

Article

The Practicality of Deep Learning Algorithms in COVID-19 Detection: Application to Chest X-ray Images

Abdulaziz Alorf 

Department of Electrical Engineering, College of Engineering, Qassim University, Buraydah 51452, Saudi Arabia; aalorf@qu.edu.sa; Tel.: +966-56-111-6270

Abstract: Since January 2020, the outbreak of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) has affected the whole world, producing a respiratory disease that can become severe and even cause death in certain groups of people. The main method for diagnosing coronavirus disease 2019 (COVID-19) is performing viral tests. However, the kits for carrying out these tests are scarce in certain regions of the world. Lung conditions as perceived in computed tomography and radiography images exhibit a high correlation with the presence of COVID-19 infections. This work attempted to assess the feasibility of using convolutional neural networks for the analysis of pulmonary radiography images to distinguish COVID-19 infections from non-infected cases and other types of viral or bacterial pulmonary conditions. The results obtained indicate that these networks can successfully distinguish the pulmonary radiographies of COVID-19-infected patients from radiographies that exhibit other or no pathology, with a sensitivity of 100% and specificity of 97.6%. This could help future efforts to automate the process of identifying lung radiography images of suspicious cases, thereby supporting medical personnel when many patients need to be rapidly checked. The automated analysis of pulmonary radiography is not intended to be a substitute for formal viral tests or formal diagnosis by a properly trained physician but rather to assist with identification when the need arises.

Keywords: COVID-19; SARS-CoV-2; radiography; CT scans; X-ray; deep learning



Citation: Alorf, A. The Practicality of Deep Learning Algorithms in COVID-19 Detection: Application to Chest X-ray Images. *Algorithms* **2021**, *14*, 183. <https://doi.org/10.3390/a14060183>

Academic Editors: Christian Mata Miquel, Gilberto Ochoa-Ruiz and Raul Benitez

Received: 6 May 2021
Accepted: 7 June 2021
Published: 13 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since January 2020, the outbreak of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) has affected the entire world, with devastating consequences. Infection with SARS-CoV-2 causes a respiratory disease that triggers a wide range of reactions among the infected, from asymptomatic infections to severe disease and death. No vaccinations or treatments are yet known as of October 2020. The disease is mainly spread through droplets of saliva or the nasal discharge of an infected person. Thus far, it has exhibited a high rate of transmission.

Although it is not clear whether the disease can be spread during the pre-symptomatic stage, its reproductive number is estimated to be around 2.5, surpassing that of influenza and other better-known viruses. Moreover, the fatality rate also appears to be much higher than that of seasonal influenza, especially in elderly people and people with certain existing physical conditions. This makes it especially difficult to confine and control how it spreads across multiple regions in the world. As of June 2020, the disease had spread to more than 200 countries, producing more than 6 million cases and 430,000 deaths [1].

Since the outbreak started, there have been multiple research efforts attempting to provide answers to the many questions concerning the virus and containing its spread. These questions include how long SARS-CoV-2 remains viable on different surfaces [2], the effects of temperature on how the virus spreads [3,4], whether or not it can be transmitted by asymptomatic people [5], how it affects the cardiovascular system [6], and whether or not certain substances are appropriate for treating the disease [7]. The authors of [8]

provide a comprehensive survey of the various research efforts that have been taking place in the last few months.

On the other hand, since coronavirus disease 2019 (COVID-19) is a new viral disease, presumably no one had immunity and most people are considered susceptible to infection, leaving very high numbers of suspicious cases that need to be diagnosed. In the context of the pandemic and a generalized fear of being infected or spreading the disease, many people with symptoms such as cough, fever, sneezing, or throat pain presume that they might be infected and approach medical centers for medical diagnosis. Furthermore, in many buildings with public access, such as airports, new protocols for allowing entrance have been established. For instance, in many countries, if one has even a slight fever, one is considered to be potentially infected with COVID-19 and hence not allowed into shopping centers or municipal buildings.

In this scenario, medical personnel must determine whether a person is really infected or not as soon as possible in order to (1) take proper care of the infected person and (2) prevent the further dissemination of the disease. This necessitates methods that can diagnose the disease or at least provide more indicators that help to determine the actual medical condition of the person presumably infected with COVID-19. Based on these indicators, medical staff can determine whether a suspicious case deserves to be considered highly likely to be infected (which requires special care and treatment) or whether it can be dismissed as a potential COVID-19 infection.

Many techniques based on machine learning have been proposed for controlling the disease. These techniques solve various problems related to recognizing if a person is wearing a facemask [9,10], non-contact sensing for COVID-19 [11], the automatic detection of the disease [12–16], the prediction of the number of infected cases [17], and optimizing the process of the admission of COVID-19 patients to hospitals [18].

The main method used to formally diagnose COVID-19 consists of viral tests. Samples from an individual's respiratory system (such as swabs from the inside of the nose) are examined to determine whether the patient is currently infected with SARS-CoV-2 [19,20]. Some of these tests can provide results in a matter of hours, whereas other tests must be sent to a laboratory for analysis, providing results in a few days. These tests tend to be costly and can consume a significant amount of resources for certain populations or countries [21]. Moreover, due to the worldwide situation, these tests are scarce in many countries, making it difficult to conduct them on a wider scale. Hence, they cannot always be performed upon the recognition of the first symptom suggesting potential infection with COVID-19.

Since COVID-19 generates a respiratory disease that mainly affects the lungs, medical staff usually resort to the use of visual images of the lungs using computed tomography (CT) and X-rays (RX) to assess the medical conditions of their patients. People with respiratory diseases usually exhibit distinct marks in their lungs that can be spotted by a trained eye. These marks exhibit different patterns according to the type of disease affecting the patient [22–25].

In cases where a person exhibits fever or any other symptom and viral tests are difficult to perform, the analysis of CT scans or RX images can provide strong support to confirm or refute the hypothesis that someone is infected with COVID-19. Nonetheless, as previously stated, highly trained personnel are required to provide accurate interpretations of the aforementioned images and diagnose whether or not the patient is infected with COVID-19.

This rationale leads to the question of whether or not we can train machines that can interpret images to identify patients with COVID-19. The same question holds for other pulmonary diseases, but due to the pandemic being ongoing (as of June 2020), we mainly aim to detect the aforementioned disease. It is highly important to emphasize that it is by no means implied that a machine-obtained interpretation of a pulmonary image can be a substitute for a more comprehensive diagnosis made by a highly trained physician. However, this technique could eventually be used to help medical personnel to quickly obtain an indication of whether the symptoms of a given patient imply infection with COVID-19.

Related Work

There have recently been many remarkable efforts to exploit machine learning tools to facilitate the analysis of images for the diagnosis of COVID-19. These efforts include initiatives to collect worldwide images and make them public [26–30] as well as attempts to automate the analysis of CT scans or lung X-ray images. Mazzilli et al. [15] developed a method for automatically evaluating lung conditions using CT scans to determine COVID-19 infection. Their method is based on individually optimized Hounsfield unit (HU) thresholds. Alshazly et al. [16] investigated the practicality of using deep learning models trained on chest CT images to detect COVID-19 efficiently. In [31], the authors offer a review of techniques from the image processing field that are being used extensively to identify COVID-19 in CT and RX images. Guo et al. [32] employed machine learning tools to predict the infectivity of COVID-19. In [33], a forecasting model for predicting the spread of COVID-19 is proposed. The studies [34–36] focus on training convolutional neural networks (CNNs) to detect COVID-19 in CT chest scans.

Artificial neural networks (ANNs) are biologically inspired algorithms that mimic the computational aspects of the human brain. They contain multiple connected computational units (resembling individual neurons) that transmit and receive information according to specific rules, rules that dictate the operation modes of the networks. CNNs are one type of ANN that have been proven to be very powerful for computer vision applications [37,38]. In recent years, a new class of ANNs named spiking neural networks (SNNs) has also been intensively investigated because of their promising computational capabilities [39,40]. However, the generic and effective training methods for these networks still remain cumbersome and, therefore, their fields of application are much more specific than those of their predecessors. CNN algorithms have the advantage that they can be run very efficiently on standard computers that are equipped with graphics processing units, without the need to use dedicated platforms based on neuromorphic systems [41,42].

Generally speaking, CT scans can be more accurate for revealing lung conditions and diagnosing related diseases than RX images. However, CT scans can be almost one order of magnitude more expensive than their RX counterparts, and they also take much longer complete (about 40 min) than X-ray imaging, which can be completed in about 5 min. Moreover, CT scanners are scarce in certain regions of the world, whereas radiography systems are much more common [43]. Therefore, there is also great motivation to explore the feasibility of automatically identifying COVID-19 in RX images. The main aim of this work was to contribute evidence that supports the following hypothesis: *Deep learning can be used to recognize COVID-19 infections from lung X-ray images.*

Duran-Lopez et al. [12] developed a deep-learning-based system called COVID-XNet, which preprocesses chest X-ray images and then classifies them as normal or COVID-19-infected. Bourouis et al. [13] proposed using a Bayesian classifier framework to classify chest X-ray images as either infected or normal. Features are extracted from segmented lungs and then modeled using a proposed mixture model; finally, the Bayesian classifier is applied to perform the classification. Alam et al. [14] combined/fused two types of features (early fusion) and then used them to train a VGGNet classifier for the detection of COVID-19 from chest X-ray images. The fused features were extracted from X-ray images using a histogram-oriented gradient (HOG) and convolutional neural networks (CNNs). Narin et al. [44] analyzed the performance of three CNN models—ResNet50, InceptionV3, and Inception ResNetV2—for detecting COVID-19 from RX pulmonary images. However, only 50 COVID-19-positive and 50 COVID-19-negative images were used in the study. Farooq et al. [45] also used a ResNet50 CNN trained with the COVID-Net dataset [26] to distinguish COVID-19 from bacterial pneumonia, non-COVID-19 viral pneumonia, and normal cases. Since the original set had only 68 COVID-19-positive images, they had to perform significant data augmentation. Due to the scarcity of available images, in [46] a capsule network model is proposed for examining lung RX images. In [47], the authors developed a CNN architecture that leverages the projection–expansion–extension design.

Zebin et al. [48] implemented three CNN-based classifiers for pulmonary radiographies by using VGG16, ResNet50, and EfficientNetB0 as backbones. In [49], based on the pre-trained AlexNet CNN, a relief feature selection mechanism is implemented that is then used with a support vector machine to perform the classification. Goel et al. [50] proposed a CNN, named OptCoNet, whose hyperparameters were optimized using the grey wolf optimizer (GWO) algorithm. In [51], a ResNet CNN architecture was utilized to derive three different classifiers, which were then assembled to obtain a more accurate classifier. Jain et al. [52] compared three architectures of CNNs for the classification of pulmonary radiographies into three classes: normal, COVID-19, and other pneumonia.

In this work, we collected images from all the relevant sources that we could find [26,28,29,53–55]. Since the number of available images has been continually increasing, we have added more and more images to our model but also had to discard a significant number of them due to constraints that will be elaborated on in Section 2.

Pneumonia can occur due to several different causes. It can be caused by bacteria or viruses, and there are even multiple types of viral pneumonia. Hence, we intended to train our model to distinguish cases of COVID-19 from cases of any other viral or bacterial disease. Consequently, we had to procure a set of images that, in addition to images of COVID-19 infection, also included images of bacterial infections, viral infections (other than COVID-19), and no infection. We collected images from all relevant open access sources that we found until November 2020. These sources can be found in [26,28,53–55] and were used among many other similar publications [12–14,48–52].

It is worth mentioning that, in general, these efforts are still in the research stage and not yet intended as production models ready for clinical diagnosis. In order to attain that level, many more properly labeled images should be made available for training and testing. We are working continuously to improve them as new data become available. However, the main goal of this work was to illustrate the process and feasibility of developing serious classifiers of X-ray images to indicate the existence of COVID-19 infection. Another goal of this work was to compare the performance of two renowned CNN architectures, VGG16 [56] and Xception [57]. The main contribution of this article consists of the description of two CNN-based classifiers that can distinguish pulmonary radiographies that include COVID-19-infected lungs from those that do not. Our novel classifiers exhibit a performance comparable to that found in other similar state-of-the-art studies [12–14,48–52]. In fact, our classifiers even slightly outperform some of these others in certain metrics.

The remainder of this paper is structured as follows. Section 2 describes the image collection and preparation processes as well as the architecture of the convolutional neural network implemented for the classifier. Section 3 elaborates on the obtained results and discusses potential directions for improvement. Finally, Section 4 presents our main highlights and observations about this work.

2. Materials and Methods

This work aimed to develop algorithms that could determine whether pulmonary radiography images belong to people infected with COVID-19. To this end, we made use of artificial neural networks (ANNs). ANNs are computing algorithms that are somewhat inspired by biological neural networks. Generally speaking, ANNs constitute sequences of mathematical operations that are applied to given inputs, yielding corresponding outputs. These mathematical operations are usually deterministic, meaning that in a given neural network, the same input will always yield the same output. These mathematical operations are defined in terms of numerical parameters, usually referred to as weights. The weights characterize how the preliminary results obtained at each step in the network are propagated forward during the sequence of operations.

There are multiple types of neural network, mainly differing in terms of the sequences and types of mathematical operations performed on their inputs. Among them, convolutional neural networks (CNNs) constitute a specific type that is well suited for image

processing and computer vision. CNNs implement sequences of convolutional operations on their input images, represented by tensors, to extract the most relevant features that characterize the images. CNNs may have thousands of weights. Unlike spiking neural networks, CNNs do not take into consideration the signal propagation—i.e., signals are assumed to propagate instantaneously across the network.

In order to use CNNs for image classification, one should determine what weights are optimal for the classification problem at hand. Determining the optimal weights of a CNN is analogous to computing the optimal coefficients of a polynomial function for best fitting. This process is known as training. In order to train a CNN such that it can detect whether pulmonary radiography exhibits signs of COVID-19, several radiographies from COVID-19-infected and uninfected people should be collected. These images will be used with their corresponding labels (infected or uninfected) to adjust the weights of a CNN (the training process). As a result, the network can properly classify new images as COVID-19-infected or uninfected. More details are provided in the following.

2.1. Image Collection

The search for and collection of a high number of images is a key process in these studies. In order to obtain a classifier model that can be generalized well for unseen images, we need to procure sufficient images for training and learning the distinct features that characterize any targeted label. By generalizing well, we mean that it can properly classify images that were not used in the training process.

There is no conclusive answer as to how many images are required to develop a model of this type. However, if it performs satisfactorily on the testing set, assuming that the set is sufficiently representative of the many possible RX pulmonary images, we can expect that it will perform satisfactorily with other unseen images.

Since lung infections can have multiple causes (not only COVID-19), it is necessary to endow the model with the ability to distinguish COVID-19 infections from other illnesses. Hence, it was necessary to include cases of other possible conditions within the training and testing sets. Great effort was made to obtain as many good-quality images as possible of various types of pathologies. Figures 1 and 2 illustrate samples of pulmonary radiographies of uninfected and COVID-19-infected patients, respectively.

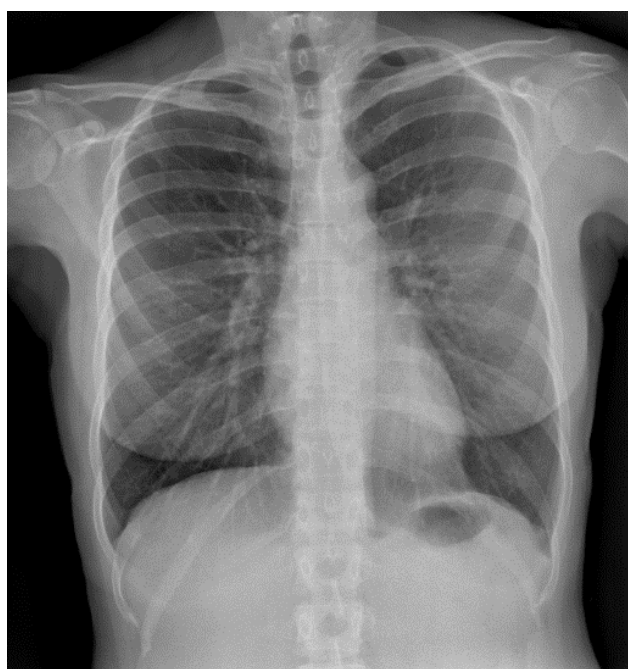


Figure 1. Pulmonary radiography of an uninfected person.

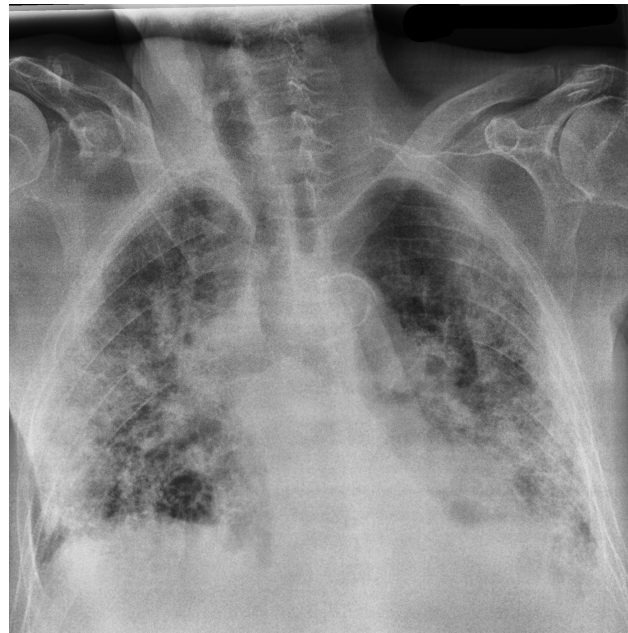


Figure 2. Pulmonary radiography of a person infected with COVID-19.

Since the outbreak is quite recent, there are few publicly available sources of images. However, there are a few highly valuable initiatives to make images publicly available through repositories for practitioners. The images used in this work were extracted from the following sources: [26,28,29,53–55]. Table 1 lists the number of images extracted from each source. The first column denoted as “Ref.” indicates the reference from which the images were retrieved. The second column shows the number of images that were initially downloaded from the mentioned source. We opted to constrain the geometry of the images used in our developments (this is elaborated on in Section 2.2). The third column of Table 1 states the number of retrieved images that satisfied the imposed constraints. Finally, the fourth column, named “Label Details”, enumerates the labels associated with the used images.

Table 1. Labels of the collected images and sample counts.

Ref.	Num. of Retrieved Images	Images Satisfying Size Constraints	Label Details
[28,29]	542	217	168 images labeled as “COVID-19” 12 images labeled as “No Finding” 3 images labeled as “Legionella” 6 images labeled as “SARS (no COVID-19)” 6 images labeled as “Pneumocystis” 6 images labeled as “Streptococcus” 1 image labeled as “Chlamydophila” 3 images labeled as “Escherichia Coli” 5 images labeled as “Klebsiella” 2 images labeled as “Pneumonia” 3 images labeled as “Varicella” 1 image labeled as “Lipoid” 1 image labeled as “Influenza”
[26]	55	15	9 images labeled as “COVID-19” 1 image labeled as “No Finding” 5 images labeled “Unclassified”

Table 1. Cont.

Ref.	Num. of Retrieved Images	Images Satisfying Size Constraints	Label Details
[53]	238	11	2 images labeled as "COVID-19" 6 image labeled as "No Finding" 3 images labeled "Unclassified"
[54]	5906	151	18 images labeled as "COVID-19" 100 images labeled as "No Finding" 21 images labeled as "Bacteria" 12 images labeled as "Other Viruses"
[55]	2683	2683	205 images labeled as "COVID-19" 1341 image labeled as "No Finding" 1137 images labeled as "Pneumonia"

2.2. Image Preprocessing

Not every image we found could be used in this work. A couple of constraints were imposed in an attempt to obtain a solid classifier. Firstly, not every image found in the repositories was an X-ray radiograph. Some of them were computed tomograms, and since this work only aimed to develop a classifier for RX, the CT images were removed from the collected set. Three types of views were found among the RX images: posterior–anterior (PA), anterior–posterior (AP), and lateral views. Among the images found, the number of lateral views was much lower than the number of AP and PA ones. Moreover, lateral views are usually less useful for diagnosis. Therefore, we discarded the lateral views and pursued the construction of a classifier that handles only PA and AP views.

In general, convolutional neural networks (and other types of neural networks) require inputs of a fixed and predefined size. However, among the collected PA and AP images, there were images of multiple sizes and aspect ratios $AR = h/w$, where h is the height of the image and w denotes its width, both measured by their numbers of pixels. Therefore, the images had to be resized to a common size for appropriate usage. If images with different original aspect ratios are resized to a common aspect ratio, at least one will necessarily lose its initial AR . This might affect the appearance of distinct features that might indicate the presence of COVID-19. Hence, we tried to avoid significant changes in the aspect ratios with respect to their original values. We decided to use images that originally had square (or almost square) shapes—i.e., an equal number of pixels for the width and height. Since we resized the images into square ones, in order to avoid the images becoming highly distorted, we selected only images whose original aspect ratios were very close to 1: $0.95 \leq AR \leq 1.05$. These images were resized to a new width and height of $h = w = 400$ pixels each. We picked the value of $h = w = 400$ because, among the images that satisfied the constraint on AR , there were many good-quality images with a h and w both larger than 400 (in this statement, the quality that we refer to is a quality assessed visually, avoiding blurred images, images with too many annotations, or images where the lungs appeared significantly shifted from the center). In order to avoid expanding the images, possibly generating pixelation and misleading information, we only allowed images whose sizes could be maintained or reduced. Hence, images that initially did not satisfy the constraint on the aspect ratio or that had at least one dimension smaller than 400 pixels were removed from the set.

Lastly, to achieve more robust results in every iteration of the training process, the images were rotated by random angles of up to 5 degrees clockwise or counterclockwise. Figure 3 illustrates the whole process sequence.

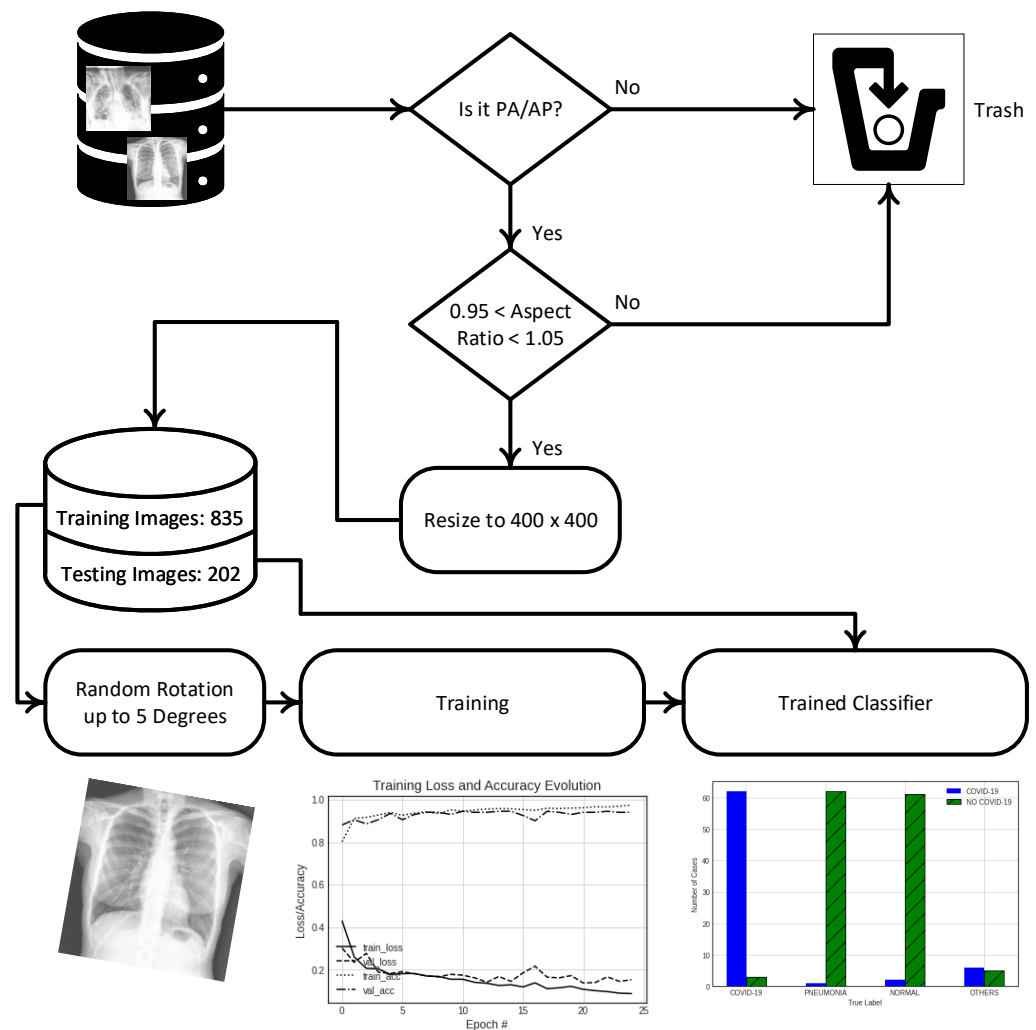


Figure 3. Schematic view of the entire process from data collection to testing.

2.3. Distribution of Samples in the Training and Testing Sets

The total number of usable images was 3077 (summing up the third column of Table 1), with the labels given according to Table 1. However, we opted to keep a dataset with similar numbers of COVID-19-positive and -negative cases to avoid biasing the training process. According to Table 1, there were 402 positive cases, and we randomly picked 400 “No Finding” cases, 200 “Pneumonia” cases, and 35 cases with infections that were neither “COVID-19” nor “Pneumonia”. In total, we selected 1037 images, which were split into a testing and a training set. The former consisted of 200 images that were carefully balanced in number with the following distribution:

- 65 images labeled as “COVID-19”;
- 63 images labeled as “Pneumonia”;
- 63 images labeled as “Normal”;
- 2 images labeled as “SARS (no COVID-19)”;
- 1 image labeled as “Legionella”;
- 2 images labeled as “Pneumocystis”;
- 2 images labeled “Streptococcus”;
- 1 image labeled as “Escherichia coli”;
- 1 images labeled as “Klebsiella”;
- 1 image labeled as “Varicella”;
- 1 image labeled as “Lipoid”.

The other 837 images constituted the training set. Within the training set, 200 images were used as the validation set. In order to assess the robustness of the training process, a cross-validation-like process was performed. Every network was trained multiple times, each with a different random split for training and validation. After every training process, the obtained performance was measured in a previously unseen testing set, yielding a similar performance for every training process. These results are described in Section 3.

2.4. Convolutional Neural Network Based on the VGG16 Architecture

This work implemented two different CNN architectures for the sake of comparison. Firstly, we implemented a convolutional neural network based on the VGG16 architecture. This architecture was proposed by the Visual Geometry Group at the University of Oxford, who won the ILSVRC (ImageNet) competition in 2014 [58], and has since become widely used owing to its excellent performance. It was described in the seminal work by Simonyan and Zisserman [56] and, since then, has been implemented in a variety of applications [59–61]. A schematic view of the architecture of this CNN is presented in Figure 4.

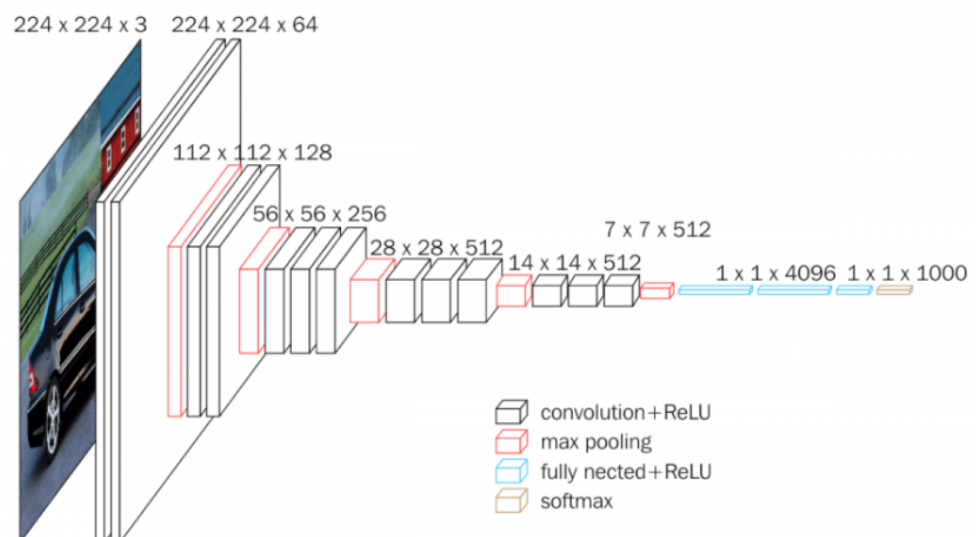


Figure 4. Schematic diagram of the VGG16 CNN architecture. Credit: <https://neurohive.io/en/popular-networks/vgg16/> (accessed on 15 November 2020).

In the original implementation, the input of the first convolutional layer has a fixed size of 224×224 and is of RGB type. The image is processed through an array of 13 successive convolutional layers with max pooling layers between every two or three convolutional layers, as displayed in Figure 4. A total of 3×3 convolutional filters are used in every convolutional layer, with a stride of 1 pixel and a padding of 1 pixel. Spatial pooling is performed in a total of five max-pooling layers. The max pooling is performed through windows of 2×2 pixels with a stride of 2 pixels. After the convolutional layers, there are three dense layers with 4096 nodes each, followed by a final softmax layer. The hidden layers are built with rectified linear unit (ReLU) activation functions.

In this work, the VGG16 network was implemented using the Keras API [62]. Keras is a programming interface for Tensorflow, which is a Google open source library for machine learning applications [63]. Keras has a built-in implementation of the VGG16 network that allows for image sizes different from the original configuration of 224×224 pixels, which allowed us to define a model for an image size of 400×400 . Keras also enables the user to remove the three fully connected layers at the end of the network that the original version of VGG16 has. In this work, these layers were removed, and the output of the convolutional layers was passed through an average-pool-type filter with a size of 2 pixels by 2 pixels. Subsequently, a single fully connected layer of 64 nodes was added

with the ReLU activation function and, finally, a two-node output layer with the softmax activation was added [64]. Note that, although we trained the CNN to distinguish images of lungs infected with COVID-19 from images of lungs with other pathologies, the pursued classifications were still binary: either infected with COVID-19 or not.

In this implementation, the weights that were optimized were those of the fully connected layers, while the pre-trained weights of the convolutional layers remained at their original values [56]. The weights were optimized by using the built-in Adam optimizer [65], for which the learning rate lr was scheduled as $lr(k) = 0.001 / (1 + d \cdot k)$, where k is the iteration number and $d = 0.00002857142$. Finally, the used loss function \mathcal{J} was “binary cross-entropy”, which is defined in Equation (1), where y_i denotes the label of the observation i , \hat{y}_i indicates the predicted value for the observation i , and N represents the number of observations in a given batch. For this architecture, the model was initially trained for 35 epochs with a batch size of eight observations, although at about epoch 25, steady-state values of the loss function were attained.

$$\mathcal{J} = -\frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (1)$$

For the implementation of this CNN, the mean of the tensors corresponding to every image in the training set was subtracted from every tensor in both the training and testing sets.

2.5. Convolutional Neural Network Based on the Xception Architecture

Another architecture assessed in this work is Xception. Xception was developed at Google by F. Chollet. It is based on the concept of “inception modules” and is endowed with 36 convolutional layers for feature extraction [57]. This architecture seems very promising, since it outperformed other precedent networks with a similar number of parameters on large datasets such as ImageNet [58] and another Google internal dataset denoted by JFT.

Like VGG16, this architecture can be also implemented through the Keras framework [66]. In its default implementation, the size of the input images is 299 pixels \times 299 pixels, and it has 3 channels. However, disabling the default top layer of this CNN makes the use of different image sizes possible. We opted to disable the default top layer to keep the same input size of 400 pixels \times 400 pixels (as used in Section 2.4) and replaced the disabled layer with another top fully connected layer of 64 nodes.

In the resulting architecture, the image is processed through an array of 14 blocks of successive convolutional layers. Each block is composed differently, combining *depthwise separable* convolutional layers with ReLU activation functions and 3 \times 3 layers of max pooling. As with the previous architecture (Section 2.4), the output of the convolutional layers was passed through an average-pool-type filter with a size of 2 pixels by 2 pixels and the last fully connected layer of 64 nodes with the ReLU activation function. Likewise, the output layer had two nodes with softmax activation.

The weights that were optimized were those of the fully connected layers, while the weights of the convolutional layers were set as those pre-trained on the ImageNet set. The weights were optimized using the Adam optimizer [65] with the learning rate lr and the loss function \mathcal{J} , as specified in Section 2.4. For this architecture, the model was trained for 25 epochs, with a batch size of 30 observations. For the implementation of this CNN, the intensity of every pixel in the input images was rescaled to lie in the interval [0, 1].

3. Results and Discussion

Both CNNs were trained and tested on the same sets. As previously described, the training process was run multiple times. Each time, the total training set (834 images) was randomly split into 200 images for validation and 634 images for training. After every training process, the performance of the obtained CNN was evaluated using the testing set. The obtained results were very similar for all the experiments, providing an indication of

the robustness of the training process. Tables 2 and 3 display the obtained performance for each of the conducted experiments for each CNN architecture.

Table 2. Performance obtained across the 10 experiments for the VGG16-based CNN.

	Sensitivity	Specificity	F_1
Experiment 1	95.4%	97.6%	95.4%
Experiment 2	96.9%	97.6%	96.1%
Experiment 3	94.2%	100%	97.0%
Experiment 4	96.7%	99.2%	97.7%
Experiment 5	97.0%	100%	98.5%
Experiment 6	96.9%	98.4%	96.9%
Experiment 7	96.9%	97.6%	96.1%
Experiment 8	96.9%	98.4%	96.9%
Experiment 9	92.9%	100%	96.3%
Experiment 10	96.9%	97.6%	96.1%

Table 3. Performance obtained across the 10 experiments for the Xception-based CNN.

	Sensitivity	Specificity	F_1
Experiment 1	100%	97.6%	97.7%
Experiment 2	95.6%	100%	97.7%
Experiment 3	97.0%	99.2%	97.7%
Experiment 4	97.0%	100%	98.5%
Experiment 5	95.6%	100%	97.7%
Experiment 6	95.4%	97.6%	95.4%
Experiment 7	97.0%	99.2%	97.7%
Experiment 8	96.6%	93.9%	91.9%
Experiment 9	97.0%	100%	98.5%
Experiment 10	96.9%	98.4%	96.9

The following paragraphs elaborate on the results of Experiment 1 for both CNNs. Figure 5 illustrates the progress regarding the prediction capability of the VGG16 CNN as the training advanced. It can be seen that, after 15 epochs of training, the loss function attained a steady value of $\mathcal{J} \sim 0.1$. Likewise, the accuracy computed over the testing set was around 94%.

Examining the distributions of the VGG16 CNN predictions over the original labels reveals the following for one of the aforementioned ten experiments. There were originally 65 cases labeled as “COVID-19”, of which 62 (95.4%) were predicted as such and 3 were wrongly identified as false negatives. On the other hand, out of the 63 cases labeled as “Pneumonia”, only 1 (1.6%) was wrongly predicted as COVID-19, whereas the other 62 were properly classified as no COVID-19. With respect to the 63 test cases labeled as “Normal”, 2 (3.2%) were wrongly identified as COVID-19, while 61 were properly determined as no COVID-19. For this analysis, we grouped all the other 11 cases as “Others”, which should be properly predicted as no COVID-19. The idea of incorporating the latter images grouped as “Others”, as well as pneumonia images, into the testing set was to explore the performance of the classifier with images of lungs affected by diseases other than COVID-19. However, the number of pulmonary images in the group “Others” was considerably smaller than the number of images of lungs infected with COVID-19 or pneumonia; hence, they were grouped together.

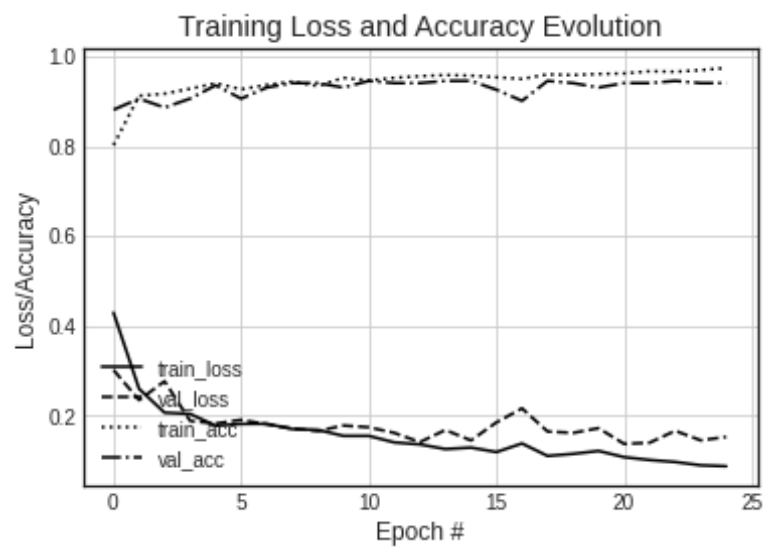


Figure 5. Evolution of the training of VGG16 CNN.

Out of the 11 testing set images grouped as “Others”, 6 were wrongly predicted as COVID-19 and the other 5 were properly classified as no COVID-19. Since there were only 26 of these images in the training set, perhaps the classifier could not properly learn to distinguish them from cases of COVID-19. This would support our first hypothesis that the number of cases of these types in the training set was not sufficient for appropriate training to develop the ability to classify them as no COVID-19. Figure 6 illustrates the distribution of the results, which is also summarized in the confusion matrices in Table 4 (including cases labeled as “Others”) and Table 5 (excluding cases labeled as “Others”). Note that the confusion matrices present only two categories, where the positive category refers to cases of COVID-19 infection while the negative category refers to all cases that were not COVID-19-positive.

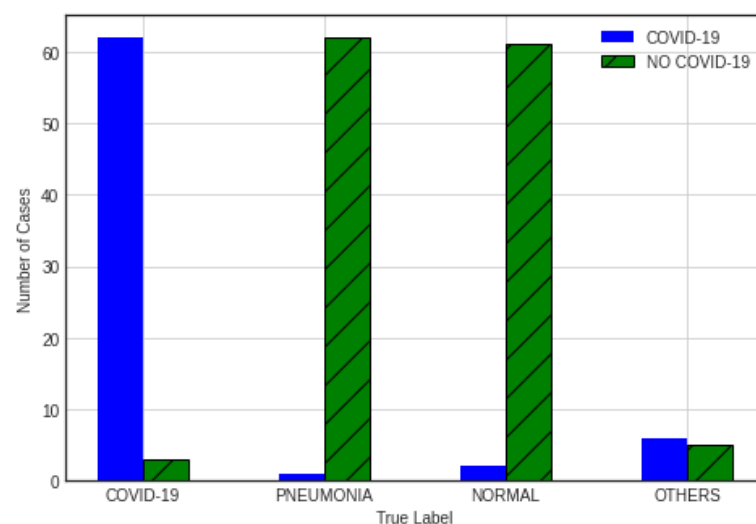


Figure 6. Distribution of the classification results.

Table 4. Confusion matrix for the entire testing set: CNN VGG16.

	Pred. Positive	Pred. Negative
True Positive	62	3
True Negative	9	128

Table 5. Confusion matrix for the testing set without the cases labeled as “Others”: CNN VGG16.

	Pred. Positive	Pred. Negative
True Positive	62	3
True Negative	3	123

Based on the aforementioned numbers and considering only the groups “COVID-19”, “Pneumonia”, or “Normal” for the detection of COVID-19, the following statistics were obtained:

- Sensitivity (Recall) $\triangleq \frac{TP}{TP+FN} = \frac{62}{65} = 95.4\%$;
- Specificity $\triangleq \frac{TN}{TN+FP} = \frac{123}{126} = 97.6\%$;
- $F_1 \triangleq \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = 95.4\%$.

where TP, TN, FP, and FN denote the number of true positive, true negative, false positive, and false negative cases, respectively. Recall that positive and negative refer to COVID-19-positive (infected) and COVID-19-negative (uninfected) cases.

The Xception CNN performed slightly better than the VGG16 architecture. The respective confusion matrices are displayed in Tables 4 and 6. Comparing Tables 4 and 6, it is observed that the Xception-based CNN did not produce any false negative case and, consequently, every positive case was classified as such. The VGG16-based CNN yielded three false negative cases. False negative cases indicate people who are classified as uninfected when they are actually infected; keeping this number as low as possible is thus crucial to avoid misleading treatments. In this sense, considering the tests conducted in this work, Xception looks more promising. This is reflected in the sensitivity statistic, which jumped from 95.4% for VGG16 to 100% for Xception. Regarding the classification of the actual negative cases, the proportion of tests correctly classified was very similar in both cases.

Table 7 yields the following statistics for the Xception CNN (disregarding cases that were not “COVID-19”, “Pneumonia”, or “Normal”):

- Sensitivity (Recall) $= \frac{65}{65} = 100\%$;
- Specificity $= \frac{123}{126} = 97.6\%$;
- $F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = 97.7\%$.

The progress attained during the training process is depicted in Figure 7.

Table 6. Confusion matrix for the entire testing set: CNN Xception.

	Pred. Positive	Pred. Negative
True Positive	65	0
True Negative	10	127

Table 7. Confusion matrix for the testing set without the cases labeled as “Others”: CNN Xception.

	Pred. Positive	Pred. Negative
True Positive	65	0
True Negative	3	123

Regarding the architectures of the Xception and VGG16 networks, it is worth mentioning that they differ in terms of the number of parameters each CNN has. The Xception-based CNN consisted of a total of 25,580,266 parameters, 4,718,786 of which were trainable. By contrast, the VGG16-based CNN had 15,894,530 parameters, only 1,179,842 of which were trainable. Roughly speaking, this provides an indication that the Xception architecture implemented herein had a better capacity to learn to classify the images than the VGG16 version.

Duran-Lopez et al. [12] developed a deep-learning-based system called COVID-XNet, which preprocesses chest X-ray images and then classifies them as normal or infected with

COVID-19. To train their systems, they used a total of 6926 images, with 5541 images for training and 1385 images for validation. The COVID-XNet system achieved a 92.53% sensitivity and 96.33% specificity. These values are lower than those obtained by our model, given that our model was trained using a much smaller training set.

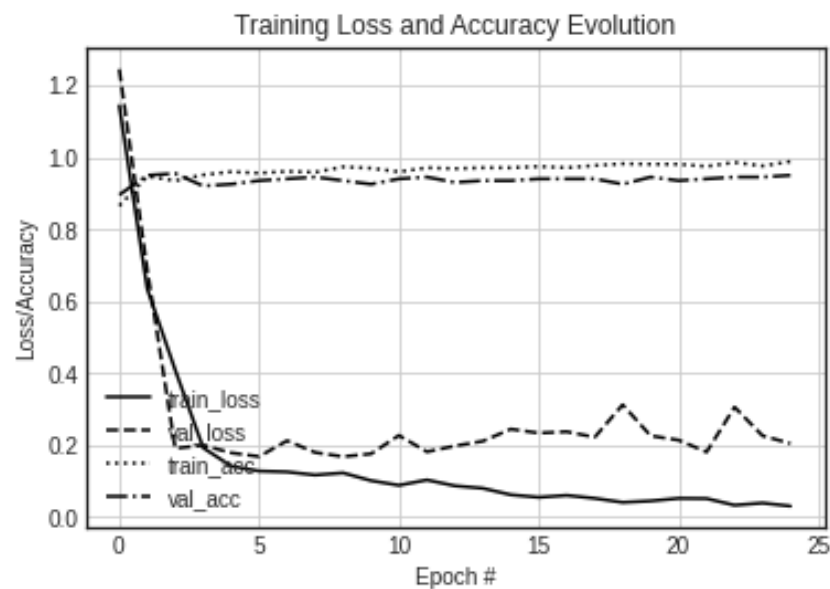


Figure 7. Evolution of the training of the Xception CNN.

Alam et al. [14] extracted HOG and CNN features from X-ray images and then fused them to train a VGGNet classifier for COVID-19 detection. They used a total of 5090 images, with 4072 images for training and 1018 images for testing. The proposed model achieved a sensitivity of 93.65% and a specificity of 95.7%. Again, these values are lower than those obtained by our model, given that our model was trained using a much smaller training set. Additionally, it is worth mentioning that fusing more features will generally result in better performance, as shown in our previous work [67]. This conclusion somewhat limits the contribution of the work proposed by Alam et al. [14].

In [44], the authors compared the performance of three CNN models: ResNet50, InceptionV3, and Inception ResNetV2. ResNet50 clearly outperformed the other two models. It attained a mean value for sensitivity of 96%, a mean value for specificity of 100%, and a mean value for F_1 of 98%. These values are very similar to those obtained in this study. However, in this work, we used a total of 1037 images, with 200 images for the testing set, while they used a total of 100 images for both the training and testing sets. In [45], they pursued a four-category classification (Bacterial Pneumonia, Viral Pneumonia (other than COVID-19), Normal (uninfected), and COVID-19), but for the sake of this analysis, we mapped their results onto either COVID-19-positive or COVID-19-negative cases (including every case that was not COVID-19). Again, the obtained results are very similar to those obtained in this paper. However, those results were obtained on a dataset including 68 COVID-19 cases, with only eight of them in the testing set. Performing a similar comparison with [47] and grouping their classification results into COVID-19-positive and negative, a sensitivity of 91%, specificity of 99.5%, and $F_1 = 94.8%$ would be obtained.

In [48], three different architectures of CNNs were analyzed for the detection of COVID-19, with the performance being evaluated in terms of accuracy, which is defined as

$$\text{Accuracy} \triangleq \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

The accuracy values achieved by each architecture are 90%, 94.3%, and 96.8%. In [50–52], the obtained accuracy values were 97.78%, 95.5%, and 97%, respectively. All of these values are comparable to the ones presented in this work. The obtained accuracy levels herein are 97% for the VGG-16-based architecture and 98.4% for the Xception-based architecture.

Bourouis et al. [13] proposed a Bayesian classifier framework to classify chest X-ray images as infected or normal. They applied their proposed framework on three different datasets and achieved the highest accuracy (90.33%) using the largest dataset, called CXR-Pneumonia. This dataset contains 5856 images, of which 5216 images are used for training, 16 images for validation, and 624 images for testing. Their obtained accuracy is much lower than the accuracy we achieved using the Xception-based architecture (98.4%). Note that our model was trained using a much smaller training set.

The results presented in this paper, as well as those presented in the papers cited, suggest that, with sufficient training, it is actually feasible to develop CNN-based classifiers with which to categorize PA and AP X-ray lung images as infected with COVID-19 or not infected. This is one of the most relevant conclusions that should be drawn from this work. Note that the obtained classifiers depend upon the labeled images used for training. Images of people infected with COVID-19 in the very early stages of the disease might have been labeled as negative despite actually being positive. Cases such as these can therefore be misclassified as negative. Hence, it is important to point out that the high potential of these methods resides in their ability to detect cases where the condition of the lungs affected by the disease is such that the pulmonary images exhibit visual characteristics typical for COVID-19-positive cases.

Finally, it is worth mentioning that, once the model was trained, the time it took to classify ten images in a row was measured as approximately 270 milliseconds on a laptop with 16 GB of RAM and an Intel Core i7 vPro 8th Generation CPU. Hence, considering the infrastructure required for running these types of models, this shows that our method is feasible and affordable for any medical institution. Our annotated codes can be found at [68].

Explainability of the Classification Models

With the models derived and exhibiting good classification metrics, we intended to visualize what triggers a trained model to classify a given image as COVID-19-positive or negative. This question is in the context of *explainability and interpretability* analysis. Trained CNNs (and other deep learning models in general) are frequently perceived as “black box” operators that map inputs (images) onto predicted labels. However, practitioners often do not exactly understand what specific features in the image induced the model to classify the image under a given category.

Identifying these features constitutes an important step in the debugging process, which should ensure that the model has learned to look at the pixels (features) that convey the appropriate information to classify the image. It might occur that, in the training process, a model trained to associate certain patterns with the labels can actually mislead the classification process. For instance, one can train a model to distinguish images with cats labeled as “CATS” from images with dogs labeled as “DOGS”. If the images with cats were all obtained outside on sunny days and the images with dogs were all taken indoors, the model could wrongly learn that blue skies should be associated with the label “CATS” and that the lack of blue skies should be linked to the label “DOGS”. If the testing set was obtained from the same set of images—i.e., the testing images of cats also have blue skies and the images of dogs are indoors—the model will likely correctly classify the cats as “CATS”, but only because of the blue skies rather than because of the cats shown in the images. Likewise, it will classify images with dogs as “DOGS” due to the features pertaining to the indoor environment. However, in this case, the same model would probably classify an image of a dog walking in a park on a sunny day as “CATS”.

Therefore, it is important to confirm that the features (pixels) that a given model actually uses to classify images are appropriate. Several approaches have been developed for this purpose [69–71]. In this work, we applied an approach called GradientExplainer to

compute the so-called *Shapley Additive exPlanations* (SHAP) scores [71–75]. Given an image with its associated prediction, for each feature in the image, a SHAP score constitutes a measurement of the influence of having a certain value in that feature on the obtained prediction in comparison to the prediction that would be made if that feature took some baseline value. Features with high positive SHAP are features that increase the probability of making such a prediction, whereas negative SHAP scores reduce the probability of classifying the image as such. In other words, the areas of an image with a concentration of high SHAP values constitute areas that convey relevant information to the classification model. Using [74], we implemented this technique to find the most relevant pixels in some of the classified images.

Figures 8 and 9 display four original pulmonary radiographs—two of patients diagnosed with COVID-19 and two of uninfected people. Each of these images was properly predicted with the derived classifier. For each of these images, the corresponding SHAP values were computed, and the results are illustrated in Figures 10–13.

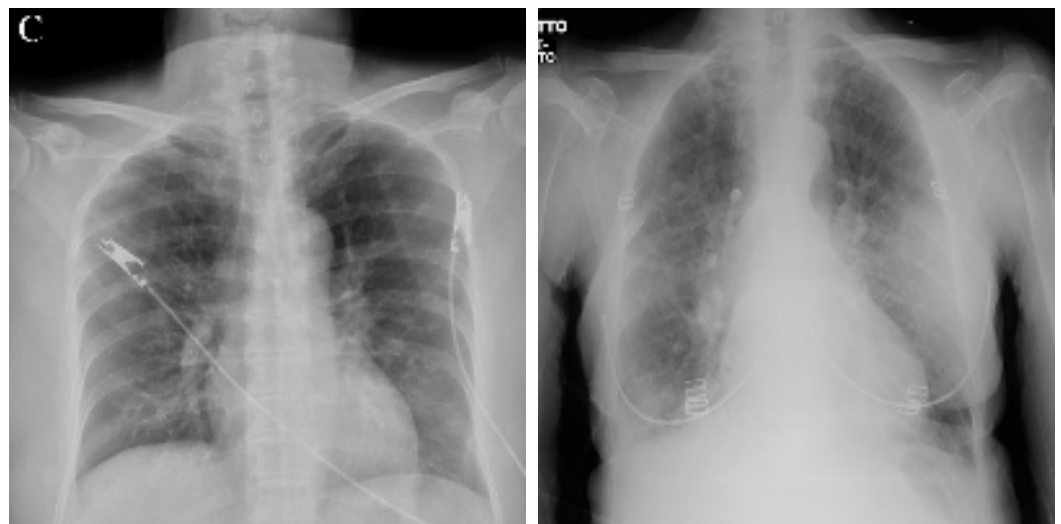


Figure 8. Examples of radiography images of patients infected with COVID-19 (positive examples).

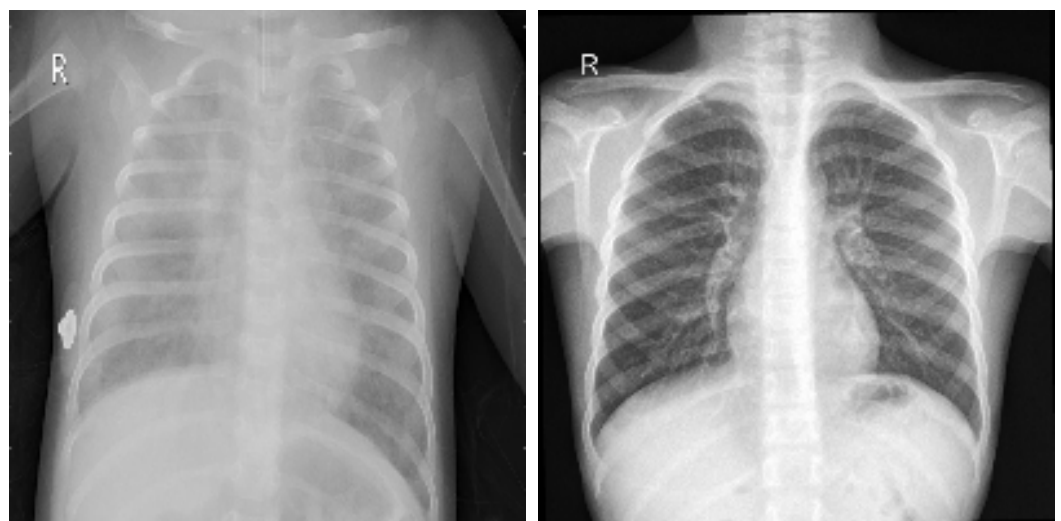


Figure 9. Examples of radiography images of patients not infected with COVID-19 (negative examples).

The top plots in Figures 10–13 depict a representation of the input to layer number 10 of the CNN. The mid-positions in the illustrations of Figures 10 and 11 reveal clouds of magenta points with high SHAP values that indicate the locations of the most relevant pixels for classifying images as COVID-19-positive. Accordingly, if these images were

to be classified as COVID-19-negative or NORMAL, these areas would attain very low SHAP values, which is illustrated in the bottom images in Figures 10 and 11. Note the side color bar that indicates the mapping from colors to SHAP values. According to the color bar marked as “SHAP Value”, the more intense the magenta color is, the higher the SHAP scores are and the influence that those pixels had in the classification. Hence, clouds with a high density of magenta points reveal zones in the image that convey significant information used by the classifier.

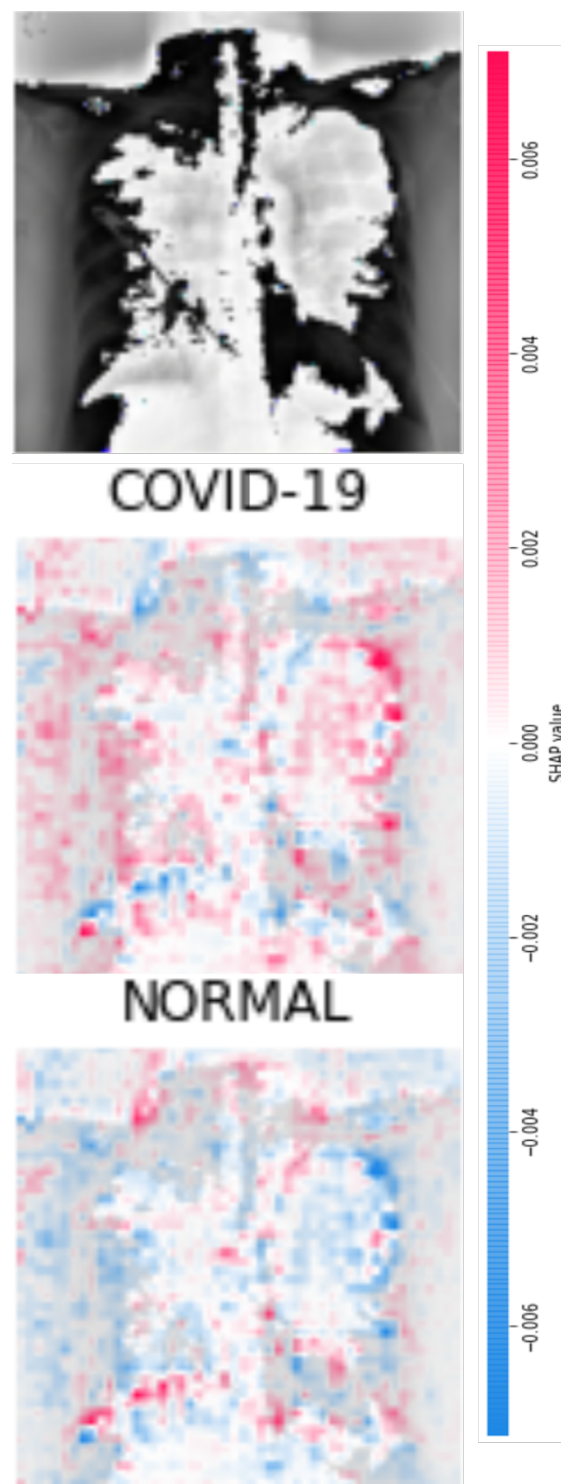


Figure 10. SHAP scores for the image displayed in Figure 8 [left].

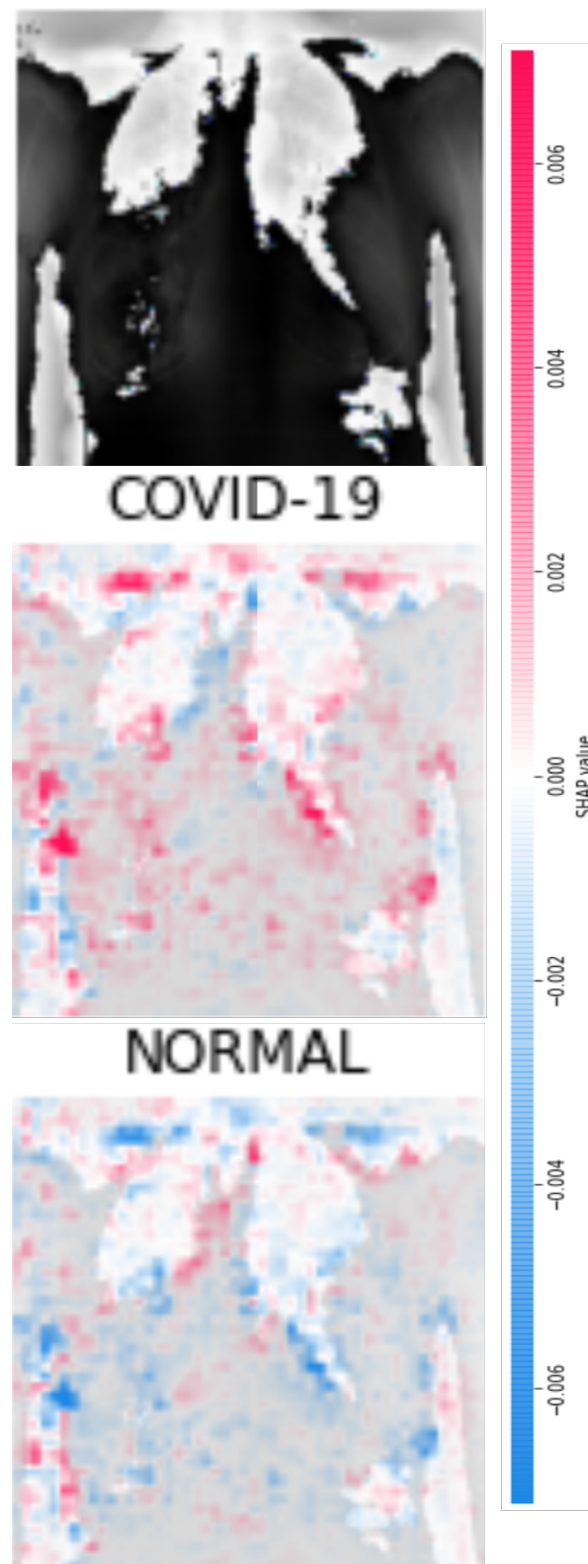


Figure 11. SHAP scores for the image displayed in Figure 8 [right].

On the other hand, Figures 12 and 13 show the SHAP values computed for the cases of Figure 9, which were classified as NORMAL. In these cases, the mid plots show the zones of magenta pixels that convey relevant information for classifying the images as NORMAL. Accordingly, if these images were classified as images of lungs infected with COVID-19, these same zones would attain very low SHAP values, as illustrated in the bottom plots of Figures 12 and 13.

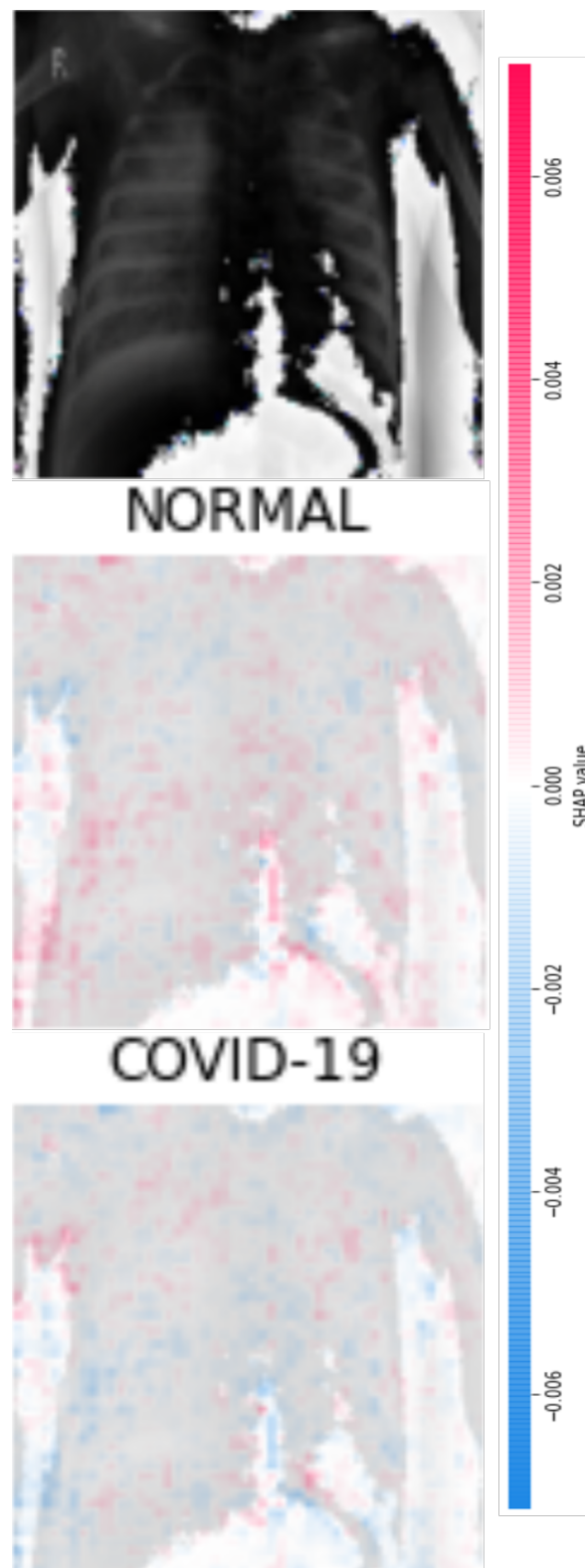


Figure 12. SHAP scores for the image displayed in Figure 9 [left].

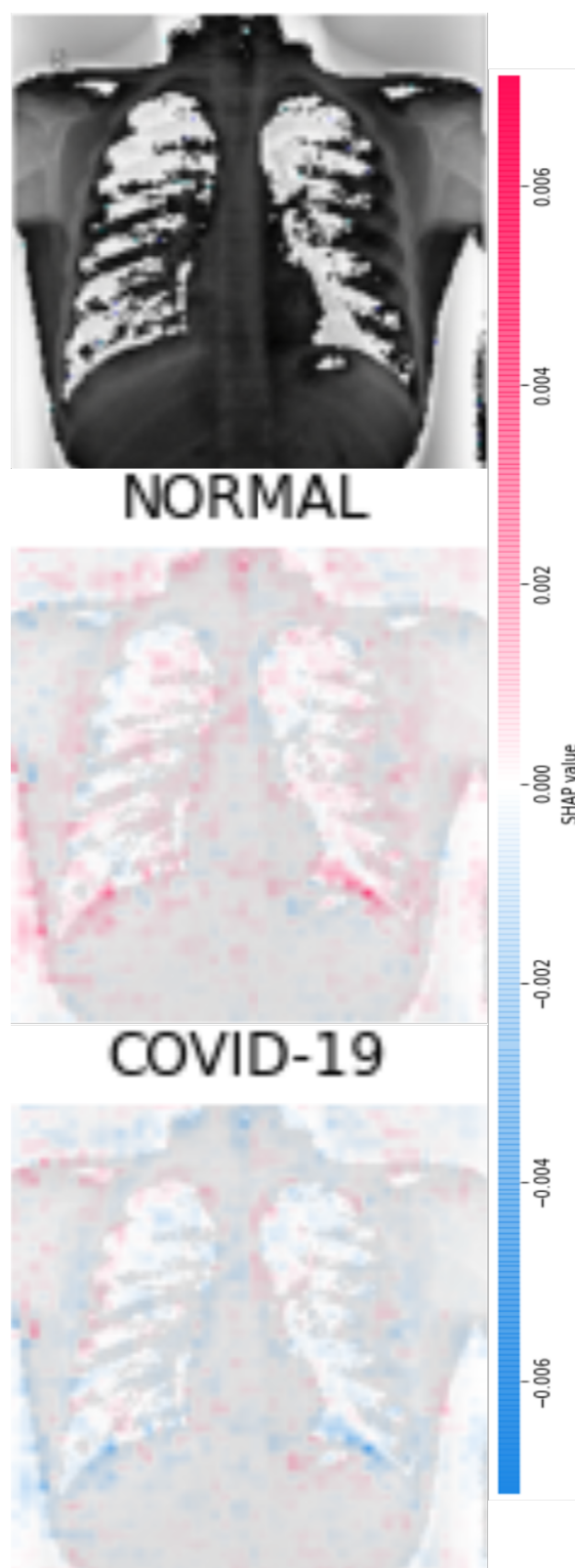


Figure 13. SHAP scores for the image displayed in Figure 9 [right].

By observing the middle plots of Figures 10–13, many colored points can be noticed in various sectors of the images. For instance, in Figure 10, clusters of magenta points (higher SHAP values) are found in the upper half of the right lung (as seen by the reader) and in the central areas of the left lung. Likewise, in Figure 13, there are high concentrations of relevant pixels in the bottom parts of the lungs. In general, we identified that clusters of

pixels with relevant information were found within the areas of the lungs, as should be expected for the detection of pulmonary diseases from lung images.

4. Conclusions

This work has shown that building a deep learning classifier that can distinguish pulmonary radiography images of patients exhibiting COVID-19 infection from radiography images of patients not infected with COVID-19 is highly feasible. Moreover, patients with other conditions such as non-COVID-19 pneumonia can be properly identified if sufficient cases are used in the training.

Compared to work with similar aims, the CNN architectures analyzed in this work exhibited very high performance metrics. Our work, along with many other studies (using similar metrics), has clearly proven that the development of CNN-based classifiers for production stages appears to be highly feasible.

These classifiers are not intended to act as a substitute for formal diagnosis by physicians. They are meant to assist medical personnel in cases where viral tests are scarce and the number of suspected cases is so high that the visual inspection of every radiograph might generate a bottleneck in the process.

This research constitutes the initial steps in the development of classifiers that could be released to production stages. It is needless to say that further training and testing with images from many more sources are required.

Funding: This research was funded by the University Agency for Graduate Studies and Scientific Research and the Deanship of Scientific Research at Qassim University.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The author would like to thank the editor and anonymous reviewers for spending their valuable time reviewing and strengthening his paper. In addition, the author would like to express his deepest appreciation to A. Lynn Abbott and Kadir Aşkın Peker for their endless valuable suggestions on how to improve his paper. Finally, special thanks go to the University Agency for Graduate Studies and Scientific Research and the Deanship of Scientific Research at Qassim University for supporting this research.

Conflicts of Interest: The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. World Health Organization (WHO). Coronavirus disease (COVID-19) Pandemic. 2019. Available online: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019> (accessed on 15 November 2020).
2. Suman, R.; Javaid, M.; Haleem, A.; Vaishya, R.; Bahl, S.; Nandan, D. Sustainability of coronavirus on different surfaces. *J. Clin. Exp. Hepatol.* **2020**, *10*, 386–390. [[CrossRef](#)] [[PubMed](#)]
3. Wang, J.; Tang, K.; Feng, K.; Lin, X.; Lv, W.; Chen, K.; Wang, F. High temperature and high humidity reduce the transmission of COVID-19. *arXiv* **2020**, arXiv:2003.05003.
4. Sajadi, M.M.; Habibzadeh, P.; Vintzileos, A.; Shokouhi, S.; Miralles-Wilhelm, F.; Amoroso, A. Temperature, humidity, and latitude analysis to estimate potential spread and seasonality of coronavirus disease 2019 (COVID-19) *JAMA Netw.* **2020**, *3*, e2011834. [[CrossRef](#)] [[PubMed](#)]
5. Bai, Y.; Yao, L.; Wei, T.; Tian, F.; Jin, D.Y.; Chen, L.; Wang, M. Presumed asymptomatic carrier transmission of COVID-19. *J. Am. Med. Assoc. (JAMA)* **2020**, *323*, 1406–1407. [[CrossRef](#)]
6. Zheng, Y.Y.; Ma, Y.T.; Zhang, J.Y.; Xie, X. COVID-19 and the cardiovascular system. *Nat. Rev. Cardiol.* **2020**, *17*, 259–260. [[CrossRef](#)]
7. Colson, P.; Rolain, J.M.; Lagier, J.C.; Brouqui, P.; Raoult, D. Chloroquine and hydroxychloroquine as available weapons to fight COVID-19. *Int. J. Antimicrob. Agents* **2020**, *55*, 105932. [[CrossRef](#)]
8. Kakodkar, P.; Kaka, N.; Baig, M.N. A comprehensive literature review on the clinical presentation, and management of the pandemic coronavirus disease 2019 (COVID-19). *Cureus* **2020**, *12*, e7560. [[CrossRef](#)]
9. Batagelj, B.; Peer, P.; Štruc, V.; Dobrišek, S. How to correctly detect face-masks for COVID-19 from visual information?. *Appl. Sci.* **2021**, *11*, 2070. [[CrossRef](#)]

10. Qin, B.; Li, D. Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. *Sensors* **2020**, *20*, 5236. [CrossRef]
11. Taylor, W.; Abbasi, Q.H.; Dashtipour, K.; Ansari, S.; Shah, S.A.; Khalid, A.; Imran, M.A. A review of the state of the art in non-contact sensing for COVID-19. *Sensors* **2020**, *20*, 5665. [CrossRef]
12. Duran-Lopez, L.; Dominguez-Morales, J.P.; Corral-Jaime, J.; Vicente-Diaz, S.; Linares-Barranco, A. COVID-XNet: A custom deep learning system to diagnose and locate COVID-19 in chest X-ray images. *Appl. Sci.* **2020**, *10*, 5683. [CrossRef]
13. Bourouis, S.; Alharbi, A.; Bouguila, N. Bayesian learning of shifted-scaled Dirichlet mixture models and its application to early COVID-19 detection in chest X-ray images. *J. Imaging* **2021**, *7*, 7. [CrossRef]
14. Alam, N.A.; Ahsan, M.; Based, M.A.; Haider, J.; Kowalski, M. COVID-19 detection from chest X-ray images using feature fusion and deep learning. *Sensors* **2021**, *21*, 1480. [CrossRef] [PubMed]
15. Mazzilli, A.; Fiorino, C.; Loria, A.; Mori, M.; Esposito, P.G.; Palumbo, D.; de Cobelli, F.; del Vecchio, A. An automatic approach for individual HU-based characterization of lungs in COVID-19 patients. *Appl. Sci.* **2021**, *11*, 1238. [CrossRef]
16. Alshazly, H.; Linse, C.; Barth, E.; Martinetz, T. Explainable COVID-19 detection using chest CT scans and deep learning. *Sensors* **2021**, *21*, 455. [CrossRef] [PubMed]
17. Zisad, S.N.; Hossain, M.S.; Hossain, M.S.; Andersson, K. An integrated neural network and SEIR model to predict COVID-19. *Algorithms* **2021**, *14*, 94. [CrossRef]
18. AbdelAziz, A.M.; Alarabi, L.; Basalamah, S.; Hendawi, A. A multi-objective optimization method for hospital admission problem—A case study on COVID-19 patients. *Algorithms* **2021**, *14*, 38. [CrossRef]
19. Wang, W.; Xu, Y.; Gao, R.; Lu, R.; Han, K.; Wu, G.; Tan, W. Detection of SARS-CoV-2 in different types of clinical specimens. *J. Am. Med. Assoc. (JAMA)* **2020**, *323*, 1843–1844. [CrossRef]
20. Corman, V.M.; Landt, O.; Kaiser, M.; Molenkamp, R.; Meijer, A.; Chu, D.K.; Bleicker, T.; Brünink, S.; Schneider, J.; Schmidt, M.L.; et al. Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR. *Eurosurveillance* **2020**, *25*, 2000045. [CrossRef] [PubMed]
21. Lloyd-Sherlock, P.; Ebrahim, S.; Geffen, L.; McKee, M. Bearing the brunt of COVID-19: Older people in low and middle income countries. *Br. Med. J. (BMJ)* **2020**, *368*, m1052. [CrossRef]
22. Salehi, S.; Abedi, A.; Balakrishnan, S.; Gholamrezaezhad, A. Coronavirus disease 2019 (COVID-19): A systematic review of imaging findings in 919 patients. *Am. J. Roentgenol. (AJR)* **2020**, *215*, 87–93. [CrossRef] [PubMed]
23. Chung, M.; Bernheim, A.; Mei, X.; Zhang, N.; Huang, M.; Zeng, X.; Cui, J.; Xu, W.; Yang, Y.; Fayad, Z.A.; et al. CT imaging features of 2019 novel coronavirus (2019-nCoV). *Radiology* **2020**, *295*, 202–207. [CrossRef] [PubMed]
24. Kanne, J.P. Chest CT findings in 2019 novel coronavirus (2019-nCoV) infections from Wuhan, China: Key points for the radiologist. *Radiology* **2020**, *295*, 16–17. [CrossRef] [PubMed]
25. Ai, T.; Yang, Z.; Hou, H.; Zhan, C.; Chen, C.; Lv, W.; Tao, Q.; Sun, Z.; Xia, L. Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases. *Radiology* **2020**, *296*, E32–E40. [CrossRef] [PubMed]
26. GitHub. Figure 1 COVID-19 Chest X-ray Dataset Initiative. 2020. Available online: <https://github.com/agchung/Figure1-COVID-chestxray-dataset> (accessed on 15 November 2020).
27. Yang, X.; He, X.; Zhao, J.; Zhang, Y.; Zhang, S.; Xie, P. COVID-CT-Dataset: A CT scan dataset about COVID-19. *arXiv* **2020**, arXiv:2003.13865.
28. Cohen, J.P.; Morrison, P.; Dao, L. COVID-19 image data collection. *arXiv* **2020**, arXiv:2003.11597.
29. Cohen, J.P.; Morrison, P.; Dao, L.; Roth, K.; Duong, T.Q.; Ghassemi, M. COVID-19 image data collection: Prospective predictions are the future. *arXiv* **2020**, arXiv:2006.11988.
30. Kalkreuth, R.; Kaufmann, P. COVID-19: A survey on public medical imaging data resources. *arXiv* **2020**, arXiv:2004.04569.
31. Shi, F.; Wang, J.; Shi, J.; Wu, Z.; Wang, Q.; Tang, Z.; He, K.; Shi, Y.; Shen, D. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for COVID-19. *IEEE Rev. Biomed. Eng.* **2020**, *14*, 4–15. [CrossRef]
32. Guo, Q.; Li, M.; Wang, C.; Wang, P.; Fang, Z.; Tan, J.; Wu, S.; Xiao, Y.; Zhu, H. Host and infectivity prediction of Wuhan 2019 novel coronavirus using deep learning algorithm. *bioRxiv* **2020**. [CrossRef]
33. González-Crespo, R.; Herrera-Viedma, E.; Dey, N.; Fong, S.J.; Li, G. Finding an accurate early forecasting model from small dataset: A case of 2019-nCoV novel coronavirus outbreak. *Int. J. Interact. Multimed. Artif. Intell. (IJIMAI)* **2020**, *6*, 132–140. [CrossRef]
34. Xu, X.; Jiang, X.; Ma, C.; Du, P.; Li, X.; Lv, S.; Yu, L.; Ni, Q.; Chen, Y.; Su, J.; et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering* **2020**, *6*, 1122–1129. [CrossRef]
35. Chen, J.; Wu, L.; Zhang, J.; Zhang, L.; Gong, D.; Zhao, Y.; Chen, Q.; Huang, S.; Yang, M.; Yang, X.; et al. Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography: A prospective study. *Sci. Rep.* **2020**, *10*, 19196. [CrossRef]
36. Gozes, O.; Frid-Adar, M.; Greenspan, H.; Browning, P.D.; Zhang, H.; Ji, W.; Bernheim, A.; Siegel, E. Rapid AI development cycle for the coronavirus (COVID-19) pandemic: Initial results for automated detection & patient monitoring using deep learning CT image analysis. *arXiv* **2020**, arXiv:2003.05037.
37. Khan, S.; Rahmani, H.; Shah, S.A.A.; Bennamoun, M. *A Guide to Convolutional Neural Networks for Computer Vision*, 1st ed.; Morgan & Claypool: San Rafael, CA, USA, 2018.

38. Hongtao, L.; Qinchuan, Z. Applications of deep convolutional neural network in computer vision. *J. Data Acquis. Process.* **2016**, *31*, 1–17. [[CrossRef](#)]
39. Lobov, S.A.; Mikhaylov, A.N.; Shamshin, M.; Makarov, V.A.; Kazantsev, V.B. Spatial properties of STDP in a self-learning spiking neural network enable controlling a mobile robot. *Front. Neurosci.* **2020**, *14*, 88. [[CrossRef](#)]
40. Yang, S.; Gao, T.; Wang, J.; Deng, B.; Lansdell, B.; Linares-Barranco, B. Efficient spike-driven learning with dendritic event-based processing. *Front. Neurosci.* **2021**, *15*, 97. [[CrossRef](#)]
41. Yang, S.; Deng, B.; Wang, J.; Li, H.; Lu, M.; Che, Y.; Wei, X.; Loparo, K.A. Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 148–162. [[CrossRef](#)] [[PubMed](#)]
42. Yang, S.; Wang, J.; Deng, B.; Liu, C.; Li, H.; Fietkiewicz, C.; Loparo, K.A. Real-time neuromorphic system for large-scale conductance-based spiking neural networks. *IEEE Trans. Cybern.* **2019**, *49*, 2490–2503. [[CrossRef](#)] [[PubMed](#)]
43. GitHub. COVID-19 Chest X-ray Model. 2020. Available online: <https://github.com/aildnon/covid-cxr> (accessed on 15 November 2020).
44. Narin, A.; Kaya, C.; Pamuk, Z. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *arXiv* **2020**, arXiv:2003.10849.
45. Farooq, M.; Hafeez, A. COVID-ResNet: A deep learning framework for screening of COVID19 from radiographs. *arXiv* **2020**, arXiv:2003.14395.
46. Afshar, P.; Heidarian, S.; Naderkhani, F.; Oikonomou, A.; Plataniotis, K.N.; Mohammadi, A. COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images. *arXiv* **2020**, arXiv:2004.02696.
47. Wang, L.; Wong, A. COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *arXiv* **2020**, arXiv:2003.09871.
48. Zebin, T.; Rezvy, S. COVID-19 detection and disease progression visualization: Deep learning on chest X-rays for classification and coarse localization. *Appl. Intell.* **2021**, *51*, 1010–1021. [[CrossRef](#)]
49. Turkoglu, M. COVIDetectionNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble. *Appl. Intell.* **2021**, *51*, 1213–1226. [[CrossRef](#)]
50. Goel, T.; Murugan, R.; Mirjalili, S.; Chakrabartty, D.K. OptCoNet: An optimized convolutional neural network for an automatic diagnosis of COVID-19. *Appl. Intell.* **2021**, *51*, 1351–1366. [[CrossRef](#)]
51. Misra, S.; Jeon, S.; Lee, S.; Managuli, R.; Jang, I.-S.; Kim, C. Multi-channel transfer learning of chest X-ray images for screening of COVID-19. *Electronics* **2020**, *9*, 1388. [[CrossRef](#)]
52. Jain, R.; Gupta, M.; Taneja, S.; Hemanth, D.J. Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Appl. Intell.* **2021**, *51*, 1690–1700. [[CrossRef](#)]
53. GitHub. Actualmed COVID-19 Chest X-ray Dataset Initiative. 2020. Available online: <https://github.com/agchung/Actualmed-COVID-chestxray-dataset> (accessed on 15 November 2020).
54. Kaggle. Covid-19 Detection from Lung X-rays. 2020. Available online: <https://www.kaggle.com/eswarchandt/covid-19-detection-from-lung-x-rays/> (accessed on 15 November 2020).
55. Kaggle. COVID-19 Radiography Database (Winner of the COVID-19 Dataset Award). 2020. Available online: <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> (accessed on 15 November 2020).
56. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
57. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2017.
58. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis. (IJCV)* **2015**, *115*, 211–252. [[CrossRef](#)]
59. Tindall, L.; Luong, C.; Saad, A. Plankton Classification Using VGG16 Network. 2017. Available online: <https://pdfs.semanticscholar.org/7cb1/a0d0d30b4567b771ad7ae265ab0e935bc41c.pdf> (accessed on 15 November 2020).
60. Ashraf, K.; Wu, B.; Iandola, F.; Moskewicz, M.; Keutzer, K. Shallow networks for high-accuracy road object-detection. *arXiv* **2016**, arXiv:1606.01561.
61. Antony, J.; McGuinness, K.; O’Connor, N.; Moran, K. Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks. In Proceedings of the International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; IEEE: New York, NY, USA, 2016.
62. Keras. VGG16 and VGG19. 2015. Available online: <https://keras.io/api/applications/vgg/#vgg16-function> (accessed on 15 November 2020).
63. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for large-scale machine learning. In Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, 2–4 November 2016; USENIX Association: Berkeley, CA, USA, 2016.
64. PyImageSearch. Available online: <https://www.pyimagesearch.com> (accessed on 15 November 2020).
65. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
66. Keras. Xception. 2017. Available online: <https://keras.io/api/applications/xception/> (accessed on 15 November 2020).

67. Alorf, A.; Abbott, A.L.; Peker, K.A. Facial attribute classification: A comprehensive study and a novel mid-level fusion classifier. In Proceedings of the International Conference on Biometrics Theory, Applications and Systems (BTAS), Tampa, FL, USA, 23–26 September 2019; IEEE: New York, NY, USA, 2019.
68. Abdulaziz Alorf. Codes for COVID-19 Detection. 2020. Available online: https://drive.google.com/drive/folders/1Wol6DQABuEOTBzTK-spBEu6z_QO6e-mE?usp=sharing (accessed on 10 June 2021).
69. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv* **2013**, arXiv:1312.6034.
70. Selvaraju, R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: New York, NY, USA, 2017.
71. Lundberg, S.; Lee, S. A unified approach to interpreting model predictions. In Proceedings of the International Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017.
72. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [[CrossRef](#)] [[PubMed](#)]
73. Lundberg, S.M.; Nair, B.; Vavilala, M.S.; Horibe, M.; Eisses, M.J.; Adams, T.; Liston, D.E.; Low, D.K.W.; Newman, S.F.; Kim, J.; et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nat. Biomed. Eng.* **2018**, *2*, 749–760. [[CrossRef](#)] [[PubMed](#)]
74. GitHub. SHAP. 2016. Available online: <https://github.com/slundberg/shap> (accessed on 15 November 2020).
75. Kaggle. Machine Learning Explainability Home Page. 2020. Available online: <https://www.kaggle.com/dansbecker/shap-values> (accessed on 15 November 2020).