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

Pedro R. A. S. Bassi*, Romis Attux

**Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, University of Campinas - UNICAMP. 13083-970, Campinas, SP, Brazil. E-Mail: p157007@dac.unicamp.br.* *Acknowledgments: This work was partially supported by CNPq (process 308811/2019-4) and CAPES.*

Abstract

We evaluated the generalization capability of deep neural networks (DNNs), trained to classify chest X-rays as COVID-19, normal or pneumonia, using a relatively small and mixed dataset.

We proposed a DNN architecture to perform lung segmentation and classification. It stacks a segmentation module (U-Net), an original intermediate module and a classification module (DenseNet201). We compared it to a DenseNet201.

To evaluate generalization, we tested the DNNs with an external dataset (from distinct localities) and used Bayesian inference to estimate the probability distributions of performance metrics, like F1-Score.

Our proposed DNN achieved 0.917 AUC on the external test dataset, and the DenseNet, 0.906. Bayesian inference indicated mean accuracy of 76.1% and [0.695, 0.826] 95% HDI with segmentation and, without segmentation, 71.7% and [0.646, 0.786].

We proposed a novel DNN evaluation technique, using Layer-wise Relevance Propagation (LRP) and the Brixia score. LRP heatmaps indicated that areas where radiologists found strong COVID-19 symptoms and attributed high Brixia scores are the most important for the stacked DNN classification.

External validation showed smaller accuracies than internal validation, indicating dataset bias, which segmentation reduces. Performance in the external dataset and LRP analysis suggest that DNNs can be trained in small and mixed datasets and detect COVID-19.

Keywords:

COVID-19 detection, Layer-wise Relevance Propagation, Lung Segmentation, Deep Neural Networks, Bayesian Inference

1. Introduction

Diagnosis is an important aspect for controlling COVID-19 spread and helping infected patients. The most common diagnosis method is reverse transcriptase-polymerase chain reaction (RT-PCR) (Wang et al. (2020)). However, this method is expensive, requires a considerable amount of time and is at high demand.

X-ray is one of the cheapest and most available COVID-19 alternative detection methods, mostly when we consider the disease spread in developing countries. But the images' analysis can be complicated. Thus, we think that artificial intelligence will be able to help in the creation of a reliable system to help clinicians in this task.

Deep neural networks (DNN) for COVID-19 detection were already proposed (Shoeibi et al. (2020)). However, some researchers raised concerns about the possibility of bias. 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 reveals that dataset biases may influence DNNs trained with mixed datasets. We do not think this test alone shows that the biases are strong enough to highly influence DNN decisions (a DNN is a very flexible model: if we delete the relevant information in the X-rays, it may be able to learn even very small dataset particularities). But we agree that the study proves the existence of dataset bias.

In March 2021, we still cannot find an open and large COVID-19 X-ray dataset, with all images collected from the same sources. This would be the best case scenario, as different classes would not present different biases. But COVID-19 classification datasets are generally relatively small and mixed, i.e., different classes have different sources, (Shoeibi et al. (2020)). Our objective is to understand how bias affects a DNN classifying pneumonia, healthy individuals and COVID-19 in a dataset like this. Therefore, we used external testing and validation (holdout) databases, whose X-rays as not from the hospitals that provided the training images. Furthermore, we analyzed if utilizing lung segmentation reduces bias and improves generalization.

We presented a large DNN, containing 3 stacked modules. The segmentation module is an U-Net (Ronneberger et al. (2015)), trained beforehand to receive X-rays and output segmentation masks (which are white in the lung regions and black everywhere else). Then comes an original intermediate module, which uses the U-Net output and the input image to erase the unimportant X-rays regions, and performs batch normalization. Finally, the classification module, a 201-layers dense neural network (Huang et al. (2016)), returns the probabilities of the X-ray containing healthy lungs, pneumonia or COVID-19. We compare this network to a DenseNet201.

We trained for classification with twice transfer learning, downloading ImageNet (Deng et al. (2009)) pretrained classification networks, training them on a large lung disease classification database (Wang et al. (2017)) and then 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. This lowers the reliability of these measurements when predicting the classifier real-world performance. Therefore, we quantified the measurements' uncertainty, using a Bayesian model (Zhang et al. (2015)) to estimate the performance metrics probability distributions and their statistics. We expanded the model in Zhang et al. (2015) to also estimate class specificity and mean specificity.

We used layer-wise Relevance Propagation (LRP, Bach et al. (2015)) to create X-ray heatmaps, showing which areas most contributed to the classification and which were more representative of other classes. These maps help us to better understand how DNNs make decisions, improving interpretability. They also show if the proposed DNN is truly ignoring the unimportant information outside the lungs and allow us to compare how the two trained models are classifying the images. Finally, the maps may be helpful for a clinician in finding the COVID-19 signs in an X-ray and evaluating the DNN prediction.

In this study, we introduced a new technique to compare DNN's analysis of COVID-19 X-rays to radiologists', using LRP and X-rays scored with the Brixia scoring system (a methodology created for radiologists to semi-quantitatively score COVID-19 severity in six lung zones, Borghesi and Maroldi (2020)). Do they 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 with higher Brixia scores?

2. Methods

2.1. The source databases

In this section, we describe the databases we utilized to create the datasets that we used in this study.

2.1.1. NIH ChestX-Ray14

ChestX-Ray14 is a very large dataset of frontal chest X-rays, containing 112120 images, from 30805 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 that is higher than 90% (Wang et al. (2017)).

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.

925 images, showing healthy patients, were extracted from this 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.

2.1.2. Montgomery and Shenzen datasets

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.

The Montgomery images also 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. 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.

2.1.3. COVID-19 database

COVID-19 image data collection (Cohen et al. (2020)) is one of the largest COVID-19 X-ray databases to date. The dataset also contains other kinds of pneumonia, such as MERS, SARS and bacterial, but we did not use them in this study. From this dataset, we obtained 475 COVID-19 X-rays (all the frontal COVID-19 X-rays).

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 the largest public collection of COVID-19 chest X-rays we found, and is also well documented. For example, 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. We have information about disease severity on some of them: from 87 patients, the 79.3% survived; from 118 patients, 61.9% needed ICU; from 77 patients, 61% were intubated; from 107 patients, 62.6% needed supplemental oxygen.

2.1.4. CheXPert

The CheXPert database contains images from the Stanford University Hospital. It has 224313 chest X-rays, from 65240 patients, showing 13 lung diseases or no findings (Irvin et al. (2019)). As in the NIH ChestX-Ray14 dataset, the images were automatically labeled, by the database authors, using Natural Language Processing to analyze radiological reports. The labels' estimated accuracy is also above 90%. The exceptions, in the images we used, are 8 pneumonia X-rays and 26 normal X-rays, which were manually labeled by three board certified radiologists (these images are part of the original CheXPert test dataset).

We used part of the CheXPert database in our classification dataset, as part of the external validation. 79 pneumonia and 79 healthy images were used, including the ones manually labeled by three radiologists. The normal images correspond to 73 patients, with a mean age of 49.5 years (with standard deviation of 18.5 years) and 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.

2.2. The segmentation dataset

This dataset was used to train an 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). As the youngest patient in the COVID-19 database was 20 years old, we only used X-rays from adult patients in the other classes as well.

The normal and tuberculosis images were all the X-rays in Montgomery and Shenzen datasets that had corresponding segmentation masks. The pneumonia X-rays were randomly selected from the NIH ChestX-Ray14 images. The COVID-19 images were randomly taken from the COVID-19 database (Cohen et al. (2020)).

As targets, this dataset contains segmentation masks 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 section 2.6.3.

2.2.1. Segmentation dataset subdivisions

We separated the segmentation dataset in 3: training, validation and testing. We used them to train our 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 only one subdivision.

For testing we used 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 here because this class is not present in our classification dataset, 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 generate a better segmentation neural network.

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. We kept both datasets balanced.

2.3. 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 925 normal images from ChestX-Ray14 (randomly selected, among adults). Pediatric patients were excluded because the COVID-19 database youngest patient is 20 years old and we thought that adding children to the other classes could create bias (training the DNN not to associate children with COVID-19).

2.4. External classification dataset

We used the external classification dataset for validation (holdout) 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 have 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 very small.

The images for the normal and pneumonia classes were extracted from the CheXPert database. 79 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.

2.5. Image preprocessing

We loaded the X-rays in greyscale, converted them to BGR (3 channels), applied histogram equalization and normalized the pixel values between 0 and 1. We also resized the images to 224x224.

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.

2.6. Training for segmentation

2.6.1. 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 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 an 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 CNNs.

An 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 in multiple points. More information can be found in Ronneberger et al. (2015).

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

2.6.2. Training with the Montgomery and Shenzen databases

We trained an 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 also 28 left or right) and horizontal flipping (50% chance).

We 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 also used a NVidia RTX 3080 GPU, with mixed precision.

Using the segmentation masks as targets, we trained the U-Net with cross-entropy loss, stochastic gradient descent (SGD) with 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.

2.6.3. 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 the 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 redone. 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.

2.6.4. 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 our segmentation dataset.

We created a new U-Net, with the same structure as the last one (Ronneberger et al. (2015)), to be trained using the segmentation dataset. For this process we used data augmentation (online) to avoid overfitting. All images were randomly rotated (between -40 and 40 degrees), 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).

We 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 figure 1.

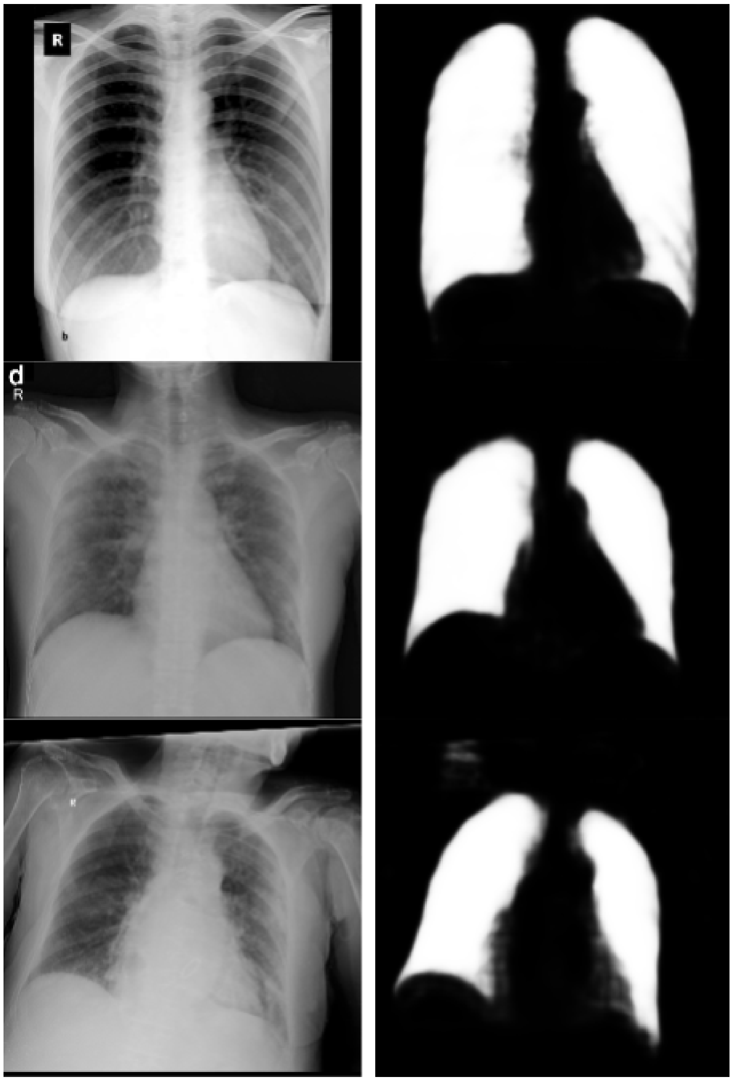


Figure 1: Examples of masks (created by the U-Net) and the corresponding COVID-19 X-rays.

2.7. 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 classification in the same manner, described in sections 2.7.2 and 2.7.3.

2.7.1. 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 network 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, we remove the unimportant regions from the X-ray and keep the lungs. Lastly, the module performs batch normalization on the multiplication output, our objective with this operation is to improve the DNN generalization.

The intermediate module output enters the second neural network (classification module), a DenseNet201 (Huang et al. (2016)), which predicts the chances of COVID-19, pneumonia or normal.

We decided to use a dense neural network as our classification module because it is a large DNN with good results in image classification (Huang et al. (2016)) and because its architecture was very successful in lung disease classification, obtaining F1 scores in pneumonia detection that surpassed radiologists', in Rajpurkar et al. (2017). The DenseNet201 was downloaded already pretrained on ImageNet (Deng et al. (2009)), a very large image classification dataset, with millions of samples.

Figure 2 shows our network structure and its three modules.

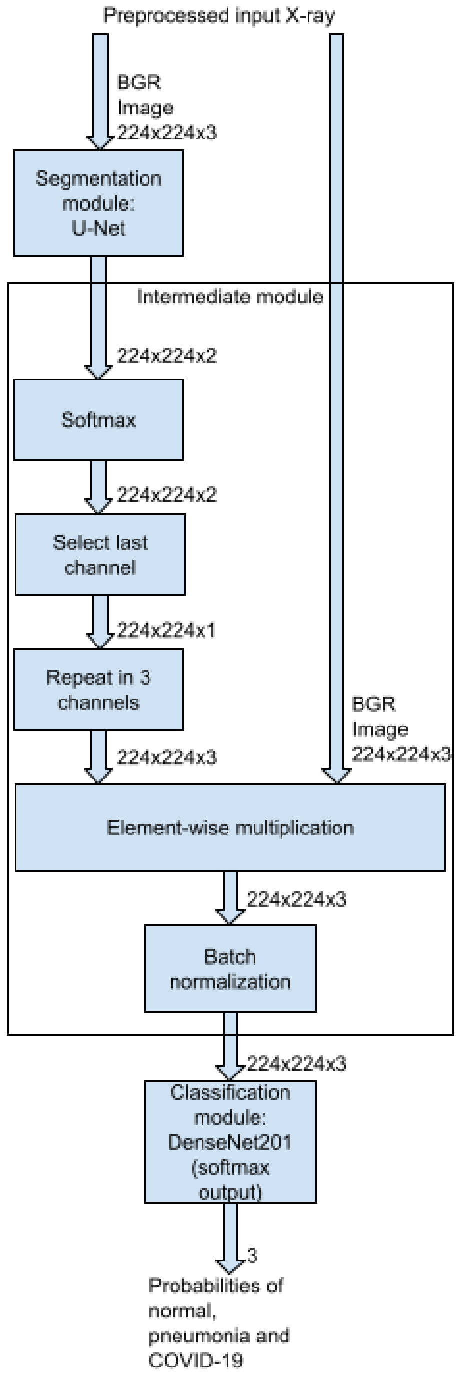


Figure 2: The structure of our proposed stacked DNN, for segmentation of lungs and classification.

2.7.2. 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 (2021b)). 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 original ones).

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

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.

2.7.3. 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 we added a dropout of 50% before it (in preliminary tests, we observed that regularization improved accuracy on the external datasets).

We also utilized 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 in 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, and flipping, with 50% chance, and not removing the original figures). In order 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. We decided to use these numbers 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 little amount of the images (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 were made. Thus, every X-ray was used during the training process. At each epoch the neural network received 11610 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 we could observe a clear overfitting. 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.

2.8. 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. (2016)), 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.

This map shows how each input feature (like a pixel in an image) affects the DNN output (Bach et al. (2015)). This propagation is conservative, a neuron receives a certain amount of relevance from its posterior layer and must propagate 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.

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 also provide insights about the X-rays, improving their own analysis.

We can start the relevance propagation by any output neuron. This decision defines the meaning of the colors in a heatmap (Montavon et al. (2019)). For example, if we start LRP by the neuron that classifies the COVID-19 class, red areas in the heatmap will indicate regions that the DNN associated with COVID-19, and blue areas will indicate 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. (2019)), 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 nothing went wrong.

2.9. The Bayesian performance evaluation

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)):

$\boldsymbol{\mu} \sim \mathrm{Dir}(\boldsymbol{\beta})$

$\boldsymbol{n} \sim \mathrm{Mult}(N, \boldsymbol{\mu})$

$\boldsymbol{\theta}_j \sim \mathrm{Dir}(\boldsymbol{\alpha}_j)$ for j=1,...,M

$\boldsymbol{c}_j \sim \mathrm{Mult}(n_j, \boldsymbol{\theta}_j)$ for j=1,...,M

$\psi = f(\boldsymbol{\mu}, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M)$

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 μ_j indicates the probability of a new sample belonging to class j. $\boldsymbol{\beta}$ indicates the hyper-parameters of the μ_j prior distribution. Choosing $\boldsymbol{\beta}$=[1,1,1] defines an 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. $\boldsymbol{\alpha}_j$ 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 an uniform prior.

ψ represents a function, calculated (in a deterministic manner) 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):

$tn_j = \sum_{u \neq j} \sum_{v \neq j} N \mu_u \theta_{u,v}$
$fp_j = \sum_{u \neq j} N \mu_u \theta_{u,j}$

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 $\boldsymbol{\theta}$:

$\text{Specificity}_j = tn_j/(tn_j + fp_j) = (\sum_{u \neq j} \sum_{v \neq j} \mu_u \theta_{u,v})/(\sum_{u \neq j} \sum_{v=1}^{M} \mu_u \theta_{u,v})$
Mean Specificity$= (1/M)(\sum_{j=1}^{M} \text{Specificity}_j)$

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 (Hoffman and Gelman (2011)), with 4 chains, 10000 tuning samples and 100000 samples after tuning.

3. 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).

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 (2004). 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 bootstraping is the suggested method for it (Hand and Till (2004)). We can

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

Table 1: Stacked DNN confusion matrix

		Predicted Class		
		Normal	Pneumonia	COVID-19
Real Class	Normal	43	0	7
	Pneumonia	14	24	12
	COVID-19	6	0	44

Table 2: Confusion matrix for the DNN without segmentation

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 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	0.74	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]

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

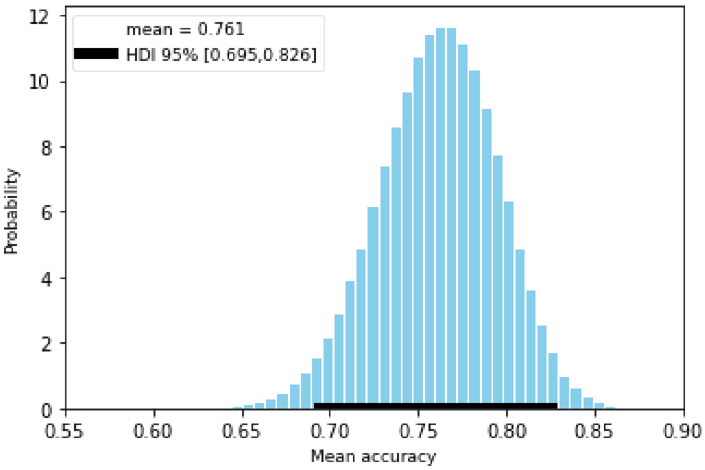


Figure 3: Mean accuracy posterior probability density estimation for the DNN with segmentation.

not use bootstraping in this study, as we are using an external test dataset and we have a small amount of COVID-19 X-rays.

In figures 3 to 6 we show the Bayesian estimations of mean accuracy (equal to miF1) and macro-averaged F1-Score. In figures 7 to 10 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 Bassi and Attux (2021a).

4. Discussion

In a previous study, we utilized a dataset that was very 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 dataset (Bassi and Attux (2021b)). 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 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, validation and test) and two images from the same patient were not allowed to be present

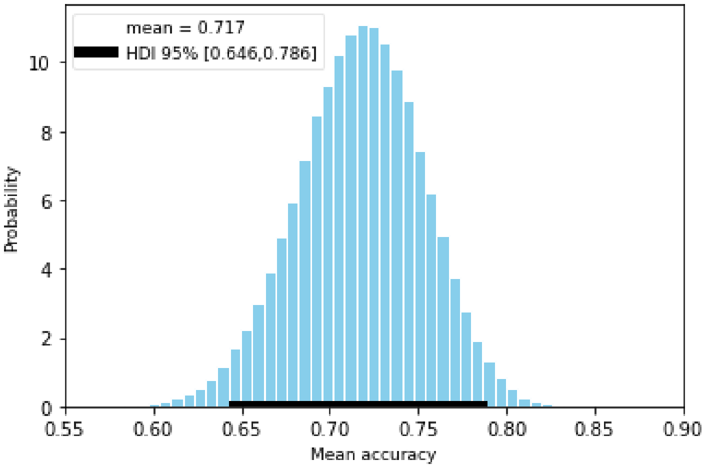


Figure 4: Mean accuracy posterior probability density estimation for the DNN without segmentation.

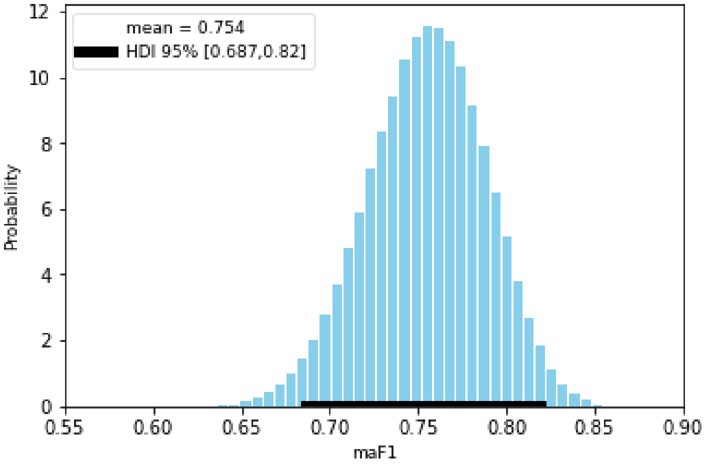


Figure 5: Macro-averaged F1-Score probability density estimation for the DNN with segmentation.

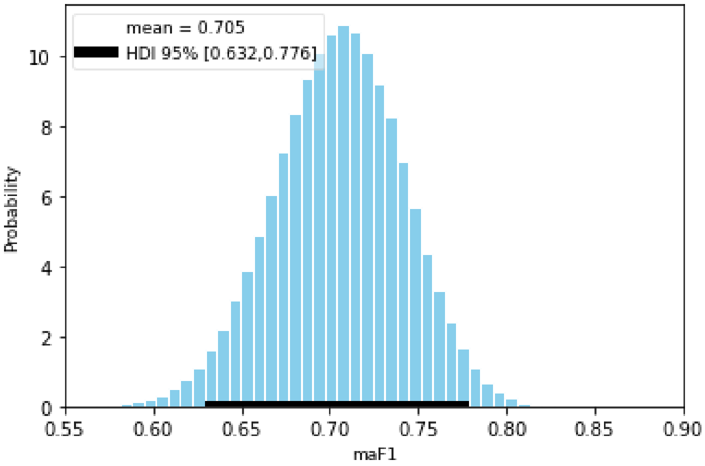


Figure 6: Macro-averaged F1-Score probability density estimation for the DNN without segmentation.

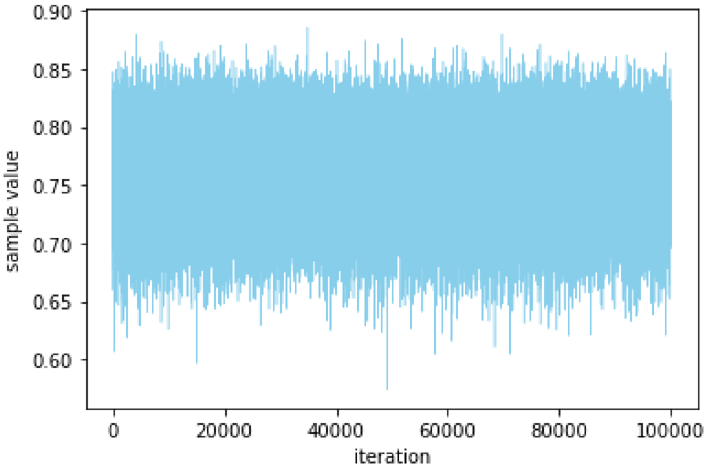


Figure 7: Mean accuracy trace plot for the DNN with segmentation.

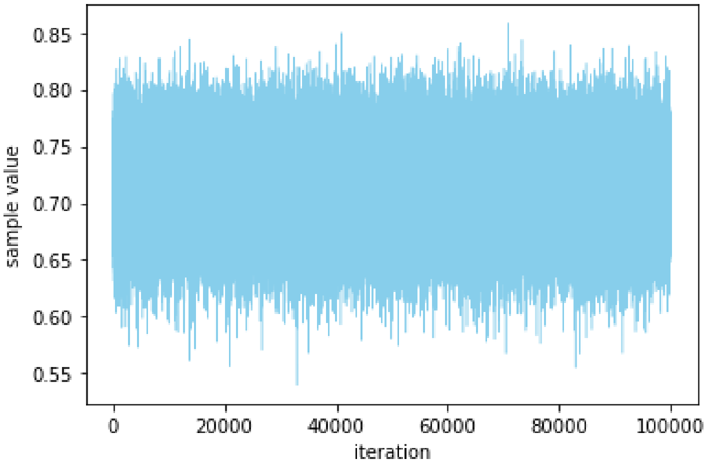


Figure 8: Mean accuracy trace plot for the DNN without segmentation.

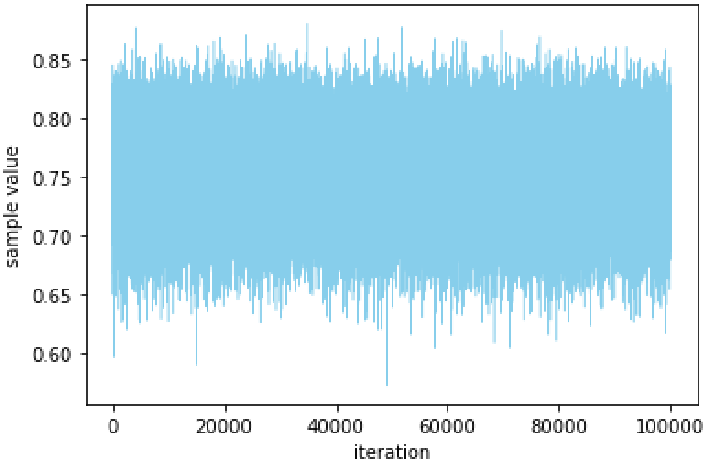


Figure 9: Macro-averaged F1-Score trace plot for the DNN with segmentation.

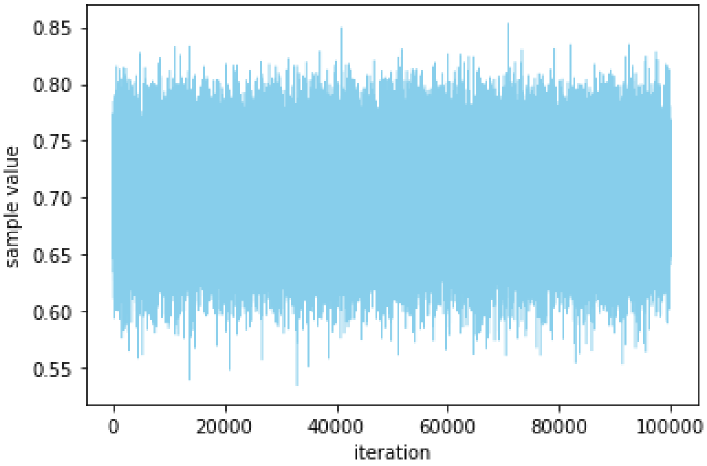


Figure 10: Macro-averaged F1-Score trace plot for the DNN without segmentation.

in two different datasets. We also used a patient split in Bassi and Attux (2021b). We conclude that evaluating DNNs in an external dataset can show significantly smaller accuracies, indicating that bias can artificially improve DNN performance metrics when working with mixed datasets and internal validation.

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

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 low number of the patients with a disease were miss-classified as healthy by our model.

We note that the 95% 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 figures 3 to 6. We suppose that the strongest reason for the large intervals is the small size of the test dataset, as using more test samples would increase the performance metrics confidence.

4.1. LRP and comparison with radiologists' analysis (using 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

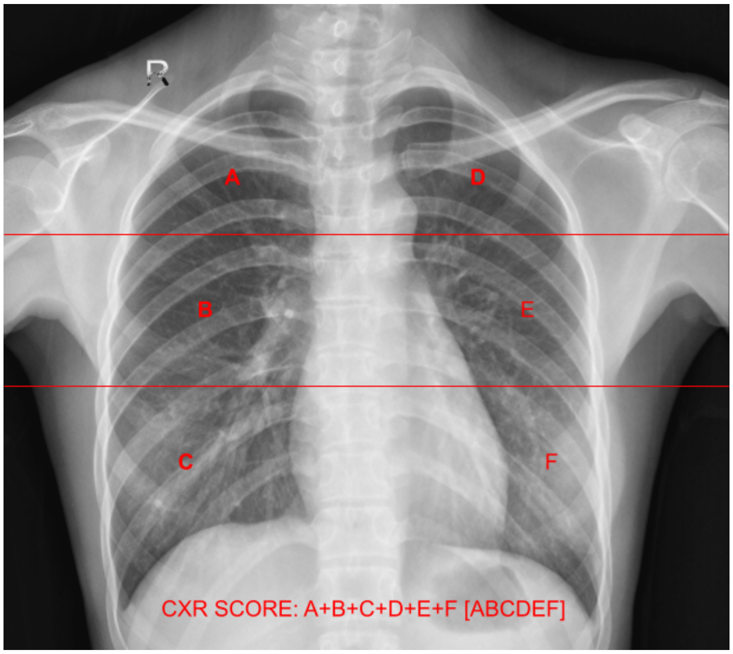


Figure 11: Illustration of the lung zones for the Brixia score and how the score is presented (based in Borghesi and Maroldi (2020)).

(2020), was created to grade the severity of COVID-19 cases. To score a chest X-ray, the radiologist divides the lungs in 6 parts (A, B, C, D, E and F), 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 anatomical landmarks, the authors suggest dividing the lungs in three equal zones. For each zone, the radiologist attributes a partial score, from 0 to 3. 0 means no abnormalities, 1 means interstitial infiltrates, 2 interstitial and alveolar infiltrates, with interstitial predominance, and 3 interstitial and alveolar infiltrates, with alveolar predominance. The overall Brixia score (from 0 to 18) is the sum of the partial scores (A+B+C+D+E+F), which are also presented, between square brackets, from A to F ([ABCDEF]). The system authors discovered that the score of patients who died was significantly higher than from discharged patients (Borghesi and Maroldi (2020)). Figure 11 illustrates the lines, zones and score presentation.

We propose comparing X-rays scored by radiologists, using the Brixia score, with heatmaps, created by LRP. The maps shows 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 COVID-19 symptoms. Checking this 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 mas show stronger symptoms and could increase the COVID-19 probability predicted by the DNN. Therefore, we may also be able check if there is a correlation between images with high Brixia scores and the higher predicted probabilities.

Besides presenting the scoring system, Borghesi and Maroldi (2020) also shows examples of COVID-19 X-rays, already scored by radiologists. These images are also part of our training dataset. In figure 12 we analyze, with our stacked DNN, three of them (the ones that had nothing written over the lungs). 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 figure 12 were taken from a 72-years-old man 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 very 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 figure 12 we see that the higher the 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 figure 12 also corroborates with the conclusion above.

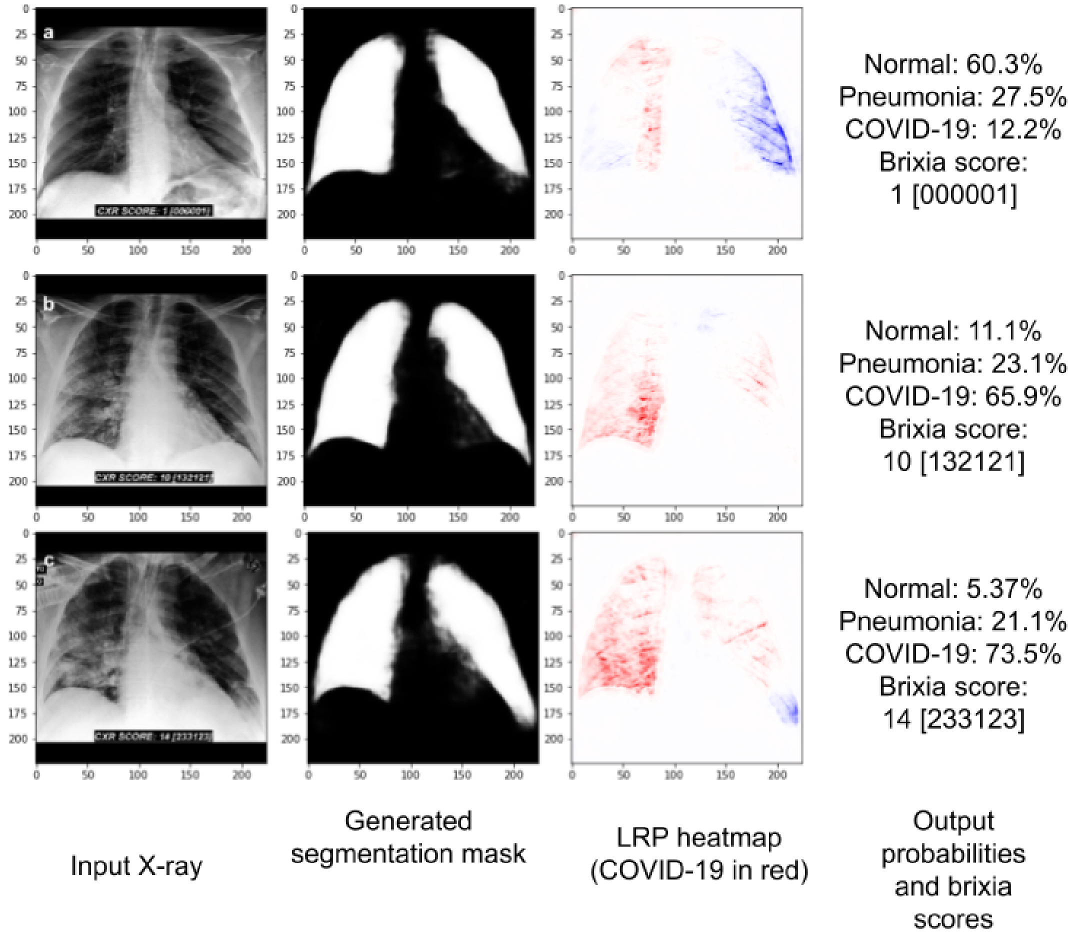
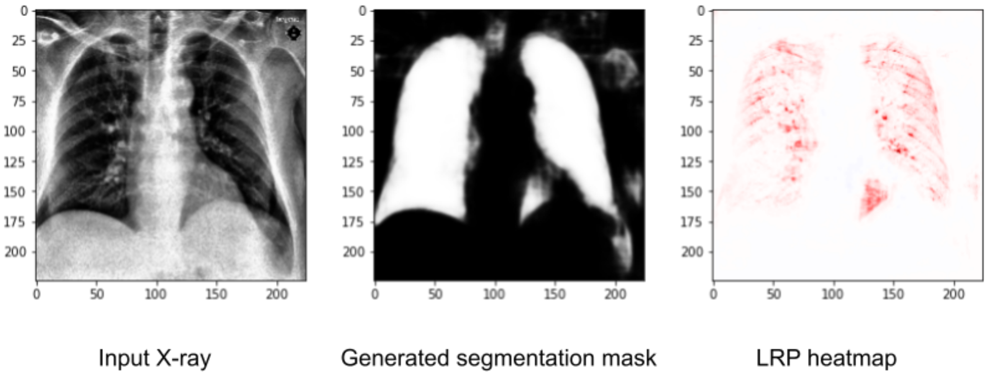


Figure 12: Stacked DNN analysis. From left to right: COVID-19 X-ray, mask generated by the segmentation module, relevance map and network predicted probabilities along Brixia scores (attributed by radiologists). The X-rays are training dataset images, available in high resolution in Borghesi and Maroldi (2020).



Input X-ray

Generated segmentation mask

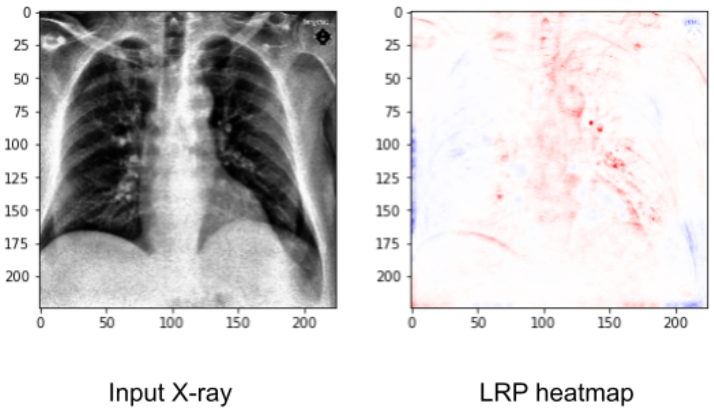
LRP heatmap

Figure 13: Stacked DNN analysis. From left to right: COVID-19 X-ray, mask generated by the segmentation module and relevance map. The X-ray is an image from our external test dataset, correctly classified as COVID-19.

In both heatmaps, we observe more relevance in the right lung, and it also has higher 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, the one with the higher 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 figure 12 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 also showed that our segmentation module and intermediate module work as intended, maintaining almost all relevance in the lung regions (as can be seen in figures 12 and 13). Figure 13 also analyzes the stacked DNN. It shows a COVID-19 input X-ray, the generated mask and LRP heatmap. But, unlike in figure 12, 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 figure 14, a LRP analysis for



Input X-ray

LRP heatmap

Figure 14: DenseNet201 analysis. From left to right: COVID-19 X-ray and relevance map. The X-ray is an image from our external test dataset, correctly classified by the DNN without segmentation.

the same X-ray in figure 13, 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 in the map were associated with the COVID-19 class, while blue areas were associated with the other classes.

We observe, in figure 14, 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 important areas might indicate dataset biases learned by the DNN. However, some COVID-19 signs, indicated in the heatmap in figure 13 can still be seen on figure 14 (mostly on the left lung).

5. 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 separating a database in test, validation and training datasets). Our previous study (which used a database similar to our current classification training dataset and utilized internal validation), Bassi and Attux (2021b), and many other works that detected COVID-19 using

DNNs without external validation (Shoeibi et al. (2020)) showed accuracies above 90%. This performance discrepancy may indicate that utilizing mixed datasets creates bias, which artificially improves internal validation accuracies and performance metrics, as the study in Maguolo and Nanni (2020) suggests.

The utilization of 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 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 would larger COVID-19 X-ray databases 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, because these regions' relevance was very small (showing almost no color in the maps). Comparing X-rays scored by radiologists using the Brixia score to our stacked DNN outputs and heatmaps showed that, normally, regions with higher partial scores also had higher COVID-19 LRP relevance. Also, X-rays with higher overall scores were associated with higher COVID-19 predicted probabilities. This may indicate that radiologists and our stacked DNN look for the same signs of COVID-19 in a radiography. Unfortunately, our DNN could not correctly classify an X-ray where radiologists also found few symptoms of COVID-19 (the upper X-ray in figure 12, with a small overall Brixia score, of only 1).

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

Although we conclude that mixed dataset bias is significant, our DNNs' performance on an external dataset and LRP analysis indicates 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 a large, open and high quality COVID-19 X-ray database, with all classes collected from the same sources, to better avoid dataset bias, improve generalization and increase performance metrics reliability. Our DNNs' performance in 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. Furthermore, even though we utilized external validation, clinical tests are needed to further ensure that the performances we observed in this study are replicable in a real-world scenario.

References

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K.T., Montavon, G., Samek, W., Müller, K.R., Dähne, S., Kindermans, P.J., 2019. innvestigate neural networks! Journal of Machine Learning Research 20, 1–8. URL: http://jmlr.org/papers/v20/18-540.html.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W., 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE 10, 1–46. URL: https://doi.org/10.1371/journal.pone.0130140, doi:10.1371/journal.pone.0130140.

Bassi, P.R.A.S., Attux, R., 2021a. Covid-19 detection with lung segmentation. URL: https://github.com/PedroRASB/COVID-19-detection-with-lung-segmentation.

Bassi, P.R.A.S., Attux, R., 2021b. A deep convolutional neural network for covid-19 detection using chest x-rays. arXiv:2005.01578.

Borghesi, A., Maroldi, R., 2020. Covid-19 outbreak in italy: experimental chest x-ray scoring system for quantifying and monitoring disease progression. La radiologia medica 125. doi:10.1007/s11547-020-01200-3.

Cai, Q., Liu, X., Guo, Z., 2018. 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), pp. 1–6.

Candemir, S., Jaeger, S., Palaniappan, K., Musco, J.P., Singh, R.K., Xue, Z., Karargyris, A., Antani, S., Thoma, G., McDonald, C.J., 2014. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. IEEE Transactions on Medical Imaging 33, 577–590.

Cohen, J.P., Morrison, P., Dao, L., 2020. Covid-19 image data collection. arXiv 2003.11597 URL: `https://github.com/ieee8023/covid-chestxray-dataset`.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09.

Hand, D., Till, R., 2004. A simple generalisation of the area under the roc curve for multiple class classification problems. Machine Learning 45, 171–186.

Heo, S.J., Kim, Y., Yun, S., Lim, S.S., Kim, J., Nam, C.M., Park, E.C., Jung, I., Yoon, J.H., 2019. Deep learning algorithms with demographic information help to detect tuberculosis in chest radiographs in annual workers' health examination data. International Journal of Environmental Research and Public Health 16, 250. doi:`10.3390/ijerph16020250`.

Hoffman, M.D., Gelman, A., 2011. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. `arXiv:1111.4246`.

Howard, J., Ruder, S., 2018. Universal language model fine-tuning for text classification, pp. 328–339. doi:`10.18653/v1/P18-1031`.

Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2016. Densely connected convolutional networks. `arXiv:1608.06993`.

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.A., Halabi, S.S., Sandberg, J.K., Jones, R., Larson, D.B., Langlotz, C.P., Patel, B.N., Lungren, M.P., Ng, A.Y., 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. `arXiv:1901.07031`.

Jaeger, S., Karargyris, A., Candemir, S., Folio, L., Siegelman, J., Callaghan, F., Xue, Z., Palaniappan, K., Singh, R.K., Antani, S., Thoma, G., Wang, Y., Lu, P., McDonald, C.J., 2014. Automatic tuberculosis screening using chest radiographs. IEEE Transactions on Medical Imaging 33, 233–245.

Maguolo, G., Nanni, L., 2020. A Critic Evaluation of Methods for COVID-19 Automatic Detection from X-Ray Images. arXiv e-prints , arXiv:2004.12823arXiv:2004.12823.

Malivenko, G., 2018. pytorch2keras. URL: https://github.com/nerox8664/pytorch2keras.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, K.R., 2019. Layer-wise relevance propagation: An overview, in: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer International Publishing, pp. 193–209.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D.Y., Bagul, A., Langlotz, C., Shpanskaya, K.S., Lungren, M.P., Ng, A.Y., 2017. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. CoRR abs/1711.05225. URL: http://arxiv.org/abs/1711.05225, arXiv:1711.05225.

Ronneberger, O., P.Fischer, Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer. pp. 234–241. URL: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a. (available on arXiv:1505.04597 [cs.CV]).

Sakai, T., 2006. 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, New York, NY, USA. p. 525–532. URL: https://doi.org/10.1145/1148170.1148261, doi:10.1145/1148170.1148261.

Salvatier, J., Wiecki, T., Fonnesbeck, C., 2016. Probabilistic programming in python using pymc3 doi:10.7287/PEERJ.PREPRINTS.1686V1.

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

Stirenko, S., Kochura, Y., Alienin, O., Rokovyi, O., Gordienko, Y., Gang, P., Zeng, W., 2018. Chest x-ray analysis of tuberculosis by deep learning with segmentation and augmentation, in: 2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO), pp. 422–428.

Thomas, A.W., Heekeren, H.R., Müller, K.R., Samek, W., 2019. Analyzing neuroimaging data through recurrent deep learning models. Frontiers in Neuroscience 13, 1321. URL: http://dx.doi.org/10.3389/fnins.2019.01321, doi:10.3389/fnins.2019.01321.

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

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M., 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3462–3471.

Yang, Y., Tresp, V., Wunderle, M., Fasching, P.A., 2018. Explaining therapy predictions with layer-wise relevance propagation in neural networks, in: 2018 IEEE International Conference on Healthcare Informatics (ICHI), pp. 152–162.

Zhang, D., Wang, J., Zhao, X., 2015. Estimating the uncertainty of average f1 scores. Proceedings of the 2015 International Conference on The Theory of Information Retrieval .