# COVID-19 detection through X-Ray chest images

Diego Hernandez
*Department of Electronics, Telecommunications. and Informatics.*
*University of Aveiro*
Aveiro, Portugal
dc.hernandez@ua.pt

Rodrigo Pereira
*Department of Electronics, Telecommunications. and Informatics.*
*University of Aveiro*
Aveiro, Portugal
rodrigo.pereira@ua.pt

Petia Georgevia
*Department of Electronics, Telecommunications. and Informatics.*
*University of Aveiro*
Aveiro, Portugal
petia@ua.pt

***Abstract*—The new COVID-19 virus has proven to be a real threat to the humanity. In this work we propose a machine learning approach to identify cases of infected patients through X-Ray images of their lungs. Due to the scarceness of the available data and limited computational power, we come up with two approaches: i) Build a custom Convolutional Neural Network (CNN) from scratch, with large data set of historical not COVID-19 pulmonary X-Rays. Tune the final layers with COVID-19 X-Ray images; ii) Apply transfer learning through pretrained CNN models (ResNet, VGG, DenseNet) and fine tuning with COVID-19 data. The second approach allowed us to reach around 90% accuracy on this challenging task.**

***Keywords*—COVID-19, Transfer Learning, VGG, ResNet, DenseNet.**

## I. Introduction

Late in 2019, in the city of Whuan (China), was reported the first infection by the new Corona Virus (SARS-CoV-2). Since then, the virus has spread around the world, becoming the worst pandemics humanity has faced in this century. Testing and isolating carriers of this virus has proven to be crucial to stop it. The current means to test individuals consist of a Polymerase Chain Reaction (PCR) Throat Swab test, that holds a sensitivity of 99% and a specificity of 98%, if preformed correctly [1]. But the testing capability of each country is still a problem.

Our hypothesis is that despite the effects of this virus, similar to pneumonia, there might be a differentiating factor in the lungs of the patient. This factor may distinct, even if slightly, from the effects of pneumonia.

The objective of the present work is to figure out if there are common characteristics between the lungs X-rays images taken from COVID-19 patients, that differentiates themselves, from the X-rays images taken from patients that does not have COVID-19. This is achieved, through deep learning algorithms that attained decent levels of accuracy.

The paper is organized as follows. In Section II we present briefly the related work that we were based on. Data set and the computational resources are presented in section III. In sections IV and V the COVID-19 detection is considered as binary and multi-class problem, respectively. Finally, in Section V, conclusions are drawn.



## II. Related Work

SARS-CoV-2 is a new virus and COVID-19 is a new disease, but beyond being a new biological and etiological entity, it is also the first time we deal with a worldwide pandemic in the era of big data. Thus, several calls have been made to the academic community to respond to the COVID-19 pandemic with data science, artificial intelligence and machine learning [2]–[5].

However, the problem of COVID-19 detection through X-Ray chest images is a new one and to the best of our knowledge so far there is no previous work. Though we did not find any related papers, we took an inspiration by the paper [6], where the dataset Chexpert was used to classify multi-labeled X-Ray images applying the ResNet50 Convolutional Neural Network (CNN) architecture.

## III. Data and Computational Resources

### *A. Data Retrieval*

Available data about COVID-19 patients is still not sufficient, however, the Italian Society of Medical and Interventional Radiology has made available a limited number of X-Ray images of patients infected with COVID-19 (*https://www.sirm.org/category/senza-categoria/covid-19/*). From 70 different cases, we selected 58 with a frontal perspective, as shown in Fig. 1. The second data source used in this study was the

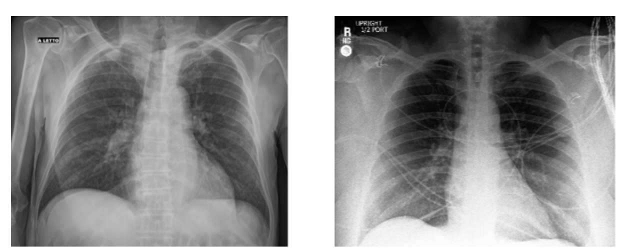

Fig. 1. Dataset 1 samples: X-Ray images of covid19 infected patient (left image) and healthy patient (right image)

large data-set of pulmonary X-Rays, named ChexPert (*https://stanfordmlgroup.github.io/competitions/chexpert/*), provided by the University of Stanford. Details about the data are presented in Fig. 2.

| Pathology | Positive (%) | Uncertain (%) | Negative (%) |
|---|---|---|---|
| No Finding | 16627 (8.86) | 0 (0.0) | 171014 (91.14) |
| Enlarged Cardiom. | 9020 (4.81) | 10148 (5.41) | 168473 (89.78) |
| Cardiomegaly | 23002 (12.26) | 6597 (3.52) | 158042 (84.23) |
| Lung Lesion | 6856 (3.65) | 1071 (0.57) | 179714 (95.78) |
| Lung Opacity | 92669 (49.39) | 4341 (2.31) | 90631 (48.3) |
| Edema | 48905 (26.06) | 11571 (6.17) | 127165 (67.77) |
| Consolidation | 12730 (6.78) | 23976 (12.78) | 150935 (80.44) |
| Pneumonia | 4576 (2.44) | 15658 (8.34) | 167407 (89.22) |
| Atelectasis | 29333 (15.63) | 29377 (15.66) | 128931 (68.71) |
| Pneumothorax | 17313 (9.23) | 2663 (1.42) | 167665 (89.35) |
| Pleural Effusion | 75696 (40.34) | 9419 (5.02) | 102526 (54.64) |
| Pleural Other | 2441 (1.3) | 1771 (0.94) | 183429 (97.76) |
| Fracture | 7270 (3.87) | 484 (0.26) | 179887 (95.87) |
| Support Devices | 105831 (56.4) | 898 (0.48) | 80912 (43.12) |

Fig. 2. Dataset 2: Chexpert meta data

### B. *Data Augmentation*

Given the small amount of cases we had for COVID-19, we resorted to data augmentation as a strategy to enable the increase of diversity of available data for the training models, as exemplified in Fig. 3. The python library OpenCV was used to accomplish the augmentation operations such as cropping, horizontal flipping, Gaussian noise. Keras library was also used for re-scaling of the pixel's channels intensities.

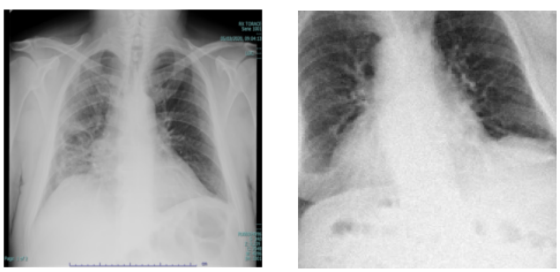

Fig. 3. X-ray Images: original (left image), generated (right image)

### C. *Resources*

The deep learning models were trained in Google Colab environment (*https://colab.research.google.com/notebooks/intro.ipynb*). Colab is a cloud service allowing AI developers to train neural network models on a Tesla K80 GPU for free, allowing to handle efficiently complex computational tasks. It also allows each user to have available about 24 GB of RAM and 110 GB of disc memory. However, Colab does not guarantee the resource availability, in some occasions we weren't able to use GPU because of a overused GPU limitation. For more information about Google Colab limitations, check *https://research.google.com/colaboratory/faq.html*. In order to handle memory restrictions, we ended up loading approximately 6000 images for testing purposes and 27000 x-ray lung images to train the models.

## IV. Binary classification problem

Due to the scarceness of the X-Ray images of the lungs of patients diagnosed with COVID-19, even after the augmentation, we applied two approaches.

### A. *Custom End-to-End trained CNN model*

A Deep Learning (DL) CNN model was designed and trained from scratch to detect healthy and not healthy lungs, with the extensive amount of historical (not COVID-19) pulmonary X-Rays from Chexpert. After this stage (training stage 1), the model weights were freezed and more layers were inserted and trained with COVID-19 X-Ray images (training stage 2). The final architecture of the optimized end-to-end trained (training stages 1 and 2) custom model is depicted in Fig. 4.
Data set was divided into 80% for training and 20% for testing. 20% from the training data was left for validation. In order to satisfy the conflict between memory restrictions and image resolution, the images were re-scaled from 300x300 px to 150x150 px. The final (test) performance achieved by this model is summarised in Table I. The poor performance both on training (74% accuracy) and on validation data (68% accuracy) indicated under-fitting issues. Increasing the complexity of our custom model (i.e. training deeper CNN) may possibly improve its performance, however the vanishing gradient problem made the search for the best combination of hyper-parameters extremely long (weeks). Hence, in order to speed up the training process we applied the concept of Transfer Learning.

### B. *Transfer Learning*

The programming environment Keras has provided several deep learning models alongside with their optimized weights. These pre-trained models (many of them trained with the large ImageNet database) has boosted significantly the computer vision research. For the present work, we selected two of the most widely used in competitions deep models, namely ResNet50 and VGG16 and were able to accomplish the feature extractions and fine-tuning.

*1) ResNet50:* The Residual Neural Network architecture (ResNet) was proposed by a Microsoft research team in 2016 [7]. What differentiates ResNet from other architectures, is the use of the skip or identity layers. Stacking more convolutional layers should, in theory, produce better results, since the network would discover higher (more complex) level of features. However, in practice, there is a certain threshold where adding more layers decreases the model's performance. Skip connection (short cut) takes the activation from one layer and feed it to another layer much deeper in the network. Skip connection build networks that enable to train very deep structures (even over 100 layers).

As the base model, we imported from Keras ResNet50, with 50 pre-trained layers. We removed the output layer, "freezed" the weights of the base model (by changing the trainable flag to False), and added new trainable layers at the end -

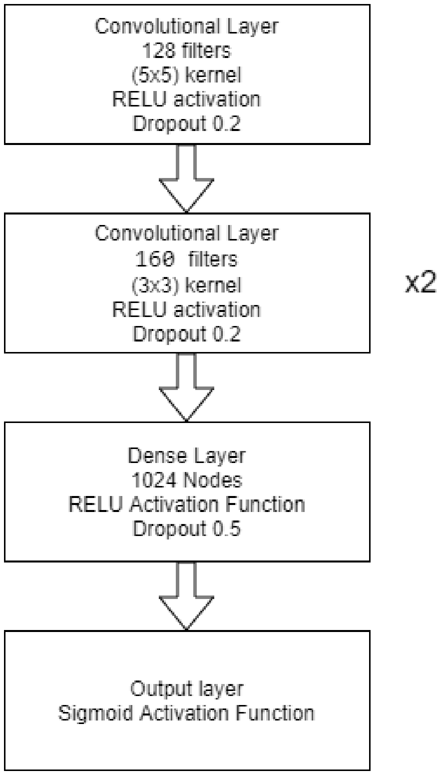


Fig. 4. Final Model 1: end-to-end trained custom architecture

hidden dense layers and an output sigmoid layer. A number of different models with fixed ResNet50 base and varying hyper-parameters were trained: number of added hidden dense layers (1, 2, 3, 4), number of neurons per layer (8, 16, 32, 64, 128). The results with respect to the Accuracy on validation data are summarised in Fig. 5.

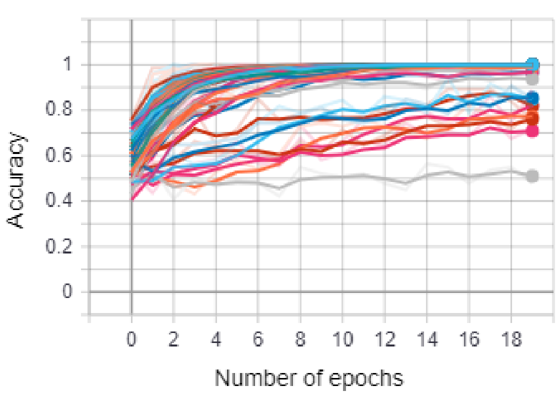


Fig. 5. Search for the best ResNet50 hyper-parameters

The final model is depicted in Fig. 6. The activation functions implemented in the extra layers was Rectifier Linear Unit [`ReLU`]. Adding Dropout Layer [8], did not improve the results, therefore it was not added in the final architecture. In order to get better fine tuning in the weight changes during the training process, `Adam` optimizer was chosen with learning rate of 0.0005 and the binary cross entropy loss function. The final (test) performance achieved by Model 2 is summarised in Table I.

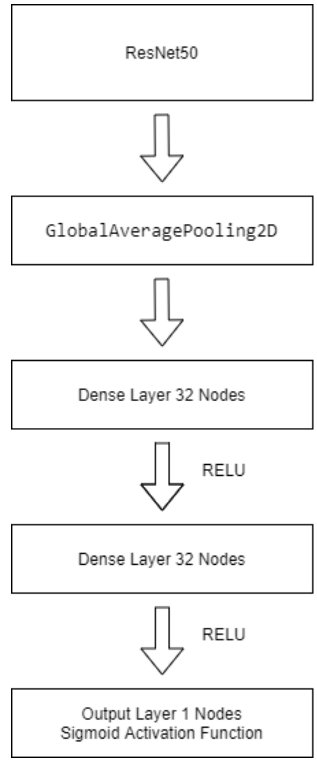


Fig. 6. Final Model 2: ResNet50 + fine tuning (ResNet50+)

*2) VGG16:* : The Oxford's Visual Geometry Group (VGG) proposed in 2015 [9], a deep convolutional network for object recognition, trained with ImageNet database. Nowadays, VGG continues to outperform other CNN on many different tasks and datasets outside of ImageNet. The image is passed through a stack of convolutional layers, using filters with small receptive fields: 3x3 and 1x1. Spatial pooling is carried out to reduce the size of the representations, speed up the computations and make the features more robust. Max-pooling or Average-Pooling is performed over 2x2 pixel window, with stride 2. The intuition behind is that large numbers mean there is some strong feature detected in this part of the image, which is not present in another part. Whenever this feature is detected it remains preserved in the output. VGG pooling is carried out by five layers, however not all convolutional layers are followed by a max or average pooling.

Two VGG architectures were proposed: with 16 trainable layers (VGG16) and with 19 trainable layers (VGG19). For the present study both models demonstrated similar performance, therefore the smaller version VGG16 was chosen. The same procedure was followed as with the ResNet50 model. First, we imported the pre-trained base VGG16 model, removed the output layer and added new layers at the end - dense layers and an output sigmoid layer.

A set of different models with fixed VGG16 base and varying hyper-parameters were trained: number of hidden layers, number of neurons in the layer, dropout rates, learning rates, number of epochs. After analyzing the results we ended up going forward with the following architecture presented in Fig. 7. The performance of Model 3 (Fig. 7) trained with `Adam` optimizer, learning rate of 0.0003 and binary cross entropy loss function is shown in Fig. 8. Note that in contrast to Model 1 (Custom CNN), Model 2 (VGG16+) clearly exhibits overfitting issues, with accuracy metrics on training data much higher than the low and wavy accuracy on test data.

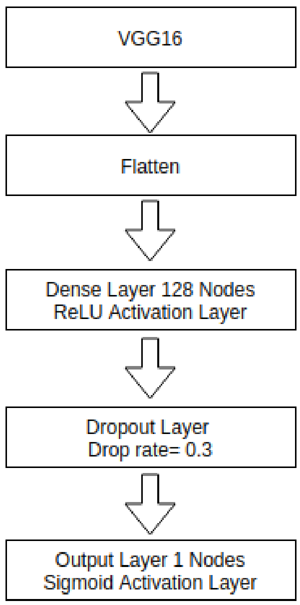


Fig. 7. Final Model 3: VGG16 + fine tuning ((VGG16+)

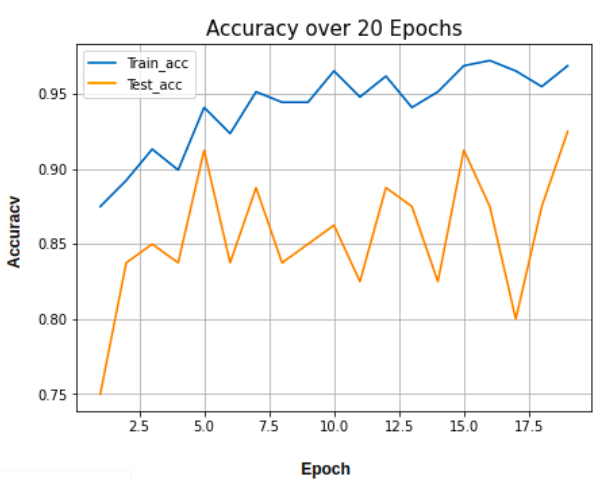


Fig. 8. Model 3 (VGG16+) performance

## V. Multi-class scenario

In the previous section we trained three models (Model 1, 2, 3) to predict if a patient has COVID-19 or not, given a digital x-ray image of the patient's lungs in a frontal

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| Model 1 (Custom CNN) | 0.68 | 0.67 | 0.63 | 0.65 |
| Model 2 (ResNet50+) | 0.9063 | 0.90 | 0.9167 | 0.9072 |
| Model 3 (VGG16+) | 0.8229 | 0.8039 | 0.8541 | 0.8282 |
| Model 4 (DenseNet121+) | 0.834 | 0.89 | 0.67 | 0.7619 |

TABLE I
Model performance metrics on test data

perspective. Now, we extend the problem and consider multi-class classification to determine if the patient is healthy, has pneumonia not originated by COVID-19 or has COVID-19.

DenseNet architecture was chosen as more relevant for this prediction problem. The convolutional networks can be deeper, more accurate and efficient to train if they contain shorter connections between layers close to the network input and those close to the output. Dense Convolutional Network (DenseNet) connects each layer to every other layer in a feed-forward fashion, [10], [11]. DenseNet is an extension of ResNet, however unlike ResNet that sums the output feature maps of the layer with the incoming feature maps, DenseNet concatenates them. The network consists of Dense Blocks (Fig. 9), where the dimension of the feature maps remains constant, but the number of filters varies. The filters, called Transition Layers, are in charge of the down-sampling with batch normalization, a convolution with kernel 1x1 and a pooling layer with kernel 2x2. In a Dense Network, every layer has access to its preceding feature maps. Hence, each layer is adding new information to this collective knowledge.

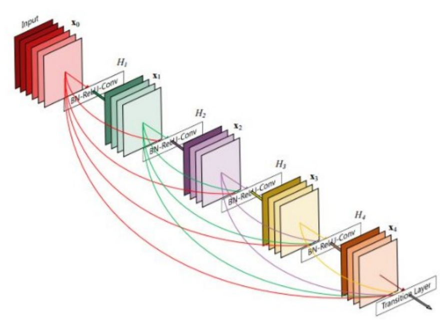


Fig. 9. Dense Block

We imported a pretrained DenseNet base model, removed the output layer, "freezed" the weights of the base model, and added new trainable layers at the end of the network, as well as a new output layer. As previously, a set of models with fixed base and varying hyper-parameters were trained. The final architecture, presented in Fig. 10, was obtained with base network of 121 layers (DenseNet121) trained with `Adam` optimizer, learning rate of 0.0005 and categorical cross entropy loss function. The results of training and validation are summarised in Fig. 11. Note that in contrast to Model 3 (VGG16+), Model 4 (DenseNet121+) is less prone to overfitting, with closer accuracy metrics on training and validation data. However, at the testing phase, this model reveals to be

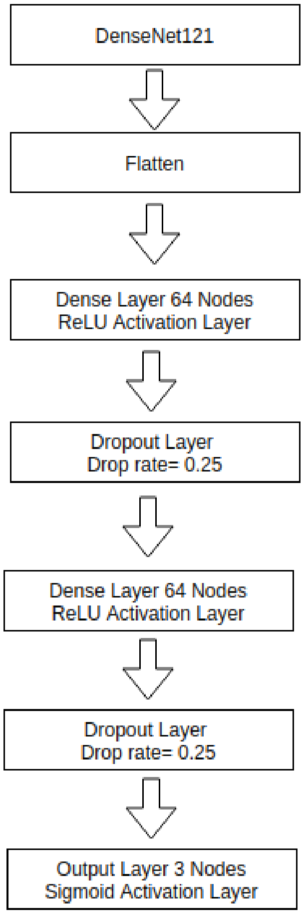


Fig. 10. Final Model 4 (DenseNet121+): DenseNet121 + fine tuning.

less confident and outputs a high number of false positive and false negative as shown in table II.

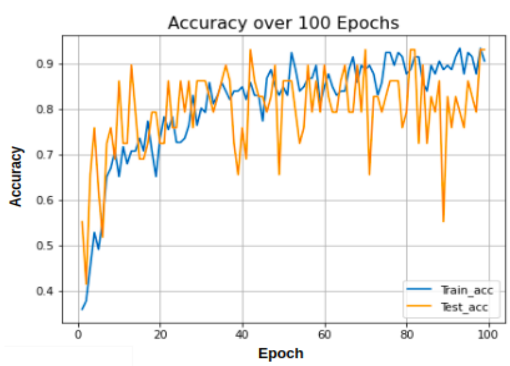


Fig. 11. Model 4 (DenseNet121+) performance.

## VI. Conclusions

In this paper we presented a proof of concept hypothesis that COVID-19 infected patients can be detected based on their X-ray chest images. In order to overcome the problem with the lack of sufficient COVID-19 related data we investigated two approaches. First, use a large data set of not COVID-19 X-ray chest images (ChexPert) and train a custom CNN. Then, fine tune the model with the small COVID-19 data. This approach produced Model 1, that did not achieve promising detection of COVID-19 infected patients. The second solution was to import pre-trained deep learning models and fine tune them with the COVID-19 data. Here we designed and tuned two models based on ResNet50 (Model 2) and VGG16 (Model 3), respectively. Both models outperformed Model 1 as shown in Table I.

Next, we decided to further extend the second approach to detect if the patient is healthy, has pneumonia not originated by COVID-19 or has COVID-19. For this more challenging, multi-class scenario, we designed Model 4 based on DenseNet121 architecture. Though the model demonstrates better quality than Model 1, it outputs far too many positive or negative false predictions.

Nevertheless, this promising study opens a new way to test and detect COVID-19 infected patients.

| | | Number of Images | Prediction | | |
|---|---|---|---|---|---|
| | | | **Healthy** | **Pneumonia** | **Covid19** |
| **Actual** | **Healthy** | 12 | 41.7% | 33.3% | 25% |
| | **Pneumonia** | 24 | 33.3% | 45.8% | 20.8% |
| | **Covid19** | 12 | 33.3% | 25% | 41.7% |

TABLE II
Model 4 (DenseNet121+) performance on test data.

## Acknowledgment

This work was funded by National Funds through the FCT - Foundation for Science and Technology, in the context of the project UID/CEC/00127/2019.

## References

[1] Echavarria M, Sanchez JL, Kolavic-Gray SA, et al. *Rapid detection of adenovirus in throat swab specimens by PCR during respiratory disease outbreaks among military recruits.* J Clin Microbiol. 2003;41(2):810–812. doi:10.1128/jcm.41.2.810-812.2003

[2] Van der Schaar, Mihaela, Responding to COVID-19 with AI and machine learning.http://www.wanders-lab.com/NewWebsite/COVID.html,

[3] Rapid Assitance in Modelling the Pandemic RAMP https://epcced.github.io/ramp/.

[4] Real Time Epidemic Datathon, https://www.epidemicdatathon.com/references

[5] C. Anastassopoulou, L. Russo, A. Tsakris C. Siettos, Data-based analysis, modelling and forecasting of the COVID-19 outbreak. PLOS ONE, vol.15 (3), 2020, doi.org/10.1371/journal.pone.0230405.

[6] Baltruschat, Ivo Nickisch, Hannes Grass, Michael Knopp, Tobias Saalbach, Axel. (2018). Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification. Scientific Reports. 10.1038/s41598-019-42294-8.

[7] He, K., Zhang, X., Ren, S., Sun, J. (2016). *Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* doi: 10.1109/cvpr.2016.90

[8] Nitish S, Geoffrey H, Alex K,Ilya S,Ruslan S, et al. (2014) *Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15 (2014) 1929-1958*

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[10] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/cvpr.2016.90

[11] DenseNets http://www.pabloruizruiz10.com/resources/CNNs/DenseNets.pdf