Alma Mater Studiorum · University of Bologna

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

Automatic Pipeline for the Identification of Ground Glass Opacities 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

Contents

Introduction

Chapter 1

Ground Glass Identification Pipeline

Since the end of 2019, COVID-19 has widely spread all over the world. Up to now the gold standard for the identification of the pathology is the RT-PCR even if it is reported that its sensitivity might not be enough for COVID-19 identifications [2] and requires a lot of time to provide results.
Has been observed that several chest CT scans collected from COVID-19 patients shown bilateral patchy shadows or ground glass opacity (GGO) in the lung [2] [**ART:Wang**], which makes interesting tp investigates this technique to help diagnosis, monitoring the course of the disease and check the recovery of healed patients, since the GGO pattern may change according to the state of the disease [5]. Austin in Glossary of terms for CT of the lungs [4] 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. This kind of lesion is not exclusive of COVID-19 but can be associated to many other disorders like edema, bacteria infection or alveolar haemorrhage. However the study of the particular patterni in combination with other techniques may help early diagnosis of this pathology and the monitoring of the recovery, has shown by [5], since the GGO pattern and the involving of lung parenchima changes according to the severity of the disesase and the recovery stage.

So becomes very important to identify this kind of lesions for the reasons given before. Up to know the identification of these lesions is made mainly by manual or semi-automatic segmentation, both of them are time consuming, error prone and subjective, since require the interaction of specialized operators. To overcome this issues an automatic way to obtain these information is desirable, since allows to obtain measures that do not depend n operator subjectivity; moreover it is desirable to obtain segmentation results in a small amount of time, which is not compatible with manual or semi-automatic segmentation.

In this chapter I will describe in details the implementation of a segmentation pipeline which allows a fast and automatic segmentation of GGO.

1.1 Pipeline Description

The aim of this work is to develop a pipeline which provides a good segmentation in a small amount of time; in order to achieve these aim, the pipeline must have the following characteristics:

- **Fully Automated:** to remove the dependency from an external operator, and so the subjectivity of the segmentation;

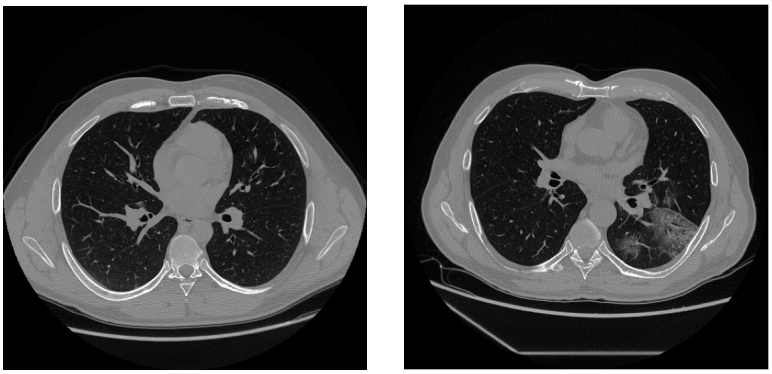


Figure 1.0.1: *CT scan of thorax for an healthy patient(left) and a COVID-19 affected one(right) in which we can observe a huge amount of GGO in the right lung*

Figure 1.1.1: *Groudf Glass Opacities 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 [***DATA:MOSMED***] dataset*

- **Fast:** in order to compete with certified software and to provides a segmentation in few minutes.

During the developing, the first problem we have to face was the lack of information: since COVID-19 is new disease, there aren't much data available. I've worked mainly on CT scans of 83 different patients provided by sant'Orsola hospital with manual segmentation, but also public dataset like MOSMED and ZENODO where used as benchmark. For each patient was available a manual segmentation, but even if some label seem to be of good quality and accurate, some other will present some errors and misclassified areas. In the end low number of patients and the quality of labels have discouraged us to use a supervised learning approach like classifiers or neural networks, an other reason is that a supervised approaches like Artificial Neural Network are computationally expansive and requires a lot of hardware resources and time.
In the end a completely unsupervised technique was used, which doesn't requires to provides already segmented images as prior knowledge.
An other thing to takes into account during the developing, is that the lesions may have different patterns according to the stage of the disease or the patients as in Figure **??** and usually these patterns are spatially disconnected, so to perform the segmentation a pixel classification techniques was used.

In the end the basic idea was to use the Color Quantization as medical imaging segmentation , which aims to to identify the different type of tissue and lesions by grouping them by color similarity. In particular we aims to assign to each structure inside the lung a characteristic colors and label each voxel by identify it as belonging to the tissue with the most similar characteristic color. This approach is justified since exist a relation between the kind of tissue and the color used to display it in a CT scan, given by Hounsfield Unit.

Since it is unlikely to find a structure with a single voxel extension, I've used the multi-channel characteristics of digital images to takes into accounts also the neighbouring voxels.

In this section I will describe how color quantization works for image segmentation, how the color space was build in order to incorporate also neighbouring information and the final structure of the segmentation pipeline.

so we have to associate each pixel color to a particular tissue and, as we will see in the section below this will be done by using the Hounsfield Unit. In particular we will find the characteristic color of each lung tissue and assign each pixel to the tissue of the most similar characteristic color.

In this section I will discuss have applied the color quantization and I will describe the main structure of the developed pipeline.

1.1.1 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 **??**.

Figure 1.1.2: *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 [8].

In this work I've applied this technique to segment CT scans of patients affected by COVID-19. Use this technique as medical image segmentation implies that each different tissue is assigned to a particular color(properly it is a range of colors since the image is affected by noise and also the tissue may not have the same density in each point)so must

exist a relationship between the kind of tissue and the color used to represent it. In case of CT scans this relation is given by the Hounsfield Units(HU) : voxels colors are proportional to HU, which are defined as a linear transformation of the linear attenuation coefficient(μ). HU normalize the μ of a particular tissue according to a reference one, usually water(μ_{H_2O}), ss we can see in equation 1.1 :

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

Where μ_{H_20} is the linear attenuation coefficient of the water, μ is the linear attenuation coefficient of the tissue in the voxel and k is a multiplicative constant, which can be 1000 or 1024 depending on the manufacturer of the CT scan. In the end each color results proportional to the linear attenuation coefficient, different from each tissue, so exist a relation between the GL and the tissue type which is exploit by its intensity, that makes this techniques available.

Color quantization and the properties of digital images allow us to consider also other properties of the image besides the single voxel intensity. This purpose can be achieved by building a suitable color space:
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. Gray scale images requires only one channel, so each pixel has a numeric values whose range may change according to the image format. On the other hand color images requires 3 channels, and the value of each channel represent the level of the primary color stored in this particular channel, so each color is represented by 3 different values, according to Young model.
In our case we have decided to use more than 3 channels and each one of them incorporates different information about the image lijke features like image edges and median neighbouring voxel intensity. This information are useful since lesions areas involves many closest voxel, 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 centroid estimation a simple kmeans clustering was used, since it provides a suitable segmentation with good time performances and it is efficiently implemented for multi-channel images in OpenCV [6]. Kmeans clustering requires a prior knowledge about the number of cluster, which in our case is given by the anatomical structure of the lung, so each cluster will correspond to a different anatomical structure:

- Total lung parenchima;
- Bronchial and vessels;
- **GGO**, which is the cluster in which we are interested in;
- Eventual noise;

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: int this way the estimation step, that we will call "training", needs to be performed only once, so can be time expansive since is not involved in the actual segmentation.

1.1.2 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. As I've said before the pipeline

is divided into two main phases, one for the centroids estimation, which is performed only once, and one of the actual segmentation, which is performed at least once for each scan to segment. Before each of these step we need also a preliminary step to achieve pre processing and a lung region isolation, useful to focus the segmentation in the region of interest and to avoid false positives.
In the end the pipeline structure is divided in three main blocks as we can see in Figure **??** :

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

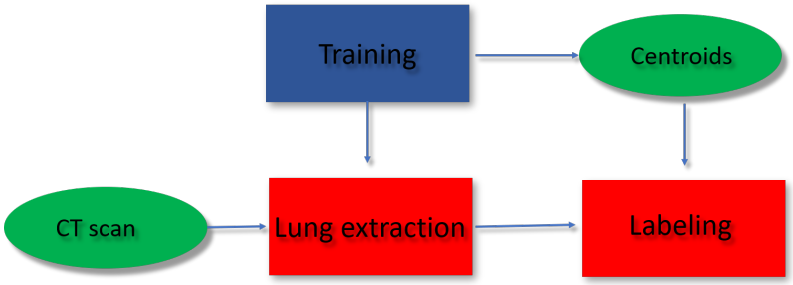


Figure 1.1.3: *Flow chart of the main structure of the developed pipeline. The training process, which allows the estimation of the centroids, is perfromed only one time.*

Pre Processing and Lung Extraction

This preliminary step is performed before both training and labeling.
First of all performs a registration of the HU on a common space, in order to overcomes 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.
This process is followed by a segmentation fo the lung regions, which allows to remove all the extra lung regions avoiding the formation of false positives. During this process a particular attention was paid on the removal of the main main bronchial structures, which can interfere with the actual segmentation, and the preservation of the lung regions which are the ones in which we are interested in.

Training

This step involves the estimation of the centroids for each tissue. This step requires different patients, in order to have a good statistical representation of each structure, this the kmeans requires an homogeneous representation of each cluster. This makes the entire process time consuming and computational expansive,but, as I've said before, this process need to be performed only once, so isn't involved in the actual segmentation and doesn't affect the segmentation time.
The implementation of this step involves the building of the multichannel image, to takes into account the different features of the images like neighborhooding voxel intensity or edges, the managing of the over represented clusters like the background and the actual centroids estimation via kmeans clustering. All of these steps will be discuss in details in the next session.

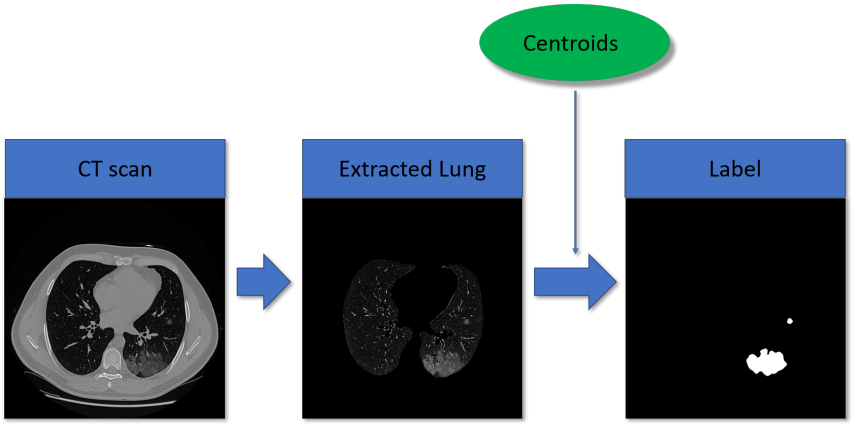


Figure 1.1.4: *Actual segmentation step, from left to right we can see the input image stack, the isolated lung regions and the final label. To performed the labeling a set of pre-computed centroids was used.*

Labeling

This step involves the actual segmentation. The script which perform it requires as inputs the image 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 end of the procedure only the cluster corresponding to the GGO are provided. 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. To found the closest centroids we simply compute the euclidean distance between each voxel and each centroids and assign the vocxel to the closest cluster.

In the end, once we have estimated the centroids, the segmentation pipeline will results of only two steps, as shown in Figure **??** , in which we can observe the flowchart of each step with an image that shown the partial results.

1.2 Pipeline Implementation

In this chapter I will go into details about the actual pipeline implementation. First of all I will briefly describe the developing environment and the libraries used for the code implementation. After that I will describe how each step of pipeline 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 o input and output images and the other operation I've aminly used OpenCV [6] and SimpleITK.
Since python is an high level lenguage, it allows an easy and fast implementation of the ode, 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 [7]an 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 ... . 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 medical software like 3DSlicer.

1.2.1 SimpleITK and OpenCV

To perform image processing operations, like application of blurring filters or morphological operations, and to perform the color quantization I've used two different libraries, which provides a large collection of optimized tools for medical image processing and computer vision. Each tool was optimized, so it will help to takes the segmentation times as short as possible.

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 tolls 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 perform the color quantization, by performing the kmeans clustering on a multi channel image.
This library allow to works only on a single image, so each operation must be repeated for each image of the stack. This means that we are working in a 2D, since each slice is processed independently.

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 Toll 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 SimpleITK consider the image like spatial object and not like arrays of values, it store also infomations about voxel spacing, size and origins, provided as wall 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. This library allow also to process the whole image volume, allowing 3D operations, which are used into many steps of the pipeline.

1.2.2 Lung Extraction

Lung extraction is the first step of the pipeline that is performed by homonymus script. As input the CT scan to process in each format supported by SimpleITK and will provide as output the isolated lung regions as default in '.nii' format.
To achievement of the lung extraction involves 3 main steps:

1. **Pre-Processing** : Which register the HU in a common space, crop the outliers, and denoise the image;

2. **Thresholding and reconstruction** : Which is the actual segmentation, that takes care to preserve all the intra-lung regions exept for the main bronchial structures.

3. **Selection of lung** : allows the exclusion of all the extra-lung organs like intestine.

Pre-processing

As I've said before, 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 the padding values, to obtain a square $N \times N$ image for each slice, are different according to the scan, for instance in the CT scan in Figure **??**(a) the padding value is $-3000HU$ and the air value is -1024. So the first thing to do is to make the padding value and the air value equal for each scan considered and shift them to 0, because for the de-noising operation we need to works with unsigned int 16-bit gray scale image.
May also happen that some Hu are out of range, that because some patients may have metallic prosthesis that make the so called *metallic artifacts*. However we haven't to worry about this kind of artifacts because they will involves only body regions and not the lung ones,and so are removed during this step. In Figure **??**(a) we can see the histogram of a CT scan before the pre processing, were it is clear the peak around $-3000\,HU$ of the padding values and the peck around $-1024\,HU$ of the air value. In Figure **??**(b) we can observe the histogram of the same scan after this pre-processing step, it is clear that both the padding and air values are set equal and shifted to zero, making positives all the other units.

Before starting the actual lung segmentation, we need to performs a de-noising operations, which allow us to increase the differences between the GL of the body regions, and to sharp the edges. According to the procedure described in [1], I've used the bit plane slices. This approach allows to use the way in which each numerical values is stored in the computer by converting each values with its binary representation, this allows us to construct an image in which each voxel intensity is given only by the bits representing the regions multiplied by their significance.

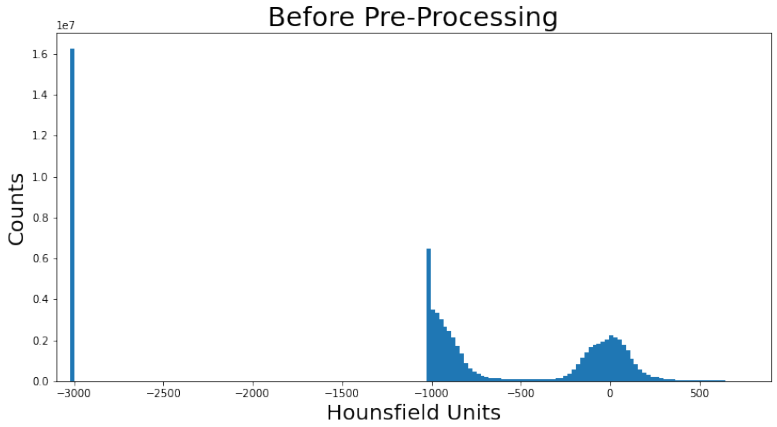
In Figure **??** are displayed the images for each bit. As we can see all the bits from the 1st to the 11th doesn't carry any useful information but only noise, on the other and the bits from the 12th to the 16th aren't used in the image representation. In the end the de-noised image is constructed by using only the 9th, 10th and 11th bits times their significance that is defined as $significance = 2^{bit}$ In this way we have constructed an image in which the noise is highly reduced and the different regions are well separated.
After these procedures the scan is ready for the actual lung segmentation.

Thresholding and reconstruction

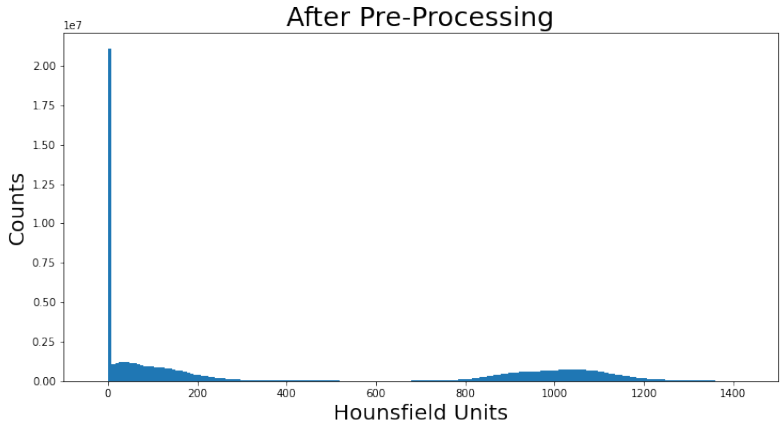
For the actual lung segmentation we have found that the most suitable technique is simply a global fixed threshold, since now the GL of the different body regions are well separated. This allow us to include all the intra-lung regions including the GGO, which are usually dropped if an adaptive threshold, like otsu, is performed. However after this process also extra lung regions, like intestine are segmented as lung. This is the reason why we need the last step to select only the lung.

Lung Selection

In Figure **??**(a), I've reported a 3D reconstruction of the selected regions after the second step. As we can see the intestine is selected with the lung, which are connected by trachea and bronchi.



(a) Histogram of a CT scan before registration



(b) Histogram of a CT scan after the registration

Figure 1.2.1: *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.*

On the other end the intestine is disconnected from the lung regions, since are structures anatomically disconnected. So we can use this characteristics to select only the lung. To perform this step I've used a suitable function from SimpleITK, which allow us to found the connected components by considering the whole image tensor, this means that we are able to found the connected components by considering all the 3 dimensions. At the end of the process, the components with the higher volume is the one corresponding to the lung and trachea regions, so we can simply select this component to achieve a correct selection of lung, as we see in Figure **??**(b).

In this way a lung mask is created and its applied to the pre-process image. THe results will be saved in '.nii' format, which allows to preserve the spatial information about the voxel.

1.2.3 Training

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

To achieved the estimation of centroids, a kmeans clustering of the multichannel images of several CT scan from different patients is performed. Since to achieve a correct estimation a huge amount of scan must be provided, this task is time consuming an computational expansive, however is performed only once and isn't directly involved in the actual segmentation, so doesn't affect the segmentation time.
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 informations 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;
- Median Blurred;
- Edges
- Standard Deviation map.

In Figure **??** I've displayed the 4 different channel of the image. The pure image will provides information about the tissue displayed in the single voxel; the median blurred allow us to consider information about the tissue surrounding each particular voxel, since lesions usually involves several group of voxels. The std map consist into the replacement of each voxel alue with the standard deviation of its neighborhood. This allow us to discriminates between the various regions and the edges, since we assume an intra-region homogeneity. Moreover this channel allow us to better discriminates bronchial regions from lesions, since bronchi have an elongated and thin structure that read in an higher std value on the neighborhood. In the end the last channel is a median blur of the edge map, which allow us to enhance the lesion regions, since presents an high number of edges.

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 subsamples. The creation of several subsamples 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 subsamples and cluster them independently, after that a clustering on the estimated centroids is performed.

Clustering

This step consist into the performing of the kmeans 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 subsaple, 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 kmeans ++ algorithm, which allows to improve speed and accuracy of the clustering algorithm [3]. During this task we have to manage some issues. As we can see from Figure **??** the number of voxel with $GL = 0$ is several order of magnitude higher than for other GL. As prior we know that these voxels belongig from background, so this cluster is over rapresented. Since kmenas cluster requres na homogenous 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. Kmeans 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 4 cluster are an optimal choice, and the considered structures are the following:

- Lung Parenchima;
- Ground Glass Opacities;
- Bronchi;
- Eventual noise and motion artifacts.

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 pseudocode of this script is reported in algorithm **??**

1.2.4 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 sum of square :

$$cluster = \arg\min_{S} \sum_{i=1}^{k} \sum_{S} \|x - \mu_i\| \tag{1.2}$$

where x is the color vector of the voxel and μ is the $i_t h$ centroids. During this process the background is automatically assigned to the 0 label, since we know as a prior that its value is 0.
Once this process is done, we can select only the label corresponding to the GGO, and our segmentation is achieved. After this step a refinement is performed a simple median filter is applied to each image of the stack, which allows the removal of misclassified voxels. The pseudocode of this script is reported in algorithm **??** I've tested this algorithm on three different dataset, and the results are described in the following chapter.

Algorithm 1: 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)
```

Algorithm 2: 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
```

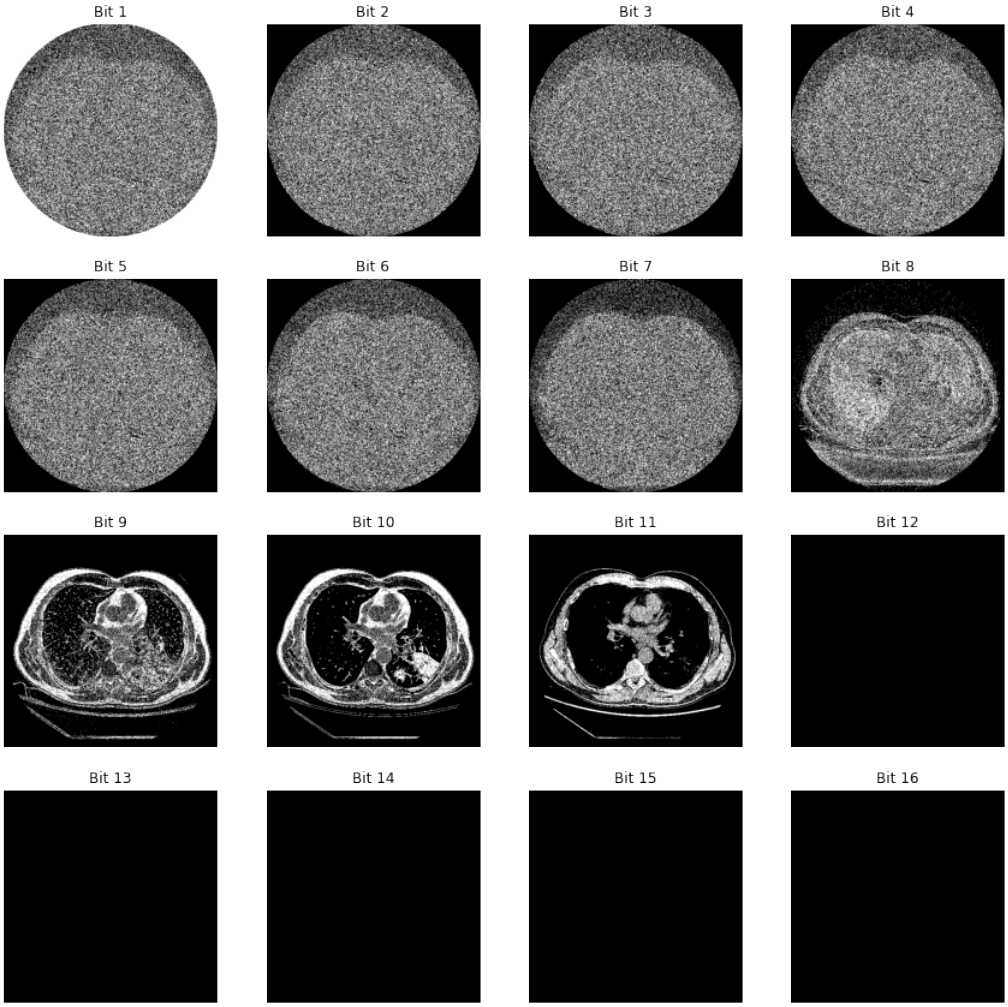
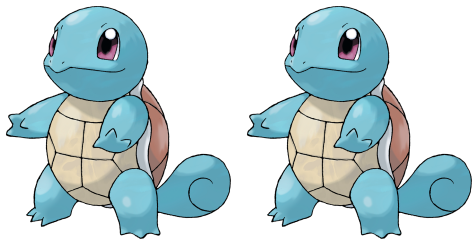


Figure 1.2.2



(a) (b)

Figure 1.2.3

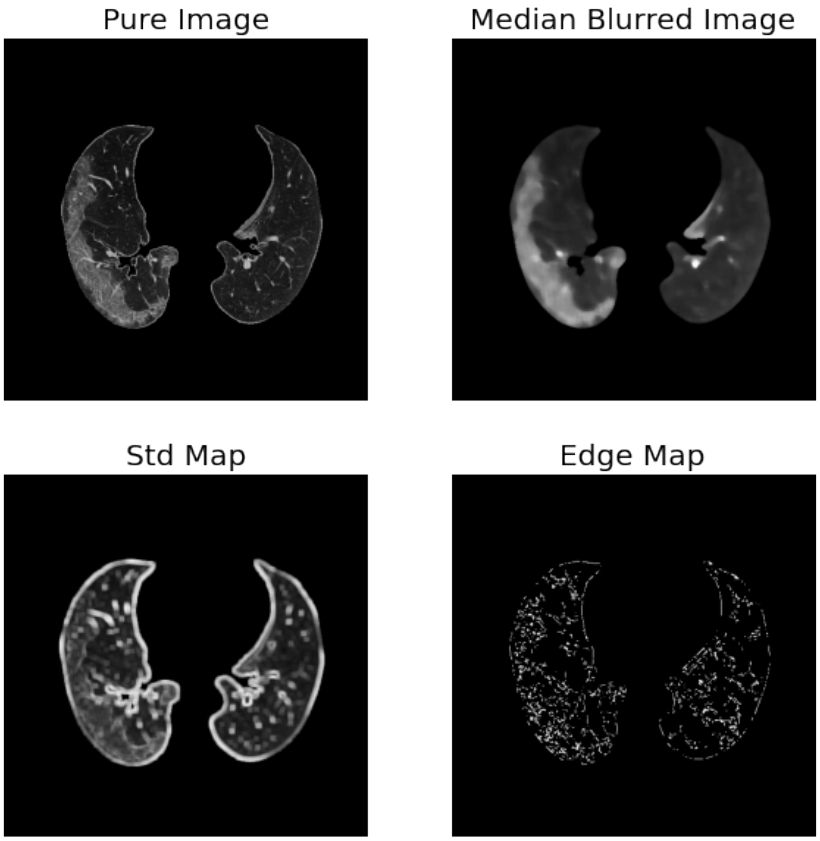


Figure 1.2.4

Chapter 2

Results

In this chapter I will present the pipeline results. First of all I will describe the characteristics of each of the dataset used as benchmark, after that I will discuss the results about time performances and accuracy of the developed method.
The centroids used for the segmentation were computed using the 83 CT scans kindly provided by Sant'Orsola hospital. This centroids were used to segment both healthy patient and patient affected by COVID-19, in order to verify that no GGO regions were found in helathy patients and the GGO are correctly identified in the other scans. To achieve this purpose I've match the labels estimated by the pipeline with some manual segmentation made by expert radiologist.

2.1 DataSet Description

In this section I will describe the main characteristic of the used dataset. The dataset are described by providing the general image characteristics, the sample used and the segmentation modes. If presents also some metadata statistics were given.

2.1.1 Sant'Orsola

Sant'Orsola data was the ones mainly considered in this work. The consist into 83 anonimysed CT scans from 83 different patients affected by COVID-19, manually labeled by interns; and 5 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

2.1.2 MOSMED

MosMed is a dataset which contains 1110 anonymised CT scan of human lung from both patients affected by COVID-19 in several stages fo the disease, and healthy constrols. 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 cathegories, from healty patients to the most several cases. Each scan of the dataset is saved in *.nfti* format and diring

the conversion from the original dicom series only 1 image every 10 was preserved. The resulting dataset have the following characteristics:

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

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

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

Of these five cathegories only 50 annotations are available, mostly invlves only the patients of CT-1 group, which is the only one used for the performances checking, since 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.

2.1.3 ZENODO

This dataset consist into 20 CT scans of patients affected by COVID-19, labeled by two expert radiologist and verificated by and expert radiologist. The anatomic sturctures labeld are the left and right lung and the infections regions. Each files is in '.nii' format and no metadata was available.

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

2.2 Time Performances

2.3 Accuracy Comparison

In order to check the performances of the pipeline, I've performed a comparison between the areas estimated by the pipeline itself and the one belonging from the manual segmentation. In order to match the labels and found how much they are according, I've used the Intersection over Union metrics, also known as Jaccard score. The intersection over union is a very suitable metrics in this case, mostly because allows to overcome the issue of the over-represented background:
Since the number of voxel of the non-GGO is several order of magnitude higher than the GGO ones, metrics like Pixel Accuracy will fails. IoU encodes the shape properties of the objects under comparison, e.g. the widths, heights and locations of two bounding boxes, into the region property and then calculates a normalized measure that focuses on their areas (or volumes); in this way the IoU is invariant to the scale of the considered problem.

The Intersection over Union is defined as follows:

$$IoU = \frac{\|A \cap B\|}{\|A \cup B\|}$$

Where A, $B \in S \subseteq R$ are, in our case, the estimated labels and the ground truth.

An other test that was made was to segment scans from healty patients and check that no GGO was detected. This kind of test was made on the 5 healthy patients from Stant'Orsola and on the $CT-0$ group of MOSMED dataset.

Healthy Control

MOSMED

ZENODO

Sant'Orsola

Chapter 3

Review on Image Segmentation techniques

Image segmentation consist in the partitionng of an image into non overlapping, consinstent 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 plays a crucial role in many medical-imaging applications by automating or facilitationg 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 foundamental 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 shuch as clustering or thresholding.

3.1 Review on Image Segmentation Methods

During the years, several segmentation methods have been developed based on a lot of different approaches These metods 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.

3.1.1 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 differetiate between tissue regions represented within the image [**INP:Withey**]. Figure **??** show an histogram of a scalar image with two classes, threshold based approach attempts to determine an intensity value, called *threshold* which separate the desidered classes [**ART:Pham**]. So to achieve the segmentation ve can group all the pixels with intensity higher than the threshold in one class an all the remaining in the other class.

Figure 3.1.1: *Caption*

The threshold value is usually setting by visual assesment, 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 multiclass 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 contranst in intensities. Threshold doesn't takes into account the spatial characteristic if the image, so it is sensitive to noise and intensity inhomogeneities, that corrupt the image histogram of the image and making difficult the separation [**ART:Pham**]. To overcome these difficulties several variation of thresholding have been proposed besed on local intensities and connectivity.
Threshold is usually used as initial step in sequency of image processing operations, followed by other segmentation technique that improve the segmentation quelity. Since threshold use only intensity information, can be considered a pixel classification technique.

3.1.2 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**].

Figure 3.1.2: *Caption*

3.1.3 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 occurence of each class. Generally this quantity aren't known, so we need a set of training data to usa as reference.
There are different techninques wich 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 quantificable features. It is computational efficent and can be applied to multichannel images. This approcach 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.

3.1.4 Clustering

Clustering approach is similar to classifiers one but in an unsupervised faishon, so doesn't require a training dataset. Clustering iteratively alternate between segmenting tha image and characterizing the proprieties 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 segmentats the image by classifying each pixel in the class with the closest mean;
- **Fuzzy C-means:** this algorinthm 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 Maximul 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 [**ART:Pham**].

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

3.1.6 Artificial Neural Networks

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.

3.1.7 Deformable Model

Deformable Model use an artificial, closed, contour/surface able to expand or contract over time and conforme 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 smootness constraint that providesrobustness to noise and spurioous edges.
However this approach requires a manual interaction to place the appropriate set of parameters.

Bibliography

[1] et al. A A Abdullah. "Lung Cancer Nodule Extraction and 3D Modeling using Bit-Plane Slice and Outlining". In: *International Journal of Engineering Research Technology (IJERT)* 5 (Sept. 2016). ISSN: 2278-0181.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] Riccardo Biondi Nico Curti Enrico Giampieri Gastone Castellani. *COVID-19 Lung Segmentation*. https://github.com/RiccardoBiondi/segmentation. 2020.

[8] Celal Ozturk, Emrah Hancer, and Dervis Karaboga. "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.