Alma Mater Studiorum · University of Bologna

**School of Science**
**Department of Physics and Astronomy**
**Master Degree in Physics**

# Automatic Pipeline for the Identification of Lung Lesions on CT Images of Patients Affected by COVID-19

**Supervisor:**
**Prof. Gastone Castellani**

**Co-supervisor:**
**Dr.Nico Curti**

**Submitted by:**
**Riccardo Biondi**

Academic Year 2019/2020

# Abstract

SARS-CoV-2 virus has widely spread all over the world since the beginning of 2020, and it is responsible of the COrina VIrus Disease (COVID). This disease affects lung areas and it causes respiratory illness.

Patienst affected by this pathology have shown a particular pattern of ground glass opacities (GGO) and consolidation (CS) in chest CT scans and that are made in relation both with the stage of the disease and its severity; moreover the study of these pattern on healed patient was used to check the actual recovery.

Up to now the gold standard for the identification and quantification of these areas is made by manual or semiautomatic segmentation which are subjected to the operator experience moreover are time consuming. Considering this scenario, an automatic segmentation of these areas is required. In this work I will present an automatic pipeline for the segmentation of GGO and CS. The pipeline achieves the segmentation by applying the color quantization as medical image segmentation technique, obtaining a classification based on voxel intensities. Exploiting the properties of digital multichannel images, I was able to incorporates also other image properties related to the voxel neighborhood.

The proposed approach, tested on three datasets, has shown to achieve good segmentation on chest CT with typical lesions.

# Contents

# Introduction

Since the end of 2019, COVID-19 has widely spread all over the world. Up to now the gold standard for the diagnosis of this disease are the reverse transcription-polymerase chain reaction (RT-PCR) and the gene sequencing of sputum, throat swab and lower respiratory tract secretion [1].

An initial prospective made by Huang et al. on chest CT scans [2] of patients affected by COVID-19, has shown the 98% of examined patients have bilateral patchy shadows or ground glass opacity (GGO) and consolidation(CS); moreover the severity, shape and involved percentage of lung were in relation with the stage of the disease [3]. In the end, other studies have monitored the change on volume and shape of these features on healed patients [4] in order to monitor their actual recovery. In Figure 1 are compared slices of patient with a different severity of the disease. From left to right we see an increment in the volume of GGO.

**Figure 1:** *Groud Glass Opacity and Consolidation on chest CT scans of COVID-19 affected patients with different severity of the disease. From left to right we can observe an increment of the GGO areas in the lung*

GGO and CS are not exclusive of COVID-19, but may be also caused by pulmonary edema, bacterial infection, other viral infection or alveolar haemorrage [5]. The combination between CT scan information and other diagnostic techniques like the RT-PCR mentioned above, may help the diagnosis, the monitoring of the course of the disease and the checking of the recovery in healed patients; moreover the study of these patterns may help to understand the infection pathogenesis, which is not well known since COVID-19 is a new disease.

Identification and quantification of these lesions in chest CT scans is a fundamental task. Up to now the segmentation is made in a manual or semiautomatic way, which are time consuming(several hous or days) and subjected to the operator experience, since involves the interaction with trained personnel. moreover these kind of segmentation cannot be reproduced. To overcome these issues an automatic and fast way for the segmentation is required.

This wirk of thesis, made in collaboration with the Department of Diagnostic and Preventive Medicine of the Poloclinico Sant'Orsola - Malpighi, aims to develop an automatic pipeline for the identification of GGO and CS in COVID-19 affected patients. The work was based and tested on chest CT scans provided by Sant'Orsola, but also public repositories [6] [7] where used as benchmark.

We start the discussion by understanding what a CT image is, its physical meaning and digital representation; so a brief review on the main image segmentation techniques will be presented, describing the main features of them. More details will be given on the techniques used for the actual implementation.

The discussion will continue by describing the main pipeline characteristics and the main pipeline structure. We will see how color quantization was used to achieve the segmentation and how the digital image properties were used in order to take into account different image features. We also discuss how a preliminary lung segmentation will help the performances of the segmentation. After that we will continue the discussion by describing in details each step of the pipline and after that how are implemented.

In the end we will discuss the segmentation results. The pipeline performances were checked trough different method, like visual comparison with other segmentation techniques, quantitative comparison against manual annotation and blind evaluation by experts. Also the segmentation achieved on healthy control was considered as benchmark.

# Chapter 1

# Image Segmentation techniques

Image segmentation consists in the partitioning of an image into non overlapping consistent regions that are homogeneous respect to some characteristics, such as intensity or texture [8]. The results of segmentation can be used to perform feature extraction, that provides fundamental information about organs or lesion volumes, cell counting, etc. If the patient perform several analysis during time, image segmentation is a useful tool to monitor the evolution of particular lesions or tumors during a therapy. Nowadays several non-invasive medical imaging techniques are available, such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) or X-Ray imaging, that provide a map of the subject anatomy. Image segmentation plays a crucial role in many medical-imaging applications by automating or facilitate the delineation of anatomical structures and other regions of interest [8]. Manual segmentation is possible, but it is time consuming and subject to operator variability making the results difficult to reproduce [9]. Therefore an automatic or semi-automatic methods are preferable.

The major difficulty in medical image segmentation is the high variability in medical images. First and foremost, the human anatomy itself shows major modes of variation. Furthermore many different modalities (X-ray, CT, MRI, etc.) are used to create medical images [10].

In this chapter I will provide a brief introduction about medical images, focusing mainly on Computed Tomography, which is the technique used tp acquire the images segmented in this work. This part is followed by a discussion on the main image segmentation techniques.

## 1.1 Medical Images

A medical image is the representation of internal structure or function of anatomic region in the form of array of picture elements (pixels/voxels). This discrete representation results from a process that map each numerical value in a position of the space. The number of pixels used for this representation is an expression of the details with which the anatomy of function can be depicted. The physical meaning of the data changes according to the image acquisition modality (CT, MRI, PET, etc.); for instance, we can consider the computed tomography (CT) case:

Computed Tomography (CT) is a medical imaging technique which aims to reproduce cross-section images and the 3D anatomy of the examined subject. Each

data represents the capability of the corresponding volume to attenuate an x-ray beam. In order to match results from different scans the beam attenuation is measured in Housfied Units(HU) :

$$CT - number = k \times \frac{\mu - \mu_{H_2O}}{\mu_{H_2O}} \tag{1.1}$$

Where $\mu$ is the linear attenuation coefficient of the tissue, $\mu_{H_20}$ is the linear attenuation coefficient of the water, took as a reference, and $k$ is a constant which can be 1000 or 1024 according to manufacturer scan. The linear attenuation coefficient of the air is considered as 0, so the corresponding CT number is $-1000$; for the bones, that have a density double than water, the CT number is 1000.

Medical images can be characterized by 5 properties : *pixel depth, photometric, interpretation, metadata and pixel data.*

**Pixel depth** is the number of bits used to encode the information of each pixel. Each value of the tensor is integer of floating point number belonging to a domain related to the image format. It is related to the memory space necessary to store the image and the amout of information we want to store in each pixel. Higher bit allows to store more information, but requires more memory [**ART:Larobina**]. The most common are $[0, 255]$ for $8-$bit integer image, of $[0, 1]$ for float. Also other formats are available, like $16-$bit integers, widely used to represent medical images.

**Photometric Interpetation** The photometric interpretation specifies how the pixel data should be interpreted for the correct image display as a mono95 chrome or color image. We have to introduce the concept of *number of channel*. Monochrome images have only one sample per pixel (GL intensity). To encode color information into pixels, we typically need multiple samples per pixel and to adopt a color model that specifies how to obtain colors combining the samples. Color may be used to encode blood flow direction and velocity in doppler ultrasound [**ART:Larobina**]. In this works I have used this feature to consider also voxels neighbouring information.

**Metadata** Are the information that describe the image. Metadata is tipically stored at the beginning of the the file and contains (at least) the image matrix dimension and photometric interpretation. In medical image formats it contains also informations how the image was produced or about the patient [**ART:Larobina**].

**Pixel Data** Numerical values of the pixels that are stored according to data type.

### 1.1.1 Medical Image Formats

Image file formats provide a standard way to store the information describing an image in a computer file. Moreover file format describes how the image data are organized inside the image file and how the pixel should be interpreted by a software for the correct loading and visualization.

Medical file formats can be divided in two categories: one that try to standardize the images generated by diagnostic modality(e.g. DICOM), the other that try to facilitate the post processing analysis (e.g. Nifti). Both of these type of images stores image data and metadata at the beginning of the file [**ART:Larobina**].

**DICOM** , acronymes for Digital Imaging and COmmunications in Medicine, is not only a file format but also a network communication protocol. The added value of its adoption in terms of access, exchange, and usability of diagnostic medical images is, in general, huge. Dicom file format establish that the pixels data cannot be separated from the description of the medical procedure which lead to the formation of the image itself. The header also contains patient informations such as name, gender, age, etc. So the header allows the image to be self descriptive. DICOM is born for only 2D images, so a 3D volume is described by a series of files containing the single slices [**ART:Larobina**].

**Nifti** primary gol is to provide coordinated and targeted service, training, and research to speed the development and enhance the utility of informatics tools related to neuroimaging. This file format uses the header to store informations about image orientation image center and origin. This avoid left-right brain hemisphere ambiguity. Even if this file is born for neuroimaging,can be used also to store other kind of iamges like chest CT. The format is supported by many viewers and image analysis software like 3D Slicer, ImageJ, and OsiriX [**ART:Larobina**].

## 1.2 Review on Image Segmentation Methods

During the years, several segmentation methods have been developed and they are based on a lot of different approaches. These methods can be classified into several way, for example we can divide them into *supervised* or *unsupervised* if they requires or not a set of training data, or can be classified according to the used information type, like *Pixel classification methods*, which use only information about pixel intensity, or *Boundary following* methods, which use edge information, etc. In this section I will provide a brief review on the main segmentation methods, organized in the same way as in [8], that divides the methods in 8 categories:

1. Thresholding,
2. Region growing,
3. Classifiers,
4. Clustering,
5. Markov Random Fields models,
6. Artificial Neural Networks,
7. Deformable Models,
8. Atlas guided approaches.

### 1.2.1 Thresholding

Thresholding approach is very simple and it basically segments a scalar image by creating a binary partitioning of image intensities [8]. It can be applied on an image to distinguish regions with contrasting intensities and thus differentiate between

tissue regions represented within the image [9]. Figure 1.1 shows an histogram of a scalar image with two classes, threshold based approach attempts to determine an intensity value, called *threshold* which splits the desired classes [8]. To achieve the segmentation we can group all the pixels with intensity higher than the threshold in one class an all the remaining ones into other class.

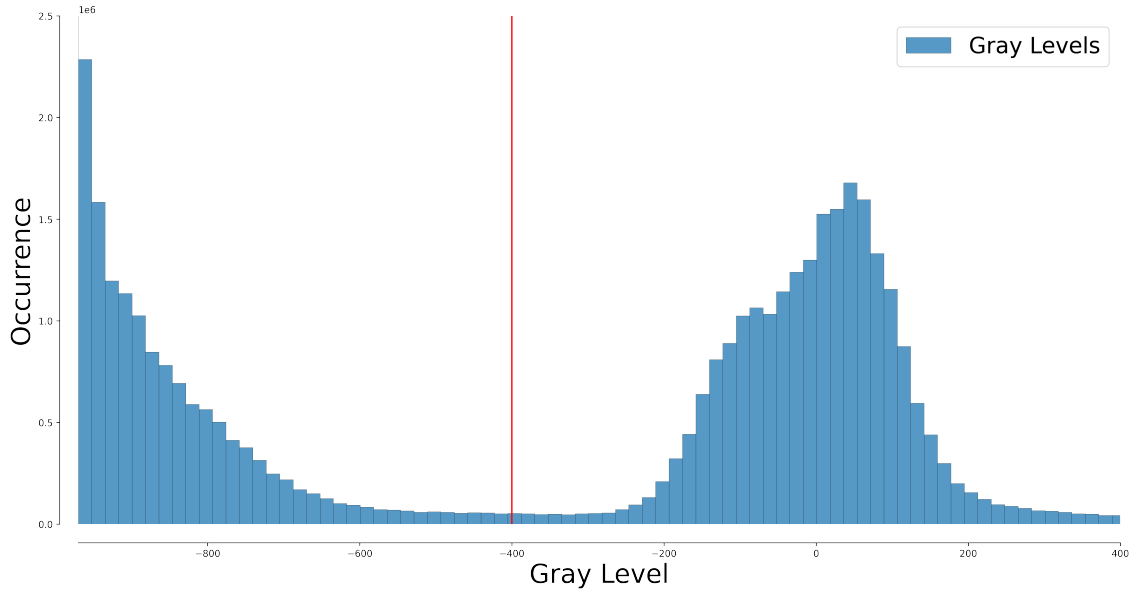



**Figure 1.1:** *Histogram of a GL image with two well delineated regions.The threshold value(red line) was setvisually at -400 HU*

The threshold value is usually setting by visual assessment, but can also be automatized by algorithms like Otsu one. Sometimes may happen that more than two classes are present in the image, so we can set more than one threshold values in order to achieve a multi-class segmentation, also in this case there are algorithms to automatized this process, like an extension of the previous one called *multi Otsu threshold*.

This is a simple but very effective approach to segment images when different structures have an high contrast in intensities. Threshold does not takes into account the spatial characteristics of the image, so it is sensitive to noise and intensity inhomogeneity, that corrupt the image histogram of the image and that make difficult the separation [8]. To overcome these issues several variations of threshold have been proposed based on local intensities and connectivity.

Threshold is usually used as initial step in sequency of image processing operations, followed by other segmentation techniques that improve the segmentation quality. Since threshold uses only intensity information, it can be considered a pixel classification technique.

### 1.2.2 Region Growing Approach

Region growing approach allows to extract connected regions from an image. This algorithm starts at seed location in the image (usually manually selected) and checks the adjacent pixels against a predefined homogeneity criteria [9], based on intensity, and/or edges. If the pixels met the criteria, they are added to the region. A continuos application of the rule allows the region to grow.

Like thresholding, region growing is used in combination with other image segmentation operations, and it usually allows the delineation of small and simple structures such as tumor and lesions [8].

Regions growing can also be sensitive to noise: the extracted regions may have holes or even become disconnected. May also happen that disjoint areas become connected due to partial volume effect.

When we use this approach we have to consider that for each region we want to segment a seed must be planted. There are some algorithms, related to region growing, that does not require a seed point, like split and merge one. Split and merge operates in a recursive fashion. The first step is to check the pixel intensity homogeneity: if they are not homogeneous, the region is splitted into two equal sized sub-regions. This step leads to an oversegmentation, so a merging step is performed, which merge together adjacent regions with similar intensities [9].

### 1.2.3 Deformable Model

Deformable Models use an artificial, closed, contour/surface able to expand or contract over time and conforme to a specific image feature [9]. This approach is physically motivated model-based technique for the detection of region boundaries [8].
The curve/surface is placed near the desidered boundary and it is deformed by the action of internal and external forces that act iteratively. The external forces are usually derived from the image.
This approach has the capability to directly generate closed parametric curves or surfaces from images and and incorporate smootness constraint that provides robustness to noise and spurioous edges.
However this approach requires a manual interaction to place the appropriate set of parameters.

### 1.2.4 Markov Random Field

Markov Random Field (MRF) is not a proper segmentation method, but it is a statistical model that is used within segmentation methods which models the spatial interaction between neighbouring pixels. It is often incorporated in clustering algorithms such as K-means with a Bayesian prior probability.
This model is used when most pixels belong to the same class as their neighbouring pixels, in this casesany anatomical structure that consist of only one pixel has a very low probability of occourring [8].
A difficulty of this model is that it is very sensitive to the parameters that controls the strenght of the spatial interactions. An other MRF disavvantage is that it is a very computationally expansive algorithms. However, despite these disavantages, MRF are widely used to model segmentation classes and intensity inhomogeneities [8].

### 1.2.5 Classifiers Approach

Classifiers approaches use statistical pattern recognition techniques to segment images by using a mixture model which assumes that each pixels belongs to one of a prior known set of classes [9]. To assign each pixel to the corresponding class, it

use the so called *feature space*, which is the space of any function of the image. An example of 1D feature space is the image histogram.

The features of each pixel form a pattern that is classified by assigning a probability measure for the inclusion of each pixel in each class [9].
This approach assumes a prior knowledge about the total numbers of features in the image and the probability of occurence of each class. Generally this quantity is not prior known, so we need a set of training data to use as reference.
There are different techniques which use this approach:

- **k-Nearest Neighborhood** : each pixel is classified in the same class as the training data with the closest intensity;
- **Maximum likelihood or Bayesian** : Assume that pixel intensities are independent samples from a mixture of probability distributions and the classification is obtained by assigning each pixel to the class with the highest posterior probability.

This approach requires a structure to segment with distinctive and quantificable features. It is computational efficient and can be applied to multichannels images. This approach does not consider a spatial modelling and it requires a manual interaction to obtain the training data that must be several since the use of the same training set for a large number of scans can lead to biased results.

### 1.2.6 Clustering

Clustering approach is similar to the classifiers one, but in an unsupervised faishon, so does not require a training dataset. Clustering iteratively alternates between the image segmentation and the its còasses properties characterization. In this way we can say that clustering approach trains itself by using the available data information. We can identify 3 main clustering algorithms:

- **k-means clustering:** partition $n$ observations in $k$ clusters iteratively computing a mean intensity for each class. It segments the image by classifying each pixel in the class with the closest mean
- **Fuzzy C-means:** this algorithm generalizes the K-means clustering in order to achieve soft-segmentation;
- **Expectation Maximization:** uses the same clustering principle as k-means by assuming that each observation belongs to a Gaussian mixture model. It is an iterative method that seek to to find maximum likelihood estimates means, covariances and mixing coefficients of the mixture model.

It does not require training data, but if suffers of an high sensitivity to the initial parameters and it does not incorporate spatial models: it is a pixel classification technique [8].

The most used algorithm for clustering is the k-means clustering, which seeks to assign each pixel to a particular cluster aiming to minimize the average square distance between pixels in the same cluster [11]. Each cluster is described by a vector representing its center, so this technique is described as a centroid model [12]. The

labeling is performed by assigning each point to the cluster with the nearest centroid. Each point is assigned to the cluster with the nearest center.

Given an integer $k$ and a set of $n$ data points from $\mathbb{R}^d$, the k-means clustering seeks to find the $k$ centers that minimize a potential function given by the sum of squares:

$$\Phi = \sum_{x \in S} \min \|x - c\|^2 \tag{1.2}$$

Where $S \subset \mathbb{R}^d$ is a set of points. In this work $\mathbb{R}^d$ will be the colors space and $S$ is the space of color of each voxel.

The steps of the algorithm are the following:

1. Randomly initialize the values of the $k$ centers,
2. Generates the $k$ clusters associating each point to its nearest centroid
3. Re-compute the centroid for each cluster until the centroid does not change.

Arthur and Vassilvitskii [11] have pointed out that this algorithm is not accurate and can produce arbitrarily bad clusters. So they have developed a method, the "k-means++" which improves the clustering accuracy by made an accurate choice of the initial cluster centers.

They pointed out that the bad clustering is caused to the fact that $\frac{\Phi}{\Phi_{opt}}$ is unbounded even if the number of clusters and points are fixed, where $\Phi_{opt}$ is the potential function in the optimal centroids case. They proposed a variant for the choosing of the centroids: They randomly chose omly the first centroid. To chose the other, they weight the initial points according to the distance square $(D(x)^2)$ from the closest center already chosen. So the final algorithm is equal to the k-means except for the initial centroids selection that is made as follows:

1. Take one center $c_1$, chosen uniformly at random from $S$.
2. Take a new center $c_i$, choosing $x \in Si$ with probability $\frac{D(x)^2}{\sum_{x \in S} D(x)^2}$
3. Repeat the step 2 till $k$ centers are chosen
4. Proceed like a classical k-means clustering.

They have proved that this approach leads to better results in less time. For more details refer to [11].

### 1.2.7 Color Quantization

Color quantization is the process of reducing the number of colors in a digital image preserving significant information. In other world quantization process should not cause significant information loss in the image. The color quantizes image should be very close to the original image visually, as in Figure 1.2.

Color quantization plays an important role in many application fields such as segmentation, compression, color texture analysis, watermarking, text localization/detection, non photorealistic rendering and content-based retrieval [13].

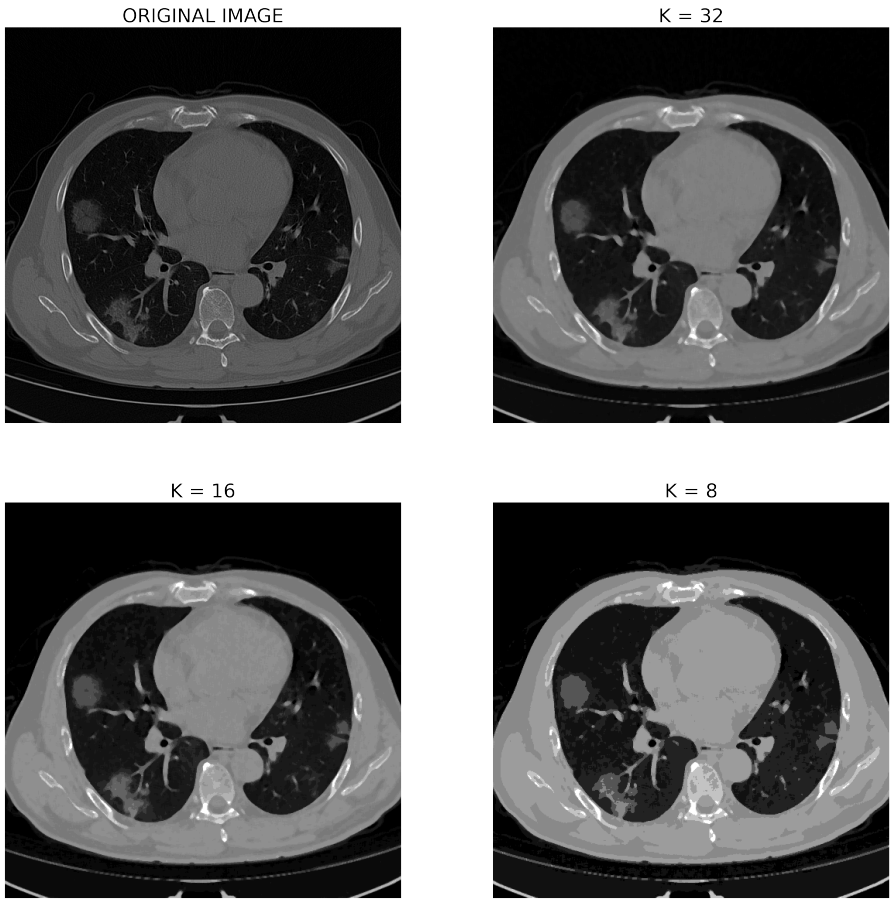



**Figure 1.2:** *Color quantized GL image. We observe the original image, a 32 color image which look similar to the original one, a 16 colors image and 8 colors image. Tissue are grouped by color similarity. Reduce the color to the number of cluster will*

Color quantization is used for image segmentation. To this purpose the algorithm aim to reduce the number oin the image is assigned an unique characteristic color. This is the case of medical image segmentation, in which each image color represents a particular characteristic of the tissue displayed (i.e in x-ray represent$\mu$). To perform this technique, different algorithms may be used to group the colors, like clustering algorithm or the principal component analysis.

### 1.2.8 U-Net

Artificial Neural Networks(ANN) are a computational architecture derived from neural physiological models [9]. This architecture is made by interconnected artificial neurons able to perform elementary operations. For imagery analysis are usually used Convolutional Neural Networks(CNN), also known as shift invariant or space invariant artificial neural networks (SIANN).

In biological image processing the so called U-Net are usually used. U-Net is a kind of convolutional neural network which allows to overcome the requirement of many training data. That because large training dataset are not always available, like often happens in medical and biological segmentation purposes.

Convolutional Networks are usually applied to classification tasks. In biological

and medical purposes, the segmentation should include localization and a class label should be assigned on each voxel. To achieve this purposes, in 2015 for the ISB cell tracking challenge, Olaf Ronneberger, Philipp Fischer, and Thomas Brox have developed this kind of network [14].

The U-Net models are modified and simplified version of a fully connected convolutional neural network, able to work with only small set of training samples.

The whole structure is divided into two main parts:

- Contraction path (*encoder*) : sequence of convolutional and pooling layers, which aim to extract features and reduce the input dimensionality.
- Expansion Path (*decoder*) : second set of convolutional and up-sampling layers, to reconstruct the feature map and consequently the segmentation mask, which aims to process the extracted features

The contractive path and the extractive path. The contracting path is a typical convolutional network that consists of repeated application of convolutions, each followed by a rectified linear unit (ReLU) and a max pooling operation. During the contraction, the spatial information is reduced while feature information is increased. The expansive pathway combines the feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features from the contracting path [14].

Decode path tends to lose some of the higher level features that the encoder learned: Using shortcut connection, the output of the encoding layers are directly passed to the decoder layer, preserving the important features [15].

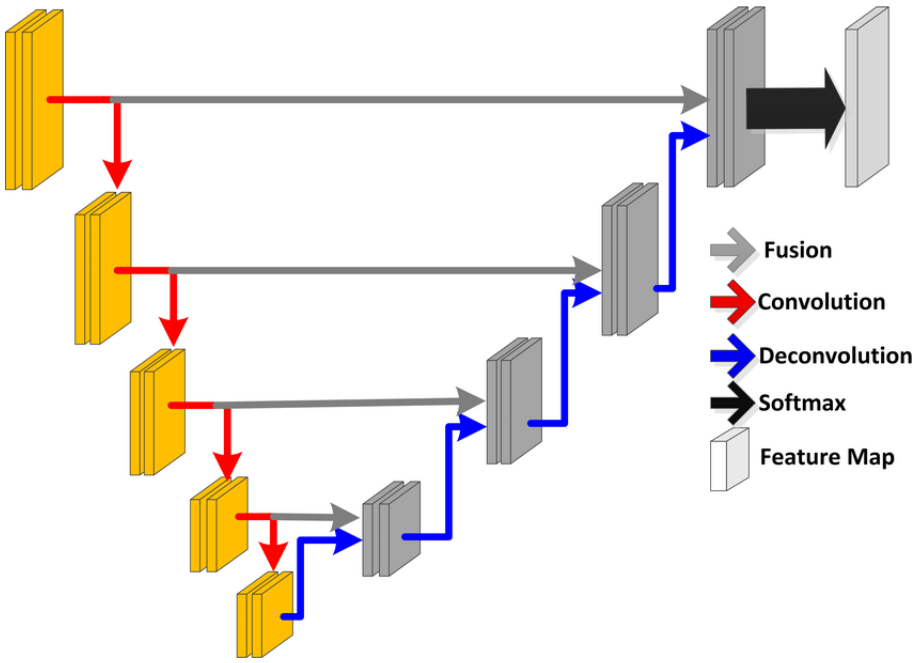



**Figure 1.3:** *UNEt network architecture: We can see the U-shape made by symmetricity between expansion and contraction path. The gray arrows indicates the shortcut used to prevent information loss*

One of the important modification of this network is that in the upsampling part have also a large number of feature channels, which allow the network to propagate

context information to higher resolution layers, as a consequence the expansive path is symmetric to the contractive one, making the U shape, as we can see in the structure diplayed in Figure 1.3. In a U-Net the segmentation map only contains the pixels for which the full context is available [14].

In order to work with few training data,we have to make a huge data augmentation, by applying an elastic deformation to the training images.MainSegmentation

# Chapter 2

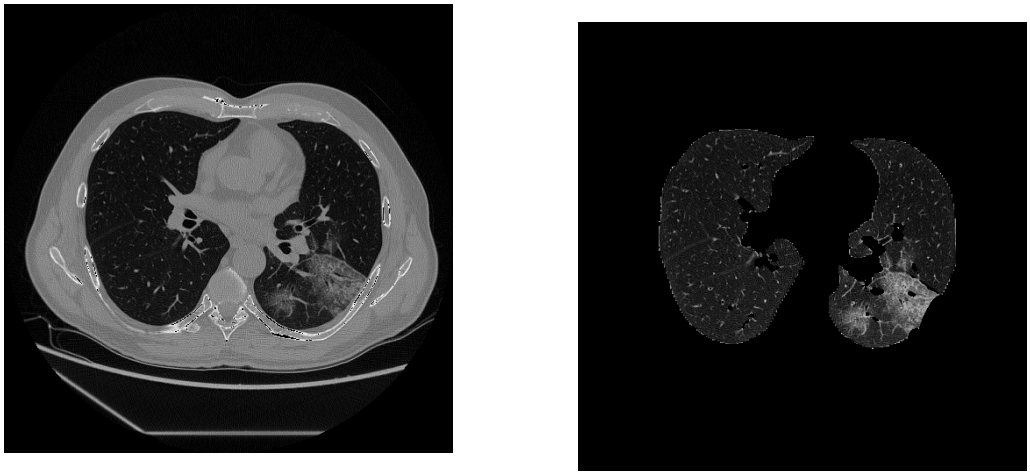
# Pipeline

The aim of these work of thesis is the developing of a automatic pipeline for the segmentation of Ground Galass Opacities and Consolidation areas in chest CT scans of patients affected by COVID-19.

Austin in *Glossary of terms for CT of the lungs* [16] has defined the GGO and CS as :

"*Hazy increased attenuation of lung, but with preservation of bronchial and vascular margins; caused by partial filling of air spaces, interstitial thickening, partial collapse of alveoli,normal expiration, or increased capillary blood volume, which is different from consolidation in which bronchovascular margins are obscured.*

.



**(a)** *Chest CT scan of a patient affected by COVID-19*

**(b)** *Isolated lung regions*

**Figure 2.1:** *Original Chest CT scan(a), we can clearly see the GGO and CS regions. Moreover all extra -lung regions are present. In (b) we can see the same slice but after the lung segmentation. We can see that each different region has a similar gray level.*

The starting point are the chest CT scans (Figure 2.1a). Firstly I have removed all the extra lung regions like body, bronchial structures and CT tube which are a potential source of false positives. Once we have obtained a scan as Figure 2.1b, we can

notice that the different structures are characterized by similar gray level: the basic idea was to use the color quantization for medical image segmentation, grouping voxels based on color similarity, and assigning to each tissue a characteristic color. This can be done since in CT scan exists a relationship between the tissue in the voxels and the Gray Levels used to display it, given by the Hounsfiend Units(eq 1.1): the colors are proportional to HU, which are defined as a linear transformation of the linear attenuation coefficient($\mu$).

We can consider different properties beside the single voxel intensity. As we can see, lesion areas involves many closest voxels. It is interesting to incorporates also neighborhood voxel information in the color quantization. Moreover, contrast between sick and healthy areas may change according to the severity of the disease. As a consequence it is interesting to incorporates also different gamma of the image, in order to enhance these regions.

Adopting a color model for the digital image, allows to encode color information in the pixels. Color can be used to encode also voxel neighborhood information. Each image channel stores a different image function.

Once I have build the color space, I have to find the characteristic color of each tissue under study, which is represented by a centroids in the color space. I have created a training dataset made of CT scans from different patients and applied a k-means clustering, since it provides a good balance between segmentation performance and computational efficiency. K-means clustering requires a prior knowledge about the number of cluster, which in our case is given by the anatomical structure of the lung. I have consider a different cluster for each anatomical structure :

- Lung Parenchima;
- Edges;
- Vessel surrounding bronchial structures;
- Bronchi.
- Ground Glass Opacities and Consolidation;

Once I have estimated the centroids for each tissue, I have used them for the segmentation: each voxel of the image is assigned to the cluster of the closest centroids.

The final pipeline structure is summarized in Figure 2.2.

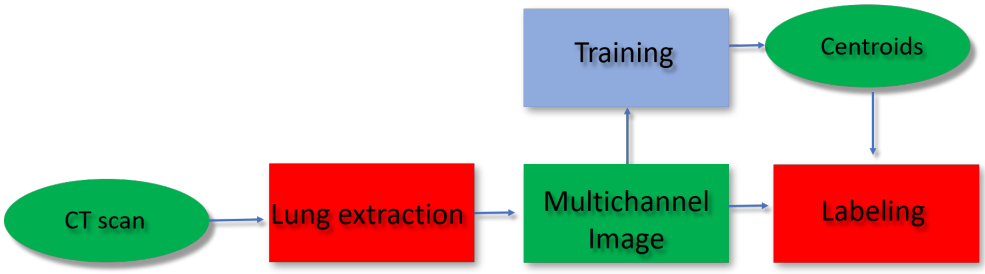



**Figure 2.2:** *Actual segmentation step, from left to right we can see the input image stack, the isolated lung regions and the final label. To performed the labeling a set of pre-computed centroids was used.*

Once the centroids are estimated, the training step does not needs to be repeated. Moreover, in the implementation, the building of the multichannel image is incorporated in the labeling step. As a consequence the final pipline structure looks like in Figure 2.3.

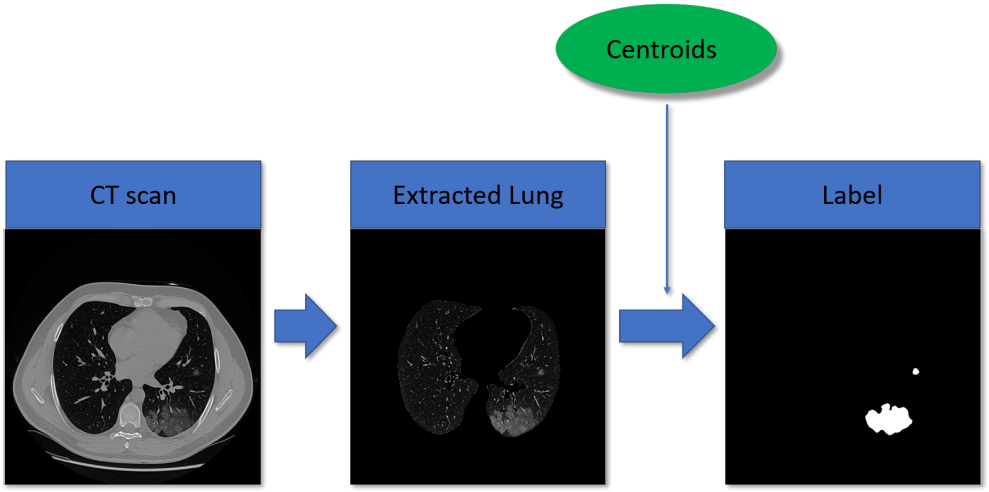



**Figure 2.3:** *Actual segmentation step, from left to right we can see the input image stack, the isolated lung regions and the final label. To performed the labeling a set of pre-computed centroids was used.*

## 2.1 Description

As we have seen, the pipeline is divided into many blocks, each one performing a different task. In this section I will go in deep on how each of these task is achieved. This section involves only a description, the implementation will be discuss in the next one.

### Lung Extraction

This preliminary step is performed before both training and labeling and involves the managing of the HU, the isolation of lung regions and the removal of the main bronchial structures.

The registration of the HU on a common space is necessary to overcome the issues that may raise from the different padding values and multiplicative constant for HU computation (equation 1.1) used by the different manufacturer of the CT scans. The $k$ constant in the HU definition (equation 1.1) may change according to the scan manufacturer or scan model. During the scan acquisition, all the regions outside the CT tube aren't sampled, so to obtain a square $N \times N$ image for each slice some padding values are added, which different values according to the scan manufacturer. This registration allows to overcome some issues that may raise during the conversion of the image from 16 to 8 bit unsigned integers, required to operates with OpenCV functions.

Lung segmentation is a pivotal pre-processing step in many image analysis such as identification and classification of lung pathologies [14]. The lung isolation allow us to fund a mask for the lung regions, and thus excluding all the body regions, the

CT tube and the extra-lung organs like intestine and heart, avoiding the formation of false positives.

Automatic lung segmentation algorithm are typically developed and tested on limited datasets and usually over a limited spectrum of visual variability by containing mainly cases without severe pathologies [14]. Rule based approach, like thresholding, region growing, ect, usually fails for CT sans of patients with severe Interstitial Lung Disease (ILD), as we can see in Figure 2.4. To achieve the lung segmentation I've used a pre-trained U-Net [14] [17].

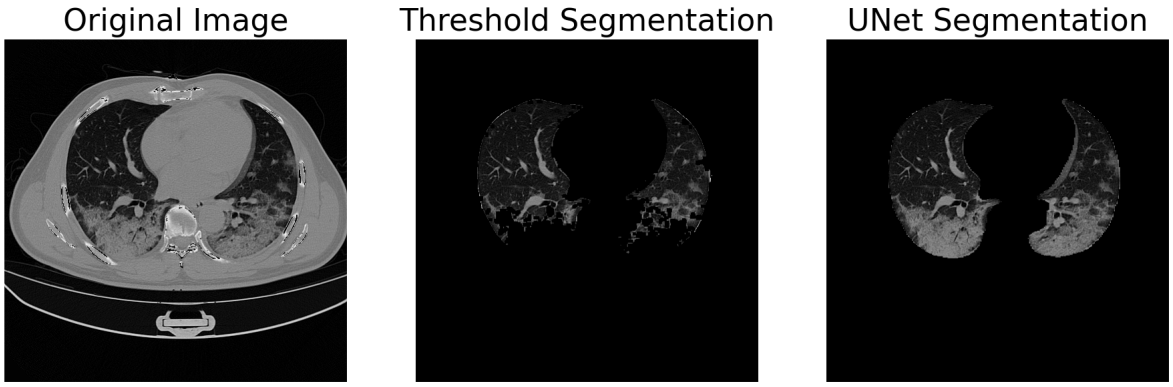



**Figure 2.4:** *From left to right the original CT scan of a patient with severe ILD, the lung segmented by threshold and it's connected components, and lung segmentation achieved by the pre-trained U-Net. We can clearly see the missing areas in the first segmentation, correctly identified in the U-Net one.*

This kind of segmentation includes in the lung region also motion artifacts and bronchial structures, which we will see are the principal causes of false positives. To achieve a better segmentation a refinement process is performed, which aims to remove the main bronchial structures form the selected lung regions.

Bronchi have an elongated shape respect to the other structure which usually are rounded: the basic idea was to use this kind of information. In order to perform this task, I have computed the covariant matrix of the derivative in a neighborhood (2.1) and the corresponding eigenvalues.

$$M = \begin{bmatrix} \sum_{S(p)} (dI/dx)^2 & \sum_{S(p)} dI/dx dI/dy \\ \sum_{S(p)} dI/dx dI/dy & \sum_{S(p)} (dI/dy)^2 \end{bmatrix} \tag{2.1}$$

If a particular regions has an elongated shape, one of the eigenvalues (corresponding to the eigenvector in the direction of the structure) will be higher than the other, otherwise both eigenvalues have are lower. I have applied this filter on each slice of the scans and took a map of the maximum eigenvalues.

In Figure 2.5, I've displayed the image after the lung segmentation by the neural network, and the corresponding eigenvalues map. As we can see the higher values of the map corresponds to the edges of the main bronchial structures. To create the mask for these structures I have applied a fixed threshold to map. Since the main bronchial structures are large, this process is able to remove only part of the edges, but the inner structure is preserved. In order to refine the segmentation, this process is repeated a second time, allowing a more accurate exclusion of the structures.

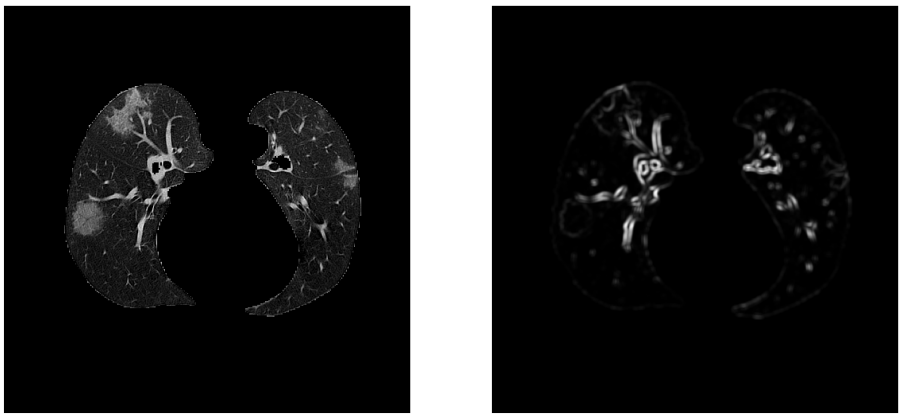


**Figure 2.5:** *From left to right: lung regions selected by the U-Net; ,maximum eigenvalues map of the lung. As we can see the U-Net does not exclude the bronchial structure from the lung, on the other hand, the maximum eigenvalues map delineates very well these regions. I have used this map to remove the unwanted bronchial regions.*

## Multi Channel Image

This step involves the preparation of images, with the building of the multi channel image that incorporates neighbouring and edges information. The used image is composed by 4 channel built as follows:

- Pure image after Contrast Limited Adaptive Histogram Equalization (CLAHE), with a block size of $10 \times 10$ pixels
- Image after a median blurring with kernel size equal to 11 pixes
- Image after a gamma correction with $\gamma = 1.5$
- Standard filtered image with a kernel of size 3 pixels

In Figure 2.6 I have displayed the 4 different channels of the image. Each channel allow us to consider different information.

The histogram equalized image and the gamma corrected allows to take into account information about the single voxel. The histogram equalization is applied in order to enhance the image contrast by improving the GL usage. For each slice the histogram is equalized considering only a $10 \times 10$ area, in order to take care of the over-amplification of the contrast.

The gamma correction is a non linear operation and is used to decode the luminance, and to made in evidence the less evident lesions.

The median blurring allow us to consider also the information of the neighborhood voxels, allowing the reduction of the outliers. The usage of the filter is justified since the lesions involves several closest voxels.

The last channel cosist in the image after the application of a local standard deviation filter. This filter consist in the replacement of each pixel value with the standard deviation of its neighborhood. It help us to distinguish the bronchial structures and motion artifacts not removed during the lung segmentation.

Each image channel is normalized according to means and standard deviation of the whole scan; because of the k-means clustering.

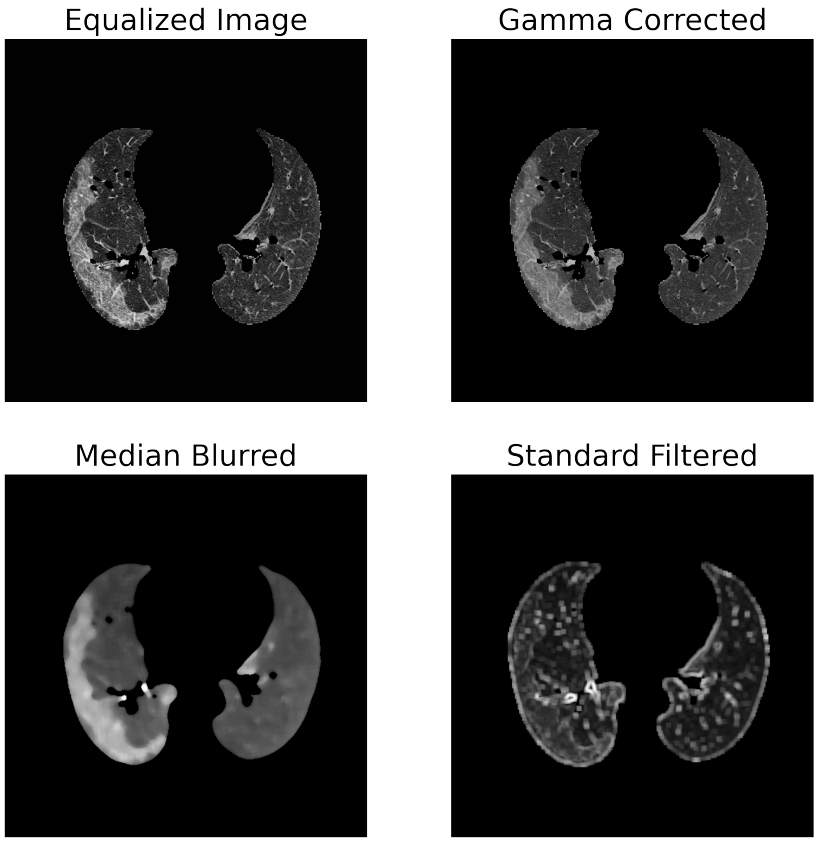



**Figure 2.6:** *Channels of the image. From left to right and from top to bottom the image after the histogram equalization, the gamma correction, the median blurring and the std filtering. These channels allow us to consider information about single voxel, neighbouring voxels and their variability. The histogram eqaulization is applied over small* $10 \times 10$ *areas to avoid over-amplification of the contrast. The gamma correction was performed using* $\gamma = 1.5$*, and the median and std filter kernel sizes are respectively* 11 *and* 3*. Each image is normalized according to mean and standard deviation of the whole scan.*

## Training

This step involves the estimation of the centroid set. To achieve this purpose, a k-means clustering was performed on a training set made on multichannel images from many CT scans. During this I have to takes into account that k-means clustering requires a balance in the different cluster representation. As we can see in Figure 2.7, the main part of the image is composed to background, that si an over-represented cluster. Moreover we can observe that the amount of involved lung volume may change according to the severity of the disease: the first scan presents a low amount pf GGO and CS; se second one have a larger region involved.

To overcome these issues, I have simply removed the background from the segmentation and carefully selected the scans of the training set to ensure a balanced representation of each cluster. The main rule used for the selection is that in each scan must be present a large lesion areas and also a well representation of artifacts, in order to takes into account all the possible features.

The images of the training set are then shuffled and splitted into many subsam-

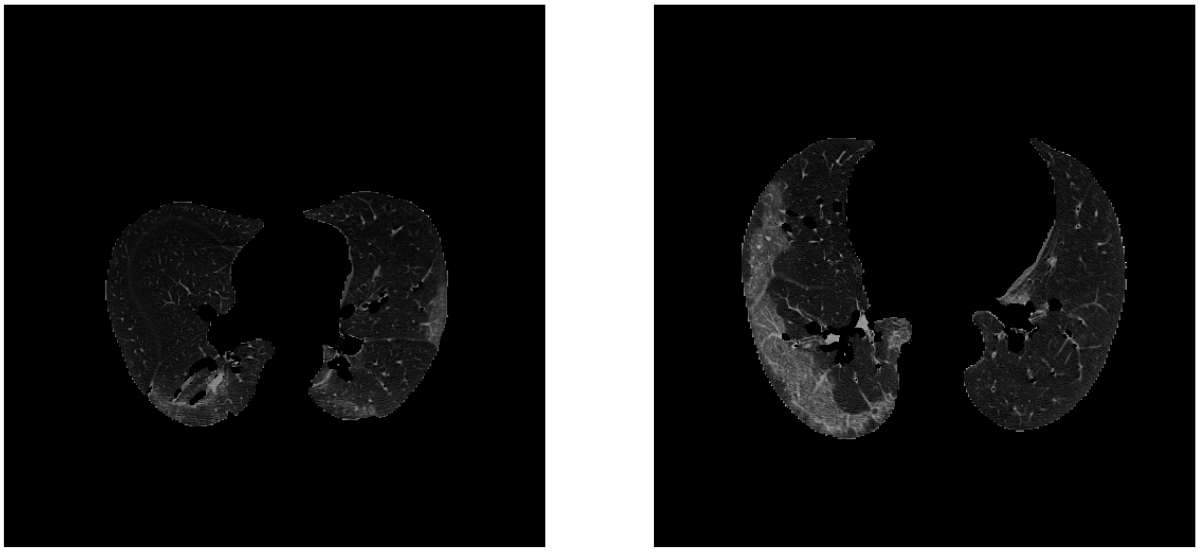


**Figure 2.7:** *Lung regions with different ground glass areas. We can notice that the main part of the image is composed only by the background. From left to right respectively a scan with small lesion areas and one with a large lesion.*

ples and the clustering is performed independently on each one of them. As a final centroid set I have consider the one that minimize the intra cluster variance. This step is justified since the results of the k-means clustering depends on the initial choice centroids. During the minimization of the potential function the algorithm may find a local minimum rather than the global one. The different clustering allows to estimate more than one possible solution and to chose the best one. Moreover the creation of several sub-samples is made since the creation of a single, large array with several images is not always possible, since requires a huge quantity of memory to be allocated.

## Labeling

This step requires as input the multichannel image and a pre-estimated centroid set. It involves the assigning of each voxel to the cluster corresponding to the nearest centroid and the selections of the one corresponding to GGO and CS areas. In this way we are performing a pixel classification by assigning the regions to a particular labels according only to their intensities information: this allow us to group on the same cluster objects that are spatially disconnected as often happen in the medical imaging field.

The distance between voxel colors and each centroid is defined as the euclidean distance:

$$d(x_j, c_i) = \sqrt{(x_j - c_i)^2}$$

Where $x_j$ is the color vector for the $j_{th}$ voxel and $c_i$ is the $i_{th}$ centroid.

## 2.2 Implementation

I've implemented the pipeline described before using python, which is an high level object oriented programming language. To perform all necessary operations (image filtering, input/output managing, etc.) I've used three libraries: OpenCV [18], SimpleITK [19] and Numpy [20].

The whole code is open source and available on GitHub [21] and the pipeline installation is automatically tested on both Windows and Linux by using AppveyorCI and TravisCI. The installation is managed by setup.py, which provides also the full list of dependencies. The code documentation was generated by using sphinx and its available online (`https://covid-19-ggo-segmentation.readthedocs.io/en/latest/?badge=latest`).

The pipeline provides 3 scripts to perform the lung extraction, the training and the labeling. Moreover a default centroid set is provided. The provided scripts allowsto to perform lung segmentation and labeling on a single scan. Till now I've used the (silent) hypothesis that the segmentation involves only one scan. In order to automatize the segmentation on several CT scans, powershell and bash script are provided.

An important point is the input and output managing. The pipeline operations involves only the voxel array, but the input image formats provided also additional information like voxel spacing, image origin and direction. These information must be preserved and incorporated in the output, since are necessary to preserve the compatibility with medical image processing software. To achieve this purpose I have implemented input and output method based on the implemented in SimpleITK. The input method allows to read an image in the medical image format and get the voxel array and the spatial information. The output one allows tosave the image in medical image format by setting also voxel array and spatial information.

The construction of the multi-channel image is embedded both in training and labeling script. This step requires the application of 4 filters on the image and to stack the together the results. For the image filtering I've used the corresponding function implemented in OpenCV. These functions can be applied only on a single image; so I have applied them slice by slice along the axial direction. In this way each images is processed independently from the others: No 3D structure is exploited. To stack the images I've simply used the Numpy stack method. Notice that the OpenCV functions works only with 8−bit gray scale images. The input image must be converted from HU to GL image. This step is performed at the end of the lung extraction.

### 2.2.1 Lung Extraction

This script aims to achieve the HU registration, the lung segmentation and the bronchial removal. It takes as input the CT scan in each format supported by SimpleITK. As output will returns the stack after the lung extraction in *nifti* format. Its workflow is summarized in1.

The lung mask was created by applying the a pre trained U-Net by using the suitable method from [17].

In order to achieve th registration, first of all the values less than the air value are setted to this value, after that the air value is subtracted from all the voxels. These

**Algorithm 1:** Lung Extraction

**Data:** Volume (CT scan)
**Result:** Volume with extracted lung

mask←ApplyUNetVolume(Volume)
volume←(volume $<$ 1200) =air_hu
volume←ShiftMinimumTo1(volume)
/* Start the bronchial removal */

eigen← maxEigenvalues(lung)
bronchial_mask← (eigen $<$ threshold)
lung_woBronchi← (lung $\circ$ bronchial_mask)
/* Refine the bronchial mask */

eigen←maxEigenvalues(lung_woBronchi)
bronchial_mask←(eigen $<$ threshold)
lung_woBronchi← (lung_woBronchi $\circ$ bronchial_mask)

operation are performed on the image tensor by using NUMPY ndarray method. After each value is rescaled in $[0, 255]$.

Listing 2.1: HU registering function

```
1
2  import numpy as np
3
4  def shif_and_crop(tensor) :
5
6    tensor[tensor <  -1200] = tenor[tensor > -1200].min()
7    tensor = tensor - tensor.min()
8
9    return tensor
10
```

More inreresting is the estimation of the eignevalues map. To achieve this purpose I've implemented a function based on CV2.CORNEREIGENVALSANDVEC in OPENCV. This function takes as input an 8−bit gray scale image, so we have to rescale the GL value of the obtained image tensor.

In the end the computing of the eigenvalues map is made by the following function:

Listing 2.2: HU registration function

```
1
2  import cv2
3  import numpy as np
4  from functools import patial
5
6  def max_eigenvalues_map(tensor, block_size, kernel_size) :
7
8    func = partial(cv2.cornerEigenValsAndVecs,
9            blockSize = block_size,
10             ksize = kernel_size)
11   res = np.array(list(zip(*list(map(func, tensor)))))
```

```
12      res = res.transpose(1, 0, 2, 3)
13      res = res[:, :, :, :2]
14      max_eigenvals = np.max(res, axis = 3)
15
16      return max_eigenvals
17
18
```

Once the required parameter were setted, the OpenCV function is iterate over the axial slices. The following transposition and selection of tensor elements, are needed since the function returns also the eigenvectors, in which we are not interested in.

At the end of the whole procedure, the resulting tensor is saved into a medical image formats.

### 2.2.2 Training

This script allows to estimate the set of centroids. It requires as input a path to the training dataset. Each training scan must be saved in a format supported by SimpleITK. This script will returns as output the set of centroids in *.npy* format.

This script will read the training dataset, construct the multichannel image, create the subsamples and cluster them independently. Notice that during the clustering process the background must be removed. These script is summarized in 2.

**Algorithm 2:** training

**Data:** CT scans with Extracted lung
**Result:** Centroid matrix

**foreach** *scan* ∈ *input_scans* **do**
    read the scan
    sample←image_array
**end**

/* prepare subsamples */
sample← build_multichannel(sample)
sample←shuffle(sample)
subsamples←split(sample)
/* start the first clustering */
**foreach** *Sub* ∈ *subsamples* **do**
    center←kmeans(sub, number of centroids)
    centroid_vector, internal_variance←append(center)
**end**

/* Refinement */
centroid_matrix←centroid_vactor(min(*internal_variance*))

The clustering process is managed by the KMEANS_ON_SUBSAMPLES fucntions, implemented as follows.

Listing 2.3: kmenas_on_subsamples

```
1
2  import cv2
3  import numpy as np
4  from tqdm import tqdm
5
6  def kmeans_on_subsamples(imgs,  n_centroids, stopping_criteria,
7               centr_init, weight = None) :
8
9    if weight is not None :
10     vector = np.asarray([ el[w != 0]
11               for el, w in zip(imgs, weight)],
12               dtype = np.ndarray)
13   else :
14     vector = np.asarray([el.reshape((-1, el[-1]))
15                for el in imgs],
16                dtype=np.ndarray)
17
18   centroids = []
19   ret = []
20   for el in tqdm(vector) :
21
22     r, _, centr = cv2.kmeans(el.astype(np.float32), n_centroids,
23                 None, stopping_criteria,
24                 10, centr_init)
25     centroids.append(centr)
26     ret.append(r)
27
28   final_center = centroids[np.argmin(ret)]
29
30   return np.asarray(final_center, dtype= np.float32)
31
32
```

This function takes as input the array containing the subsamples, and apply kmeans clustering on each of them, storing all the values. To remove the background it use a weight tensor. Te values of this tensor must be 0 for the voxel corresponding to the background, and 1 otherwise. At the end of the procedure, the centroid set which provide the minum value of the sum of square error is selected.

### 2.2.3 Labeling

This is the last step of the pipeline and involves assigning of each voxels to the cluster corresponding to the nearest centroids, in this way an hard segmentation is achieved.

The script takes as input the CT scan after the lung extraction and it build the multichannel image as described before. After that it assigns each voxel to the cluster of nearest centroids, which is the one that minimize the distance :

$$cluster = \arg\min_{S} \sum_{i=1}^{k} \sum_{S} \|x - \mu_i\| \tag{2.2}$$

where $x$ is the color vector of the voxel and $\mu$ is the $ith$ centroid. During this process the background is automatically assigned to the 0 label, by passing a mask

which assume 0 on the volxel background and 1 for the other one. At the end of the assignment, only the cluster corresponding to GGO and CS is selected. To summarize the process, the pseudocode of the script is reported in algorithm 3 I've tested this algorithm on three different dataset, the results are described in the next chapter.

**Algorithm 3:** Pseudo-code for the labeling script

**Data:** CT scan to label, centroids
**Result:** GGO label
image←build_multi_channel
/* Compute distances and found the minimum */
**foreach** $c \in centroids$ **do**
  distances← $\|image - c\|^2$
**end**
labels← arg min $(distances)$

The assignement process is performed by the IMLABELING function, which takes care to assign the background to 0, if the suitable parameter is passed. The function is implemented as follows:

Listing 2.4: imlabeling

```
1
2  import numpy as np
3
4  def imlabeling(image, centroids, weight = None) :
5
6
7    if weight  is not None :
8      distances = np.asarray([np.linalg.norm(image[weight != 0] -c,
     axis = 1)
9                  for c in centroids])
10
11     weight[weight != 0] = np.argmin(distances, axis = 0)
12     return weight
13   else :
14     distances = np.asarray([np.linalg.norm(image -c, axis = 3)
15                 for c in centroids])
16     labels = np.argmin(distances, axis = 0)
17     return labels
18
19
20
```

## 2.3 Optimization

During each step of the pipeline we have to set different parameters, like the kernel size for median and std filter, as well as the number of centroids to use fro the segmentation. In this section I will birefly describe how each one of these parameters was optimized, in order to obtain the best segmentation performances.

### 2.3.1 Number of Clusters

The designed algorithm for the centroids estimation is the k-means clustering that requires a prior knowledge about the number of clusters to use. This is very important since a bad choice will badly affect the whole segmentation results. In order to chose the proper number of clusters, I've consider two different sources of information: the anatomical knowledge about the lung and the internal variability of the lung.

From anatomical knowledge about the lung, we can derive 5 clusters, corresponding to:

- Lung Parenchima;
- Edges;
- Vessel surrounding bronchial structures;
- Bronchi.
- Ground Glass Opacities and Consolidation;

Notice that the background of the image is not considered as a cluster since it is removed from the segmentation for the reasons explained before. In order to verify that this number of clusters is the best one, I have considered the internal cluster variability.

Clustering techniques try to group the data into different clusters in order to maximize the difference between points in different clusters and to maximize the similarity within each cluster. If the number of centroids is less then the clusters one, the similarity within each cluster is low. Increasing the number of centroids, will reduce reduce the internal variability till 0 (if number of clusters is equal to the number of points). This means that after a a certain point the diminishing of the internal variability is no more significant, since do not correspond to the good number of clusters but only to the increasing of their number.

To found the correct number of clusters we are seek to for a number of clusters which still provides a small amount of internal variability.

To achieve this purpose, the clustering was repeated several times increasing the number of clusters and for each iteration the internal variability was measured by the sum of squares estimate error (SSE) :

$$SSE = \sum (x_i - c_j)^2 \tag{2.3}$$

Once this task is completed, the results was printed in Figure 2.8.

The optimal number of cluster is the one that corresponds to the elbow of the curve. It is difficult to find this feature from visually, I took the numerical value of the elbow considering the point which maximize the distance between the right joining the first and the last point. From this analysis I have found that the optimal number of clusters is 5, which corresponds to the same that I have found considering the lung anatomy.

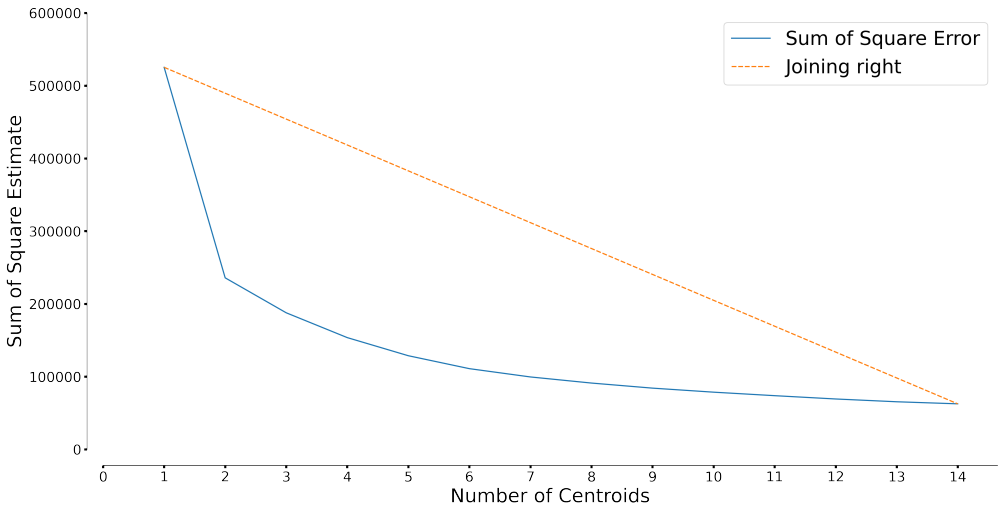



**Figure 2.8:** *Intra-cluster variance vs the number fo clusters(blue); right joining firts and last point(orange). We can se the elbow shape of the graph; the elbow location in the one which maximize the distance between the right joining line and the elbow curve.*

### 2.3.2 Kernel Size

During the building of the multi channel image, I had to compute different image features, that requires the setting of different parameters, like median or standard filter kernel sizes. In order to achieve the best segmentation, I have performed an optimization step that aims to find the parameters that allows to obtain the best segmentation.

Notice that this process is not necessary and a good segmentation can be achieved also by setting this parameters manually.

In order to perform the optimization, I have used SCIKIT-OPTIMIZE [22], more specifically the GC_MINIMIZE METHOD.

This method seeks to perform a Bayesian optimization by using a Gaussian process to approximate an objective function. The function values are assumed to follow a multivariate Gaussian. The covariance of the function values are given by a GP kernel between the parameters. Then a smart choice to choose the next parameter to evaluate can be made by the acquisition function over the Gaussian prior which is much quicker to evaluate [22].

In the end an objective function is minimized. I have used the objective function defined in 4 in which is also reported the whole minimization pseudocode. The used function aim to minimize $1 - IoU$ where the IoU(Intersection over Union) is computed between the output labels and a reference one :

$$IoU = \frac{Area\,of\,Overlap}{Area\,of\,Union}$$

I have decided to use the IoU instead of accuracy since the number of pixels concerning the labeled object would be very few against the number of pixels related to the background. Thus, the label would be a matrix with a large amount of zeros (background) and only few ones (object). In this case the standard metric functions have to consider an unbalanced number of samples; so the solution was to use the

IoU which measures the ration between the Intersection and Union of the output labels and the binary ground truth [15]:

**Algorithm 4:** Parameter Optimization Algorithm

**Data:** Test scans, Ground Truth
**Function** `objective`($parameters, ref\_labels, CT\_scan$)`:`
    $labels \leftarrow$ segment($CT\_scan$)
    iou = IoU($labels$, $ref\_labels$)
    **return** $1 - iou$
$best_parameters \leftarrow$gc_minimize(objective, n_calls, n_random_init)

This process allows to optimize the parameters in order to obtain the better results. As reference labels I have used the ones evaluated as gold standard from five experts radiologists at least 2 years of experience.

This procedure allows only to tune the parameters for a better segmentation, but the learning process remains unsupervised.

# Chapter 3

# Results

The developed pipeline was trained and tested on three datasets (Sant'Orsola, ZENODO and MOSMED), described in the first section. The main method for the evaluation were a quantitative comparison whit a gold standard segmentation, made and validate by 5 experts with at least 2 years of experience. In collaboration with Sant'Orsola, the pipeline results were compared with some semiautomatic segmentation by a blind evaluation made by experts. As a control also segmentation on healthy patient were made, in order to ensure that no lesion were detected. This kind of segmentation was useful also to find the main source of errors.

## 3.1 DataSet Description

This section is dedicated to the description of the datasets used for the developing and test of the pipeline. The description includes general image characteristics and some metadata. If within the datasets are provided also some manual annotation, also the segmentation masks are described.

### 3.1.1 Sant'Orsola

Sant'Orsola data was the ones mainly considered in this work. It consist into 83 anonimyzed CT scans from 83 different patients affected by COVID-19 and 8 scans from healthy controls. Within these scans, also manual annotation were provided. These annotations were obtained with a semi-automatic approach. The built of each annotation may requires several hours.

The series are distributed as follows:

| Property | Value |
|---|---|
| Number of Scans | 83 |
| Distribution by sex(M/F/O) | 66.3/33.7/0 |
| Distribution by age(min/median/max) | 35/60/89 |

For 5 scans were provided also other semi-automatic segmentations. These segmentation were obtained by refining the initial segmentation of a certified software. The building of these labels has required several days. In the end this results are validated by 5 experts with at least 2 years of experience. These 5 segmentation represents the gold standard used as ground truth.

### 3.1.2 MOSMED

MosMed is a dataset which contains 1110 anonymized CT scan of human lung from both patients affected by COVID-19 in several stages fo the disease, and healthy controls. A small subset of this scans is labeled. The scans are obtained between 1st March and 25th of april 2020 by different Russian hospitals. This dataset was born with educational and AI developing purposes. The studies are divided into 5 cathegories, from healty patients to the most severe cases. Each scan of the dataset is saved in *.nfti* format and during the conversion from the original dicom series only 1 image every 10 was preserved. The resulting dataset have the following characteristics:

| **Property** | **value** |
|---|---|
| Number of Scans | 1110 |
| Distribution by sex(M/F/O) | 42/56/2 |
| Distribution by age(min/median/max) | 18/47/87 |
| Number of studies in each cathegory | 254/648/125/45/2 |

The CT scans are organized into 5 cathegories, depending on the percentage of the involved lung volume :

| **Class** | **Description** |
|---|---|
| CT-0 | Normal lung tissues |
| CT-1 | presence of GGO, lung parenchima involved less than 25% |
| CT-2 | GGO, involvement of lung parenchima in $25 - 50\%$ |
| CT-3 | GGO and consolidation, involvement of lung parenchima in $50 - 75\%$ |
| CT-4 | GGO, consolidation and reticular changes, lung parenchima involved more than 75% |

Of these five cathegories only 50 annotations are available, mostly involving only the patients of CT-1 groups. Scans have been annotated by the experts of Research and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Health Care Department.

### 3.1.3 ZENODO

This dataset consists into 20 CT scans of patients affected by COVID-19, labeled by two expert radiologists and verified by third one. The anatomic structures labeled are the left and right lung and the infections regions. Each files is in nifti format and no metadata were available.

Unfortunately only an half of the scans are in HU, the remaining are in 8-bit gray scale, which is not suitable to verify the pipeline since requires it as input an image in HU.

## 3.2 Accuracy

In this section I will comapare the pipeline segmentation against the annotations. The pipeline was trained over 10 CT scans, carefully selected from the available datasets. This ensure a balanced cluster representation. Using this set of centroids

I have segmented the CT scans of the 3 available datasets, and match the achieved results whit the annotations, when available.

The automatic pipeline was run on the servers of the Department of Physics and Astronomy (DIFA), and its able to achieve a segmentation in less than 2 minutes.

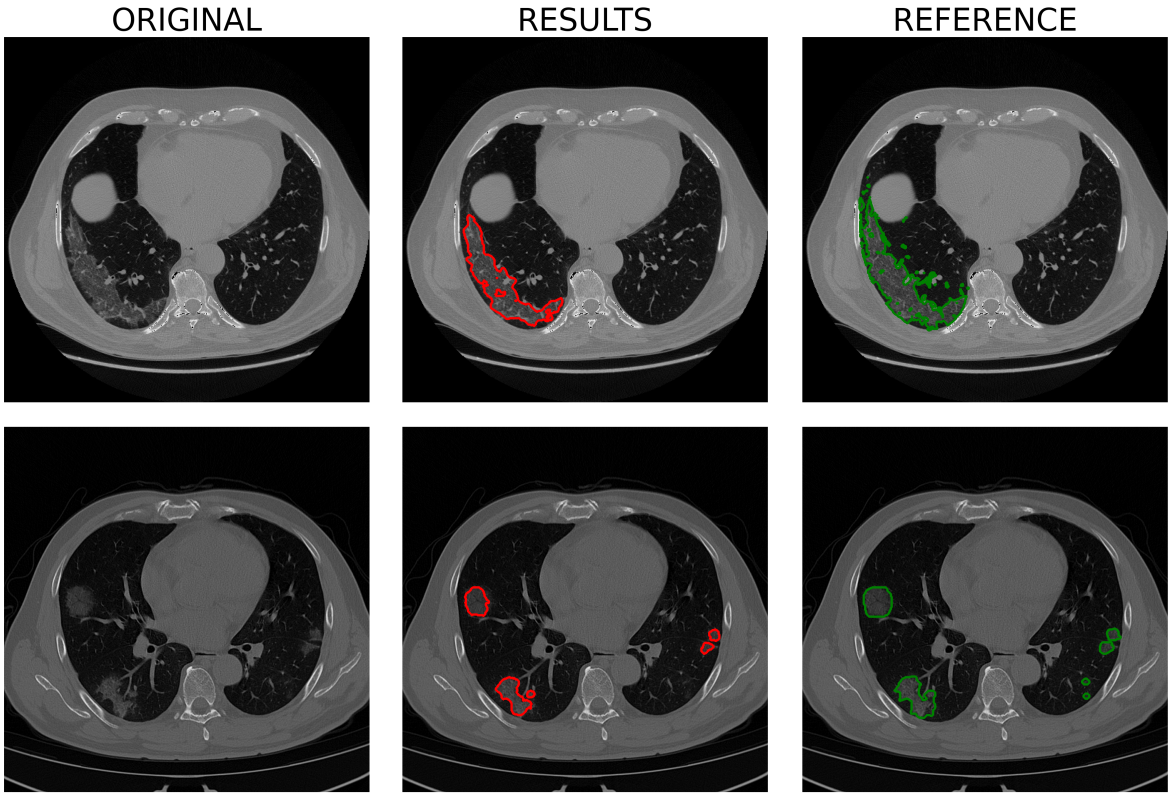



**Figure 3.1:** *Comparison between the achieved segmentation (red) and the reference (green). We clearly see that the GGO ans CS areas are well identified and segmentented*

In Figure 3.1 I've reported a comparison between the achieved results and the reference segmentation. We an observe that GGO and CS areas are correctly segmented.

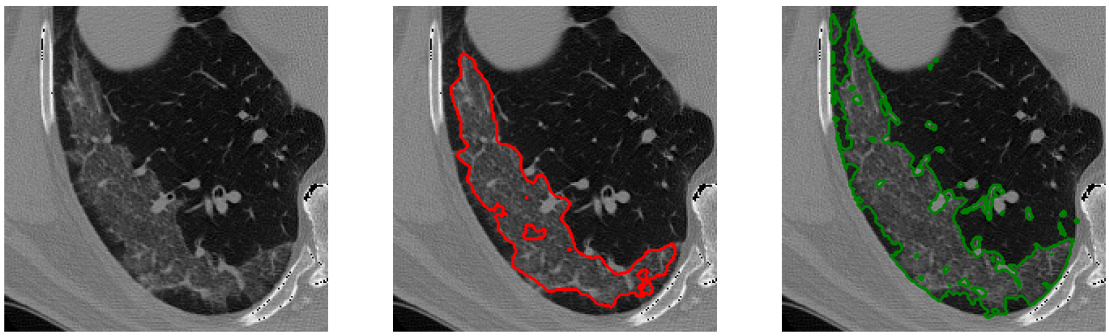


**Figure 3.2:** *Focus on lesion area. From left to right we can observe the original image, the predicted lesion areas and the reference one. We can observe that the annotation identify a larger area. The annotation identify also some spots outside the lesions which seems to be healthy tissue*

In Figure 3.2 I have reported a zoom on the identified lesions areas. As we can see both the pipeline segmentation and the annotations correctly identify the areas of interest. We can observe that in the semi-automatic method there are some spot which do not seem to belong to a lesion area. Moreover the automatic segmentation seems to identify a lesions with less areas.

In Figure 3.3 I've reported a 3D rendering of the lung, with the identified lesions. In pink we can see the manual annotation, in green the segmentation achieved by the pipeline. From these images is more clear that the total estimated volume is different between the two methods. In particular the annotations incorporates a larger volume than the pipeline segmentation.

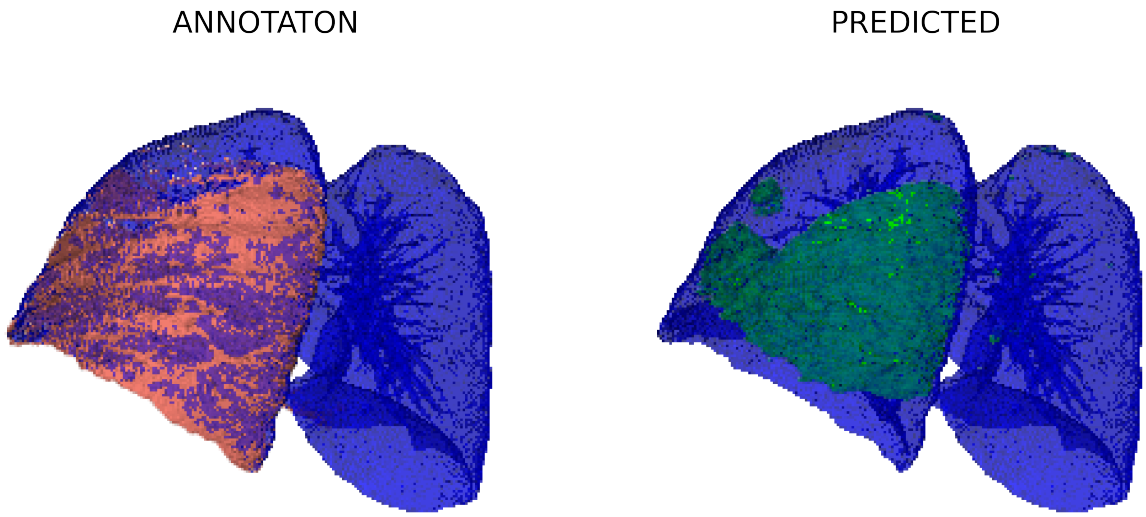



**Figure 3.3:** *3D representation of lesions. From left to right the manual annotation (pink) and the pipeline segmentation (green). We can observe that the segmented size of the area is different.*

In order to assess the quality of the achieved segmentation, I've used two different method.

First of all I've compared both the pipeline segmentation and the annotations with a gold standard. In this way I was able to measure the proportion of the areas correctly identified.

After that the segmentation was submitted to five experts in order to made a blind evaluation, comparing the pipeline segmentation with the reference one.

In the section below I will describe these kind of measures.

### 3.2.1 Comparison with Manual Annotations

In order to obtain quantitative measures about the performances of the developed pipeline, I have compared the results with the manual annotation provided by the Department of Diagnostic and Preventive Medicine of the Poloclinico Sant'Orsola - Malpighi.

The comparison was made using sensitivity and and specificity scores.

**Sensitivity** refers to the ability to correctly detect ill areas. It is defined as the number of voxel classified as lesions among all the lesions. in other world is defined as the number of true positives over the total number of lesions voxels (True POsitives + False negtives) :

$$Sensitivity : \frac{TruePositive}{TruePositive + FalseNegatives} \tag{3.1}$$

**specificity** relates to the ability to correctly reject healthy areas. Is defined as the number of rejected pixels(True Negative) agains the total numner of healthy areas (True Negative + False Positives) :

$$spcificity : \frac{TrueNegative}{TrueNegative + FalsePositives} \tag{3.2}$$

As ground truth I have considered the 5 gold standard segmentations.
The results of these measures are displayed in table 3.1

| | Predicted | | Annotation | |
|---|---|---|---|---|
| | Sensitivity | Specificity | Sensitivity | Specificity |
| Patient 1 | 0.412 | $\sim 1.00$ | 0.676 | 0.999 |
| Patient 2 | 0.399 | $\sim 1.00$ | 0.698 | 0.995 |
| Patient 3 | 0.570 | $\sim 1.00$ | 0.653 | 0.999 |
| Patient 4 | 0.512 | $\sim 1.00$ | 0.325 | 0.999 |
| Patient 5 | 0.628 | $\sim 1.00$ | 0.974 | 0.999 |

**Table 3.1:** *Sensitivity and Specificity for the pipeline segmentation and annotation. As a ground truth was used a sem-automatic segmentation made and evaluated by* 5 *experts with at least* 2 *years of experience.*

The first thing we can notice is that both segmentation and annotation method have an high specificity. So are both able to correctly reject healthy pixels. Moreover we can see from the sensitivity that the annotation is still better to recognize GGO and CS areas. However, in case of annotation I have to point out that this values are subjected to the operator experience. The automatic method is not, since does not requares any trained operator.

In Figure 3.4, we can see a comparison between the ground truth (blue), the predicted segmentation (pink) and the annotation (yellow). As we can see the GGO and CS areas are correctly classified. We can notice that no healthy area is misclassified as lesion. e can also see that, even if the pipleline identifies a smaller lesion area, the segmentation results are in agreement with ground truth.

I have provide an other comparison in Figure 3.5. In this case we can see that the identified area is smaller for the annotation.

### 3.2.2 Expert Evaluation

An other measure of the quality of the segmentation used in this work has been blind evaluation. Five expert with at least 2 years of experience have compared the pipeline segmentation and the reference labels provided within the dataset (obtained by semiautomatic technique).

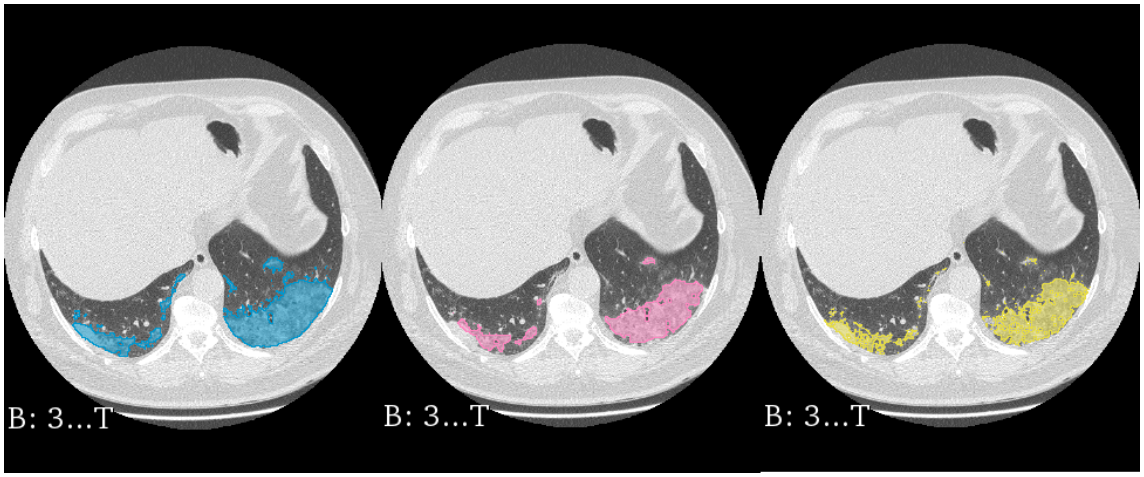



**Figure 3.4:** *Comparison between the ground truth (blue), the pipeline segmentation (pink) and the manual annotation (yellow). We can see that the GGO an CS areas are correctly identified.*

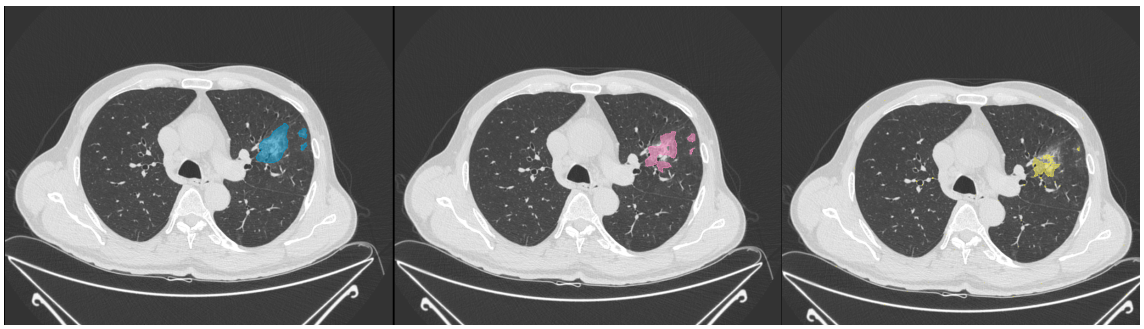


**Figure 3.5:** *Comparison between the ground truth (blue), the pipeline segmentation (pink) and the manual annotation (yellow). We can see that the segmentation obtained by the automatic pipeline is better that the one of the manual annotation.*

To perform the evaluation we have randomly select 40 patients from the three datasets and organized the scans as follows: for each patient we have displayed, slice by slice, two images: the scans with the pipeline segmentation (the one to test), and the semi-automatic labels provided within the dataset (control) as in Figure 3.6.

The scans, organized in this way, was evaluated by experts, to which has been asked to decide, slice by slice, which segmentation is better or if the quality is equal.

The core of this method is that the expert don't know which scan corresponds to a specific segmentation technique, so this will lead to an ideally unbiased evaluation. In order to ensure that, the order of segmentation was shuffled between patients.

### 3.2.3 Healty Control

In this subsection I will discuss the results of the segmentation made on 8 healthy controls scans. This kind of test was performed as benchmark, in order to ensure that, in absence of lesion, no areas were detected. On the other hand, this kind of segmentation is useful since allows to find the main sources of misclassified areas, which we will see are represented by motion artifacts

In Figure 3.7 I have displayed two segmented scans from healthy control. We can

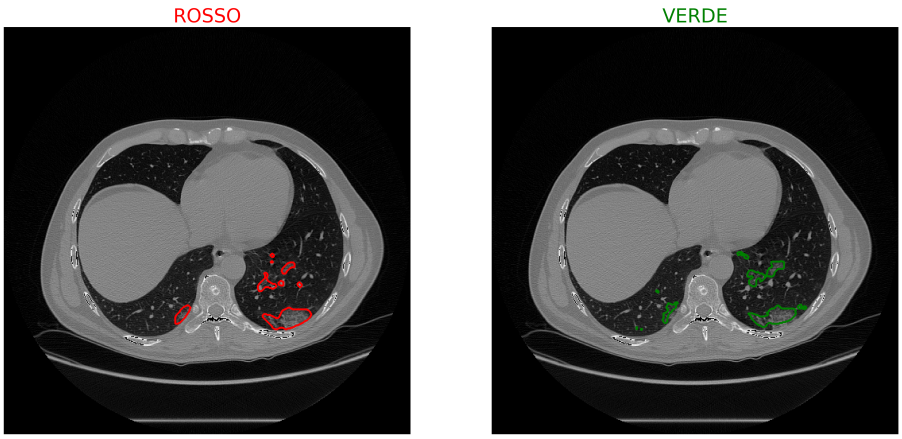



**Figure 3.6:** *Example of comparison submitted to the experts. The two images report the segmentation of the same slices achieved with two different techniques: Manual and pipeline. Was asked to the expert to decide which one is better, without knowing which technique is used to obtain the segmentation.*

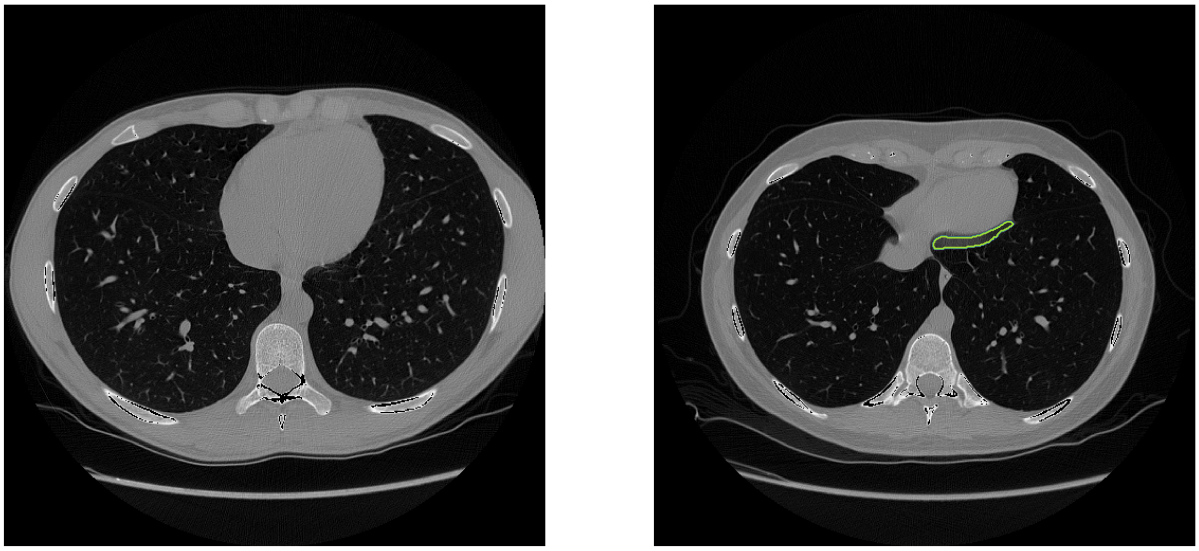


**Figure 3.7:** *Segmented CT scans from healthy controls: Without motion artifacts(left), with motion artifacts(right). In the first case we can see that no area is identified since no lesion is present; In second one we can see how, in presence of motion artifacts, some false positives may raise.*

see that, in a normal case, no lesion areas were detected; but in presence of artifacts, some misclassified regions appears.

This can be clearly seen by considering the percentage of misclassified areas. In normal case, the percentage of misclassified voxels is less than 0.01%; but, in presence of artifact, increase of 10 times.

This result suggests that the main source of false positives is the presence of motion artifacts, caused by heartbeat and respiratory cycle. However we have seen that these doesn't affect too much the specificity of the method. Further improvements may seek to remove these regions, in order to achieve a more accurate segmentation.

In the end we have verified that, in the most common case, no lesion areas were identified as we expect. On the other hand, in presence of motion artifacts, these

may be misclassified as lesions and we have observed that this is one of the main source of false positives. However the specificity of the pipeline remains very high.

# Conclusions

In this work I've developed a fully automated pipeline for the identification of lesions (GGO and CS) in chest CT scans of patient affected by COVID-19. The whole pipeline achieves the segmentation by using the color quantization technique, that allows to exploit the color similarity between voxels belonging from the same tissue. The multichannel nature of digital images has allowed also to consider other properties besides the single voxel intensity: also neighboring voxel information are takes into account.

As a preliminary step, a lung segmentation is performed by using a pre trained U-Net, followed by a bronchial removal, which aims to reduce the rate of false positives. To achieve the segmentation, we have found the characteristic color of each tissue by performing a k-means clustering in the color space. The actual segmentation is achieved by assigning each voxel to the cluster corresponding to the nearest color.

The pipeline, tested on three different datasets, has provided good segmentations for the typical lesion cases with an accurate identification of GGO and CS. The proposed algorithm has also shown some limitations due to the eventual presence of motion artifacts, caused by heartbeat and respiratory cycle; this behaviour was observed also on segmentation made on healthy controls, which have shown some misclassified points in these areas.

In the end we have developed a fully automated pipeline which achieve a good segmentation for the most typical cases.

Color quantization has shown to be a suitable approach to face this kind of problems, and some improvements, like removal of motion artifacts are still possible.

# Bibliography


[1] Zhao Fu and et al. "CT features of COVID-19 patients with two consecutive negative RT-PCR tests after treatment." In: *Scientific reports* (July 2020). DOI: 10.1038/s41598-020-68509-x.

[2] Huang C, Wang Y, and et al. "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China". In: *Lancet* (Feb. 2020). DOI: 10.1016/S0140-6736(20)30183-5.

[3] Adam Bernheim et al. "Chest CT Findings in Coronavirus Disease-19 (COVID-19): Relationship to Duration of Infection". In: *Radiology* 295.3 (2020). PMID: 32077789, p. 200463. DOI: 10.1148/radiol.2020200463. eprint: https://doi.org/10.1148/radiol.2020200463. URL: https://doi.org/10.1148/radiol.2020200463.

[4] Tao Ai et al. "Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases". In: *Radiology* 296.2 (2020). PMID: 32101510, E32–E40. DOI: 10.1148/radiol.2020200642. eprint: https://doi.org/10.1148/radiol.2020200642. URL: https://doi.org/10.1148/radiol.2020200642.

[5] Collins J Stern E J. "Ground-glass opacity at CT: the ABCs". In: *American Journal of Roentgenology* 169 (1997), pp. 355–366. DOI: doi:10.2214/ajr.169.2.9242736.

[6] Ma Jun et al. *COVID-19 CT Lung and Infection Segmentation Dataset*. Version Verson 1.0. Zenodo, Apr. 2020. DOI: 10.5281/zenodo.3757476. URL: https://doi.org/10.5281/zenodo.3757476.

[7] N.A. Vladzymyrskyy A.V. Ledikhova N.V. Gombolevskiy V.A. Blokhin I.A. Gelezhe P.B. Gonchar A.V. Morozov S.P. Andreychenko A.E. Pavlov and Chernina V.Y. *MosMedData: Chest CT Scans With COVID-19 Related Findings*. Version Verson 1.0. 2020. URL: https://mosmed.ai/.

[8] Dzung L. Pham, Chenyang Xu, and Jerry L. Prince. "Current Methods in Medical Image Segmentation". In: *Annual Review of Biomedical Engineering* 2.1 (2000). PMID: 11701515, pp. 315–337. DOI: 10.1146/annurev.bioeng.2.1.315. eprint: https://doi.org/10.1146/annurev.bioeng.2.1.315. URL: https://doi.org/10.1146/annurev.bioeng.2.1.315.

[9] D. Withey and Z. J. Koles. "A Review of Medical Image Segmentation: Methods and Available Software". In: 2008.

[10] Dr J P Chaudhari Pooja V. Supe Prof. K. S. Bhagat. "Image Segmentation and Classification for Medical Image Processing". In: *International Journal on Future Revolution in Computer Science & Communication Engineering* 5.1 (), pp. 45–52.

[11] David Arthur and Sergei Vassilvitskii. "K-Means++: The Advantages of Careful Seeding". In: vol. 8. Jan. 2007, pp. 1027–1035. DOI: 10.1145/1283383.1283494.

[12] Laurence Morissette and Sylvain Chartier. "The k-means clustering technique: General considerations and implementation in Mathematica". In: *Tutorials in Quantitative Methods for Psychology* 9 (Feb. 2013), pp. 15–24. DOI: 10.20982/tqmp.09.1.p015.

[13] Ozturk Celal, Hancer Emrah, and Karaboga Dervis. "Color Image Quantization: A Short Review and an Application with Artificial Bee Colony Algorithm". In: *Informatica* 25.3 (2014), pp. 485–503. ISSN: 0868-4952. DOI: 10.15388/Informatica.2014.25.

[14] Johannes Prayer Forian Pan Jeanny Röhrich Sebastian Prosch Helmut Langs Georg Hofmanninger. "Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem". In: *European Radiology Experimental* 4 (2020). ISSN: 1. DOI: 10.1186/s41747-020-00173-2. URL: https://doi.org/10.1186/s41747-020-00173-2.

[15] Nico Curti. *Implementation and optimization of algorithms in Biomedical Big Data Analytics*. https://nico-curti2.gitbook.io/phd-thesis/. 2019.

[16] J. Austin et al. "Glossary of terms for CT of the lungs: Recommendations of the Nomenclature Committee of the Fleischner Society". In: *Radiology* 200 (Sept. 1996), pp. 327–31. DOI: 10.1148/radiology.200.2.8685321.

[17] Hoa Nguyen Johannes Hofmanninger. *Automated lung segmentation in CT under presence of severe pathologies*. https://github.com/JoHof/lungmask. 2020.

[18] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[19] Bradley Lowekamp et al. "The Design of SimpleITK". In: *Frontiers in Neuroinformatics* 7 (2013), p. 45. ISSN: 1662-5196. DOI: 10.3389/fninf.2013.00045. URL: https://www.frontiersin.org/article/10.3389/fninf.2013.00045.

[20] Travis E Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006.

[21] Biondi Riccardo et al. *COVID-19 Lung Segmentation*. https://github.com/RiccardoBiondi/segmentation. 2020.

[22] Tim Head et al. *scikit-optimize/scikit-optimize*. Version v0.8.1. Sept. 2020. DOI: 10.5281/zenodo.4014775.