Explainable Detection of COVID-19 from Chest X-Ray Images

Franco Ruggeri **Fredrik Danielsson** **Muhammad Tousif Zaman** **Milan Jolić**

Abstract

Convolutional networks have proven to be very effective when it comes to image classification and they are especially promising when it comes to medical image analysis. For this project we set out to train two deep convolutional neural networks (ConvNets), the popular ResNet50 [1] and COVID-Net [2], to classify chest X-ray (CXR) images of medical patients with COVID-19.

We tested whether transfer learning from the large-scale ImageNet dataset [3] and HAM10000 [4], a dataset of skin cancer images, is useful for classifying COVID-19 using CXR images, and we tried to apply data augmentation to reduce overfitting. Furthermore, we boosted our models with explainability, using Grad-CAM [5] and Integrated Gradients [6]. This turns out to be very important for using the model in a real context, as support tool to doctors.

Our results essentially prove that ConvNets can be effectively used to classify COVID-19 cases, showcasing good accuracy, precision and recall. We also show that the predictions of our models are based on meaningful aspects, by means of the above-said explainable AI methods.

1 Introduction

COVID-19 pandemic is still at large in many countries, taking toll in both casualties and economic and medical resources. As such, efficiency is key in containing the disease before we are fully capable to combat it. As two main methods of screening are reverse transcriptase-polymerase chain reaction (RT-PCR) testing and radiography examination. The second method utilizes chest X-ray scanning (CXR).

What makes CXR incredibly useful is three-fold: its speed, availability and portability, solving issues such as transmission during transport or lack of tests. Most of the time consumed in CXR scanning falls on examining the images to find indicators of disease, which can often be subtle. Therefore, using neural networks and interpretability methods like Grad-CAM or Integrated Gradients as tools in medical work can make finding those indicators easier.

In this work, we compare the popular ResNet50 [1] with COVID-Net [2], an architecture tailored for this task. The goal in this part is to investigate the benefits of common techniques such as transfer learning and data augmentation. Moreover, we applied two explainable AI methods, Grad-CAM [5] and Integrated Gradients [6], in order to check whether the predictions of the classifiers are based on meaningful aspects. This turns out to be essential when it comes to deploying a model in the real world, both to speed-up the doctor's final assessment and to know when to ignore a prediction.

Our results are good and confirm the effectiveness of ConvNets for image classification. Our best model reached an accuracy of 90%, with precision and recall for COVID-19 cases of 75% and 78%, respectively.

The whole project was developed using the popular and powerful software library TensorFlow. It is open source and available on GitHub.

2 Related Work

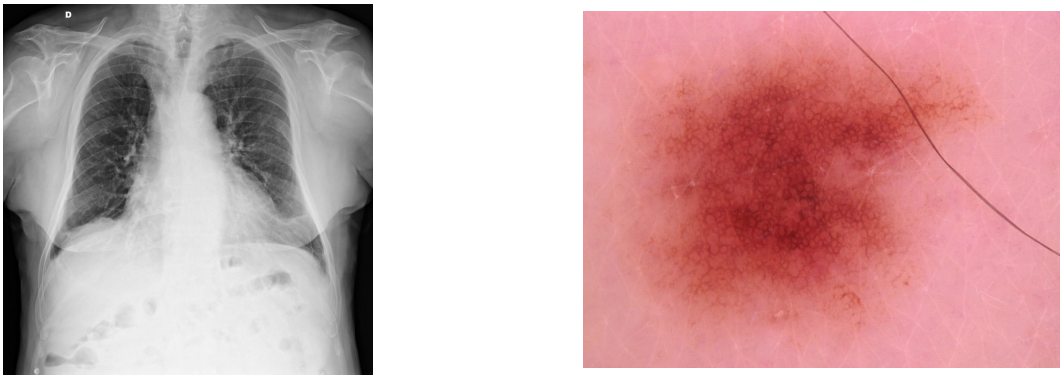
Considering the current pandemic, there are many papers that focus on working with CXR and computed tomography (CT) images and neural networks combined with other machine learning methods.

One of these is [7], which utilizes Grey Level Co-occurrence Matrix (GLCM), Local Directional Pattern (LDP), Grey Level Run Length Matrix (GLRLM), Grey-Level Size Zone Matrix (GLSZM), and Discrete Wavelet Transform (DWT) algorithms for feature extraction and Support Vector Machines for classification, showing 99.68% accuracy for GLSZM and 10-fold cross-validation.

Other research combining NNs with another machine learning technique is [8], which combines ConvNet with three decision trees. First tree decides whether the CXR image is normal or abnormal, while second and third tree classify the existence of signs of tuberculosis and COVID-19 respectively. Third tree shows 95% accuracy.

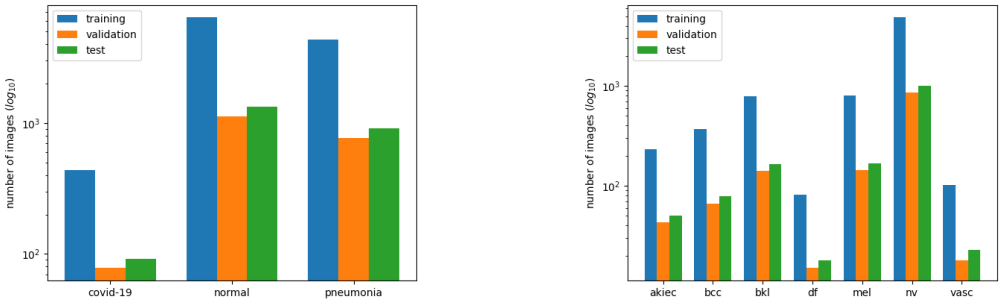
The others are similar in work to [2] and most showcase high level of accuracy (90% range). Some of these include [9] and [10].

3 Data



(a) (b)

Figure 1: Example of image for COVIDx (a) and HAM10000 (b).



(a) (b)

Figure 2: Distribution of images for each class in the training ($\approx 70\%$), validation ($\approx 15\%$), and test ($\approx 15\%$) sets in COVIDx (a) and HAM10000 (b) datasets. The split was done by means of stratified sampling to preserve the distribution over the classes. Note that the y-axis is in log scale.

We used two datasets, COVIDx [2] and HAM10000 [4], which are described in the following subsections. Examples of images and distributions are shown in figures 1 and 2, respectively.

In order to cross-validate and test our models, we utilized hold-out, splitting both datasets in training ($\approx 70\%$), validation ($\approx 15\%$), and test ($\approx 15\%$) sets. The split was done by means of stratified sampling, so the distribution across the classes was preserved. Importantly, the split was done by sampling the IDs (patient ID for COVIDx, lesion ID for HAM10000) instead of the images, so images with the same ID were put in the same subset. This ensures a more realistic and less biased generalization assessment.

Furthermore, Tensorflow provides the weights pretrained on ImageNet [3] for popular architectures like ResNet50, which we used. Thus, ImageNet was indirectly useful. Unfortunately we were not allowed to download it and for this reason we could not pretrain COVID-Net.

3.1 COVIDx

COVIDx [2] is a combination of several open-source datasets and contains CXR images labelled as COVID-19, normal, or pneumonia. Coming from different sources, the images had different formats (DICOM, JPEG, PNG, BMP) and number of channels (gray-scale or RGB), so we decided to preprocess them converting everything to RGB and PNG, for easier management.

We used this dataset for all our experiments. As we can see in figure 2a, it is highly imbalanced, with very few COVID-19 images. This led us to use metrics and strategies suitable for imbalanced classification problems, as discussed in sections 4.3.1 and 4.3.2.

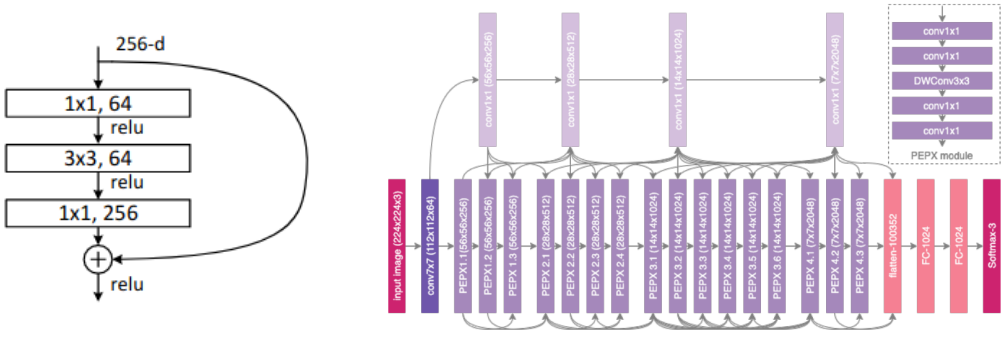
3.2 HAM10000

HAM10000 is an open-source dataset containing skin cancer images labelled with 7 classes. A more detailed description goes beyond the scope of this work, but can be found in [4].

This dataset was utilized for pretraining and verifying that transfer learning from skin cancer to COVID-19 detection is useful.

4 Methods

4.1 Architectures



(a) (b)

Figure 3: Basic block of ResNet50 [1] (a) and complete COVID-Net architecture [2] (b).

4.1.1 ResNet50

ResNet50 is one of the state-of-the-art deep residual networks [4] presented during the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2015. It exploits the idea of deep residual learning with skip connections and makes use of batch normalization and bottleneck structures. The basic block is shown in figure 3a.

Since it is currently one of the best choices for image classification and the weights pretrained on ImageNet are available in TensorFlow, enabling transfer learning, we selected it as one of the two architectures for our experiments.

4.1.2 COVID-Net

The COVID-Net architecture was originally generated with a human-machine collaborative design [2]. Basically, the human knowledge on the state-of-the-art residual connections was given as constraint to a generative tool, which optimized the architecture for the COVID-19 detection task. As shown in figure 3b, the resulting network uses heavily a bottleneck pattern called PEPX (Projection, Expansion, Projection, Extension), with an efficient depth-wise convolution in the middle.

Unfortunately, the authors did not provide low-level details, so we decided on trying the following methods in order to implement it, based on state-of-the-art architectures such as ResNet50, as well as trial and error:

- Batch normalization, ReLU activation, and L_2 regularization for the fully connected layers.
- Max pooling 3x3 with stride 2 at the end of each block (where the spatial dimensionality decreases).
- No activation and batch normalization for the convolutional layers.
- Inputs rescaled to the range [-1,1].

4.2 Training techniques for ConvNets

4.2.1 Transfer learning

Deep architectures need huge amounts of labelled data for training. A technique heavily exploited in computer vision to overcome this is transfer learning. We examined its effect when applied with ImageNet and HAM10000. The steps we followed are:

1. Pretrain the model on ImageNet or HAM10000.
2. Replace the layer on top with a linear classifier with the correct number of units.
3. Train until convergence the linear classifier on top, freezing the rest of the network.
4. Fine-tune some layers, unfreezing them and training with a very small learning rate.

Step 1 is not necessary with ResNet50 and ImageNet, since the weights are already available in TensorFlow. Step 2 is required as COVIDx has a different number of classes than ImageNet and HAM10000. Last but not least, step 3 is critical and cannot be skipped, because the randomly-initialized weights of the linear classifier on top cause very large gradient updates at the beginning, that would ruin the pretrained weights of the other layers.

4.2.2 Data augmentation

Going deeper increases inevitably the number of parameters, exposing to the risk of overfitting. For image classification, one of the most common regularization techniques is data augmentation. We applied the following augmentations:

- Random rotation in the range [-10, +10] degrees.
- Random shift in the range [-10%, +10%], both horizontally and vertically.
- Random brightness in the range [90%, 110%].
- Random zoom in the range [85%, 115%].
- Random horizontal flip.

The values were not explicitly tuned for reasons of time. We used instead values that are reasonable for this task according to [2].

4.2.3 Settings

We used cross entropy as loss and Adam as optimizer, with an initial learning rate of 0.0001, reduced by a factor of 1/100 for fine-tuning. We trained for 30 epochs + 10 when fine-tuning, selecting the best model during training according to the validation macro-averaged F1-score, as discussed in section 4.3.1. Furthermore, $\lambda = 0.01$ was used for L_2 regularization and batch size was set to 32.

Good values for learning rate, amount of L_2 regularization (λ), and number of layers to fine-tune were found with a coarse search. A finer search was beyond the scope of the project and was not done due to time constraints.

4.3 Dealing with imbalaned datasets

4.3.1 Metrics

In COVIDx, COVID-19 is the minor class (see section 3.1) as well as the one we are most interested in. For this reason and since the validation and test sets have the same imbalance as the training set, evaluating the model with a simple accuracy would be absolutely wrong and other metrics need to be considered.

Moreover, when it comes to monitoring training or comparing models, it is very convenient to have metrics with a scalar as output, so that it can be plotted as learning curve or comparisons are straightforward. There are two ways of averaging values computed for different classes: macro-average and micro-average. In imbalanced tasks in which the minor class is very important, macro-average is the best choice, as it computes the metric independently for each class and then takes the average, resulting in the same importance for each class.

Thus, we decided to use macro-averaged F1-score as main metric for monitoring training. For additional insights, we tracked also macro-averaged AUC and, for the COVID-19 class, precision and recall.

Regarding the evaluation, we used macro-averaged F1-score as well as F1-score, precision, and recall for the COVID-19 class. We also plotted a ROC curve on the data binarized for the COVID-19 class (i.e. true if labelled as COVID-19, false otherwise), since it gives very useful indications of the performance, in terms of true positives to COVID-19 and false negatives to COVID-19, that the model can reach just by tuning the output threshold.

4.3.2 Class weights

Training with an imbalanced dataset can cause the model to learn to only classify the dominating classes. One technique to compensate for this fact is to use class weights. Basically, a factor depending on which class the sample belongs to is added to the loss function.

In order to obtain the compensation, the class weights have to be inversely proportional to the class sizes. We set them using the following formula:

$$w_k = \frac{1}{n_k} \frac{\sum_{k=1}^{K} n_k}{K} \tag{1}$$

where n_k is the number of samples for the class k in the training set and K is the number of classes. The second constant factor just helps to have a range of loss values similar to the case without class weights.

4.4 Explainability

4.4.1 Grad-CAM

Until rather recently, deep learning networks have mainly been treated as black boxes, meaning that interpretability and explainability were basically not considered. Although this is perhaps still true for large parts of deep networks, Grad-CAM (Gradient-weighted Class Activation Mapping) offers a way to visualize what features the network considers relevant in a image [5].

The main idea behind Grad-CAM is that the last convolutional layer of a ConvNet offers the best trade-off between high-level information in an image and detailed spatial information (something which convolutional layers preserve very well compared to fully connected layers). At a high level, Grad-CAM works by taking the derivate of the class scores of the last convolutional layer (before applying softmax) w.r.t the feature map activations of that layer. We then apply a global-average-pooling (the width and height dimensions of the last layer) in order to capture relative importance of the weights. Then as a final step, we apply ReLU; the intuition behind this is that we are looking for features that have a positive influence on the classes we are interested in and it is likely that a pixel which has a negative global-average-value belongs to a different category [5].

4.4.2 Integrated Gradients

Integrated Gradients (IG) is another technique used to solve the interpretability issue. Paragraph below explains how IG works.

A black image is utilized as a baseline for our algorithm. We interpolate the series of images in between our baseline picture and the picture we are explaining. When getting the scores of all these pictures based on our Neural Network there exist both stagnant and changing continuations in terms of score between two neighbouring pictures. The changing continuations are important as they represent the information that affects the learning of our Neural Network the most. We call those interesting gradients. By integrating those interesting gradients we get the Integrated Gradients of the picture.

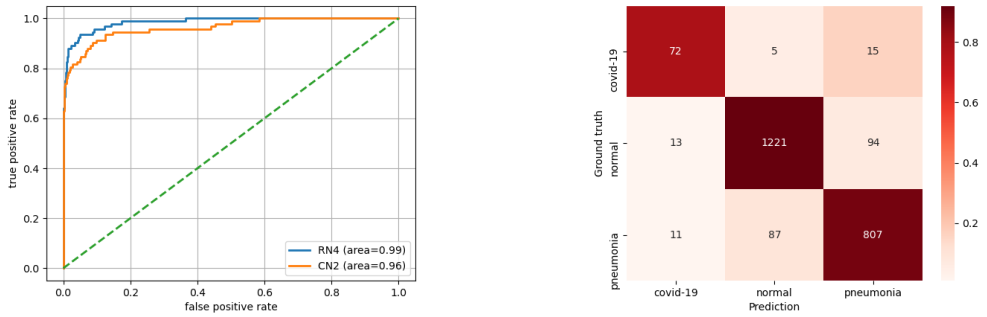
Integrated gradients are approximated using the trapezoid rule: we average the gradients, g, using the formula: $\Sigma_{k=1}^{m} g \times \frac{1}{m}$, and then we scale the averaged gradients with respect to the original image: $(x_i - x_i^{'}) \times \Sigma_{k=1}^{m} g \times \frac{1}{m}$, to obtain the integrated gradients.

5 Experiments

5.1 Performance

Table 1: Settings for the models and their performance.

							Macro-average	COVID-19		
ID	Transfer learning	Class weights	Augmentation	Fine-tuning	Free params	Accuracy	F1-score	Precision	Recall	F1-score
RN1	ImageNet	No	No	No	6,146	89	83	93	59	72
RN2	ImageNet	No	No	Yes	15,240,195	91	87	84	72	77
RN3	ImageNet	Yes	No	No	6,146	88	81	57	80	67
RN4	ImageNet	Yes	No	Yes	15,240,195	90	86	75	78	77
RN5	ImageNet	Yes	Yes	No	6,146	86	76	48	62	54
RN6	ImageNet	Yes	Yes	Yes	15,240,195	88	80	53	78	63
RN7	-	Yes	No	-	23,525,507	87	82	73	70	71
RN8	-	Yes	Yes	-	23,525,507	85	76	52	65	58
CN1	HAM10000	Yes	No	No	3,075	76	64	31	47	37
CN2	HAM10000	Yes	No	Yes	98,917,443	85	80	60	80	69
CN3	-	Yes	No	-	103,827,715	78	68	28	95	43



(a) (b)

Figure 4: ROC curve for data binarized on COVID-19 class computed for RN4 and CN2 (a) and confusion matrix computed for RN4 (b). The confusion matrix is labelled with absolute numbers, but colored with values normalized by row (more significant due to the imbalance).

Table 1 contains all the models we trained together with the respective performance.

Comparing RN1-4, we can notice that class weights improve the recall on COVID-19, while worsening the precision, as expected. In today's situation and considering that COVID-19 is contagious, it is probably preferable to have a high recall, in order to isolate possible infected patients. For this reason, we trained the rest of models using class weights.

Regarding transfer learning, pretraining both on ImageNet and on HAM10000 turns out to be beneficial compared to training from scratch, as proved by RN1-8 and CN1-3. The same results

show that fine-tuning boosts the performance of the pretrained models. A further observation is that the number of fine-tuned layers (set with a coarse search, see section 4.2.3) is quite high. This is reasonable because CXR images are really different from ImageNet and skin cancer ones, and consequently many convolutional layers need to be tuned.

Moreover, we find it surprising that data augmentation does not improve the generalization capabilities, both when training from scratch and when using pretrained weights, rather it makes performance slightly worse, as can be seen by comparing models RN3-8. Thus, we decided to train the rest of the networks without it. Possible reasons are that we did not tune the ranges of augmentation or the fill mode (set to *constant*) is not adequate. An expert would be helpful to know which augmentations are realistic for CXR images. A deep investigation of this behavior was not done for reasons of time.

In figure 4a we compare qualitatively RN4 and CN2, which we consider the best ResNet50 and COVID-Net, respectively, due to the high F1-score and recall for the COVID-19 class. Both models provide very good performance, but ResNet50 slightly outperforms COVID-Net. However, this does not prove that ResNet50 is better than COVID-Net, since RN4 was pretrained on ImageNet, that is a much larger and general dataset than HAM10000, so more suitable for transfer learning. Unfortunately, as mentioned in section 3, we were unable to gain access to the ImageNet database, the people responsible from Stanford/Princeton university did not reply to our inquiry about ImageNet, so a more fair comparison could not be done.

Finally, figure 4b shows the confusion matrix for RN4. The strong diagonal proves once again the very good generalization obtained for all the classes, including the minor COVID-19 one.

5.2 Explainability

(a) (b)

Figure 5: Grad-CAM and IG applied to two predictions on test images made by RN4.

The result of Grad-CAM and IG applied to two predictions on test images of RN4 is shown in figure 5. In both cases, both methods highlight pixels in correspondence of the lungs, proving that the model learnt meaningful features of the dataset. Moreover, in a real context in which the model is used as support tool, these methods can help the doctor to focus immediately on the important parts, speeding up and maybe improving the diagnosis.

It is difficult to understand why Grad-CAM and IG highlight different regions of interest. Although the two methods have the same intuitive intention, they differ a lot in their approach and especially when we consider this kind of high dimensional, noisy data, it is not surprising that they find different regions of interest. Grad-CAM is an exact method, while IG is an approximation and numerical precision, from the trapezoid method, could be one issue among many contributing to different results.

6 Conclusion

We can say that this project was a massive learning experience as we have learnt how to utilize a standard framework for deep learning, TensorFlow, together with the Keras API, while also showing what was our primary goal - that standard deep learning techniques can be utilized to solve the issue of classification of COVID-19 cases. Overall, we think that we managed what we set out to do. We familiarized ourselves, and gained some very interesting and useful experience, with some advanced deep learning network architectures, data augmentation techniques, explainability methods and more.

As for future extensions, some of those include finer random search of the hyperparameters, more investigating of data augmentation for this application, or integrated gradients where we did not inspect how the number of interpolated images affects the features shown as relevant. Finally, another extension can be getting access to more datasets, one of which is ImageNet, which we could use to pretrain COVID-Net and compare more fairly with the more popular ResNet50.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[2] Linda Wang, Zhong Qiu Li, and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. 2020. URL https://arxiv.org/abs/2003.09871.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1), Aug 2018. ISSN 2052-4463. doi: 10.1038/sdata.2018.161. URL http://dx.doi.org/10.1038/sdata.2018.161.

[5] Ramprasaath R. Selvaraju et al. Grad-cam: Visual explanations from deep networks via gradient-based, localization. 2019. URL https://arxiv.org/abs/1610.02391.

[6] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. 2017. URL https://arxiv.org/abs/1703.01365.

[7] Mucahid Barstugan, Umut Ozkaya, and Saban Ozturk. Coronavirus (covid-19) classification using ct images by machine learning methods. 2020. URL https://arxiv.org/ftp/arxiv/papers/2003/2003.09424.pdf.

[8] Seung Hoon Yoo, Hui Geng, Tin Lok Chiu, Siu Ki Yu, Dae Chul Cho, Jin Heo, Min Sung Choi, Il Hyun Choi, Cong Cung Van, Nguen Viet Nhung, Byung Jun Min, and Ho Lee. Deep learning-based decision-tree classifier for covid-19 diagnosis from chest x-ray imaging. 2020. URL https://www.frontiersin.org/articles/10.3389/fmed.2020.00427/full.

[9] Abolfazl Zargari Khuzani, Morteza Heidari, and S. Ali Shariati. Covid-classifier: An automated machine learning model to assist in the diagnosis of covid-19 infection in chest x-ray images. 2020. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7273278/.

[10] Asmaa Abbas, Mohammed M. Abdelsamea, and Mohamed Medhat Gaber. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. 2020. URL https://link.springer.com/article/10.1007/s10489-020-01829-7.