

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 |
| 1.3 Review on Image Segmentation Methods | 9 |
| 1.3.1 Classifiers Approach | 9 |
| 1.3.2 Clustering | 10 |
| 1.3.3 Color Quantization for Medical Image Segmentation | 11 |
| 1.3.4 U-Net | 12 |
| 2 Infection Identification Pipeline | 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 | 24 |
| 2.2.4 Labeling | 27 |
| 2.3 Optimization of Parameters | 29 |
| 2.3.1 Estimation of the Number of Clusters | 29 |
| 2.3.2 Kernel Size Optimization | 31 |
| 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 Time Performances | 34 |
| 3.3 Results | 35 |
| 3.3.1 Healty Control | 36 |
| 3.3.2 Zenodo and Mosmed | 36 |
| 3.3.3 Gold Standard | 36 |

| | | |
|-------|--------------|----|
| 3.3.4 | Double Blind | 36 |
| 3.3.5 | Discussion | 36 |

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 [9].

Many COVID-19 affected patients have shown ground glass opacities(GGO) and consolidation(CS) in chest CT, for which the severity, shape and involved percentage of lung parenchima are made in relation with the stage of the disease [4]. As shown by Huang [6], initial prospective analysis have shown that the 98% of examined patients have bilateral patchy shadows or ground glass opacity (GGO) and consolidation(CS) in lung. Other study have monitored the change on volume and shape of these features on healed patients [1] 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.

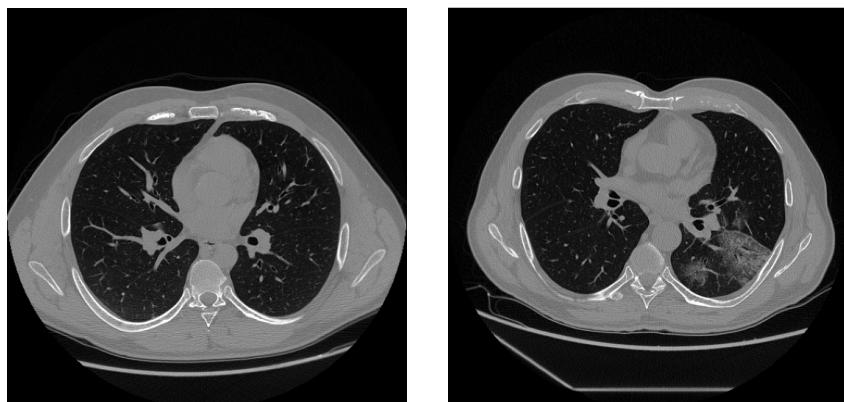


Figure 1: *CT scan of thorax for an healthy patient(left) and a COVID-19 affected one(right) in which we can observe a huge amount of GGO in the right lung*

GGO and CS are not exclusive of COVID-19, but may be also caused by pulmonary edema, bacterial infection, other viral infection or alveolar haemorrhage [11]. However 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. The study of these patterns may help to understand the infection pathogenesis, which is not well known since COVID-19 is a new disease.

Austin in Glossary of terms for CT of the lungs [3] define the Ground Glass Opacities as *hazy increased attenuation of lung, 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, and the consolidation as Homogeneous increase in pulmonary parenchymal attenuation that obscures the margins of vessels and airway walls.. For the reason given before, the identification of this kind of lesions in CT scans of lung is very important. 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. 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 COVID19 affected patients. The developed pipeline is completely unsupervised and its based on the color quantization, so the relation between the HU and the different linear attenuation coefficient of tissues is exploited. In order to takes into account other features, like the spatial extension of the lesion areas, or the different shape of the lung structure, a suitable color space was build.

The pipeline was implemented in python and to perform the different operations different image processing libraries where used : *SimpleITK* and *OpenCV*.

The results of the segmentation was verified in double blind by expert assessment, matching them with manual segmentation. Also a quantitative measures was taken from 5 gold standard manual segmentation, which represent the actual ground truth. Also two public dataset(ZENODO [13], MOSMED [15]) are used 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 [17]. 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 [17]. Manual segmentation is possible, but is time consuming and subject to operator variability; making the results difficult to reproduce [20], 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 [18].

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 [17] 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 [17]. It can be applied on an image to distinguish regions with contrasting intensities and thus differentiate between tissue regions represented within the image [20]. 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 [17]. 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.

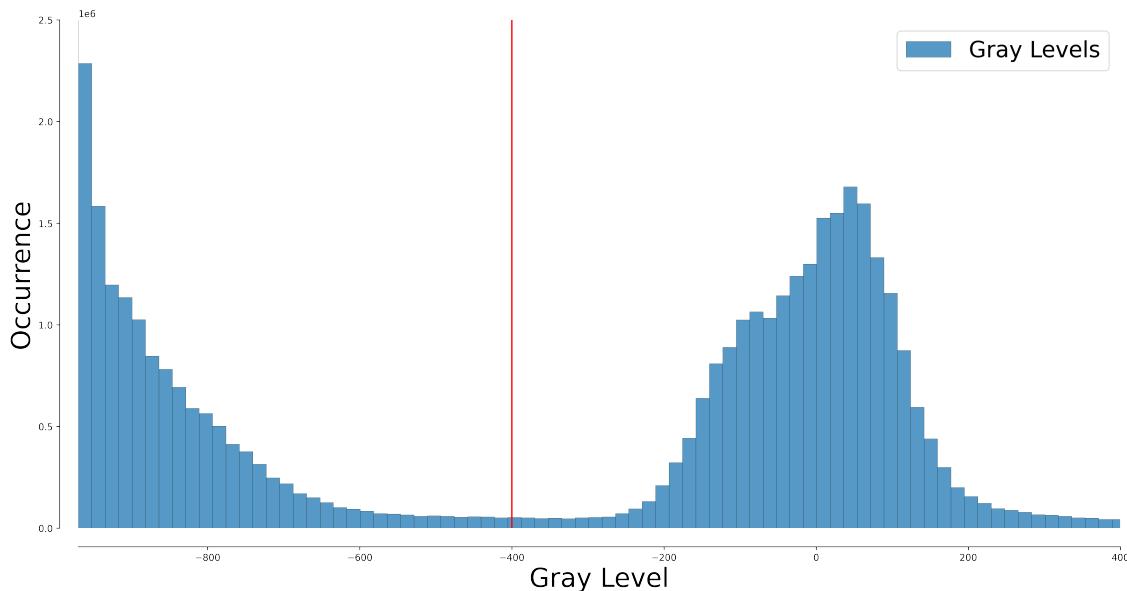


Figure 1.1: Histogram of a GL image with two well delineated regions. The threshold value(red line) was set visually at -400 HU

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 if the image, so it is sensitive to noise and intensity inhomogeneity, that corrupt the image histogram of the image and making difficult the separation [17]. 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 [20], 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 [17].

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 [20].

1.2.3 Deformable Model

Deformable Model use an artificial, closed, contour/surface able to expand or contract over time and conforme to a specific image feature [20]. This approach is physically motivated model-based thechnique for the detection of region boundaries [17].

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 occurring [17].

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 disavantages, MRF are widely used to model segmentation classes and intensity inhomogeneities [17].

1.3 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 ways, for example we can divide them into *supervised* or *unsupervised* if they require 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 [17] 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.

1.3.1 Classifiers Approach

Classifiers approaches use statistical pattern recognition techniques to segment images by using a mixture model that assumes each pixels belonging to one of a known set of classes [20]. 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 features of each pixel form a pattern that is classified by assigning a probability measure for the inclusion of each pixel in each class [20].

This approach assumes a prior knowledge about the total number in the image and the probability of occurrence of each class. Generally these quantities 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 assigning each pixel to the class with the highest posterior probability.

This approach requires a structure to segment with distinct and quantifiable features. It is computationally efficient and can be applied to multichannel images. This approach doesn't consider a spatial modelling and needs 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.

1.3.2 Clustering

Clustering approach is similar to classifiers one but in an unsupervised fashion, so it 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 requires training data, but suffer to an high sensitivity to the initial parameters and do not incorporates spatial model, so it is a pixel classification technique [17].

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 [2]. A vector representing the mean is used to describe each cluster, so this technique is described as a centroid model [14]. 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 [2] 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 choose
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 [2]

1.3.3 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.

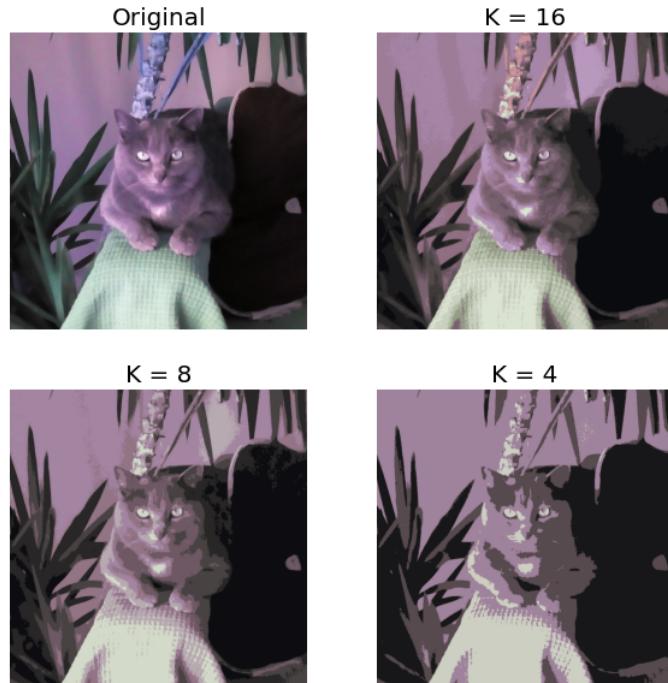


Figure 1.2: Color quantized RGB image. We observe the original image, a 16 color image which look similar to the original one, a 8 colors image and 4 colors image

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 [7].

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.

1.3.4 U-Net

Artificial Neural Networks are formed by using artificial neurons derived from physiological models [20]. 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 colled 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 [10]. 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 [8].

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 [10].

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

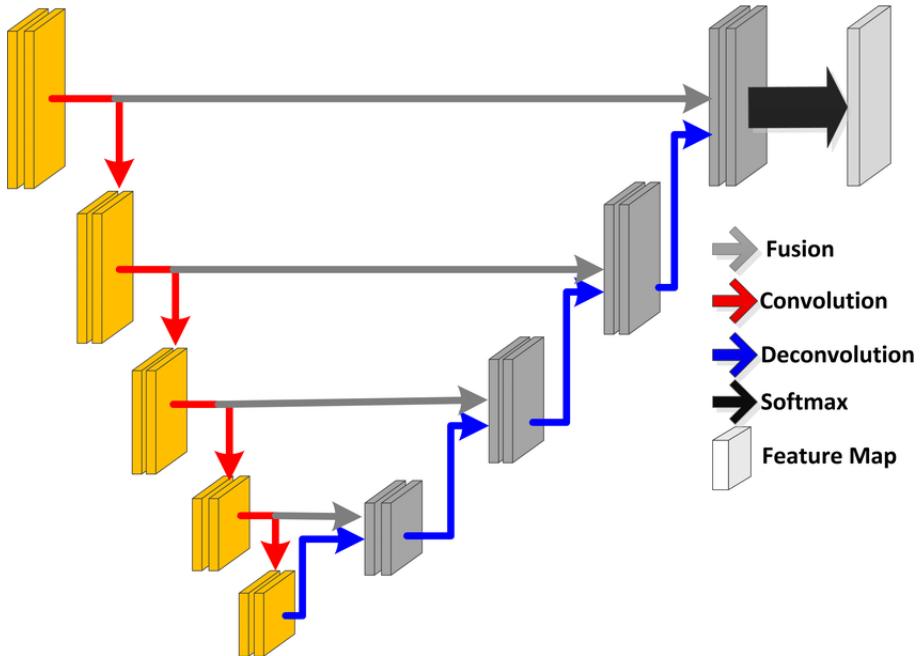


Figure 1.3

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 [10].

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

Infection Identification Pipeline

In this chapter I will discuss the developed pipeline. In the first section I will describe the basic idea behind the pipeline and describe the main structure, paying attention on the aims and the problems managed by each block. like the artifact removal. In the second section is described the actual implementation, by giving details about how each step is achieved. I've also provide a brief description of the used frameworks and tools. In the end the routines for the optimization of the internal parameters is described, given details about how the correct number of cluster was estimated.

2.1 Pipeline Description

The aim of these thesis is the developing of a pipeline for the identification of GGO and CS areas in chest CT scans of COVID-19 affected patients. The pipeline 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 provides a segmentation in few minutes.

The segmentation of the lesion areas is made by unsupervised technique, se doesn't requires to provide hte expected outcomes. The whole pipeline was developed and tested on CT scans kindly provided by Sant'Orsola Hospital. Also the public datasets were used as benchmark. The used datasets are manually annotated; the provided labels are used to check the pipeline performances.

During the pipeline developing we have to takes into account is that the infection regions may have different patterns according to the stage of the disease or recovery, as we can see in Figure 2.1, and usually these patterns are spatially disconnected; so we have decided to use a pixel classification technique.

The basic idea was to use the color quantization technique for the segmentation, grouping voxel based on color similarity, by 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 to display it, given by the Hounsfiend Units(eq 1.1:

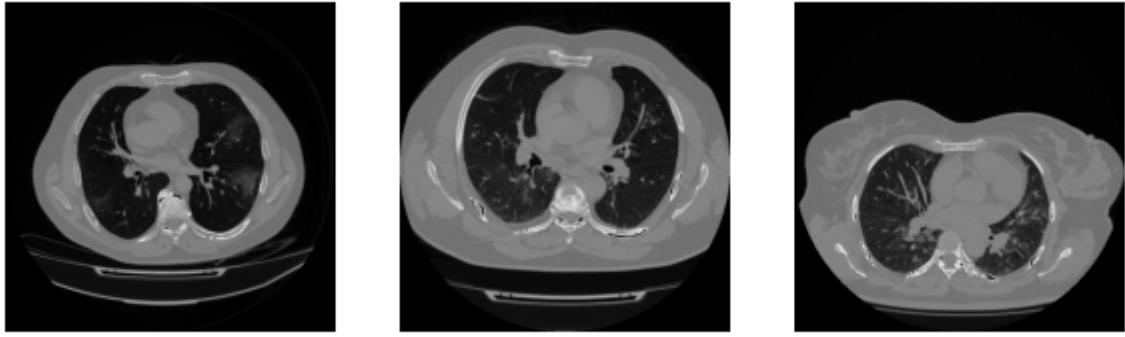


Figure 2.1: Ground Glass Opacity of COVID-19 affected patients with different severity of the disease. From left to right this scans belong to CT-1, CT-2 and CT-4 category of MOSMED [15] dataset

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 allow 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 the different channel are used to takes in account different properties, exploited by the application of different filters. 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. We have also used this features to discriminate between other lung regions like bronchi by exploit shape information.

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 [5]. 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 use that 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.

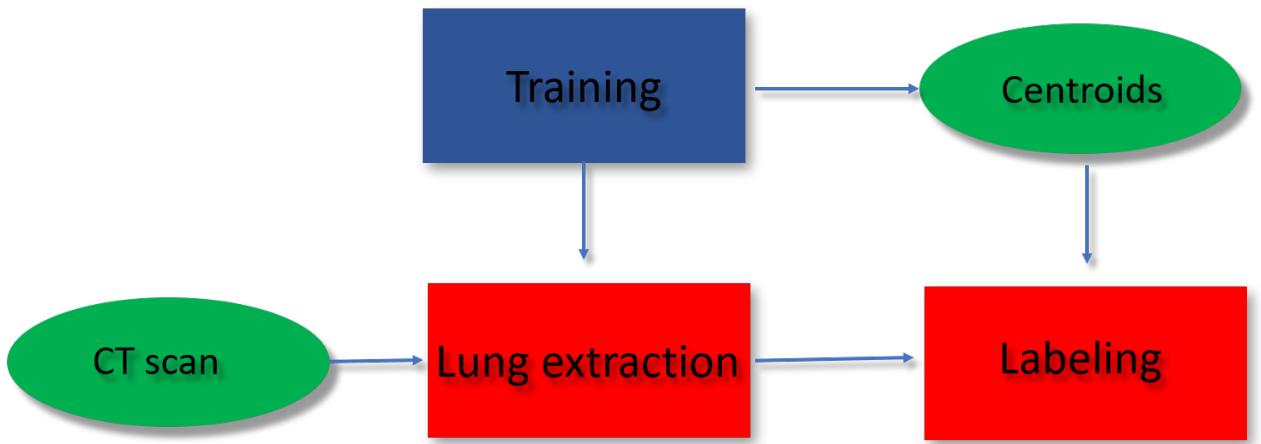


Figure 2.2: Flow chart of the main structure of the developed pipeline. The training process, which allows the estimation of the centroids, is performed only one time.

Pre Processing and Lung Extraction

This preliminary step is performed before both training and labeling; involves the managing of the HU, the isolation of lung regions and the removal of the 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 classification, identification and classification of lung pathologies [10]. 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 fomation 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 [10]. Rule based approach, like thresholding, region growing, ect, usually fails for CT sans of patients with severe ILD, as we can see in Figure 2.3. So, to achieve the lung segmentation I've used pre-trained UNet [10] [12], that provide a good segmentation of the lung regions. The UNet is a supervised approach, however is used only on this preliminary step, as we will see

the identification of infection regions is made with an unsupervised approach.

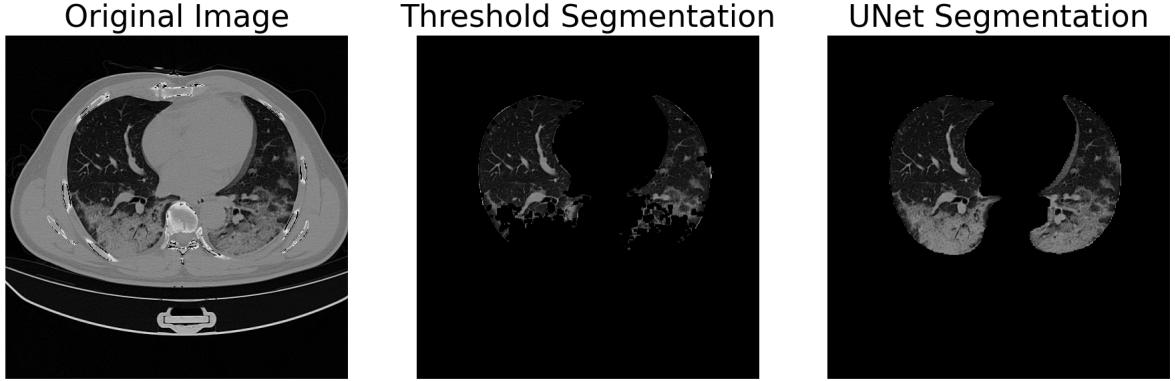


Figure 2.3: From left to right the original CT scan of a patient with severe ILD, the lung segmented by threshold and connected components, UNet lung segmentation. We can clearly see the missing areas in the first segmentation, corrected identified in the UNet one.

This kind of segmentation include in the lung region also motion artifacts and air-pockets, 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 form 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 we have to manage this problem. Moreover the patient that have a low involvement of lung parenchima have the cluster corresponding to the infection underrepresented, to overcome this issue a careful selection of the patient used for the training was performed. In summary, the implementation of this step involve the building of the multi-channel image, which allow us to takes into account also the neighbouring 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 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

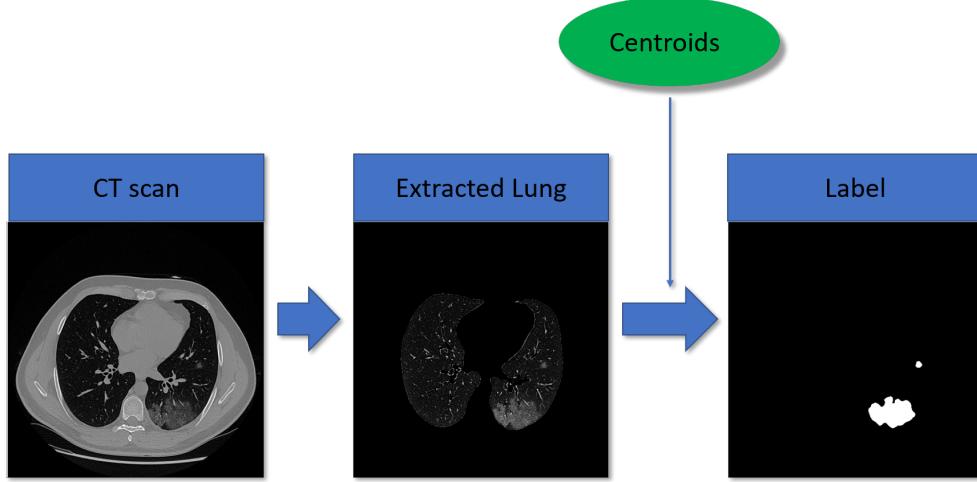


Figure 2.4: Actual segmentation step, from left to right we can see the input image stack, the isolated lung regions and the final label. To performed the labeling a set of pre-computed centroids was used.

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 describe 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 [5] and SimpleITK; for the other operations involving the image tensor I've used Numpy [16]

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 [19] 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 at <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 [5]. 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 the lung extraction, I've used the pre trained Unet available here on GitHub. In the end, to perform all the other operation on the image array, I've used numpy [16]

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. 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 kmeans implementation offered by the library allows to cluster multi channel images in an efficient way.

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 availableSimpleITK.

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. For more information on our governance approach, please see our Governance Document. [16].

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 network provided 3 pre-trained models :

- **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**

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)

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, to reducing the number of clusters and avoiding the formation of false positives by removing external structures which can be source of errors. In order to achieve this purpose, we have decided to use a pre-trained neural networks [12] 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.

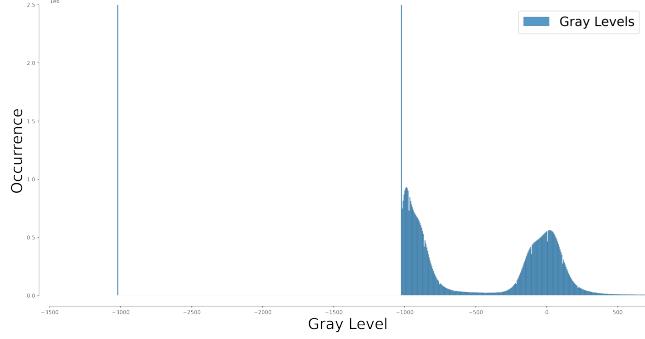
Once we have found a suitable mask for the lung, a managing of HU must be performed. The k constant in the HU definition (equation 1.1) may change according to the scan manufacturer or scan model; moreover, during the scan acquisition, all the regions outside the CT tube aren't sampled, so to obtain 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.

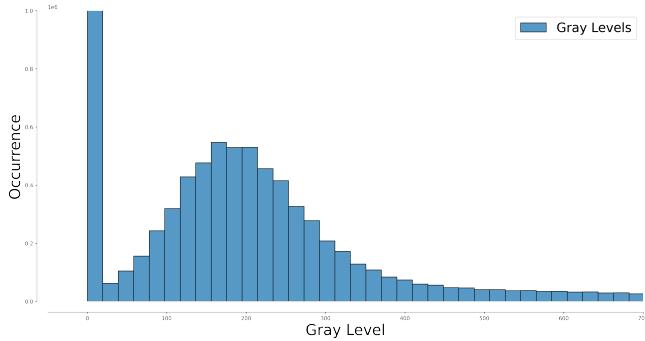
Now, we have found the mask for the lung regions, and we have managed the HU, so by a simple element wise multiplication we are able to isolate the lung from the rest of the body, removing also the organs, like heart and intestine, presents in the CT scans.

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 eigenvection in the direction of the structure) will have an higher values, otherwise both eigenvalues have a lower values. So we have



(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.

applied this filter on each slice of the scans and took the value of the maximum eigenvalues.

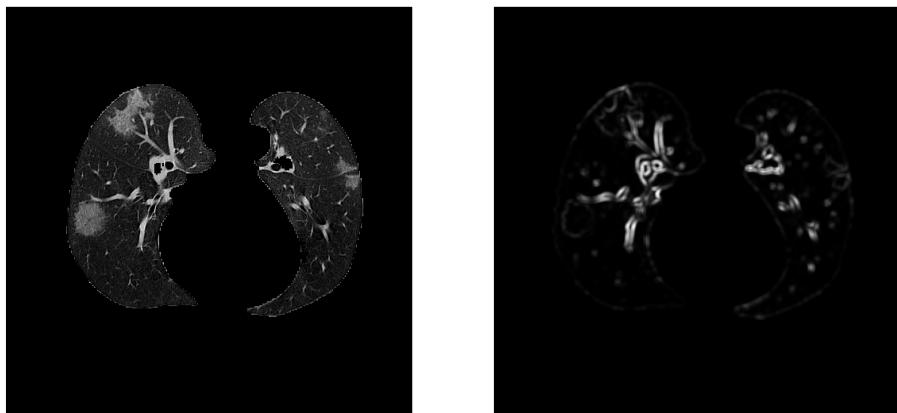


Figure 2.6: From left to right: Lung regions selected by the UNet; Maximum eigenvalues map of the lung. As we can see the UNet does not exclude the bronchial structure from the lung, on the other hand, the maximum eigenvalues map delineates very well these regions. We have used this map to remove the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

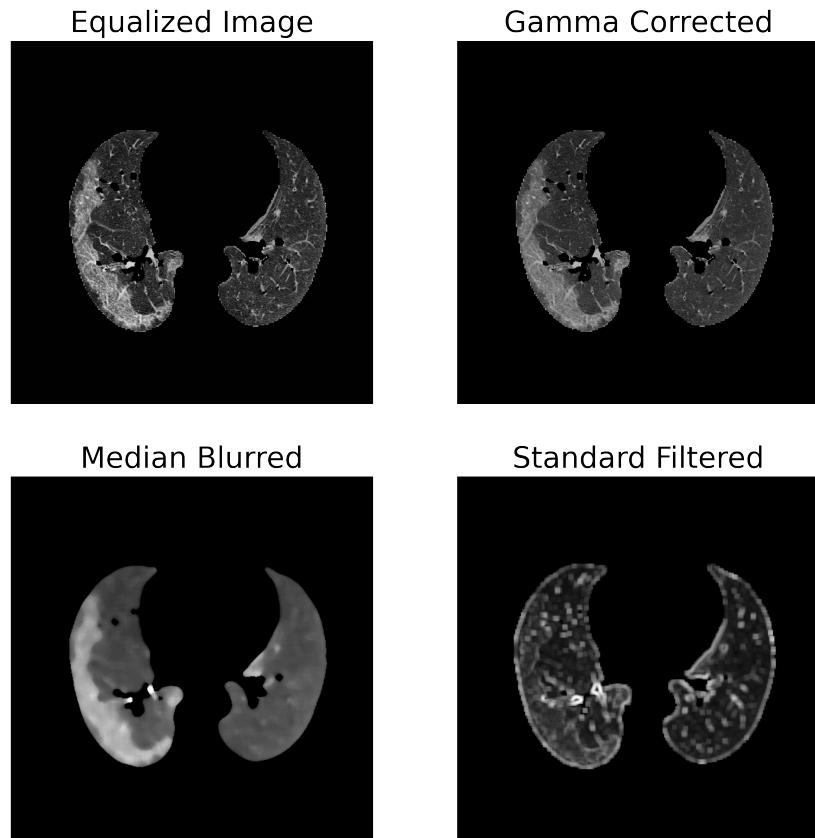


Figure 2.7: *Channels of the image. From left to right and from top to bottom the image after the histogram equalization, the gamma corrected image, the median blurred image and the std filtered image. These channels allow us to consider informations about the single voxel but also about the neighbouring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

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 make it evident 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.

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.

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. 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 [2]. 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 belong 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.

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:

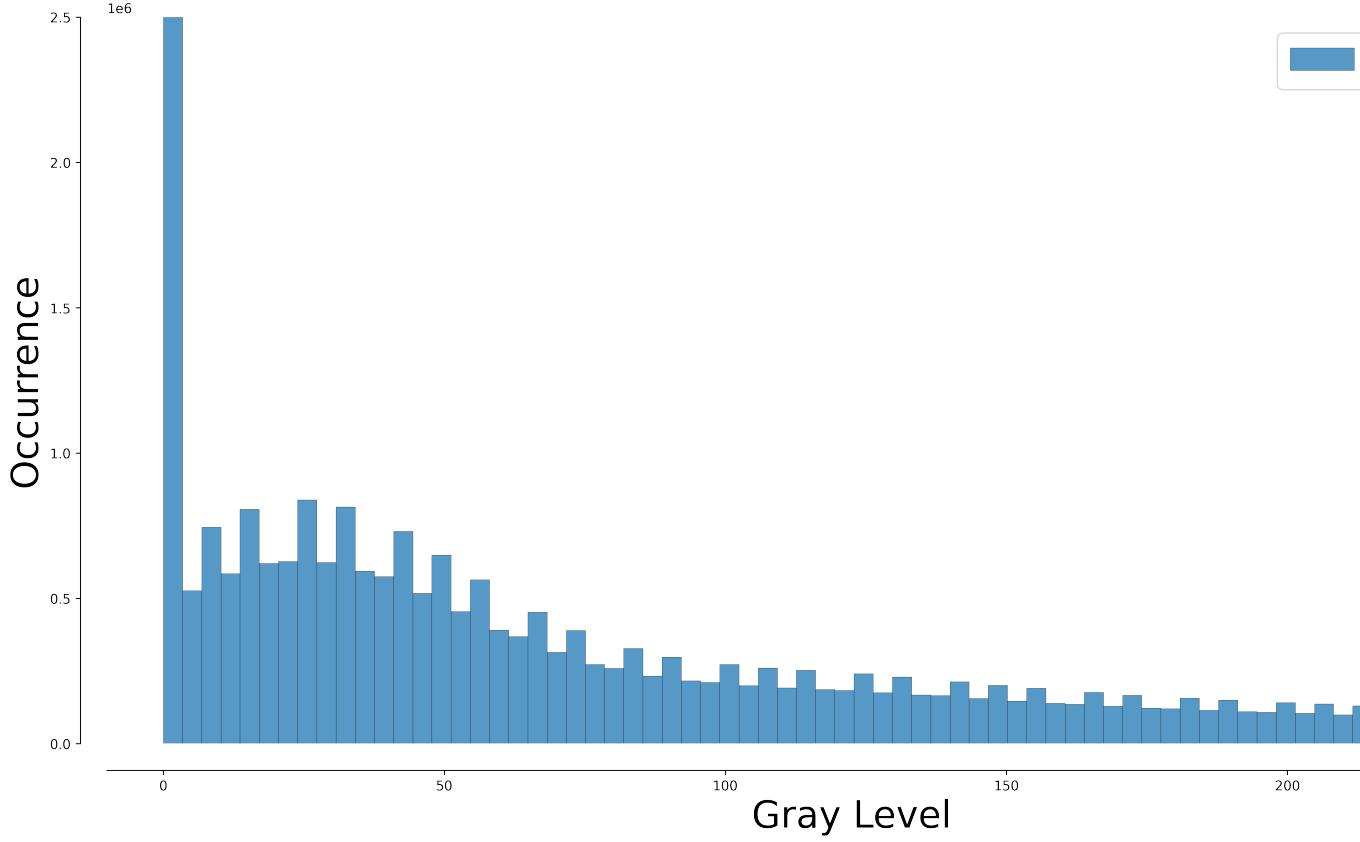


Figure 2.8: Histogram of the multichannel image to cluster, we can clearly see the overrepresented cluster at 0 GL

- 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.

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.

Algorithm 2: Pseudo-code for the training script

```

Function shuffle_and_split(images, numberofsubsamples):
    images←shuffle(images)
    output←split(images, number of subsamples )
    return output
End Function

Function kmeans_on_subsamples(subsamples, numberofcentroids):
    centroids <- []
    foreach Sub ∈ subsamples do
        | center←kmeans(sub, number of centroids)
        | centroids←append(center)
    end
    return centroids
End Function

Data: CT scans with Extracted lung
Result: Centroid matrix
foreach scan ∈ input_scans do
    | read the scan
    | sample←image_array
end
sample← build_multichannel(sample)
subsamples←shuffle_and_split(sample, number of subsamples)
centroid_vector←kmeans_on_subsamples(subsamples, n_centroids)
centroid←kmeans_clustering(centroid_vector, n_centroids)

```

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 :

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

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

```

Function imlabeling(image, centroids):
    foreach c ∈ centroids do
        | distances←  $\|image - c\|^2$ 
    end
    labels← arg min (distances)
    return labels
End Function

Data: CT scan to label, centroids
Result: GGO label
image←build _multi _channel
labels←imlabeling(image, centroids)
ggo← labels = GGO label

```

2.3 Optimization 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.

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 to 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 the the correct choice of the cluster.

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 error (SSE) :

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

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



Figure 2.9: 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 results that the best number of clusters is

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. To properly set these parameters I've build a routine to estimate the best values.

To build this routine I've used the tools given by scikit-optimize library. The idea behind the optimization process is very simple but requires some ground truth labels. I've simply generates many set of parameters, train the centroids with this set and label some images, after that I've compute the according of this labels with ground truth ones. After that I've simply chose the set of parameter which guarantees the best according with the references. The whole procedure is sketched in Figure 4

Algorithm 4: Parameter Optimization Algorithm

```

Data: Test scans, Ground Truth
old_iou ← 0
best_parameters ← None
do
    parameters←generates set of parameters
    labels← segmentation(parameters)
    iou←IoU(labels, ground_truth)
    if  $iou > old\_iou$  then
        | best_parameters← parameters
    end
    old_iou←iou
while iter < max_iter

```

This process allows to optimize the parameters in order to obtain the best results as possible. As reference labels I've used the one provided by the public datasets ZENODO and MOSMED. This optimization allows to achieve a better segmentation, but isn't necessary and doesn't interfere to the actual centroids estimation, keeping the techniques unsupervised.

Chapter 3

Results

In this chapter, after a description of the used dataset, I will discuss the results of the segmentation. The pipeline was mainly tested on the samples kindly provided by Sant'Orsola hospital, even if also some test on MOSMED and ZENODO was made. The results involves both timing and quality of the segmentation. The centroids used for the segmentation where trained over the 10 CT scans belonging from the different dataset. The scans was selcted in order to have a good representation of the different features in the lung, like anatomical structures, bronchial and artifacts. I've also used the some healthy scan patient to ensure that no lesion areas are identified.

As reference I've used some manual segmentation performed by expert radiologist. To match the ground truth and the label under test I've used as metrics the intersection over union(IoU).

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. This dataset was used to train the model by the centroids estimation and also to verify the time performances of the pipeline vs the one of a certified software. 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 |

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, which is the only one used for the performances checking, since it is the only one with the annotations. 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 another expert radiologist. The anatomic structures labeled are the left and right lung and the infectious 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 Time Performances

As I've said before the time performances are a relevant parameter of this pipeline. In this section I will discuss the segmentation time. Since for this work I've to

segment several scans (over than 100), I've performed the two main step (lung extraction and labeling) separately, and compute the time segmentation time for each slice and phase. The timing for the training step was not measured since this step is performed only once, so doesn't affect the total segmentation time.

The time performances are measured by performing the segmentation on the DIFA servers.

In Figure ?? I've plotted the times of the labeling step for each scan versus the number of slices of the scan. As we can see the timing increase linearly with the number of slices. In order to find the segmentation time for the single slice, I've performed a linear fit, the desired time will correspond to the angular coefficient. From this analysis I've measured that the segmentation time 0.074 sec for each slice of the CT scan.

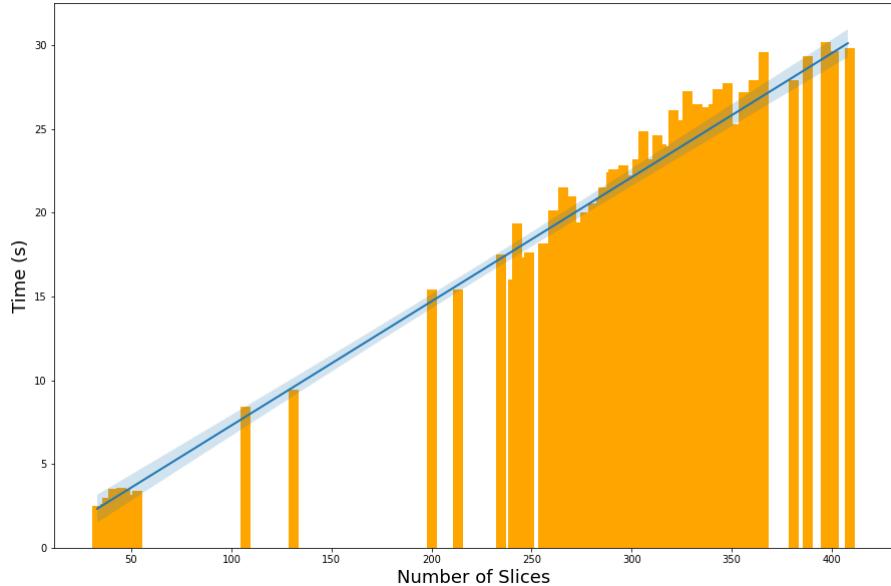


Figure 3.1: Total labeling time vs number of slice for each scan of each dataset

3.3 Results

In this section I will describe and discuss the segmentation results. The segmentation was made on 123 patients belonging from three different dataset:

- Sant'Orsola
- ZENODO [13]
- MOSMED [15]

For each scans also manual segmentation was available, however only 5 scans provides labels such that can be considered as gold standard. Together with COVID-19 affected patients, also healthy controls were provided, which we have used as benchmark to ensure that no lesion areas were detected, or at least to have a measure

of false positives.

A visual assessment between the labels provided by ZENODO and MOSMED and our segmentation was made, in order to make a comparison . Moreover two more relevant check were made :

- Matching between our segmentation and gold standard;
- Double blind check between our segmentation and manual ones.

In the end, in the last part, we discuss the results.

3.3.1 Healty Control

In this part I will discuss the results of the segmentation of healthy controls. The segmentation was performed on 8 scans, by using the same method described before. In

3.3.2 Zenodo and Mosmed

3.3.3 Gold Standard

3.3.4 Double Blind

3.3.5 Discussion

Bibliography

- [1] 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>.
- [2] 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.
- [3] 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.
- [4] 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>.
- [5] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb's Journal of Software Tools* (2000).
- [6] 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.
- [7] 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.
- [8] Nico Curti. *Implementation and optimization of algorithms in Biomedical Big Data Analytics*. <https://nico-curti2.gitbook.io/phd-thesis/>. 2019.
- [9] 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.
- [10] 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>.

- [11] 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.
- [12] Hoa Nguyen Johannes Hofmanninger. *Automated lung segmentation in CT under presence of severe pathologies*. <https://github.com/JoHof/lungmask>. 2020.
- [13] 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>.
- [14] 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.
- [15] 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/>.
- [16] Travis E Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006.
- [17] 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>.
- [18] 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.
- [19] Biondi Riccardo et al. *COVID-19 Lung Segmentation*. <https://github.com/RiccardoBiondi/segmentation>. 2020.
- [20] D. Withey and Z. J. Koles. “A Review of Medical Image Segmentation: Methods and Available Software”. In: 2008.