

Our solution employs custom segmentation models, a public classification model, and public lung segmentation models.

The solution can be split into 4 steps.

Step #1: classification for the presence of lung opacity by a public model, taken from torchxrayvision [1] package.

Step #2: a mix of 3 segmentation models. They are trained on public data: QataCov19 [2], a Covid dataset from Mendeley [3], VinDr[4], and RSNA Pneumonia [5]. The first model is a ConvNext-based[6] FPN[7] network, initialized from CLIP pretrained on in-house image-text paired Xray dataset. As an input, it takes equalized images resized to (336, 336). Two other models are nnUNet [8], trained on different resolutions (336 and 768) and with different mix of models.

The first ConvNext-based model is trained on QataCov19, Mendeley segmentation, and RSNA Pneumonia datasets, where the bounding boxes are filled to obtain the masks.

The 768 nnUNet was trained in two steps: first, using the segmentation datasets QataCov19 and Mendeley, and then, on the data pseudo-labeled by the model from the first step (VinDr), the bounding boxes from which are used to refine predicted segmentation masks.

The 336 nnUNet was trained on QataCov19 and Mendeley data, but with a randomly applied histogram equalization.

Steps 2 and 1 provide 4 classification models. Their average prediction reveals whether there are signs of lung opacity in the image or not. If not, we proceed with the lung post-processing; if yes, we move to steps 3 and 4.

Step #3: the masks of 3 segmentation models are joined. Connected components from these masks are extracted and processed by another segmentation model, also nnUNet, which was trained on the crops containing lung opacities. This step refines previous masks, as they were obtained in a relatively low resolution.

Step #4: we then blend 4 segmentation outputs and post-process by lungs. We treat each segmentation model as an expert. There are several approaches to blend the intermediate outputs.

Approach 4.1 - «Zones»:

To smooth the values of the output probability map, we assign different values to the regions marked by the four consecutive runs of the four models. The values in these regions are optimized during the internal validation.

Approach 4.2 - «Weighted»:

We take the probability maps and sum them up together with the weights that are inversely proportional to their independent weighted log-loss values (evaluated on

14 studies from the calibration phase). This provides us with a smooth probability map.

After that, to smoothen the small probabilities further, we apply post-processing by a dilated lung mask, increasing the values for the pixels inside this mask. We run this twice with different values for the lung pixels.

- [1] - <https://github.com/mlmed/torchxrayvision>
- [2] - <https://www.kaggle.com/datasets/aysendegerli/qatacov19-dataset>
- [3] - <https://data.mendeley.com/datasets/36fjrg9s69/1>
- [4] - <https://www.nature.com/articles/s41597-022-01498-w>
- [5] - [https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge/data?select=stage\\_2\\_detailed\\_class\\_info.csv](https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge/data?select=stage_2_detailed_class_info.csv)
- [6] - <https://arxiv.org/abs/2201.03545>
- [7] - <https://arxiv.org/abs/1612.03144v2>
- [8] - <https://github.com/MIC-DKFZ/nnUNet>

Submission 1: «Zones» with higher values for lung pixels (0.3 on positive samples, 0.1 on negative samples, background is always 0.05)

Submission 2: «Weighted» sum of probabilities + lungs post-processing as in Submission 1.

Submission 3: «Zones» with lower values for lung pixels (0.15 on positive samples, 0.05 on negative samples, background is 0.05 on positive and 0 on negative samples)