

Article

COVID-19 classification on chest X-ray images using Deep Learning Methods

Marios Constantinou¹, Themis Exarchos^{1*}, Aristidis G. Vrahatis¹, Panagiotis Vlamos¹

¹ Bioinformatics and Human Electrophysiology Laboratory, Dept of Informatics, Ionian University, Corfu, 49132, Greece

* Correspondence: Themis Exarchos, exarchos@ionio.gr

Abstract: Coronavirus disease since December 2019 has significantly affected millions of people. Given the effect this disease has on the pulmonary systems of humans, there is a need for chest radiographic imaging (CXR) for monitoring the disease and preventing further deaths. Several studies have been shown that Deep Learning models can achieve promising results for COVID-19 diagnosis towards the CXR perspective. However, this research field is at an initial stage since there is a limited number of large CXR repositories regarding COVID-19. In this study, five deep learning models were analyzed and evaluated with the aim of identifying COVID-19 from chest X-Ray images. More specifically, we utilized the ResNet50, ResNet101, DenseNet121, DenseNet169 and InceptionV3 using Transfer Learning. All models were trained and validated on the largest publicly available repository for COVID-19 CXR images. Furthermore, they were evaluated on unknown data as well, that was not used for training or validation, authenticating their performance, and clarifying their usage in a medical scenario. All models achieved satisfactory performance where ResNet101 was the superior model achieving 96% in Precision, Recall, and Accuracy. Our outcomes show the potential of deep learning models on COVID-19 medical offering a promising way for the deeper understanding of COVID-19.

Keywords: Deep Learning; COVID-19; ResNet50; ResNet101; DenseNet121; DenseNet169; InceptionV3; Transfer Learning; Chest X-Rays

1. Introduction

In December 2019, the first case of Coronavirus 2019 (COVID019) was reported in Wuhan, China. Until now, the virus affected millions of people, showing almost 630 million cases and 6,5 million deaths worldwide [1]. The most common symptoms of COVID-19 are fever, cough, fatigue, headache, dizziness, sputum, and dyspnea. Consequently, some patients sustained further damage to their respiratory system; specifically, lesions were detected in the lower lobes of both lungs. Severe cases of COVID-19 can result in acute respiratory distress syndrome or complete respiratory failure [2].

Given the solemnity of COVID-19, reliable and swift diagnosis is extremely important. There have been numerous methods for the detection of COVID-19. The primary method is reverse-transcription polymerase chain reaction (RT-PCR) [3]. These tests suffer from high false-positives or false-negatives due to sample contamination, virus mutations or user error during sample extraction [4]. As a result, several studies [5], [6] suggested on using Computed Tomography (CT-Scans) for performing diagnosis, since it showed higher accuracy. Consequently, it was shown that the majority of COVID-19 cases share similar radiographic features, such as bilateral abnormalities and multifocal ground-glass opacities, mostly at the lower lung lobes during the early stages for the disease and at the final stages pulmonary consolidation was observed [7]. However, compared to CT-Scans, chest x-rays are cheaper and faster in image generation, furthermore it is an accessible method for medical imaging and the body gets exposed to less radiation during the procedure [8]. Chest x-rays are already used as a diagnostic tool against COVID-19 [9].

There has been an immense growth in Machine Learning the past few years. Specifically, in medicine, it's used for various tasks, such as classification of cardiovascular diseases, diabetic retinopathy and other [10]–[12]. Respectively, the revolutionary performance of the convolutional neural network (CNN), has enabled medical experts to use it on many tasks such diagnosis of skin lesion, detection of brain tumor and breast cancer [13]–[15].

Applying Deep Learning models on chest x-ray (CXR) images has proven beneficial where various researchers showed auspicious results in the diagnosis of pulmonary diseases including COVID-19 pneumonia. Notably, Rajpurkar et al. [16] developed a new CNN architecture called CheXNet based on DenseNet121 for the classification of 14 different pulmonary diseases by training it on over 100,000 x-ray images. They reported that their method exceeds average radiologist performance on the F1 metric. Similarly, in [17] the authors proposed a method for automatic detection of COVID-19 pneumonia from CXR images using pre-trained convolutional neural networks, reaching accuracy ~99%. In addition, Keidar et al. [18] proposed a deep learning model for the detection of COVID-19 from CXR images and clustering of similar images to the model's result. Lastly, in [19] a method for the detection of COVID-19 is proposed using various Deep Learning models and a support vector machine (SVM) as a classifier.

Correspondingly, additional studies proposed methods for the automatic diagnosis of COVID-19, from CXR images using Deep Learning [20]–[23]. Their methods revealed high performance in detecting COVID-19; although, they possess a few flaws. Foremost, all the mentioned studies had finite number of COVID-19 CXR images. This can affect the training and evaluation performance of these methods, resulting in improper generalization for future data. In addition, they didn't use external unseen data for evaluation of their methods.

The goal of this study is the comparative evaluation of Deep Learning methods on COVID-19 CXR image classification and their potential to be used as decision-making tools for COVID-19 diagnosis. Our analysis is performed using five deep learning models covering various state-of-the-art architectures. We also applied all models in the largest (by the time of writing and to the best of our knowledge) dataset [24].

2. Materials and Methods

The COVID-QU dataset [24] is used for this study and it consists of 33,920 CXR images from 3 different classes. More specifically, COVID-19 contains 11,956 images of coronavirus positive patients, non-COVID19 contains 11,263 images of viral or bacterial pneumonia patients and lastly, Normal contains 10,701 healthy images. Moreover, COVID-QU contains only posterior to anterior (PA) and anterior to posterior (AP) x-ray images. Lastly, although this dataset contains the corresponding lung masks of each image, they were not used for this study. The sources that were used for the compilation of this dataset are found below in detail:

1. COVID-19 CXR dataset

This dataset consists of 11,956 COVID-19 positive x-ray images. To compile this dataset, various sources were accessed. Specifically, 10,814 images were taken from the BIMCV-COVID19+ [25] database, then 183 images were taken from a German medical school [26], 559 images were taken from SIRM [27], GitHub [28], Kaggle [29] and Eurorad [30]. Lastly, 400 images were taken from another COVID-19 repository [31].

2. RSNA CXR dataset

This dataset consists of 8,851 healthy and 6,012 lung opacity x-ray images from the RSNA CXR [32] repository, where the lung opacity images belong in the non-COVID19 class of the COVID-QU dataset.

3. Chest X-Ray Pneumonia dataset

The Chest X-Ray Pneumonia [33] dataset was used to access 1,300 viral pneumonia, 1,700 bacterial pneumonia and 1,000 healthy X-Ray images. Viral and bacterial pneumonia images belong to the non-COVID19 class of the COVID-QU dataset.

4. PadChest dataset

From the PadChest [34] dataset, 4,000 healthy and 4,000 pneumonia x-ray images were used. The 4,000 pneumonia images belong to the non-COVID19 class of the COVID-QU dataset.

5. Montgomery and Shenzhen CXR lung masks datasets

The Montgomery dataset [35] consists of 80 healthy and 58 tuberculosis x-ray images along with their lung masks, and the Shenzhen dataset [36] consists of 326 normal and 336 tuberculosis x-ray images where 566 of the total 662 images are accompanied by their lung masks.

6. QaTa-Cov19 CXR infection mask dataset

The QaTa-Cov19 [37] dataset consists of almost 120,000 CXR images with their ground-truth infection masks. The researchers who created COVID-QU used these masks to train and evaluate their segmentation models that generated the rest of the segmentation masks.

Table 1 presents the distribution of data across 3 subsets grouped into 3 classes. In detail, the train subset consists of 21,715 CXR images, split into: COVID-19 with 7,658 images, non-COVID19 with 7,208 images, and Normal with 6,849 images. Furthermore, the validation set consists of 5,417 CXR images, split into: COVID-19 with 1,903 images, non-COVID19 with 1,802 images, and Normal with 1,712 images. Lastly, the test set consists of 6,788 CXR images, split into: COVID-19 with 2,395 images, non-COVID19 with 2,253 images, and Normal with 2,140 images.

Table 1. Overall distribution of the data used in this study.

#	Subset	COVID-19	Non-COVID19	Normal	Total
1	Train	7,658 – 35%	7,208 – 33%	6,849 – 32%	21,715 – 64%
2	Validation	1,903 – 35%	1,802 – 33%	1,712 – 32%	5,417 – 16%
3	Test	2,395 – 35%	2,253 – 33%	2,140 – 32%	6,788 – 20%

The proposed approach for this study is demonstrated on COVID-19 classification from CXR images. In *Figure 1*, the general pipeline for the classification system is shown, where the first step is the configuration of the dataset into 3 subsets i.e., train, validation, and test sets. Step 2 consists of defining the model and all its functions where data is loaded, augmented, and pre-processed, and all the layers are frozen except the classifier. Afterwards, metrics, optimizer and callbacks are defined, and the model gets compiled. Step 3 consists of training only the classifier using the pre-trained weights of each model respectively. Thereafter in step 4, fine-tuning is performed where a specific number of layers are unfrozen, and the models are trained again. Lastly, in step 5 the models are evaluated on the test set.

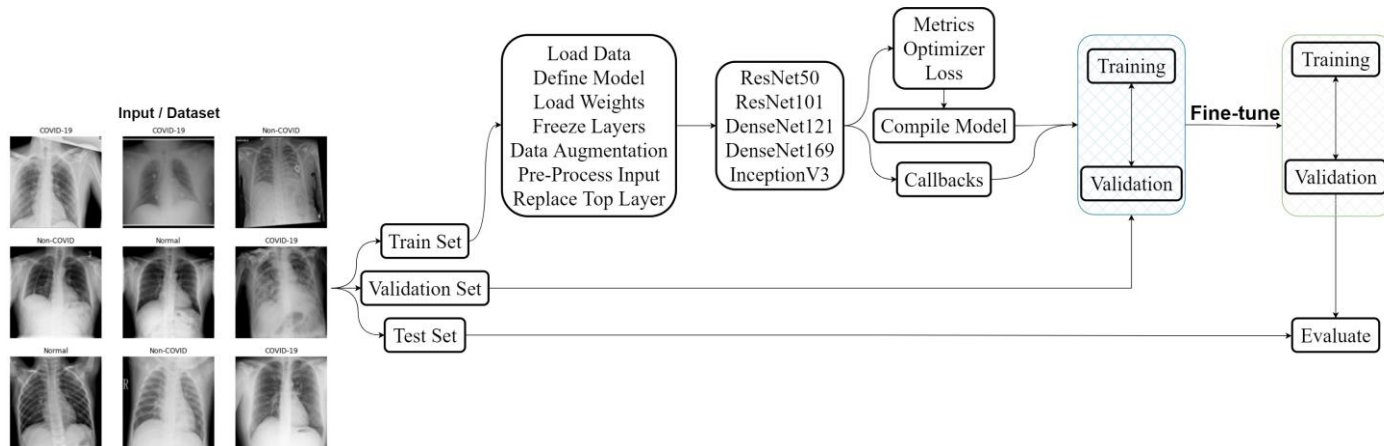


Figure 1. The proposed pipeline of this study.

Setup and Tools

The programming language that was used for the project is Python 3.10.2 in combination with Visual Studio Code as a code editor. Furthermore, regarding software version control, GitHub along with SourceTree was used. Tensorflow and Keras are used for the creation and training of these models. Training was performed on a personal computer with the following specs: AMD Ryzen 5600X, 16GB Ram 3200mhz, an RX Vega 64 8 and Windows 10. Since the graphics card is not compatible with Tensorflow, the training process was performed on the CPU.

Models and architectures for COVID-19 Classification

Regarding COVID-19 classification, 5 state-of-the-art Convolutional Neural Networks (CNNs) were evaluated on COVID-19 classification from CXR images: 2 variants based on the ResNet [38] architecture; ResNet50, and ResNet101, then 2 based on the DenseNet [39] architecture; DenseNet121, and DenseNet169, and lastly one based on the InceptionV3 [40] architecture. All models were pre-trained on the ImageNet dataset that consists of 1000 classes and millions of images.

ResNet - Residual Network

The ResNet - Residual Network architecture [38] was proposed as a solution to the vanishing/exploding gradients problem that deep neural networks suffer. Respectively, this architecture consists mostly of residual blocks as show in Figure 2, and batch normalization layers, where each residual block contains convolution layers and shortcut connections.

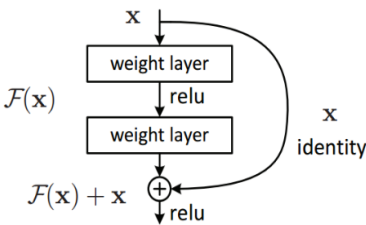


Figure 2. A residual block of ResNet. Source: *Deep Residual Learning for Image Recognition* [38].

DenseNet

The DenseNet architecture [39] as show in Figure 3, was introduced by G. Huang et al., in 2018 where each layer is connected to every other layer in a feed-forward manner. Furthermore, for each layer, the feature maps of all former layers are used as inputs and its own feature maps are used as inputs for the succeeding layers. Lastly, DenseNet solves the problem of vanishing gradients and reduces the number of parameters considerably.

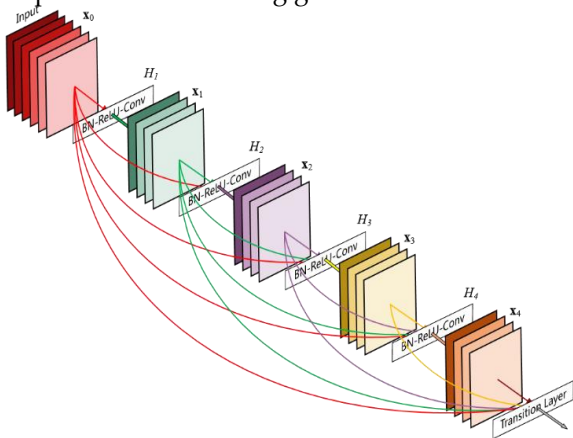


Figure 3. An overview of the DenseNet Architecture. Source: *Densely Connected Convolutional Networks* [39].

InceptionV3

InceptionV3 [40] was introduced by Szegedy et al., in 2015. The fundamental characteristic of this network is the Inception Module as shown in Figure 4. This module consists of convolutions in various sizes such as 1x1, 3x3, and 5x5. Lastly, a pooling and concatenation layer is included.

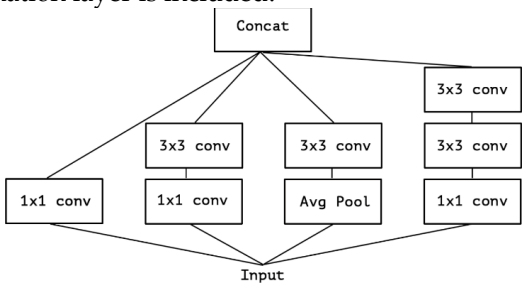


Figure 4. An inception block of InceptionV3 architecture. Source: *Learning Robust Scene Classification Model with Data Augmentation Based on Xception* [41].

Image pre-processing

We utilized data augmentation methods, such as *random rotation* ($\pm 10^\circ$) and *random horizontal flip* (Figure 5), to deal with overfitting issues. The augmentations were applied randomly on each image, meaning that some images will only be rotated, flipped horizontally or both, as it is shown on Figure 5. These methods were applied on each image during model training on the training set and not before, leaving the original dataset intact without changes. Each architecture requires a specific image size, therefore, all images were resized to 224x224 and for ResNet and DenseNet models using bilinear interpolation. On the other hand, InceptionV3 can work with various sizes, therefore, no resizing was needed.



Figure 5: The figure above is an example of the augmentations that is performed on each image. Image A is flipped and rotated, and image B is only rotated slightly.

Model Definition

As previously mentioned, five models are trained and evaluated on CXR images. Ergo, a template was created and used for all models with only a few changes in each instance. Foremost, the base model is defined with the pre-trained weights of ImageNet and without the included classifier since a custom one is added later. Following, all layers of the base model were frozen. The model's input is defined, then data augmentation is applied and lastly, it's pre-processed where the values of the input image are normalized to 0 and 1 or -1 and 1, depending on the architecture.

The last step is to define the new classifier. In detail, the classifier consists of 3 layers. The first one is a Global Average Pooling layer, or in the case of InceptionV3 a flatten layer, followed by a Dropout layer with a factor of 0.2, and lastly, a 3-unit Dense layer with the softmax activation function show in equation (1) and the HeNormal kernel initializer.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (1)$$

Evaluation Metrics and Callbacks

Several metrics were used to monitor the performance of each model. Specifically, *Categorical Accuracy*, *Precision*, *Recall* and *F1-Score* as shown in Equations 2-5, along with True Positives, True Negatives, False Positives and False Negatives. Regarding the optimization method, Adam was used with an initial learning rate of $4e^{-3}$, 0.9 for beta 1, 0.999 for beta 2, and $1e^{-07}$ for epsilon. Lastly, categorical cross entropy was used as a loss function as shown in Equation 6.

$$\text{Categorical Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

$$\text{F1} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

$$\text{Loss} = - \sum_{i=1}^N y_i * \log \hat{y}_i \quad (6)$$

Categorical Accuracy represents the number of correct predictions divided by the total number of predictions. Precision represents the ratio of correctly classified positive samples to the total number of classified positive samples. Recall is the ratio between the numbers of positive samples correctly classified as positive to the total number of positive samples. In this study, *Recall* was the primary metric.

Callbacks

The last task before the initial training of each model is to define all the required callbacks. In this study, the callbacks *Model Checkpoint*, *Early Stopping*, *Reduce Learning Rate on Plateau*, *Tensorboard* and *CSVLogger* were used. In detail, *model checkpoint* was setup to save only the weights of each model, *Early Stopping* was setup with an 8-epoch patience and to restore the model's best weights. Afterwards, *Reduce Learning Rate on Plateau* was setup to reduce the learning rate by a factor of 0.2 as shown in Equation 7 with a 3-epoch patience.

$$\text{new lr} = \text{initial_lr} * \text{factor} \quad (7)$$

Regarding model visualization, *Tensorboard* was used to monitor the training performance of each model.

Model Training and fine-tuning

After every function, parameter and callback has been setup, the initial training can commence where all the layers are frozen except the classifier. All models were set to be trained for 100 epochs. Consequently, none of them were trained for 100 epochs, because the callback *Early Stopping* ends their training if no improvement in performance is observed. Following the initial model training, the fine-tuning phase takes place where some layers of each model are unfrozen and are trained again. Table 2 shows in detail the number of parameters of each model after layer unfreezing.

Table 2. The total parameters of each model along with the trainable and non-trainable parameters after unfreezing some layers

Parameters	ResNet50	ResNet101	DenseNet121	DenseNet169	InceptionV3
Total	23,564,800	42,632,707	7,040,579	12,647,875	22,023,971
Trainable	14,970,880	25,040,899	5,527,299	11,059,843	17,588,163
Non-Trainable	8,593,920	17,591,80	1,513,280	1,588,032	4,435,808

Once the layers are unfrozen, the model is trained for around 10-15 epochs with the same callbacks, loss function and metrics. The only difference is in the optimizer function; although Adam was used during fine-tuning, the learning was set to $4e^{-4}$.

3. Results

In this chapter, the training and evaluation performance is demonstrated and compared across all models. The following tables show the metrics that were discussed above with the addition of the Support column where it shows the number of samples for each class. It can be observed that all 3 classes had a similar number of samples, therefore eliminating the problem of class imbalance.

ResNet50

Table 3 shows that ResNet50 managed to achieve 97% Precision, Recall and F1-Score regarding class COVID-19. Although, its performance drops significantly for the classes non-COVID19 and Normal. Overall, its Recall reached 95%.

Table 3. ResNet50 Evaluation results

	Precision	Recall	F1-score	Support
COVID-19	0.97	0.97	0.97	2395
Non-COVID19	0.95	0.94	0.94	2253
Normal	0.94	0.94	0.94	2140
Accuracy			0.95	6788
Macro avg	0.95	0.95	0.95	6788
Weighted avg	0.95	0.95	0.95	6788

Furthermore, regarding class COVID-19, ResNet50 performed exceptionally well, as shown in Figure 6. Although, its performance degraded regarding the other 2 classes, with a similar number of errors.

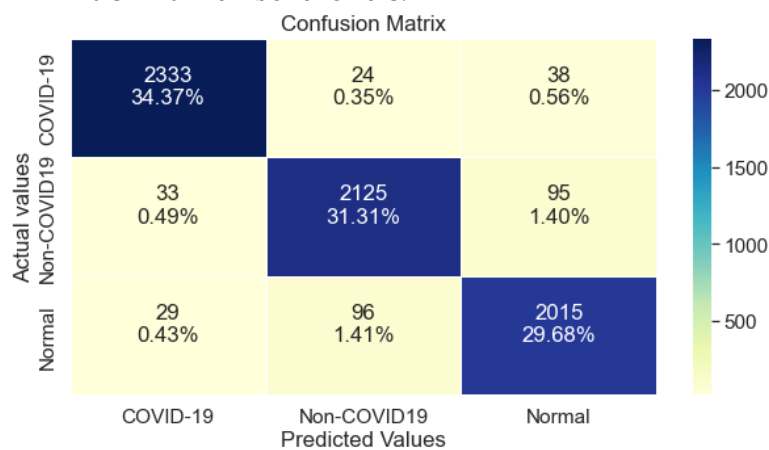


Figure 6. ResNet50 Confusion Matrix.

ResNet101

Furthermore, ResNet101 as shown in Table 4, managed 99% Precision, 96% Recall, and 98% F1- Score regarding class COVID-19. Similar to ResNet50, a drop in performance is observed regarding the classes non-COVID19 and Normal. Lastly, it reached 96% in Recall.

Table 4. Resnet101 Evaluation results

	Precision	Recall	F1-score	Support
COVID-19	0.99	0.96	0.98	2395
Non-COVID19	0.95	0.95	0.95	2253
Normal	0.93	0.95	0.94	2140
Accuracy			0.96	6788
Macro avg	0.96	0.96	0.96	6788
Weighted avg	0.96	0.96	0.96	6788

Regarding the Confusion Matrix that is shown in *Figure 7*, it is clear that compared to ResNet50, ResNet101 performed equally well on class COVID-19 while also maintaining a balanced performance regarding the classes non-COVID19 and Normal.

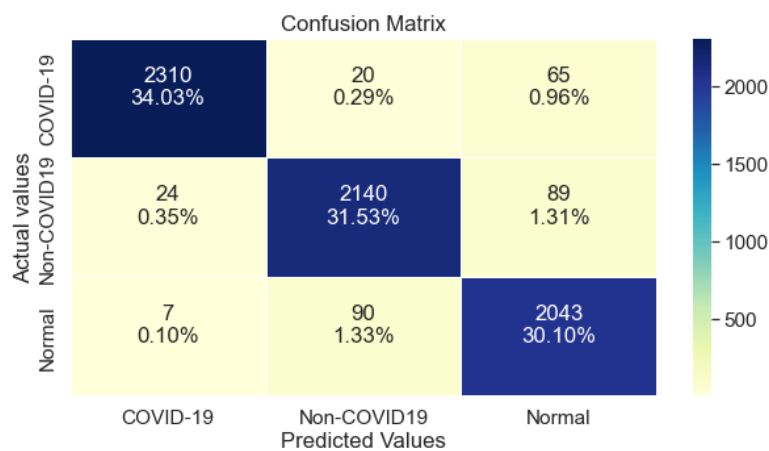


Figure 7. ResNet101 Confusion Matrix

DenseNet121

DenseNet121, as shown in *Table 5*, managed to achieve 99% Precision, 94% Recall and 96% F1-Score regarding the class COVID-19. Furthermore, a significant drop in Precision and Recall is observed for class non-COVID19 and Normal where it achieved 86% and 87% respectively. The achieved Recall for this model is 93%.

Table 5. DenseNet121 Evaluation results

	Precision	Recall	F1-score	Support
COVID-19	0.99	0.94	0.96	2395
Non-COVID19	0.86	0.97	0.91	2253
Normal	0.95	0.87	0.91	2140
Accuracy			0.93	6788
Macro avg	0.93	0.93	0.93	6788
Weighted avg	0.93	0.93	0.93	6788

The confusion matrix shown in *Figure 8*, DenseNet121 made many misclassifications regarding the classes Normal and COVID-19, where the model's prediction classified images, as non-COVID19 in both cases. With reference to *Table 5*, this drop in performance is also shown by the significant drop of Precision and Recall in the classes COVID-19 and Normal respectively.

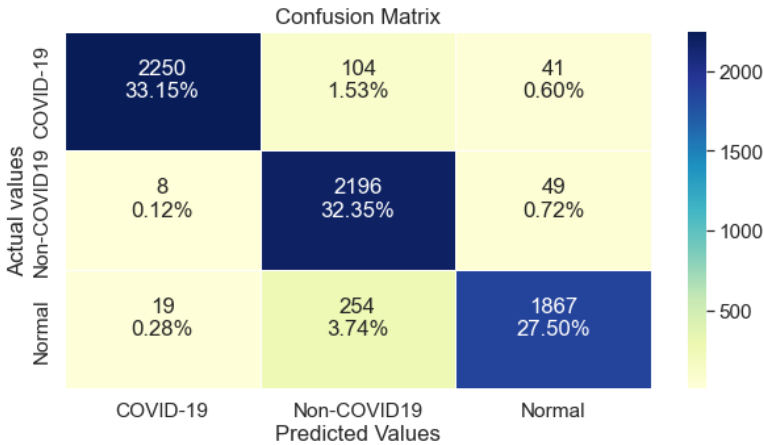


Figure 8. DenseNet121 Confusion Matrix

DenseNet169

DenseNet169 as shown in Table 6, had similar performance with DenseNet121 despite having a larger computational capacity. Regarding the class COVID-19, 99% Precision, 93% Recall and 96% F1-Score were reported. Compared to DenseNet121, it managed to surpass its performance regarding the class non-COVID19 but had a significant drop in its Precision regarding the class Normal. Overall, it's Accuracy reached 94%.

Table 6. DenseNet169 Evaluation results.

	Precision	Recall	F1-score	Support
COVID-19	0.99	0.93	0.96	2395
Non-COVID19	0.95	0.92	0.94	2253
Normal	0.88	0.96	0.92	2140
Accuracy			0.94	6788
Macro avg	0.94	0.94	0.94	6788
Weighted avg	0.94	0.94	0.94	6788

Concerning the Confusion Matrix of DenseNet169, it is evident that misclassifications were made regarding the classes non-COVID19 and COVID-19, where it classified images as Normal although the correct class was either COVID-19 or non-COVID19.

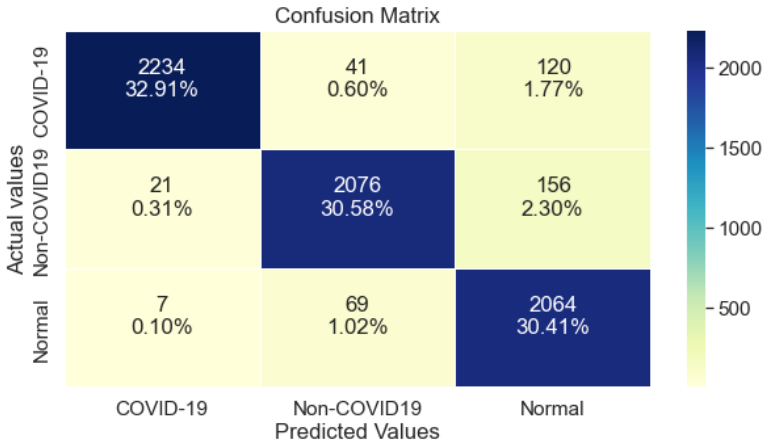


Figure 9. DenseNet169 Confusion matrix.

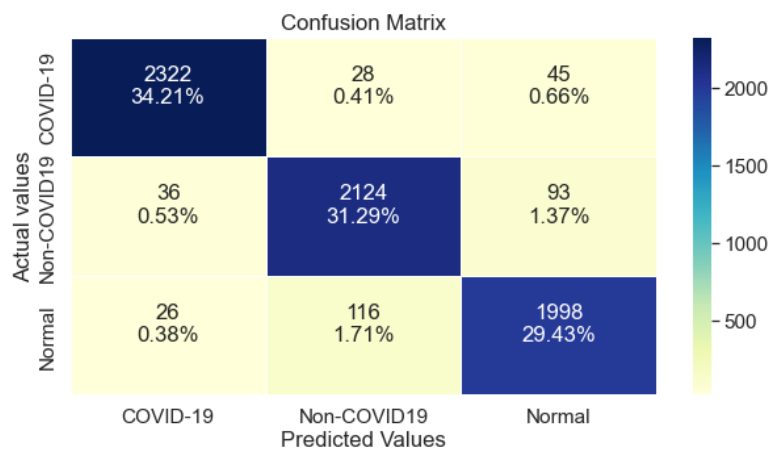
InceptionV3

InceptionV3 as shown in Table 7, managed 97% Precision, 97% Recall and 97% F1-Score regarding the class COVID-19. It's performance on non-COVID19 and Normal is slightly lower but balanced across all metrics. The overall Accuracy of this model is 95%.

Table 7. InceptionV3 Evaluation results.

	Precision	Recall	F1-score	Support
COVID-19	0.97	0.97	0.97	2395
Non-COVID19	0.94	0.94	0.94	2253
Normal	0.94	0.93	0.93	2140
Accuracy			0.95	6788
Macro avg	0.95	0.95	0.95	6788
Weighted avg	0.95	0.95	0.95	6788

With reference to the Confusion Matrix of this model, its performance was low regarding the class Normal, where it classified a significant number of images as non-COVID19. Similarly, the class non-COVID19 is troublesome, where many images were classified as Normal. Concerning the class COVID-19, it performed adequately with minimal error.



Overall Performance

In this study, the key metric for the classification is *Recall*, on the grounds that the identification of COVID-19 positive images is important, hence the requirement for high *Recall* on each model. All models reached high Recall values (>93%), where the top performer was ResNet101 with 96% score on all metrics as shown on Table 4 and 8; notwithstanding, it had the largest number of trainable parameters which translates to a larger computational capacity compared to the other 4 models.

4. Discussion

It is beyond doubt that COVID-19 affected millions of humans worldwide jeopardizing their health, while at the same time pushing health care services to their limit. Fast and accurate identification of positive COVID-19 cases is extremely essential for the prevention of virus spread. CXR imaging is publicly available at a low cost while producing fast results compared to the more commonly used methods such as RT-PCR tests and CT scans. Thus, numerous studies on COVID-19 identification from CXR images using deep learning methods showed excellent results. However, some of them used limited data for

training and evaluation. Consequently, a model will probably not be able to generalize well to new, unseen data with insubstantial training making its usage in a clinical scenario deficient. In this study, a system is proposed for the automatic detection and diagnosis of COVID-19 from CXR images using deep learning methods. To achieve this, the largest COVID-19 CXR dataset with COVID-19 images was used to train and evaluate 5 different deep learning models on COVID-19 identification.

The proposed methods of this study delivered high results in COVID-19 identification as shown in *Table 8*, attaining equal or more of 93% in Precision and Recall scores. The best performer was *ResNet101*, achieving 96% scores across all metrics.

Table 8. Compiled Results from all models.

Model	Accuracy	Precision	Recall	Trainable Parameters
ResNet50	95%	95%	95%	14,970,880
ResNet101	96%	96%	96%	25,040,899
DenseNet121	93%	93%	93%	5,527,299
DenseNet169	94%	94%	94%	11,059,843
InceptionV3	95%	95%	95%	17,588,163

Henceforward, the plan for this study is to apply lung segmentation and localization on CXR images to increase the classification accuracy of this system, making it more robust and enabling it to generalize even better to new CXR images. Furthermore, another goal is to test the system against professional radiologists and see how well it performs. Furthermore, collaborating with professional radiologists will also result on the acquisition of valuable feedback from them, regarding the usability of this system in a clinical environment, as a decision-making tool.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. Conceptualization, M.C. and T.E.; methodology, M.C., A.V. and P.V. software, M.C., T.P. validation, M.C. and P.V. writing—original draft preparation, M.C. and T.E. writing—review and editing, A.V. and P.V. All authors have read and agreed to the published version of the manuscript." Please turn to the [CRediT taxonomy](#) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: "This research received no external funding"

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: <https://www.kaggle.com/datasets/anasmohammedtahir/covidqu>

Conflicts of Interest: The authors declare no conflict of interest.

References

- [1] "Coronavirus Graphs: Worldwide Cases and Deaths - Worldometer." <https://www.worldometers.info/coronavirus/worldwide-graphs/#total-cases> (accessed Jun. 17, 2022).
- [2] Y. ALIMOHAMADI, M. SEPANDI, M. TAGHDIR, and H. HOSAMIRUDSARI, "Determine the most common clinical symptoms in COVID-19 patients: a systematic review and meta-analysis," *J Prev Med Hyg*, vol. 61, no. 3, pp. E304–E312, Oct. 2020, doi: 10.15167/2421-4248/jpmh2020.61.3.1530.
- [3] A. Tahamtan and A. Ardebili, "Real-time RT-PCR in COVID-19 detection: issues affecting the results," *Expert Review of Molecular Diagnostics*, vol. 20, no. 5, pp. 453–454, May 2020, doi: 10.1080/14737159.2020.1757437.

4. [4] T. Iftner, A. Iftner, D. Pohle, and P. Martus, "Evaluation of the specificity and accuracy of SARS-CoV-2 rapid antigen self-tests compared to RT-PCR from 1015 asymptomatic volunteers." *medRxiv*, p. 2022.02.11.22270873, Feb. 13, 2022. doi: 10.1101/2022.02.11.22270873.
5. [5] T. Ai *et al.*, "Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases," *Radiology*, vol. 296, no. 2, pp. E32–E40, Aug. 2020, doi: 10.1148/radiol.2020200642.
6. [6] S. Salehi, A. Abedi, S. Balakrishnan, and A. Gholamrezanezhad, "Coronavirus Disease 2019 (COVID-19): A Systematic Review of Imaging Findings in 919 Patients," *AJR Am J Roentgenol*, vol. 215, no. 1, pp. 87–93, Jul. 2020, doi: 10.2214/AJR.20.23034.
7. [7] C. Huang *et al.*, "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China," *The Lancet*, vol. 395, no. 10223, pp. 497–506, Feb. 2020, doi: 10.1016/S0140-6736(20)30183-5.
8. [8] D. J. Brenner and E. J. Hall, "Computed Tomography — An Increasing Source of Radiation Exposure," *New England Journal of Medicine*, vol. 357, no. 22, pp. 2277–2284, Nov. 2007, doi: 10.1056/NEJMra072149.
9. [9] F. Shi *et al.*, "Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation, and Diagnosis for COVID-19," *IEEE Rev Biomed Eng*, vol. 14, pp. 4–15, 2021, doi: 10.1109/RBME.2020.2987975.
10. [10] A. S. Alghamdi, K. Polat, A. Alghoson, A. A. Alshdadi, and A. A. Abd El-Latif, "A novel blood pressure estimation method based on the classification of oscillometric waveforms using machine-learning methods," *Applied Acoustics*, vol. 164, p. 107279, Jul. 2020, doi: 10.1016/j.apacoust.2020.107279.
11. [11] M. Hammad, A. Maher, K. Wang, F. Jiang, and M. Amrani, "Detection of abnormal heart conditions based on characteristics of ECG signals," *Measurement*, vol. 125, pp. 634–644, Sep. 2018, doi: 10.1016/j.measurement.2018.05.033.
12. [12] H. Khalil *et al.*, "Classification of Diabetic Retinopathy types based on Convolution Neural Network (CNN)," *Menoufia Journal of Electronic Engineering Research*, vol. 28, no. ICEEM2019-Special Issue, pp. 126–153, Dec. 2019, doi: 10.21608/mjeer.2019.76962.
13. [13] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, Art. no. 7639, Feb. 2017, doi: 10.1038/nature21056.
14. [14] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, "Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks." *arXiv*, Jun. 03, 2017. doi: 10.48550/arXiv.1705.03820.
15. [15] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, "Deep Learning to Improve Breast Cancer Detection on Screening Mammography," *Sci Rep*, vol. 9, no. 1, Art. no. 1, Aug. 2019, doi: 10.1038/s41598-019-48995-4.
16. [16] P. Rajpurkar *et al.*, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," *arXiv*, arXiv:1711.05225, Dec. 2017. doi: 10.48550/arXiv.1711.05225.
17. [17] M. E. H. Chowdhury *et al.*, "Can AI Help in Screening Viral and COVID-19 Pneumonia?," *IEEE Access*, vol. 8, pp. 132665–132676, 2020, doi: 10.1109/ACCESS.2020.3010287.
18. [18] D. Keidar *et al.*, "COVID-19 classification of X-ray images using deep neural networks," *Eur Radiol*, vol. 31, no. 12, pp. 9654–9663, Dec. 2021, doi: 10.1007/s00330-021-08050-1.
19. [19] M. Toğaçar, B. Ergen, and Z. Cömert, "COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches," *Computers in Biology and Medicine*, vol. 121, p. 103805, Jun. 2020, doi: 10.1016/j.compbiomed.2020.103805.
20. [20] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Computers in Biology and Medicine*, vol. 121, p. 103792, Jun. 2020, doi: 10.1016/j.compbiomed.2020.103792.
21. [21] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Phys Eng Sci Med*, vol. 43, no. 2, pp. 635–640, Jun. 2020, doi: 10.1007/s13246-020-00865-4.
22. [22] L. Wang, Z. Q. Lin, and A. Wong, "COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Sci Rep*, vol. 10, no. 1, Art. no. 1, Nov. 2020, doi: 10.1038/s41598-020-76550-z.
23. [23] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection," *IEEE Access*, vol. 8, pp. 91916–91923, 2020, doi: 10.1109/ACCESS.2020.2994762.
24. [24] "COVID-QU-Ex Dataset." <https://www.kaggle.com/datasets/cf77495622971312010dd5934ee91f07ccbcf-dea8e2f7778977ea8485c1914df> (accessed Aug. 25, 2022).
25. [25] "BIMCV-COVID19 – BIMCV." <https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/> (accessed Aug. 25, 2022).
26. [26] "covid-19-image-repository," *GitHub*. <https://github.com/ml-workgroup/covid-19-image-repository> (accessed Aug. 25, 2022).
27. [27] "COVID-19 DATABASE - SIRM." <https://sirm.org/category/covid-19/> (accessed Aug. 25, 2022).
28. [28] J. P. C. PhD, "iee8023/covid-chestxray-dataset." Aug. 23, 2022. Accessed: Aug. 25, 2022. [Online]. Available: <https://github.com/iee8023/covid-chestxray-dataset>
29. [29] "COVID-19 Radiography Database." <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database> (accessed Aug. 25, 2022).
30. [30] "Eurorad.org," *Eurorad - Brought to you by the ESR*. <https://www.eurorad.org/homepage> (accessed Aug. 25, 2022).

31. [31] A. Haghanifar, "COVID-CXNet." Aug. 05, 2022. Accessed: Aug. 25, 2022. [Online]. Available: <https://github.com/armiro/COVID-CXNet>
32. [32] "RSNA Pneumonia Detection Challenge." <https://kaggle.com/competitions/rsna-pneumonia-detection-challenge> (accessed Aug. 25, 2022).
33. [33] "Chest X-Ray Images (Pneumonia)." <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia> (accessed Aug. 25, 2022).
34. [34] "PADCHEST – BIMCV." <https://bimcv.cipf.es/bimcv-projects/padchest/> (accessed Aug. 25, 2022).
35. [35] S. Jaeger *et al.*, "Automatic tuberculosis screening using chest radiographs," *IEEE Trans Med Imaging*, vol. 33, no. 2, pp. 233–245, Feb. 2014, doi: 10.1109/TMI.2013.2284099.
36. [36] S. Candemir *et al.*, "Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration," *IEEE Trans Med Imaging*, vol. 33, no. 2, pp. 577–590, Feb. 2014, doi: 10.1109/TMI.2013.2290491.
37. [37] A. Degerli *et al.*, "COVID-19 infection map generation and detection from chest X-ray images," *Health Inf Sci Syst*, vol. 9, no. 1, p. 15, Apr. 2021, doi: 10.1007/s13755-021-00146-8.
38. [38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." arXiv, Dec. 10, 2015. doi: 10.48550/arXiv.1512.03385.
39. [39] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks." arXiv, Jan. 28, 2018. doi: 10.48550/arXiv.1608.06993.
40. [40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision." arXiv, Dec. 11, 2015. doi: 10.48550/arXiv.1512.00567.
41. [41] H. Chen, Y. Yang, and S. Zhang, "Learning Robust Scene Classification Model with Data Augmentation Based on Xception," *Journal of Physics: Conference Series*, vol. 1575, p. 012009, Jun. 2020, doi: 10.1088/1742-6596/1575/1/012009.