Research on Biomedical Engineering
https://doi.org/10.1007/s42600-022-00242-y

ORIGINAL ARTICLE

# Covid-19 detection using chest X-rays: is lung segmentation important for generalization?

**Pedro R. A. S. Bassi**[1,2,3] · **Romis Attux**[1]

Received: 18 January 2022 / Accepted: 21 September 2022

## Abstract

**Purpose** We evaluated the generalization capability of deep neural networks (DNNs) in the task of classifying chest X-rays as Covid-19, normal or pneumonia, when trained in a relatively small and mixed datasets.

**Methods** We proposed a DNN to perform lung segmentation and classification, stacking a segmentation module (U-Net), an original intermediate module and a classification module (DenseNet201). To evaluate generalization capability, we tested the network with an external dataset (from distinct localities) and used Bayesian inference to estimate the probability distributions of performance metrics. Furthermore, we introduce a novel evaluation technique, which uses layer-wise relevance propagation (LRP) and Brixia scores to compare the DNN grounds for decision with radiologists.

**Results** The proposed DNN achieved 0.917 AUC (area under the ROC curve) on the external test dataset, surpassing a DenseNet without segmentation, which showed 0.906 AUC. Bayesian inference indicated mean accuracy of 76.1% and [0.695, 0.826] 95% HDI (high-density interval, which concentrates 95% of the metric's probability mass) with segmentation and, without segmentation, 71.7% and [0.646, 0.786].

**Conclusion** Employing an analysis based on LRP and Brixia scores, we discovered that areas where radiologists found strong Covid-19 symptoms are the most important for the stacked DNN classification. External validation showed smaller accuracies than internal, indicating difficulty in generalization, which is positively affected by lung segmentation. Finally, the performance on the external dataset and the analysis with LRP suggest that DNNs can successfully detect Covid-19 even when trained on small and mixed datasets.

**Keywords** Covid-19 detection · Layer-wise relevance propagation · Lung segmentation · Deep neural networks · Bayesian inference · Chest X-rays

## Introduction

Diagnosis is an important aspect for controlling Covid-19 spread and helping infected patients. Nowadays, an active SARS-CoV-2 infection is normally diagnosed with the detection of its ribonucleic acid (RNA) genome (usually employing reverse transcription–quantitative polymerase chain reaction, RT–qPCR, or, alternatively, using next-generation sequencing, NGS, or isothermal nucleic acid amplification assays) or with antigen tests, which assess the presence of viral proteins (Mercer and Salit 2021). Having higher sensitivity (50–70% in a real clinical scenario) and specificity (~99%), to this date (August 2022), the gold-standard Covid-19 diagnosis method is RT-qPCR (Wang et al. 2020; Mercer and Salit 2021). However, this method is expensive, requires a considerable amount of time, and is at high demand during infection peaks.

X-ray is one of the cheapest and most available Covid-19 alternative detection methods, mostly when the disease spread in developing countries is considered. Covid-19 has characteristic signs, which can be observed in chest X-rays, like bilateral radiographic abnormalities, ground-glass opacity, and interstitial abnormalities (Guan et al. 2020).

✉ Pedro R. A. S. Bassi
pedro.salvadorbassi2@unibo.it

[1] Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, University of Campinas - UNICAMP, Campinas, SP 13083-970, Brazil

[2] Present Address: Alma Mater Studiorum - University of Bologna, 40126 Bologna, BO, Italy

[3] Present Address: Istituto Italiano Di Tecnologia, 16163 Genoa, GE, Italy

Published online: 02 November 2022

However, the images' analysis is not an easy task. Therefore, artificial intelligence may be able to help in the creation of a reliable system to help clinicians in this endeavor.

Deep neural networks (DNN) for Covid-19 detection were already proposed by many studies (Shoeibi et al. 2020). However, some researchers raised concerns about the possibility of bias falsely improving the reported results. Maguolo and Nanni (2020) mixed different chest X-ray datasets, removed most of the lungs from the images, and trained DNNs to classify to which dataset the images belonged. They were able to obtain high accuracies and, according to them, this study reveals that dataset biases may influence DNNs trained with mixed datasets, reducing their generalization capability. We do not think this test alone is enough evidence to conclude that the biases can strongly influence DNN decisions, because a deep neural network is a very flexible model: if the relevant information in the X-rays is deleted, it may be more prone to learn even tiny dataset particularities. However, we agree that the study proves the existence of bias in mixed datasets.

Accordingly, a review (López-Cabrera et al. 2021) has shown that if the DNNs are allowed to analyze the entire X-ray, they tend to focus on areas outside of the lungs. The study suggested that the DNNs pay attention to X-ray features that are not representative of the disease symptoms (like text outside of the lungs), i.e., they focus on image characteristics that represent bias. Analyzing these features, the DNN can achieve high accuracy on the training dataset and standard test databases, which are independent and identically distributed (i.i.d.) in relation to the training data (they present the same founts of bias as the training dataset). However, the DNNs do not properly generalize to real-world scenarios or out-of-distribution (o.o.d.) datasets, whose X-rays are gathered from external sources in relation to the training samples. This phenomenon is known as shortcut learning, and the review shows that, in Covid-19 detection, performances on i.i.d. test databases can be unrealistically high (López-Cabrera et al. 2021).

Open and large Covid-19 X-ray datasets, with all images collected from the same sources, are still not very common, making the study of mixed datasets relevant. Databases with the aforementioned characteristics represent the best-case scenario, as different classes would not present different biases. But Covid-19 classification datasets tend to be relatively small and mixed, i.e., different classes have dissimilar sources (Shoeibi et al. 2020). The objective in this study is to understand how, in a dataset like this, bias affects a DNN classifying healthy individuals, Covid-19 and pneumonia, which is a disease that also creates abnormalities in chest X-rays, such as airspace consolidation, poorly defined small centrilobular nodules, and bilateral asymmetric ground-glass opacity (Kim et al. 2002). Therefore, to remove any effect of dataset bias in our reported results, we used external testing and validation (hold-out) databases, whose X-rays were not from the hospitals that provided the training images. We can also refer to the external datasets as out-of-distribution (o.o.d.) in relation to the training database. Furthermore, we analyzed if the utilization of lung segmentation improves performance on the external test dataset, which would indicate a reduction of bias and improved generalization.

In this study, we use a large DNN, which performs lung segmentation, and then classifies the segmented images. We trained for classification with twice transfer learning, downloading ImageNet (Deng et al. 2009) pretrained classification networks, training them on a large X-ray database showing many lung diseases (Wang et al. 2017), and finally on our dataset (including Covid-19, normal and pneumonia).

We evaluated our networks with traditional performance measurements (point estimates). But, due to the small number of available Covid-19 X-rays, our test dataset is small, lowering the performance metrics' reliability for prediction of real-world behavior. Therefore, we quantified the measurements' uncertainty, using a Bayesian model (Zhang et al. 2015) to estimate the performance metrics probability distributions and their statistics, e.g., 95% high density intervals (an interval containing 95% of the metric probability mass, and whose points have probabilities that are higher than any point outside of it). We expanded the model in Zhang et al. (2015) to also estimate class specificity and mean specificity.

We employed a technique called layer-wise relevance propagation or LRP (Bach et al. 2015) to create heatmaps of the X-rays, showing which areas most contributed to the DNN classification, and which were more representative of other classes. These maps allow for a better understanding of how DNNs make decisions, improving interpretability. They also show if the proposed DNN is truly ignoring the unimportant information outside the lungs, and they allow a clearer understanding of the lung segmentation impact on classifier behavior. Furthermore, the maps may be helpful for a clinician in finding the Covid-19 signs in an X-ray and evaluating the DNN prediction.

This study presents a large DNN, containing 3 stacked modules. The segmentation module is a U-Net (Ronneberger et al. 2015), trained beforehand to receive X-rays and output segmentation masks (images that are white in the lung regions and black everywhere else). Afterwards, we utilize an original intermediate module, which uses the U-Net output and the input image to erase the unimportant X-ray regions, and performs batch normalization. Finally, the classification module, a 201-layers dense neural network (Huang et al. 2017), returns the probabilities of the X-ray containing healthy lungs, pneumonia, or Covid-19. A common DenseNet201 (without segmentation) is employed for comparison.

This work introduces a new technique to compare DNN's analysis of Covid-19 X-rays to radiologists', which is based on LRP and X-rays scored with the Brixia scoring system. The Brixia score is a methodology created for radiologists to semi-quantitatively score Covid-19 severity in six lung zones (Borghesi and Maroldi 2020). Please refer to "The Brixia score" section for a detailed explanation of the scoring system. Based on LRP heatmaps and the Brixia score, we will answer the following questions: do DNNs and radiologists look at the same Covid-19 signs? Is there a correlation between areas where radiologists find more severe symptoms to areas with more relevance in heatmaps? Do DNNs predict higher Covid-19 probabilities in X-rays that radiologists considered more severely affected by the disease?

The main contribution of this paper consists in a profound analysis of the effects of mixed datasets and lung segmentation on generalization in the field of Covid-19 detection, using a test dataset created by external sources (with respect to the training dataset). Novel aspects of the analysis are the utilization of Bayesian inference to estimate the performance metrics probability distributions and the comparison of LRP heatmaps with X-rays analyzed using the Brixia score. Furthermore, we suggested a modular DNN architecture, composed of two state-of-the-art DNNs and an original intermediate module. The proposal is flexible: researchers can utilize just our trained segmentation module, along with the intermediate one, attach it to an alternative classification network and train for classification. This can provide a simple and fast way to create other DNNs that perform segmentation and classification, thus, our trained DNNs are available for download at https://github.com/PedroRASB/Covid-19-detection-with-lung-segmentation.

## Methods

In this section, we explain the employed datasets, the data processing and augmentation procedures, the deep neural networks, their training schemes, and, finally, the LRP strategy and Bayesian model used to analyze this study's results.

### The source databases

In sections "NIH ChestX-ray14 dataset (Wang et al. 2017)," "Montgomery and Shenzen datasets (Jaeger et al. 2014)," "Covid-19 dataset (Cohen et al. 2020)," and "CheXPert dataset (Irvin et al. 2019)," we describe the open and anonymized databases that we utilized as data sources to create the datasets employed in this study, which we explain in "The segmentation dataset," "Classification training dataset," and "External classification dataset" sections.

#### NIH ChestX-ray14 dataset (Wang et al. 2017)

ChestX-ray14 is an exceptionally large dataset of frontal chest X-rays, containing 112,120 images, from 30,805 patients, showing 14 different lung diseases, as well as healthy individuals. The dataset was originally created by the US National Institutes of Health and the authors automatically labeled it with Natural Language Processing, using radiological reports. The labels have an estimated accuracy higher than 90% (Wang et al. 2017). The open database is available at the following link: https://nihcc.app.box.com/v/ChestXray-NIHCC.

It is an unbalanced dataset and a single patient can have more than one disease, therefore, classifying the database is a multi-label classification problem. The dense neural network CheXNet (Rajpurkar et al. 2017) was trained on this dataset.

Nine hundred twenty-five images showing healthy patients were extracted from the database and used in our classification training dataset. Those images correspond to 925 different patients, with mean age of 46.8 years (with 15.6 years of standard deviation) and who are 54.3% male. Additionally, 1295 ChestX-ray14 images, showing patients with pneumonia, were also included in our classification training dataset. They correspond to 941 patients, with a mean age of 48 years (standard deviation of 15.5 years), and who are 58.7% male.

#### Montgomery and Shenzen datasets (Jaeger et al. 2014)

This database was created by the National Library of Medicine, National Institutes of Health, Bethesda, Maryland, USA, in collaboration with the Department of Health and Human Services, Montgomery County, Maryland, USA and with Shenzhen No. 3 People's Hospital, Guangdong Medical College, Shenzhen, China (Jaeger et al. 2014). The X-rays taken in Shenzen show 336 normal cases and 326 tuberculosis cases. In the Montgomery images, there are 80 normal cases and 58 tuberculosis cases. Please refer to the following website to request access to the open database: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4256233.

The Montgomery images came with segmentation masks, created under the supervision of a radiologist (Candemir et al. 2014; Jaeger et al. 2014). The dataset authors segmented the images excluding the lung part behind the heart, and following some anatomical landmarks, such as the ribs, the heart boundary, aortic arc, pericardium line, and diaphragm (Candemir et al. 2014; Jaeger et al. 2014).

The authors in Stirenko et al. (2018) created segmentation masks for most of the Shenzen database; and they are similar to the Montgomery's (e.g., they also exclude the lung part behind the heart).

The healthy patients in the Montgomery and Shenzen database have a mean age of 36.1 years (with standard deviation of 12.3 years) and are 61.9% male. Their X-rays were used in our classification training dataset.

### Covid-19 dataset (Cohen et al. 2020)

Covid-19 image data collection (Cohen et al. 2020) is one of the most utilized Covid-19 X-ray databases, making it an interesting candidate for this study analysis. The dataset also contains other kinds of pneumonia, including viral variants (such as Middle East respiratory syndrome/MERS, and severe acute respiratory syndrome/SARS) and bacterial pneumonia, but we did not employ them in this work. From the dataset, we obtained 475 Covid-19 X-rays, representing all the frontal Covid-19 X-rays. Please find the open dataset on https://github.com/ieee8023/covid-chestxray-dataset.

It is a public open dataset, whose images were collected from public sources or indirectly from hospitals and clinicians (Cohen et al. 2020). It is also one of the largest public collections of Covid-19 chest X-rays we could find by the date that we started training the DNNs. Furthermore, it is also well documented, e.g., it contains information about patient age, gender, and the image source.

The images we utilized correspond to 295 Covid-19 patients, with a mean age of 42.5 years (with standard deviation of 16.5 years) and who are 64.5% male. Information about disease severity is available for some of them: from 87 patients, 79.3% survived; from 118 patients, 61.9% needed ICU; from 77 patients, 61% were intubated; from 107 patients, 62.6% needed supplemental oxygen.

### CheXPert dataset (Irvin et al. 2019)

The CheXPert database contains images collected from the Stanford University Hospital. It has 224,313 chest X-rays, from 65,240 patients, showing 13 lung diseases or no findings (Irvin et al. 2019). As in the NIH ChestX-ray14 dataset, the database authors automatically labeled the images, employing Natural Language Processing to analyze radiological reports. The labels' estimated accuracy is also above 90%. As exceptions, those of the original CheXPert test dataset were manually labeled by three board-certified radiologists. To request access to the open database, please utilize the following website: https://stanfordmlgroup.github.io/competitions/chexpert/.

We used part of the CheXPert database in our classification dataset, as part of the external validation. Seventy-nine pneumonia and 79 healthy images were used, including the ones manually labeled by three radiologists (8 of the pneumonia X-rays and 26 normal X-rays). The normal images correspond to 73 patients, with a mean age of 49.5 years (with standard deviation of 18.5 years), who are 56.2% male. The pneumonia images correspond to 61 patients, with mean age of 61.9 years (standard deviation of 18.1 years), and who are 60.7% male.

## The segmentation dataset

This dataset was used to train a U-Net to segment the lungs in frontal chest X-ray images. It contains images of Covid-19 (327), pneumonia (327), normal lungs (327) and tuberculosis (282). Pediatric patients were excluded from this study, because the Covid-19 database youngest patient is 20 years old. Thus, we hypothesized that allowing the presence of children in other classes could create bias during classification (training the DNN not to associate children with Covid-19).

The normal and tuberculosis images were all the X-rays in Montgomery and Shenzen datasets that had corresponding segmentation masks; therefore, in the segmentation dataset, we excluded the normal and tuberculosis X-rays without segmentation targets. The pneumonia X-rays were randomly selected from the NIH ChestX-ray14 images, and the Covid-19 images were randomly taken from the Covid-19 database (Cohen et al. 2020).

As targets, this dataset contains a segmentation mask for each X-ray. For the healthy and tuberculosis images, the masks were already provided in the Montgomery database and in Stirenko et al. (2018), for the Shenzen dataset. We created the other segmentation masks (for pneumonia and Covid-19). The mask creation process will be described with more details in "Creating the masks for the segmentation dataset" section.

### Segmentation dataset subdivisions

We separated the segmentation dataset in 3: training, validation, and testing. We employed them to train the U-Net with hold-out validation. The dataset subdivisions were random, but we performed a patient split: if we had more than one image from the same patient, all of them were used in a single subdivision. For testing, we selected 150 images, 50 from each class (pneumonia, Covid-19, and normal, with 10 from Montgomery and 40 from Shenzen). We did not include tuberculosis images in the testing dataset because this class is not present in our classification database, thus the U-Net performance on it was not as relevant. But they were included in training and validation because we thought that more images would improve the network's segmentation performance.

To create the training and validation datasets, we removed the test images, then randomly selected 80% of the remaining X-rays as training and 20% as validation, while keeping both datasets balanced.

## Classification training dataset

We used this dataset to classify chest X-ray images in one of three classes: healthy, pneumonia, or Covid-19. It consists of frontal X-rays and has 1295 images of healthy subjects, 1295 of pneumonia patients and 396 of Covid-19 patients. Unlike the segmentation dataset, which had masks, this dataset has simple classification labels: Covid-19, normal, or pneumonia.

The coronavirus images were all Covid-19 frontal X-rays in Cohen et al. (2020), except for the ones from Hannover Medical School, Hannover, Germany (they will be used in the external testing and validation datasets). The pneumonia X-rays were NIH ChestX-ray14 images labeled as pneumonia and with adult patients. Finally, the healthy images were all normal images from the Montgomery and Shenzen databases (with adult patients), along with 925 normal images from ChestX-ray14 (randomly selected, among adults).

## External classification dataset

We used the external classification dataset for validation (hold-out) and testing when training for Covid-19 detection.

We did not get the external Covid-19 images from another coronavirus database, because, as the current availability of Covid-19 X-rays is still limited, different datasets can share the same images. Instead, we separated the Covid-19 image data collection (Cohen et al. 2020) in two, according to geographical location. We chose all the images from Hannover Medical School (Hannover, Germany) for the external dataset because there are 79 images from this locality, a reasonable amount to create a validation and a test dataset (considering the small number of Covid-19 images), and because there are only 3 other images from Germany in the entire dataset (from Essen and Berlin). Therefore, the chance of a patient from Hannover having X-rays in another hospital from our database is exceedingly small.

The images for the normal and pneumonia classes were extracted from the CheXPert database. Seventy-nine images from each class were randomly selected, among the adult patients. We included, in the external dataset, all the normal and pneumonia images labeled by the three radiologists.

We divided the external dataset in two, for test and validation. The test dataset included 50 images from each class, and the validation dataset, 29. The division was random, but we did not allow X-rays from a single patient to be in more than one dataset.

## Image preprocessing

Original image sizes varied between datasets or sometimes even within the same dataset, and we decided to utilize the input shape of 224 × 224, with 3 channels. This is the DenseNet original input size, and the shape that we successfully used in our previous work with Covid-19 detection in X-rays (Bassi and Attux 2021). With 3 channels, we can take better advantage of transfer learning, due to the convolutional kernel shapes; images larger than 224 × 224 would be more detailed, but they would cause the simulations to be much slower, and a large input shape with a small training dataset can make the data very sparse in the input space, aggravating the problem of overfitting (Trunk 1979). Therefore, although we think that the exploration of different input shapes is an important research topic in the context of Covid-19 detection, we used the ImageNet standard of 224 × 224, because this choice had already been successful (Bassi and Attux 2021), and because it does not detract from the main purpose of this paper, which is to analyze generalization on an external dataset and the effects of lung segmentation.

When we loaded the X-rays, we converted them to grayscale and single-channel images (using OpenCV), with pixel values ranging from 0 to 255. We did this to remove any color information from the datasets, as some images had slight color variations, which could become a source of bias. As the DenseNet original input size is 224 × 224 with 3 channels, we converted the grayscale images to RGB (replicating the single-channel pixel values into three channels). Afterwards, we applied histogram equalization and normalized the pixel values between 0 and 1. Finally, we resized the X-rays to 224 × 224.

In the external test and validation datasets, as well as the segmentation datasets, we made the images square (if they were not already) by adding black bars in their borders, before resizing. We used the black bars to avoid changing the X-rays aspect ratio. Furthermore, as we did not use the bars in the classification training dataset, the DNNs (especially the one without segmentation) could not learn to identify them.

## Training for segmentation

### The U-Net

The U-Net architecture was proposed in Ronneberger et al. (2015), as a DNN for segmentation in biomedical databases. Therefore, it was designed to perform well using a small quantity of annotated samples and a large amount of data augmentation. For example, the authors in Ronneberger et al. (2015) used the DNN to segment neuronal structures in electron microscopic stacks, winning the International Symposium on Biomedical Imaging (ISBI) cell tracking challenge in 2015. As we had a relatively small amount of lung X-rays with masks, the U-Net seemed like a good option for lung segmentation.

The architecture was already used for this purpose. In Heo et al. (2019), the authors used a U-Net to successfully segment lungs in chest X-rays, generating masks that were used to create a new dataset, with images that contained only the lungs (and black pixels outside them). Afterwards, they classified these images as tuberculosis or non-tuberculosis, utilizing convolutional neural networks (CNNs).

A U-Net is a fully convolutional DNN with two symmetric paths, a contracting path, which captures context in the image, and an expanding path, which allows precise localization. The paths are connected at multiple points by skip connections. More information can be found in Ronneberger et al. (2015).

Our U-Net implementation is the same as the original (shown in Fig. 1 of Ronneberger et al. 2015); it has 5 blocks in each path, each one with two 2D convolutions and ReLU activation.

### Training with the Montgomery and Shenzen databases

We trained a U-Net with the Shenzen and Montgomery datasets, using their manually created segmentation masks as targets. We randomly selected 70% of the images for training, 20% for validation (hold-out), and 10% for testing. We used data augmentation in the training dataset, multiplying the number of images by 8 (the original images were not removed), with random rotations (between − 40 and 40 degrees), translations (with a maximum of 28 pixels up or down and 28 left or right), and horizontal flipping (50% chance).

During every training procedure in this work, we used the validation error to estimate the DNN with the best generalization capability, and this network was then evaluated on the test dataset. Furthermore, the hold-out validation error was also used in preliminary tests to determine training parameters, such as learning rate, number of epochs, and weight decay (L2 regularization). We also note here that we conducted all training procedures and network implementations described in this paper using PyTorch, a Python library specialized in neural networks. We employed a NVidia Ray Tracing Texel eXtreme (RTX) 3080 Graphics Processing Unit (GPU), with mixed precision.

Using the segmentation masks as targets, we trained the U-Net with cross-entropy loss, stochastic gradient descent (SGD), momentum of 0.99, and mini-batches of size 8. We began by training the network for 200 epochs with a learning rate (lr) of $10^{-4}$. Afterwards, we changed the rate to $10^{-5}$ and used a reduce on plateau learning rate scheduler, reducing the lr by a factor of 10 if our validation loss did not decrease in 20 epochs. We trained in this configuration for 200 epochs more.

We used mean intersection over union (IoU) to measure the U-Net test performance. IoU is a similarity measurement between two images, to calculate it we change the DNN output mask, transforming any value below 0.5 in 0 and over or equal 0.5 in 1. We then find the intersection of this binary image and the target mask (the area where both are 1), and divide it by their union (the area where the target or the output is 1). Thus, the maximum IoU is 1, when the two images are equal. Calculating the mean IoU for all testing X-rays, we can quantitatively measure the DNN segmentation performance.

After this training process, we achieved a mean test IoU of 0.927 in the Montgomery and Shenzen datasets. We also checked the generated images to have a qualitative measure of performance, and we found the U-Net satisfactorily segmented the lungs.

### Creating the masks for the segmentation dataset

In our segmentation dataset, we only had segmentation masks for the Shenzen and Montgomery images. Thus, we still needed to create masks for the Covid-19 and pneumonia images.

We used the U-Net trained before (in the Montgomery and Shenzen datasets) to help us in this task. We began by using the DNN to generate automated masks for pneumonia and Covid-19 images. We transformed these masks in binary, changing any value over or equal to 0.5 to 1 and below 0.5 to 0.

Then, we manually edited the automated masks, removing imperfections and comparing them with the X-rays. The ones that were not good enough were deleted and manually recreated. As in the Montgomery and Shenzen masks, we excluded areas behind the heart and used anatomical landmarks (like the ribs and the diaphragm) to create our masks. In total, we created 327 masks for the pneumonia class and 327 masks for the Covid-19 class.

### Training with the segmentation dataset

With the Montgomery and Shenzen masks and the new masks for the Covid-19 and pneumonia images, we had targets for every X-ray in the segmentation dataset.

We created a new U-Net, with the same structure as the last one (Ronneberger et al. 2015), to be trained employing the segmentation dataset. For this process, we used data augmentation (online) to avoid overfitting. All images were randomly rotated (between − 40 and 40°), translated (maximum of 28 pixels up or down and 28 left or right) and horizontally flipped (with a 50% chance). This augmentation multiplied the training dataset size by 15 and we did not remove the original images.

We trained the U-Net using cross-entropy loss, stochastic gradient descent (SGD) with momentum of 0.99, and mini-batches of size 5. We used a learning rate of $10^{-4}$ and trained for 400 epochs (when the DNN was already overfitting).

The DNN ended up with 0.864 mean intersection over union in the test dataset. We analyzed the generated masks and found that they correctly indicated the lung areas. Most of the DNN mistakes were generating brighter regions in the gastric bubble area and in the lung region behind the heart. You can see examples of the generated masks, created with Covid-19 X-rays, in Fig. 1.

## Training for classification

We trained two DNNs for classification: a stacked network (which also performs segmentation) and a DenseNet201. The dense network and the stacked DNN classifier module have the same structure, a DenseNet201. For this reason and to better compare the networks, we trained them for

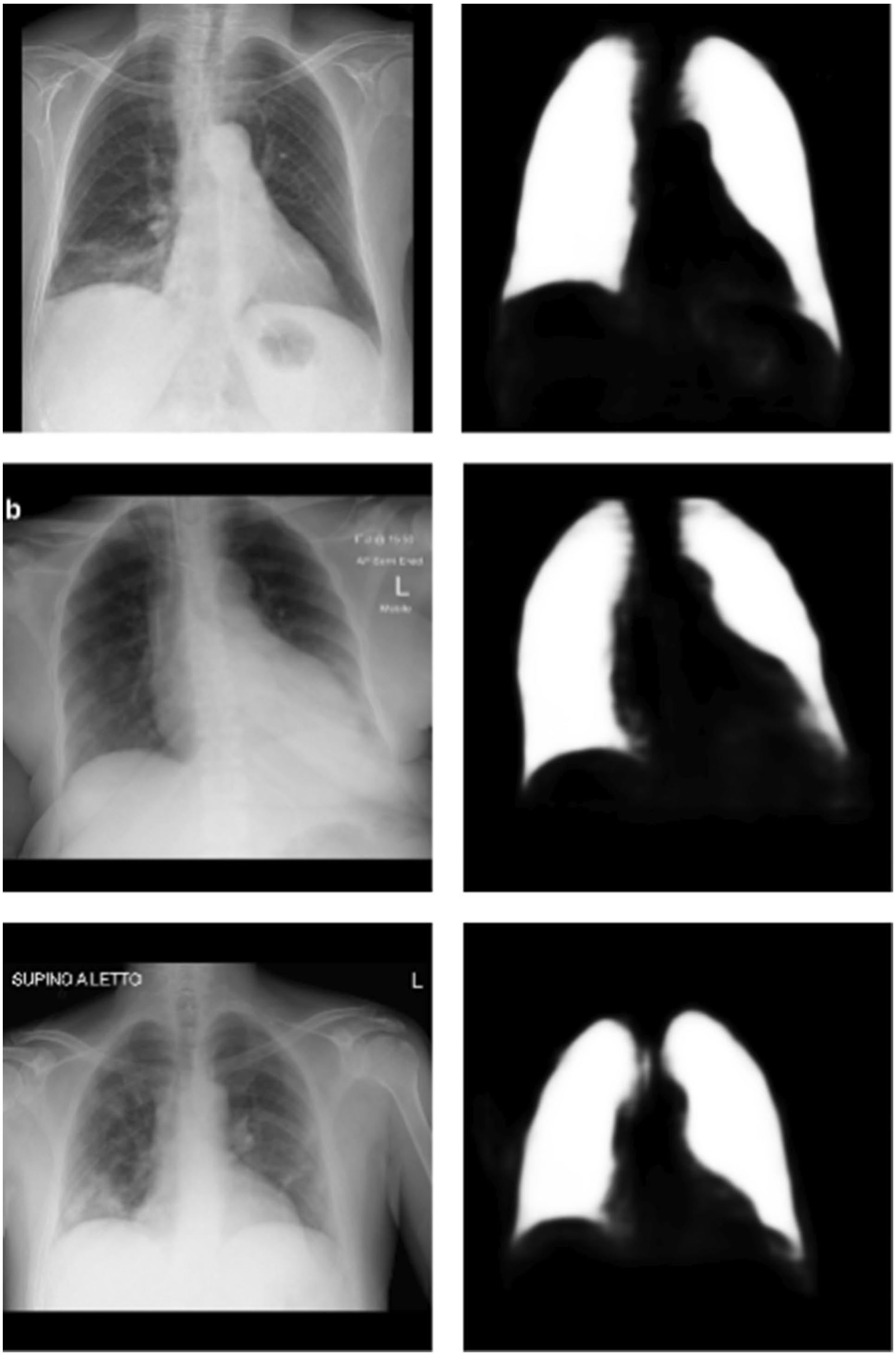

**Fig. 1** Examples of masks (created by the U-Net) and the corresponding Covid-19 X-rays. The images were gathered from the segmentation test dataset. From top to bottom, they represent a 70-year-old female on the first day of Covid-19 symptoms, a 67-year-old female on day 8 of symptoms, and a 40-year-old male on day 10

classification in the same manner, which is described in "Pretraining with the ChestX-ray14 dataset" and "Training with the classification dataset" sections.

### The stacked DNN creation

To perform lung segmentation and classification, we propose an architecture composed of stacked modules. The first one (segmentation module) is the U-Net, already trained on the segmentation dataset. This DNN receives an X-ray and outputs a segmentation mask, where high values indicate lung regions and low values refer to areas without importance. The segmentation module parameters will always be frozen when training for classification.

After the segmentation module comes the intermediate module that we designed. It applies a softmax function to the U-Net output, takes only the last dimension of the softmax result (which displays the important regions of the image with high values), and replicates it to create an image with 3 channels. Afterwards, the module performs an element-wise multiplication of this image and the input X-ray. Thus, it removes the unimportant regions from the X-ray while keeping the lungs. Lastly, the module performs batch normalization on the multiplication output; our objective with this operation is to improve the DNN generalization.

Therefore, with batch normalization, the classifier input is normalized for each training mini-batch. BatchNorm is mostly used to make training deep neural networks faster, by reducing the problem of internal covariance shift. However, it also makes the DNN output for a single example non-deterministic, creating a regularization effect, which improves generalization (Ioffe and Szegedy 2015). Its creators discovered that the technique's regularization effect can even reduce the need for other regularization methods, like dropout (Ioffe and Szegedy 2015).

The intermediate module output enters the second neural network, the classification module, a DenseNet201 (Huang et al. 2017) that predicts the chances of Covid-19, pneumonia, or normal images.

We decided to use a dense neural network as our classification module because it is a large DNN with reliable results in image classification (Huang et al. 2017) and because its architecture was highly successful in lung disease classification, obtaining F1-Scores in pneumonia detection that surpassed radiologists', in Rajpurkar et al. (2017). Note that the F1-Score is defined as the harmonic mean between precision and recall. Considering a certain class as positive and the remaining classes as negative, we can define the number of true positives (tp) as the number of positive samples correctly classified, false negatives (fn) as the number of positive samples incorrectly classified as negative, and false positives (fp) as the number of negative samples incorrectly classified as positive. With these definitions in mind, it is possible to calculate the class precision (P), recall (R), and F1-score (F1). The three equations below summarize the concepts. The DenseNet201 was downloaded already pretrained on ImageNet (Deng et al. 2009), an exceptionally large image classification dataset, with millions of samples.

$$P = \frac{tp}{tp+fp}$$

$$R = \frac{tp}{tp+fn}$$

$$F1 = \frac{2PR}{P+R}$$

Figure 2 shows our network structure and its three modules.

### Pretraining with the ChestX-ray14 dataset

We trained our DNN using a twice transfer learning approach, which is similar to the one that we used in a previous Covid-19 detection study (Bassi and Attux 2021). Another work that used twice transfer learning in a medical classification problem is Cai et al. (2018), which applied the technique for mammogram classification.

Our approach consists in a transfer learning with three steps: we download ImageNet pretrained DenseNet201s (to be used as a classification DNN or the classification module of our stacked DNN), train the networks on the large ChestX-ray14 database, and then on our classification dataset (smaller, with the Covid-19, pneumonia, and normal classes). We expect training on ChestX-ray14 to improve generalization of the DNNs, as it is a large X-ray database with a similar task to Covid-19 detection (classification of 14 lung diseases and healthy patients).

In the ChestX-ray14 dataset, the only augmentation technique that we applied was horizontal flipping (with 50% chance). Unlike the augmentation we performed in the other datasets, in this case, the new images substituted the original in the mini-batch (in the other datasets, the augmented images were added to the mini-batch along the originals).

We used the test dataset reported by the database authors as our test dataset and randomly separated the remaining images, selecting 20% for validation (hold-out). We did not allow two images from the same patient to be present in more than one dataset.

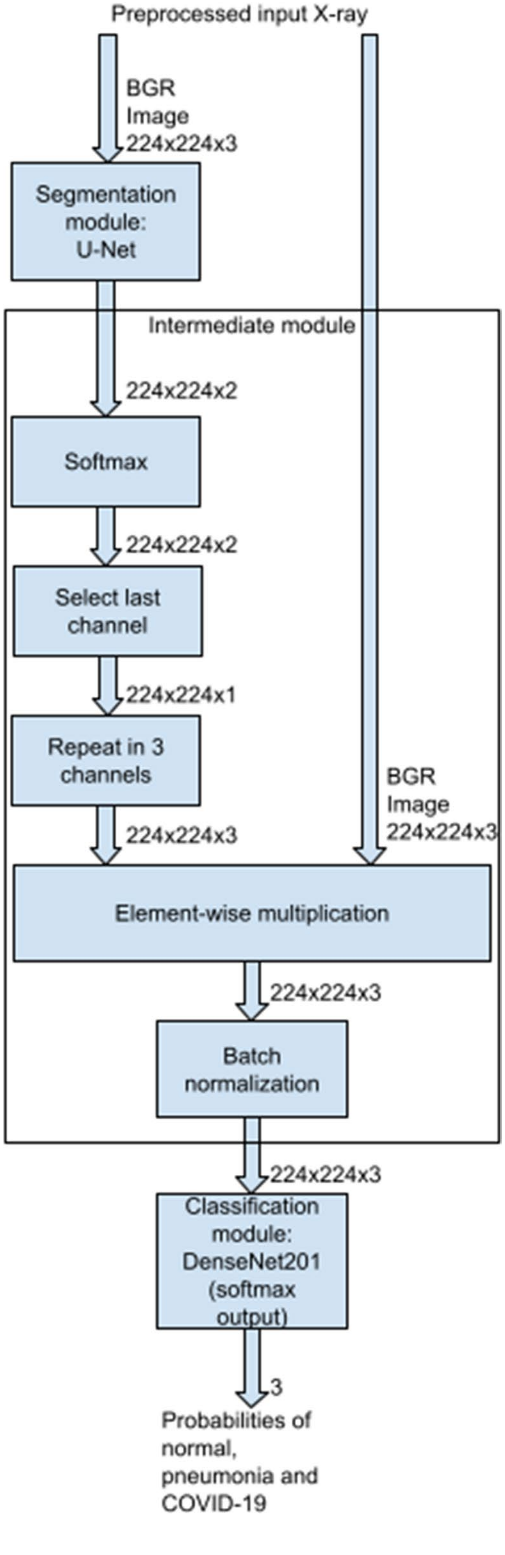

**Fig. 2** The structure of our proposed stacked DNN, for lung segmentation followed by classification

As classifying this dataset is a multi-label classification problem, we substituted the DNNs' last layer for one with 15 neurons and used PyTorch's binary cross-entropy loss with logits. We trained the networks using SGD, with momentum of 0.9 and mini-batches of size 64. We started by training only the last layer, with a learning rate of $10^{-3}$, for 20 epochs. Then, we unfroze all layers (except for the segmentation module's, when training the stacked DNN) and trained for 80 epochs, with a learning rate of $10^{-4}$. In the end of this process, both DNNs were already overfitting.

### Training with the classification dataset

In this step, we started with the DNNs (DenseNet201 and stacked DNN) that we trained in the ChestX-ray14 dataset and we performed the last stage of twice transfer learning: training on our classification dataset to classify the Covid-19, pneumonia, and normal classes. We substituted the networks' last layer by one with 3 neurons and added a dropout of 50% before it (in preliminary tests, we observed that regularization improved accuracy on the external datasets).

We also employed online data augmentation in the training dataset, to avoid overfitting and to balance the database. The augmentation process was similar to the one we used during the U-Net training (i.e., generating new images with random translation, up to 28 pixels up or down, left, or right, rotation, between – 40 and 40 degrees, horizontal flipping, with 50% chance, and not removing the original figures). To obtain almost the same number of images in the three classes, we multiplied the number of normal and pneumonia images by 3 and of Covid-19 images by 10, numbers that we decided to use after some preliminary tests. The multiplications did not produce exactly the same number of images for each class: they created 3885 normal and pneumonia training images, and 3960 Covid-19 training images. To feed the DNN balanced mini-batches, a small quantity of the Figures (90 of the 3960 Covid-19 augmented images and 15 of the 3885 pneumonia and normal images) were left out of training, but in every epoch a new selection of these images was made. Thus, every X-ray was used during the training process. At each epoch, the neural network received 11,610 training images (3870 for each class). The external validation and test datasets were not augmented.

We used cross-entropy loss, as the optimizer we chose SGD, with momentum of 0.9, and mini-batches of size 30. We trained the DNNs with hold-out validation, until overfitting was clearly observable. We started by training only the networks' last layer, for 20 epochs, with learning rate of $10^{-5}$ and weight decay of 0.01. We then trained all layers (except for the segmentation module, when training the stacked DNN), for 240 epochs, with weight decay of 0.05 and differential learning rates (the learning rate started at

$10^{-5}$ in the last layer was divided by 10 for each dense block before it, achieving the smallest value in the DenseNet first layer) (Howard and Ruder 2018). Each epoch in this stage took about 200 s in our NVidia RTX 3080 GPU.

## Layer-wise relevance propagation

DNNs are large and complex structures and it can be hard to interpret why they make decisions and classifications. Although they have a high capacity to classify images (Huang et al. 2017), in medical applications we want to have a better understanding of how it is making its choices.

Layer-wise relevance propagation is a technique that makes DNNs more explainable and understandable by humans. It propagates a value called relevance from the network output layer until its first layer, creating a heatmap, with the same format as the DNN input shape. This map associates a relevance value to each input feature (like a pixel in an image), showing how it affects the DNN output (Bach et al. 2015). The relevance propagation is approximately conservative, a neuron receives a certain amount of relevance from its posterior layer and propagates almost the same quantity to the layer below it (Montavon et al. 2019). For example, if a neuron receives 10 relevance and there are three neurons in the previous layer, it can propagate relevance values of 5, 2, and 3, but not 5, 2, and 4 (as it does not sum 10). Therefore, the amount of explanation in the heatmap is directly related to what can be explained by the DNN output. We cite two uses of LRP in medical contexts: in neuroimaging (Thomas et al. 2019) and explaining therapy predictions (Yang et al. 2018). LRP has more than one rule that can be utilized to propagate relevance, and we can apply different rules in different DNN layers to produce better heatmaps.

We used LRP to investigate if the DNNs were correctly interpreting symptoms of the diseases and to check if areas outside of the lungs were properly being ignored. We also think that giving these maps to clinicians along the DNN predictions may help them to evaluate the DNN classification and provide insights about the X-rays, improving their own analysis.

We can start the relevance propagation by any output neuron and the meaning of the colors in a heatmap depends on which neuron we choose (Montavon et al. 2019). In this study, we have output neurons with indexes 0, 1, and 2, predicting the classes normal, pneumonia, and Covid-19, respectively. When we start the relevance propagation by an output neuron that predicts a certain class, red areas (i.e., positive relevance) in the heatmap will show regions that the DNN associated with that class, and blue areas (i.e., negative relevance) will have been associated with the other classes. As an example, if we start LRP by the neuron that classifies the Covid-19 class (index 2), red areas in the heatmap will indicate regions associated with Covid-19, and blue areas will show regions associated with the other classes (normal and pneumonia). Normally, we start propagation by the neuron with the highest output, i.e., the predicted class.

When analyzing the stacked DNN, we only applied LRP to the classification module, because we only wanted to know which X-ray features were important to classify the image, not to create the segmentation mask.

To implement LRP, we used the Python library iNNvestigate (Alber et al. 2018), which already works with the DenseNet201 that we used as our classification module and as the DNN without segmentation. We chose the preset A-flat (a selection of propagation rules for the network layers) because it generated more interpretable results. To apply LRP to the classification module, we first needed to unstack our DNN. Furthermore, iNNvestigate is a library created to work with Keras and we created our DNNs using PyTorch. Thus, we used another library, called py2keras Malivenko (2018) to convert our classification module to Keras, before applying LRP. Accuracy was checked after conversion to make sure it was successful.

## The Brixia score

To compare our stacked DNN analysis with radiologists,' we will use the Brixia score. This scoring system, presented in Borghesi and Maroldi (2020), was created to grade the severity of Covid-19 cases. To score a chest X-ray, the radiologist divides the lungs into 6 parts, using two horizontal lines. The upper line is drawn at the inferior wall of the aortic line, and the other line at the level of the right pulmonary vein. If it is difficult to identify the anatomical landmarks, the authors suggest dividing the lungs into three equal zones. For each of the 6 zones, the radiologist attributes a partial score, from 0 to 3, with higher values indicating higher severity. 0 means no abnormalities in the zone, 1 means interstitial infiltrates, 2 interstitial and alveolar infiltrates, with interstitial predominance, and 3 interstitial and alveolar infiltrates, with alveolar predominance. The lines and the 6 regions are illustrated in Fig. 3. Note that a single letter in red (A, B, C, D, E, or F) represents a partial score for one of the 6 regions. The overall Brixia score (from 0 to 18) is the sum of all partial scores (A + B + C + D + E + F). The 6 partial scores are presented between square brackets, from A to F ([ABCDEF]), after the overall Brixia score. At the bottom of Fig. 3, we show in red how the score is presented. Below it, in white, there is the actual Brixia score for the example X-ray, which presents a 72-year-old male diagnosed with Covid-19, 4 days after hospitalization. The X-ray was scored by radiologists and is presented in Borghesi and Maroldi (2020). The

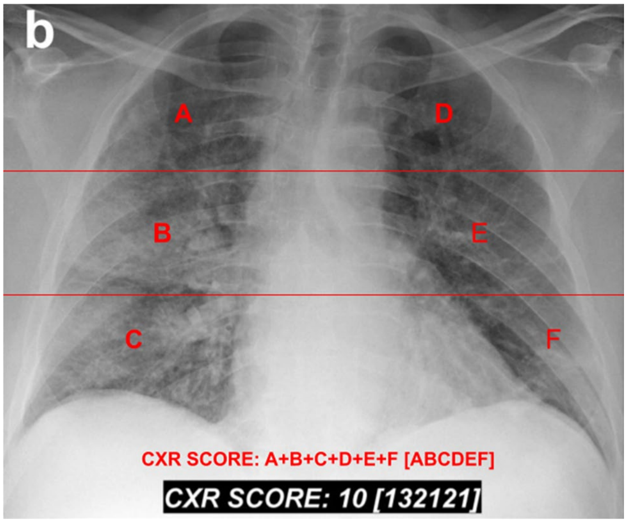

**Fig. 3** Illustration of the lung zones for the Brixia score, and the score presentation (bottom, in red), based in Borghesi and Maroldi (2020). The actual Brixia score for this X-ray, attributed by radiologists, is shown in white (bottom). The X-ray presents a 72-year-old man with Covid-19 in the fourth day of hospitalization

scoring system authors discovered that the score of later deceased patients was significantly higher than from discharged patients (Borghesi and Maroldi 2020).

## The Bayesian performance evaluation

The study in Zhang et al. (2015) proposed a Bayesian model to estimate the probability distribution of F1-Scores in the context of multi-class classification problems (when we have more than two classes and any sample can only be assigned to a single class).

The model can be summarized as (Zhang et al. 2015):

$$\begin{aligned}
&\boldsymbol{\mu} \sim Dir(\boldsymbol{\beta}) \\
&\boldsymbol{n} \sim Mult(N, \boldsymbol{\mu}) \\
&\boldsymbol{\theta}_j \sim Dir(\boldsymbol{\alpha}_j) \qquad \mathrm{for} j = 1, \ldots, M \\
&\boldsymbol{c}_j \sim Mult\left(n_j, \boldsymbol{\theta}_j\right) \qquad \mathrm{for} j = 1, \ldots, M \\
&\psi = f(\boldsymbol{\mu}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M)
\end{aligned}$$

where $N$ is the test dataset size (150 in this study), $M$ the number of classes (3), Dir() represents the Dirichlet distribution, and Mult() the multinomial.

$\boldsymbol{n}$ is a random vector, with size M, $n_j$ estimates the number of samples in class j, if we collected a new test dataset (of size N). $\boldsymbol{\mu}$ is also a random vector with size M and $\mu_j$ indicates the probability of a new sample belonging to class j. $\boldsymbol{\beta}$ indicates the hyper-parameters of the $\boldsymbol{\mu}$ prior distribution. Choosing $\boldsymbol{\beta}$ as [1,1,1] defines a uniform prior, as we and the authors of Zhang et al. (2015) did.

$\boldsymbol{c}_j$ is a random vector of size M and $c_{j,k}$ estimates the number of class j samples classified as class k. Thus, the $c_{j,k}$ elements provide an expected confusion matrix, for a new test dataset. $\boldsymbol{\theta}_j$ is a random vector of size M, $\theta_{j,k}$ estimates the probability of classifying a sample from class j as class k. $\boldsymbol{\alpha}_j$ is a vector with M hyper-parameters, defining the $\boldsymbol{\theta}_j$ prior distribution. As in Zhang et al. (2015), we chose all elements in these vectors as 1, creating a uniform prior.

ψ represents a function, calculated (deterministically) using the posterior probability distributions of $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$. Zhang et al. (2015) provides functions to estimate many performance measurements: class precision ($P_j$), class recall ($R_j$), macro-averaged F1-Score (maF1), and micro-averaged F1-Score (miF1). In a multi-class single-label classification problem, miF1 is identical to the overall accuracy (Sakai 2006). With a balanced test dataset, like our test database, it is also identical to the average accuracy. Therefore, we used the miF1 posterior probability distribution to estimate our accuracy reliability.

We expanded the Bayesian model to also estimate the specificity for each class and their arithmetic mean. Therefore, we expressed the metrics as functions of $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ and calculated them using these parameters posterior distributions. Zhang et al. (2015) defines functions for $tn_j$ and $fp_j$ (true negatives and false positives in the class j contingency table):

$$\begin{aligned}
tn_j &= \sum\nolimits_{u \neq j} \sum\nolimits_{v \neq j} N \mu_u \theta_{u,v} \\
fp_j &= \sum\nolimits_{u \neq j} N \mu_u \theta_{u,j}
\end{aligned}$$

Therefore, using the equations above and the definition of specificity, we can deduce the equations that define the class specificity and the mean specificity (macro-averaged) as functions of $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$:

$$\begin{aligned}
Specificity_j &= \frac{tn_j}{tn_j + fp_j} = \frac{\sum_{u \neq j} \sum_{v \neq j} \mu_u \theta_{u,v}}{\sum_{u \neq j} \sum_{v=1}^{M} \mu_u \theta_{u,v}} \\
Mean\ Specificity &= \frac{1}{M} \sum\nolimits_{j=1}^{M} Specificity_j
\end{aligned}$$

The Bayesian model takes only the classifier confusion matrix as input, which it uses to create the likelihoods for $\boldsymbol{c}_j$ and $\boldsymbol{n}$.

We computed the posterior probability distributions with Markov chain Monte Carlo (MCMC), utilizing the Python library PyMC3 (Salvatier et al. 2016). We used the No-U-Turn Sampler (Homan and Gelman 2014), with 4 chains, 10,000 tuning samples, and 100,000 samples after tuning.

## Results

Table 1 shows the confusion matrix for our stacked DNN, and Table 2 for the DNN without segmentation (we created both matrices using the external test database).

**Table 1** Stacked DNN confusion matrix

| | | Predicted Class | | |
|---|---|---|---|---|
| | | Normal | Pneumonia | Covid-19 |
| Real class | Normal | 38 | 7 | 5 |
| | Pneumonia | 8 | 32 | 10 |
| | Covid-19 | 2 | 0 | 48 |

**Table 2** Confusion matrix for the DNN without segmentation

| | | Predicted Class | | |
|---|---|---|---|---|
| | | Normal | Pneumonia | Covid-19 |
| Real class | Normal | 43 | 0 | 7 |
| | Pneumonia | 14 | 24 | 12 |
| | Covid-19 | 6 | 0 | 44 |

Tables 3 and 4 show performance metrics in the external test dataset, for the DNNs with and without segmentation, respectively. In the second column (score), we show performance scores, calculated in the traditional and deterministic manner, using the confusion matrix. The other columns refer to statistics of the metrics' posterior distributions, estimated using Bayesian inference. They are mean, standard deviation (std), Monte Carlo error, and 95% high-density interval (HDI). The HDI is defined as an interval with 95% of the distribution probability mass and any point in this interval has a probability that is higher than any point outside the HDI.

We calculated, with the test dataset, the multi-class area under the ROC curve (AUC) using macro averaging and the pairwise comparisons approach from Hand and Till (2001). The stacked DNN achieved 0.917 AUC and the DenseNet201, 0.906. We do not present interval estimations for multi-class AUC because defining its confidence interval is not a simple task, and bootstrapping is the suggested method for it (Hand and Till 2001). We cannot use bootstrapping in this study, as we are using an external test dataset and we have a small number of Covid-19 X-rays.

In Fig. 4, we show the Bayesian estimations of mean accuracy (equal to miF1) and macro-averaged F1-Score. In Fig. 5, we display the corresponding trace plots (for only one MCMC chain). These plots exclude the tuning samples.

Our trained DNNs are available for download at https://github.com/PedroRASB/Covid-19-detection-with-lung-segmentation.

## Discussion

In a previous study, we utilized a dataset that was similar to our classification training database. We also trained dense neural networks (without segmentation), but we did not perform validation and testing on an external database (Bassi and Attux 2021). There, we could achieve accuracies above 90%, as is common in many Covid-19 detection studies, which also use internal validation, i.e., they randomly divide a single dataset in testing, validation, and training (Shoeibi et al. 2020). Furthermore, in preliminary tests using the stacked DNN that we proposed here, but without external validation, we could also achieve accuracies above 90%. We note that, in our previous study and in the preliminary tests, our classification training database was divided in three datasets (for training,

**Table 3** Performance metrics for the DNN with segmentation. The score values are traditional point estimates. The other values were obtained with Bayesian inference

| Metric | Score | Mean | std | MC error | 95% HDI |
|---|---|---|---|---|---|
| Mean accuracy or miF1 | 0.787 | 0.761 | 0.034 | 0.0 | [0.695, 0.826] |
| Macro-averaged F1-score | 0.781 | 0.754 | 0.034 | 0.0 | [0.687, 0.82] |
| Macro-averaged precision | 0.791 | 0.764 | 0.034 | 0.0 | [0.697, 0.829] |
| Macro-averaged recall | 0.787 | 0.761 | 0.032 | 0.0 | [0.698, 0.823] |
| Macro-averaged specificity | 0.893 | 0.88 | 0.017 | 0.0 | [0.848, 0.912] |
| Normal precision | 0.792 | 0.765 | 0.059 | 0.0 | [0.648, 0.877] |
| Normal recall | 0.76 | 0.736 | 0.06 | 0.0 | [0.617, 0.85] |
| Normal F1-score | 0.776 | 0.748 | 0.048 | 0.0 | [0.654, 0.839] |
| Normal specificity | 0.9 | 0.887 | 0.031 | 0.0 | [0.825, 0.943] |
| Pneumonia precision | 0.821 | 0.786 | 0.063 | 0.0 | [0.66, 0.902] |
| Pneumonia recall | 0.64 | 0.623 | 0.066 | 0.0 | [0.493, 0.75] |
| Pneumonia F1-score | 0.719 | 0.692 | 0.054 | 0.0 | [0.586, 0.795] |
| Pneumonia specificity | 0.93 | 0.915 | 0.027 | 0.0 | [0.861, 0.964] |
| Covid-19 precision | 0.762 | 0.742 | 0.054 | 0.0 | [0.636, 0.844] |
| Covid-19 recall | 0.96 | 0.925 | 0.036 | 0.0 | [0.854, 0.985] |
| Covid-19 F1-score | 0.85 | 0.822 | 0.038 | 0.0 | [0.746, 0.894] |
| Covid-19 specificity | 0.85 | 0.84 | 0.035 | 0.0 | [0.769, 0.906] |

**Table 4** Performance metrics for the DNN without segmentation. The score values are traditional point estimates. The other values were obtained with Bayesian inference

| Metric | Score | Mean | std | MC error | 95% HDI |
|---|---|---|---|---|---|
| Mean accuracy or miF1 | 0.74 | 0.717 | 0.036 | 0.0 | [0.646, 0.786] |
| Macro-averaged F1-score | 0.729 | 0.705 | 0.037 | 0.0 | [0.632, 0.776] |
| Macro-averaged precision | 0.794 | 0.758 | 0.032 | 0.0 | [0.696, 0.82] |
| Macro-averaged recall | 0.74 | 0.717 | 0.033 | 0.0 | [0.653, 0.781] |
| Macro-averaged Specificity | 0.87 | 0.858 | 0.017 | 0.0 | [0.825, 0.891] |
| Normal precision | 0.683 | 0.667 | 0.058 | 0.0 | [0.553, 0.779] |
| Normal recall | 0.86 | 0.83 | 0.051 | 0.0 | [0.728, 0.924] |
| Normal F1-score | 0.761 | 0.738 | 0.045 | 0.0 | [0.647, 0.824] |
| Normal specificity | 0.8 | 0.792 | 0.039 | 0.0 | [0.714, 0.867] |
| Pneumonia precision | 1.0 | 0.926 | 0.05 | 0.0 | [0.829, 0.998] |
| Pneumonia recall | 0.48 | 0.472 | 0.068 | 0.0 | [0.34, 0.605] |
| Pneumonia F1-score | 0.649 | 0.622 | 0.063 | 0.0 | [0.497, 0.743] |
| Pneumonia specificity | 1.0 | 0.981 | 0.013 | 0.0 | [0.955, 1.0] |
| Covid-19 precision | 0.698 | 0.682 | 0.057 | 0.0 | [0.569, 0.792] |
| Covid-19 recall | 0.88 | 0.849 | 0.049 | 0.0 | [0.752, 0.938] |
| Covid-19 F1-score | 0.779 | 0.755 | 0.044 | 0.0 | [0.667, 0.839] |
| Covid-19 specificity | 0.81 | 0.802 | 0.039 | 0.0 | [0.726, 0.876] |

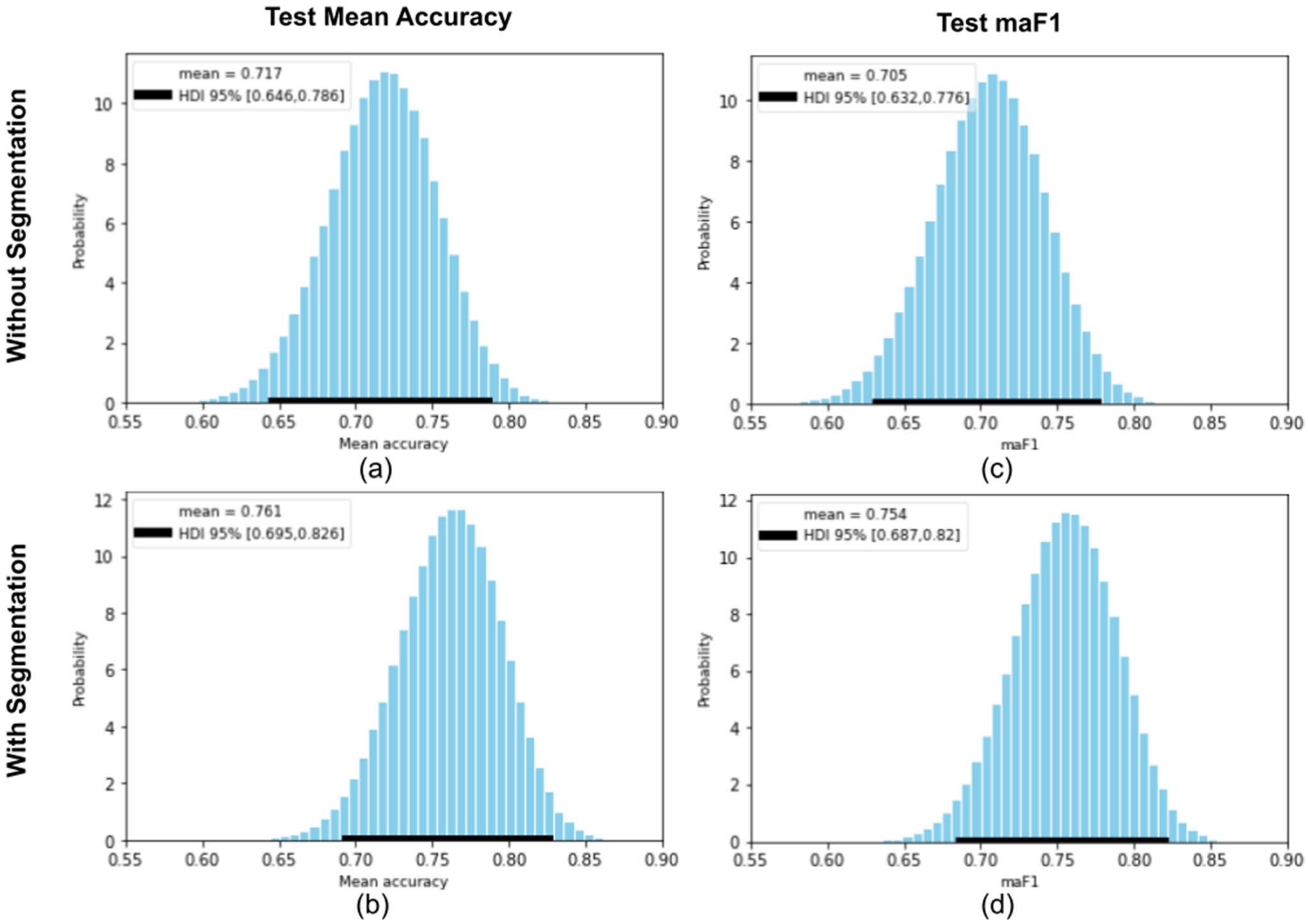

**Fig. 4** Posterior probability density estimations for test mean accuracy (subfigures **a** and **b**) and macro-averaged F1-Score (**c** and **d**), considering the DNNs with (**b** and **d**) and without segmentation (**a** and **c**)

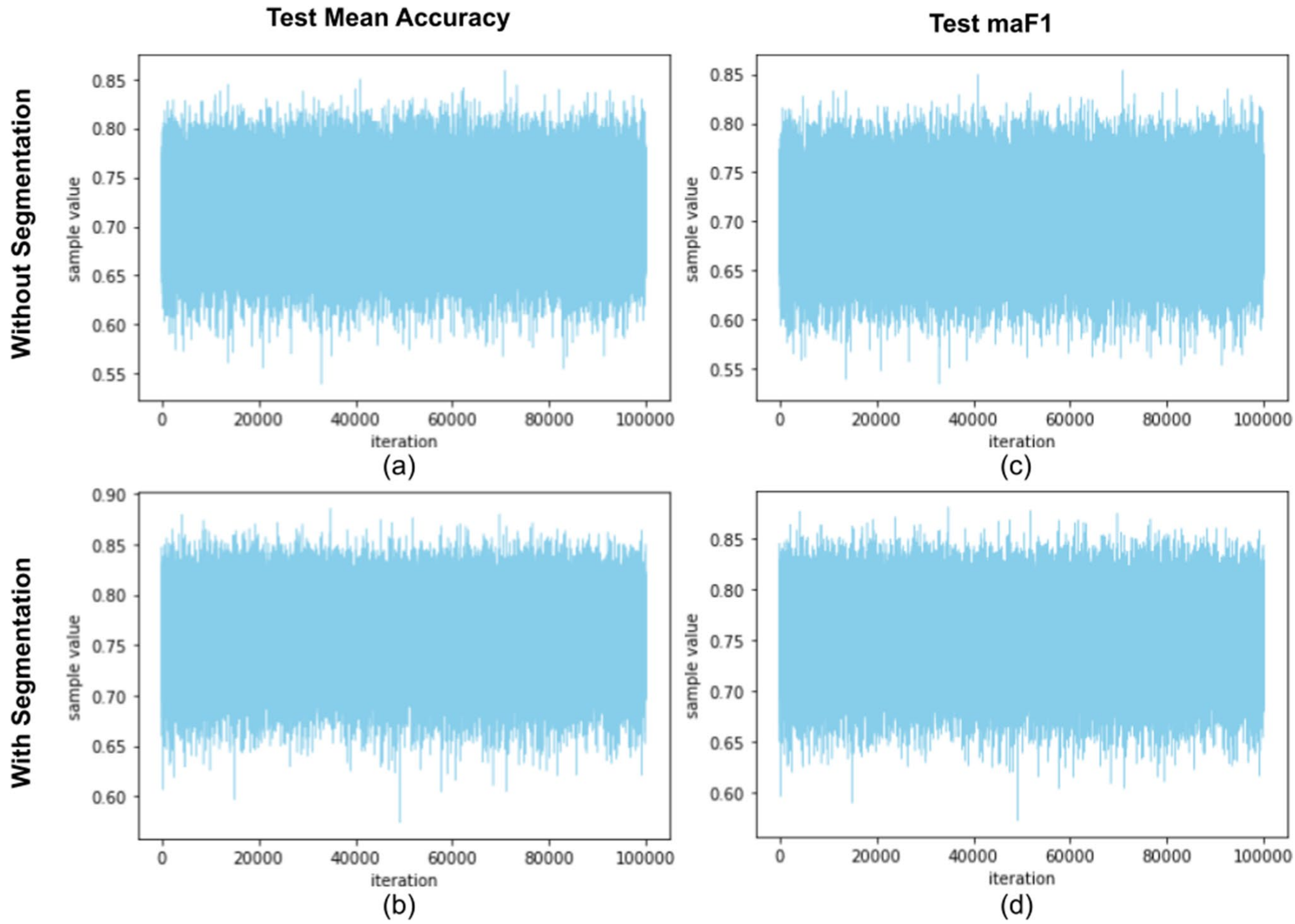

**Fig. 5** Trace plots for test mean accuracy (subfigures **a** and **b**) and macro-averaged F1-Score (**c** and **d**), considering the DNNs with (**b** and **d**) and without segmentation (**a** and **c**)

validation and test) and two images from the same patient were not allowed to be present in two different datasets. We conclude that evaluating DNNs in an external dataset can show significantly smaller accuracies, indicating that bias can hinder generalization when working with mixed datasets, and that internal validation results may not reflect performance when analyzing data from other hospitals and locations.

Furthermore, when we compare the results of our stacked DNN and the DenseNet201, we observe that segmentation influences the model generalization capability, increasing mean accuracy score on the external test dataset by 4.7%, and the Bayesian estimation mean by 4.4%.

Some works have also used lung segmentation for Covid-19 detection in chest X-rays. A recent study (Rahman et al. 2020) used a modified U-Net to segment the X-rays beforehand, it then applied an image enhancement technique (like histogram equalization) and classified the segmented X-ray with different DNNs. Like in this study, their work utilized a mixed database, but, unlike our work, they constructed their test dataset randomly, using five-fold cross validation. As can be seen in other works that applied internal validation (Shoeibi et al. 2020), their study obtained high accuracies, around 95%. But surprisingly, their results showed that lung segmentation reduced test accuracy and F1-scores (by about 1%). This result strongly contrasts with our findings (4.7% accuracy improvement with segmentation), and, although the utilization of our intermediate module might have positively influenced our performances with segmentation, we do not think that it is the main cause for this discrepancy. Instead, we think that the different test methodologies in the two papers caused the different results. In our study, lung segmentation reduced dataset bias, improving generalization and performances on the external test dataset. However, this reduction of dataset bias may decrease performance when it is measured with internal validation, possibly explaining why lung segmentation reduced accuracy and F1-Score in Rahman et al. (2020).

The normal class specificity shows the percentage of unhealthy patients that were not classified as healthy. The score value of 90%, in Table 3, indicates that a relatively small number of the patients with a disease were miss-classified as healthy by our model.

We note that the 95% high-density intervals (HDIs) are relatively large, e.g., for mean accuracy with the stacked DNN the interval length is 0.131. This can also be observed in Fig. 4. The strongest reason for the large intervals is the small size of the test dataset, and using more test samples would increase the performance metrics confidence.

## LRP and comparison with radiologists' analysis (using the Brixia score)

We propose comparing X-rays scored by radiologists, using the Brixia score, with heatmaps, created by LRP. The maps show how much relevance in classification each part of the X-rays has. Therefore, if we start the propagation by the neuron that classifies Covid-19, areas that have larger and darker red regions indicate where the DNN found more severe Covid-19 symptoms. Checking these areas' partial Brixia scores may indicate if the DNNs look for the same signs of Covid-19 as radiologists do. Furthermore, more severe cases of Covid-19 may show stronger symptoms and could increase the Covid-19 probability predicted by the DNN. Therefore, we may also be able to check if there is a correlation between images with high overall Brixia scores and the higher predicted probabilities.

Besides presenting the scoring system, Borghesi and Maroldi (2020) also show examples of Covid-19 X-rays, already scored by radiologists. These images are also part of our training dataset. In Fig. 6, we analyze, with our stacked DNN, three of them (the ones that had nothing written over the lungs). The X-rays displayed in this section were processed as indicated in "Image preprocessing" section. The figure presents the X-rays, the generated segmentation masks, the LRP heatmaps, the network outputs and the Brixia scores (given by radiologists), with the partial scores in brackets. We note that relevance propagation began at the neuron that classifies Covid-19; therefore, red areas indicate regions that the DNN associated with Covid-19, while blue areas were associated with the pneumonia or the normal class.

All X-rays in Fig. 6 were taken from a 72-year-old man diagnosed with Covid-19. The one in the first row is from the day of admission, one day after the onset of fever (Borghesi and Maroldi 2020). We observe that the X-ray shows little signs of Covid-19, as the Brixia score is exceptionally low, at one. This should make classification more challenging, and, indeed, our DNN could not correctly classify this image, predicting the normal class, but with only 60.3% probability. The patient had a rapid disease progression; the second and third rows show X-rays at days 4 and 5 post-hospitalization, respectively (Borghesi and Maroldi 2020). Our DNN correctly classified both X-rays, with Covid-19 probabilities of 65.9% and 73.5%.

Unlike the Brixia score, our network is not designed to analyze disease severity. But we observe that X-rays showing more severe and apparent symptoms (thus, with higher Brixia scores) also increase the DNN confidence for the Covid-19 class. In Fig. 6, we see that the higher the overall Brixia score, the higher the Covid-19 predicted probability. This indicates a similarity between the symptoms that the radiologists look for and the ones that our DNN analyses.

An analysis of the partial scores and the heatmaps of the two correctly classified X-rays in Fig. 6 also corroborates with the conclusion above. In both heatmaps, we observe more relevance in the right lung, and it also has higher partial Brixia scores. The middle heatmap shows that, in the right lung, the DNN found more Covid-19 signs in regions b and c, which also have higher partial Brixia scores; in the left lung, we see more relevance in the E region, which also contains the highest partial score. In the lower heatmap, in the right lung, there is again more relevance in regions b and c, which also present higher Brixia scores. The f region of the lower heatmap in Fig. 6 has 3 Brixia score, but is blue in our heatmap. The reason for this is that the region was mostly associated, by our DNN, with the pneumonia class (this region is very red if we start LRP by the neuron that classifies pneumonia).

LRP analysis showed that our segmentation module and intermediate module work as intended, containing almost all relevance in the lung regions (as can be seen in Figs. 6 and 7). Figure 7 also analyzes the stacked DNN. It shows a Covid-19 input X-ray, the generated mask and LRP heatmap. But, unlike in Fig. 6, this radiography is from the external test dataset. We observe that the segmentation mask is not perfect, but the areas outside the lungs are not very bright and are mostly ignored by the DNN, as the heatmap shows. Again, this X-ray was correctly classified (89.6% probability of Covid-19) and the red areas in the heatmap were associated, by the neural network, with the Covid-19 class.

We can further understand the differences between the two DNNs (with and without segmentation) when we analyze them using layer-wise relevance propagation. Therefore, we show, in Fig. 8, an LRP analysis for the same X-ray in Fig. 7, but created using the DenseNet201 (without segmentation) instead of the stacked DNN. We note that this DNN correctly classifies the image, but it assigned a much lower Covid-19 probability, of 46.2%. Red areas on the map were associated with the Covid-19 class, while blue areas were associated with the other classes.

We observe, in Fig. 8, that there is relevance outside of the lungs. Its existence may explain why the stacked DNN has better generalization (4.7% higher accuracy on the external test dataset) than the network without segmentation. The relevance outside of the critical areas might indicate dataset biases learned by the DNN. However, some Covid-19 signs, indicated in the heatmap in Fig. 7, can still be seen on Fig. 8 (mostly on the left lung).

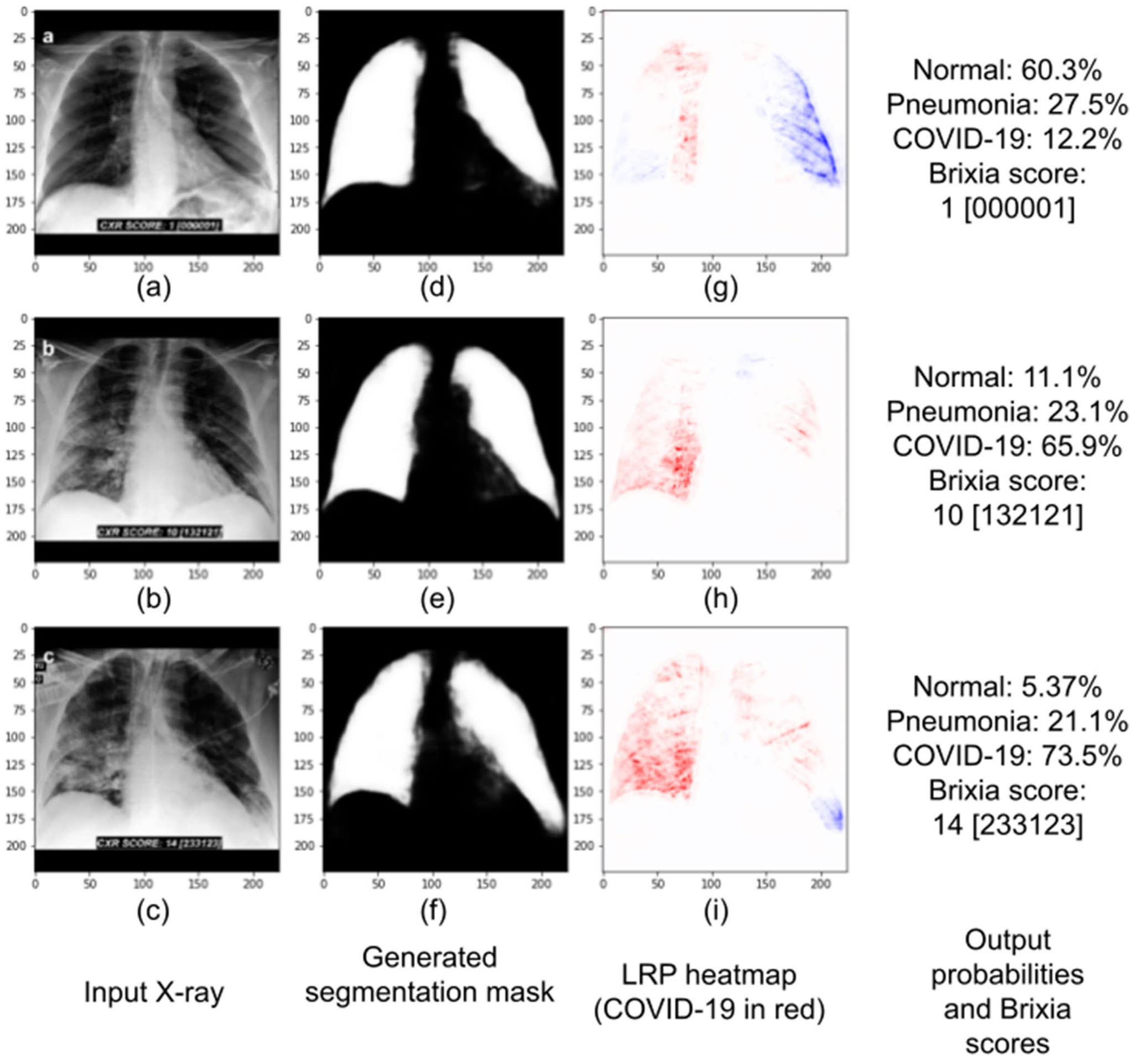

**Fig. 6** X-rays (subfigures **a**, **b**, and **c**) from a male 72-year-old Covid-19 patient. They were created in the first (**a**), fourth (**b**), and fifth (**c**) days of hospitalization. Masks (**d**, **e**, and **f**), heatmaps (**g**, **h**, and **i**), and class probabilities (right) consider the stacked DNN. In the heatmaps, red colors indicate areas that the DNN associated to Covid-19, while blue areas were associated to the pneumonia or normal classes. We observe a clear correlation between Brixia scores and red regions in the heatmaps. Moreover, as the disease progressed, Brixia scores increased, the DNN predicted higher Covid-19 probability, and heatmaps became redder

## Conclusion

First, we observe that our mean accuracy score on the external test dataset, using the stacked DNN, was 78.7% and, without segmentation, 74%. These values are significantly lower than the accuracies calculated using internal validation (i.e., randomly splitting a database in test, validation, and training datasets). Our previous study (which used a database like our current classification training dataset and employed internal validation, without lung segmentation) (Bassi and Attux 2021), and many other works that detected Covid-19 using DNNs without external evaluation (Shoeibi et al. 2020) showed accuracies above 90%. Although a loss of performance is expected when a DNN is trained on one database and tested on another, this accuracy discrepancy may indicate that utilizing mixed datasets creates bias and shortcut learning (López-Cabrera et al. 2021), which improves internal validation accuracies and

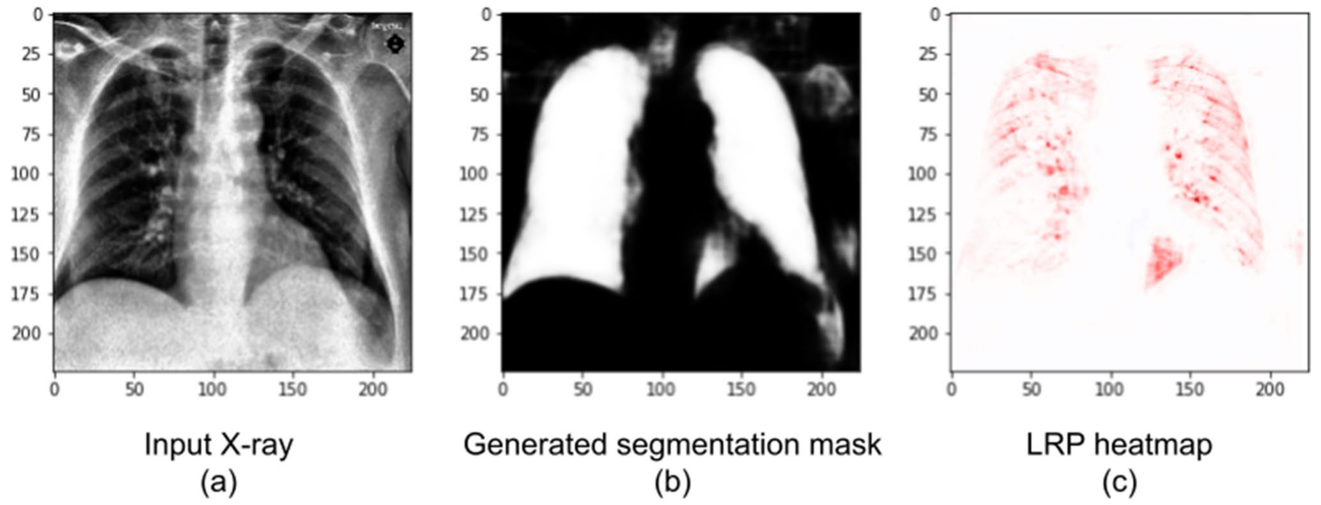

**Fig. 7** The X-ray (**a**) is an image from our external test dataset (unlike Fig. 6, which presented training X-rays), correctly classified as Covid-19. It presents a male patient in the first day of Covid-19 symptoms. The mask (**b**) and the heatmap (**c**) were created with the stacked DNN. Red colors indicate areas that the DNN associated to Covid-19, while blue areas were associated to the classes pneumonia or normal

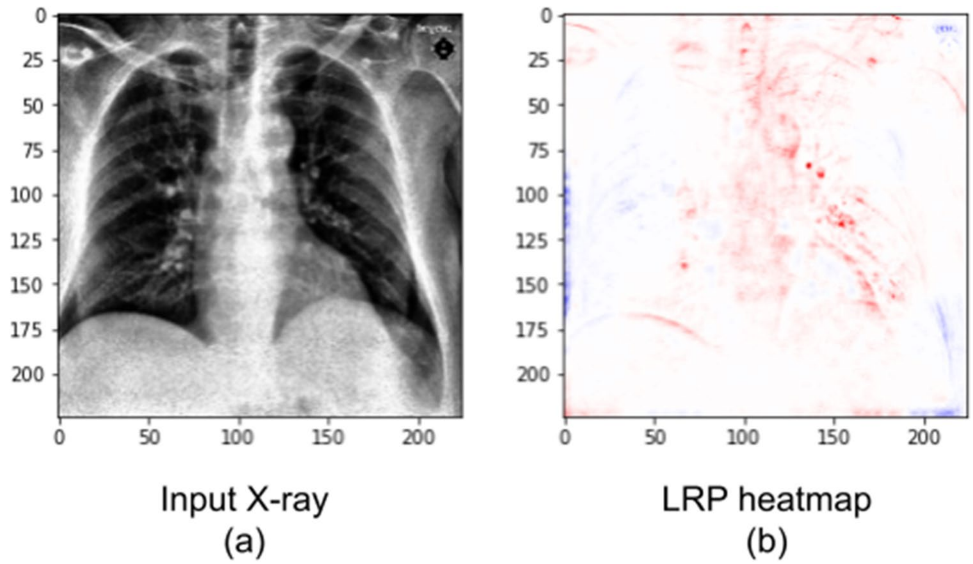

**Fig. 8** Covid-19 X-ray (**a**) and heatmap (**b**). Unlike Figs. 6 and 7, this heatmap was created with the DNN without lung segmentation. The X-ray is an image from our external test dataset, correctly classified by the network, and it is the same X-ray seen in Fig. 7. Red colors indicate areas that the DNN associated to the Covid-19 class, while blue areas were associated to pneumonia or normal

performance metrics, as the study in Maguolo and Nanni (2020) suggests. These extremely high accuracies may not hold up when images from other hospitals, locations, and datasets are analyzed, as we have seen in this work.

The utilization of lung segmentation, performed by our stacked DNN architecture, improved generalization, increasing mean accuracy score on the external test dataset by 4.7% (or 4.4%, when considering the Bayesian estimations means). Other techniques that may have helped mitigating mixed dataset bias in this study were histogram equalization (in the input X-rays), batch normalization (in our intermediate module), removing pediatric patients from the datasets (because the youngest patient in the Covid-19 class is 20 years old), utilizing an external validation dataset, regularization (dropout and weight decay), twice transfer learning, and data augmentation.

Bayesian estimation of the DNNs' performance metrics allowed us to quantify the reliability of the reported metrics. We observed relatively large 95% high-density intervals, caused by the small size of the test dataset (150 images). This emphasizes both the importance of making interval estimations in the context of Covid-19 detection, and how beneficial larger Covid-19 X-ray databases would be.

Layer-wise relevance propagation allowed us to generate heatmaps and analyze how our DNNs performed their classification. The stacked DNN heatmaps indicated that the networks successfully ignored areas outside the lungs, as these regions' relevance was exceedingly small, showing almost no color in the maps. Comparing X-rays scored by radiologists using the Brixia score with our stacked DNN outputs and heatmaps demonstrated that, normally, regions with high partial Brixia scores also had high Covid-19 LRP relevance. Furthermore, X-rays with higher overall scores were associated with higher Covid-19 predicted probabilities. These observations point out that radiologists and our stacked DNN look for the same signs of Covid-19 in a radiography.

Performing LRP in a DenseNet201 without segmentation indicated that, although lung areas were relevant and considered, the DNN also paid attention to regions outside of the lungs. This again suggests that segmentation can reduce dataset bias and improve generalization.

Although we conclude that mixed dataset bias is significant, our DNNs' performance on an external dataset and LRP analysis indicate that it can be partially avoided. On the external test dataset our stacked network had 0.916 AUC and, using the Bayesian model, we estimated a macro-averaged F1-Score with mean of 0.754 and 95% high density interval of [0.687, 0.82].

This study shows the need for large, open, and high-quality Covid-19 X-ray databases, with all classes collected from the same sources, to better avoid dataset bias, improve generalization, and increase performance metrics reliability. Our DNNs' performance on the external dataset suggests that, even with small and mixed datasets, DNNs can be successfully trained to detect Covid-19, if appropriate measures to avoid bias are taken. However, we must note that even though we utilized an external test dataset, clinical tests are needed to further ensure that the performances we observed in this study are replicable in a real-world scenario.

This work employed a non-standard testing strategy, evaluating the DNNs on an external, out-of-distribution dataset. Therefore, we mitigated the effect of bias in the reported results, and more realistically assessed the potential of deep learning to become an auxiliary tool to help clinicians in Covid-19 detection. Moreover, using this evaluation strategy, we demonstrated the importance of lung segmentation for DNN generalization, a capability that is paramount for the neural network applicability in a real clinical scenario. Finally, our novel analysis with the Brixia score and LRP heatmaps allowed a more profound understanding of the deep neural network decision rules, increasing its trustworthiness, a quality that is crucial for medical applications.

**Acknowledgements** This work was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (process 308811/2019-4) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

**Funding** Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement.

## Declarations

**Competing interests** The authors declare no competing interests.

## References

Alber M, Lapuschkin S, Seegerer P, Hägele M, Schütt KT, Montavon G, Samek W, Müller KR, Dähne S, Kindermans PJ. iNNvestigate neural networks! ArXiv. 2018:1808.04260. https://doi.org/10.48550/arxiv.1808.04260. Accessed Aug 2022.

Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One. 2015;10. https://doi.org/10.1371/journal.pone.0130140.

Bassi PRAS, Attux R. A deep convolutional neural network for covid-19 detection using chest x-rays. Res Biomed Eng. 2021. https://doi.org/10.1007/s42600-021-00132-9.

Borghesi A, Maroldi R. Covid-19 outbreak in Italy: experimental chest x-ray scoring system for quantifying and monitoring disease progression. Radiol Med. 2020;125. https://doi.org/10.1007/s11547-020-01200-3.

Cai Q, Liu X, Guo Z. Identifying architectural distortion in mammogram images via a se-densenet model and twice transfer learning. In: 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI); 2018. p. 1–6. https://doi.org/10.1109/CISP-BMEI.2018.8633197.

Candemir S, Jaeger S, Palaniappan K, Musco J, Singh R, Xue Z, Karargyris A, Antani S, Thoma G, Mcdonald C. Lung segmentation in chest radiographs using anatomical atlases with non-rigid registration. IEEE Trans Med Imaging. 2014;33. https://doi.org/10.1109/TMI.2013.2290491.

Cohen JP, Morrison P, Dao L. Covid-19 image data collection. ArXiv. 2020;2003:11597.

Deng J, Dong W, Socher R, Li L, Kai L, Li F. Imagenet: A large-scale hierarchical image database. IEEE conference on computer vision and pattern recognition. 2009;2009:248–55. https://doi.org/10.1109/CVPR.2009.5206848.

Guan W, Ni Z, Hu Y, Liang W, Ou C, He J, Liu L, Shan H, Lei C, Hui DS, Du B, Li L, Zeng G, Yuen KY, Chen R, Tang C, Wang T, Chen P, Xiang J, et al. Clinical characteristics of coronavirus disease 2019 in China. N Engl J Med. 2020;382. https://doi.org/10.1056/NEJMoa2002032.

Hand DJ, Till RJ. A simple generalisation of the area under the roc curve for multiple class classification problems. Mach Learn. 2001;45. https://doi.org/10.1023/A:1010920819831.

Heo SJ, Kim Y, Yun S, Lim SS, Kim J, Nam CM, Park EC, Jung I, Yoon JH. Deep learning algorithms with demographic information help to detect tuberculosis in chest radiographs in annual workers' health examination data. Int J Environ Res Public Health. 2019;16. https://doi.org/10.3390/ijerph16020250.

Homan MD, Gelman A. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. J Mach Learn Res. 2014;15(1):1593–623.

Howard J, Ruder S. Universal language model fine-tuning for text classification; 2018. https://doi.org/10.18653/v1/P18-1031.

Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017;2017:2261–9. https://doi.org/10.1109/CVPR.2017.243.

Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv: 2015;1502.03167

Irvin J, Rajpurkar P, Ko M, Yu Y, Ciurea-Ilcus S, Chute C, Marklund H, Haghgoo B, Ball R, Shpanskaya K, Seekins J, Mong D, Halabi S, Sandberg J, Jones R, Larson D, Lan-glotz C, Patel B, Lungren M, Ng A. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. Proceedings of the AAAI Conference on Artificial Intelligence. 2019;33. https://doi.org/10.1609/aaai.v33i01.3301590.

Jaeger S, Karargyris A, Candemir S, Folio L, Siegelman J, Callaghan F, Xue Z, Palaniappan K, Singh RK, Antani S, Thoma G, Wang YX, Lu PX, McDonald CJ. Automatic tuberculosis screening using chest radiographs. IEEE Trans Med Imaging. 2014;33. https://doi.org/10.1109/TMI.2013.2284099.

Kim EA, Lee KS, Primack SL, Yoon HK, Byun HS, Kim TS, Suh GY, Kwon OJ, Han J. Viral pneumonias in adults: Radiologic and pathologic findings. RadioGraphics. 2002;22. https://doi.org/10.1148/radiographics.22.suppl_1.g02oc15s137.

López-Cabrera J, Portal Diaz J, Orozco R, Lovelle O, Perez-Diaz M. Current limitations to identify Covid–19 using artificial intelligence with chest x-ray imaging (part ii). the shortcut learning problem. Heal Technol. 2021;11. https://doi.org/10.1007/s12553-021-00609-8.

Maguolo G, Nanni L. A critic evaluation of methods for Covid-19 automatic detection from X-ray images. 2020. ArXiv 2004:12823.

Malivenko G. pytorch2keras; 2018. https://github.com/nerox8664/pytorch2keras. Accessed 01 Mar 2021

Mercer T, Salit M. Testing at scale during the covid-19 pandemic. Nat Rev Genet. 2021;22:1–12. https://doi.org/10.1038/s41576-021-00360-w.

Montavon G, Binder A, Lapuschkin S, Samek W, Müller KR. Layer-wise relevance propagation: an overview. In: Explainable AI: interpreting, explaining and visualizing deep learning: Springer International Publishing; 2019. p. 193–209.

Rahman T, Khandakar A, Qiblawey Y, Tahir A, Kiranyaz S, Kashem SBA, Islam MT, Maadeed SA, Zughaier SM, Khan MS, MEH C. Exploring the effect of image enhancement techniques on covid-19 detection using chest x-rays images. arXiv. 2020:2012.02238.

Rajpurkar P, Irvin J, Zhu K, Yang B, Mehta H, Duan T, Ding DY, Bagul A, Langlotz C, Shpanskaya KS, Lungren MP, Ng AY. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. ArXiv. 2017;1711:05225.

Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W, Frangi A, editors. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015, Lecture Notes in Computer Science, vol. 9351. Cham: Springer; 2015. https://doi.org/10.1007/978-3-319-24574-4_28.

Sakai T. Evaluating evaluation metrics based on the bootstrap. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery; 2006. p. 525–32. https://doi.org/10.1145/1148170.1148261.

Salvatier J, Wiecki T, Fonnesbeck C. Probabilistic programmingin python using pymc3. PeerJ Comput Sci. 2016. https://doi.org/10.7717/peerj-cs.55.

Shoeibi A, Khodatars M, Alizadehsani R, Ghassemi N, Jafari M, Moridian P, Khadem A, Sadeghi D, Hussain S, Zare A, Sani ZA, Bazeli J, Khozeimeh F, Khosravi A, Nahavandi S, Acharya UR, Shi P. Automated detection and forecasting of covid-19 using deep learning techniques: A review. arXiv. 2020:2007.10785.

Stirenko S, Kochura Y, Alienin O, Rokovyi O, Gordienko Y, Gang P, Zeng W. Chest x-ray analysis of tuberculosis by deep learning with segmentation and augmentation. In: 2018 IEEE 38th International Conference on Electronics and Nanotechnology; 2018.

Thomas AW, Heekeren HR, Müller KR, Samek W. Analyzing neuroimaging data through recurrent deep learning models. Front Neurosci. 2019;13. https://doi.org/10.3389/fnins.2019.01321.

Trunk GV. A problem of dimensionality: A simple example. IEEE Trans Pattern Anal Mach Intell PAMI-1. 1979:306–7.

Wang W, Xu Y, Gao R, Lu R, Han K, Wu G. Detection of sars-cov-2 in different types of clinical specimens. JAMA. 2020. https://doi.org/10.1001/jama.2020.3786.

Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017;2017:3462–71. https://doi.org/10.1109/CVPR.2017.369.

Yang Y, Tresp V, Wunderle M, Fasching PA. Explaining therapy predictions with layer-wise relevance propagation in neural networks. In: 2018 IEEE International Conference on Healthcare Informatics; 2018. https://doi.org/10.1109/ICHI.2018.00025.

Zhang D, Wang J, Zhao X. Estimating the uncertainty of average f1 scores. In: Proceedings of the 2015 International Conference on The Theory of Information Retrieval (ICTIR '15); 2015. p. 317–20. https://doi.org/10.1145/2808194.2809488.