most relevant pixels, focusing learning on the ROI, causing the stabilization of the classification throughout the training process.
- From a visual analysis of the results obtained over the 210 training processes, it is concluded that:
 - In the U-NET + CE model, there is 1 case in which pixels of the kidney class are classified outside their ROI;
 - In the U-NET + FL model, there is 1 case in which pixels of the kidney class are classified outside their ROI;
 - In the ATT U-NET + CE model, there are 0 cases in which pixels of the kidney class are classified outside their ROI;
 - In the ATT U-NET + FL model, there are 9 cases in which the kidneys appear visibly deformed, 10 cases in which pixels of the kidney class are classified outside their ROI, and 2 in which there was no segmentation.

- Given the visual analysis of the classification results obtained and the evaluation of the metrics results, it was concluded that the model with the worst results was the Attention U-Net using FL as loss function;
- Within the four models used to segment the kidney, evaluating the results of all metrics in the classification process, it is concluded that the model that presents the best results is the Attention U-Net model using CE as a loss function.

Multi-Class classification results When looking at the experimental results for multi-class classification results from tables A.2, A.3, A.4, A.6, A.7, A.8 and A.10, we can conclude:

- The use of the CE loss function is the loss function that presents the best segmentation performance. Learning the characteristics of the images and updating the weights is done by minimizing the cost function. The models trained from minimizing the focal loss cost function converge very quickly (during the first 10 epochs), partially neglecting the rest of the training process. Figure A.3 shows the graphs of the evolution of the loss function throughout the training process for the case in which the best segmentation performance is achieved in each model. It should be noted that despite the focal loss reaching lower minimum values than the CE, for the case in question, these values are not synonymous with a better segmentation performance;
- The architecture that presents the best ranking results given by all evaluation metrics is Attention U-Net;
- The standard deviation obtained for the validation methodology used for the results of all metrics is smaller in the case of Attention U-Net. This architecture uses the attention mechanism that aims to assign greater weight to the most relevant pixels, focusing the learning process on the ROI, causing the stabilization of the classification throughout the training process.
- The model that presents the best results throughout the training process is the U-Net with CE as a loss function. Since the results of the classification process are better with Attention U-Net, it would be expected that this architecture also produces better results during the training process. However, learning multi-class segmentation is a process of high computational complexity. Given that both networks were designed with the same configurations, it is assumed that Attention U-Net did not reach an optimal state. By increasing the number of epochs, learning rate, and learning rate decay, the Attention U-Net architecture is expected to show better results;
- From a visual analysis of the results obtained over the 210 training processes, it is concluded that:
 - In the U-NET + CE model, there are 3 cases in which pixels of the kidney class are classified outside their ROI.
 - In the U-NET + FL model, there are 3 cases in which pixels of the kidney class are classified outside their ROI.

- In the ATT U-NET + CE model, there are 0 cases in which pixels of the kidney class are classified outside their ROI.
- In the ATT U-NET + FL model, there are 3 cases in which pixels of the kidney class are classified outside their ROI.
- Given the visual analysis of the classification results obtained and the evaluation of the metrics results, it was concluded that the model that presented the worst results was the U-Net using FL as loss function;
- Within the four models used to make a multi-class segmentation, evaluating the results of all metrics in the classification process, it is concluded that the model that presents the best results is the Attention U-Net using CE as loss function.

Conclusion

- Attention U-Net produced the best-achieved result for binary segmentation with a cross-entropy loss function. Achieved 0.966 ± 0.009 at dice coefficient and 0.934 ± 0.016 at Jaccard index;
- The highest achieved result for three class segmentation was produced by Attention U-Net with cross-entropy loss function;
- Binary segmentation models outperformed the proposed coarse-to-fine approach;
- Focal loss function best fits in multi-class segmentation approaches;
- Attention U-Net has shown more importance in multi-class segmentation approaches.

5.2 Comparison with the state-of-the-art

Table 5.3 presents the results of the two best models proposed by this dissertation and other approaches that are part of the state-of-the-art in the segmentation of the kidneys in abdominal MRI images. Only studies proposed from 2015 onwards are presented. In all studies presented, the result obtained relative to the coefficient dice and description of the used dataset are reported. The results obtained for the Jaccard index and precision in the studies that include these results as evaluation metrics are reported. Each approach is briefly described in subsection 3.3.2.

Kline's method makes a multi-observation by the voting scheme on each pixel of the result obtained by 11 different U-Net configurations. This approach is only possible with large datasets since the primary dataset has been divided into ten secondary datasets. This approach presents the best results; given its morphology, it is very robust. In practice, the results obtained in this study are slightly better but with a much stronger standard deviation.

In the approach presented by Van Gasten, the segmentation is achieved using U-Net and presents very similar results. The remaining approaches show considerably lower results. It should be noted that

the method proposed by Kline, van Gasten, or Bevilacqua is performed on the kidneys of patients with ADPKD or PKD.

In the approaches that present results for several organs, it is worth mentioning the model proposed by Chen in which he reached a kidney segmentation of 0.951 using Dense U-Net and compared it with the results obtained using U-Net.

Table 5.3: Comparison of the DSC and IoU with the state-of-the-art methods

Method	<i>DSC</i>	<i>IoU</i>	Dataset	Train/Test	Validation
Multi-observer U-NET [111]	0.97±0.01	0.94±0.03	2D:256 x 256	2000/400	holdout
U-NET (proposed)	0.966±0.009	0.934±0.016	2D:256 x 256	20/1	LOOCV
ATT U-NET (proposed)	0.966±0.007	0.934±0.014	2D:256 x 256	20/1	LOOCV
U-NET [112]	0.96±0.02	0.92±0.03	2.5D:256 x 256 x 3	352+88/100	5-fold CV
Dense U-NET [116]	0.954±0.008	0.913±0.015	3D:256 x 160 x 20	66+16/20	holdout
Modified U-NET	0.86±0.08	0.76±0.11	2D:256 x 256 x 3	316+105/105	5-fold CV
CNN	0.52	-	2D:256 x 256 x 3	316+105/105	-
FCN [115]	0.780	-	2D:256 x 256 x 3	111/27	-

Chapter 6

Conclusions

6.1 Achievements

In conclusion, this dissertation addressed the segmentation of kidneys in abdominal MRI images using the U-Net and Attention U-Net architectures. The objectives were successfully achieved, including obtaining an approach for kidney segmentation, comparing the results of U-Net and Attention U-Net, evaluating different loss functions, and exploring binary and multi-class segmentation as a coarse-to-fine approach.

The results demonstrated that the Attention U-Net model outperformed U-Net, contributing to state-of-the-art kidney segmentation. The Cross-Entropy loss function showed the best results among the studied functions, while the Focal Loss function requires further parameterization exploration. Binary segmentation yielded better results for small datasets with limited epochs, while multi-class segmentation with a contour class did not significantly improve kidney segmentation.

Importantly, this study showed that a large dataset is not mandatory, and training neural networks with excessive epochs is not essential. These findings challenge some prevailing assumptions in the field.

The results support the consideration of Attention U-Net as a valuable architecture for developing methodologies and tools that utilize kidney segmentation in abdominal MRI images as a biomarker for kidney assessment. Further research can explore the optimization of the architecture's parameterization and investigate the potential of the Focal Loss function with refined parameters.

6.2 Future Work

Regarding the proposed models, some changes should be investigated in the future to improve their performance. This section outlines the key areas for the investigation to further advance kidney segmentation in MRI images using deep learning.

Optimization of the Proposed Algorithm The algorithm that combines Attention U-Net with the Cross-Entropy loss function has demonstrated the best results in our study. However, further optimization can be performed to minimize the loss function. Specifically, studies on learning rate decay and the number of epochs should be prioritized to improve convergence and enhance the overall accuracy of the models.

Parameterization of the Focal Loss Function Our research utilized the focal loss function with $\alpha = 0.25$ and $\gamma = 2$. However, further investigations should be conducted to explore different parameterizations of this loss function for medical image segmentation. Specifically, varying the α value while maintaining the γ value constant can provide insights into the importance attributed to different classes.

Evaluation on Larger Datasets While our study achieved promising results with a small dataset, evaluating the proposed models on larger datasets is important. Emphasizing datasets with a substantial number of samples will enable a quantitative analysis of the impact of dataset size on model performance. This investigation can shed light on the scalability and generalizability of the proposed methods.

Calculation of Kidney Volume To enhance the clinical relevance of our research, the calculation of kidney volume based on the obtained segmentation should be implemented. Performing instance segmentation on each kidney can yield accurate volume measurements, thereby contributing to potential clinical applications.

Exploration of Pre-trained Backbone Models To leverage pre-existing knowledge learned from large-scale datasets, we suggest employing pre-trained weights of backbone models, such as ResNET, EfficientNet, or ImageNet, for feature extraction in our models. This approach has the potential to improve model performance and efficiency.

Residual Attention U-Net We propose implementing the Residual Attention U-Net to expand the comparison of U-Net-based architectures. This architecture has exhibited promising results in various computer vision tasks and may contribute to advancing the state-of-the-art in kidney segmentation.

Investigation of the Dice Loss Function Exploring the Dice loss function as an alternative to the Cross-Entropy and focal loss functions would provide further insights into its effectiveness in addressing the challenges specific to kidney segmentation.

By pursuing these future research directions, we anticipate significant advancements in the performance, robustness, and clinical applicability of the proposed models for kidney segmentation in MRI images using deep learning.

Bibliography

- [1] N. M. Selby, P. J. Blankestijn, P. Boor, C. Combe, K.-U. Eckardt, E. Eikefjord, N. Garcia-Fernandez, X. Golay, I. Gordon, N. Grenier, et al. Magnetic resonance imaging biomarkers for chronic kidney disease: a position paper from the european cooperation in science and technology action parenchima. *Nephrology Dialysis Transplantation*, 33(suppl_2):ii4–ii14, 2018.
- [2] K. Sharma, C. Rupprecht, A. Caroli, M. C. Aparicio, A. Remuzzi, M. Baust, and N. Navab. Automatic segmentation of kidneys using deep learning for total kidney volume quantification in autosomal dominant polycystic kidney disease. *Sci. Rep.*, 7(1):2049, May 2017.
- [3] P. Niyishaka and C. Bhagvati. Image splicing detection technique based on illumination-reflectance model and lbp. *Multimedia Tools and Applications*, 80:2161–2175, 2021.
- [4] Z. Zhao, H. Chen, and L. Wang. A coarse-to-fine framework for the 2021 kidney and kidney tumor segmentation challenge. In *Kidney and Kidney Tumor Segmentation: MICCAI 2021 Challenge, KiTS 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings*, pages 53–58. Springer, 2022.
- [5] D. Li and Y. Du. *Artificial Intelligence with Uncertainty*. CRC Press, 2nd edition, 2016. ISBN:978-1315366951.
- [6] S. Dick. Artificial Intelligence. *Harvard Data Science Review*, 1(1), jul 1 2019. <https://hdsr.mitpress.mit.edu/pub/0aytgrau>.
- [7] H. Al-Sahaf, Y. Bi, Q. Chen, A. Lensen, Y. Mei, Y. Sun, B. Tran, B. Xue, and M. Zhang. A survey on evolutionary machine learning. *Journal- Royal Society of New Zealand*, 05 2019. doi: 10.1080/03036758.2019.1609052.
- [8] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL <https://doi.org/10.1038/nature14539>.
- [9] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL <https://doi.org/10.1038/nature14539>.
- [10] A. M. IESC. Artificial intelligence, machine learning, and deep learning: Same context, different concepts, Apr 2018. URL <https://master-iesc-angers.com/artificial-intelligence-machine-learning-and-deep-learning-same-context-different-concepts/>.

- [11] D. Bzdok, M. Krzywinski, and N. Altman. Machine learning: supervised methods. *Nature Methods*, 15(1):5–6, Jan 2018. ISSN 1548-7105. doi: 10.1038/nmeth.4551. URL <https://doi.org/10.1038/nmeth.4551>.
- [12] IBM-Cloud-Education. Unsupervised learning, 2020. URL <https://www.ibm.com/cloud/learn/unsupervised-learning>.
- [13] Y. Chen, J. Wang, and R. Krovetz. An unsupervised learning approach to content-based image retrieval. In *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, volume 1, pages 197–200 vol.1, 2003. doi: 10.1109/ISSPA.2003.1224674.
- [14] I. SALIAN. Nvidia blog: Supervised vs. unsupervised learning, 2018. URL <https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning>.
- [15] H. Han, W. Ma, M. Zhou, Q. Guo, and A. Abusorrah. A novel semi-supervised learning approach to pedestrian reidentification. *IEEE Internet of Things Journal*, 8(4):3042–3052, 2021. doi: 10.1109/JIOT.2020.3024287.
- [16] M. L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.
- [17] J. Feng, M. Huang, L. Zhao, Y. Yang, and X. Zhu. Reinforcement learning for relation classification from noisy data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.12063. URL <https://ojs.aaai.org/index.php/AAAI/article/view/12063>.
- [18] R. van Loon. Machine learning explained: Understanding supervised, unsupervised, and reinforcement learning. <https://datafloq.com/read/machine-learning-explained-understanding-learning/>, Jan. 2018. Accessed: 2023-5-10.
- [19] D. J. Hemanth, O. Deperlioglu, and U. Kose. An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. *Neural Computing and Applications*, 32(3):707–721, Feb 2020. ISSN 1433-3058. doi: 10.1007/s00521-018-03974-0. URL <https://doi.org/10.1007/s00521-018-03974-0>.
- [20] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):7–19, 2015. doi: 10.1109/TASLP.2014.2364452.
- [21] R. Szeliski. *Computer Vision*, chapter 5, pages 235–271. 1868-0941. Springer London, Springer-Verlag London Ltd., part of Springer Nature 2011, 1 edition, 2011. ISBN 978-1-84882-935-0. doi: 10.1007/978-1-84882-935-0.

- [22] S.-C. Wang. *Artificial Neural Network*, pages 81–100. Springer US, Boston, MA, 2003. ISBN 978-1-4615-0377-4. doi: 10.1007/978-1-4615-0377-4_5. URL https://doi.org/10.1007/978-1-4615-0377-4_5.
- [23] R. Bala and D. A. Kumar. Classification using ann : A review. 2017.
- [24] Cusabio. Get an Overview of Neuron Cells- CUSABIO. <https://www.cusabio.com/Cell-Marker/Neuron-Cell.html>. [Accessed 07-Feb-2023].
- [25] A. Sumari, A. Wuryandari, M. Darusman, and N. Utama. The performance of supervised and unsupervised neural networks in performing aircraft identification tasks. 04 2009.
- [26] I. Basheer and M. Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1):3–31, 2000. ISSN 0167-7012. doi: [https://doi.org/10.1016/S0167-7012\(00\)00201-3](https://doi.org/10.1016/S0167-7012(00)00201-3). URL <https://www.sciencedirect.com/science/article/pii/S0167701200002013>. Neural Computing in Micrbiology.
- [27] P. L. Fernández-Cabán, F. J. Masters, and B. M. Phillips. Predicting roof pressures on a low-rise structure from freestream turbulence using artificial neural networks. *Frontiers in Built Environment*, 4, 2018. ISSN 2297-3362. doi: 10.3389/fbuil.2018.00068. URL <https://www.frontiersin.org/articles/10.3389/fbuil.2018.00068>.
- [28] S. Khan, H. Rahmani, S. Shah, and M. Bennamoun. *A Guide to Convolutional Neural Networks for Computer Vision*. Number 1 in Synthesis Lectures on Computer Vision. Morgan Claypool Publishers, 2018. doi: 10.2200/S00822ED1V01Y201712COV015.
- [29] K. Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 1969. doi: 10.1109/TSSC.1969.300225.
- [30] P. Rani, S. Kotwal, J. Manhas, V. Sharma, and S. Sharma. Machine learning and deep learning based computational approaches in automatic microorganisms image recognition: Methodologies, challenges, and developments. *Archives of Computational Methods in Engineering*, 29(3):1801–1837, May 2022. ISSN 1886-1784. doi: 10.1007/s11831-021-09639-x. URL <https://doi.org/10.1007/s11831-021-09639-x>.
- [31] L. Pauly, H. Peel, S. Luo, D. Hogg, and R. Fuentes. Deeper networks for pavement crack detection. 07 2017. doi: 10.22260/ISARC2017/0066.
- [32] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning, 12 2020.
- [33] J. Chang, P. Arbeláez, N. Switz, C. Reber, A. Tapley, J. L. Davis, A. Cattamanchi, D. Fletcher, and J. Malik. Automated tuberculosis diagnosis using fluorescence images from a mobile microscope. In N. Ayache, H. Delingette, P. Golland, and K. Mori, editors, *Medical Image Computing*

- and *Computer-Assisted Intervention – MICCAI 2012*, pages 345–352, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33454-2.
- [34] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
 - [35] I. J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. <http://www.deeplearningbook.org>.
 - [36] B. Chen, W. Deng, and J. Du. Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4021–4030, 2017. doi: 10.1109/CVPR.2017.428.
 - [37] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.
 - [38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
 - [39] A. Tiwari. Chapter 2 - supervised learning: From theory to applications. In R. Pandey, S. K. Khatri, N. kumar Singh, and P. Verma, editors, *Artificial Intelligence and Machine Learning for EDGE Computing*, pages 23–32. Academic Press, 2022. ISBN 978-0-12-824054-0. doi: <https://doi.org/10.1016/B978-0-12-824054-0.00026-5>. URL <https://www.sciencedirect.com/science/article/pii/B9780128240540000265>.
 - [40] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
 - [41] F. van Beers, A. Lindström, E. Okafor, and M. Wiering. Deep neural networks with intersection over union loss for binary image segmentation. 02 2019. doi: 10.5220/0007347504380445.
 - [42] R. Padilla, S. L. Netto, and E. A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020. doi: 10.1109/IWSSIP48289.2020.9145130.
 - [43] R. Zhao, B. Qian, X. Zhang, Y. Li, R. Wei, Y. Liu, and Y. Pan. Rethinking dice loss for medical image segmentation. *2020 IEEE International Conference on Data Mining (ICDM)*, pages 851–860, 2020.
 - [44] E. Tiu. Metrics to Evaluate your Semantic Segmentation Model. <https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2>, 2019. [Accessed 07-Feb-2022].

- [45] T. Eelbode, J. Bertels, M. Berman, D. Vandermeulen, F. Maes, R. Bisschops, and M. Blaschko. Optimization for medical image segmentation: Theory and practice when evaluating with dice score or jaccard index. *IEEE Transactions on Medical Imaging*, PP:1–1, 06 2020. doi: 10.1109/TMI.2020.3002417.
- [46] Y. Kurmi and V. Chaurasia. Multifeature-based medical image segmentation. *IET Image Processing*, 12(8):1491–1498, 2018. doi: <https://doi.org/10.1049/iet-ipr.2017.1020>. URL <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-ipr.2017.1020>.
- [47] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.
- [48] J. Blumberg and G. Kreiman. How cortical neurons help us see: visual recognition in the human brain. *J Clin Invest*, 120(9):3054–3063, Sept. 2010.
- [49] M. Mishra. Convolutional neural networks, Sep 2020. URL <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>.
- [50] F. Altaf, S. M. S. Islam, N. Akhtar, and N. K. Janjua. Going deep in medical image analysis: Concepts, methods, challenges, and future directions. *IEEE Access*, 7:99540–99572, 2019. doi: 10.1109/ACCESS.2019.2929365.
- [51] K. Patel. Convolution Neural Networks — A Beginner’s Guide. <https://towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa6> 2020. [Accessed 24-Mar-2022].
- [52] J. Yan, L. Mu, L. Wang, R. Ranjan, and A. Zomaya. Temporal convolutional networks for the advance prediction of enso. *Scientific Reports*, 10:8055, 05 2020. doi: 10.1038/s41598-020-65070-5.
- [53] V. Suárez-Paniagua and I. Segura-Bedmar. Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction. *BMC Bioinformatics*, 19, 06 2018. doi: 10.1186/s12859-018-2195-1.
- [54] D. A. Van Valen, T. Kudo, K. M. Lane, D. N. Macklin, N. T. Quach, M. M. DeFelice, I. Maayan, Y. Tanouchi, E. A. Ashley, and M. W. Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLOS Computational Biology*, 12(11):1–24, 11 2016. doi: 10.1371/journal.pcbi.1005177. URL <https://doi.org/10.1371/journal.pcbi.1005177>.
- [55] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25.

- Curran Associates, Inc., 2012. URL <https://proceedings.neurips.cc/paper/2012/file/459a4ddcb586f24efd9395aa7662bc7c-Paper.pdf>.
- [56] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857, 2017. URL <http://arxiv.org/abs/1704.06857>.
- [57] M. A. Al Mamun and I. Kadir. *an-Eye: SAFE NAVIGATION IN FOOTPATH FOR VISUALLY IMPAIRED USING COMPUTER VISION TECHNIQUES*. PhD thesis, 06 2020.
- [58] Z. Niu, G. Zhong, and H. Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2021.03.091>. URL <https://www.sciencedirect.com/science/article/pii/S092523122100477X>.
- [59] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert. Attention u-net: Learning where to look for the pancreas, 2018. URL <https://arxiv.org/abs/1804.03999>.
- [60] M. H. Hesamian, W. Jia, X. He, and P. Kennedy. Deep learning techniques for medical image segmentation: Achievements and challenges. *Journal of Digital Imaging*, 32(4):582–596, Aug 2019. ISSN 1618-727X. doi: 10.1007/s10278-019-00227-x. URL <https://doi.org/10.1007/s10278-019-00227-x>.
- [61] N. Sharma and L. M. Aggarwal. Automated medical image segmentation techniques. *J Med Phys*, 35(1):3–14, Jan. 2010.
- [62] J. Egger, A. Pepe, C. Gsaxner, Y. Jin, J. Li, and R. Kern. Deep learning-a first meta-survey of selected reviews across scientific disciplines, their commonalities, challenges and research impact. *PeerJ Comput. Sci.*, 7(e773):e773, Nov. 2021.
- [63] L. Jing, Y. Chen, and Y. Tian. Coarse-to-fine semantic segmentation from image-level labels. *CoRR*, abs/1812.10885, 2018. URL <http://arxiv.org/abs/1812.10885>.
- [64] S. Wang, C. Li, R. Wang, Z. Liu, M. Wang, H. Tan, Y. Wu, X. Liu, H. Sun, R. Yang, X. Liu, J. Chen, H. Zhou, I. Ben Ayed, and H. Zheng. Annotation-efficient deep learning for automatic medical image segmentation. *Nature Communications*, 12(1):5915, Oct. 2021.
- [65] F. Luo, B.-B. Gao, J. Yan, and X. Li. A coarse-to-fine instance segmentation network with learning boundary representation, 2021. URL <https://arxiv.org/abs/2106.10213>.
- [66] J. Lee, T. Ilyas, H. Jin, J. Lee, O. Won, H. Kim, and S. J. Lee. A pixel-level coarse-to-fine image segmentation labelling algorithm. *Scientific Reports*, 12(1):8672, May 2022.
- [67] J. Sharkey, L. Scarfe, I. Santeramo, M. Garcia-Finana, B. K. Park, H. Poptani, B. Wilm, A. Taylor, and P. Murray. Imaging technologies for monitoring the safety, efficacy and mechanisms of action of cell-based regenerative medicine therapies in models of kidney disease. *European Journal of Pharmacology*, 790:74–82, 2016.

- [68] H. R. Torres, S. Queiros, P. Morais, B. Oliveira, J. C. Fonseca, and J. L. Vilaca. Kidney segmentation in ultrasound, magnetic resonance and computed tomography images: A systematic review. *Computer methods and programs in biomedicine*, 157:49–67, 2018.
- [69] S. University. Normal kidney, Sagittal view (Ultrasound). <https://web.stanford.edu/dept/radiology/radiologysite/site376.html>. [Accessed 13-Mar-2022].
- [70] G. Koulouris. CT Scan of the Kidney — Melbourne Radiology. <https://www.melbournerradiology.com.au/diagnostic-imaging/ct-scan-kidney/>, 2022. [Accessed 28-Oct-2022].
- [71] R. S. R. George, J. Dela Cruz. MRI kidneys (renal) planning — MRI kidneys protocol— indications for MRI kidneys scan. <https://mrmaster.com/PLAN%20MRI%20KIDNEYS.html>, 2023. [Accessed 13-Feb-2023].
- [72] C. L. Chapman, H. W. Hess, R. A. Lucas, J. Glaser, R. Saran, J. Bragg-Gresham, D. H. Wegman, E. Hansson, C. T. Minson, and Z. J. Schlader. Occupational heat exposure and the risk of chronic kidney disease of nontraditional origin in the united states. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 321(2):R141–R151, 2021.
- [73] W. Brisbane, M. R. Bailey, and M. D. Sorensen. An overview of kidney stone imaging techniques. *Nat Rev Urol*, 13(11):654–662, Aug. 2016.
- [74] N. R. Council et al. Health risks from exposure to low levels of ionizing radiation: Beir vii, phase i, letter report. 1998.
- [75] D. Cosgrove. Ultrasound contrast agents: an overview. *European journal of radiology*, 60(3): 324–330, 2006.
- [76] J. L. Zhang, H. Rusinek, H. Chandarana, and V. S. Lee. Functional mri of the kidneys. *Journal of magnetic resonance imaging*, 37(2):282–293, 2013.
- [77] D. C. Preston. Magnetic Resonance Imaging (MRI) of the brain and spine: Basics. <https://case.edu/med/neurology/NR/MRI%20Basics.htm>, 2006. [URL revised 07/04/16], [Accessed 09-Jun-2022].
- [78] W. Huda. Radiation doses and risks in chest computed tomography examinations. *Proceedings of the American Thoracic Society*, 4(4):316–320, 2007.
- [79] C. Türk, T. Knoll, A. Petrik, K. Sarica, M. Straub, and C. Seitz. European association of urology guidelines on urolithiasis. *Eur Urol*, 69:468, 2015.
- [80] P. F. Fulgham, D. G. Assimos, M. S. Pearle, and G. M. Preminger. Clinical effectiveness protocols for imaging in the management of ureteral calculous disease: Aua technology assessment. *The Journal of urology*, 189(4):1203–1213, 2013.

- [81] F. G. Zöllner, M. Kociński, L. Hansen, A.-K. Golla, A. Trbalić, A. Lundervold, A. Materka, and P. Rogelj. Kidney segmentation in renal magnetic resonance imaging - current status and prospects. *IEEE Access*, 9:71577–71605, 2021. doi: 10.1109/ACCESS.2021.3078430.
- [82] D. Turco, S. Severi, R. Mignani, R. Magistroni, and C. Corsi. Geometry-independent assessment of renal volume in polycystic kidney disease from magnetic resonance imaging. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, Aug. 2015.
- [83] F. G. Zöllner, M. Kociński, L. Hansen, A.-K. Golla, A. Š. Trbalić, A. Lundervold, A. Materka, and P. Rogelj. Kidney segmentation in renal magnetic resonance imaging-current status and prospects. *IEEE access*, 9:71577–71605, 2021.
- [84] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi: 10.1109/TSMC.1979.4310076.
- [85] Q. Salih and A. Ramli. Region based segmentation technique and algorithms for 3d image. In *Proceedings of the Sixth International Symposium on Signal Processing and its Applications (Cat.No.01EX467)*, volume 2, pages 747–748 vol.2, 2001. doi: 10.1109/ISSPA.2001.950259.
- [86] E. A. Hanson and A. Lundervold. Local/non-local regularized image segmentation using graph-cuts: application to dynamic and multispectral MRI. *Int. J. Comput. Assist. Radiol. Surg.*, 8(6): 1073–1084, Nov. 2013.
- [87] S. P. *Morphological image analysis: principles and applications*. Springer Berlin, Heidelberg, 2003.
- [88] F. G. Zöllner, E. Svarstad, A. Z. Munthe-Kaas, L. R. Schad, A. Lundervold, and J. Rørvik. Assessment of kidney volumes from mri: acquisition and segmentation techniques. *American Journal of Roentgenology*, 199(5):1060–1069, 2012.
- [89] A. Belgherbi, I. Hadjidi, and A. Bessaid. Morphological segmentation of the kidneys from abdominal ct images. *J. Mech. Med. Biol.*, 14(05):1450073, Oct. 2014.
- [90] J. Huang, X. Yang, Y. Chen, and L. Tang. Ultrasound kidney segmentation with a global prior shape. *Journal of Visual Communication and Image Representation*, 24(7):937–943, 2013. ISSN 1047-3203. doi: <https://doi.org/10.1016/j.jvcir.2013.05.013>. URL <https://www.sciencedirect.com/science/article/pii/S1047320313001132>.
- [91] L. Bokacheva, H. Rusinek, J. L. Zhang, and V. S. Lee. Assessment of renal function with dynamic contrast-enhanced MR imaging. *Magn. Reson. Imaging Clin. N. Am.*, 16(4):597–611, viii, Nov. 2008.
- [92] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022. doi: 10.1109/TPAMI.2021.3059968.

- [93] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, Aug 2018. ISSN 1869-4101. doi: 10.1007/s13244-018-0639-9. URL <https://doi.org/10.1007/s13244-018-0639-9>.
- [94] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *Ieee Access*, 9:82031–82057, 2021.
- [95] A. Abdelrahman and S. Viriri. Kidney tumor semantic segmentation using deep learning: A survey of State-of-the-Art. *J Imaging*, 8(3), Feb. 2022.
- [96] A. H. Andriy Myronenko. 3d kidneys and kidney tumor semantic segmentation using boundary-aware networks, 2019.
- [97] D. B. Efremova, D. A. Konovalov, T. Siriapisith, W. Kusakunniran, and P. Haddawy. Automatic segmentation of kidney and liver tumors in ct images, 2019.
- [98] J. Guo, W. Zeng, S. Yu, and J. Xiao. Rau-net: U-net model based on residual and attention for kidney and kidney tumor segmentation. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 353–356, 2021. doi: 10.1109/ICCECE51280.2021.9342530.
- [99] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang. Unet++: A nested u-net architecture for medical image segmentation. In D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00889-5.
- [100] X. Xie, L. Li, S. Lian, S. Chen, and Z. Luo. Seru: A cascaded se-resnext u-net for kidney and tumor segmentation. *Concurrency and Computation: Practice and Experience*, 32(14):e5738, 2020.
- [101] J. Heo. Automatic segmentation in abdominal CT imaging for the kiTS21 challenge, 2021. URL <https://openreview.net/forum?id=n6DR2TdGLa>.
- [102] L. B. Cruz, J. D. L. Araújo, J. L. Ferreira, J. O. B. Diniz, A. C. Silva, J. D. S. de Almeida, A. C. de Paiva, and M. Gattass. Kidney segmentation from computed tomography images using deep neural network. *Computers in Biology and Medicine*, 123:103906, 2020. ISSN 0010-4825. doi: <https://doi.org/10.1016/j.compbimed.2020.103906>. URL <https://www.sciencedirect.com/science/article/pii/S0010482520302523>.
- [103] Y. Zhang, Y. Wang, F. Hou, J. Yang, G. Xiong, J. Tian, and C. Zhong. Cascaded volumetric convolutional network for kidney tumor segmentation from CT volumes. Oct. 2019.
- [104] A. Hatamizadeh, D. Terzopoulos, and A. Myronenko. Edge-gated cnns for volumetric semantic segmentation of medical images, 2020.

- [105] W. Zhao, D. Jiang, J. Peña Queralta, and T. Westerlund. Mss u-net: 3d segmentation of kidneys and tumors from ct images with a multi-scale supervised u-net. *Informatics in Medicine Unlocked*, 19:100357, 2020. ISSN 2352-9148. doi: <https://doi.org/10.1016/j.imu.2020.100357>. URL <https://www.sciencedirect.com/science/article/pii/S2352914820301969>.
- [106] G. Santini, N. Moreau, and M. Rubeaux. Kidney tumor segmentation using an ensembling multi-stage deep learning approach. a contribution to the kits19 challenge, 2019.
- [107] Z. Chen and H. Liu. 2.5d cascaded semantic segmentation for kidney tumor cyst, 2021. URL https://openreview.net/forum?id=d5WM4_asJC1.
- [108] T. He, Z. Zhen, P. Chenhao, and H. Liqin. A two-stage cascaded deep neural network with multi-decoding paths for kidney tumor segmentation, 2021. URL <https://openreview.net/forum?id=c7kCK-E-B1>.
- [109] H. Wei, Q. Wang, W. Zhao, M. Zhang, K. Yuan, and Z. Li. Two-phase framework for automatic kidney and kidney tumor segmentation. In *Submissions to the 2019 Kidney Tumor Segmentation Challenge: KiTS19*. University of Minnesota Libraries Publishing, 2019.
- [110] J. Cheng, J. Liu, L. Liu, Y. Pan, and J. Wang. A double cascaded framework based on 3D SEAU-Net for kidney and kidney tumor segmentation. In *Submissions to the 2019 Kidney Tumor Segmentation Challenge: KiTS19*. University of Minnesota Libraries Publishing, 2019.
- [111] T. L. Kline, P. Korfiatis, M. E. Edwards, J. D. Blais, F. S. Czerwiec, P. C. Harris, B. F. King, V. E. Torres, and B. J. Erickson. Performance of an artificial multi-observer deep neural network for fully automated segmentation of polycystic kidneys. *J Digit Imaging*, 30(4):442–448, Aug. 2017.
- [112] M. D. A. van Gastel, M. E. Edwards, V. E. Torres, B. J. Erickson, R. T. Gansevoort, and T. L. Kline. Automatic measurement of kidney and liver volumes from MR images of patients affected by autosomal dominant polycystic kidney disease. *J Am Soc Nephrol*, 30(8):1514–1522, July 2019.
- [113] V. Bevilacqua, A. Brunetti, G. D. Cascarano, A. Guerriero, F. Pesce, M. Moschetta, and L. Gesualdo. A comparison between two semantic deep learning frameworks for the autosomal dominant polycystic kidney disease segmentation based on magnetic resonance images. *BMC Medical Informatics and Decision Making*, 19(9):244, Dec. 2019.
- [114] A. Brunetti, G. D. Cascarano, I. De Feudis, M. Moschetta, L. Gesualdo, and V. Bevilacqua. Detection and segmentation of kidneys from magnetic resonance images in patients with autosomal dominant polycystic kidney disease. In D.-S. Huang, K.-H. Jo, and Z.-K. Huang, editors, *Intelligent Computing Theories and Application*, pages 639–650, Cham, 2019. Springer International Publishing. ISBN 978-3-030-26969-2.
- [115] M. F. Bobo, S. Bao, Y. Huo, Y. Yao, J. Virostko, A. J. Plassard, I. Lyu, A. Assad, R. G. Abramson, M. A. Hilmes, and B. A. Landman. Fully convolutional neural networks improve abdominal organ segmentation. *Proc SPIE Int Soc Opt Eng*, 10574, Mar. 2018.

- [116] Y. Chen, D. Ruan, J. Xiao, L. Wang, B. Sun, R. Saouaf, W. Yang, D. Li, and Z. Fan. Fully automated multiorgan segmentation in abdominal magnetic resonance imaging with deep neural networks. *Med Phys*, 47(10):4971–4982, Aug. 2020.
- [117] E. Koh, E. R. Walton, and P. Watson. VIBE MRI: an alternative to CT in the imaging of sports-related osseous pathology? *Br J Radiol*, 91(1088):20170815, Mar. 2018.
- [118] N. M. Rofsky, V. S. Lee, G. Laub, M. A. Pollack, G. A. Krinsky, D. Thomasson, M. M. Ambrosino, and J. C. Weinreb. Abdominal mr imaging with a volumetric interpolated breath-hold examination. *Radiology*, 212(3):876–884, 1999.
- [119] M. Xu, S. Yoon, A. Fuentes, and D. S. Park. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, 137:109347, 2023. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2023.109347>. URL <https://www.sciencedirect.com/science/article/pii/S0031320323000481>.
- [120] M. Irfan and I. Hameed. Deep learning based classification for healthcare data analysis system. pages 1–6, 10 2017. doi: 10.1109/BESC.2017.8256396.
- [121] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. Albumen-tations: Fast and flexible image augmentations. *Information*, 11(2), 2020. ISSN 2078-2489. doi: 10.3390/info11020125. URL <https://www.mdpi.com/2078-2489/11/2/125>.
- [122] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In S. Ourselin, L. Jostkowicz, M. R. Sabuncu, G. Unal, and W. Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46723-8.
- [123] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, and K. H. Maier-Hein. nnu-net: Self-adapting framework for u-net-based medical image segmentation, 2018.
- [124] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.
- [125] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.
- [126] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.
- [127] A. LUNTZ. On estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetika*, 1969. URL <https://cir.nii.ac.jp/crid/1573387449978580096>.

- [128] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson. Machine learning algorithm validation with a limited sample size. *PLOS ONE*, 14(11):1–20, 11 2019. doi: 10.1371/journal.pone.0224365. URL <https://doi.org/10.1371/journal.pone.0224365>.

Appendix A

Tables, figures, and schemes

A.1 Algorithm pseudo-code to obtain contour class from manually annotated ground truth

Algorithm 1 Contour Generator

```
1: function CONTOURGENERATOR(mask, bordersize, erosions)
2:   ErosionKernel = [3,3]
3:   ErodeImage = Erode(mask, ErosionKernel, erosion)           ▷ Create an eroded image
4:   DilationKernel = [2× bordersize + 1, 2× bordersize + 1]
5:   DilateImage = Dilate(ErodeImage, DilationKernel)
6:   DilateLine = Where(DilateImage== 255, 127.5, DilateImage) ▷ Replace 255 values to 127 for all
   pixels
7:   ContourImage = Where(ErodeImage> 127.5, 255, DilateLine) ▷ In the above DilateLine, convert
   the eroded object parts to pixel value 255
8:   return ContourImage
9: end function
10: for mask in maskdiretory do           ▷ For cycle to apply ContourGenerator function to every mask in
   directory
11:   ContourImage = ContourGenerator(mask, 3, 1)
12:   SaveImage(ContourImage)
13: end for
```

A.2 Pipeline illustration of the processes that the original dataset was subjected to the training process.

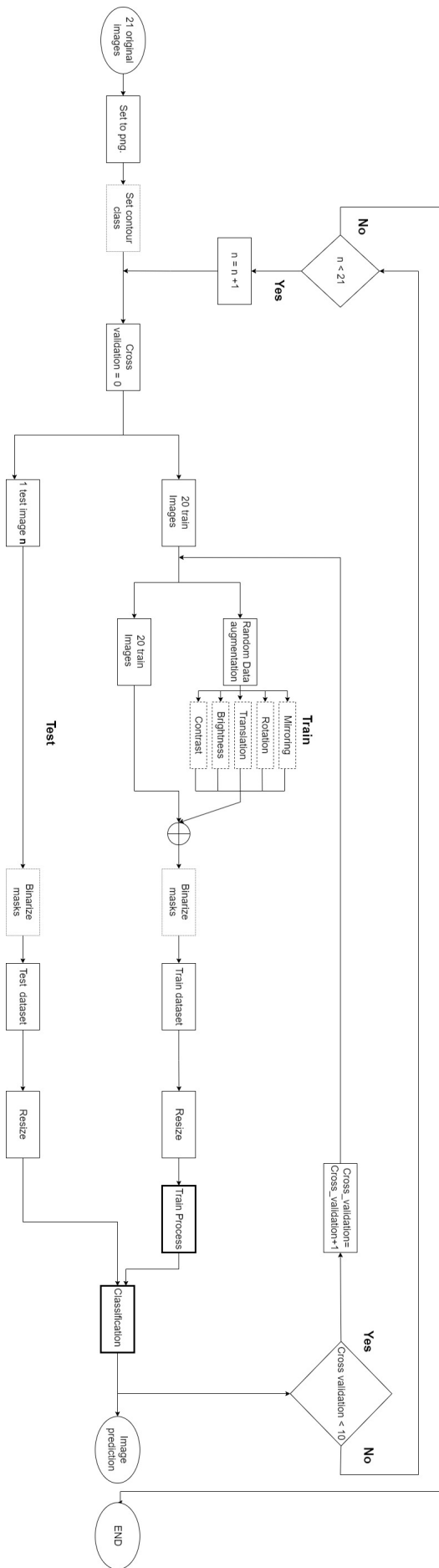


Figure A.1 : Pipeline illustration of the processes that the original dataset was subjected to the training process.

A.3 Train results

A.3.1 2 class

Table A.1: Evaluation results of Dice coefficient, Jaccard Index, Accuracy, Precision, Recall, Loss function and training time of the two classes segmentation models.

Segmentation Model	<i>DSC</i>	<i>IoU</i>	Accuracy	Precision	Recall	Loss	Time(s)
U-NET + CE	0.97742±0.00233	0.95584±0.00445	0.99823±0.00018	0.97655±0.00260	0.97828±0.00214	0.00500±0.00061	360.77383±9.52538
U-NET + FL	0.96969±0.00374	0.94118±0.00704	0.99767±0.00028	0.98861±0.00165	0.95147±0.00573	0.00055±8.605e-5	364.79818±2.74240
ATT U-NET + CE	0.97780±0.00185	0.95658±0.00354	0.99826±0.00015	0.97287±0.00242	0.98279±0.00133	0.12198±0.00064	473.16325±3.47271
ATT U-NET + FL	0.95541±0.06458	0.91867±0.06598	0.99677±0.00265	0.98323±0.02288	0.93180±0.06555	0.00663±0.00105	470.67213±3.08325

A.3.2 3 class

Table A.2: Evaluation results of Accuracy, Loss function and classification time of the multi-class segmentation models

Segmentation Model	Accuracy	Loss	Time (s)
U-NET + CE	0.99693 ±0.00031	0.00779 ±0.00076	369.44202±1.48043
U-NET + FL	0.99636 ±0.00046	0.00019 ±2.601e-5	366.54499±1.60239
ATT U-NET + CE	0.99655 ±0.00033	0.06887 ±0.00109	482.74754 ±2.52171
ATT U-NET + FL	0.99599 ±0.00043	0.00063 ±3.620e-5	484.26839 ±2.47802

Table A.3: Evaluation results of Dice coefficient and Jaccard Index of the multi-class segmentation models for each class in the training process.

Segmentation Model	<i>DSC</i>				<i>IoU</i>			
	Mean	Class 1	Class 2	Class 3	Mean	Class 1	Class 2	Class 3
U-NET + CE	0.96299±0.00370	0.99923±8.058e-5	0.91409±0.00865	0.97563±0.00238	0.93093±0.00641	0.99847±0.00016	0.84190±0.01459	0.95243±0.00452
U-NET + FL	0.95591±0.00566	0.99908±0.00012	0.89748±0.01321	0.97116±0.00368	0.91880±0.00959	0.99817±0.00023	0.81428 ±0.02162	0.94396±0.00694
ATT U-NET + CE	0.95828±0.00396	0.99914±8.795e-5	0.90315±0.00926	0.97255±0.00253	0.92279±0.00674	0.99828±0.00018	0.82353±0.0153	0.94658±0.00479
ATT U-NET + FL	0.95129±0.00531	0.9990±0.00011	0.88677±0.01236	0.96808±0.00350	0.91099 ±0.00880	0.99800±0.00022	0.79681±0.01969	0.93816±0.00654

Table A.4: Evaluation results of Precision and Recall of the multi-class segmentation models for each class in the training process

Segmentation Model	Precision				Recall			
	Mean	Class 1	Class 2	Class 3	Mean	Class 1	Class 2	Class 3
U-NET + CE	0.96304±0.00374	0.99923±8.135e-5	0.91407±0.00896	0.97581±0.00230	0.96294±0.00367	0.99924±8.210e-5	0.91412±0.00846	0.97545±0.00254
U-NET + FL	0.95502±0.00597	0.99913±0.00011	0.89392±0.01477	0.97202±0.00327	0.95681±0.00537	0.99904±0.00013	0.90107±0.01188	0.97030 ±0.00430
ATT U-NET + CE	0.958±0.00386	0.99914±9.94e-5	0.90110±0.00907	0.97375±0.00256	0.95856±0.00409	0.99914±8.33e-5	0.90520±0.00969	0.97135±0.00264
ATT U-NET + FL	0.94964±0.00551	0.99908±0.00011	0.88025±0.01326	0.96960±0.00341	0.95297±0.00519	0.99892±0.00012	0.89342±0.01186	0.96657±0.00379

A.4 Test results

A.4.1 2 class

Table A.5: Evaluation results of Dice coefficient, Jaccard Index, Accuracy, Precision, Recall, Loss function and classification time of the two classes segmentation models.

Segmentation Model	<i>DCS</i>	<i>IoU</i>	Accuracy	Precision	Recall	Loss	Time(s)
U-NET + CE	0.96577±0.00871	0.93393±0.01608	0.99732±0.00102	0.96723±0.01201	0.96469±0.01920	0.00843±0.00461	0.44850±0.10193
U-NET + FL	0.95985±0.01119	0.92301±0.02035	0.99689±0.00128	0.98021±0.00988	0.94074±0.02307	0.00096±0.00055	0.44170±0.09179
ATT U-NET + CE	0.96557±0.00740	0.93354±0.01377	0.99728±0.00092	0.96149±0.01337	0.97005±0.01637	0.12585±0.00332	0.67557±0.17771
ATT U-NET + FL	0.94399±0.09431	0.90222±0.09393	0.99605±0.00418	0.96967±0.09633	0.92027 ±0.09534	0.00731±0.00175	0.67212±0.15161

A.4.2 3 class

Table A.6: Evaluation results of Accuracy, Loss function and classification time of the multi-class segmentation models

Segmentation Model	Accuracy	Loss	Time (s)
U-NET + CE	0.99463±0.00203	0.00843±0.00461	0.44216±0.07360
U-NET + FL	0.99462±0.00203	0.00096±0.00055	0.43956±0.08247
ATT U-NET + CE	0.99486±0.00175	0.12585±0.00332	0.72847±0.22369
ATT U-NET + FL	0.99470±0.00206	0.00731±0.00175	0.69617±0.18541

Table A.7: Evaluation results of Dice coefficient and Jaccard Index of the multi-class segmentation models for each class

Segmentation Model	<i>DSC</i>				<i>IoU</i>			
	Mean	Class 1	Class 2	Class 3	Mean	Class 1	Class 2	Class 3
U-NET + CE	0.93667±0.01519	0.99861±0.00069	0.85398±0.03686	0.95743±0.00960	0.88755±0.02369	0.99722±0.00137	0.74694±0.05487	0.91850 ±0.01758
U-NET + FL	0.93627±0.01568	0.99861±0.00069	0.85289±0.03843	0.95729±0.00951	0.88696±0.02424	0.99722±0.00138	0.74542±0.05670	0.91824±0.01746
ATT U-NET + CE	0.93905±0.01277	0.99867±0.00061	0.85989±0.031	0.95858±0.00841	0.89114±0.02018	0.99735±0.00122	0.75550±0.04684	0.92057 ±0.01551
ATT U-NET + FL	0.93747±0.01528	0.99862±0.00076	0.85617±0.03647	0.95761±0.01036	0.88878±0.02369	0.99724±0.00151	0.75023±0.05388	0.91886±0.01883

Table A.8: Evaluation results of Precision and Recall of the multi-class segmentation models for each class

Segmentation Model	Precision				Recall			
	Mean	Class 1	Class 2	Class 3	Mean	Class 1	Class 2	Class 3
U-NET + CE	0.93605±0.01938	0.99872±0.00061	0.85294±0.04655	0.95649±0.01963	0.93780±0.01369	0.99849±0.00123	0.85608±0.03756	0.95884±0.01535
U-NET + FL	0.93486±0.02020	0.99879±0.00062	0.84998±0.04746	0.95583±0.02139	0.93823±0.014	0.99843±0.00133	0.85693±0.03968	0.95933±0.01593
ATT U-NET + CE	0.93937±0.01670	0.99869±0.00073	0.86072±0.04042	0.95869±0.01840	0.93924±0.01281	0.99865±0.00114	0.86013±0.03405	0.95894±0.01582
ATT U-NET + FL	0.93676±0.01907	0.99871±0.00088	0.85318±0.04420	0.95841±0.02155	0.93873±0.01513	0.99853±0.00130	0.86025±0.03929	0.95740±0.01716

A.5 Loss funtion results

A.5.1 2 class

Table A.9: Loss function evolution from the highest dice coefficient result for each of two class methods

Method	Epoch = 1,	2,	3,	5,	10,	15,	30,	45,	60,	75
U-NET + CE	0.2855	0.0947	0.0475	0.02022	0.0076	0.0056	0.0044	0.0043	0.0043	0.0043
U-NET + FL	0.0176	0.0030	0.0017	0.0010	0.0006	0.0005	0.0004	0.0004	0.0004	0.0003
ATT U-NET + CE	0.5743	0.4932	0.4404	0.3540	0.1965	0.1427	0.1233	0.1210	0.1208	0.1207
ATT U-NET + FL	0.0984	0.0648	0.0506	0.0328	0.0119	0.0074	0.0058	0.0056	0.0056	0.0056

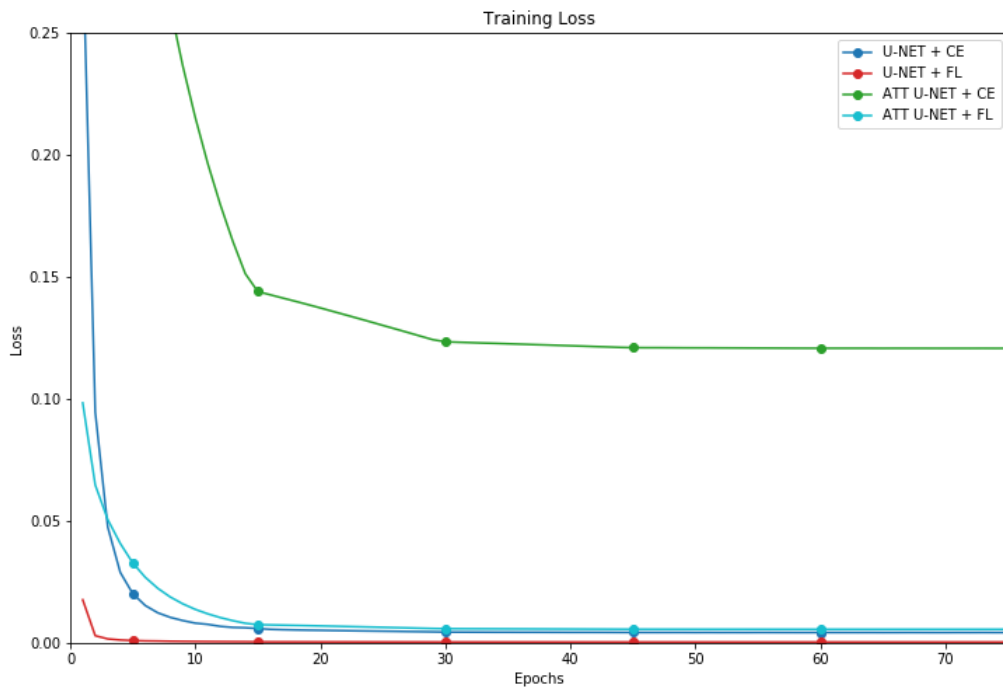


Figure A.2: Loss function evolution from the highest dice coefficient result for each of two class methods.

A.5.2 3 class

Table A.10: Loss function evolution from the highest dice coefficient result for each of three class methods

Method	Epoch = 1,	2,	3,	5,	10,	15,	30,	45,	60,	75
U-NET + CE	0.2917	0.0664	0.0346	0.0195	0.0102	0.0077	0.0064	0.0062	0.0062	0.0062
U-NET + FL	0.0100	0.0015	0.0007	0.0004	0.0002	0.0002	0.0002	0.0002	0.0001	0.0001
ATT U-NET + CE	0.8053	0.6015	0.4905	0.3311	0.1323	0.0855	0.0690	0.0668	0.0665	0.0664
ATT U-NET + FL	0.0246	0.0123	0.0083	0.0012	0.0008	0.0006	0.0006	0.0006	0.0006	0.0005

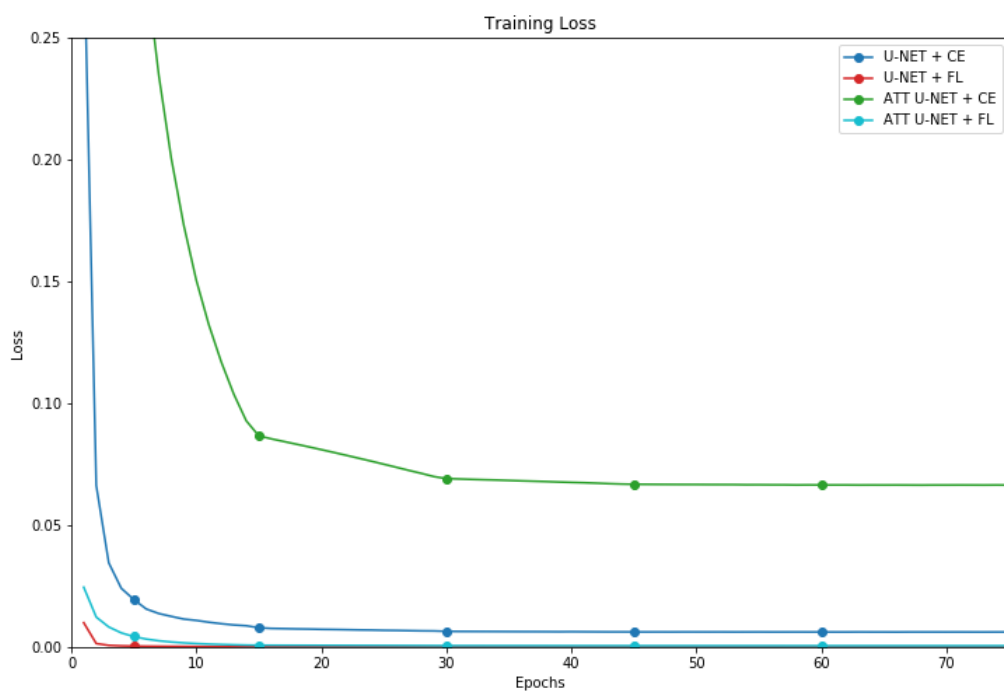


Figure A.3: Loss function evolution from the highest dice coefficient result for each of three class methods.