

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. This virus affect lung areas and causes respiratory illness.In this scenario is highly desirable a method to identify in CT images the lung injuries caused by COVID-19. The approach proposed here is based on color quantization to identify the infection regions(GGO and Consolidation). We have segmented CT scans belonging from 3 different dataset and match the results with manual segmentation made by experts. The proposed approach aims to segment the scan basing on the voxel and the

Contents

Abstract

Introduction	3
1 Review on Image Segmentation techniques	5
1.1 Medical Images	5
1.2 Review on Image Segmentation Methods	6
1.2.1 Thresholding	7
1.2.2 Region Growing Approach	8
1.2.3 Deformable Model	8
1.2.4 Markov Random Field	8
1.2.5 Classifiers Approach	9
1.2.6 Clustering	9
1.2.7 Color Quantization for Medical Image Segmentation	11
1.2.8 U-Net	12
2 Lung Lesion Segmentation	15
2.1 Pipeline Description	15
2.1.1 Pipeline Structure	16
2.2 Pipeline Implementation	19
2.2.1 Frameworks	20
2.2.2 Lung Extraction	22
2.2.3 Training	25
2.2.4 Labeling	28
2.3 Tuning of Parameters	29
2.3.1 Estimation of the Number of Clusters	29
2.3.2 Kernel Size Optimization	30
3 Results	33
3.1 DataSet Description	33
3.1.1 Sant'Orsola	33
3.1.2 MOSMED	34
3.1.3 ZENODO	34
3.2 Results	34
3.2.1 Visual Comparison	35
3.2.2 Healty Control	36
3.2.3 Gold Standard	36
3.2.4 Expert Evaluation	38

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 [ART:Zhao].

An initial prospective made by Huang and al. on chest CT scans [ART:Huang] 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 [ART:Bernheim]. In the end, other study have monitored the change on volume and shape of these features on healed patients [ART:Ai] in order to monitoring their actual recovery. In Figure 1 are compared slices of an healthy control and a patient affected by COVID-19. We can clearly see the GGO and CS regions in the lung of the second lung image from left.



Figure 1: Ground 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 increasing of the involvement of the involved lung volume.

GGO and CS are not exclusive of COVID-19, but may be also caused by pulmonary edema, bacterial infection, other viral infection or alveolar haemorrhage [ART:Collins]. 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 and subjective, since involves the interaction with trained personnel; so an automatic and fast way for the identification of this features

is desired.

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

We start the discussion by understanding what is CT image, 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 takes 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 describe in details the actual pipeline implementation, going in deep on how each step is actually achieved. Also a brief description about the used frameworks will be given.

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 with a ground truth segmentation and blind evaluation by experts. Also the segmentation achieved on healthy control was considered as benchmark.

Chapter 1

Review on Image Segmentation techniques

Image segmentation consist in the partitionng of an image into non overlapping, consinistent regions that are homogeneous respect to some characteristics such as intensity or texture [ART:Pham]. Nowadays several non-invasive medical imaging techniques are available, such as Computed Tomography(CT), Magnetic Resonance Imaging (MRI) or X-Ray imaging,. that provides a map of the subject anatomy. Image segmentation play a crucial role in many medical-imaging applications by automating or facilitate the delineation of anatomical structures and other regions of interest [ART:Pham]. Manual segmentation is possible, but is time consuming and subject to operator variability; making the results difficult to reproduce [INP:Withey], so automatic or semi-automatic methods are preferable.

A major difficulty of 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 [ART:Pooja].

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, for example, a therapy.

This chapter contains a brief introduction on medical digital images and a brief review on the image segmentation techniques such as clustering or threshold.

1.1 Medical Images

Digital images are represented by 3 dimensional tensor, where the three dimensions corresponds respectively to height, width and number of channel, where channel refers to the number of image components. For instance a Gray level image is represented with a $h \times w$ matrix, on the other hand, an RGB image is composed by three $h \times w$ matrices, each of them represent a different primary. Each value of the tensor is in a range that change according to the image format. The most common are [0, 255] for 8-bit integer image, of [0, 1] for float. However also other formats are available, like 16-bit integers, widely used to represent medical images.

Medical images provides a map of the subject anatomy and so are computed on a uniformly x-y-z spatial space. At each point the data is represented by a 16-bit integer. The meaning of the data change according to the image acquisition modality(CT, MRI, PET, etc.).

In computed tomography(CT), which is a technique that aims to reproduce cross-section images and the 3D anatomy of the examined subject, each data represent the capability of the corresponding volume to attenuate an x-ray beam. In order to be able to match results from different scans the beam attenuation is measured in Hounsfield Units(HU) :

$$CT - number = k \times \frac{\mu - \mu_{H_2O}}{\mu_{H_2O}} \quad (1.1)$$

Where μ is the linear attenuation coefficient of the tissue, μ_{H_2O} 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 .

The resulting image tensor have a 16-bit depth and each voxel value is in relation with the displayed tissue.

1.2 Review on Image Segmentation Methods

During the years, several segmentation methods have been developed based on a lot of different approaches. These methods can be categorized in 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 [ART:Pham] 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 basically segments a scalar image by creating a binary partitioning of image intensities [ART:Pham]. It can be applied on an image to distinguish regions with contrasting intensities and thus differentiate between tissue regions represented within the image [INP:Withey]. Figure 1.1 show an histogram of a scalar image with two classes, threshold based approach attempts to determine an intensity value, called *threshold* which separate the desired classes [ART:Pham]. So to achieve the segmentation we can group all the pixels with intensity higher than the threshold in one class and all the remaining in the other class.

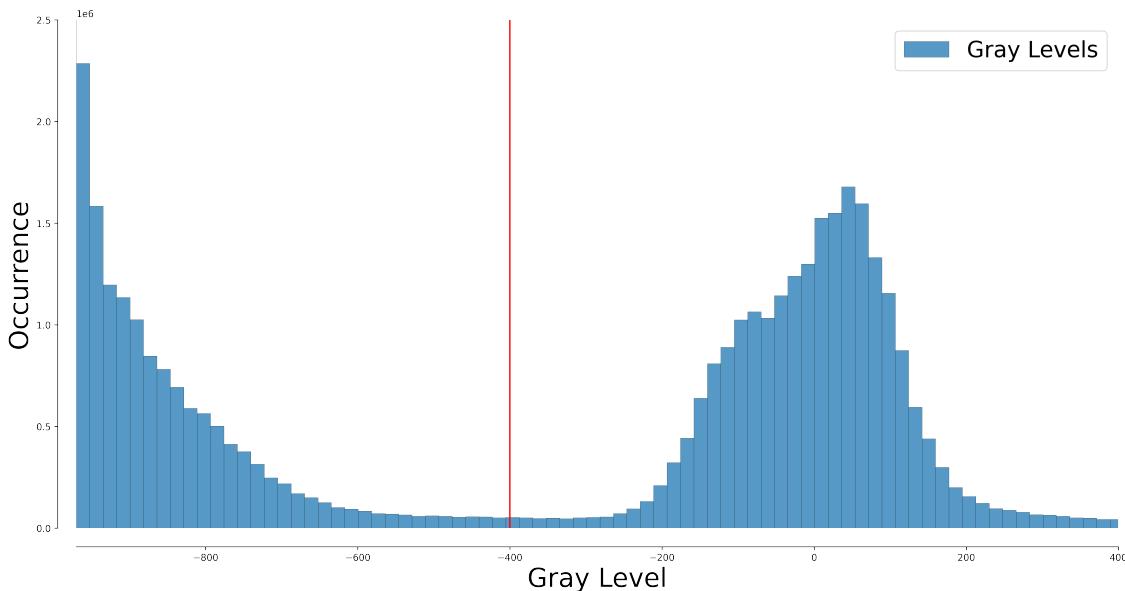


Figure 1.1: Histogram of a GL image with two well delineated regions. The threshold value(red line) was set visually at -400 HU

The threshold value is usually setting by visual assessment, but can also be automatized by algorithm 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 this 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 doesn't takes into account the spatial characteristic of the image, so it is sensitive to noise and intensity inhomogeneity, that corrupt the image histogram of the image and making difficult the separation [ART:Pham]. To overcome these difficulties several variation of threshold have been proposed based on local intensities and connectivity.

Threshold is usually used as initial step in sequence of image processing operations, followed by other segmentation technique that improve the segmentation quality. Since threshold use only intensity information, 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 start at seed location in the image(usually manually selected) and check the adjacent pixels against a predefined homogeneity criterion [**INP:Withey**], based on intensity, and/or edges. If the pixels met the criterion, they are added to the region. A continuos application of the rule allow the region to grow.

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

Regions growing can also be sensitive to noise so extracted regions may have holes or even become disconnected. May also happen that separate rion becomes 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 algorithm, 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 [**INP:Withey**].

1.2.3 Deformable Model

Deformable Model use an artificial, closed, contour/surface able to expand or contract over time and confrom to a specific image feature [**INP:Withey**]. This approach is physically motivated model-based thechnique for the detection of region boundaries [**ART:Pham**].

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 an also incorporate smoothness 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 its a statistical model that's used within segmentation methods that model the spatial interaction between neighbouring pixels. It's often incorporated in clustering algorithms such as K-means with a Bayesian prior probability.

This model is used because most pixels belong to the same class as their neighbouring pixels, this means that any anatomical structure that consist of only one pixel has a very low probability of occourring [**ART:Pham**].

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 requires computationally intensive algorithms. However, despite these disavan-

tages, MRF are widely used to model segmentation classes and intensity inhomogeneities [ART:Pham].

1.2.5 Classifiers Approach

Classifiers approaches use statistical pattern recognition techniques to segment images by using a mixture model that assume each pixels belonging to one of a known set of classes [INP:Withey]. To assign each pixel to the corresponding classes, use the so called *feature space*, which is the space of any function of the image like intensity. An example of 1D feature space is image histogram.

The feature of each pixel form a pattern that is classified by assign a probability measure for the inclusion of each pixel in each class [INP:Withey].

This approach assume a prior knowledge about the total number in the image and the probability of occurrence of each class. Generally this quantity aren't 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 assign each pixel to the class with the highest posterior probability.

This approach requires a structure to segment with distinct and quantifiable features. It is computational efficient and can be applied to multichannel images. This approach doesn't consider a spatial modelling and need 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 classifiers one but in an unsupervised fashion, so doesn't require a training dataset. Clustering iteratively alternate between segmenting the image and characterizing the properties of each class. In this way we can say that clustering approach train itself by using the data available information. We can identify 3 main clustering algorithms:

- **k-means clustering:** that iteratively compute a mean intensity for each class and segments the image by classifying each pixel in the class with the closest mean;
- **Fuzzy C-means:** this algorithm generalize the K-means clustering in order to achieve soft- segmentation;
- **Expectation Maximization:** use the same clustering principle as k-means by assuming that the pixel follows a Gaussian mixture model. It iterates between posterior probability and compute the the Maximum Likelihood estimates for the means, covariances and mixing coefficients of the mixture model.

This approach doesn't require training data, but suffer from high sensitivity to the initial parameters and do not incorporate spatial model, so it is a pixel classification technique [ART:Pham].

The most used algorithm for clustering is the k-means clustering, which seek to assign each point to a particular cluster in a way that minimize the average square distance between points in the same cluster [Arthur2007]. A vector representing the mean is used to describe each cluster, so this technique is described as a centroid model [ART:Morisette]. Each point is assigned to the cluster with the nearest mean.

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

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

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

The steps of the algorithm are the following:

1. Select the value of k as initial centroids
2. Form k cluster by allocating every point to its most nearest centroid
3. Recalculate the centroid for each cluster until the centroid does not change.

Arthur and Vassilvitskii [Arthur2007] have pointed that this algorithm is not accurate and can produce arbitrarily bad clusters. So they have developed a popular algorithm, 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 Φ_{opt} is the potential function in the optimal centroids case. They have proposed a variant for the choosing of the centroids, instead of chose the centroids randomly, the 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 $x \in S$ 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 [Arthur2007]

1.2.7 Color Quantization for Medical Image Segmentation

Color quantization is the process of reducing the number of colors in a digital image. The main objective of quantization process is that significant information should be preserved while reducing the number of colors in an image, in other word quantization process shouldn't cause significant information loss in the image. Color quantization, accepted as a pre-processing application, is used to reduce the number of colors in images with minimum distortion such that the reproduced image should be very close to the original image visually, as in Figure 1.2.

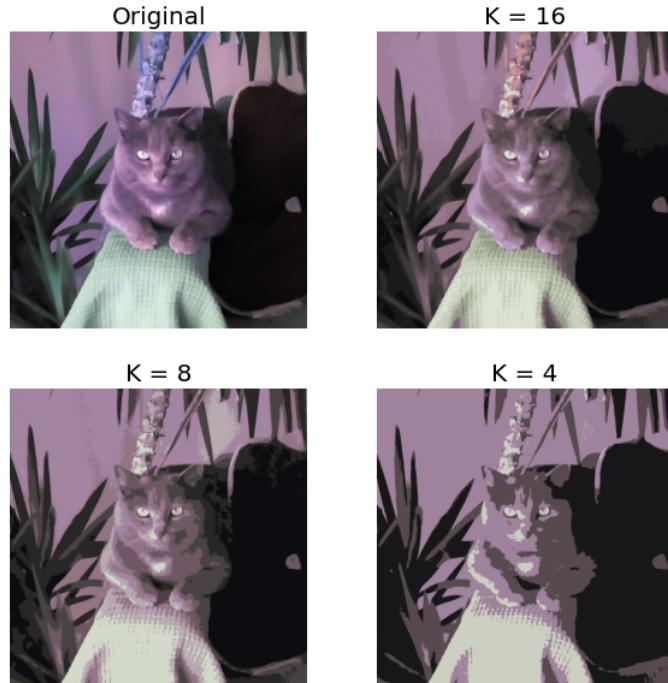


Figure 1.2: Color quantized RGB image. We observe the original image, a 16 color image which look similar to the original one, a 8 colors image and 4 colors image

Color quantization play an important role in many filed of applications such as segmentation, compression, color texture analysis, watermarking, text localization/detection, non photorealistic rendering and content-based retrieval [ART:Ozturk].

Color quantization may be used for image segmentation. Use that for this purpose means to redice the numner of colors to the number of the different objects we seek to segment. This means that is is based on the assumption that to each different object class in the image is assigned an unique characteristic color. This is the case of image segmentation, in which each image color represent a particular characteristic of the tissue displayed(i.e in x-ray represent μ). 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 are formed by using artificial neurons derived from physiological models [INP:Withey]. Neural Networks are made by nodes that simulate a biological learning. Each node of the network it is able to perform an elementary operation. 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 usually is used the so called modelU-Net, which is a kind of convolutional neural network which allows to overcome the two main drawbacks of this kind of networks :

- Needs of a huge size fo training data
- Size of the considered network.

That because, for medical and biological segmentation purposes, training dataset with a huge size are not available.

Convolutional Networks usually are used nn classification tasks, which requires only one label. For 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 [ART:Johannes]. This kind of network is a modification and simplification of a fully convolutional neural network, making it suitable to works with few 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 size and the segmentation mask, which aims to process the extracted features

Decode path tends to lose some of the igher 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 [PhDtheis].

As I've said the network is composed by two path: 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 [ART:Johannes].

In particular the contractive path consist in the application of 3×3 convolutions, each of them followed by a rectified linear unit(ReLU). and a 2×2 pooling operation. very step in the expansive path consists of an upsampling of the feature map followed

by a 2×2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1×1 convolution is used to map each 64-component feature vector to the desired number of classes

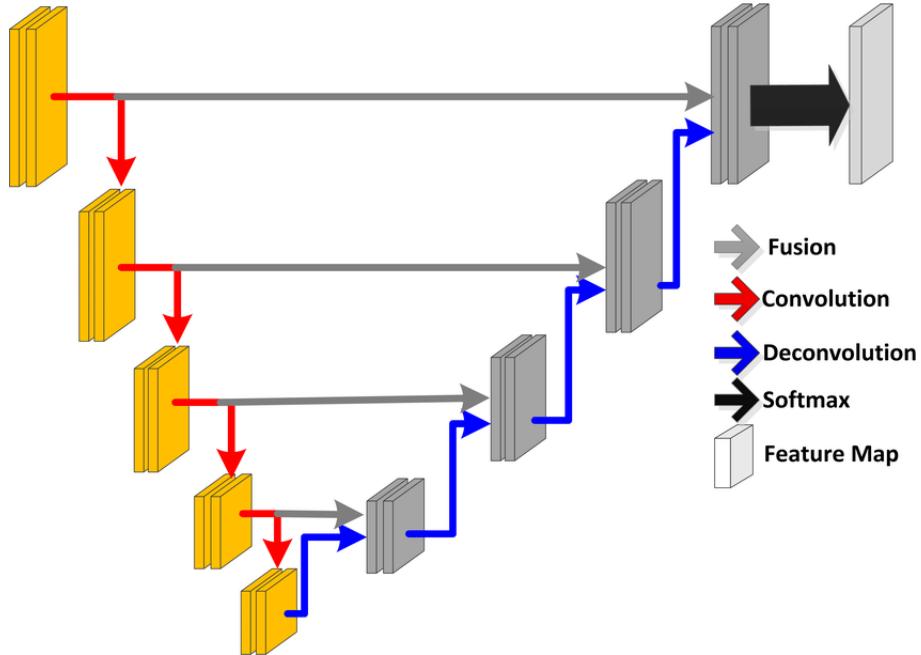


Figure 1.3

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 displayed in Figure 1.3. In a U-Net the segmentation map only contains the pixels for which the full context is available [ART:Johannes].

In order to work with few training data, this network makes a huge data augmentation, by applying an elastic deformation to the training images, that allow the network to learn invariance to such kind of transformation. .MainSegmentation

Chapter 2

Lung Lesion Segmentation

In this chapter I will discuss the pipeline developed for the segmentation of GGO and CS areas in chest CT scans of patients affected by COVID-19.

In the first section I will first start defining GGO and CS and then I will describe how color quantization was used to achieve the segmentation and the main pipeline structure; paying attention on the aims of each block.

In the second section I will go into details on the actual pipeline implementation, describing how each step is achieved. Also a brief description of the used frameworks and tools was provided.

In the end I will describe the routines used for the tuning of parameters, used to achieve a best segmentation performances.

2.1 Pipeline Description

The aim of this thesis work is to develop a pipeline for identification of GGO and CS areas in chest CT scans of patients affected by COVID-19, that aims to have the following characteristics:

- **Fully Automated:** to remove the dependency from an external operator, and so the subjectivity of the segmentation;
- **Fast:** in order to compete with certified software and to provide a segmentation in few minutes.

Before going in deep and starting the description, I will define this kind of regions. Austin in [ART:Austin] has defined the Ground Glass Opacities as an "*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. An example of these areas is displayed in Figure 2.1.

The segmentation is based on an unsupervised technique, so doesn't require to provide the expected outcomes. As we can see in the chest CT scan in Figure 2.1 the different structures are characterized by similar gray level: the basic idea was to use the color quantization for medical image segmentation, grouping voxel based on color similarity, and assign to each tissue a characteristic color. This can be done since in CT scan exist a relationship between the tissue in the voxel and the Gray Level used



Figure 2.1: *Chest CT image of a patient affected by COVID-19. We can see GGO and CS on the right lung. We can appreciate how the different structures, like lesions, are identified by similar color, so the basic segmentation idea was to group voxel by color similarity.*

to display it, given by the Hounsfield Units(eq 1.1), so colors are proportional to HU, which are defined as a linear transformation of the linear attenuation coefficient(μ).

Color quantization and the properties of digital images allows to consider also other properties of the image besides the single voxel intensity: As I've said before, in digital image processing, images are represented with a 3D tensor, in which the first two dimensions represent the height and width of the image and the last one the number of channels. In this work we have used the different channels to takes in account different properties, exploited by the application of suitable filters on the original scan. This allow us to consider also neighboring voxels, that is really suitable for the segmentation since the lesions areas involves many closest voxels, not only a single one.

Once we have build the color space, we have to found the characteristic color of each tissue under study, which is represented by a centroids in the color space. In order to perform this task and achieve the centroids estimation, a simple k-means clustering was used, since it provides a suitable segmentation with good time performances and it is efficiently implemented for multi-channel images in OpenCV [**OpenCV**].

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; so we can consider a different cluster for each anatomical structure. Once we have estimated the centroids for each tissue, we have used them for the actual segmentation, by assign each voxel to the cluster of the closest centroids: in this way the estimation step, that we will call "train", needs to be performed only once, so can be time expansive since is not involved in the actual segmentation.

2.1.1 Pipeline Structure

In this section I will discuss the general structure of the pipeline, more details about the actual implementation will be given in the next chapter. To perform the color quantization I've to found the characteristic color(centroids in the color space) of each tissue and use them for the actual segmentation, dividing the pipeline in two main steps.

Before each of these steps we need a preliminary phase that aim to isolate the lung regions in order to exclude the extra lung areas and reduce the false positives and motion artifacts. In the end the pipeline structure is divided in three main blocks as we can see in Figure 2.2 :

- **Pre-Processing and lung extraction:** Preliminary step, involves registration of HU, isolation of lung regions and removal of bronchial structures and air pockets.
- **Training :** estimation of the centroids, is performed only ones;
- **Labeling :** assignment of each voxel to the cluster of the closest centroids, it is the actual segmentation.

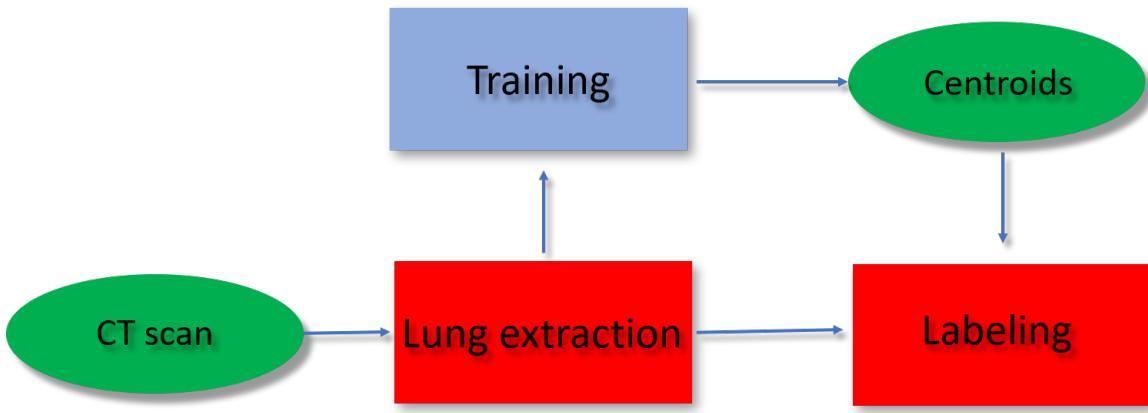


Figure 2.2: Flow chart of the main structure of the developed pipeline. The training process, which allows the estimation of the centroids, is performed only one time.

Pre Processing and 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. Lung segmentation is pivotal pre-processing step in many image analysis such as identification and classification of lung pathologies [ART:Johannes]. The lung isolation allow us to found a mask for the lung regions, excluding so 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 dataset and usually over a limited spectrum of visual variability by containing mainly cases without severe pathologies [ART:Johannes]. Rule based approach, like threshold, region growing, ect, usually fails for CT sans of patients with severe Interstitial Lung Disease(ILD), as we can see in Figure 2.3.

To achieve the lung segmentation I've used pre-trained UNet [ART:Johannes] [REP:lungmask] that provide a good segmentation of the lung. The UNet is a supervised approach

and it is used only on this preliminary step, as we will see the identification of infection regions is made with an unsupervised approach.

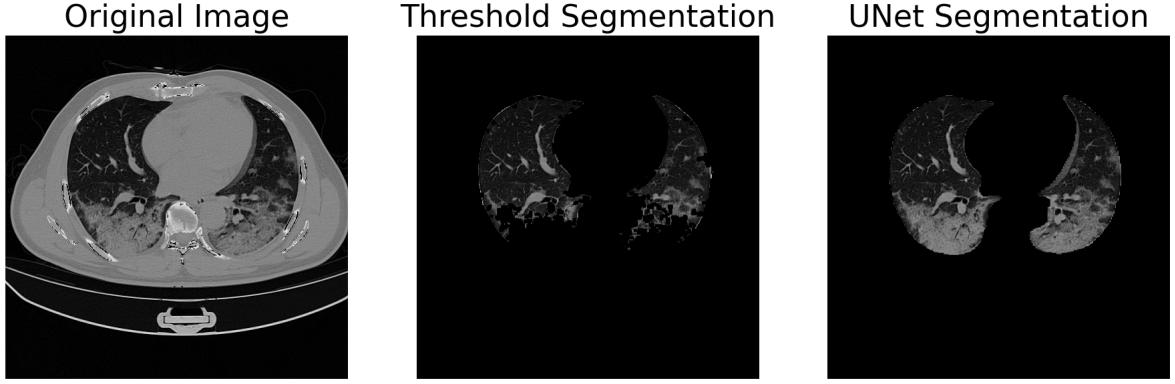


Figure 2.3: From left to right the original CT scan of a patient with severe ILD, the lung segmented by threshold and connected components, and lung segmentation achieved by the pre-trained UNet. We can clearly see the missing areas in the first segmentation, corrected identified in the UNet one.

This kind of segmentation include in the lung region also motion artifacts and bronchial structures, which we will see are the principal cause of false positives. To achieve a better segmentation a refinement process is performed, which aims to remove the main bronchial structures from the selected lung regions. Up to now no motion artifact removal routines is implemented, so we will see that this kind of artifacts will be the main source of misclassified points.

Training

This step involves the estimation of the centroids for each tissue. To achieve this purpose we have chose to perform a clustering by using the k-means algorithm.

We have to takes into account that the k-means clustering requires an homogeneous representation for each cluster. As we will see, the number of voxel representing the background is several order of magnitude higher than that the other; moreover patients with a low involvement of lung volume have the cluster corresponding to the infection underrepresented. To overcome the first issue we have simply removed the background from from the segmentation, instead, to overcome the second one, we have carefully selected the scans of the training set to ensure an homogeneous representation of each cluster.

In summary, the implementation of this step involve the building of the multi-channel image, which allow us to takes into account also the neighborhood information; the managing of the over represented clusters and the actual centroids estimation.

Labeling

This step involves the actual segmentation. The script which perform it requires as inputs the CT scans after the lung extraction, and the previously estimated

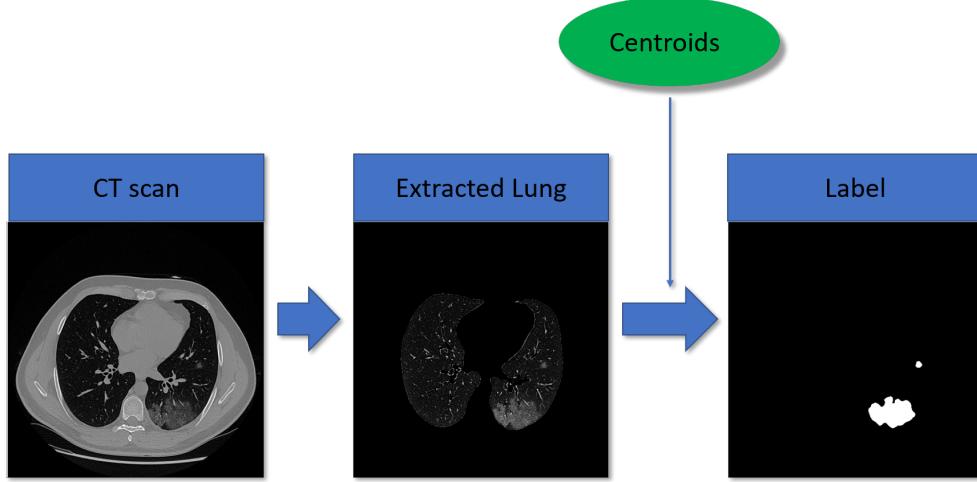


Figure 2.4: Actual segmentation step, from left to right we can see the input image stack, the isolated lung regions and the final label. To performed the labeling a set of pre-computed centroids was used.

centroids. This block of the pipeline simply assign each voxel to the cluster corresponding to the nearest centroids and the select only the one corresponding to GGO and CS. In this way we are performing a pixel classification by assign regions to a particular labels according only to intensities information, without exploiting spatial information: this allow us to group on the same cluster objects that are spatially disconnected as often happen in medical imaging field.

The distance between voxel color and each centroid is defined as 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.

To summarize, once the centroids are estimated, the segmentation pipeline will results in 2 main steps : **lung extraction** and **labeling**, as shown in Figure 2.4 in which we can observe the flowchart of each step with an image that shown the partial results.

2.2 Pipeline Implementation

In this chapter I will discuss in details the actual pipeline implementation. First of all I will briefly describe the used framework for the actual implementation. After that I will describe step by step each block of the pipeline, describing how each task is achieved.

The whole pipeline was implemented by using python, which is an high level object oriented programming language and to perform the necessary image processing operations, the managing of input and output images, and the other operation I've mainly used OPENCV [**OpenCV**] and SIMPLEITK; for the other operations involving the image tensor I've used NUMPY [**Numpy**].

Since python is an high level language, it allows an easy and fast implementation

of the code, on the other hand working with optimized image processing libraries written in C++ allows to prevent the lack of performances.

The whole code is open source and available on GitHub [**REP:CTLungSeg**] 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>). To automatize the segmentation on multiple CT scans are provided bash and powershell script and, even if the centroids are already estimated, a training script is provided, in order to allow the user to estimate its own set.

The whole pipeline is organized into three scripts, which performs the main tasks:

- lung_extraction
- train
- labeling

The usage of SimpleITK to manage input and output file, allows the compatibility with medical image formats and the preservation of the spatial information.

2.2.1 Frameworks

In order to perform all the necessary image processing operations both involving 2D and 3D filters, to perform the color quantization and to manage the input and output medical image format, I've used mainly two libraries for image processing and computer vision. These libraries has been written in C++ but has multi language support. I've performed all the 2D image processing operations like median blurring or filter application by using OPENCV [**OpenCV**]. For the managing of medical image formats, ensuring the preservation of voxel spatial information, and for the 3D operations, I've used SIMPLEITK. To perform all the other operation on the image array, I've used NUMPY [**Numpy**].

For the preliminary lung extraction, I've used the pre trained Unet available on GitHub(<https://github.com/JoHof/lungmask>).

OpenCV

OPENCV, acronym for Open Source Computer Vision, is an open source computer vision and machine learning software library. OPENCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. This library is implemented in C++, however bindings are available for PYTHON, JAVA and MATLAB/OCTAVE. This library can use also hardware acceleration like Integrated Performance Primitives, and also CUDA and OPENGL based GPU interfaces are available.

I've used the tools from this library to perform all the processing that involves the single image and, most important, to perform the color quantization, since the k-means implementation offered by the library allows to cluster multi channel images in an efficient way.

SimpleITK

SIMPLEITK is a simplified programming interface to the algorithms and data structures of the Insight Toolkit (ITK) that support many programming languages. The library provides a simplified interface to use Insight Tool Kit(ITK) library. Insight Tool Kit (ITK) is an open source library which provides an extensive suite of tools for image analysis, developed since 1999 by US National Library of Medicine of the National Institutes of Health. This library provides tool useful to works also with N-dimensional images. This library provides a powerful tools for the reading and writing of the image. Since ITK, ans so SIMPLEITK, consider the image like spatial object and not like arrays of values, it store also information about voxel spacing, size and origins, provided as well as the array, this makes us able to works only with the array by using NUMPY or OPENCV, by preserving the spatial information of the image.

Numpy

NUMPY is an open source project aiming to enable numerical computing with Python. It was created in 2005, building on the early work of the Numerical and Numarray libraries. NUMPY is developed in the open on GitHub, through the consensus of the NUMPY and wider scientific Python community. [Numpy].

Pre-Trained UNet

This network is a pre trained UNet which allows an Automated lung segmentation in CT under presence of severe pathologies. The whole code is written in python and it is based on torch and torchvision libraries. The repository offers 4 pre trained models for different kind of segmentation, like single lung lobe segmentation and the extraction of lung in presence of severe ILD. Each model perform the segmentation slice by slice.

The used network is a U-Net, that is a modification of the convolutional neural network architecture useful for medical and biological field, because is developed to works also with a small training dataset. The allowed pre-trained models are the follows :

- **R231** : This model, trained on a dataset that cover a wide range of visual variability, performs a segmentation on individual slice and extract the right and left lung lobes including airpockets, tumor and effusion, wothout including the trachea.
- **LTRCLobes** : This model, trained on a subset of LTRC dataset, perform an individual segmentation fo lung lobes, but have limitad performances in case of severe ILD.
- **LTRCLobes_R231** : Model which fuse the two previous one. Fills the false negative of LTCTLobes by using R231, but is computational expansive.
- **COVID231-Web** : Augmentation of R231 with COVID-19 slices that were mapped back from regular imaging formats to HU The data was collected and prepared by MedSeg.

The network was trained in order to takes the maximum flexibility with respect of the field of view, in order to enable the segmentation without a prior localization of the organs.

The model is trained on 231 CT scans collected from PACS, selected following these criteria:

- Random Sampling (57 scans)
- Sampling from Image Phenotypes (71 scans)
- Manual selection of edge cases :
 - Fibrosis (28 scans)
 - Trauma (20 scans)
 - Other Pathology (55)

That ensure a proper variability of the training dataset.

The **COVID231-Web** was used developed to incorporates collections of slices and case reports from the web are often cropped, annotated or encoded in regular image formats so that the original hounsfield unit (HU) values can only be estimated. Since the scans used in this works do not fall within these cases, we have used the the **R231** model, which works very well with COVID-19 CT.

2.2.2 Lung Extraction

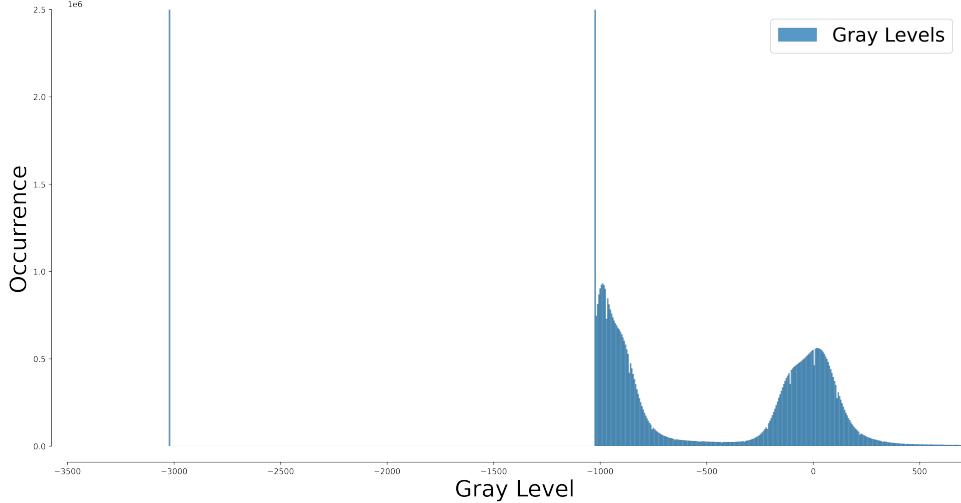
This is a pre-processing step , which involves the creation of a lung mask and the managing of the Hounsfield Unit. The creation of a mask for the lung regions aims to reduce the number of clusters and avoid the formation of false positives by removing externalbfmask which code is open source. The use of the neural network allows to obtain a good lung segmentation also for patients with severe ILD, from which some lung areas will be excluded with method like threshold and connected components, as we can see in Figure 2.3.

Once we have found a suitable mask for the lung, a managing of HU must be performed. The k constant in the HU definition (equation 1.1) may change according to the scan manufacturer or scan model; moreover, 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: for instance in the CT scan in Figure 2.5(a) the padding value is $-3000HU$ and the air value is -1024 .

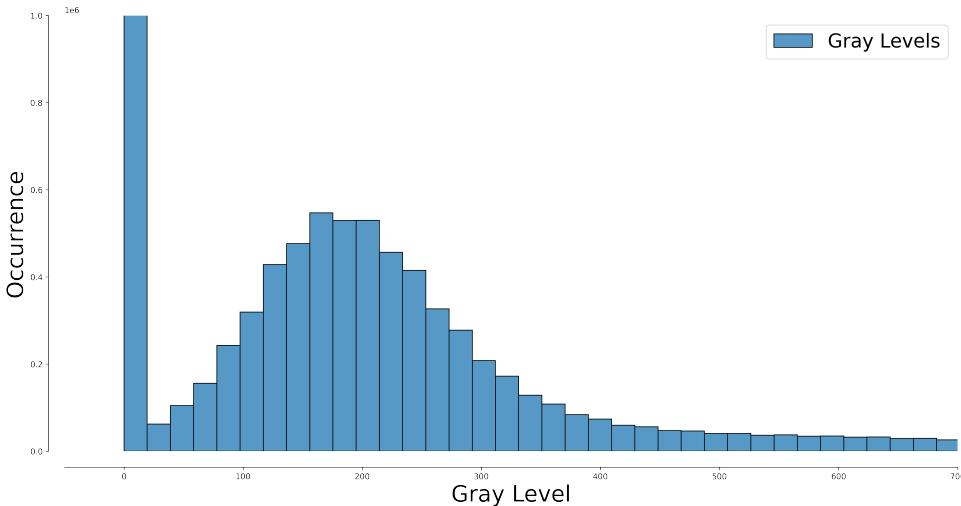
The first thing to do is to make the padding value and the air value equal for each scan considered and shift them to 1: in this way we have registered the HU for scan from different manufactures in a common space, as we can see in Figure 2.5(b).

May happen that some patient have metallic prothesis, that lead of HU out of range. Since usually this implant are outside the lung, are removed after the mask application, so no other step are needed.

Now, we have found the mask for the lung regions, and we have managed the HU, so by a simple element wise multiplication we are able to isolate the lung from the rest of the body, removing also the organs, like heart and intestine, presents in the CT scans.



(a) Histogram of a CT scan before registration



(b) Histogram of a CT scan after the registration

Figure 2.5: Histogram of voxel values before and after the pre-processing. We can observe that before the pre processing there are some HU out of range, which are the values used to fill the regions outside the tube, and the air value is around -1000 HU according to HU definition. After the rescaling we can observe that all the values are non-negatives.

We have observed that the presence of the airpockets and bronchial structure in the lung regions is one of the main source fo false positives, so an extra step was added in order to remove as much as possible this kind of structures.

Air pockets have an elongated shape, respect to the other structure which usually are rounded, so the basic Idea was to use this kind of information. By using the cornerEigenValsAndVecs function implemented in OPENCV. This particular filter compute the covariant matrix of the derivative in a neighborhood and the corresponding eigenvalues. If a particular regions have an elongated shape, one of the eigenvalues (corresponding to the eigenvector in the direction of the structure) will have an higher values, otherwise both eigenvalues have a lower values. So we have applied this filter on each slice of the scans and took the maximum eigenvalues.

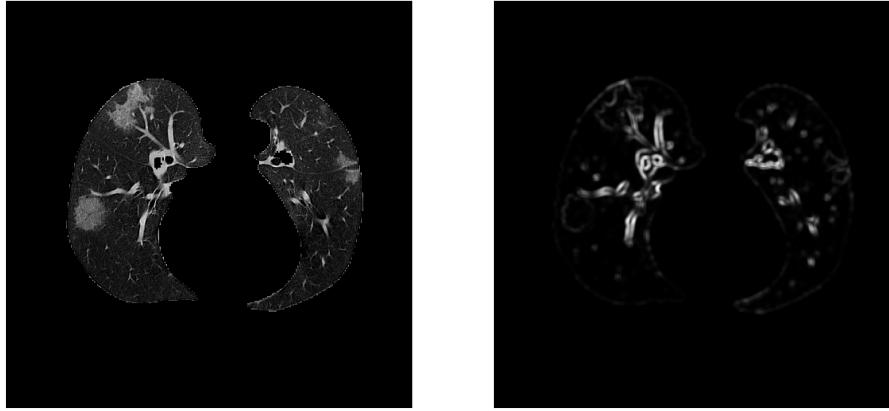


Figure 2.6: From left to right: Lung regions selected by the UNet; Maximum eigenvalues map of the lung. As we can see the UNet does not exclude the bronchial structure from the lung, on the other hand, the maximum eigenvalues map delineates very well these regions. We have used this map to remove the unwanted bronchial regions.

In Figure 2.6, 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 a simple threshold on the map was taken. 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.

Once lung regions are extracted, we are ready to perform the actual segmentation, or, if we haven't already estimate the centroids, performing the training step.

Algorithm 1: Pseudo-code for the lung extraction script

Data: Volume(CTScan)

Result: Volume with extracted lung

```

mask ← ApplyUNetModel(Volume)
volume← (volume < 1200) = volume air values
volume ← ShiftMinimumTo1(volume)
lung ← (volume o mask)
/* Start the bronchial removal */
```



```

eigen ← maxEigenvalues(lung)
alveolar_mask← (eigen < threshold)
lung_woBronchi ← (lung o alveolar_mask)
/* Refine the alveolar removal */
```



```

eigen ← maxEigenvalues(lung_woBronchi)
alveolar_mask← (eigen < threshold)
lung_wo_bronchi ← (lung_woBronchi o alveolar_mask)
```

2.2.3 Training

This step consist in the estimation of the centroids of the color space. May be really time consuming, but it is performed only once, so during the actual segmentation the corresponding script isn't run.

To achieved the estimation of centroids, a k-means clustering of the multichannel images of several CT scan from different patients is performed. In order to achieve a uniform representation of each cluster, the scans included in the training set were carefully selected from the 3 datasets used in this work. The main rule used for the selection is that in each scan must me present a huge amount of infection areas and also a well representation of artifacts, in order to takes into account all the possible features.

The achievement of the training task involves two main steps :

1. **Preparation of images** : involves the building of the multi channel images, and the registration in a common space;
2. **Clustering** : Actual clustering, involves also the managing of the background problem.

Preparation of Images

This step involves the preparation of images, with the building of the multi channel image that incorporates neighbouring and edges information as well as the registration in a common space and the managing of an allocation memory problem.

As I've said the multichannel image is build to incorporate more information during the clustering. We have found that a 4 channel image will provides good segmentation results. The 4 channel of the image are built as follows :

- Pure image after histogram equalization;
- Median Blurring of the equalized image;
- Gamma corrected image;
- Standard filtered image.

In Figure 2.7 I've displayed the 4 different channel of the image. Each channel allow us to consider different information

The histogram equalized image and the gamma corrected allows to takes 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 operating in small image regions, in order to takes 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. Both of this filters involves the single voxel.

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

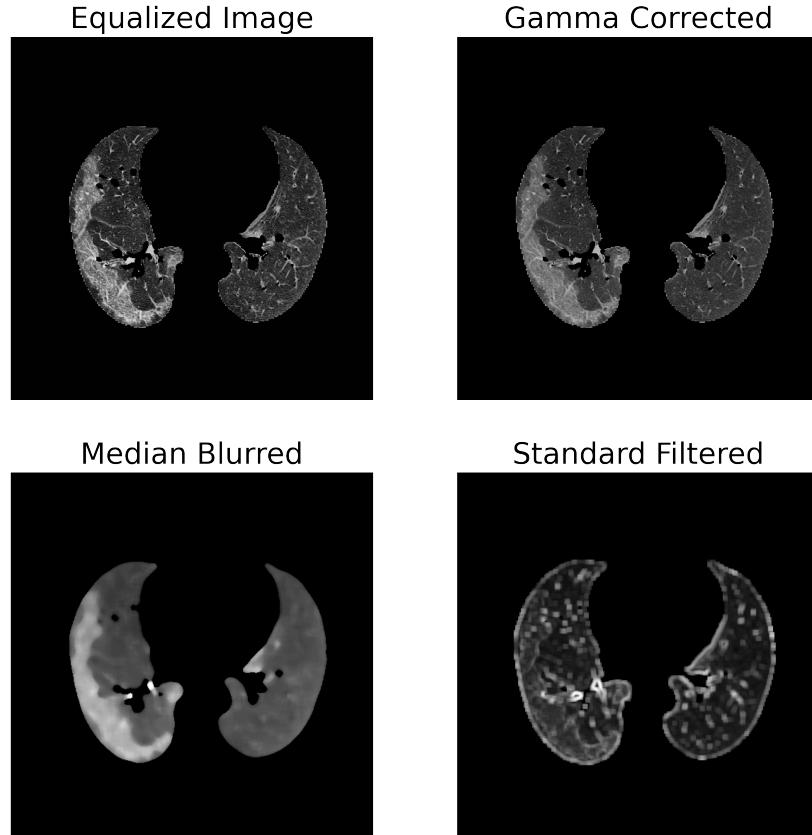


Figure 2.7: *Channels of the image.* From left to right and from top to bottom the image after the histogram equalization, the gamma corrected image, the median blurred image and the std filtered image. This channels allow us to consider informations about the single voxel but also about the neighbouring voxels and their variability.

The last channel used is the image after the application of a local standard deviation filter which consist in the replacement of each pixel value with the standard deviation of its neighborhood, help us to distinguish the remaining bronchial and artifact structures.

Each image channel is normalized according to means and standard deviation of the whole scan.

The first step consist into the construction of the multichannel image of for each input series, after that all the images are shuffled and divided into several sub-samples. If the provided training dataset is huge, may happen that the required memory cannot be allocated. Divide the training set into several subsamples lead to less memory requirement, allowing the training. From this step we obtain a several set of centroids, on which a second clustering is performed, in order to found the final centroid matrix.

The creation of several sub-samples is made since the creation of a single, huge array with several images is not always possible, since requires a huge quantity of memory to be allocated, so we have chose to divide all the images into several sub-samples and cluster them independently, after that a clustering on the estimated centroids is performed.

Clustering

This step consist into the performing of the k-means clustering for the centroids estimation. To perform this task I've used the OpenCV algorithm, which provides an optimized implementation of the algorithm for multi channel images. A first clustering is applied on each sub-sample, resulting in a set of centroids for each one of them. On this set is applied a second clustering, which provides the actual centroids. In both of the clustering, the initial centroids set is initialized by using the k-means ++ algorithm, which allows to improve speed and accuracy of the clustering algorithm [Arthur2007].

During this task we have to manage some issues. As we can see from Figure 2.8 the number of voxel with $GL = 0$ is several order of magnitude higher than for other GL . As prior we know that these voxels belonging from background, so this cluster is over represented. Since kmeans cluster requires an homogeneous representation for each cluster, this may raise problem during the centroids estimation. In order to overcome this issue we have simply removed this voxels from the clustering.

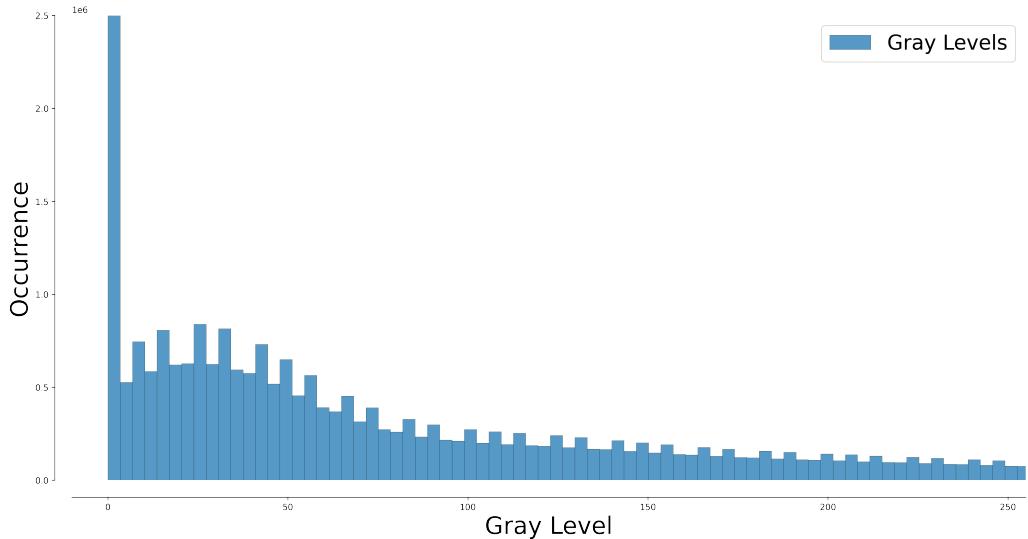


Figure 2.8: Histogram of the multichannel image to cluster, we can clearly see the overrepresented cluster at 0 GL

An other problem may be the estimation of the correct number of clusters. k-means clustering requires a prior knowledge on the number of clusters which is a crucial choice. In our case the anatomical knowledge about the lung may help, since we can consider one cluster for each anatomical structure. In the end we have found that 5 clusters are an optimal choice, and the considered structures are the following:

- Lung Parenchima;
- Edges;
- vessel surrounding bronchial structures;
- Ground Glass Opacities and consolidation;
- Bronchi.

We don't need a cluster to represent the background, since as I've said before the corresponding voxel aren't takes into account during the clustering. In the end a set of centroids for each subsamples was estimated and a second clustering was performed, to found the optimal centroids. This process takes a lot of time, but once we have estimated the optimal centroid set, we haven't to repeat it.

The whole step is summarized in the pseudocode in Figure 2.

Algorithm 2: Pseudo-code for the training script

Data: CT scans with Extracted lung

Result: Centroid matrix

```

foreach scan  $\in$  input_scans do
    | read the scan
    | sample  $\leftarrow$  image_array
end

/* prepare subsamples */ 
sample  $\leftarrow$  build_multichannel(sample)
sample  $\leftarrow$  shuffle(sample)
subsamples  $\leftarrow$  split(sample)
/* start the first clustering */ 
foreach Sub  $\in$  subsamples do
    | center  $\leftarrow$  kmeans(sub, number of centroids)
    | centroid_vector  $\leftarrow$  append(center)
end

/* Refinement */ 
centroid_matrix  $\leftarrow$  kmeans_clustering(centroid_vector, n_centroids)

```

2.2.4 Labeling

This is the last step of the pipeline, which involves the actual segmentation. This task is performed by simply assign each voxel 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 build the multichannel image as described before. after that will assign each voxel to the cluster of nearest centroids, which is the centroids that minimize the distance :

$$cluster = \arg \min_S \sum_{i=1}^k \sum_S \|x - \mu_i\| \quad (2.1)$$

where x is the color vector of the voxel and μ is the i th centroid. During this process the background is automatically assigned to the 0 label, since we know as a prior that its value is 0.

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  $c \in \text{centroids}$  do
| distances←  $\|image - c\|^2$ 
end
labels← arg min (distances)

```

2.3 Tuning of Parameters

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 actual 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 Estimation of the 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 source 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;
- Ground Glass Opacities and consolidation;
- Bronchi.

Notice that the background of the image isn't 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've considered the internal cluster variability.

Clustering techniques try to group the data in 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 cluster is too low, the similarity within each

cluster is low and increasing the number of clusters will reduce the internal variability, that will be 0, when the number of cluster is equal to the number of samples. This means that after 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) :

$$SSE = \sum (x_i - c_j)^2 \quad (2.2)$$

Once this task is completed, the results was printed in Figure 2.9.

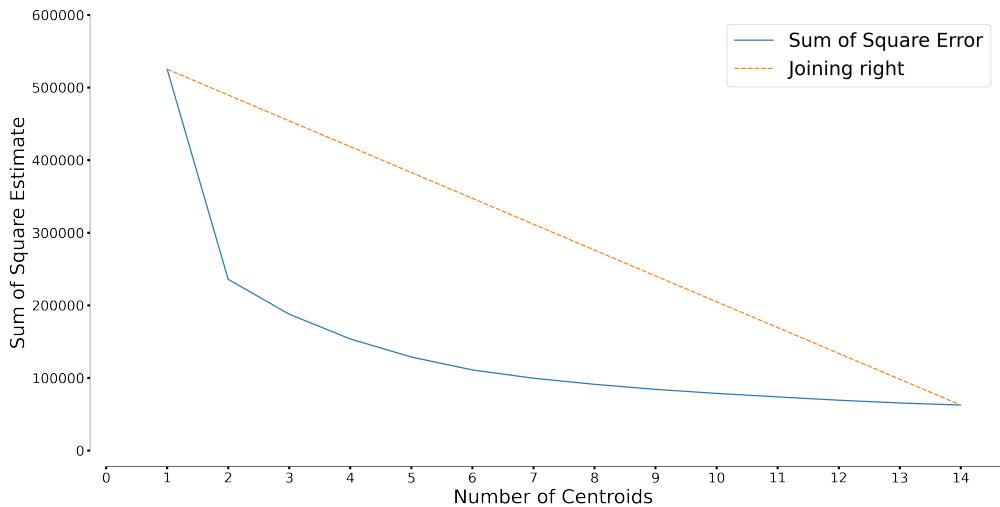


Figure 2.9: Intra-cluster variance vs the number fo clusters(blue); right joining firsts and last point(orange). We can se the elbow shape of the graph; the elbow location in the one which maximize the distance between the right joining line and the elbow curve.

The optimal number of cluster is the one that corresponds to the elbow of the curve. Since found this feature is visually difficult , I've took as elbow the point which maximize the distance between the right joining the first and the last point. From this analysis we have found that the optimal number of clusters in 5, which corresponds to the same that we have found considering the lung anatomy.

2.3.2 Kernel Size Optimization

During the building of the multi channel image, we have to compute different image features, that requires the setting of different parameters, like median or standard filter kernel size. In order to achieve the best segmentation, we have performed an optimization step. This step aims to found 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've used SCIKIT-OPTIMIZE [**skopt**], more specifically the `gc_minimize` method.

This method seek 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 [**skopt**].

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

I've 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 [**PhDtheis**]:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (2.3)$$

Algorithm 4: Parameter Optimization Algorithm

Data: Test scans, Ground Truth

Function `objective(parameters, ref_labels, CT_scan):`

```

    labels ← segment(CT_scan)
    iou = IoU(labels, ref_labels)
    return 1 - iou
  
```

`best_parameters ← gc_minimize(objective, n_calls, n_random_init)`

This process allows to optimize the parameters in order to obtain the best results as possible. As reference labels I've used the ones valuated as gold standard from expert radiologist.

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

Chapter 3

Results

In this final chapter I will presents and discuss the results of the segmentation achieved by developed the pipeline.

In the first section I will describe the dataset used for the implementation of the pipeline and the segmentation. After that i will show the actual segmentation results.

The segmentation performances were measured by visual assessment, expert blind evaluation and quantitative comparison with the gold standard segmentation. As a control also segmentation on healthy patient were made, in order to ensure that no lesion were detected.

The centroid set used to obtain the segmentation is estimated on 10 CT scans carefully selected from the available dataset, in order to obtain a uniform representation of each cluster.

3.1 DataSet Description

This section is dedicated to the description of the dataset used for the developing and test of the pipeline. The description includes general image characteristics and some metadata. If within the dataset are provided also some ground truth manual segmented labels, also the segmentation modes are described.

3.1.1 Sant'Orsola

Sant'Orsola data was the ones mainly considered in this work. The consist into 83 anonymized CT scans from 83 different patients affected by COVID-19, manually labeled by interns; and 8 healthy control. 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 this dataset were available 5 manual segmentation, made and validated by expert. These segmentation were used as the ground truth for quantitative evaluation.

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 of 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 hospital. This dataset is born with educational and AI developing purpose. The studies are divided into 5 categories, from healthy patients to the most severe cases. Each scan of the dataset is saved in *.nifti* 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 category	254/648/125/45/2

As I've said before, the CT scans are divided into 5 categories, depending on the percentage of the involved lung parenchyma :

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

Of these five categories only 50 annotations are available, mostly involves only the patients of CT-1 group. 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 an expert radiologist. The anatomic structures labeled are the left and right lung and the infected regions. Each file is in '*.nii*' format and no metadata was available.

Unfortunately only 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 as input an image in HU.

3.2 Results

In this section I will describe the results. Each one of the three available datasets was segmented. The segmentation, performed on DIFA servers takes less than 2 min for each patient. The achieved segmentation will be matched with the manual segmentation available with the scans. Three different methods were used :

- Visual comparison, made collaboration with the Department of Diagnostic and Preventive Medicine of the Poloclinico Sant'Orsola - Malpighi,
- Blind expert evaluation
- Quantitative comparison with the gold standard

Also segmentation on healthy controls was taken into account, in order to check that no area was selected and estimate the false positives.

3.2.1 Visual Comparison

In this subsection I will discuss the segmentation results. The segmentation obtained by the pipeline were compared with the manual segmentation provided within the dataset. This comparison aims to match the different modality for patient that presents a regular lesion pattern.

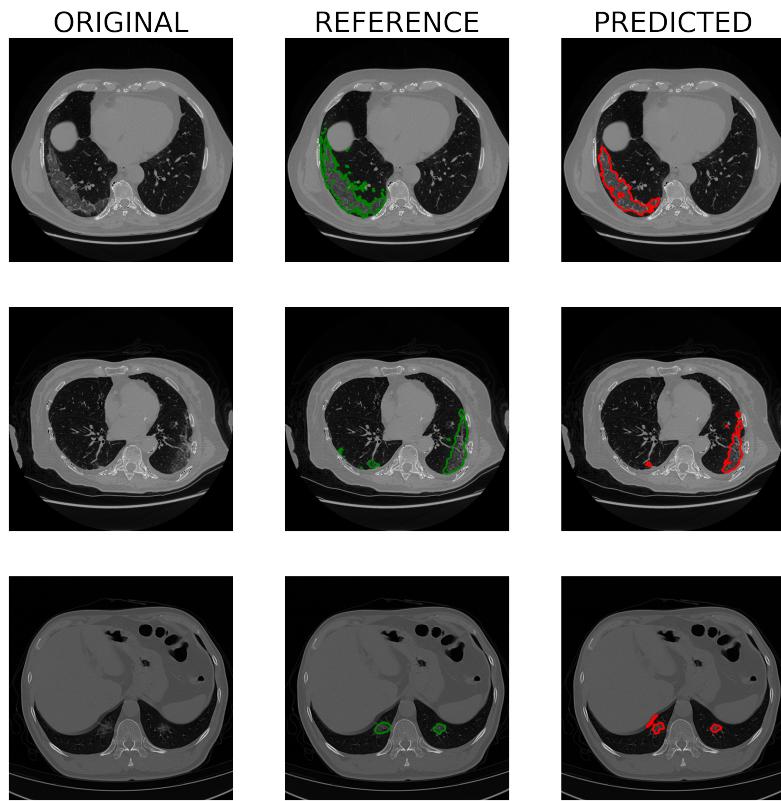


Figure 3.1: Comparison between the lesion areas detected by the developed pipeline(red) and the reference ones(green). We can see how the the lesion areas are correctly detected.

In Figure 3.1 I've reported a comparison between the segmentation achieved by the developed pipeline(red) and the reference labels(green) for three scans of patient with a typical lesion pattern. We can see how the lesion areas are correctly identified. We can observe how in each case the edges of the label estimated by the pipeline are more detailed respect to the reference ones; and the first case the achieved segmentation presents less false positives.

In the last scan we can observe how in presence of motion artifacts the pipeline may presents false positives; but the lesion areas are well segmented.

3.2.2 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. As we will see, some misclassified areas will raise, so we have used this kind of scans to detect the main source of false positives.

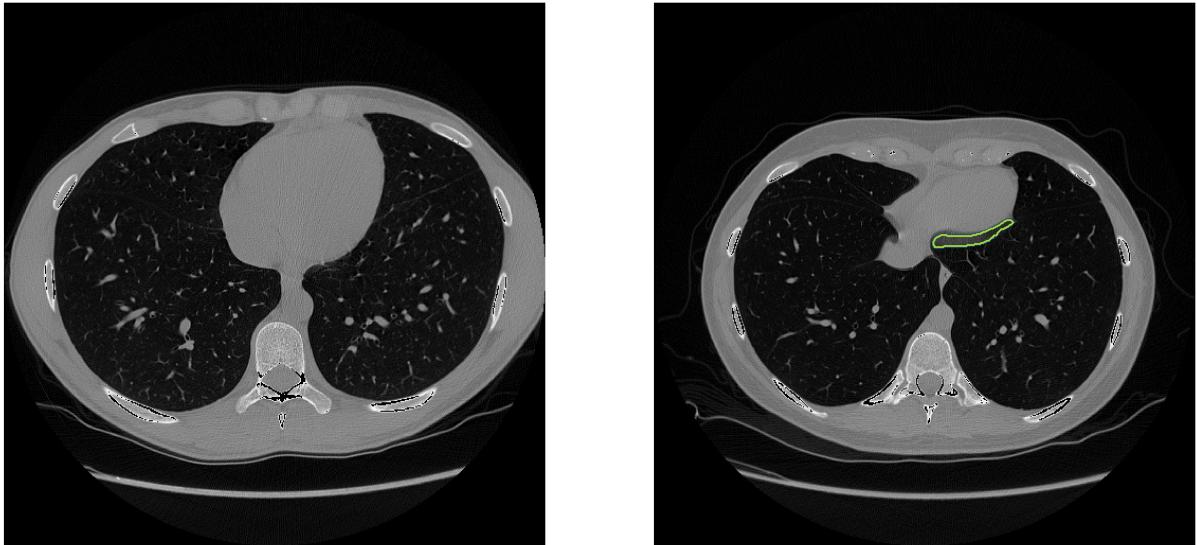


Figure 3.2: Segmented CT scans from healthy controls: Without motion artifacts(left), with motion artifacts(right). On the fist case we can see that no area is identified since no lesion is present; In second one we can see how, in presence of motion artifacts, some false positives may raise.

In Figure 3.2 I've displayed two segmented scans from healthy control. We can see that, in a normal case, no lesions areas were detected; but in presence of artifacts, some misclassified regions appears.

This results suggest that the main source of false positives is the presence of motion artifacts, caused by heartbeat and respiratory cycle.

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.

3.2.3 Gold Standard

In this section I will discuss the quantitative match between the achieved segmentation and the gold standard. The gold stanrd used involves 5 semiautomatic segmentation made by a certified software and refined and checked by experts. This segmentation was provided by the Department of Diagnostic and Preventive Medicine of the Poloclinico Sant'Orsola - Malpighi.

The comparison was made for all the scans by Dice similarity coefficient.

To define this kind of metric we consider the segmentation labels as a boolean mask, in which *True* corresponds to the identified lesion area, and *False* to other areas.

This mask is compared to the actual ground truth, given by the gold standard, so we are able to compute the True positives (TP), True negative (TN), false positives(FP) and false negatives(FN) rate. Once we have computed these quantities, we are able to compute the dice score coefficient, given by :

$$DSC = \frac{2TP}{2TP + FP + FN} \quad (3.1)$$

This coefficient, which is in $[0, 1]$, will provide a measure of the goodness of the segmentation.

Dice coefficient were computed for all the five patients and the results are reported in table 3.1.

	DSC	TP (%)	TN (%)	FP(%)	FN (%)
Patient 1	0.519	0.033	99.90	0.048	0.014
Patient 2	0.559	0.417	98.93	0.031	0.627
Patient 3	0.708	0.649	98.82	0.045	0.489
Patient 4	0.546	0.0298	99.92	0.021	0.028
Patient 5	0.765	0.269	99.56	0.006	0.159

Table 3.1: Table of the dice coefficient for each gold standard patient. The accurate manual segmentation was taken as ground truth.

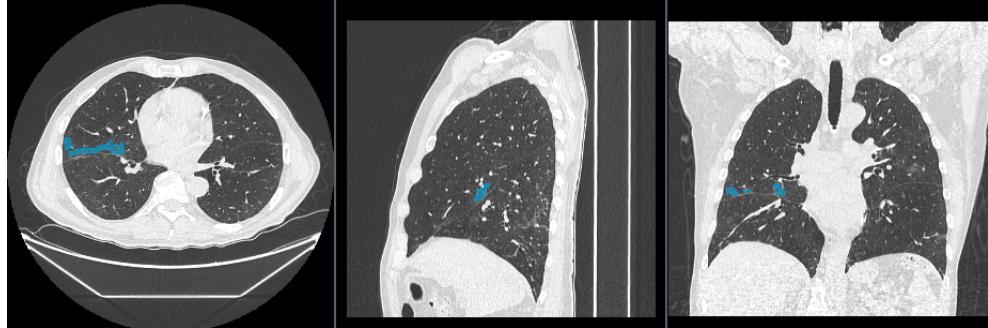
The first thing that we can notice is that the percentage of the true negative is much higher than the other.; that because the number of pixels concerning the labeled object would be very few against the number of pixels related to the background.

An other thing we can notice is that not all the patient have similar agreement: Patient 1, 2, and 4 show DSC near the 0.55, the remaining two near 0.73. This may be caused to the fact that the chosen patients presents different severity of the disease and lesion characteristics: as we will see the developed method performs well in case of normal or severe lesions, but have some difficult in case of low contrast between lesions and normal lung regions.

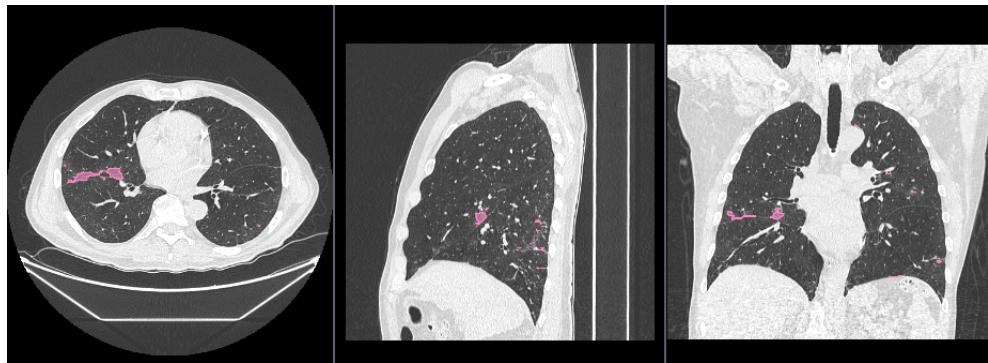
In Figure 3.3 I've reported a comparison between the ground truth (blue) and the pipeline segmentation (pink) for the first patient in axial, sagittal and coronal view. We can see how the main lesion areas are correctly identified, however a low underestimation of the total lesion volume is present, together with the presence of false positives in the lowest back regions.

This is a particular case, since the involvement of lung parenchima is very low, as is the contrast between lesions and healthy lung.

In Figure 3.4 I've reported a comparison between the ground truth (blue) and the pipeline segmentation (pink) for the third patient in axial, sagittal and coronal view. The patient presents a high involvement of lung regions. The lesion areas, which in this case presents a high contrast with the lung regions, are correctly selected. In this case no misclassified points are detected, which is in agreement with the percentage detected by the dice score.



(a) Ground truth for Axial, sagittal and coronal view of the first patient



(b) Predicted lesions areas for Axial, sagittal and coronal view of the first patient

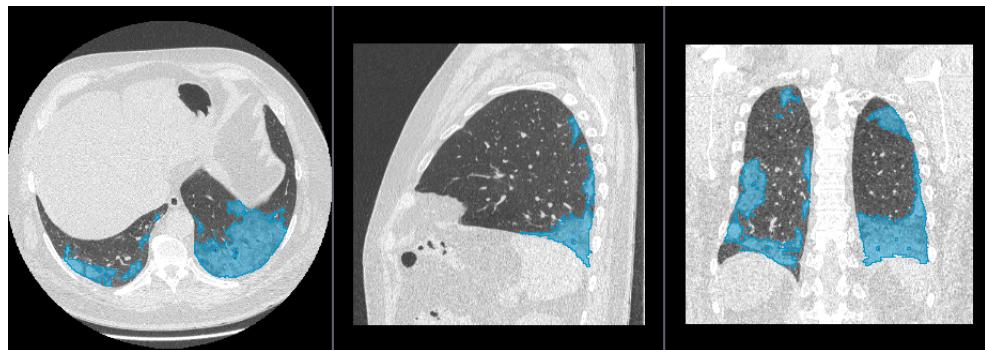
Figure 3.3: Comparison between the gold standard segmentation(blue) and the pipeline results(pink) for axial, sagittal and coronal view of a patient with a low involvement of lung parenchima. We can see that te main lesion areas are identified, even if an underestimation of the total volume is present together with some small misclassified points.

3.2.4 Expert Evaluation

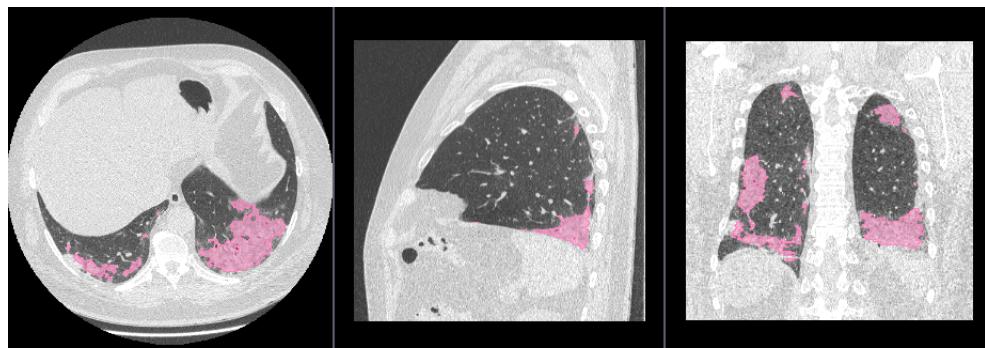
One control that was made to check the quality of the achieved segmentation, was a blind expert evaluation. We have randomly select 40 patients from the three dataset and organized the scans as follows:

For each patient we have displayed, slice by slice, two images: the scans with the pipeline segmentation(the one to test), and the manual labels provided within the dataset (control) as in Figure 3.5.

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 doesn't know which scan correspond to a specific segmentation technique. In order to ensure that, the order of segmentation was shuffled between patients.



(a) Ground truth for Axial, sagittal and coronal view of the third patient



(b) Predicted lesions areas for Axial, sagittal and coronal view of the third patient

Figure 3.4: Comparison between the gold standard segmentation(blue) and the pipeline results(pink) for axial, sagittal and coronal view for a patient with a large involvement of lung parenchima. We can see how the lesion areas are correctly identified

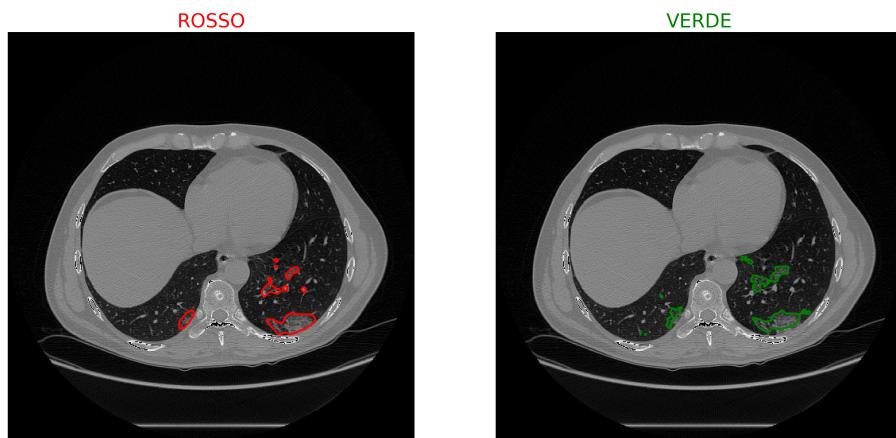


Figure 3.5: Example of comparison submitted to the experts. The two images report the segmentation of the same slices achieved with two different techniques: Manual and pipeline. Was asked to the expert to decide which one is better, without knowing which technique is used to obtain the segmentation.

Conclusions

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

As a preliminary step, a lung segmentation is performed by using a pre trained UNet, followed by a bronchial removal, which aims to reduce the rate of false positives.

The pipeline, tested on three different dataset, has provided good segmentation for the typical lesions cases with an accurate identification of GGO and CS. The proposed algorithm have also shown some limitation due to the eventual presence of motion artifacts, cased 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.