COVID-19 Detection using chxr Radiography

Hadji Yousra, Ihaddadene Sofiane and **Ouadda Tinhinane**
Master in Intelligent Systems engineering
Sorbonne University
Paris, France
Ben-Belkacem Yacine
Master in Advanced Systems and Robotic
Sorbonne University
Paris, France

Abstract

Le résumé synthétise en environ 200 mots la tâche abordées, les problèmes associés, la solution proposée, et les résultats principaux

The COVID-19 pandemic continues to have a devastating effect on health. It was found in early studies that patients affected by COVID-19 present abnormalities in chest radiography. We decide to use the open-source datasets made available by the research community and deep neural networks to detect COVID-19 from chest X-ray radiography. In this study, we use datasets composed of pneumonia, Covid-19, and normal X-Ray images. The study aims to explore different solutions by creating a deep neural network from scratch, transfer learning which is really interesting when we have small image datasets, and parallel networks which is a mix between transfer learning and RNN network with LSTM which uses metadata.The source code is available at

`https://github.com/Intelligent-Systems-MSc/mla-project-mla-team-15.git`

Keywords: Covid-19; Automatic Detections; X-Ray; Transfer Learning; Deep

1 Introduction

As part of the Advanced Machine Learning project, we are led to put into practice the theoretical and methodological concepts of deep learning seen in progress, for this we have chosen to work on "The analysis of X-ray images for the detection of COVID 19".

Covid-19 is a disease that has emerged in China, which has spread worldwide. This virus infects the lungs and causes potentially fatal respiratory syndromes. Diagnosis of this virus is usually made by real-time polymerase chain reaction (RT-PCR) and surprisingly, many researchers have attempted to diagnose COVID-19 using X-ray images since the virus attacks the lungs. The classification of thoracic X-ray images is not a new problem in artificial intelligence. Convolutional neural networks (CNN) have already achieved very high performance in the diagnosis of lung disease [14].

The idea is to produce high-performance and original solutions for the detection of COVID-19 using different architectures based on machine learning(CNN). We have recovered a set of data with associated metadata [3] that require pre-processing beforehand. As data the database is small, we used the X-rays provided by Pneumonia Kaggle dataset [7] and Covid X-rays [8]. The structure of the report is as follows: first, we address the state of the art of existing research on the subject. After that, we describe the techniques of pre-processing images and metadata as well as the descriptions of the data used. Finally, we will present the results obtained followed by a critical analysis of these and a conclusion.

2 Related work

Chest x-ray image classification is not new in deep learning. Many datasets have been created and many neural networks have been trained on those datasets and reported high performances. Since the publication of the COVID-19 dataset [11] by Cohen et al., many researchers tried to classify those images and created a test by merging this dataset with other chest x-ray image datasets. As an example, Wang et al. [18] trained Covid-Net, a new architecture introduced in their paper. A large dataset was used with 183 cases of COVID-19: 5,538 of Pneumonia and 8,066 of normal subjects. Several sources were for their data. They extracted a test base that contains 100 images of pneumonia and normal lungs, and only 31 of COVID-19. They explicitly mention that there is no patient overlap between the test and the training set, which can be very problematic in applications like this. They reach a 92 % accuracy. Gianluca Maguolo and Loris Nanni [13] had a critical evaluation of methods for COVID-19 automatic detection from X-Ray images. In their paper, they compared and evaluate several testing protocols used for automatic COVID-19 detection from X-Ray images in the recent literature and show that similar results can be obtained using X-Ray images that do not contain most of the lungs. After removing lungs from the images by turning to black the center of the X-Ray scan and training classifiers only on the outer part of the images. They deduce that several testing protocols for the recognition are not fair. It means that the neural networks are learning patterns in the dataset that are not correlated to the presence of COVID-19. Apostolopoulos et al. [10] combined the COVID-19 dataset and many others to create a larger dataset containing 224 images of COVID-19, 700 of pneumonia, and 504 negatives. They tested VGG19 on these data using 10 fold cross-validation and obtained 93.48 % accuracy, although on an unbalanced dataset. Tartaglione, Enzo and Barbano, Carlo Alberto and Berzovini, Claudio and Calandri, Marco and Grangetto, Marco [17] took up the challenge of current small size COVID data and they proposed an approach based on a fairly standard pipeline: pre-processing of the chest image and segmentation of the lungs followed by a classification model obtained by transfer learning. they show how significant can be the bias introduced by transfer-learning using larger public non-COVID CXR datasets.

3 Proposition

In this section we are going to describe the proposed approach based on a pre-processing of images and metadata then followed by a training of different architectures of advanced machine learning.
We've made **three** assumptions, that are developed later, in order to solve this kind of task :

- The first one is based on **Transfer Learning** from a model that does already exist (VGG19, VGG16, and RESNET50) using **only** the images as training data. These models are already pre-trained, therefore we have to specialize them for our task.
- The second assumption is building a **CNN from scratch** so that we can compare between this model and the transfer learning.
- The last assumption is combining the **images** and some **metadata** related to the clinical history of the patient. Using **Two parallel networks** such that one of them treats images and the other one the text (clinical notes). Or using a sort of **QR-Code** to directly stamp the data on the image.

Before testing these three assumptions, a pre-processing of the data is needed as in most of the cases in machine learning. Some of them are done on the images in order to improve their quality and robustness, and others are made on the metadata to encode the texts.

3.1 Pre-processing

Case of images:
As already mentioned above, removing bias in the data is very important in order not to confuse the covid19 criteria of others, we will test this hypothesis by learning from none pre-processed data 11 and then performing a test on masked lung data (the last line of Figure 11) to confirm the credibility of our results. But before that , lets describe the steps of pre-processing :
Histogram equalization : When acquiring a CXR, the so-called radiographic contrast depends on a large variety of actors, (human factors such as receptor sensors). Histogram equalization is a simple means to guarantee quite a uniform image dynamic in the data. (the second line of Figure 11)
Lung segmentation masks : the fact that only the lungs can be segmented eliminates possible biological sources, such as the presence of medical devices (the third line of Figure 11). In order to address this task, we had considered two methods:

The first method consists of generating masks for frontal chest X-ray images (PA[1] and AP[2]). For this task, we used a pre-trained model [15]. The proposed model performs lung segmentation from Chest X-rays using a network with variational data imputation and a U-net type segmentation network with encoder and decoder. The model is trained to predict PA/AP views. We generate masks using the "predict.py" file that takes X-ray images as input and generates the corresponding masks.

The previous method allows to generate masks only for the frontal chest X-ray images, for the other images (L[3] and Axial[4]) we made a pipeline standard of segmentation, we coded up an algorithm based on the watershed algorithm like the one presented in [6] with some modifications.

first of all, to separate different objects in an image we use marker-based watershed segmentation, we are starting by creating the internal marker by thresholding the Image and removing all regions but the biggest one. The external marker is created by morphological dilation of the internal marker then we superimpose the 2 markers with different grayscale values to get the watershed marker. In order to do the algorithm, we also need the Sobel-Gradient-Image, which describes the contours of our original scan. To perfect the segmentation a post-process is applied to reduce the remaining residues and to assemble the two largest connected components.

Case of metadata:

The metadata used are brought from [3] with its corresponding images dataset (about 800 covid and non-covid images).
They contain information about the clinical past of patients, we thought that it's a good idea to include them with our images to detect a potential infection by covid.
In order to do that we have to encode them first, we use the lab code of the RNN course (Lab 703). Therefore we convert the clinical notes into encoded vectors that will feed an RNN Network.

3.2 Training

3.2.1 Using only images datasets

We will describe here some of the existing state-of-the-art deep learning image classifiers that we have tested to accomplish the COVID-19 detection. **VGG16[5] :**

VGG16 is a convolutional neural network proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" [16]. The model achieves 92.7% test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes including X-ray images. figure 12

We took this architecture and removed the 3 fully connected layers. After this we add a convolutional layer with 64 filters with padding + average pooling. We add 3 dense layers with respectivly 128, 64, 2 neurons with softmax in the output layer. You can the new architecture in section 6 figure 13

VGG19 :

VGG19 was developed based on the convolutional neural network architecture by Oxford Robotics Institute's Karen Simonyan and Andrew Zisserman [16]. It was addressed at the 2014 Large Scale Visual Recognition Challenge (ILSVRC2014). The VGGNet performed very well on the imageNet dataset. VGG16 and VGG19 have different depths and layers. VGG19 is deeper than VGG16.The number of parameters for VGG19, however, is larger and thus more expensive than VGG16 to train the network. figure 14

We took this architecture and we did exactly the same operations as VGG16 network. You can the new architecture in section 6 figure 15

Resnet50 :

Residual neural network (ResNet) model is an improved version of convolutional neural network (CNN). ResNet adds shortcuts between layers to prevent the distortion that occurs as the network gets deeper and more complex [12]. In addition, bottleneck blocks are used to make training faster in the ResNet model [19]. ResNet50 is a 50-layer network trained on the ImageNet dataset. figure 16

1. (PA) Posterior-Anterior projection. The standard chest radiograph is acquired with the patient standing up, and with the X-ray beam passing through the patient from Posterior to Anterior (PA). The chest X-ray image produced is viewed as if looking at the patient from the front, face-to-face.
2. (AP) Anterior-Posterior projection. Sometimes it is not possible for radiographers to acquire a PA chest X-ray. This is usually because the patient is too unwell to stand. The chest X-ray image is still viewed as if looking at the patient face-to-face. [9]
3. (L) The lateral view of the chest X-ray image is performed erect left or right lateral and labeled with the side closest to the cassette. It allows for localization of suspected chest pathology when assessed in conjunction with a PA view
4. (Axial) the axial plane – view looking down through the body for images produced by CT scans which use X-rays, radiation detectors and computers
5. Visual Geometry Group Network

We remove the fully connected output layer. After this we add a convolutional layer with 64 filters with padding + average pooling and 3 dense layers with respectivly 128, 64, 2 neurons with softmax in the output layer.

From Scratch :

One of the solutions is a CNN-based algorithm from scratch. A CNN is first and foremost a neural network. An input data, in this case, an X-ray image, travels through the different layers of the network that learns to draw characteristics from it, the features, which represent the input data at a semantic level of higher until it can be classified: "this X-ray is COVID" or "this X-ray is NON-COVID".The architecture of our CNN is in section 6 figure 17.

The Pooling layer is an operation that was applied between two convolution sofas. It receives in addition the feature maps formed out of the convolution layer and its role is to reduce the size of the images while preserving their most essential characteristics. We used the average pool did not the operation is to keep at each step, the average value of the filter window. At the end of CNN's architecture of Fully Connected Layer (FC) is placed. After receiving an input vector, the FC layer successively applies a linear combination and then an activation function in the but final to classify the image input. Finally, it returns a size-2 vector corresponding to the number of classes in which each component represents the probability for the input image to belong to a class. Between each FC layer, we added Dropout to avoid over-learning.

3.2.2 Using images datasets and metadata

This solution is more like a proof of concept, the idea here is to combine images and text to diagnose diseases in patients. We can think of it as a method that makes data fusion, and it is very useful in medical diagnosis, we can imagine data fusion from different sources of information like radiographys, physiological data, physician's clinical notes ...

So the architecture (Figure 18) is actually two parallel networks :

- The first one is a VGG16 network like the one used before: it processes the images.
- The second network is a Recurrent Neural Network (RNN) with Long Shot Term Memorie cells (LSTM) that are bidirectional.
- Each network ends with a dense layer, therefore a concatenation of the last dense layers of the networks is done.
- Then the concatenated layers are connected to fully connected layers that end with a Softmax activation.

Another way to do this data fusion is by converting the metadata into an image that will be stamped on the radiography image. Then process the whole new output as an image (Figure 19).

- The first step is to increase the size of the radiography in order to create a space where to stamp the metadata
- The second step consists of creating a matrix from the metadata that we paste on the image.
- The last step is processing the whole in the VGG16.

4 Experiment

4.1 Data

4.1.1 COVID-19 dataset

Our source of COVID-19 images is the repository made available by Cohen et al. [11], which is the main source of most papers dealing with COVID-19. At the moment we are writing, it contains about 500 images of x-ray images of patients potentially positive to COVID-19. It's the only database where Metadata is available for every sample, containing the patient ID and, most of the time, the location and other notes that contain the reference to the doctor that uploaded the images. Some of the samples are in Figure 20.

4.1.2 Pneumonia Kaggle dataset

In 2017 Dr. Paul Mooney started a competition on Kaggle on viral and bacterial pneumonia classification [7]. It contained 5,863 pediatric images, Some of the samples can be seen in Figure 21.

4.1.3 COVID-19 Radiography database

A team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh along with their collaborators from Pakistan and Malaysia in collaboration with medical doctors have created a database of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. [8].

In the latest release, there are 1200 COVID-19 images. 400 of them came from another Github source [2], 183 from a Germany medical school [4] and 617 CXR image from SIRM, Github, Kaggle Tweeter [5, 1, 11]. This database also contains 1341 normal images, and 1345 pneumonia images from Pneumonia Kaggle dataset [7]. The team will continue to update this database as soon as they have new x-ray images for COVID-19 patients.

4.2 Experimental setup

4.2.1 transfert learning

We made different experiments that we are going to detail right afterward. In all experiments, our training and test sets consist of combinations of datasets that we introduced in the previous section.

experiment 1 we use all the COVID images of belonging COVID-19 DATASET (4.1.1) and for non-COVID images, we make a mixture of pneumonia and normal images from PNEUMONIA KAGGLE DATASET (4.1.2) it should be noted that for this experiment **we kept the basic images without any pre-processing**.

experiment 2 We use the same training and test datasets as in experiment 1 however this time we go through the pre-processing process first.

experiment 3 We use the same training and test datasets as in experiment 1 however this time we go through the pre-processing process first.

experiment 4 this time to check if the algorithm generalizes well. A test is carried out on a large base of images, the COVID images are from COVID-19 RADIOGRAPHY DATABASE 4.1.3 (about 960), and not COVID images are those not selected in experiment 2 from PNEUMONIA KAGGLE DATASET 4.1.2(about 5200).

4.2.2 from scratch

For this part, we conducted two experiments:

experiment 5 We use only images that belong to COVID-19 DATASET (4.1.1) pre-processed.

experiment 6 In this experiment, we train our model on a basis that we have created (see experience 2). Then we performed a test on a large image base never seen by the model to check if the algorithm manages to generalize.

4.2.3 Using Images + Metadata

experiment 7 The parallel model has been used with the COVID-19 DATASET (4.1.1) images that have been treated (masked + egalized) since it's the only one that has metadata.

experiment 8 The architecture with QR-Code (19)is used with the COVID-19 DATASET (4.1.1) images which have been treated (masked + egalised).

4.3 Results

In this section we present the results obtained for each architecture with its experiments, we start with the evolution curves of trainning and validation accuracy and loss.Then we will continue with figures of confusion matrices of all experiments.Finally a table summarizing all the results.

4.3.1 TRANSFERT LEARNING

For all the experiments performed, we use Early stopping and Reduce learning rate on plateau :

Early stopping stops training when loss of one model has stopped improving or when generalization error begins to degrade.

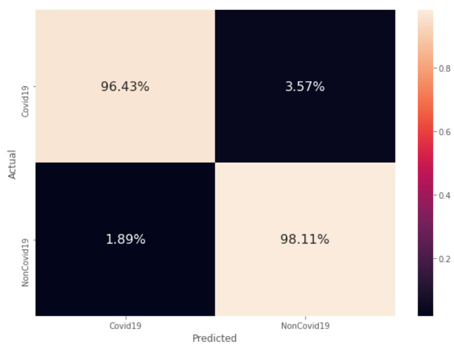
The reduction of learning rate is made by a callback witch monitors the val_loss and if no improvement is seen for a patience of 5 epochs, the learning rate is reduced by 1/2.

experiment 1

VGG16 for this experiment, we can see that for the training base the accuracy increases the loss decreases, progressively according to the epochs. Figure 23

For the test base, the curves oscillate around 90 % for accuracy and 20 % for loss. Figure 24

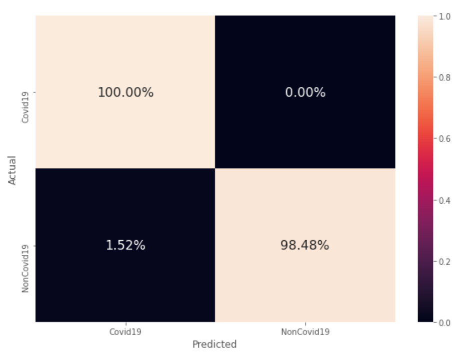
We also note that rate learning has been reduced twice. Figure 25



we can see in the confusion matrix that the results are very satisfactory with 4 errors on 108 so 96 % for the covid classification and 5 errors on 259 so 98 % for the non-covid classification. figure 1

Figure 1: confusion matrice for test data experiment1 VGG16

VGG19

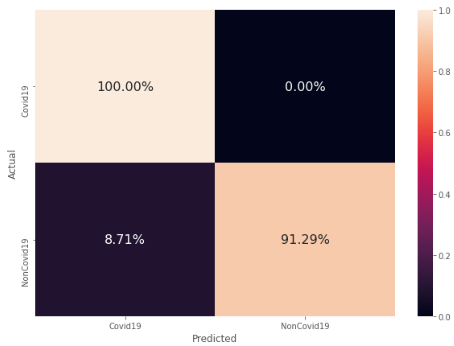


we can see in the confusion matrix that the results are very satisfactory better then VGG16 with 0 error on 112 so 100 % for the covid classification and 4 errors on 260 so 98 % for the non-covid classification.2

Figure 2: confusion matrice for test data experiment1 VGG19

Resnet50 For this experiment, we can see that both accuracies of training and test data increase but the values of the test data loss are important which means that we are in over-fitting. Figures 26 and 27

we also note that rate learning has been reduced once. figure 28



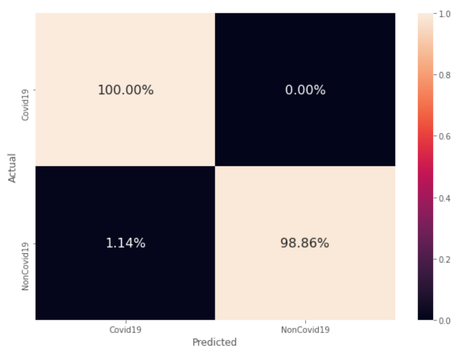
we can see in the confusion matrix that the results are less satisfactory with 81 % for the covid classification and 99 % for the non covid classification. figure 3

Figure 3: confusion matrice for test data experiment1 RESENET50

experiment 2

For this experiment, the curves remain faithful to those of experiment 1 however the accuracy curve of the validation base is sandier in this case. figures 29 and 30

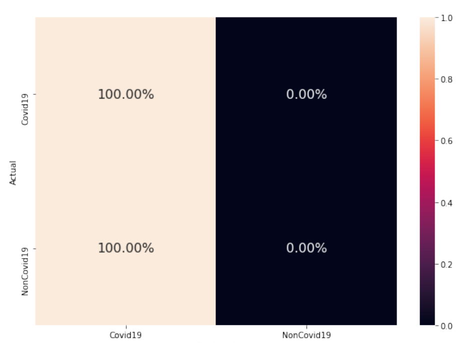
the rate was also decreased twice. figure 31



we can see in the confusion matrix that the results are more satisfactory than in experiment 1 for VGG16 with 100 % for the covid classification and 99 % for the non covid classification. figure 4

Figure 4: confusion matrice for test data experiment 2 VGG16

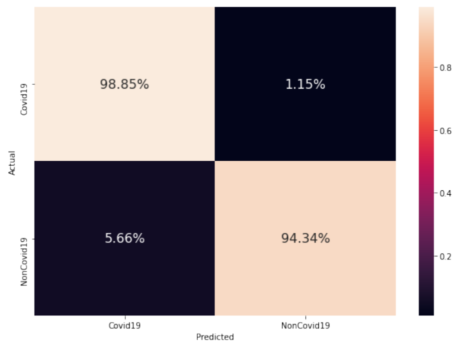
experiment 3



we can see in the confusion matrix that the algorithm classifies all the data as a single class. figure 5

Figure 5: confusion matrice for test data of masked lungs experiment3 VGG16

experiment 4



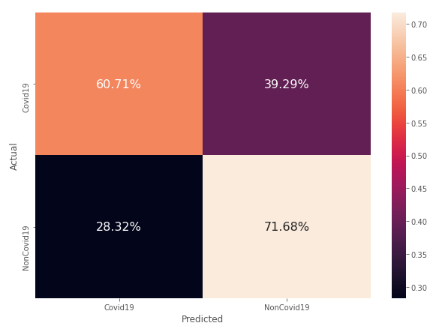
we can see in the confusion matrix that the results are very satisfactory, the algorithm generalizes well with 11 errors on 949 so 99 % for the covid classification and 295 errors on 4921 so 94 % for the non covid classification. figure 6

Figure 6: confusion matrice for big test data experiment4 VGG16

4.3.2 FROM SCRATCH

experiment 5

In this experiment we kept the same model, on the other hand, we took a richer database, and there is a remarkable improvement, the accuracy curve for the training base and validation evolve in the same way, the model generalizes better.The figures of the accuracy curves and the loss are in section 6 figures 32 et 33



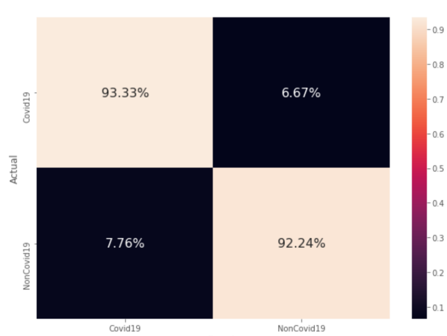
We made a prediction on 25% of data from this database, where we got a accuracy of 68%. The figure below represents the matrix of confusion, it is clear that it classifies more Non-Covid images being than Covid.

Figure 7: confusion matrix for experiment 5

experiment 6

In this experiment we kept the same model, on the other hand we took a richer database, and there is a marked improvement, the accuracy curve for the train base and validation evolve in the same way, the model generalizes better. The figures of the accuracy curves and the loss are in section 6 figures 34 and 35

In order to confirm the performance of our model, we predicted on a large test basis (see experience 4) of data never seen by our model whose confusion matrix is presented below.



confusion matrix is satisfactory, we have reached an accuracy of 93%on the test basis, which confirms that our model generalizes very well.

Figure 8: confusion matrix for big test data experiment 6

4.3.3 Metadata + Images

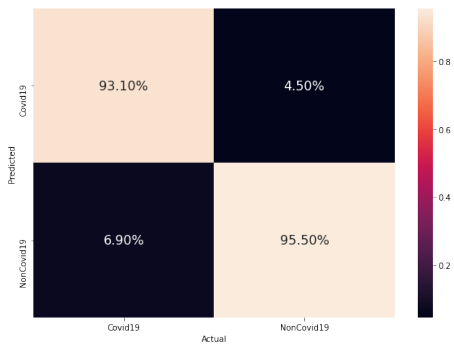


Figure 9: confusion matrix for experiment 7 : Images + Metadata + Parallel Model on validation data

In this case we have really good results using the supplementary metadata information

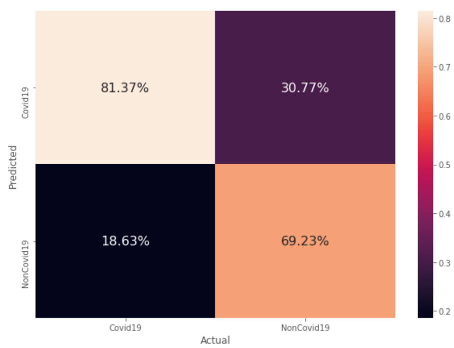


Figure 10: confusion matrix for experiment 8 : Images + QR-Metadata + VGG on validation data

We also have good results using the QR-Code alongside with the VGG16 but not as much as the parallel model above.

4.4 Discussion

Architecture	Accuracy	Sensitivity	specificity
Resnet50	93.88 %	100 %	91.29 %
VGG16	97.61 %	96.43 %	98.11 %
VGG19	98.94 %	100 %	98.48 %

Table 1: Results on test set

Architecture	Accuracy	Sensitivity	specificity
VGG16	99.20 %	100 %	98.86 %
VGG19	97.87 %	100 %	96.97 %
From Scratch	94.59 %	86.36 %	98.08 %

Table 2: Results on pre-treated test set

Architecture	Accuracy	Sensitivity	specificity
Resnet50	86.46 %	98.75 %	84.20 %
VGG16	97.30 %	97.92 %	97.18 %
VGG19	98.14 %	98.23 %	98.12 %

Table 3: Results On test BigDataset

Architecture	Accuracy	Sensitivity	specificity
VGG16	95.05 %	98.85 %	94.34 %
VGG19	93.36 %	98.65 %	92.39 %
From Scratch	92.5 %	93.33 %	92.24 %

Table 4: Results on test BigDataset Pre-treated

Architecture	Accuracy	Sensitivity	specificity
From Scratch	68.25 %	75.38 %	65.07 %
VGG16+RNN metadata	94.08 %	93.1 %	94.59 %
VGG16 + QR metadata	77.27 %	81.37 %	69.23 %

Table 5: Results on only Dataset Pre-treated

- Which learning transfer architecture is best ?

  experiment 1 allows us to compare between the VGG16, VGG19 and Resnet50 architectures, by comparing the accuracy results and the confusion matrix, the resenet50 is not very adapted to this problem with less satisfactory results, however between the VGGs, the VGG19 is deeper which explains its better results compared to VGG16. table 1

- Pre-processed data or not?

  comparing tables 1 and 2 on one side and tables 3 and 4 on the other side, VGG16 improves with pre-processed data, however the results for VGG19 remain similar in both tables.

- Lung criteria or bias criteria ?

  experiment 3 shows that by masking the lungs and testing a trained model on non-preprocessed images, the algorithm classifies them as a single class, so the algorithm does not rely on bias criteria.

- The Scratch Architecture ?

  First, we tested our architecture from scratch on the database provided by CovidDataSet (section 4.1.1). We took 20% of this database for testing and 80% for learning. The results were not satisfactory, we reached only 68% accuracy on the test basis -table 5-, we deduce that the model fails to generalize. It is suspected that the poor results obtained are due to the use of a small database. In order to verify this, we trained our model on a large database, then we made the prediction on a database where we had an accuracy of 94.59% table -2-, and an accuracy of 92.59% on the BigDataSet -table 4-, the model manages to generalize well. We confirm our assumption, and we conclude that even the "From scratch" give similar results to VGG16 and VGG19, provided it is trained on a large database.

- The use of metadata (clinical history of the patient) ?

  this can be a good idea on poor databases (not a lot of images), however we notice that there are indications on the nature of the radiography in some clinical notes (Coronavirus, covid, sars ...), people's names are also repeated, therefore the good recognition rate is due to this kind of data. Moreover, processing these data in an RNN is better than in a CNN (pasting datas as QR on the X-ray image) because the RNN with its bidirectional LSTM layers manages to make links between words, which is not the case in the CNN (absence of temporal dependency) Table 5. Finally, these results show the power of combining X-rays with diagnostics but are not significant in this problem due to the presence of obvious tips in the clinical notes that help the network.

5 Conclusion

Detecting covid from X-rays using deep learning is one of the most recent challenges. Some researchers have highlighted the possibility of successfully addressing this problem, despite the current small amount of data. In our solutions, we have remedied this problem by exploiting the metadata that allows improvements in results, we have also added pneumonia x-rays that are present enough, all its X-ray images have been pre-processed. Thanks to all these pre-processing and with the different architectures proposed, we have achieved very good results with different architectures proposed. A little error is seen in the detection of X-rays of patients with Covid, the only solution is to collect more Covid X-ray data in the hope of helping to fight the pandemic.

References

[1] https://eurorad.org.

[2] https://github.com/armiro/COVID-CXNet.

[3] https://github.com/ieee8023/covid-chestxray-dataset.

[4] https://github.com/ml-workgroup/covid-19-image-repository/tree/master/png.

[5] https://sirm.org/category/senza-categoria/covid-19/.

[6] https://www.kaggle.com/ankasor/improved-lung-segmentation-using-watershed.

[7] https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia/version/2.

[8] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

[9] https://www.radiologymasterclass.co.uk/tutorials/chest/chest_quality/chest_xray_quality_projection.

[10] Ioannis D. Apostolopoulos and Tzani A. Mpesiana. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2):635–640, Apr 2020.

[11] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection, 2020.

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

[13] Gianluca Maguolo and Loris Nanni. A critic evaluation of methods for covid-19 automatic detection from x-ray images, 2020.

[14] Kaylie Zhu Brandon Yang Hershel Mehta Tony Duan Daisy Ding Aarti Bagul Curtis Langlotz Katie Shpanskaya Matthew P. Lungren Andrew Y. Ng Pranav Rajpurkar, Jeremy Irvin. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. ArXiv Prepr. ArXiv1711.05225, 2017. https://arxiv.org/abs/1711.05225.

[15] Raghavendra Selvan, Erik B. Dam, Nicki Skafte Detlefsen, Sofus Rischel, Kaining Sheng, Mads Nielsen, and Akshay Pai. Lung segmentation from chest x-rays using variational data imputation. ICML Workshop on The Art of Learning with Missing Values, July 2020. arXiv preprint arXiv:2020.2005.10052.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[17] Enzo Tartaglione, Carlo Alberto Barbano, Claudio Berzovini, Marco Calandri, and Marco Grangetto. Unveiling covid-19 from chest x-ray with deep learning: A hurdles race with small data. *International Journal of Environmental Research and Public Health*, 17(18):6933, Sep 2020.

[18] Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images, 2020.

[19] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition, 2016.

6 appendix

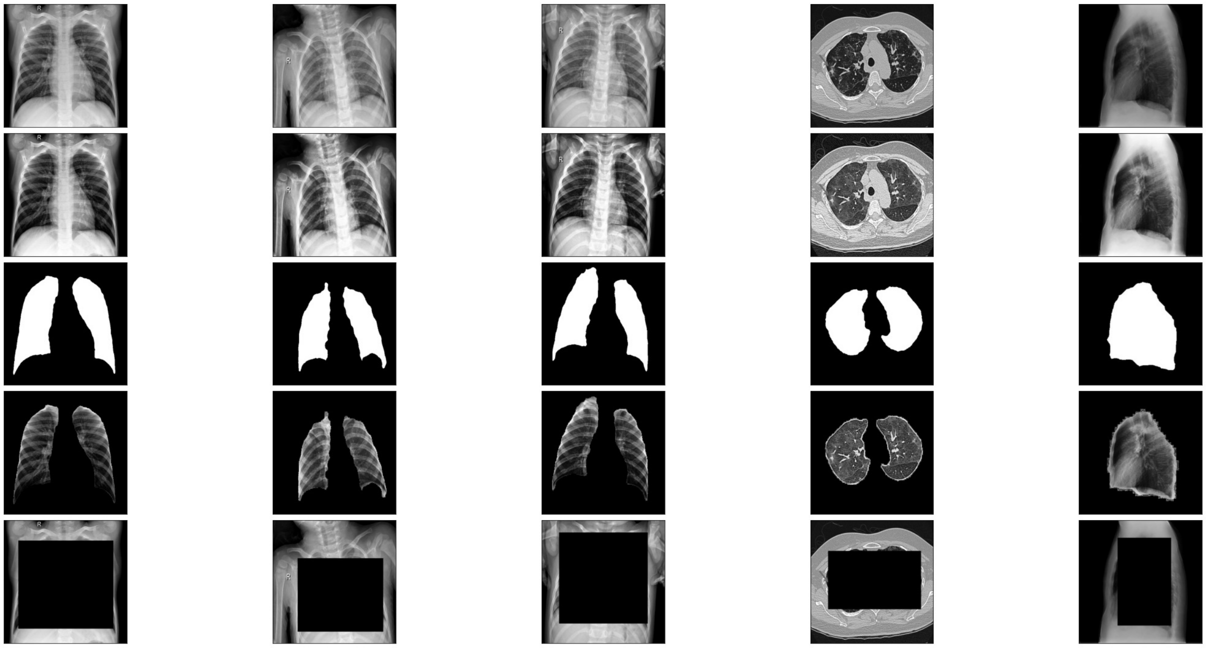


Figure 11: summary of pre-treatment steps
the first 3 columns are frontal chest X-ray then axial by CT scans and the last one is the lateral view of the chest X-ray
The first line is the basic images and the penultimate line is the result of the pre-processing.

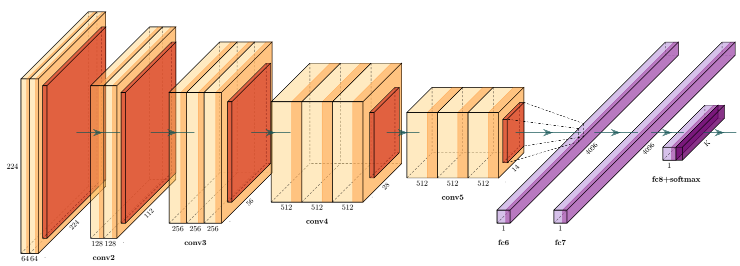


Figure 12: VGG16 Neural Network Architecture

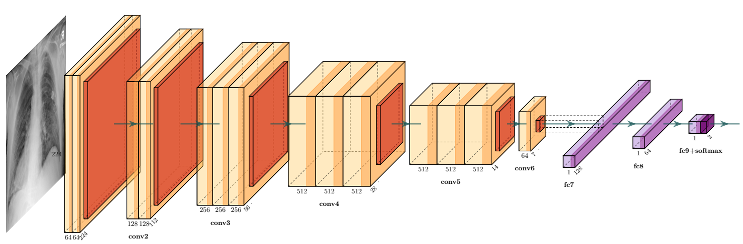


Figure 13: New VGG16 Neural Network Architecture

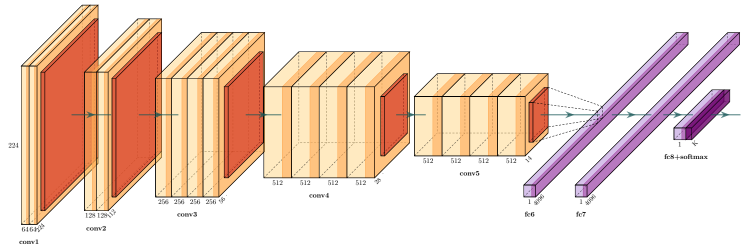


Figure 14: VGG19 Neural Network Architecture

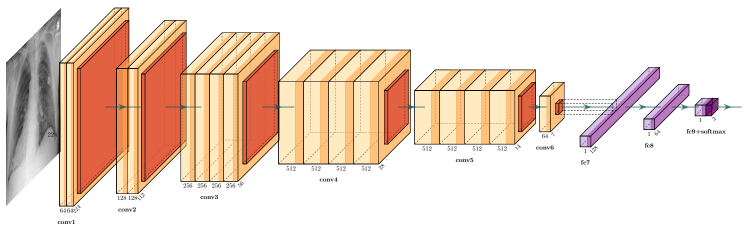


Figure 15: New VGG19 Neural Network Architecture

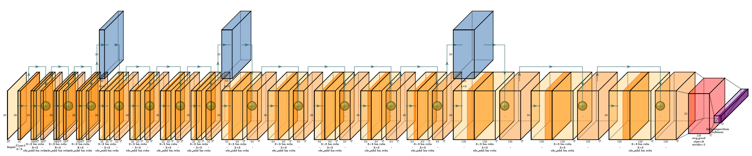


Figure 16: Resnet50 Neural Network Architecture

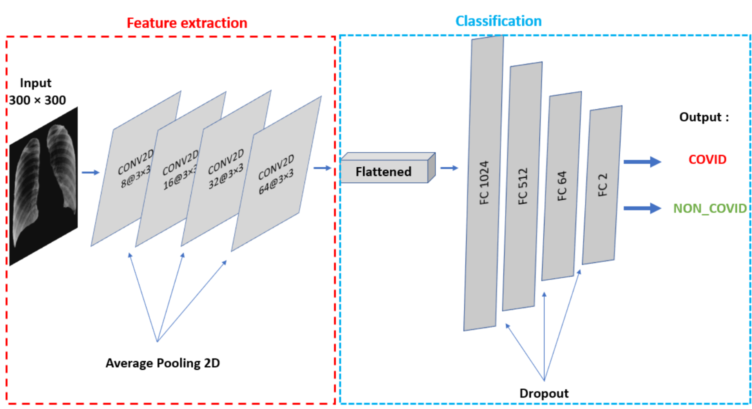


Figure 17: CNN Architecture

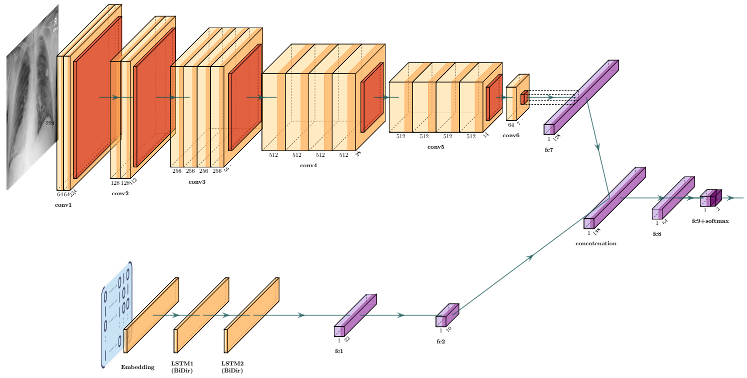


Figure 18: Parallel networks VGG19 + RNN

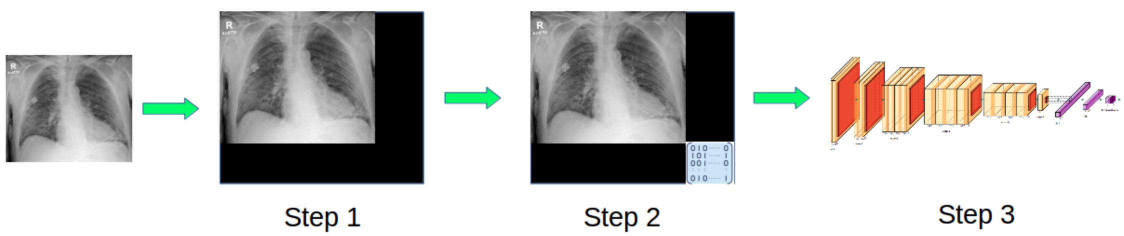


Figure 19: Creating a QR code on the image

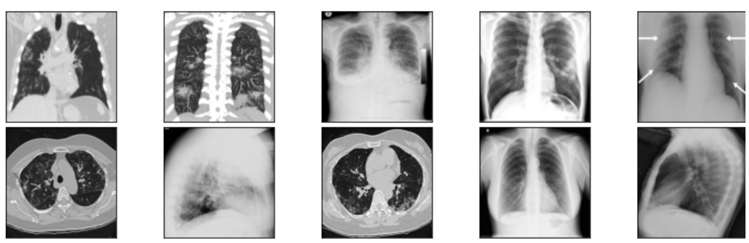


Figure 20: Samples from COVID-19 dataset

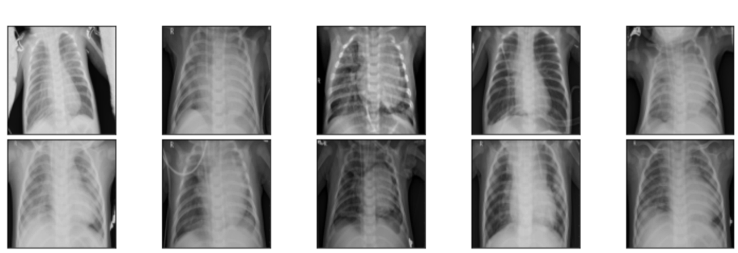


Figure 21: Samples from Pneumonia Kaggle dataset

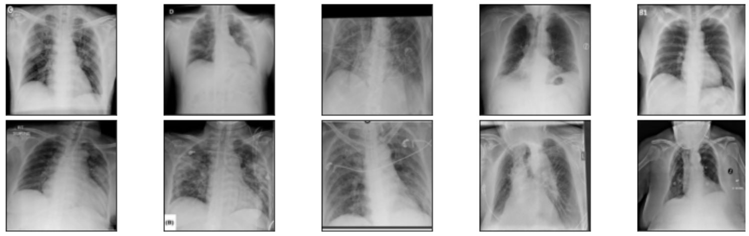


Figure 22: Samples from Pneumonia COVID-19 Radiography database

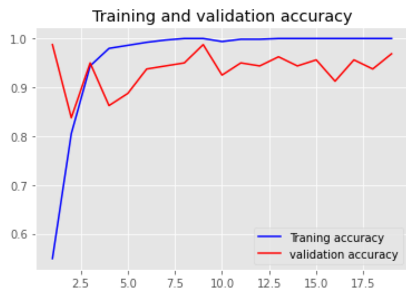


Figure 23: training and validation accuracy experiment1 VGG16

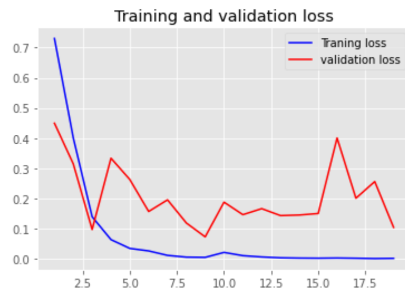


Figure 24: training and validation loss experiment1 VGG16

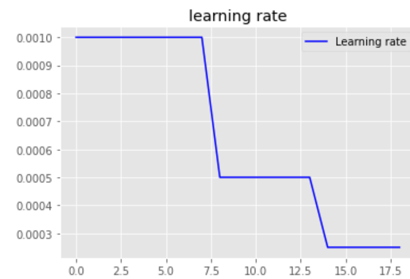


Figure 25: learning rate experiment1 VGG16

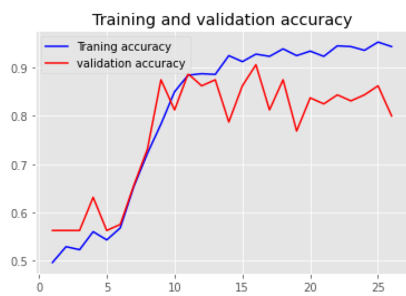


Figure 26: training and validation accuracy experiment1 Resent50

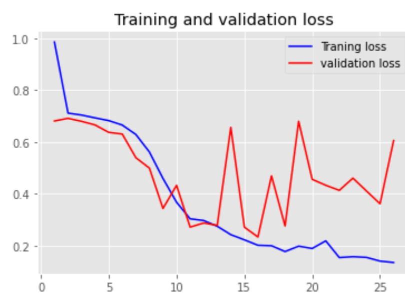


Figure 27: training and validation loss experiment1 RESENT50

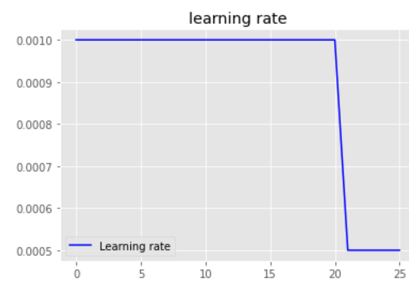


Figure 28: learning rate experiment1 RESENET50

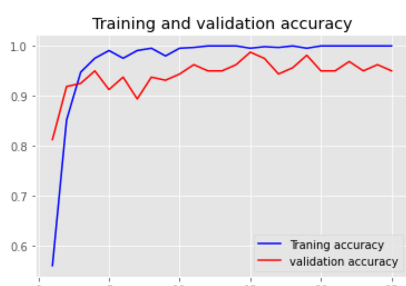


Figure 29: training and validation accuracy experiment2 VGG16

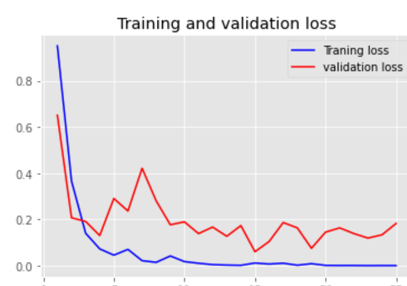


Figure 30: training and validation loss experiment2 VGG16

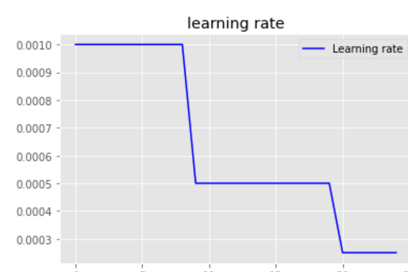


Figure 31: learning rate experiment2 VGG16

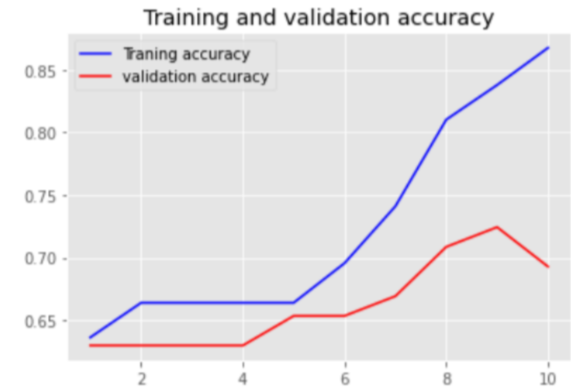


Figure 32: training and validation accuracy experiment 5

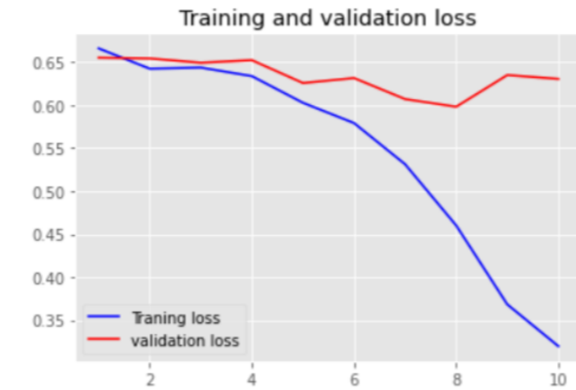


Figure 33: training and validation loss experiment 5

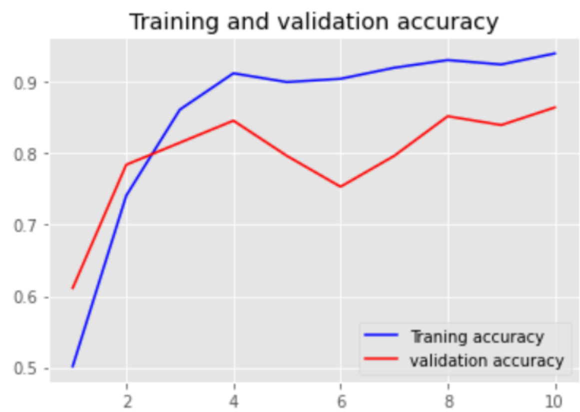


Figure 34: training and validation accuracy experiment 6

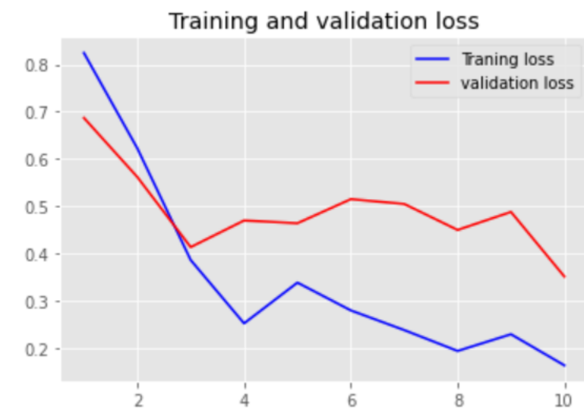


Figure 35: training and validation loss experiment 6