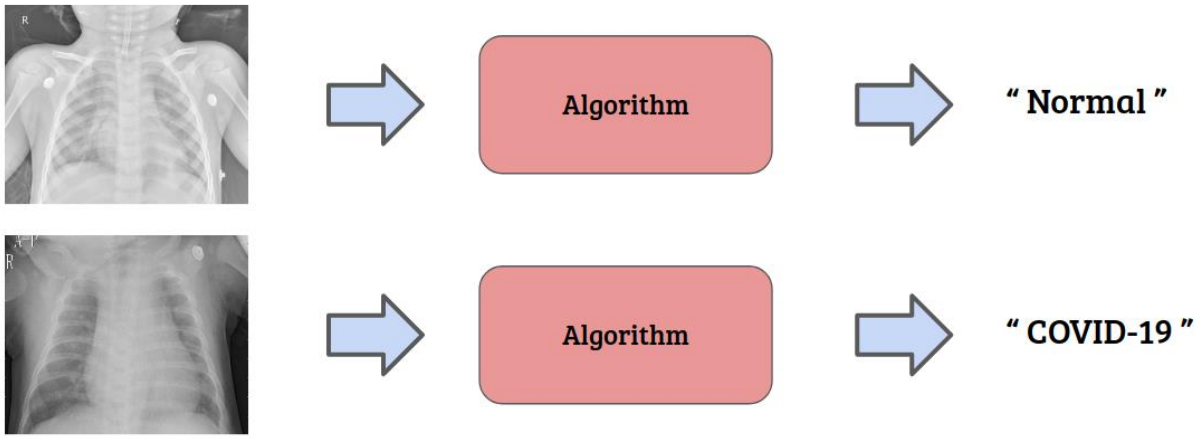
Advanced Algorithms Project

1. **Rules:** the project is not mandatory. Students who choose to carry out the project will discuss the project during the oral exam. The project is individual, and each student is required to develop an original solution which should be properly justified and sustained. The student is required to produce a report of at least 1/2 pages (plots excluded from the page count) which will include an explanation of the design choices, the issues and performances encountered during the development of the project. In addition, the student is required to provide the source code. Report and code could be sent in a zip file. In alternative, students can produce a Colab notebook with runnable code, akin to what you have seen during the laboratory session. In case you opt for the latter solution, the submission consists in sharing the link to your colab notebook besides the report (sent by email).

2. **Objectives:** the main objective of this project is to create an algorithm able to identify, given a thoracic x-ray image as input, the potential presence of illnesses. The illnesses that we target are 3 and are named: *viral infection, bacterial infection* and *COVID-19*. In order to start to understand the task, here is an example:



In this example we have two thoracic x-ray images from two different patients. The one on the top does not show any presence of illnesses, while the one on the bottom shows the presence of COVID-19. Given the first image as input, the algorithm should be able to determine the absence of illnesses. On the other hand, given the second image as input, the same algorithm should be able to determine the presence of COVID-19. By looking at these two examples, we can immediately see that this is not an easy task. Despite the differences between these images seem to be very subtle, there are some meaningful features that can be used to distinguish among them.

3. **Data:** we have ~6000 thoracic x-ray images, each one has a 256x256 pixels resolution. These images have been obtained on people affected by one of the three illnesses or

people without any illness. As usually done when developing such an algorithm, we need to make some considerations:

A. **Subdivision of the data into splits:** as said before, we have ~6000 images at our disposal. Despite this fact, we cannot use them all just to develop our algorithm, but we need to reserve a part of them in order to test the algorithm performances. For this reason, we will divide the ~6000 images into three splits: *train*, *validation* and *test*. The *train* and *validation* splits will be given with their corresponding ground truth, meaning that for each thoracic x-ray image we will also know its label. On the other hand, for the *test* split, only the images will be given. The *train* and *validation* splits will be the two splits that you can use to develop and improve the algorithm, while the *test* split will be used in order to verify the performances of the algorithm on data on which the algorithm has not been trained on and has never seen before.

   This table shows statistics about the three splits.

Split / Class	Normal	Bacterial	Viral	COVID-19	Total
train	951	1664	909	45	3569
validation	313	553	303	20	1189
test	319	569	292	11	1191
total	1583	2786	1504	76	5949

   Given the ~6000 images, we consider ~60% for *train*, ~20% for *validation* and the remaining ~20% for *test*.

B. **Number of images for each class:** as you can see from the table above, the number of images for each class is fairly different. In particular, the number of COVID-19 images is very limited. This is a very common issue which is usually found in many datasets, also known as *class imbalance*. In order to obtain the best results, it will be very important to take this class imbalance into account.

C. **Additional data:** in order not to limit the choice of the algorithm to use, we will provide you with some additional data. On top of the images, we will also provide a set of *features* which have been extracted and will be potentially meaningful in order to solve the task.

4. **Where to find the data and template for report**: the data and the template for writing the report is available through this google drive folder.
   In the “report_template” folder you will find the Latex report template. Please open the “egpaper_final.tex” file with any Latex editor you like (e.g. the open source TexMaker) and modify it. The document that needs to be submitted as report is the PDF version of that document.

In the “project_data” folder you will find 3 sub-folders, one for each split. In the *train* and *validation* folder, the data will be divided into 4 sub-folders, one for each category (*normal*, *bacterial*, *viral* and *COVID-19*). The data is composed of three parts: images, labels and features. In the test folder you will only find the images and features. Features are given as .npy files, which you can load using the numpy.load function (ref).
In order to load the data into Colab follow this guide https://colab.research.google.com/notebooks/io.ipynb.

5. **Methods to use:** you can use any of the methods which have been taught during this course. We encourage you not to limit the solution to just one method, but instead to try out different ones in order to better understand their strengths and limitations and report results obtained by each method.

6. **Note of caution:** COVID-19 is a hot topic these days and we are aware of the presence of many repositories online targeting a similar task. We encourage you not to look at those projects, and to come up with an original solution. We are aware of these repositories and if your solution will be found to be similar to a solution found online, the project will be declared invalid.

7. **How to measure the performances:** the objective of the project is to obtain the highest accuracy on the *test* set. However, you will not have access to the ground-truth for the test data, but can estimate the generalization performance on the available validation data. The metric that we will use takes into account the accuracy of each category and, in order to obtain good performances, the algorithms need to be accurate on all of them. In particular, the overall accuracy will be called $\mathbf{A_{tot}}$ and will be computed as the average of the accuracies on each category. Since we have four categories, it will be

$$A_{tot} = \frac{1}{|C|} \sum_{c \in C} A_c$$

where **C** is the set of our four categories, $|C|$ is the cardinality of the set (4 in this case) and $\mathbf{A_c}$ is the accuracy on each category, computed as

$$A_c = \frac{TP_c}{TP_c + FN_c}$$

where $\mathbf{TP_c}$ is the number of images which have been *correctly* assigned to category “**c**”, while $\mathbf{FN_c}$ is the number of images which have been *incorrectly* assigned to category “**c**”.

8. **How to measure the performances on the test set:** since you don't know the labels of the test set, you will need to send us the predicted class for each image included into the test set. Your test set accuracy result will be sent back to you. In order to motivate you we will also keep a leaderboard with all the scores. We can keep it anonymous if you don't want to show your name. We will limit the number of test submissions to 3. The reason we do this is that you should try to optimize your algorithms relying only on *train* and *validation* as much as you can, and only at the very end try to evaluate the performances on the *test* set. In order to have a standardized way of sending your test set results, please provide your prediction as a single txt file, with a line for every image. Each line will contain two items, the image_id and class_name. Just to be clear, produce and send a file that looks like this:

    0000 COVID-19
    0001 Normal
    0002 Viral
    . . .
    1190 Normal

    The total number of lines in this file has to be the number of images in the test set which is 1191, the image_id should go from 0000 to 1190, the set of possible class_names is {Normal, Bacterial, Viral, COVID-19}.