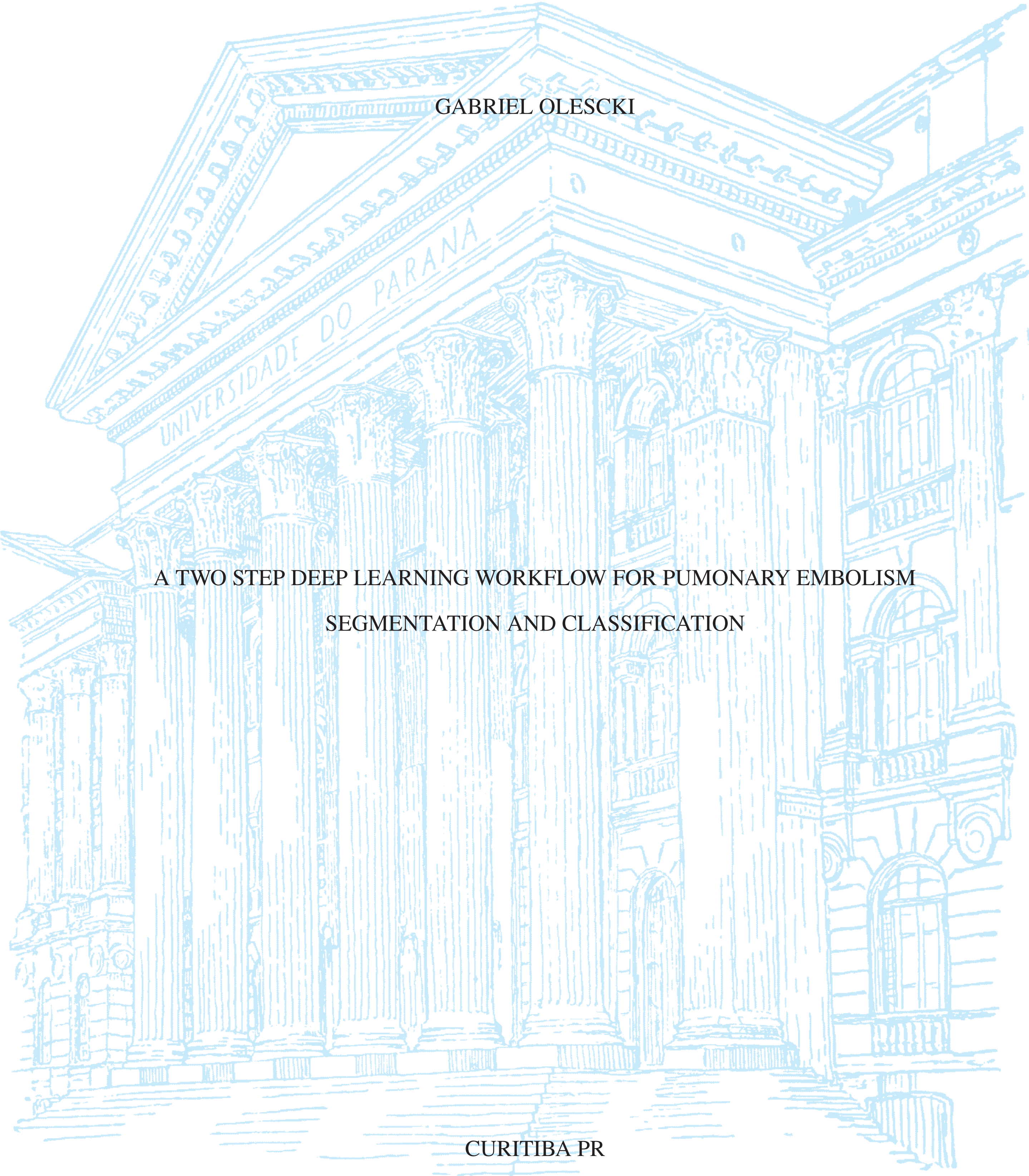
UNIVERSIDADE FEDERAL DO PARANÁ



GABRIEL OLESCKI

A TWO STEP DEEP LEARNING WORKFLOW FOR PUMONARY EMBOLISM SEGMENTATION AND CLASSIFICATION

CURITIBA PR

2021

GABRIEL OLESCKI

A TWO STEP DEEP LEARNING WORKFLOW FOR PUMONARY EMBOLISM SEGMENTATION AND CLASSIFICATION

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Lucas Ferrari de Oliveira.

CURITIBA PR

2021

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

O45a Olescki, Gabriel
A two step deep learning workflow for pumonary embolism segmentation and classification [recurso eletrônico] / Gabriel Olescki. – Curitiba, 2021.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2021.

Orientador: Lucas Ferrari de Oliveira .

1. Processamento de imagens. 2. Aprendizado do computador. 3. Inteligência artificial. 4. Embolia pulmonar. I. Universidade Federal do Paraná. II. Oliveira, Lucas Ferrari de. III. Título.

CDD: 006.3

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIENCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA -
40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **GABRIEL OLESCKI** intitulada: **A TWO STEP DEEP LEARNING WORKFLOW FOR PUMONARY EMBOLISM SEGMENTATION AND CLASSIFICATION**, sob orientação do Prof. Dr. LUCAS FERRARI DE OLIVEIRA, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 14 de Dezembro de 2021.

Assinatura Eletrônica
06/01/2022 09:45:02.0
LUCAS FERRARI DE OLIVEIRA
Presidente da Banca Examinadora

Assinatura Eletrônica
06/01/2022 14:18:06.0
DANTE LUIZ ESCUISSATO
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ -UFPR)

Assinatura Eletrônica
10/01/2022 08:53:24.0
LUIZ EDUARDO SOARES DE OLIVEIRA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

Embolia pulmonar está entre as principais causas de morte no mundo todo, de acordo com o Datasus 22% dos pacientes hospitalizados por embolia pulmonar acabam falecendo, trata-se de um trombo alojado em alguma região da vascularização arterial pulmonar. Para estabelecer o tratamento adequado e reduzir a mortalidade associada é necessário um diagnóstico rápido por parte da equipe médica, atualmente a forma de diagnóstico mais utilizado é a analise de imagens de tomografia computadorizada, devido a vários fatores como a sua velocidade de aquisição, alta disponibilidade dos equipamentos que fazem a sua captura e uma alta acurácia no diagnóstico. Um exame de tomografia computadorizada é composto de centenas de imagens que requerem a atenção do radiologista, pelo número alto de dados produzidos a análise de tais exames pode ser cansativa e levar a erros de diagnóstico devido a fadiga, ainda hoje a embolia pulmonar está entre as doenças onde se há mais erros diagnósticos. Nos últimos anos alguns sistemas computacionais de auxílio ao diagnóstico foram desenvolvidos para auxiliar radiologistas na detecção de trombos, tais sistemas têm se tornado de grande ajuda para um diagnóstico mais eficaz. *Deep learning* tem sido um dos tópicos mais comentados na área da visão computacional ultimamente, especialmente na área do processamento de imagens médicas, mais especificamente em aplicações de detecção e reconhecimento de imagens. Isso também se aplica no auxílio ao diagnóstico de embolia pulmonar, alguns trabalhos têm atingido resultados do estado da arte utilizando modelos complexos de redes neurais, que são capazes de identificar trombos dentro das imagens geradas pela tomografia, removendo outros ruídos que podem ser vistos como um falso positivo. O objetivo deste trabalho é desenvolver uma aplicação de *deep learning* capaz de encontrar tromboembolismos pulmonares em imagens de tomografias computadorizadas, a robustez do modelo permitirá que detecte trombos em exames de diferente origens. Se bem sucedido, o algoritmo produzido neste trabalho será capaz de auxiliar radiologistas em um diagnóstico rápido com uma alta probabilidade de acerto. Alguns testes preliminares já mostram que modelos de *deep learning* são capazes de discriminar embolias pulmonares, em uma base de dados pública contendo imagens de tomografia computadorizada de pulmão a rede foi capaz de encontrar vários trombos. Com um total de 35 exames, 28 foram usados para treinar o modelo e validar seus resultados, ajustando seus hiperparâmetros de acordo com os resultados, as outras 7 imagens foram utilizadas como teste, avaliado como o sistema se comporta quando recebe dados reais, atingindo um *Dice score* de 0.81 e uma acurácia de 84%, apesar de já apresentar bons resultados a modelo ainda possui espaço para melhores, pois ainda há diversos métodos de otimização que costumam melhorar os resultados das arquiteturas.

Palavras-chave: Processamento de Imagens Médicas. Deep Learning. Machine Learning.

ABSTRACT

Pulmonary embolism is one of the leading causes of death all over the world, according to Datasus the mortality rate of patients hospitalized due to pulmonary embolism is 22%. To break the clot and save the patient a fast diagnosis is required, that is the reason why computer tomography is used as a means to detect embolisms. A computed tomography exam is composed of hundreds of images that require an analysis from a radiologist, due to the high number of images this process can be tiresome and can lead to errors due to fatigue, pulmonary embolism remains one of the frequent misdiagnosis due to this fact. Over the years some computed aided systems had been developed aiming to help radiologists to see some missed clot, those systems had proven to be of great aid to an even faster diagnosis. Deep learning models have been increased significantly in many computer vision problems, especially in medical imaging, in image detection and recognition. This is also true in the classification of pulmonary embolisms, some works achieve a state of the art results by applying complex neural network models that can identify a clot from a whole tomography exam and remove any potential false positive found. The purpose of this work is to develop a deep learning application that is capable of discriminate pulmonary embolisms from a whole computed tomography volume, due to the use of deep learning a robust model can be developed that can generalize the process of embolism detection in different sources of data. If successful, this work will be able to aid radiologists in a fast diagnosis of pulmonary embolisms with a high discrimination probability. Some preliminary tests show that a deep learning architecture can discriminate pulmonary embolisms, a public dataset was used for validation of this architecture and can find several clots. With a total of 35 exams, 28 were used for training the model and validating its results, tweaking the models' hyperparameters with the results, the last 7 exams were used for testing the model, simulating how it should behave in a receiving unknown data, it achieves a Dice score of 0.81 and an accuracy of 84%, even if it got a relatively good result, it got plenty of room for improvement still, since many known improvement methods can still be applied in the architecture.

Keywords: Medical Image Processing. Deep Learning. Machine Learning.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AP	Average Precision
AUC	Area Under the Curve
CAD	Computer Aided Diagnosis
CCA	Connected-Component Analysis
CCL	Connected-Component Labeling
CNN	Convolutional Neural Network
CT	Computed Tomography
CTA	Computed Tomography Angiography
CTPA	Computed Tomography Pulmonary Angiography
FP	False Positive
FP/p	False Positive per Patient
HC	Hospital das Clínicas
HU	Hounsfield Units
IoU	Intersection over Union
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
MIL	Multiple Instance Learning
MLP	Multi Layer Perceptron
PCA	Principal Component Analysis
PE	Pulmonary Embolism
RGB	Red, Green and Blue
RNN	Recurrent Neural Networks
ReLU	Rectified Linear Unit
ResNet	Residual Network
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
UFPR	Universidade Federal do Paraná

LIST OF SYMBOLS

α	alfa, first letter of the Greek alphabet
β	beta, second letter of the Greek alphabet
γ	gama, third letter of the Greek alphabet
ϵ	episilon, fifth letter of the Greek alphabet
ω	ômega, last letter of the Greek alphabet
μ	mu, 12th letter of the Greek alphabet
σ	sigma, 18th letter of the Greek alphabet
Σ	uppercase sigma
∂	partial derivative symbol

CONTENTS

1 INTRODUCTION

Pulmonary embolism (PE) is a clinical condition where the patient has a thrombus (clot move from elsewhere in the body) in pulmonary vessels, this thrombus can harshly reduce or even interrupt the blood flow of the artery which can be fatal if left untreated (Carson et al., 1992). It's highly present, having an estimate of over 300,000 deaths per year in the US (**?**). The mortality rate (30%) can be reduced to, as low as 2% with an early diagnosis, showing that a fast and accurate diagnosis is critical to saving those lives ((Jha et al., 2013; Sadigh et al., 2011)).

The modality of choice for diagnosis of the PE is computed tomography (CT) exam, due to its fast acquisition and reliability using angiography (CTA) (Goldhaber and Bounameaux, 2012) and (Kligerman et al., 2018). The exam produces a 3-dimensional view of the pulmonary trunk, containing hundreds of images (slices) that have to be carefully analyzed by a radiologist, making the process highly time-consuming, besides that the interpretation of CTA exams requires a high degree of training due to the high amount of noise or other artifacts that can be confused for a PE. Unlike other pulmonary diseases that are observed with CT, such as lung nodules or emphysema, PEs are irregularly shaped and can appear in different regions of the lung (Rajan et al., 2020). All those factors corroborate with a high chance of diagnostic error, there is a difference of 13% of right diagnosis rate between radiologists that work overnight and in the daytime (Huang et al., 2020).

To try to mitigate those problems, and with the high occurrence of PE, a computed aided diagnosis (CAD) program is highly valuable; we propose an automated method capable of PE detection in CTA exams.

1.1 MOTIVATION

With the type of data produced from CTA exams, the fact that the diagnosis of PE is given mainly with the image observation, and the high number of recent deep learning methods using medical imaging, data-driven computed aided diagnosis of PE methods is desirable to assist the radiologist on its diagnosis, existing algorithms had already proved successful in improving the radiologist diagnosis rate (Tajbakhsh et al., 2019).

Those CAD for PE detection follow the steps: 1) Pulmonary vessel extraction from the CTA scan, 2) Generation a set of PE candidates, 3) Features extraction from the PE candidates and 4) Remove false PE from those candidates (Tajbakhsh et al., 2015), this last step is often achieved by supervised learning methods.

Early PE detection CADs tried to remove false positives (FPs) using hand-crafted features, as in (Masutani et al., 2002) and (Bouma et al., 2009), but they result in a high false positive rating. Authors in (Liang and Bi, 2007) introduced the tobogganing algorithm for the segmentation of the pulmonary vessel reducing the amount of FPs. (Ozkan et al., 2017) uses more information to get the features and uses a rule-based classifier to reduce FPs.

In more recent works, the use of convolutional neural networks for the FP reducing (Tajbakhsh et al., 2015) or in both the classification and finding PE candidates (Yang et al., 2019) have been tried, both with reducing the amount of FP.

Such methods depend on a good and discriminative dataset, but so far only one dataset is public available (Masoudi et al., 2018) with a good amount of documentation about it, however, this dataset is composed of 35 exams, of which 2 of them do not have any PE.

1.2 OBJECTIVES

This work creates a deep learning-based method capable of segmentation of PE candidates from the whole CTA exam, basing it on techniques that worked well over the years, and develop a new dataset with the annotation of PEs from radiologists from the *Hospital de Clínicas* (HC). The specific objectives are:

- To develop a segmentation method, based on Ronneberger's U-net (Ronneberger et al., 2015) aiming to localize the PE from the whole exam
- Develop a second neural network capable of classifying PE candidates from the first network into PE or non-PE
- To create a new public dataset with semantic annotation of PEs in CTA exams
- Evaluate the proposed method in with a new dataset, which will be openly available.

This work achieved a solid model for the extraction of a PE candidate in a CT exam and classifying this candidate into either a true positive PE or a false positive PE. The segmentation step of the model achieved a Dice Score of 0.81 and a Sensitivity of 0.89, and the classification step using the Random Forest classifier achieved an accuracy of 84%.

1.3 DOCUMENT STRUCTURE

This document is currently composed of four chapters, the first and current chapter aims to introduce the problem and what is the proposed objectives of this work. In chapter 2, it's introduced the fundamentals to understand the concepts applied in this work and related ones.

A review of the current state of the art and the timeline of PE detection CADs is presented in chapter 3. Chapter 4 then describes this works proposal, some early results obtained and the proposed schedule.

2 FUNDAMENTAL BACKGROUND

This chapter aims to explain some concepts and know subjects from different areas, such as computer science, mathematics, physics, and even medicine. The concepts here represented serve as a basis for understanding the methodology applied in this work (chapter 4) and other similar (chapter 3).

2.1 VENOUS THROMBOEMBOLISM

Venous thromboembolism is a syndrome that includes both deep vein thrombosis (DVT) and pulmonary embolism (PE). DVT occurs when a clot is formed, most commonly in the legs, thigh, or pelvis. A PE occurs when a detached clot travels through the bloodstream to the lungs, blocking an artery (NHLBI, 2011). Pulmonary embolism can be fatal in some severe cases due to a sudden cardiac death (Goldhaber and Bounameaux, 2012).

PE is the third most common cause of death due to sudden cardiac death, standing behind only ischemic heart disease and ischemic stroke, although it is more common (Raskob et al., 2014). PE has a frequency of 3.5 per 1000 population in high-income countries and 1.1 per 1000 population in both low- and middle-income countries (Jha et al., 2013). In the United States and Canada around 12,000 deaths were recorded in 2017, and an average of around 39,000 pulmonary embolism related deaths in Europe during 2013 and 2015 (Barco et al., 2020b) and (Barco et al., 2020a). In Brazil, autopsy data shows that PE prevalence is similar as in the United States (Terra-Filho and Menna-Barreto, 2010)

Due to a nonspecific set of symptoms the diagnosis can often be inaccurate, due to the fact that most of the symptoms are also common in other lung-related diseases (Dalen et al., 2002). The misdiagnosis however can be fatal, with a mortality rate lower than 10% between patients that were correctly diagnosed (Dalen et al., 2002).

Diagnosis of PE use mainly non-invasive diagnostic techniques (Figure 2.1). A patient with suspected PE goes through a clinical evaluation aiming to produce a clinical probability, which is decided empirically or with the help of a scoring system (Goldhaber and Bounameaux, 2012). The two main scoring systems used in this cases are the Wells score and the revised Geneva score, the Wells score can be used for DVT diagnosis also, patients are then classified as low, intermediate, or high risk based on their score (Goldhaber and Bounameaux, 2012). Low and intermediate risk patients go through the D-dimer test, which verifies the presence of the protein found in clots, patients which tested positive in the D-dimer and with and high risk score has to go through an angiotomography of the pulmonary trunk (Goldhaber and Bounameaux, 2012).

2.2 COMPUTED TOMOGRAPHY

Computed tomography has been over the years the main method for early diagnosis of several diseases (Buzug, 2011). Different from traditional x-ray images, CT exams produce a sectional view of the patient. This view comes from a reconstructed x-ray bean shaped like a cone or a fan, that spins and moves continually over a specific body part, an image reconstruction method is applied on the cast projections (or shadows), applying different techniques from physics, mathematics, and computer science, a 2D reconstruction of those shadows is obtained (Buzug, 2011).

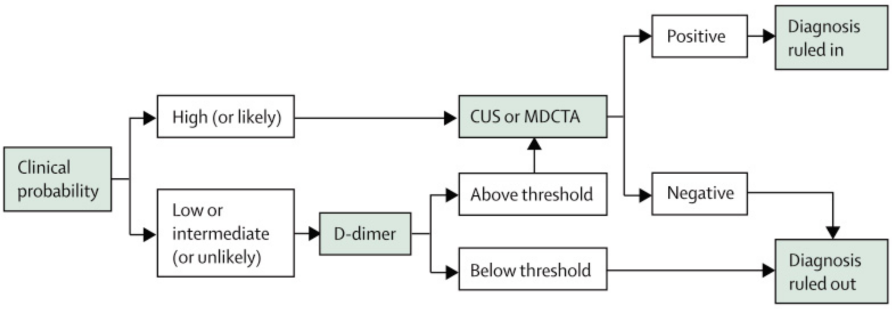


Figure 2.1: Diagnostic pipeline for deep vein thrombosis or pulmonary embolism, being compression ultrasonography (CUS) for deep vein thrombosis suspects and multi detection computed tomography angiography (MDTCA) for pulmonary embolism suspects .Source: (Goldhaber and Bounameaux, 2012)

A full CT exam is composed of hundreds of slices, each slice being one of the 2D reconstruction images. Views from different anatomical planes of the exam can be reconstructed, the coronal plane (divides the body into front and back), sagittal plane (divides the body into left and right), and axial or transverse plane (divides the upper and lower parts of the body).

The final CT image representation is a three-dimensional array, where each voxel (spatial pixel) is calculated with the attenuation value (μ) from the tissues in which the x-ray has been through (Figure 2.2), this approach has been developed by Hounsfield, that gives the name of the scale Hounsfield Units, each voxel gets its corresponding HU value from the given equation (2.1), where the attenuation values μ_{water} is the assigned HU value for water and K is a constant scaling factor, usually equals to 1000 (Seeram, 2018).

$$HU = \frac{\mu - \mu_{water}}{\mu_{water}} K \tag{2.1}$$

The scale maps different attenuation values into grayscale values, the minimum value of -1000 HU representing air, 0 HU is assigned for water, values bigger from the 0 and 1000 HU range are for different body tissues, and from above 1000 HU are for bones. There is no upper limit in the HU scale, but it ends at around 3000 HU (Seeram, 2018).

The final CT values are then converted, using digital image processing algorithms, to a grayscale image (figure 2.3, with lower attenuation values being black, higher values white and the values in between with shades of gray. Since bones attenuates more radiation, in the image representation it has as white color, while the air attenuates very little to the radiation, it has a black color (Seeram, 2018).

Since the human eye can distinguish around 30 shades of gray (Kreit et al., 2013), the HU values are normalized in different scales, usually of 256 or 512. In some applications, it is also possible to extract different intervals from the HU values, since some tissues can have similar HU values this partitioning can enhance the contrast of similar tissues (Buzug, 2011). Those intervals are defined by the *window width* and *window level*, the window width represents the lower and upper bound of the interval, and window level is the center of the range. Once defined both width and level, the image will represent in a gray-scale the values in between the lower and upper bound, enhancing the contrast of different tissues depending on the selected window (Seeram, 2018).

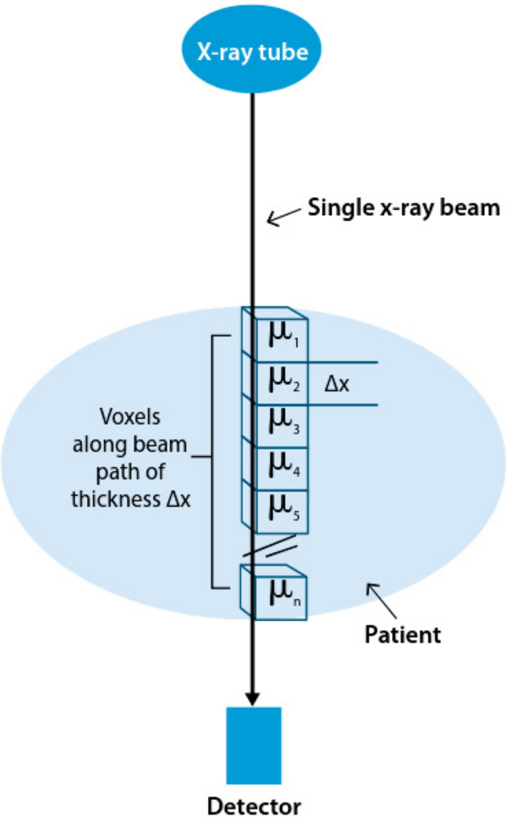


Figure 2.2: Representation of a single x-ray passing through a patient, obtaining the attenuation value (μ) from the voxels of the volume. Source: (Seeram, 2018)

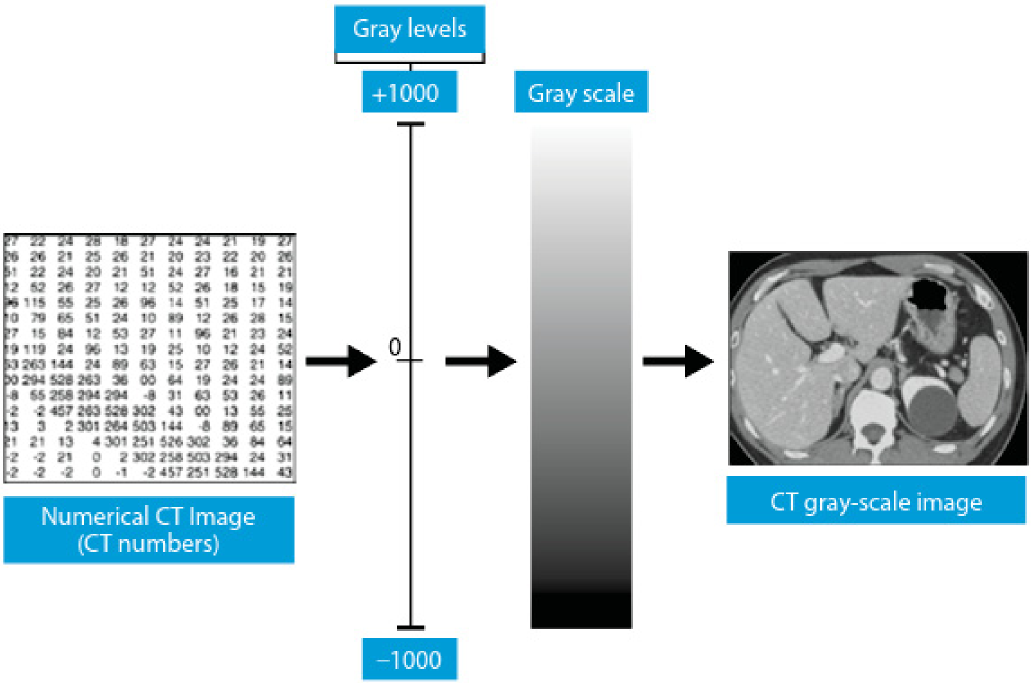


Figure 2.3: The numerical CT Image (left), representing the raw numerical values calculated from the attenuation and the normalized with μ_{water} value and its grayscale representation (right). Source: (Seeram, 2018)

2.2.1 Computed tomography angiography

Computed tomography angiography (CTA) is a technique used paired with CT imaging, aiding radiologists to better see clots or other artifacts inside the blood vessels with a CT image with enhanced contrast (Figure 2.4). It consists of injecting a contrast in the patient veins before the tomography, producing a CTA image with higher contrast in some vessels. It's used in CTs of the pulmonary trunk as a form o diagnosis for PE (Masoudi et al., 2018).

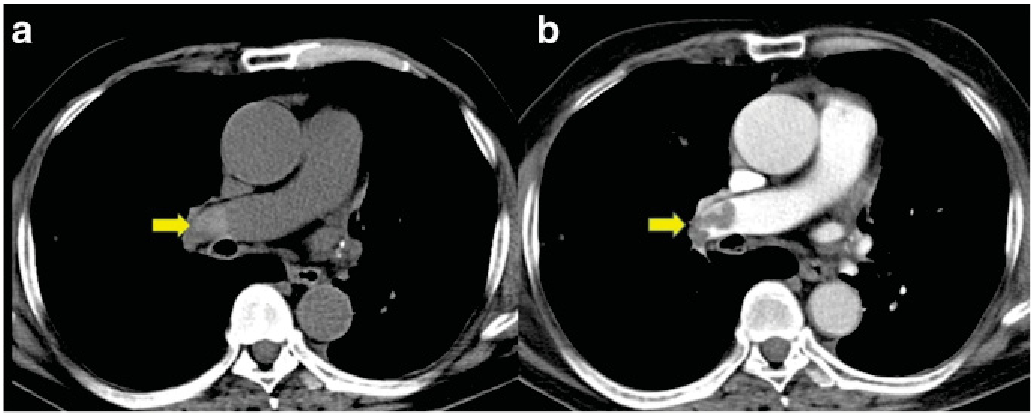


Figure 2.4: a) PE in a CT image. (b) Same PE using with a CTA exam.Source: (Chien et al., 2019)

The CT image is taken after a certain time after the radiocontrast injection, this contrast is radiosensitive, causing it to light up the area inside the blood vessel. To scan the correct area at the correct time, the scanner either automatically detects the amount of contrast present or a previous bolus injection is made, measuring the speed which it takes to reach the selected area, and then the rest of the contrast is applied (Kumamaru et al., 2010). For pulmonary CTAs it's necessary a fast acquisition since the patient needs to hold his breath during the exam, avoiding artifacts and noise from the movement (Kumamaru et al., 2010).

2.3 IMAGE PROCESSING

Image processing is a field of computer science where different algorithms and mathematical functions from linear algebra and calculus are applied to improve image quality for the human eye or another algorithm. The definition of an image according to (Gonzales and Richard, 2018) is a two dimensional function $f(x, y)$ from a plane on which x and y points are found, and each $f(x, y)$ value being its value from a discrete scale. For grayscale images only one plane of x and y is represented (one channel), while color images come from a combination of values from different planes, some examples are the RGB format, were each value from each plane represents the intensity of a color, the first plane is the red color, while the other two are the green and blue. All those intensity channels values vary from 0 to 255 of its $f(x, y)$, also known as pixel.

2.3.1 Normalization

2.3.1.1 Min-max Normalization

In order to normalize a signal array to pixel values (i.e. 0 to 255), a linear function can be applied to each of those values, mapping then to a pixel value. One algorithm that executes this task is implemented in the library *SimpleITK*, from (Yaniv et al., 2018).

The linear function is given by the following equation:

$$outputPixel = (inputPixel - inputMin).\frac{(outputMax - outputMin)}{(inputMax - inputMin)} + outputMin \quad (2.2)$$

Where *inputPixel* is the raw value to be scaled, *inputMin* and *outputMin* are the minimal and maximal raw value of the input signal, *outputMin* and *outputMax* are the biggest and smallest value for the desired output (e.g. min value of 0 and max value of 255 to scale the signal in pixels).

2.3.1.2 Standardization

Another common strategy to change the original signal value is to apply a standardization, also know as Z-score normalization. It aims to rescale the input data to a 0 mean distribution, with the values between -1 and 1. It's defined by the following equation:

$$x' = \frac{x - \mu}{\sigma} \quad (2.3)$$

Where x is the old value, μ is the mean, and σ is the standard deviation from all the data and x' is the new value of x.

It is commonly used in deep and machine learning algorithms (Grus, 2019) and it can help the convergence speed of some algorithms.

2.3.2 Segmentation

Segmentation in image processing is basically the partition of an image into pieces that satisfy a given condition, each pixel in the output partitioned image share a common feature. Segmentation algorithms can be divided into two main categories, with discontinuous or similar values as input 2.5. In the first category, it's assumed that the boundaries are heterogeneous between each other and the background, an example of this type of segmentation is *edge-based*. While the second category separates the image following predefined criteria, being a mask (a copy of the image, but it has value 1 for regions to be separated and 0 everywhere else) or a rule, being called *region-based* segmentation (Gonzales and Richard, 2018).

2.3.3 Convolution

Convolution can be simply defined by the combination of two mathematical functions in order to produce a new third one, the result function represents the modification of the first function over the second. In image processing, convolution aims to change a pixel value to a new one based on its neighbors and a kernel. Kernels are matrices with dimension equal or lower than the original image, each element of the kernel dictates its function, being for noise correction, sharping, border detection, or other specific functions (Gonzales and Richard, 2018).

The kernel aligns its central value with a pixel of the original image, then it starts sliding over the image, executing mathematical operations between the kernel's values aligned with the image, assigning a new value to pixel aligned with the center of the kernel (figure 2.6)

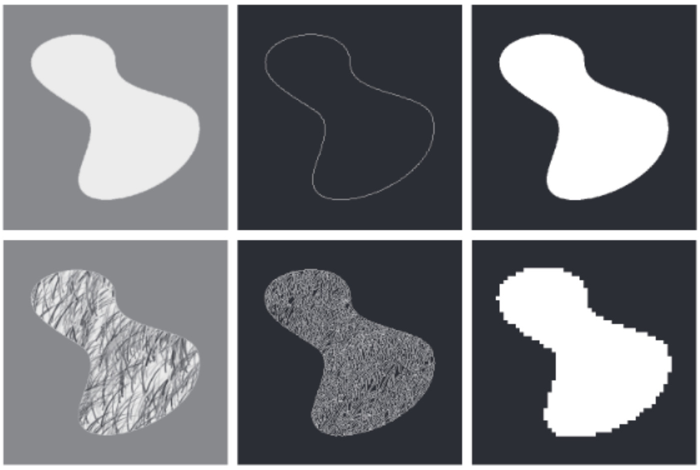


Figure 2.5: The original image in the top left, on the top middle a boundary based on discontinuity, in the top right the result of a segmentation between the last two images. At bottom left an image with a texture, at bottom middle portion the result of a discontinuity segmentation and at the bottom right portion the result of a segmentation based on the region properties. Source: (Gonzales and Richard, 2018)

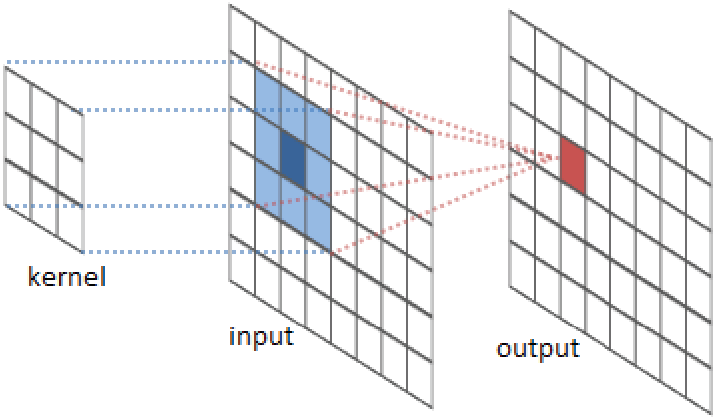


Figure 2.6: Example of a kernel (in the left) combining its values with a input image (blue), resulting in a new value for the center aligned pixel (red) Source: (Karpathy, 2018)

2.4 MACHINE LEARNING

Machine learning is a branch from the classic artificial intelligence, algorithms from this branch can learn from data and create inferences based on what it learned. Such algorithms are becoming increasingly present in our day-to-day lives, are widely used in e-commerce and product recommendation applications, computer vision, linguistics, economics, and other fields. The output of such algorithms can be presented as a function $y(x)$, where the algorithm receiving the input x generates a prediction $y(x)$, the function y is modified during the training process, where the algorithm observes the known dataset, the machine learning aims to be able to produce the correct $y(x)$ for a x not viewed during the training step (Bishop, 2006).

Conventional machine learning methods can be limited, the algorithms cannot extract meaningful features from the data, being dependent on other pre-processing methods and feature extraction algorithms (LeCun et al., 2015). The feature extraction process is usually aided by human knowledge, with sometimes the need for a specialist in the original data (e.g. radiologists in medical imaging) that helps to create meaningful and discriminative representations of the dataset.

There are three big branches of machine learning, the first being supervised learning, in which each instance of the original dataset has a label representing the class to which this instance belongs. An example is digits classification, in which each handwritten digit is assigned to a discrete label y, the aim of the machine learning algorithm, in this case, producing a prediction of an unseen instance of the handwritten dataset and assigned the correct value of y, being considered a classification problem, while in cases where the prediction is a continuous value, its called a regression problem (Bishop, 2006).

In unsupervised learning the original dataset does not have a label for each instance, in such cases machine learning algorithms must group the data into similar clusters, finding similarities in the original data. Such applications can be helpful in scenarios where no much is known in the original dataset. The last category of machine learning algorithms is reinforcement learning, which aims to find the best action for a current environment, the selection of actions are based on score attributed to each possibility, so based on positives and negatives scores the algorithm adjusts its actions (Géron, 2019).

Supervised learning algorithms need a huge amount of data to be able to give good results, otherwise, it may know only a small number of features that can differentiate the data, this is known as *overfitting*, in this case, the algorithm knows only a small portion of the real data, and cannot generalize its classification to other samples of the data (Géron, 2019). To mitigate this problem techniques such as *data augmentation* were develop, in which simple operations like rotation, translation, and noise addition in the original set can help create artificial instances of the data that can improve the generalization of the algorithm.

2.4.1 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm it's one of the simplest to understand machine learning algorithms, it is an instance learning algorithm, meaning that when evaluates a new unseen data, it compares to the rest of the know training set.

Figure 2.7: Example of the KNN algorithm, in which two features (feature 1 and feature 2) are used to discriminate triangles and squares, 3 nearest training instances from X are found through euclidean distance, in this case X is considered a triangle. Source: (Géron, 2019)

The algorithm calculates the distance (usually the Euclidean) between the new instance of unseen data to the k nearest neighbors or the k instances of data on which the distance is the lowest (figure 2.7), then it simply counts the nearest neighbors and assigns the label of the majority to the new instance (Géron, 2019).

2.4.2 Support Vector Machines

Support vector machine (SVM) is one of the most popular algorithms in the field, it's a model based algorithm, meaning that it learns a model capable to differentiate the data, creating a decision frontier.

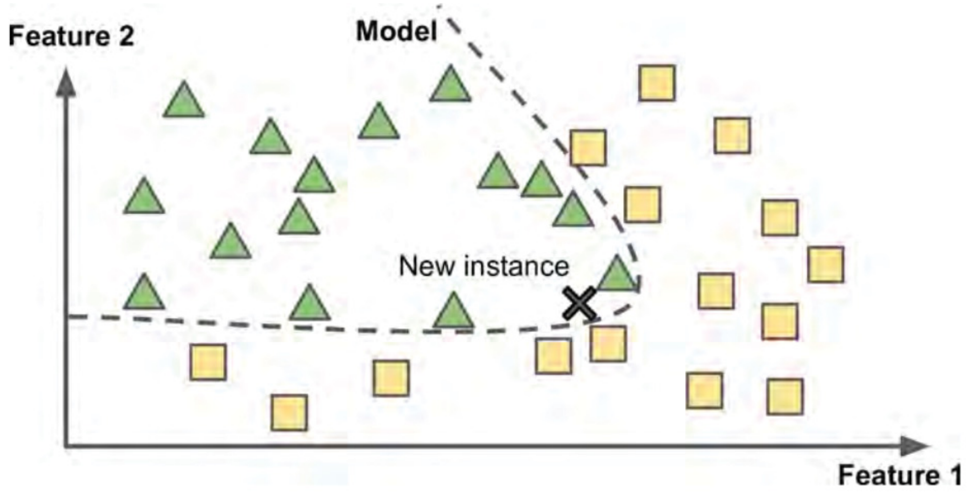


Figure 2.8: Example of the SVM algorithm with a polynomial kernel, Source: (Géron, 2019)

The classifier aims to create a margin that separates different classes based on its features, it attempts to create a wide decision frontier which is the furthest away from instances of data near to the other class. The original SVM algorithm applies a linear separation of the data, but in some cases, the data cannot be separated by a line, for these cases, other implementations of the SVM sees the data into a hyperplane, through the so-called *kernel trick*, which then the SVM can create a curve that separates the data (2.8), the polynomial and Gaussian are the most commonly used kernels (Géron, 2019).

2.4.3 Principal Component Analysis

The Principal Component Analysis (PCA) algorithm aims to reduce the dimensionality of a dataset. It works by identifying a hyperplane on which the data can be projected, by using the eigenvectors and eigenvalues it can find an axis which on which the original data has the highest variation (i.e. the data is more separated). It also finds the line that is orthogonal to the found axis, by projecting the data in the plane in which these lines are found we can see a projection of the data (Géron, 2019).

This algorithm is widely used in machine learning algorithms, it can help to reduce the number of features that represent a dataset or even helps to visualize the data. Increasing the features of a machine learning pipeline also increases exponentially the complexity of the model, each new feature adds a new dimension, needing more training examples that can fill every possibility of features, this is known as the curse of dimensionality (Géron, 2019), and that is why algorithms like PCA can be helpful in the training of machine learning.

2.4.4 Hold-out and cross-validation

In machine learning different methods of separation of data can be utilized, the optimal choice can depend of the amount of available data and its complexity.

One of the simpler methods is the *Hold-out* separation, in which the dataset is split into two parts, a training set and a test set, the training is commonly separate into a validation set.

The validation set in this case is used for observing how the model behaves when applied to unseen data, so the algorithm parameters can be adjusting according to its results. While the test set should be only used for the final evaluation of the model, representing how it should behave unknown data (Géron, 2019).

2.4.5 Cross-validation

Another evaluation method commonly used in machine learning is the cross-validation, it also splits the dataset into training and testing set, but the training set is randomly split into K predefined folds, then it trains the model using K-1 folds and uses the remaining one for validation, the repeats the training using a different fold for validation, resulting in K evaluation scores (Géron, 2019).

2.4.6 Neural networks and Deep Learning

Neural networks and deep learning have gained much coverage in the last few years, models know as Deep learning have been achieving the state of the art results in many pattern recognition and natural language (LeCun et al., 2015). Different from traditional machine learning, a deep learning architecture is built with the junction of many different layers of modules, most of these modules are for direct learning while others help the learning process.

To better understand the most works cited in chapter 3, a brief introduction to some concepts that serve as building blocks for deep learning network will be presented.

2.4.6.1 Fully Connected Network *(FCN)*

Also known as *Multi Layer Perceptron* (MLP), consists of multiple interconnected perceptrons. The perceptron is a linear classifier, in other words, it can create a line that best separates the data. It operates simply: given an input array x and a weight w and a bias b, a prediction $\hat{y}$ are created by the equation 2.2.

$$\hat{y} = w^T x + b \tag{2.4}$$

This prediction is compared to the real y value, and given the error, the weights and bias are updated. This process repeats until the prediction $\hat{y}$ is equal to y. For the fact that the perceptron creates a line, it cannot solve problems which require a non-linear separation (e.g. XOR problem) (Maier et al., 2019).

To solve problems that require a non-linear boundary, a combination of several perceptrons, each receiving a part of the input data to create a so-called *hidden unit*, a combination of *hidden units* followed by non-linear functions (figure 2.9) can distinguish more complex data, due to the introduction and can create curves as separation (Bishop, 2006).

A MLP with at least two *hidden layers* is capable to approximate any function inside a continuous function space (Cybenko, 1989), the more complex the problem the bigger number of layers are usually required.

2.4.6.2 Convolutional Neural Network *(CNN)*

Even though MLP can achieve good results, they are not recommended in applications with a huge amount of input data, since each instance of the input would need a weight, increasing the memory necessary to represent all the weights necessary. In the case of convolutional neural

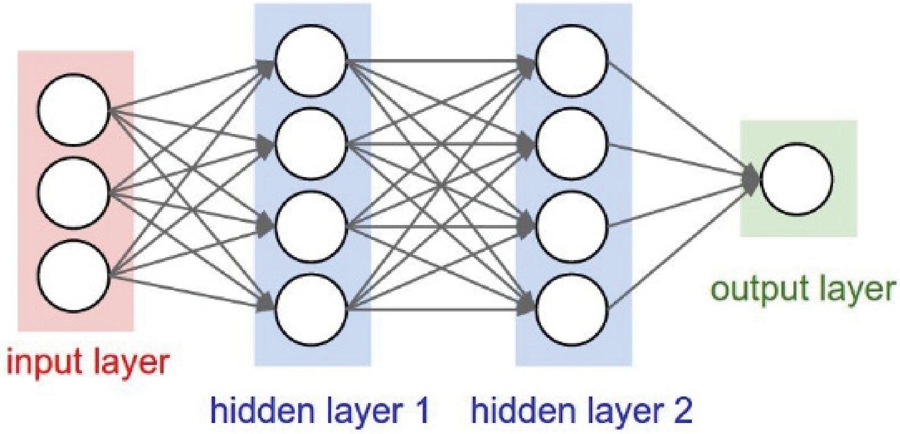


Figure 2.9: Multi Layer Perceptron with two *hidden layers*. Circles in the red layer (left) are the input x, circles in blue (middle) represent perceptrons and the last layer in green represents the prediction output $\hat{y}$ Source: (Karpathy, 2018)

networks, the individual weights of the MLP are substituted to convolution kernels (figure 2.10) (LeCun et al., 2015).

The idea of CNNs derived from classic algorithms for image classification, a defined kernel that can extract features is applied to an image. The CNN can learn the kernel that extracts the feature that best describes the input data (LeCun et al., 2015).

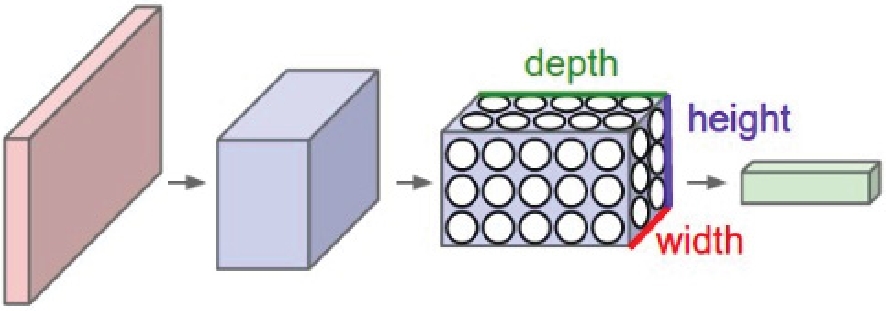


Figure 2.10: Convolutional Neural Network, in the left a input image with a single layer, after the application of two convolution kernels (blue) and generation a output in the right. Source: (Karpathy, 2018)

2.4.6.3 Pooling

The Pooling technique aims to reduce the size of an input data, attempting to summarize similar features into a single one. Works similarly to a CNN but with simpler operations. Used plenty in CNNs and other variations always to reduce the scale of the input (Maier et al., 2019).

Instead of kernels, a single function is applied into a sliding window over the input image and producing a single output, functions usually applied in the pooling operation are the max (MaxPooling) or the average (AveragePooling). An example of the MaxPooling operation can be seen in figure 2.11 (Maier et al., 2019).

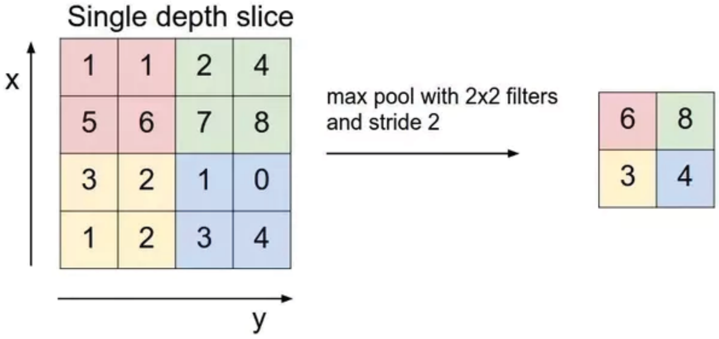


Figure 2.11: MaxPooling filter operation with a 2×2 size and stride 2. Source: (Karpathy, 2018).

2.4.6.4 *Activation functions*

As stated before, after each weighted operation in a Deep learning mode, there is a layer which applies a non-linear function to the input, the so-called activation layers. Each activation layer has its uses and can be used on different occasions.

2.4.6.5 *Sigmoid*

One of the early functions used to introduce non-linearity to neural networks, fits the input data into a curve with values between 0 and 1, defined by the equation:

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

2.4.6.6 *ReLU*

Abbreviation of Rectified linear unit, widely used in CNNs and MLP, its simply defined by the equation:

$$f(x) = max(0, x) \tag{2.6}$$

It commonly produces better results when compared with other activation functions, this can be true since the ReLU solves the vanishing gradient problem for deeper networks (LeCun et al., 2015). The vanishing gradient happens when several applications of activation functions, that reduce the maximum value of the data to 1 (i.e. sigmoid), are applied in sequence reducing the values exponentially (Maier et al., 2019).

2.4.6.7 *Leaky ReLU*

Even though the ReLU solves the vanishing gradient problem, it brings another problem called the dying ReLU. Since the weights are randomized in the first stage of training, too many negative values can be attributed to the input data, making the output of the ReLU function mostly zeros and the network can no longer learn from them.

To mitigate this problem, the Leaky ReLU was develop, it works similarly as the original Relu, but instead of giving 0 to negative values, it applies a small inclination in the line (figure 2.12), its defined by the following equation:

$$f(x) = 1(x < 0)(\alpha x) + 1(x >= 0)(x) \tag{2.7}$$

Were alpha is a constant which defines the angle of the line for the negative values, due to the fact it solves the dying ReLU problem it has been widely used in several deep learning works (Maier et al., 2019).

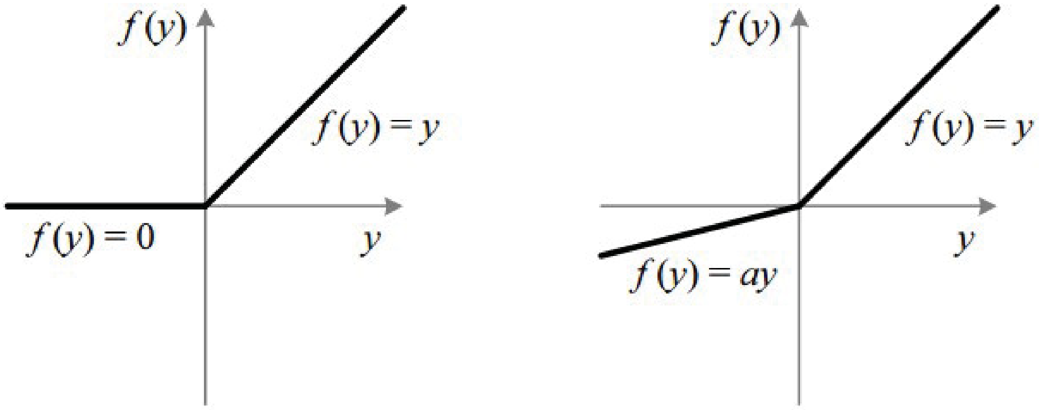


Figure 2.12: Comparison between ReLU (left) and Leaky ReLU (right). Source: (Sharma, 2017)

2.4.6.8 Loss Functions

Loss functions (also known as cost functions) are statistical functions that measure the distance between the prediction output of a CNN ($\hat{y}$) to its original value (y), the focus during training is to minimize this distance. One of the most used loss functions in Deep learning is the cross entropy, which consists of a variation of the log-likelihood function. Its described by the following equation:

$$H(p,q) = -\sum_{x} p(x)\ log\ q(x) \tag{2.8}$$

On which two probabilistic distributions p and q are compared and give a value, the more alike the distributions are the close $H(p,q)$ will be to zero (Géron, 2019).

A loss usually used for segmentation tasks is the Intersection over the Union (IoU), it aims to calculate the distance of the current segmentation to the real one, it tries to maximize the value of the intersection of the predicted mask and the real one, if the value is 1.0 it means that both segmentations are the same.

Other commonly used loss function for segmentation tasks is the Dice Loss, the loss can be described as:

$$Dice = 1 - 2\frac{|X \cap Y|}{|X| + |Y|} \tag{2.9}$$

Where X is the prediction and Y is the real value, this loss can be helpful in unbalanced datasets since it only updates when a positive sample of X is found.

2.4.6.9 Optimizers

Optimizers are functions that update the weights from a neural network based on the gradient of its loss function. Since the final goal of a deep learning model is to minimize the loss function, optimizers based on gradient descend methods give good results, the first derivative of a function tells on which direction its converging, we can reduce the value of $f(x)$ by moving x in the opposite direction of the derivative signal (Goodfellow et al., 2016). One classic algorithm that performs this task is the Stochastic gradient descent (SGD), where the gradient is weighted by a learning rate factor, which dictated the convergence speed of the model, a learning rate too high can provoke leaps that miss a minimal of the function, while a low learning rate can take too long to reach that point (Géron, 2019).

One of the know problems from optimizers is when a local minimum is reached, where $\frac{\partial y}{\partial x}$ reaches zero, the network stops learning at this points since the gradient points to no direction. To avoid reaching local minimums techniques such as *momentum* is applied, on which a calculation based on the last few gradients and the current one is made, helping find which direction the gradient was headed (Goodfellow et al., 2016)

One of the most used optimizers currently is the Adaptive Moments (Adam), which has an adaptive learning rate for different hyperparameters of the most common optimizers, even though the learning rate that multiplies the amount of influence from the gradient still needs to be found by trial and error. Adam is one of the easiest to use and gives good results when compared with other optimizers (Goodfellow et al., 2016) and (Géron, 2019).

2.4.6.10 Backpropagation

With the introduction of even deeper models, a found issue was on how to attribute credits since intermediary perceptrons do not have a value from which a loss can be calculated. To solve this problem, the *backpropagation* algorithm solves the problem to attribute credits using the gradient of the last layer, which propagates back to all the *hidden layers* (Rumelhart et al., 1986).

The algorithm is based on the chain rule, the gradient values are applied recursively through each perceptron of the network, computing the gradient with respect to the weights and propagation backward the gradient with respect to the input of the current perceptron, i.e. each neuron receives a backward pass $\frac{\partial L}{\partial y}$ from the neuron in its front, and propagates backward the new gradient ($\frac{\partial L}{\partial y}\frac{\partial y}{\partial x}$), better explained in figure 2.13 (Maier et al., 2019). With the propagated gradients, the optimizers can update the current weights of a neuron.

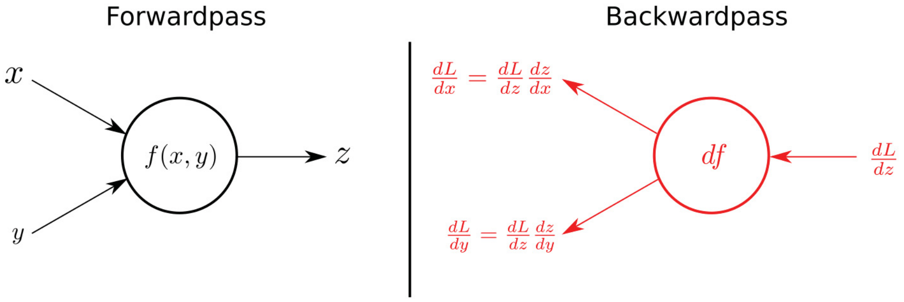


Figure 2.13: Example of a forward pass (left) and backward pass (right) of a neuron, which gets as input the values x and y and compute the output z in the forward pass. In the backward pass it propagates back the gradient with respect to the input x ($\frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$) and the input y ($\frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$) Source: (Agarwal, 2017)

2.4.6.11 Batch Normalization

Batch normalization is a technique that increases the convergence rate of a neural network, ensuring that the input of the linear operations of the network is always with a zero mean. The use of batches for training in neural networks aims to optimize the training time, instead of training one instance of the database at a time, the network is inputted with a batch of elements, usually values in a base 2 value (i.e. 32, 64, 128). The technique was proposed to address the vanishing or exploding gradient problem (Géron, 2019).

Proposed by (Ioffe and Szegedy, 2015), it normalize the data using the mean and standard deviation of the batch, where it calculates the new value of x_i from a batch $\mathcal{B} = \{x_1 ... m\}$ following the equations:

$$\mu_{\mathcal{B}} = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{2.10}$$

$$\sigma_{\mathcal{B}}^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \sigma)^2 \tag{2.11}$$

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{2.12}$$

$$y_i = \gamma \hat{x}_i + \beta \tag{2.13}$$

Which ϵ is a constant, γ and β are learnable variables, if the batch normalization harms the convergence of the network, the γ and β values will change corresponding to the received gradients.

2.4.6.12 Dropout

Neural networks tend to suffer from overfitting quite often since it can learn a good decision frontier over the training data with ease, over the years different techniques were developed aiming to mitigate overfitting. One of the most effective techniques is the dropout, which simples shuts down random neurons from the network at a given probability (figure 2.14), by doing that the networks purposefully does not learn some aspects of the training data, resulting in a more generic model (Géron, 2019).

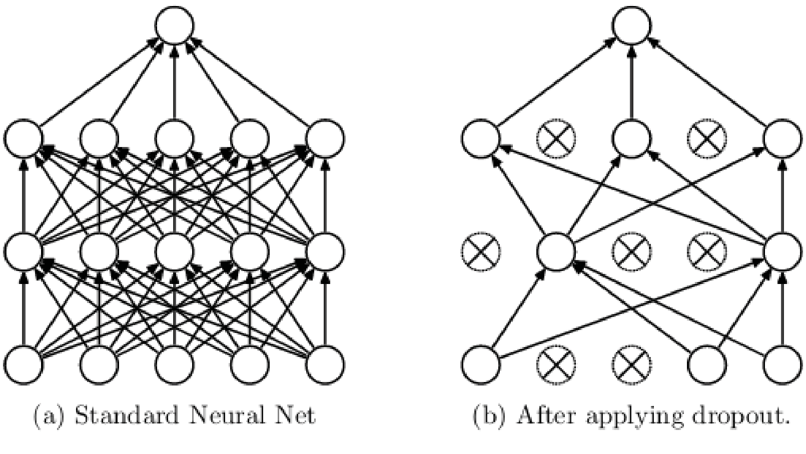

(a) Standard Neural Net
(b) After applying dropout.

Figure 2.14: Ordinary fully connected network (a) and the same network after applying a dropout (b). Fonte: (Srivastava et al., 2014)

2.4.7 Famous Deep learning architectures

Over the last years, some new deep learning architectures have been release, these advanced neural networks became notable for their wide applications, which also had an impact on medical imaging (Maier et al., 2019).

2.4.7.1 U-net

Proposed in (Ronneberger et al., 2015) this network aims to learn how to apply an image-to-image transform, one of its most common uses is to create segmentation masks. Based on a bottleneck structure, it consists of an encoder part, on which the network learns features to classify the input image, followed by a decoder part, which up-convolution layers increase the dimension of the activations from the encoder part back to the original image size (figure 2.15). The network also applies skip connections, which concatenates the outputs of some layers of the encoder part to the decoder part, with the so-called skip connections, aiming to increase the amount of information that the up-convolution layers use to upscale the activations.

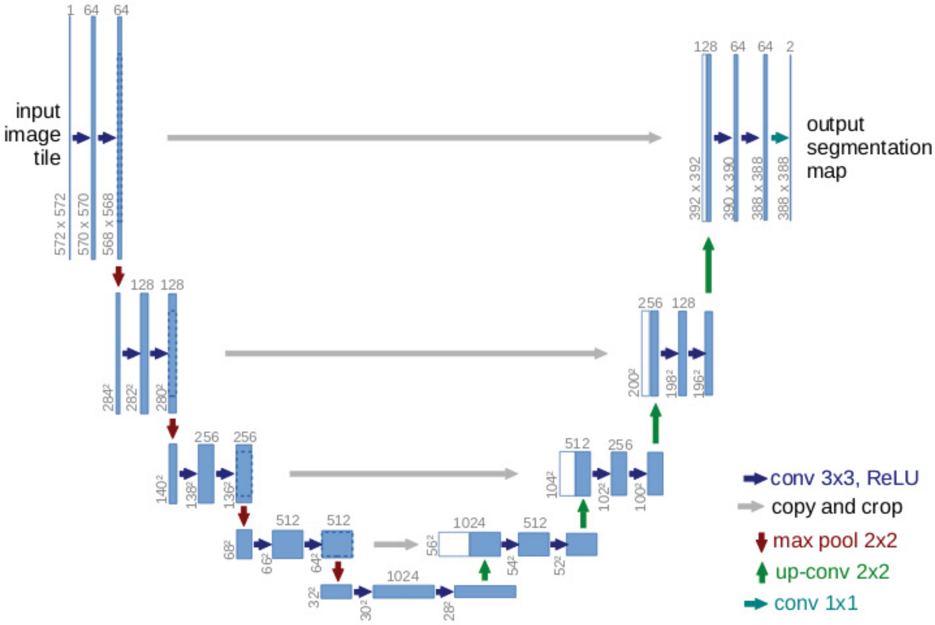


Figure 2.15: Graphical representation of the U-net, the encoder part (left) works as a typical CNN, while the decoder (right) upscales the activations, the gray arrows represent the skip connections. Source: (Ronneberger et al., 2015)

2.4.7.2 Faster R-CNN

Another quite famous segmentation network is the Faster R-CNN, which is an optimization of the R-CNN. This architecture works as an object detector, it extracts different size windows from the input image and classifies then, the original R-CNN paper (Girshick et al., 2014) finds these regions using a CNN and then classifies they using a SVM

The faster R-CNN (Ren et al., 2015) eliminates the selective searching part of the R-CNN algorithm, which was used in the feature map from the output of the CNN. It uses instead another network to predict the selected regions, which has its dimensions reshaped with a region of interest pooling layer, classifying the image within the found region and predict the values within the bounding boxes (figure 2.16).

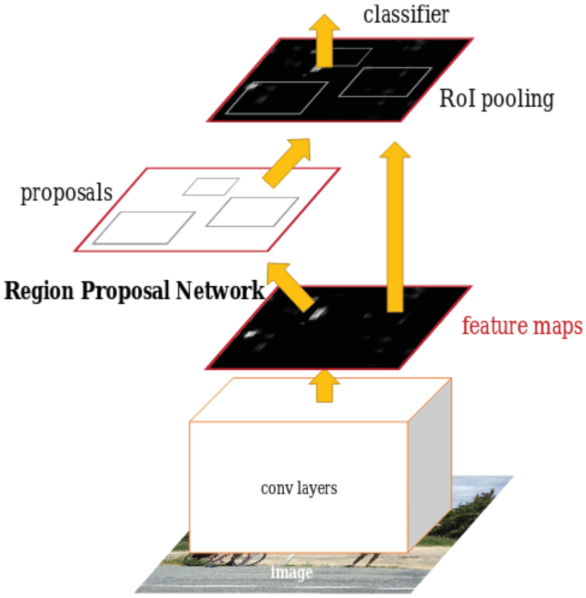


Figure 2.16: The Faster R-CNN applies a convolution layer, the feature map produce goes to another network that gets the regions of interest and classify it. Fonte: (Ren et al., 2015)

2.4.7.3 *Resnet*

The ResNet architecture was designed to aid the training of deeper networks, even with the mitigation methods described before, networks deeper than 30 layers can still suffer from unstable gradients, harming the learning process of the firsts layers. The ResNet (He et al., 2016) presents the residual blocks, the layers in a ResNet can be represented in the form of $f(x) = x + f'(x)$, adding the input of an earlier layer to the output of the last layer, this operation is called a *shortcut connection* 2.17. This connection adds a parallel branch to the network, letting its gradient flow backward and getting to the early ends of the architecture (Maier et al., 2019).

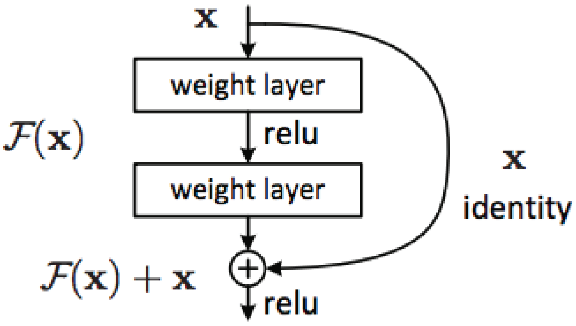


Figure 2.17: Representation of a *shortcut connection* from a ResNet, where $\{(x)$ is the network itself and x its input. Fonte: (He et al., 2016)

A variation of ResNet is the dense networks or DenseNets. Built with dense blocks that different from the ResNets, it concatenates the previous inputs to its feature map instead of adding it (Huang et al., 2017).

2.4.7.4 *Recurrent Neural Networks*

The RNN aim to process sequences of data with dependencies (Mandic and Chambers, 2001), commonly used to process texts and videos. Such networks are modeled recursively, each RNN cell has two inputs, one from its input data x and one from the last cell, which processed the last instance of the sequenced data (e.g. frames in a video), each cell then carries a memory from the last cells forward.

One of its famous variations is the LSTM, in it, a *forget gate* is implemented in each cell since longer sequences might have information at the beginning that is only relevant for instances in the end, the middle portion might not carry the memory from these early cells through the *forget gate* function (Hochreiter and Schmidhuber, 1997).

2.5 CONCLUSION

This chapter presented most of the technologies, techniques, and methods that CAD works related to pulmonary CTA images. Most of the background presented is related to machine and deep learning, since much of the papers related to computer vision, being of medical or regular images, tend to use such algorithms.

3 RELATED WORKS

This chapter reviews the literature related to the use of CAD for classification and segmentation of pulmonary embolisms, for the most part, the data used in those techniques were from CTA exams, but in some cases, CT exams were used.

While the focus of the research was not to find papers that used machine and/or deep learning techniques, most of them, end up using these approaches. Some older were also reviewed to find what worked best in each different technique that was used over time, while the focus was papers at least 5 years old, some early methods can give some good insights over the problem.

One of the earliest methods of PE classification by (Masutani et al., 2002) uses handcrafted features for both the segmentation of PE candidates and FP removal from those selected candidates. To reduce the search space of the algorithm, the pulmonary vessel was segmented from the whole exam, after the segmentation, a feature analysis for possible PEs was performed. Those candidates were grouped based on their position and raw voxel value using a connected component analysis algorithm with 26-neighbor connectivity, PEs were determined based on features collected inside of those groups. The selected characteristics were the volume, effective length, mean local contrast, the location of the candidates, and lung coverage. Based on these features, it was considered as a PE group on which the volume was bigger than 10 mm^3, effective length bigger than 10 mm, mean local contrast between 60 HU and 300 HU and voxel groups with more than 0.75 lung coverage. The database used for the tests was composed of 30 clinical cases (20 positives and 10 healthy), using a 16 mm^3 error threshold, the sensitivity was 100% with 7.7 FP/p (mean FP found in each patient), with 64 mm^3 thresholds it achieved 85% sensitivity with 2.6 FP/p and with a 95 mm^3 threshold and 1.9 FP/p, the sensitivity was not reported for this case. It was found that most of the false positives (92%) were due to soft tissues, such as lymphoid tissue, surrounding the vessel.

The following work by (Liang and Bi, 2007) also tries to segment the pulmonary vessels from the rest of the CTPA to find the PEs, it uses the tobogganing segmentation algorithm to separate the pulmonary vessels from the exam and highlight possible embolisms. The authors used a new implementation of the algorithm, called concentration oriented tobogganing, which looks at each scan row by row to find voxels with a HU value from between -50 HU and 100 HU. Once it finds a voxel, the algorithm “slides” to neighbors voxels and creates a toboggan cluster, it repeats this step to the external boundary voxels of the cluster in the attempt to find other clusters. The dataset used for evaluating the algorithm was composed of 177 exams with 872 PEs, that were divided into train and test sets, with 45 and 132 cases respectively. All the exams were processed with concentration oriented tobogganing, generating 8806 PE candidates with a huge false positive rating, giving an average of 47.5 FP/p in the training set and 40.3 FP/p in the test set. To reduce the number of false positives a multiple instance classification was developed, based on the SVM classifier, on which after at least one PE were classified correctly, a mathematical model was utilized for minimizing the classification errors of the clusters related to the correct one. As the input of this model 116 features were extracted from each candidate based on voxel intensity, 3D shape, and information of neighborhood of the cluster. With the false positive reduction step, false positive ratio was reduced to 4 FP/p in the test set and obtaining an 80% accuracy of the PEs. This work introduced tobogganing as a method of separate the PE from the pulmonary vessel, which was used in future works such as (Tajbakhsh et al., 2015), and it was also one of the first to introduce some early machine learning techniques to reduce the false positive rating.

The work by (Bouma et al., 2009) also follows the same steps as the previous works, a segmentation of the pulmonary vessel, a PE candidate selection, and FP removal, it also applied some well know machine learning algorithms to solve the task. The segmentation of the pulmonary vessels was achieved simply by applying a threshold to the exam, after that, a tracking algorithm was applied, which is described to be able to find possible gaps in the segmented vessel, the tracking algorithm has as input three different segmentations, one of the whole vessel tree, one of the unconnected vessels and the last is the whole search area inside and between the lungs. After the segmentation, for the PE candidate detection step, the authors used intensity based features of the voxels, the voxel raw value, the eigenvalues of the Hessian matrix which were inputted to a black Top-hat transform, with a 4mm dilatation and 2 mm erosion to create a binary mask for the PE candidates. Those candidate voxels were then grouped, from each candidate group a set of features were extracted for classification between positives and false positives. The three features based on the voxel intensity used for identifying candidates, the isophote curvature (with $\sigma = 0.9$ mm), its circularity, the tubularity of peripheral vessels, the distance to parenchyma, distance to the mediastinum, the connectivity of a vessel to major vessels, and its size were all used in the FP removal step. From all those features, a subset of the best features was picked using the technique of l-forward and r-backward with the tree classifier from PRTools, which showed that the stringness (which measures the tubularity near PE candidates) and the distance to parenchyma were the pair of features that best represented the PEs. Many classifiers were also tested using PRTools, for each classifier it was performed a feature selection, the most optimal classifier found was the decision tree. The dataset used for training consisted of 38 positive examples, and 19 for testing, getting a 58%, 63%, and 73% at 4, 4.9, and 15 FP/p respectively.

In (Dehmeshki et al., 2010) work, the same steps as all the previous works were followed: pulmonary artery segmentation, PE candidates selection, and FP removal, this work focus mainly to remove FP due to a poor segmentation of the pulmonary vessel. The first step was the artery segmentation, which separated voxels from the lung tissue to voxels of heart tissue using the threshold technique. A feature map is created using the eigenvalues from the Hessian matrix to remove the connectivity, then it was used a region growing algorithm, with its root to the pulmonary trunk, to increase the difference between the intensity of the heart and pulmonary artery borders. For the embolism detection, five different methods were used, the first three were based on the voxel value, one being based on the voxel raw value, once PEs inside the pulmonary vessel are darker than the background, the second uses the eigenvalues from the Hessian matrix, the third uses black top-hat operator, the other two were based on the shape, the eccentricity of the arterial cross-section and the isophote curvature components. To reduce FP, voxels on the wall of the pulmonary vessel were discarded based on the radius of the artery, soft tissue FP were removed by segmenting the airway system, being that it is commonly found in between the pulmonary vessel and the bronchial wall, and grouped voxels that were too small were considered noise. Using a dataset of 55 CTA exams, with 20 exams used as the training set and 35 for testing, the system achieved a sensitivity of 76%, 94%, and 98% at 2.3, 4.1, and 12.6 FP/p respectively.

The work proposed by (Özkan et al., 2014) aims to further optimize the pulmonary vessel segmentation to reduce the FP ratio, it follows the same steps as other works, pulmonary vessel segmentation, PE candidate selection, and FP removal. The pulmonary vessel segmentation was made in three steps, firstly the segmentation of left and right pulmonary trunks, then lobar and segmental vessel segmentation, followed by subsegmental vessel segmentation, the lungs, and the mediastinum were separated from the CT for the vessel segmentation step. After all the segmentation steps, a threshold of $1 \text{ HU} < \text{voxel} < 150 \text{ HU}$ was applied at the pulmonary vessel and the resulted components were added as PE candidates. For FP removal, very small particles were considered as noise and removed using a 3×3 median filter, if a candidate was located in only a slice it was considered as a FP and thin line shaped candidates were also considered FP. The dataset used was composed of 33 patients with a total of 445 PEs, which was not separated in train or validation, different results were from different volume thresholds, at 6.4 mm^3 the sensitivity was 95% with a 14.4 FP/p, with 16 mm^3 the sensitivity was 61% with a 12.4 FP/p and with 80 mm^3 61% of sensitivity at 8.2 FP/p obtained.

Published in the following year, the work by (Tajbakhsh et al., 2015) was one of the first papers that used convolutional neural networks (CNN) to attempt to solve the task, the paper also introduced a new representation of the selected PE, summarizing the 3D image around an embolus to a 2D image (figure 3.1), making it more compact and easier to apply data augmentation methods. The overall steps are the same as of previous works, the segmentation of the lungs, followed by a selection of PE candidates using the tobogganing algorithm, then the transformation to the proposed image representation is applied to convert the 3D representation of the PE candidate to a 2D image that is fed as input to a CNN, classifying the images into PE or non-PE. To achieve the 2D representation of the image, or as the author calls a vessel-aligned multi-planar image representation, it is estimated the orientation of the vessel on which the PE is contained. To achieved this representation, it was applied a principal component analysis (PCA) in the connected components that were considered a PE candidate (HU $\geq$ 100), with the eigenvectors v_1, v_2, v_3 from the PCA, the interpolation of the volumes along $\{v_1, v_2\}$ or $\{v_1, v_3\}$ gives a longitudinal view of the PE and the interpolation of the volume along $\{v_2, v_3\}$ gives in a cross-sectional view of the PE, each of those views was used as a channel of the final image, that served as input for the CNN (i.e. a $32 \times 32 \times 3$).

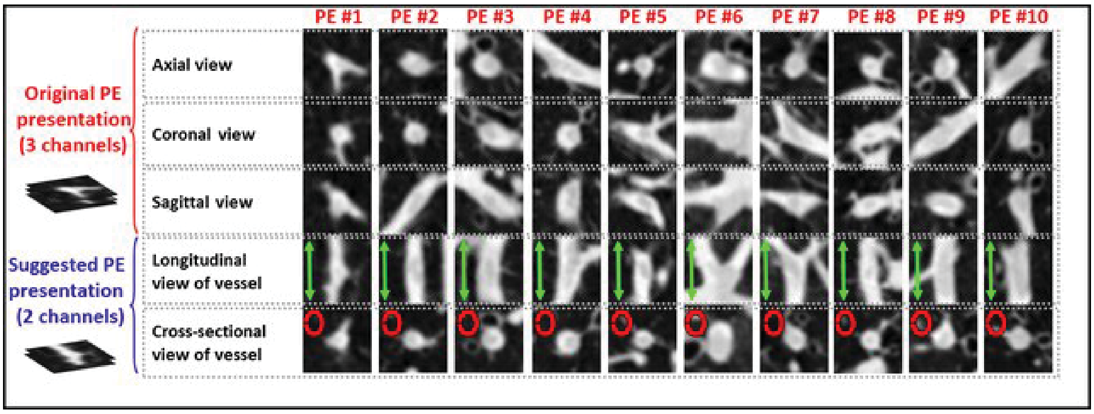


Figure 3.1: Comparison between the traditional 3D view of PEs (axial, coronal and sagittal), and the proposed 2D representation achieved using PCA. Source: (Tajbakhsh et al., 2015)

The CNN architecture was simple and straight forwarded, and not much was discussed in the paper, besides the fact that other more complex CNN models were tested but with no significant difference in the results obtained. As of the experiments, the algorithm was tested in two datasets, a private one containing 121 CTPA exams with 326 emboli, and the dataset from the PE challenge (González et al., 2020) with 20 CTPA exams and 133 emboli. For training, the databases were split into 3 even folds for cross validation, it was also applied a rotation, translation, and scaling data augmentation of the 2D representation of the PE candidates. It achieved an 83.4% sensitivity with a 2.0 FP/p with the private database and 34.6% at 2 FP/p with the challenge database at a 0 mm error threshold.

Another approach to the problem that uses machine learning can be found in (Ozkan et al., 2017), following their work from 2014 using the same database, containing 33 CTPA exams and 450 PEs, and the same steps (lung and lung vessel segmentation, PE candidate extraction and PE detection) but introducing machine learning algorithms to optimize some steps. Firstly the left and right lungs were segmented applying the Otsu threshold method, the space between the segmentation was considered the mediastinum, then with a help of a tracking algorithm (not specified by the authors) and using adaptive threshold the segmental (that reaches the lungs) and subsegmental (does not reach the lungs) vessels were segmented and added to the left and right lung segmentation. That resulted in an uneven segmentation, due to the path of the vessels, the uneven areas of the lungs were found using the first derivative of the x value of the borders, then the area between the two found peaks in the signal were connected using a tracking algorithm, closing the lungs (figure 3.2).

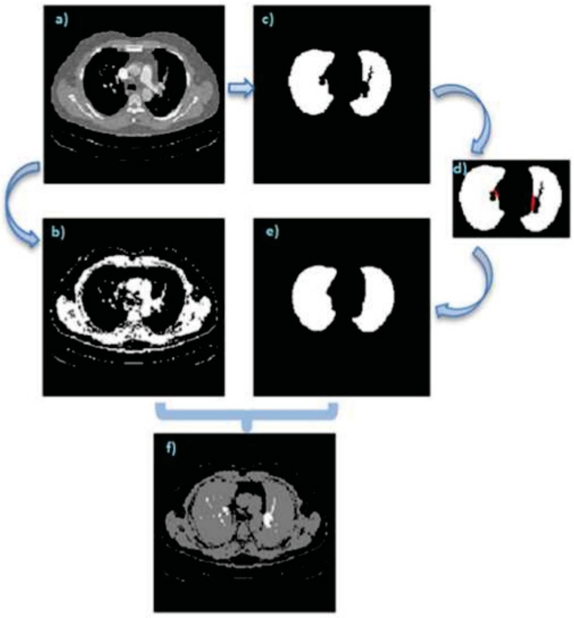


Figure 3.2: The steps of the pulmonary lungs and vessels segmentations. In (a) the original image, (b) threshold with Otsu, (c) segmented lungs without vessels, (d) uneven segmented vessels and lungs, (e) connected segmented lung with vessels and (f) application of the segmental vessels. Source: (Ozkan et al., 2017)

In the PE candidate detection step the authors used a 3D CCL algorithm on a 3D representation of the vessels (called vessel tree), after that it was also used a 3×3 median filter to remove small noise. In the next step, 14 features based on volume, size, and ratios were chosen, then a feature selection method based on the t-score was applied, the authors chose 8 features that had a t-score greater than a threshold, the features were: the maximum value of the candidate length in the 3 dimensions, the largest area in transverse cross section, the volume of each candidate, the diagonals the ratios of voxels and volume in the possible largest area. Three classification algorithms were used, the SVM, MLP, and KNN, the MLP had 2 hidden layers, with 14 and 7 neurons, SCG was used for adjusting the weights with a learning rate of 0.02 and momentum of 0.2. In the KNN, the K value of 30 was chosen by trial and error, and the SVM was used with a Gaussian kernel (RBF). For testing, the data were separated with 10-fold cross validation, using all the extracted features gave a higher accuracy but it had a worse sensitivity and FP/s in general, the results obtained with the selected features using MLP were 98.3% and a 10.2 FP/s, with KNN the sensitivity was 57.3% with 5.9 FP/s and the SVM got 73% sensitivity with 8.2 FP/s. The work had the higher sensitivity at the time, but it did not have the lower false positive rating, this is said to be due to the lower number of PEs in the data, a patient given as an example has 3 PEs but got 15 FP.

After two years, the problem was approached by (Yang et al., 2019), which also followed the same three steps as the others, but it was the first to use deep learning methods in the PE candidate selection, utilizing a model based on Ronneberger's U-net for segmentation. The output of the first network served as input to a second CNN, that aims to reduce false positives by classifying candidates as PE or not. The first network uses 3D convolution layers followed by max-pooling and four residual blocks for its encoder part, and the decoder has two deconvolutional layers, followed by one residual block and two convolutional layers, skip connections are also applied from the last two layers of the encoder part to the first two layers of the decoder part, in the second skip connection it is also concatenated a 3 channel array containing the coordinates of each voxel of the location map inputted in the residual block, forming a 131 channel feature map. To detect the PE candidate after the network output and inspired by the Faster R-CNN, three pre-defined anchor cubes for each voxel were created, each cube has a scale of 10mm, 30mm, or 60mm, which are represented in an array of five values, with its location (Δx, Δy, Δz), size and probability to have a PE inside the cube, explaining the output of the network ($3*5@24\times24\times24$) seen in the upper half of figure 3.3. For training, the input data were split in 96×96×96 overlapping cubes to fit in memory, during testing the cubes were of 208×208×208.

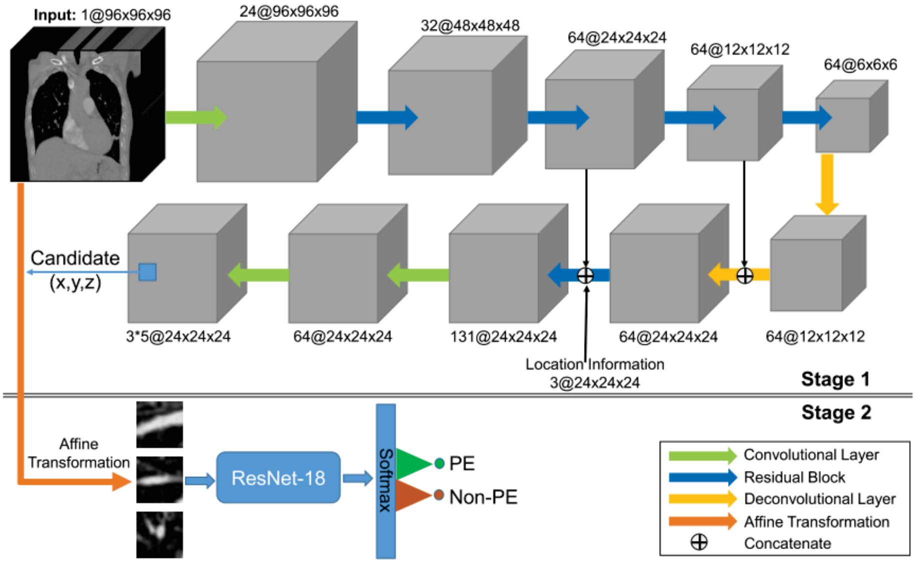


Figure 3.3: Graphic representation of the two-stage network, in the upper-half the PE candidate network and in the lower-half the FP removal network, Source: (Yang et al., 2019)

To avoid overfitting and due to the curse of dimensionality a 2.5D representation was applied, similar to the proposed in (Tajbakhsh et al., 2015), firstly the cube was binarized with a threshold of 70 HU (chosen empirically), then a PCA was applied to get the three eigenvalues that were used in a so-called vessel-alignment transformation, which results in a vessel-aligned. Three different views (coronal, sagittal, and transverse) were extracted from the resulted cube, which was concatenated in a 3 channel image to serve as input for the 2D neural network, based on the ResNet-18 (bottom half of figure 3.3). Due to a high imbalance between false positives generated in the first step, a data augmentation technique was used to increase the size of the real PE base, using scaling, random translation, and rotation. The data used for evaluating the network came from two sources, the dataset from an online PE detection challenge (González et al., 2020) which has 40 exams (20 for training and 20 for testing), and a local database called PE129 that have 99 exams combined with 30 exams from another public dataset (Masoudi et al., 2018). In the CAD-PE challenge dataset, the system got the same value of sensitivity, 75.4%, and

2 FP per scan at 0mm, 2mm, and 5mm localization error, surpassing the winner of the challenge which the data were taken. In the private data sensitivities of 76.3%, 78.9%, and 84.2% at 2 FP per scan at 0mm, 2mm, and 5mm localization error respectively.

In the work by (Liu et al., 2020), the authors utilized a single network (also based on the U-net) for the PE segmentation only, testing different probabilities thresholds for the output of the network, the paper also calculated the clot burden of each PE found, all the cases used in the dataset were of patients with acute PE. A total of 590 exams were used, being 460 positive and 130 negatives, 80% of this dataset (368 positives and 104 negative cases) was used for training and 20% for validation (92 positives and 26 negative samples). For preprocessing the data, a min-max normalization was applied, the images were processed using a window approach, but no clipping was applied, the input of the model were 512×512 images. Not much detail over the network itself was shared, but it seemed to be an implementation of the U-net pretty much like the original one, the sigmoid was used for the output activation function, and different probability thresholds were used to identify a pixel as being part of a PE, the tested thresholds were 0.1, 0.3, 0.6, 0.7 and 0.99, a clot burden calculation was applied together with the segmentation of a PE, using equation (3.1), where Ω is PE regions, p_i its the pixel i of the PEs, w_i represents the width and h_i the height of the p_i and T_i the thickness of the slice where p_i is located. For testing, another dataset was used, containing 288 exams collected in the local hospital (186 positive and 102 negative cases). The results in the test set got no significant statistical difference, getting in the most cases 93.50% sensitivity, 76.50% specificity, and 0.925 AUC, no false positive per scan was informed.

$$V = \sum_{pi \in \Omega} w_i h_i T_i \tag{3.1}$$

The paper by (Rajan et al., 2020) also does a two-stage neural network, but it uses a 2D CNN instead of 3D. It does also follow the same pipeline as the previous works, for the PE candidate generation step a 2D U-net was used and for the FP reduction step a ConvLSTM (figure 3.4). The dataset used in the experiments came from more than 100 medical imaging centers, it contains 1,874 exams with PE and 718 without. With the huge amount of collected data, a different approach to annotation was applied, over 17 radiologists marked the emboli on roughly every slice in a 10mm range, resulting in multiple slices that contained PE without annotation, this method is called “sparse annotation" by the authors. About 172 exams were discarded due to poor quality.

Different from other works with similar approaches, in (Rajan et al., 2020) a 2D CNN based U-net was used, for the input of the network a “slab" of 4 neighbors above and below the current slice (i.e. 9 channel image) was used, the network then outputs a single channel matrix containing the probability of each pixel in the middle of the slab of having a PE, the Dice coefficient was used as a metric. The second neural network had an approach of multiple instance learning (MIL) to the problem to add local context to the learning process, a Conv-LSTM network was used, that takes into consideration for learning the output of its neighbor’s inputs, the features learned by the Conv-LSTM were then used by a multiple instance learning modules, the featuring aggregation methods used for the MIL tested were *max*, *mean* and learnable self-attention functions. Multiple tests were performed in the validation set to find the best hyperparameters of the second network, the best results were achieved with the Conv-LSTM instead of a normal CNN, the Max pooling as the aggregation method and Focal loss instead of binary cross entropy, the network scored an AUC of 0.94 in the validation set and 0.84 in the test set.

In (Huang et al., 2020), the authors used a 3D CNN with pre-trained weights. Unlike the previous work that used 3D neural networks (Yang et al., 2019), the whole CT scan was used

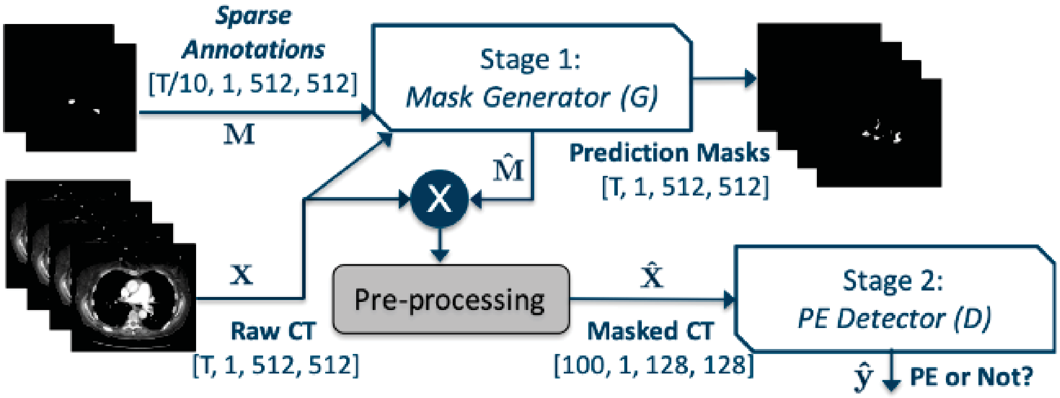


Figure 3.4: Image representation of the network used in the Pi-PE paper, where the raw CT input (X) with its mask (M) are used as input of the generator network (G) that produces a prediction mask ($\hat{M}$), that is multiplied with the raw CTA input to produce a masked CT ($\hat{X}$) used in the second network for FP removal (D) that produces the prediction ($\hat{y}$) saying whether the prediction mask was positive or false. Source: (Rajan et al., 2020)

instead of dividing it into smaller cubes, this change is said by the authors to be significant to the training. This approach also differs from other works by using a single big network, instead of the two-stage, this is connected to the fact that it classifies the CT in a window level as PE positive or negative, instead of classifying PE candidates. The data used consisted of 1797 CT studies (both PE negative and positive) from Stanford Medical Center, which was split into 1461 exams for training, 167 for validation, and 169 for testing. Another dataset was acquired with an additional 200 CT exams from the Intermountain healthcare system. The network itself was composed of multiple custom layers (figure 3.5), the so-called PENet unit (3D convolution followed by group normalization and Leaky ReLU) and Squeeze-and-Excitation (SE) blocks (Average pooling followed by two fully connected layers, with Leaky ReLU and Sigmoid as activation functions respectively), a PENet bottleneck consists of three PENet units followed by a SE block, a skip connection was applied from the input of the first PEN unit to the SE block output, and a PENet encoder is composed of multiple PENet bottleneck concatenated, the ideal depth of the network was found using cross validation on the training set.

The pre-trained weights used came from the Kinects-600 dataset, the group normalization was used instead of batch normalization to save memory, so the resizing of the training images from 512×512 to 224×224, a data augmentation of random crops of the width and height with 192×192, rotation and shifting of the slices (jitter). As for the input of the network, a sliding window method was used of 24 slices at a time, which was also found by using the validation set. The weights were updated using the SGD algorithm with a momentum of 0.9 and learning rate of 0.1, a window was considered positive if at least 4 slices had a PE, being the number of the smallest PE found in the dataset. The network achieved an AUROC of 0.84 from the test set of the Stanford dataset and 0.85 for the whole Intermountain dataset, the results were only compared to other networks famous networks, the ResNet-50, ResNetXt-101, and DenseNet-121, and it outperformed all.

In the same year, the work by (Cano-Espinosa et al., 2020) also uses a single network approach, seeing it as a segmentation rather than a classification problem. The dataset used came from the CAD-PE challenge (González et al., 2020), with a total of 80 scans, 60 being used for training and 20 for testing. Three different networks were proposed for the segmentation task, the first network was also based on a 2D U-net structure, with 4 convolution layers and deconvolution layers, followed by a fully connected layer with a sigmoid as the last activation function, each slice was inputted individually in this network. The second one, similar to the one proposed by

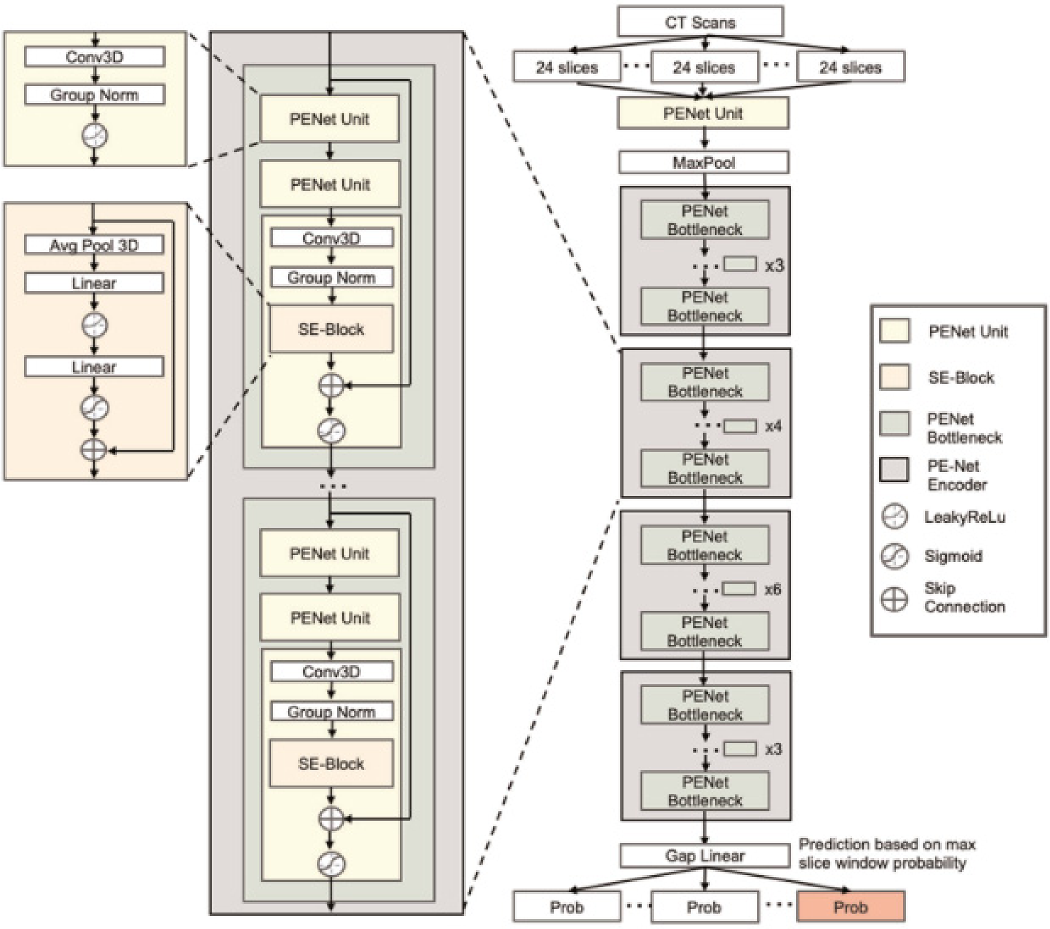


Figure 3.5: Representation of the PENet, with the PENet units, Squeeze-and-Excitation blocks, PENet bottlenecks and encoders. Source: (Huang et al., 2020)

(Rajan et al., 2020), uses a slab of 5 slices as input for a 2D U-net network, this architecture is called 2.5D by the authors. The third one, called 3D network, used the same structure as the 2.5D one, but each inputted slice had the three different planes inputted, sagittal, coronal, and transverse, the exams were sampled to be the same size (512×512×512) for this network, for the prediction of candidates, the three views were merged using the biggest value of each pixel.

During training, 5 of the 60 scans were used for validation, the HU window chosen was between -200 and 500 and then normalized in the range of 0 and 1, each one of the three models was trained for 200 epochs. The network predictions were binarized with a threshold of 0.5, then a closing operation was performed with a kernel size of 5×3×3 to eliminate small noises, lastly, a connected component analysis algorithm was used to connect individual PE candidates. Those candidates were then extracted and, using an inner distance transform, the authors were able to find the pixels that were the furthest away from the perimeter, the coordinates closest to those pixels were associated with a score of its probability of being an embolus. The results obtained in the test set were separated in error thresholds, the 2D, 2.5D, and 3D networks got, respectively, at 0mm threshold sensitivities of 0.49, 0.48 and 0.55 and FP/s of 1.5, 0.65, and 1.00, at 2mm the networks got 0.56, 0.54 and 0.61 of sensitivity and FP/s of 1.45, 0.55 and 1.00 respectively, and lastly at 5 mm error threshold the sensitivities were of 0.59, 0.63 and 0.68 and the FP/s were of 1.35, 0.50 and 0.95. Besides the fact of achieving a low FP/s, the sensitivity of the models was rather low, besides that all the models presented surpassed all the other participants in the old CAD-PE challenge.

In the work by (Long et al., 2021), in it the authors proposed a deep learning model based on the instance segmentation neural network R-CNN (Girshick et al., 2014). The proposed network, named P Mask-RCNN, aims to improve small object detection by enlarging the feature maps and anchor selection from the Region Proposal Network (RPN) module of the Mask R-CNN. Firstly the authors constructed a probability model in order to extract the probability of PE ROIs appearing in each voxel of the image, using a Gaussian mixture model (GMM) to model the probability distribution. Then, based on values of the mixture coefficient, coordinates from the Gaussian distribution were selected to extract anchors for the Mask-RCNN model 3.6. The combination of the upsampled feature maps and anchors selected by GMM are combined and fed to a fast R-CNN model. The data used to test the model came from FUMPE dataset (Masoudi et al., 2018), containing 35 patients, which 29 were used for training, 2 for validation and 4 for testing, rotation and flipping data augmentation methods were applied in order to prevent overfitting. The model obtained a average precision (AP), AP50, AP75 and Dice Score of 41.87%, 81.55%, 41.43% and 0.75, respectively. It had the best results when compared with other state-of-the-art models, such as “vanilla” Mask R-CNN and FCIS.

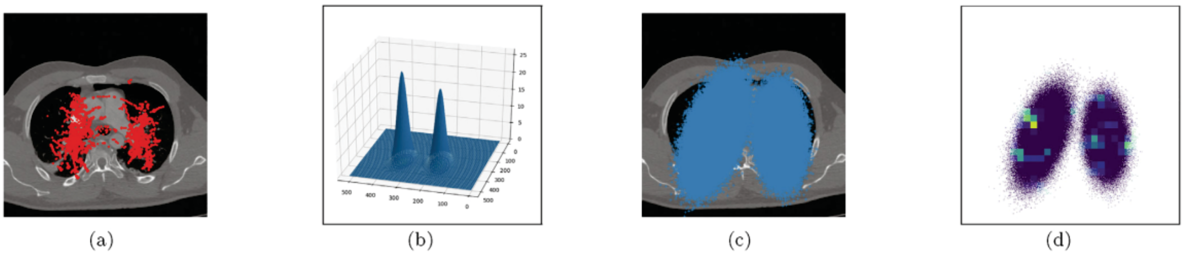


Figure 3.6: (a) PE ROIs from the whole scan. (b) GMM model built from the PE ROIs. (c) Applicaiton of the GMM model to obtain a candidate region. (d) Extraction of the anchors from the feature map. Source: (Long et al., 2021)

Table 3.1: Table summarizing reviewed papers.

Name	Author/Year	Dataset	Description	Results
Computerized detection of pulmonary embolism in spiral CT angiography based on volumetric image analysis (Masutani et al., 2002)	Masutani et al. / 2002	30 exams, 20 positive and 10 negative	Segmentation of the pulmonary vessel, CCL based on HU to find PE candidates, basic thresholding of features to exclude FP (volume, length, mean local contrast, position and lung coverage)	100% sensitivity and 7.7 FP/p at 16mm^3 error tolerance, 85% sensitivity and 2.6 FP/p at 64mm^3 error tolerance
Computer aided detection of pulmonary embolism with tobogganing and multiple instance classification in CT pulmonary angiography (Liang and Bi, 2007)	Liang and Bi / 2007	177 exams, 45 for training and 177 for testing	Segmentation of pulmonary vessel and PE candidate found using an altered version of the tobogganing algorithm, clusters found by the algorithm were classified using a mathematical model based on the SVM, 116 features were used based on voxel intensity, 3D shape, and information from neighbor clusters	80% accuracy and 4 FP/p
Automatic detection of pulmonary embolism in CTA images (Bouma et al., 2009)	Bouma et al. / 2009	57 exams, 38 for training and 19 for testing	Lung and vessel segmentation thresholding HU values, tracking algorithm to find gaps in the segmentation. Voxels intensity and values of the Hessian Matrix are used for creating the segmentation mask from on the gaps found by tracking. Features for FP removal were selected using the l-forward and r-backward methods. The best results were achieved with the decision tree classifier and distance to parenchyma and stringness as features	Sensitivities of 58%, 63% and 73% at 4, 4.9 and 15 FP/p
Fully Automatic Segmentation and Detection of Pulmonary Artery and Embolism in CTA images (Dehmeshki et al., 2010)	Dehmeshki et al. / 2010)	55 exams, 20 for training and 35 for testing	Vessel Segmentation using HU threshold, eigenvalues from Hessian to increase borders intensity. FP removal based on voxel raw value, eigenvalues from Hessian, using black top-hat operator, accentricity of the arterial cross-section and isophote curvature components	Sensitivities of 76%, 94% and 98% at 2.3, 4.1 and 12.6 FP/p
A novel method for pulmonary embolism detection in CTA images (Özkan et al., 2014)	Özkan et al. / 2014	33 exams	Segmentation of the left and right trunks, then lobar and segmental vessel segmentation followed by mediastinum separation from the lungs, threshold technique for PE. FP removal using median filter, number of slices in the component,	At 6.4mm^3, sensitivity of 95% and 14.4 FP/p, at 16mn^3 61% and 12.4 FP/p and at 80mm^3 61% and 8.2 FP/p

Table 3.1: continuation from previous page

Name	Author/Year	Dataset	Description	Results
Computer-aided pulmonary embolism detection using a novel vessel-aligned multi-planar image representation and convolutional neural networks (Tajbakhsh et al., 2015)	Tajbakhsh et al. / 2015	Two datasets, a private one with 121 exams and a public one with 20 exams	Tobogganing algorithm used to find PE candidates, change of representation of each candidate to a 2-channel using eigenvectors which served as input to an FP removal CNN	At the private database an 83.4% sensitivity with 2.0 FP/p, with the public database a sensitivity of 34.6% at 2 FP/p at a 0 mm error threshold
Automatic Detection of Pulmonary Embolism in CTA Images Using Machine Learning (Ozkan et al., 2017)	Ozkan et al. / 2017	33 exams	Otsu thresholding and a tracking algorithm to reduce search area and find PE candidates, SVM, MLP, and KNN were used to classify candidates, 10-fold cross validation for training	The MLP, KNN and SVM got respectively 98.3%, 57.3% and 73% sensitivities and 10.2, 5.7 and 8.2 FP/p
A Two-Stage Convolutional Neural Network for Pulmonary Embolism Detection From CTPA Images (Yang et al., 2019)	Yang et al. / 2019	Two datasets, a public with 20 exams and a second dataset combining 99 private exams with 30 from another public dataset	3D CNN network based on the U-net and uses residual blocks, produces a set of blocks for each voxel containing its probability of having a PE, those blocks were converted to a vessel-aligned representation and 3 views (sagittal, coronal, and axial) were extracted to serve as input to a Resnet-18 for FP removal	In the public dataset 75.4% sensitivity and 2 FP/p at 0mm, 2mm and 5mm error threshold, in the private set sensitivities were of 76.3%, 78.9% and 84.2% all with 2 FP/p at 0mm, 2mm and 5mm error
Evaluation of acute pulmonary embolism and clot burden on CTPA with deep learning (Liu et al., 2020)	Liu et al. / 2020	590 exams, 460 positives, and 130 negatives, 80% were used for training	2D U-net that segmented clots from the CT only, different errors thresholds were tested and a clot burden value were attributed to each PE found	No statistical difference between thresholds, with 93.50% sensitivity, 76.50% specificity, and 0.925% AUC, no FP were informed
Pi-PE: A Pipeline for Pulmonary Embolism Detection using Sparsely Annotated 3D CT Images (Rajan et al., 2020)	Rajan et al. / 2020	2592 exams, 1874 positives exams and 718 negative	Two step deep learning approach, first segmentation network based on U-net with 2D convolutions with different channels being the 4 neighbor slices from above and below the target slice (i.e. 9 channel image) with dice loss. The second network was a ConvLSTM as a feature extractor that was applied to a MIL algorithm. It got the best results with a max pooling aggregation method and Focal loss	AUC of 0.94 in validation set and 0.84 in test set

Table 3.1: continuation from previous page

Name	Author/Year	Dataset	Description	Results
PENet—a scalable deep-learning model for automated diagnosis of pulmonary embolism using volumetric CT imaging (Huang et al., 2020)	Huang et al. / 2020	1797 exams, 1461 used in training, 167 for validation and 169 for testing and a second test dataset with 200 exams	3D convolutional network with squeeze-and-excitation blocks, with skip connections from the input of the 3D convolution and the output of the SE blocks. Different network depths were tested. It used pre-trained weights from the Kinects-600 dataset and data augmentation.	AUROC 0.84 in the test set and 0.85 in the second test set
Computer Aided Detection of Pulmonary Embolism Using Multi-Slice Multi-Axial Segmentation (Cano-Espinosa et al., 2020)	Cano-Espinosa et al. / 2020	80 exams, 60 for training and 20 for testing	Three different networks were tested, a 2D U-net with a depth of 4. A 2D U-net with neighbor slices as input (similar to Rajan et al. work). The third was a 3D U-net with three different views (sagittal, coronal, and transverse). The best results were achieved with the 3D network.	Sensitivities of 55%, 61% and 68% with FP/p of 1, 1 and 0.95 at 0, 2 and 5 mm error threshold respectively
Probability-based Mask R-CNN for pulmonary embolism detection (Long et al., 2021)	Long et al. / 2021	35 exams from FUMPE dataset	Model based on Mask R-CNN, using upsampling of feature maps in combination with an anchor extraction technique using Gaussian Mixture Models and a Fast R-CNN for prediction, the model optimizes small object detection.	average precision (AP), AP50, AP75 of 41.87%, 81.55%, 41.43%, respectively and a Dice Score 0.75

4 PROPOSAL

The objective of this work is to develop a method capable of classifying correctly and show PE from a CTA image scan. The proposed method aims to further optimize the current networks from the state of the art. It follows the same main steps: pulmonary segmentation and false positives (FP) reduction. Besides that, this work also developed a brand new dataset alongside the radiology team of the *Hospital das Clínicas* (HC) from the UFPR.

The pipeline for detection of EPs (Figure 4.1) follows two main parts: in the first part, a U-net model will be trained to be able to select PE candidates, segmenting them from the whole exam volume. For the input of this network a neighborhood of five slices served as input, each slice representing a different channel in a image. This approach was inspired mainly by (Cano-Espinosa et al., 2020) work, which achieved the better results using a 3D representation of the input, and in the work by (Rajan et al., 2020), which uses so-called "slabs" of regions in the volume. Since a EP occupies more than one slice, the spatial information helps the training process of the network.

For the second part a machine learning pipeline was developed for extracting the features and classify candidates of EP. Right after the segmentation each EP candidate was separated, then a feature array was extracted with data about the candidate volume and it's spatial location. Then those feature arrays were trained with different machine learning algorithms classifying the candidates into PE and non-PE, effectively reducing false positives.

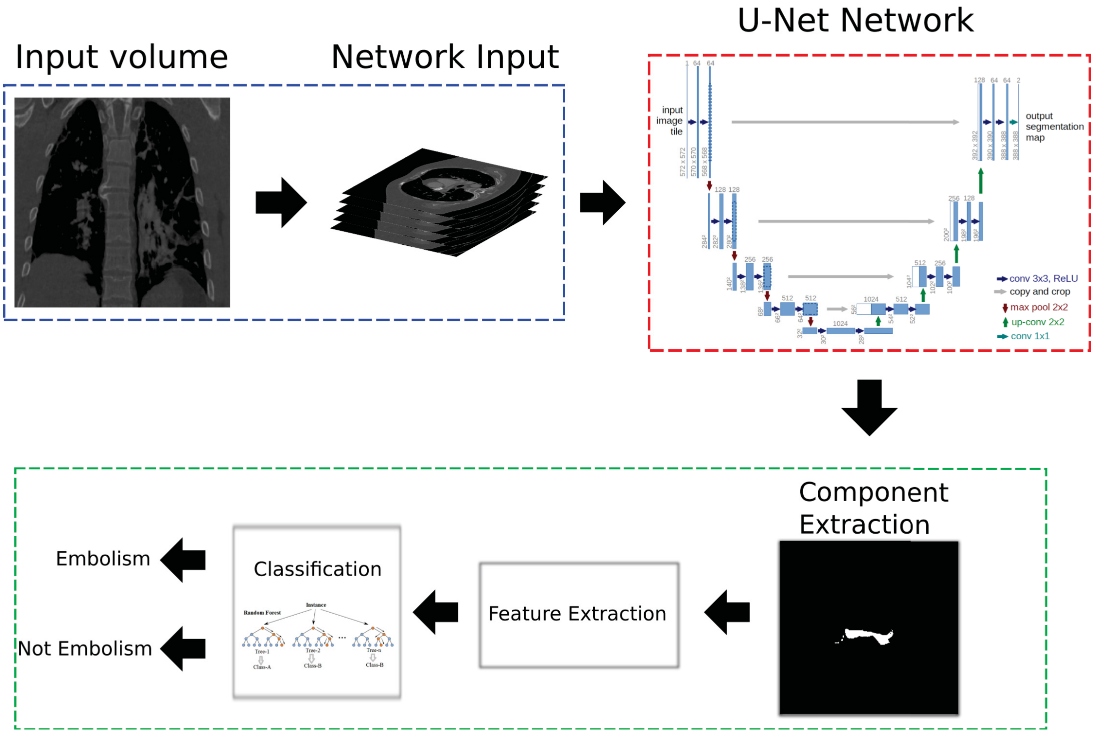


Figure 4.1: Step-by-step representation of the proposed method, in blue the steps of preprocessing and slice extraction for the input of the network, in red the representation of the U-net network adapted from (Ronneberger et al., 2015), in green the steps extraction of PE candidate, feature extraction and classification. Source: The author.

4.1 DATA

The data used for training and evaluation of the built model come from three different datasets (FUMPE, CAD-PE and HC), which all are composed of CTA images with semantic labeling of the PE. Table 4.1 summarize all three datasets, showing PE counts at intervals based on their sizes in mm^3.

4.1.1 First dataset: "FUMPE"

The first dataset available comes from (Masoudi et al., 2018), which is composed of 35 CTA, with two exams not having any PE. The segmentation masks were produced by a radiologist with 5 years of experience in the field and

Table 4.1: Count of PEs of each database within ranges of volumes in mm^3

Intervals (mm^3)	FUMPE	CAD-PE	HC
$0,1 \leq n < 0,5$	0	74	35
$0,5 \leq n < 1$	1	117	23
$1 \leq n < 2$	0	93	26
$2 \leq n < 9$	3	96	69
$9 \leq n < 38$	8	68	51
$38 \leq n < 168$	13	157	71
$168 \leq n < 740$	31	155	74
$740 \leq n < 3270$	32	80	25
$3270 \leq n < 14420$	19	54	29
$14420 \leq n \leq 63650$	11	27	5

reviewed by another with 18 years of experience. The dataset has a total of 8792 slices, and 3438 PE regions of interest, with the height and width of the *voxels* ranging between 0.52 and 0.78 mm. Most PEs (67%) are found in peripheral arteries. In addition to the markings, the authors also provided metadata, such as the proportion of the size of the right ventricle and left ventricle of the heart, whether there was reflux in the inferior *vena cava*, whether the interventricular septum is straight, the pulmonary artery diameter and the value of *Qanadli score*, which is calculated based on the position of the thrombus within the lung. The dataset is also used in combination with a private one in the work by (Yang et al., 2019).

4.1.2 Challenge dataset: "CAD-PE"

The second database, called "CAD-PE" (González et al., 2020), is composed of 91 CT angiography exams and a total of 24624 slices with *voxels* sizes ranging between 0.52 and 0.92 mm. The first 40 exams were segmented by a board of radiologists with more than 10 years of experience while the other 51 exams were segmented by a board member. The marking was done with the aid of a semi-automated method, in which the markings performed by the radiologists were of the area of interest (i.e. a rectangle in the region of the PE), and with this area of interest a semi-supervised method was used for to generate the semantic segmentation masks, each segmentation performed was then averaged to produce the final mask. Many papers from the chapter 3 utilize data from the CAD-PE challenge, which is available on their website but it no longer has the segmentation masks available for downloading, but with a new recent wave of papers for PE detection, the dataset was republished with new labeled data.

4.1.3 Our dataset: "HC"

The dataset called "HC" was develop alongside the algorithm of this work. It has 40 exams, 20 produced by the GE Revolution 512 tomographer and another 20 by the Toshiba Aquilion 64. The dataset has 14463 slices, with sizes of the *voxels* range from 0.57 and 1.0 mm. All exams were evaluated by a thoracic radiologist with 32 years of experience regarding the presence of PE and its location. After this review, the masks were created by a physician in the third year of the radiology residency. No semi-automated method was used to create the segmentation masks.

The project regarding the dataset creation was accepted by the ethics committee from HC and it's properly registered in the *Plataforma Brasil*. All of the exams selected were diagnosed with PE in the HC hospital from the time period of 05/11/2018 to 05/02/2019, the exams were captured with a Toshiba Aquilion 64 topographer, PE diagnosed exams were also collected form the DAPI clinic by a GE Revolution topographer in the time period of 09/11/2018 to 20/11/2019. From the final 40 labeled exams, 20 will come from each tomographer. The development of the dataset has been approved by the ethical committee from HC with the CAAE: 07724919.0.0000.0096.

The annotation of the dataset was made using the ITK-SNAP software, producing binary mask with same dimension as the CTA exam (Figure 4.2).

Additional metadata was also created for the dataset, for all CTA scans evaluated the largest artery involved, inferior vena cava reflux, interventricular septum flattening or paradoxical bowing, pulmonary artery trunk diameter, transverse diameters of the right ventricle (RV) and left ventricle (LV), measured between the endocardial surfaces in the largest place perpendicular to the longitudinal axis, and RV/LV relationship were listed (Tables 4.2 and 4.3). Those informations can serve as indirect signs of the presence of PE, they can be observed in CTA scans as pulmonary artery dilatation, right ventricle enlargement (increasing the relation of right and left ventricles

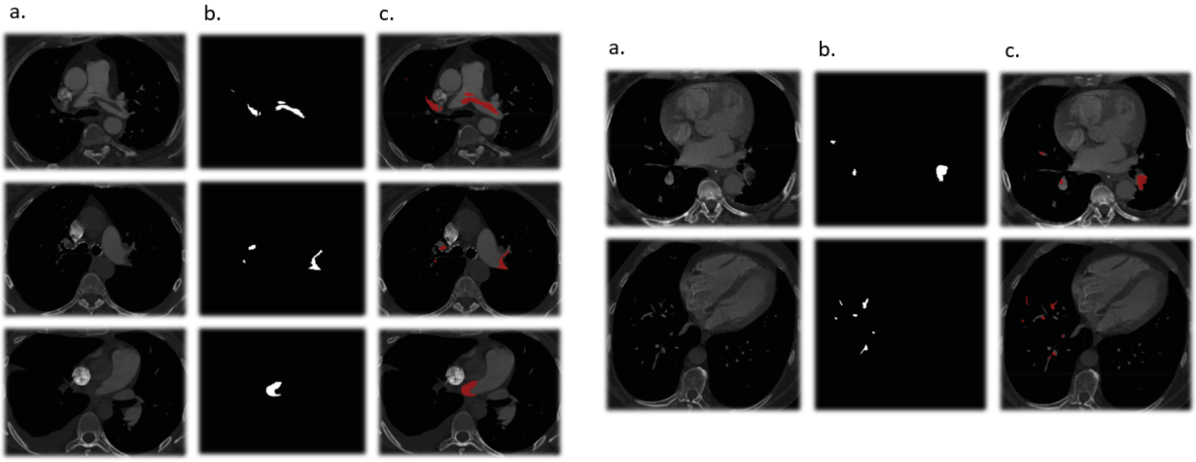


Figure 4.2: Pixel level annotation. In column (a) CTA image of PE in different anatomical locations. In column (b) pixel level annotation showing in white all pixels of the PE of the corresponding CTA image in column (a). In column (c) superposition of the corresponding image of columns (a) and (b), with the thrombus in red.

diameters), inferior vena cava contrast reflux and abnormal position of the interventricular septum, that can be flattened or even paradoxical bowed towards the left ventricle (Figure

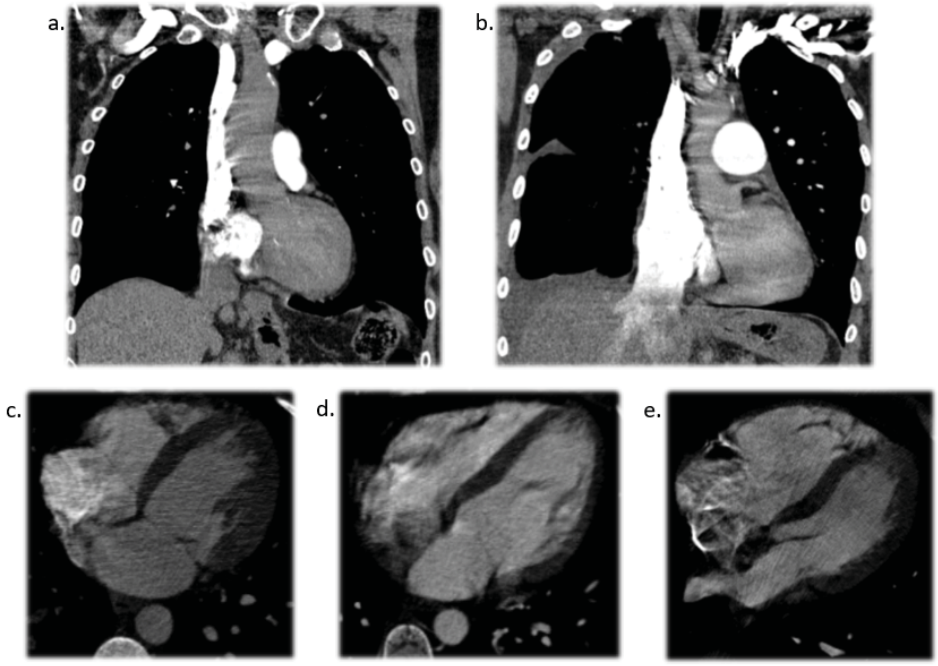


Figure 4.3: CTA features related to right heart strain and pulmonary artery hypertension. In (a) an example without IVC reflux. In (b) with IV reflux. In (c) showing normal position of interventricular septum. In (d) showing flattened interventricular septum. In (e) showing paradoxical interventricular septal bowing

4.2 PROPOSED METHOD

The technique proposed in this work is composed of two main steps: the location and segmentation of PEs candidate and the reduction of false positives of these candidate groups (Figure 4.1).

Table 4.2: Data from exams obtained with GE Revolution 512. PAD: pulmonary artery diameter. RV: right ventricle diameter. LV: left ventricle diameter. RV/LV: relation of right and left ventricles diameters. IVC Reflux: inferior vena cava reflux of contrast media. IV Septum: interventricular septum position.

Case	Genger	Age	PAD(mm)	RV(mm)	LV(mm)	RV/LV	IVC Reflux	IV Septum	The largest affected vessel
1	M	59	27	52	57	0,91	Absent	Normal	Main Artery (unilateral)
2	F	72	32	60	40	1,5	Absent	Flattening	Main Artery (unilateral)
3	M	71	36	38	32*	1,19	Present	Normal*	Lobar
4	M	62	28	33	34	0.97	Absent	Flattening	Trunk bifurcation
5	M	73	30	32	42*	0,76	Absent	Normal	Segmental
6	M	82	40	47	30	1,57	Present	Flattening	Trunk bifurcation
7	F	21	32	39	47	0,83	Absent	Normal	Segmental
8	F	82	29	32	42	0,76	Absent	Normal	Segmental
9	M	50	28	22	40	0,55	Present	Normal	Lobar
10	F	57	26	55	38	1,45	Absent	Paradoxical bowing	Sub-segmental
11	M	31	27	25	51	0,49	Absent	Normal	Segmental
12	F	36	22	30	35	0,86	Absent	Normal	Lobar
13	F	86	26	35	36	0,97	Present	Normal	Sub-segmental
14	F	57	38	50	48	1,04	Present	Flattening	Main Artery (unilateral)
15	F	32	27	44	42	1,05	Absent	Flattening	Segmental
16	M	38	30	46	50	0,92	Absent	Normal	Lobar
17	F	81	31	50	50	1,00	Present	Normal	Lobar
18	M	54	28	44	50	0,88	Absent	Normal	Lobar
19	M	41	29	40	49	0,82	Absent	Normal	Trunk bifurcation
20	M	54	29	38	64	0,59	Present	Normal	Segmental

Table 4.3: Data from exams obtained with Toshiba Aquilion 64. PAD: pulmonary artery diameter. RV: right ventricle diameter. LV: left ventricle diameter. RV/LV: relation of right and left ventricles diameters. IVC Reflux: inferior vena cava reflux of contrast media. IV Septum: interventricular septum position.

Case	Genger	Age	PAD(mm)	RV(mm)	LV(mm)	RV/LV	IVC Reflux	IV Septum	The largest affected vessel
1	F	63	24	43	57	0,75	Present	Normal	Lobar
2	F	11	20	37	40	0,92	Absent	Flattening	Trunk bifurcation
3	F	77	29	45	35	1,29	Absent	Flattening	Main pulmonary artery
4	F	38	34	52	29	1,79	Present	Paradoxical bowing	Main pulmonary artery
5	F	68	29	41	43	0,95	Absent	Flattening	Trunk bifurcation
6	F	54	28	39	54	0,72	Present	Flattening	Main pulmonary artery
7	F	58	35	33	35	0,94	Present	Normal	Lobar
8	F	64	31	37	48	0,77	Absent	Flattening	Segmental
9	M	29	28	23	31	0,74	Absent	Flattening	Segmental
10	M	68	26	52	35	1,49	Present	Paradoxical bowing	Main pulmonary artery
11	F	81	27	29	42	0,69	Present	Normal	Lobar
12	F	84	31	52	41	1,27	Present	Normal	Trunk bifurcation
13	F	41	24	32	38	0,84	Absent	Normal	Segmental
14	F	48	34	43	51	0,84	Present	Normal	Segmental
15	F	45	20	40	45	0,89	Absent	Normal	Main pulmonar artery
16	F	75	31	33	46	0,72	Present	Normal	Sub-segmental
17	F	59	31	42	36	1,17	Present	Flattening	Trunk bifurcation
18	F	42	24	42	45	0,93	Absent	Normal	Segmental
19	F	26	25	35	50	0,7	Absent	Normal	Lobar
20	F	50	25	34	52	0,66	Absent	Normal	Main pulmonar artery

4.2.1 Preprocessing

To assist the training of the proposed method, the same preprocessing methods were applied in the three datasets. First, the slice width size was changed to 1 mm for all patients in order to standardize the entries. Then, the grayscale values of the images were normalized using the Hounsfield (UH) Units scale windowing, with the window size of 350 HU and the center of the window of 150 HU, transforming the values between $[200, 500]$, then the values were normalized between 0 and 1 in order to improve the learning and convergence rate of the network. For each image, a central window in the size 364×364 was also cut, in order to remove regions around the rib cage from the images.

A data augmentation step was performed on the training base, with the objective of increasing the amount of images for a better network training, a horizontal and vertical flips with a probability of 50% and a Gaussian noise with 20% probability were applied for each image.

To feed the input to the U-net network, the total image volume was split into 5-slice overlapping windows, with each slice being a channel of the image. The central channel (i.e. channel 3) was the aim for the network segmentation.

4.2.2 U-net Model

As stated before, a U-net model was used for the preliminary results obtained. Besides being based on Ronnenberg's original U-net it does not follow the same amount of layers and hyperparameters. The model is divided into two parts, a encoder and a decoder. The encoder part is built using basic blocks, each basic block is composed of two convolutions, each followed by a batch normalization and a ReLU activation, the whole encoder is composed of five basic blocks, with a number of channels of each convolution of 64, 64, 128, 256, and 512. Then five decoder blocks are applied, each decoder block is composed of two deconvolution layers, each followed by ReLU only, the amount of channels of the convolutions are the same as in the encoder part, but decreasing. The input of each decoder block is the output of the last block concatenated with the output of the corresponding encoder block (i.e. the one with the same amount of channels). At the end of the last decoder block, a last 2D convolution is applied with a single channel as output and the Sigmoid function is applied, this is the final output of the network, having the same shape as the input image segmentation mask.

Different from the original U-net network, the proposed network receives as input an image of size $364 \times 364 \times 5$ and generates an output 364×364. The five input channels are the five neighboring slices in the CT volume, therefore the 2D convolutions are using dimensional data as the channels.

For the training step of the model, the optimizer of choice was Adam with an initial learning rate of 1e-4, which was reduced by 1e-2 every time the loss reached a plateau. The loss function used was Dice and the IoU with a threshold of 0.5 of the output values. The size of batches per iteration of training chosen was 16 images, and each image having 5 channels of neighboring slices.

4.2.3 Candidates Extraction

To find the PEs candidate , the output values of the network were rounded to 0 (values <0.5) and 1 (values ≥ 0.5), then a connected components algorithm (CCL) was applied, each component found by the CCL was considered a PE candidate. For an initial reduction of false positives, a closing operation was performed with a kernel of $5 \times 3 \times 3$, aiming to remove small particles and noise, candidates found in only one slice of the exam were also removed.

Each candidate from the output of the network was intersected with the ground truth mask, if there was any intersection (candidate mask

$$\times$$

ground truth $\neq 0$) the candidate was considered a true positive, components without any intersection were considered as false positives.

4.2.4 Feature Extraction

After the candidates' extraction step, the feature extraction of each candidate is carried out. The extraction was made to each candidate produced on the 3D CCL algorithm.

First, for each candidate, the 3D coordinates of the center of the mass (CM) were defined and selected as features. Then, the HU scale value and the output from the U-net network of these central coordinates were selected as features, both raw values. Finally, the total volume in mm^3 of each candidate was calculated, along with the larger distance along the three axis (x,y,z), which was obtained by setting the value of the median x, y and z and finding the maximum distances in the other axis. In the end, nine features (CM coordinates (x,y,z), CM HU value, CM output U-net value, volume, larger distance in axis x, y, z) were used in the classification step which were normalized using a standard scaler, leaving the values between 0 and 1. The selected features were chosen based on other works from that used machine learning for PE classification, such as (Cano-Espinosa et al., 2020), (Özkan et al., 2014), (Bouma et al., 2009), and (Liang and Bi, 2007)

4.2.5 Classification

To make the classification between PE and non-PE candidates, a combination of classifiers was trained and tested. In all tests regarding the FP classification step, the same hold-out division of the data-sets was used (train, validation, and test).

The classifiers KNN, SVM (with linear and Gaussian), Random Forest, Naive Bayes and a Multi Layer Perceptron; With some early testing using the train and validation set, some algorithms that performed poorly were discarded, keeping only the KNN, SVM with Gaussian kernel and Random Forest. The next step was to find the optimal parameters and hyper-parameters for each classifier. For the KNN classifier a range of neighbors were tested using the first 100 odd numbers beginning from 3. For the SVM classifier a grid search was performed, varying the C $(0.001, 0.01, 0.1, 1, 10)$ and gamma value $(0.001, 0.01, 0.1, 1)$, for the Random Forest, another grid search was done, using the range of parameters for the maximum depth (200 to 2000 increasing 200 each step) and number of estimators (100 to 500 increasing 40 each step). For both grid searches, the optimal parameters were found by means with 5 fold cross validation. All training of the classification step was done with a combination of all three data-sets after being segmented by the U-net network.

4.2.6 Metrics

For the evaluation of the segmentation step of the model (U-net), the Dice score, the IoU and the Sensitivity were measured. The Dice score tells the correctness of the segmentation mask when compared to the original mask, it was also used as the loss function for the network instead of the IoU since the Dice Coefficient is differentiable. The IoU was measured to clearly see the detection quality of the network, and the Sensitivity was measured in order to better compare the results with other SOTA works, the Sensitivity is a common metric in such works with a low amount of training data available.

5 RESULTS

All results were achieved using the same base pipeline, with a U-net model for segmentation, and a machine learning algorithm for classification. But two different test approaches were made, one in which the amount of data was fixed and the hyperparameters of the model was adapted for the dataset, and a second approach with a fixed model but varying the data distribution.

5.1 SEGMENTATION EVALUATION

5.1.1 Model-centric approach

The U-net segmentation network and all the training process was developed with PyTorch library ((Paszke et al., 2019)). Each one of the three datasets were tested separately using a hold-out division of 70% for training, 10% for validation and 20% for testing. Each model trained for 100 epochs, the epoch that achieved the highest IoU score in the validation set was saved for testing.

For training the optimizer used was Adam with an initial learning rate of $1e4$, which was reduced by $1e-2$ every time the loss reached plateau. The chosen loss for training was the Dice, but the IoU was also observed with a threshold of 0.5 of the output values. The size of batches per iteration of training chosen was 16 images, each image having 5 channels of neighboring slices.

A high number of other hyperparameters were also tested in each dataset, the HU window scale, the optmizer, the initial learning rate and a method (or absence) for reducing the learning rate while training, the amount of batches for training, the preprocessing steps (normalization and scaling). But only the parameters cited gave a proper result.

In the ((Masoudi et al., 2018)) dataset, the network got a 0.68 mean IoU, 0.72 Dice score and Sensitivity of 0.82 in the test set. The changes in the loss can be seen in Figure 5.1. It's observable that the validation loss quickly decreases as values below the training loss, this can be due to the amount of data used since the plot shows the mean loss during an epoch and there are more patients in the training set than in the validation set. While in the test set of the ((González et al., 2020)) achieved a mean IoU of 0.79, 0.81 mean Dice score and 0.89 Sensitivity, the training and validation losses can be observed in the Figure 5.1, showing that with more data the curves behave more conventionally. In the HC dataset, a mean IoU of 0.75, Dice of 0.78 and Sensitivity 0.78 were achieved (Table 5.1).

Table 5.1: Segmentation results the model-centric approach separately.

Dataset	Dice	IoU	Sensitivity
FUMPE	0.72	0.68	0.82
CAD-PE	0.81	0.79	0.89
HC	0.78	0.75	0.78

5.1.2 Data-centric approach

Another experimental approach was using different combinations of the selected datasets for training and testing the U-net. The network used was the same from the model-centric tests, varying just some hyperparameters that were found to be better for training with a higher amount of data.

Two different combinations of the datasets were tested, one with a combination of the CAD dataset and the FUMPE dataset for training and validation while testing in the HC dataset, and another training with the CAD and HC dataset and testing in the FUMPE dataset.

For training the model, the SGD optimizer found better results, with a initial learning rate of $1e4$ and no reduction during training. The size of the batch was also changed from the model-centric approach to 16, and a different HU window scale was used, while the model-centric approach used a WC of 150 and WL of 350, the tests with the data-centric approach found a better result with a WC of 150 and a WL of 700.

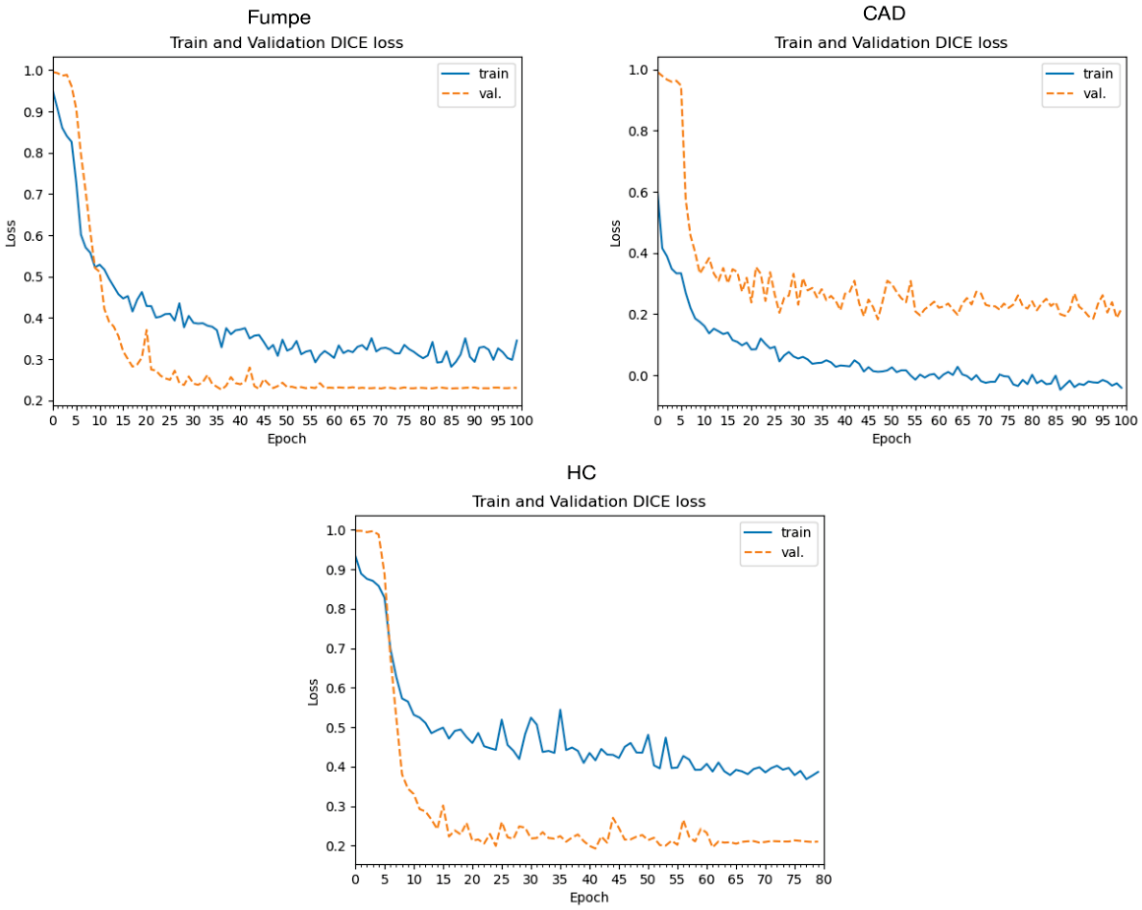


Figure 5.1: Top left: Dice loss over the epochs during the U-net training step for the (Masoudi et al., 2018) dataset. **Top right**: Dice loss over the epochs during the U-net training step for the (González et al., 2020). **Bottom:** Dice loss during training of U-net model in the HC dataset

The results achieved with each combination can be seen in Table 5.2. With no difference in the sensitivity value of each test, but a slight better result was found with a combination of the CAD and FUMPE dataset and testing in the HC dataset.

Table 5.2: Results with training the model with the two whole datasets and testing in the.

Dataset	Dice	IoU	Sensitivity
CAD+FUMPE (test HC)	0.78	0.76	0.86
CAD+HC (test FUMPE)	0.74	0.70	0.86

5.1.3 Model comparison

The results achieved in both approach's can be compared to some results of works found. The first comparison was with the paper by (Long et al., 2021), in which the authors found PEs by using a model based om the Mask R-CNN, the metrics used to evaluate the model was the Average precision with different IoU thresholds, this work was chosen for comparison because it used a single dataset for training and evaluation, the FUMPE dataset. The results can be seen in table 5.3.

The second work which the results were compared is from (Cano-Espinosa et al., 2020), which used only the CAD dataset. The authors also used a U-net based model for segmentation, reporting the sensibility achieved. The results can be seen in table 5.4.

It's observable that the model had a worse result when comparing APs to the work from (Long et al., 2021) and, it achieved a better sensibility when compared to (Cano-Espinosa et al., 2020). The amount of data used in each experiment could explain the results.The FUMPE data-set has a PE distribution less uniform than CAD-PE. This difference could explain the better results achieved using the CAD-PE data-set.

Table 5.3: Segmentation of proposed U-net vs P Mask RCNN.

Dataset	AP50	AP75	AP
P MASK RCNN	81.55	41.87	41.43
PROPOSED	0.1875	0.06	0.06

Table 5.4: Segmentation of proposed U-net vs P Mask RCNN.

Work	Sensibility
Cano Espinosa et al.	0.68
PROPOSED	0.82

5.2 FALSE POSITIVE REDUCTION EVALUATION

Each candidate from the output of the network was intersected with the ground truth mask. If there was an intersection (candidate mask × ground truth ≠ 0) the candidate was considered a true positive. The components without any were marked as false positives. These components were fed to the false positive reduction phase, applying all stages of pre-processing and feature extraction before the classification training. The classification obtained an accuracy of 83% in the CAD-PE data-set, 74% in FUMPE, and 79% HC 5.5. The same training, validation, and testing distribution of each data-set were used for the false positive reduction step.

Table 5.5: Accuracy from classification algorithms using each dataset.

	RandomForest	SVM	Naïve Bayes	KNN	MLP
FUMPE	0.77	0.71	0.63	0.73	0.72
CAD	0.84	0.82	0.74	0.81	0.81
HC	0.73	0.55	0.40	0.59	0.47

An analysis was made by the size of the components, comparing them with the ground truth (Figure 5.2). It was observed that the network has difficulties to find small components, with high false positive rates. It is also observed that in some cases the number of predicted true positives of a size interval is greater than the amount of real components, this is due to the fact that the method considers the components as true positives if there is any intersection of the candidate with any real component, so there are cases where the PEs were found but were not marked completely, generating a prediction of lower volume. The proposed method also seemed to be unable to detect smaller PEs, even in datasets with an higher representation of them (CAD-PE and Private), the model was not capable of detect most of smaller PEs. It was also observed that the Private dataset seemed to better fit in the segmentation step

The dataset used presented itself as another factor that directly interferes in the amount of false positives found, it affects also in which volume ranges these false positives are found. It is believed that this behavior is due to the amount of existing elements in each size range. The Figure 5.3 shows the effectiveness of the model in reducing the false positive components in all the three datasets, especially the components of smaller volume, but it makes some mistakes while classifying larger components, classifying true components as false positives among the higher volumes.

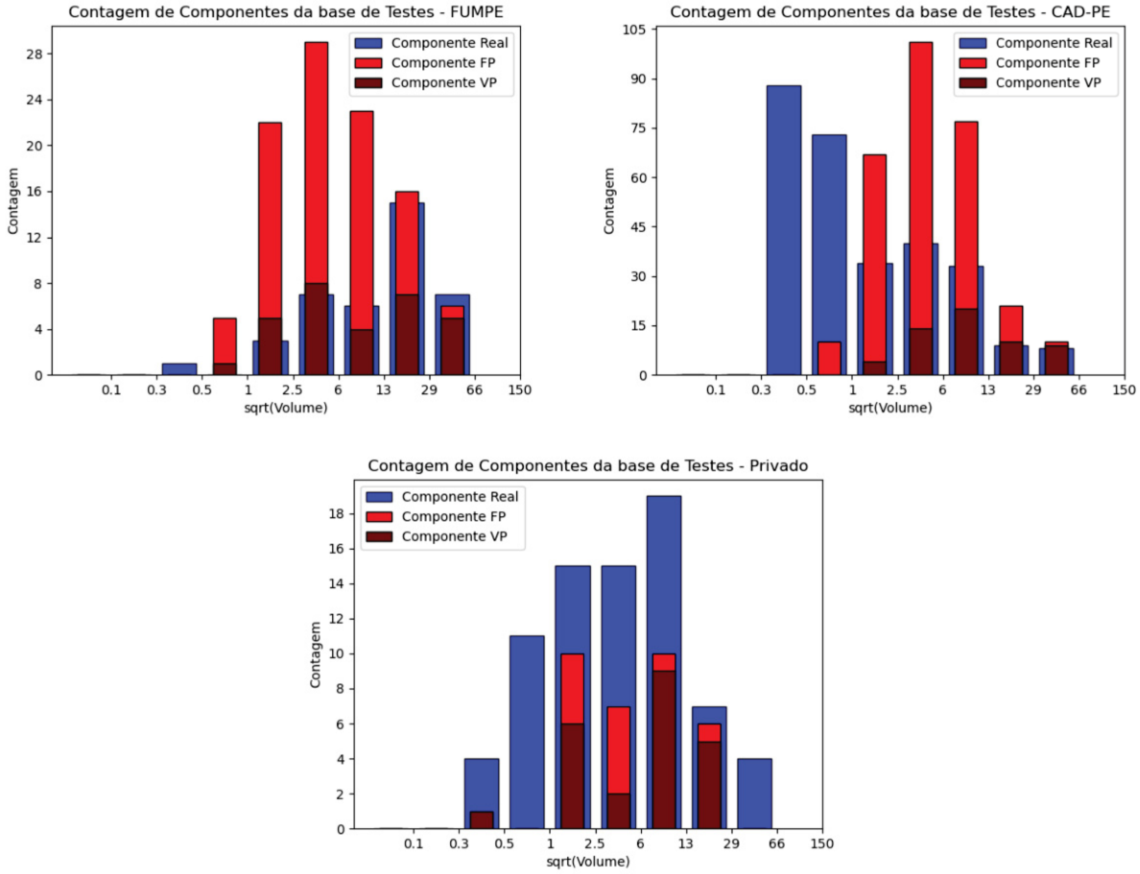


Figure 5.2: Graphs showing the square root count (y-axis) of the volumes (x-axis) of the components (PEs) found in the test set of the three datasets, in blue the true PEs (i.e. marked by the radiologist), in dark red the PEs true positives (TP) segmented by the U-net network and in yellow the false positive PEs (FP) produced by the network.

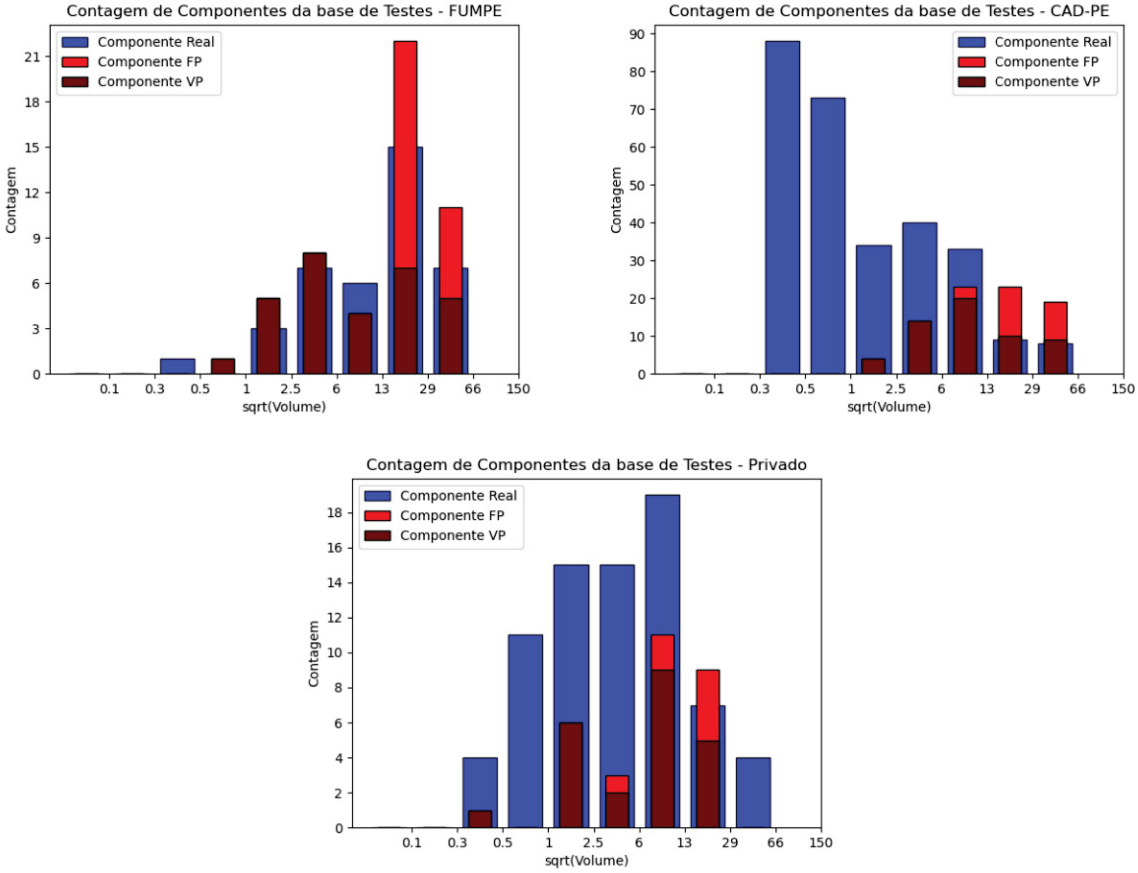


Figure 5.3: Graphs showing the square root count (y-axis) of the volumes (x-axis) of the components (PEs) found in the test set of the three datasets after applying the false positive classification method, in blue the true PEs (i.e. marked by the radiologist), in dark red the PEs true positives (TP) segmented by the U-net network and in yellow the false positive PEs (FP) produced by the network.

6 CONCLUSION

In this work, a method based on other state-of-the-art PE detection, although being able to find PE candidates and correctly classify them as false positives and true positives, it still has some limitations. The amount of false negatives, or PE that the U-net segmentation network was unable to find, especially in smaller PEs, shows that the network is not optimized to differentiate small noise or other artifacts from the thrombus. Besides that the number of false positives produced is still significant, even before the false reduction step.

The size and number of thrombi in the database directly influence the model's capacity. It is believed that it is possible to improve the results of the first step of the method with the U-net network using more tests of different hyper-parameters. The network proved to be able to distinguish PEs within the exams even though it produced some false positives, the model also proved to have difficulties in detecting smaller and more peripheral thrombi.

Even though some segmentation (Figure 6.1 (a)) achieved good results, the model still needs further optimizations and some hyperparameters can still be tweaked. The network seems to be capable to distinguish correctly PE from inside the exams, but it can produce some false negatives (Figure 6.1 (b), (c) and (d)), seeming unable to clots too small or with an odd shape. Some false positives were also observed in the test set (Figure 6.1 (e) and (f)), but some techniques like the 3D representation ((Tajbakhsh et al., 2015)) of the input or a HU window normalization can help to mitigate this problem ((Liu et al., 2020)), these false positives show the need of a second specialized FP removal network in this task.

The results between different datasets were also not that different from each other, the better overall results in the CAD dataset shows that the model performance scales with the amount of data provided for training, which was also confirmed in the experiments with a combination of datasets. It was also observed that some minor tweaks in the network were necessary when training with a low amount of data, especially in the Fumpe dataset, which the optimizer which gave the best results was the Adam, while the SGD did not mark any voxels as true positive, since there is a high amount of true negatives in each mask, a training with a "slower" optimizer such as the SGD can never reach a good result and be stuck in a local minimum. While training the network with a bigger amount of data has a similar problem but using the Adam optimizer, the loss drops drastically in the first few epochs but it does not reach a good result. The model also showed to be easily trained with different datasets and combinations,proving that it can give good results in real world applications.

6.1 FUTURE WORKS AND PUBLICATION

In order to improve the accuracy of the model in finding small components, other hyper-parameters of the network related to convolution and channel reduction can be explored, for example squeeze-and-excitation blocks as in the work of ((Huang et al., 2020)), performing model training with different combinations of databases can also increase the rate of found components. Models with 3D convolution operations also show an increase in the network segmentation rate as in ((Cano-Espinosa et al., 2020)) and ((Yang et al., 2019)). Other targeting networks can also serve as an alternative, such as Mask R-CNN. In the ((Long et al., 2021)) work the author uses a variation of the R-CNN Mask, called P Mask R-CNN, which is optimized to find small objects.

Regarding the number of false positives, approaches such as ((Tajbakhsh et al., 2015)) can increase the success rate of this step. In it, the author produced a single projection of the TEP, called 2.5D, using eigenvalues and eigenvectors, an image was generated with two channels, one longitudinal and the other transversal, of each thrombus, also explained in ((Tajbakhsh et al., 2019)) . With this representation, it is possible to classify thrombi with other image classification networks, facilitating training with the possibility of using pre-trained weights, e.g. networks such as InceptionNet ((Szegedy et al., 2017)), ResNext ((Xie et al., 2016)), EfficientNet ((Tan and Le, 2019)) and AmoebaNet ((Real et al., 2019)), all with a hit rate above 90% in the Imagenet database ((Deng et al., 2009)).

Regarding the publications related to the work on this thesis, two articles were accepted, one in the XXI Brazilian Symposium on Computing Applied to Health which took place virtually in july 2021, the title of the publication was "Detecção de Tromboembolia Pulmonar utilizando Redes Neurais Convolucionais e Extração de Características". Another work was also accepted but not yet published in the international journal Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization, with the title of "A Two Step Deep Learning Workflow for Pumonary Embolism Segmentation and Classification".

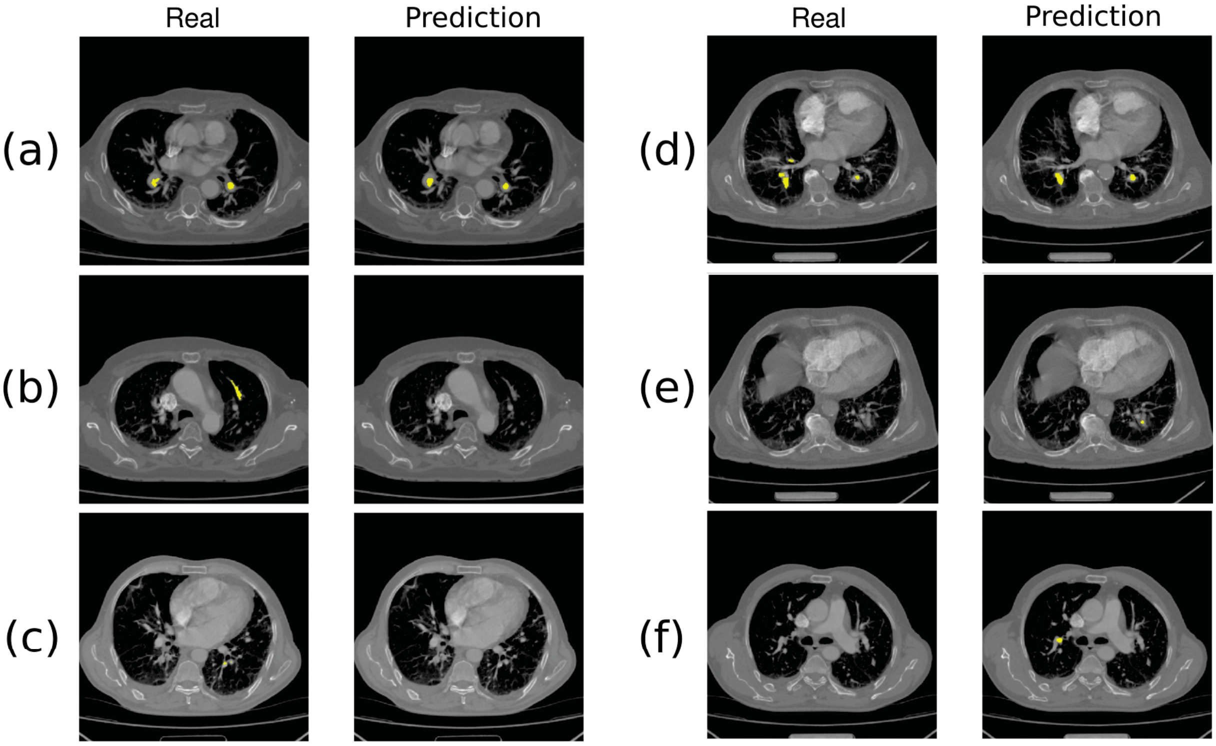


Figure 6.1: Ground truth and prediction results samples from the test set of (Masoudi et al., 2018), in (a) an example of a correctly labeled image, in (b) and (c) two examples where the network did not find PE when there are in the ground truth, in (d) only parts of the image were correctly segmented and in (e) and (f) two examples of a false positive label produced by the network.

REFERENCES

Agarwal, M. (2017). Back Propagation in Convolutional Neural Networks—Intuition and Code. `https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-and-code-714ef1c38199`. [Online; retrieved on August 2020].

Barco, S., Mahmoudpour, S. H., Valerio, L., Klok, F. A., Münzel, T., Middeldorp, S., Ageno, W., Cohen, A. T., Hunt, B. J., and Konstantinides, S. V. (2020a). Trends in mortality related to pulmonary embolism in the european region, 2000–15: analysis of vital registration data from the who mortality database. *The Lancet Respiratory Medicine*, 8(3):277–287.

Barco, S., Valerio, L., Ageno, W., Cohen, A. T., Goldhaber, S. Z., Hunt, B. J., Iorio, A., Jimenez, D., Klok, F. A., Kucher, N., et al. (2020b). Age-sex specific pulmonary embolism-related mortality in the usa and canada, 2000–18: an analysis of the who mortality database and of the cdc multiple cause of death database. *The Lancet Respiratory Medicine*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. USA: Springer-Verlag New York, Inc.

Bouma, H., Sonnemans, J. J., Vilanova, A., and Gerritsen, F. A. (2009). Automatic detection of pulmonary embolism in cta images. *IEEE transactions on medical imaging*, 28(8):1223–1230.

Buzug, T. M. (2011). Computed tomography. In *Springer Handbook of Medical Technology*, pages 311–342. Springer.

Cano-Espinosa, C., Cazorla, M., and González, G. (2020). Computer aided detection of pulmonary embolism using multi-slice multi-axial segmentation. *Applied Sciences*, 10(8):2945.

Carson, J. L., Kelley, M. A., Duff, A., Weg, J. G., Fulkerson, W. J., Palevsky, H. I., Schwartz, J. S., Thompson, B. T., Popovich Jr, J., Hobbins, T. E., et al. (1992). The clinical course of pulmonary embolism. *New England Journal of Medicine*, 326(19):1240–1245.

Chien, C.-H., Shih, F.-C., Chen, C.-Y., Chen, C.-H., Wu, W.-L., and Mak, C.-W. (2019). Unenhanced multidetector computed tomography findings in acute central pulmonary embolism. *BMC Medical Imaging*, 19(1):65.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Dalen, J. E. et al. (2002). Pulmonary embolism: what have we learned since virchow. *Natural history, pathophysiology, and diagnosis. Chest*, 122(4):1440–1456.

Dehmeshki, J., Ebrahimdoost, Y., and Qanadli, S. (2010). Fully automatic segmentation and detection of pulmonary artery and embolism in cta images.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Goldhaber, S. Z. and Bounameaux, H. (2012). Pulmonary embolism and deep vein thrombosis. *The Lancet*, 379(9828):1835–1846.

Gonzales, C. R. and Richard, W. E. (2018). *Digital Image Processing (4nd Edition)*. Pearson.

González, G., Jimenez-Carretero, D., Rodríguez-López, S., Cano-Espinosa, C., Cazorla, M., Agarwal, T., Agarwal, V., Tajbakhsh, N., Gotway, M. B., Liang, J., et al. (2020). Computer aided detection for pulmonary embolism challenge (cad-pe). *arXiv preprint arXiv:2003.13440*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Grus, J. (2019). *Data science from scratch: first principles with python*. O'Reilly Media.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Huang, S.-C., Kothari, T., Banerjee, I., Chute, C., Ball, R. L., Borus, N., Huang, A., Patel, B. N., Rajpurkar, P., Irvin, J., et al. (2020). Penet—a scalable deep-learning model for automated diagnosis of pulmonary embolism using volumetric ct imaging. *npj Digital Medicine*, 3(1):1–9.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jha, A. K., Larizgoitia, I., Audera-Lopez, C., Prasopa-Plaizier, N., Waters, H., and Bates, D. W. (2013). The global burden of unsafe medical care: analytic modelling of observational studies. *BMJ quality & safety*, 22(10):809–815.

Karpathy, A. (2018). CS231n Convolutional Neural Networks for Visual Recognition. `http://cs231n.github.io/convolutional-networks/`. [Online; Retrieved on July 2020].

Kligerman, S. J., Mitchell, J. W., Sechrist, J. W., Meeks, A. K., Galvin, J. R., and White, C. S. (2018). Radiologist performance in the detection of pulmonary embolism. *Journal of thoracic imaging*, 33(6):350–357.

Kreit, E., Mäthger, L. M., Hanlon, R. T., Dennis, P. B., Naik, R. R., Forsythe, E., and Heikenfeld, J. (2013). Biological versus electronic adaptive coloration: how can one inform the other? *Journal of the Royal Society Interface*, 10(78):20120601.

Kumamaru, K. K., Hoppel, B. E., Mather, R. T., and Rybicki, F. J. (2010). Ct angiography: current technology and clinical use. *Radiologic Clinics*, 48(2):213–235.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Liang, J. and Bi, J. (2007). Computer aided detection of pulmonary embolism with tobogganing and mutiple instance classification in ct pulmonary angiography. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 630–641. Springer.

Liu, W., Liu, M., Guo, X., Zhang, P., Zhang, L., Zhang, R., Kang, H., Zhai, Z., Tao, X., Wan, J., et al. (2020). Evaluation of acute pulmonary embolism and clot burden on ctpa with deep learning. *European Radiology*, pages 1–9.

Long, K., Tang, L., Pu, X., Ren, Y., Zheng, M., Gao, L., Song, C., Han, S., Zhou, M., and Deng, F. (2021). Probability-based mask r-cnn for pulmonary embolism detection. *Neurocomputing*, 422:345–353.

Maier, A., Syben, C., Lasser, T., and Riess, C. (2019). A gentle introduction to deep learning in medical image processing. *Zeitschrift für Medizinische Physik*, 29(2):86–101.

Mandic, D. and Chambers, J. (2001). *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. Wiley.

Masoudi, M., Pourreza, H.-R., Saadatmand-Tarzjan, M., Eftekhari, N., Zargar, F. S., and Rad, M. P. (2018). A new dataset of computed-tomography angiography images for computer-aided detection of pulmonary embolism. *Scientific data*, 5:180180.

Masutani, Y., MacMahon, H., and Doi, K. (2002). Computerized detection of pulmonary embolism in spiral ct angiography based on volumetric image analysis. *IEEE Transactions on Medical Imaging*, 21(12):1517–1523.

NHLBI (2011). Venous thromboembolism. `https://www.nhlbi.nih.gov/health-topics/venous-thromboembolism`. [Online; Retrieved on November 2020].

Özkan, H., Osman, O., Şahin, S., and Boz, A. F. (2014). A novel method for pulmonary embolism detection in cta images. *Computer methods and programs in biomedicine*, 113(3):757–766.

Ozkan, H., Tulum, G., Osman, O., and Sahin, S. (2017). Automatic detection of pulmonary embolism in cta images using machine learning. *Elektronika ir Elektrotechnika*, 23(1):63–67.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Rajan, D., Beymer, D., Abedin, S., and Dehghan, E. (2020). Pi-pe: A pipeline for pulmonary embolism detection using sparsely annotated 3d ct images. In *Machine Learning for Health Workshop*, pages 220–232.

Raskob, G. E., Angchaisuksiri, P., Blanco, A. N., Buller, H., Gallus, A., Hunt, B. J., Hylek, E. M., Kakkar, A., Konstantinides, S. V., McCumber, M., et al. (2014). Thrombosis: a major contributor to global disease burden. *Arteriosclerosis, thrombosis, and vascular biology*, 34(11):2363–2371.

Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. (2019). Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533.

Sadigh, G., Kelly, A. M., and Cronin, P. (2011). Challenges, controversies, and hot topics in pulmonary embolism imaging. *American Journal of Roentgenology*, 196(3):497–515.

Seeram, E. (2018). Computed tomography: A technical review. *Radiologic technology*, 89(3):279CT–302CT.

Sharma, S. (2017). Activation Functions: Neural Networks. `https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6`. [Online; retrieved on August 2020].

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

Tajbakhsh, N., Gotway, M., and Liang, J. (2015). Computer-aided pulmonary embolism detection using a novel vessel-aligned multi-planar image representation and convolutional neural networks. pages 62–69.

Tajbakhsh, N., Shin, J. Y., Gotway, M. B., and Liang, J. (2019). Computer-aided detection and visualization of pulmonary embolism using a novel, compact, and discriminative image representation. *Medical image analysis*, 58:101541.

Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

Terra-Filho, M. and Menna-Barreto, S. S. (2010). Recomendações para o manejo da tromboembolia pulmonar, 2010. *Jornal Brasileiro de Pneumologia*, 36:1–3.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2016). Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*.

Yang, X., Lin, Y., Su, J., Wang, X., Li, X., Lin, J., and Cheng, K.-T. (2019). A two-stage convolutional neural network for pulmonary embolism detection from ctpa images. *IEEE Access*, 7:84849–84857.

Yaniv, Z., Lowekamp, B. C., Johnson, H. J., and Beare, R. (2018). Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of digital imaging*, 31(3):290–303.