Per-Covid-19: A Benchmark Database For Covid-19 Percentage Prediction From CT-scans

Fares Bougourzi (faresbougourzi@gmail.com)
National Research Council of Italy

Cosimo Distante
National Research Council of Italy

Ouafi Abdelkrim
University of Biskra

Fadi Domaika
IKERBASQUE, Basque Foundation for Science

Abdenour Hadid
Univ. Polytechnique Hauts-de-France, Univ. Lille, CNRS, Centrale Lille, UMR 8520 - IEMN

Abdelmalik Taleb-Ahmed
Univ. Polytechnique Hauts-de-France, Univ. Lille, CNRS, Centrale Lille, UMR 8520 - IEMN

Research Article

Keywords: Covid-19, infections, Covolutional Neural Network, RT-PCR

DOI: https://doi.org/10.21203/rs.3.rs-491375/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Per-Covid-19: A Benchmark Database For Covid-19 Percentage Prediction From CT-scans

Fares Bougourzi1,+,*, Cosimo Distante1,2,+, Ouafi Abdelkrim3,+, Fadi Dornaika4,+, Abdennour Hadid5,+, and Abdelmalik Taleb-Ahmed5,+, *

1Institute of Applied Sciences and Intelligent Systems, National Research Council of Italy, 73100 Lecce, Italy
2Department of Innovation Engineering, University of Salento, 73100 Lecce, Italy
3Laboratory of LESIA, University of Biskra, Algeria
4University of the Basque Country UPV/EHU, San Sebastian, SPAIN; IKERBASQUE, Basque Foundation for Science, Bilbao, SPAIN
5Univ. Polytechnique Hauts-de-France, Univ. Lille, CNRS, Centrale Lille, UMR 8520 - IEMN, F-59313 Valenciennes, France
*faresbougourzi@gmail.com; Abdelmalik.Taleb-Ahmed@uphf.fr
+these authors contributed equally to this work

ABSTRACT

Covid-19 infection recognition is very important step in the fighting against the new pandemic Covid-19. In fact, many methods have been used to recognize the Covid-19 infection including Reverse transcription polymerase chain reaction (RT-PCR), X-ray scan and CT-scan. In addition to the recognition of the Covid-19 infection, CT-scans can provide more important information about the evolution of this disease and its severity. With the extensive number of Covid-19 infections, estimating the Covid-19 percentage can help the intensive care to free up the resuscitation beds for the critical cases and follow other protocol for less severity cases. In this paper, we propose Covid-19 percentage estimation database. Moreover, we evaluate the performance of three Covolutional Neural Network (CNN) architectures which are ResneXt-50, Densenet-161 and Inception-v3. For the three CNN architectures, we use two loss functions which are MSE and Dynamic Huber. In addition, two pretrained scenarios are investigated (ImageNet pretrained models and X-ray pretrained models). The evaluated approaches achieved promising results, where Inception-v3 with using Dynamic Huber loss function and X-ray pretrained model achieved the best performance.

Introduction

Since the end of 2019 year, the world has faced a health crisis because of the spread of Covid-19 pandemic. The crisis influenced the whole aspects of human lives. To save the infected persons lives and stop the spread of Covid-19 disease, many methods have been used to recognize the infected persons. These methods include Reverse transcription polymerase chain reaction (RT-PCR)1, X-ray scan2-4 and CT-scan5,6. Despite that the RT-PCR test is considered as the global standard method for Covid-19 diagnosis, this method has many downsides7,8. In details, RT-PCR test is time consuming and has considerable False-Negative Rate1. Using X-ray scan and CT-scan methods can replace RT-PCR test and give an efficient result in both time and accuracy2,8. However, both these methods need an expert radiologist to identify Covid-19 infection. Artificial Intelligence (AI) can provide the right solution to make this process automatic and limit the need of the radiologist to recognize the Covid-19 infection from these medical imaging. Indeed, computer vision and machine learning communities have proposed many algorithms and frameworks which have proved their efficiency. Especially by using deep leaning methods which have proved their efficiency on different tasks9 including medical imaging tasks10,11.

Compared with the other two diagnosis methods, CT-scans method has many advantages. In addition to the use of CT-scans to recognize the Covid-19 infection, they can be used for other important tasks which include quantifying the infection and monitoring the evolution of the disease which can help on the treatment and save the patient life12. Moreover, the evolution stage can be recognized where the typical signs of Covid-19 infection could be ground-glass opacity (GGO) in the early stage, and pulmonary consolidation in the late stage7,8. According to the estimated Covid-19 infection percentage from the CT-scans, the patient state can be classified into Normal (0%), Minimal (<10%), Moderate (10-25%) , Extent (25-50%) , Severe (50-75%) and Critical (> 75%)13.

The State-of-the-Art methods using CT-scans can be classified into two main tasks: Covid-19 Recognition5,6,14 and Covid-19 Segmentation7,8. In15, Zheng, C.et al. proposed DeCoVNet approach which is based on 3D deep convolutional neural Network to Detect COVID-19 (DeCoVNet) from CT volumes. The input to DeCoVNet is CT volume and its 3D
lung mask which was generated by using pre-trained UNet\textsuperscript{16}. Their proposed DeCoVNet architecture has three parts: vanilla 3D convolution, 3D residual blocks (ResBlocks) and progressive classifier (ProClf). He, K. et al. proposed a multi-task multi-instance deep network (M\textsuperscript{2}UNet) to assess the severity of COVID-19 patients.\textsuperscript{17} Their proposed approach classifies the volumetric CT-scans into two classes of severity: severe or non-severe. Their M\textsuperscript{2}UNet approach consists of a patch level encoder, a segmentation sub-network for lung lobe segmentation, and a classification sub-network for severity assessment. In\textsuperscript{18}, Yao, Q. et al. proposed NormNet architecture, which is a voxel-level anomaly modeling network, to distinguish healthy tissues from the COVID-19 lesion in the thorax area. Zhao, X. et al. proposed a dilated dual attention U-Net (D2A U-Net) approach for COVID-19 lesion segmentation in CT slices based on dilated convolution and a novel dual attention mechanism to address the issues above\textsuperscript{7}.

Most of the state-of-the-art methods have been concentrating on the recognition of Covid-19 from the CT-scans or segmentation of the infected regions. Despite these huge efforts that have been made, the state-of-the-art methods do not provide a helping tool to monitor the patient state, the evolution of the infection and the response of patient to the treatment which can play a crucial role in saving the patient life. In this paper, we propose a fully automatic approach to evaluate the evolution of Covid-19 infection from the CT-scans as regression task which can provide a richer information about the Covid-19 infection evolution. The estimation of Covid-19 percentage can help the intensive care to identify the patients that need urgent care, especially the critical and severe cases. With the extensive number of Covid-19 infections, estimating the Covid-19 percentage can help the intensive care to free up the resuscitation beds for the critical cases and follow other protocol for less severity cases.

Unlike the main stream that dealt with Covid-19 recognition and segmentation, this paper addresses the estimation of Covid-19 infection percentage. To this end, we created Per-Covid-19 database, then we used it to evaluate the performance of three CNN architectures with two loss functions and two pretrained models scenarios. In summary, the main contributions of this paper are:

- We propose Per-Covid-19 dataset for estimating the Covid-19 infection percentage for both slice-level and patient-level. The proposed database consists of 183 CT-scans with the corresponding slice-level Covid-19 infection percentage which were estimated by two expert radiologists. To the best of our knowledge, our work is the first one who propose a finer granularity of Covid-19 virus presence and solve a challenging task related to exact estimation of Covid-19 infection percentage.

- In order to test some state-of-the art methods, we evaluate the performance of three CNN architectures which are ResneXt-50, Densenet-161 and Inception-v3. For the three CNN architectures, we use two loss functions which are MSE and Dynamic Huber loss. In addition, two pretrained scenarios are investigated. In the first scenario, the pretrained models on ImageNet are used. To study the influence of using pretrained models on medical imaging task, we use the pretrained models on X-ray images.

- We make our database and codes publicly available to encourage other researchers to use it as a benchmark for their studies \textsuperscript{1}. (Last accessed on May 2\textsuperscript{nd} 2021)

**Per-Covid-19 dataset**

Our Per-Covid-19 database consists of 183 CT-scans that were confirmed to have Covid-19 infection. The patients were from both genders and aged between 27 to 70 years old. Each volumetric CT-scan of Per-Covid-19 database were taken from different patient, so the number of CT-scans equals to the number of patients. The diagnosis of Covid-19 infection is based on positive Reverse transcription polymerase chain reaction (RT-PCR) and CT-scan manifestations identified by two experienced thoracic radiologists. The CT-scans were collected from two hospitals; Hakim Saidane Biskra and Ziouch Mohamed Tolga (Algeria) from June to December 2020. Each CT-scan consists of 40-70 slices. The two Radiologists estimated Covid-19 infection percentage based on the area of infected lungs over the overall size of the lungs. From each CT scan, the radiologists selected the slices that contain clear signs of Covid-19 infection and the ones that do not contain any Covid-19 signs. In our dataset, we kept the slices that have diagnosis agreement of the two radiologists. In summary, from each CT-scan, we can have between 2-45 typical slice images. Figure 1 shows the histogram of number of CT-scans over the number of CT slices. Figure 2 shows examples of slices images with their corresponding Covid-19 infection percentage. For evaluating different machine learning methods, we divided the Per-Covid-19 database into patient-independent five folds, where each patient slices are included into one fold.

\textsuperscript{1}https://github.com/faresbougourzi/Per-Covid-19
Methods

Loss Functions
In our experiments, we used two loss functions which are Mean Squared Error (MSE) and Dynamic Huber loss. The loss functions are defined for \( N \) batch size and \( X = (x_1, x_2, \ldots, x_N) \) are the ground-truth percentages and \( \hat{X} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N) \) are their corresponding estimated percentages.

MSE is sensitive towards outliers. For \( N \) predictions, MSE loss function is defined by:

\[
L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2
\]

(1)

On the other hand, Huber loss function is less sensitive to outliers in data than \( L_2 \) loss function. For \( N \) training batch size images, Huber loss function is defined by:\(^19\):

\[
L_{Huber} = \frac{1}{N} \sum_{i=1}^{N} z_i
\]

(2)

where \( N \) is the batch size and \( z_i \) is defined by:

\[
z_i = \begin{cases} 
0.5 (x_i - \hat{x}_i)^2, & \text{if } |x_i - \hat{x}_i| \leq \beta \\
\beta |x_i - \hat{x}_i| - 0.5 \beta^2, & \text{otherwise}
\end{cases}
\]

(3)

where \( \beta \) is a controlling hyperparameter. In our experiments \( \beta \) decreases from 15 to 1 during the training.
**Figure 2.** Slices examples which have the Covid-19 infection percentages of 0%, 5%, 18%, 40%, 70% and 80%, respectively. These slices are classified as Normal, Minimal (<10%), Moderate (10-25%) , Extent (25-50%) , Severe (50-75%) and Critical (> 75%), respectively.

**Evaluation metrics**

To evaluate the performance of the state-of-the-art methods, we use slice-level metrics which are Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Pearson Correlation coefficient (PC) which are defined in equations 4, 5 and 6, respectively.

\[
\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| 
\]

\[
\text{RMSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 
\]

\[
\text{PC} = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^{n} (\hat{y}_i - \bar{\hat{y}})^2}} 
\]

Where \( Y = (y_1, y_2, ..., y_n) \) are the ground-truth Covid-19 percentages of the testing data which consists of \( n \) slices and \( \hat{Y} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_n) \) are their corresponding estimated percentages. For equation 6, \( \bar{y} \) and \( \bar{\hat{y}} \) are the means of the ground-truth percentages and the estimated ones, respectively.

In addition, we used subject-level metrics which are MAE_subj, RMSE_subj and PC_subj which are defined in equations 7, 8 and 9, respectively.

\[
\text{MAE}_{\text{subj}} = \frac{1}{s} \sum_{i=1}^{s} |y_{s_i} - \hat{y}_{s_i}| 
\]

\[
\text{RMSE}_{\text{subj}} = \frac{1}{s} \sum_{i=1}^{s} (y_{s_i} - \hat{y}_{s_i})^2 
\]
Where \( Y_s = (y_{s1}, y_{s2}, ..., y_{sn}) \) are the ground-truth means of Covid-19 percentages of each patient’s slices from the testing data. and \( \hat{Y}_s = (\hat{y}_{s1}, \hat{y}_{s2}, ..., \hat{y}_{sn}) \) are their corresponding estimated patient-level percentages (means of patient’ slices percentages). For equation 9, \( \bar{y}_s \) and \( \bar{\hat{y}}_s \) are the means of the ground-truth patient percentages and the estimated ones, respectively.

MAE and RMSE are error indicators where the smaller values indicates better performance. From other hand, PC is a statistic measurement of linear correlation between two variables \( Y \) and \( \hat{Y} \). A value of 1 means that there is a total positive linear correlation and 0 indicates no linear correlation.

**Results**

To train and test the CNN architectures (ResneXt-50, DenseNet-161 and Inception-v3), we used Pytorch\(^ {20} \) library and SGD optimizer with momentum equals 0.9 is used during the training phase. All experiments were carried out on PC with 64 GB Ram and NVIDIA GPU Device GeForce TITAN RTX 24 GB. Each CNN architecture was trained for 30 epochs with initial of \( 10^{-4} \) with decays by 0.1 every 10 epochs and batch size equals 20. In addition, we used active data augmentation by rotating the input image by random angle between (-10 to 10 degrees). In summary, our experiments are divided into two scenarios. In the first scenario, we used retrained models of ImageNet. While, in the second scenario, we used pretrained models that were trained on medical imaging task.

**First Scenario**

In the first scenario, we used three pretrained CNN architectures on ImageNet\(^ {21} \) (ResneXt-50\(^ {22} \), Inception-V3\(^ {33} \) and DenseNet-161\(^ {24} \)). Moreover, we used two loss functions which are MSE and Dynamic Huber. Table 1 summarizes the obtained results of the first scenario. From the results, we notice that for all models almost the Dynamic Huber loss gives better results then MSE loss function. This proves the efficiency of using Dynamic Huber loss function for this regression task. From the other hand, we notice that the three trained models with Huber dynamic loss achieved close results. In details, ResneXt-50 achieved the best results performance in MAE, PC\(_{subj}\), MAE\(_{subj}\) and RMSE\(_{subj}\). While, Densenet-161 achieved the best result for PC metric and Inception-v3 for RMSE metric.

**Second Scenario**

In the second scenario, we use the same models as the first scenario but this time they were trained on the recognition of Covid-19 from X-ray scans\(^ {2} \). In more details, four lung diseases plus neutral were used to train the CNN architectures\(^ {2} \). The objective of this scenario is to study the influence of the pretrained model which were trained on medical imaging task. The experimental results are summarized in Table 2. From these results, we notice that Inception-v3 achieved the best performance. Similar to the results of the first scenario, the Dynamic huber loss gives better results then MSE loss function for most of the evaluation metrics.

**Discussion**

The comparison between the first and second scenarios experiments (Tables 1 and 2) shows that the pretrained models on medical imaging task give better result than the pretrained models of ImageNet. From all experiments, we conclude that the best scenario for Covid-19 infection percentage estimation is by using Inception-v3 architecture with X-ray pretrained model and Dynamic Huber loss function.

**Conclusion**

In this paper, we introduced Per-Covid-19 database which presents Covid-19 infection percentage estimation. Moreover, we evaluated the performance of three CNN architectures which are ResneXt-50, DenseNet-161 and Inception-v3. For the three CNN architectures, we use two loss functions which are MSE and Dynamic Huber loss. In addition, we evaluate two pretrained models scenarios. In the first scenario, we used ImageNet pretrained models. In the second scenario, we used pretrained models that were trained on X-ray scans to investigate the influence of using pretrained models that were trained on medical imaging task. The experimental results show that using the X-ray pretrained models improve the results. Moreover, the experiments using Dynamic Huber loss function achieved better performance than the ones used standard MSE loss function. From other hand, Inception-v3 outperformed ResneXt-50 and Densenet-161 architectures in both scenarios.
Table 1. Five folds cross-validation results of the first scenario (pretrained ImageNet models) with using three CNN architectures ResneXt-50, Densenet-161, Inception-v3 and two loss functions (MSE, Huber).

| Architecture | Fold | PC ↑ | MAE ↓ | RMSE ↓ | PC subj ↑ | MAE subj ↓ | RMSE subj ↓ |
|--------------|------|------|-------|--------|-----------|-------------|-------------|
| ResneXt-50   |      |      |       |        |           |             |             |
| MSE          | Fold 1 | 0.9479 | 4.96 | 8.57 | 0.9645 | 3.65 | 6.41 |
|              | Fold 2 | 0.8833 | 6.27 | 12.00 | 0.9373 | 4.58 | 7.88 |
|              | Fold 3 | 0.9179 | 5.81 | 11.94 | 0.9255 | 5.14 | 10.77 |
|              | Fold 4 | 0.9495 | 4.40 | 7.03 | 0.9791 | 2.93 | 4.36 |
|              | Fold 5 | 0.9459 | 5.79 | 8.92 | 0.9666 | 4.71 | 6.30 |
|              | Mean   | 0.9289 | 5.45 | 9.69 | 0.9546 | 4.20 | 7.14 |
| ResneXt-50   | Fold 1 | 0.9516 | 4.62 | 8.26 | 0.9649 | 3.61 | 6.32 |
| MSE          | Fold 2 | 0.8820 | 6.10 | 12.14 | 0.9393 | 4.50 | 7.76 |
|              | Fold 3 | 0.9193 | 5.73 | 11.93 | 0.9309 | 5.02 | 10.60 |
|              | Fold 4 | 0.9625 | 3.85 | 6.15 | 0.9877 | 2.75 | 3.55 |
|              | Fold 5 | 0.9492 | 5.41 | 8.55 | 0.9729 | 4.02 | 5.62 |
|              | Mean   | 0.9329 | 5.14 | 9.41 | 0.9592 | 3.98 | 6.77 |
| Densenet-161 | Fold 1 | 0.9429 | 5.24 | 8.90 | 0.9520 | 4.29 | 7.23 |
| MSE          | Fold 2 | 0.8747 | 6.39 | 12.55 | 0.9299 | 5.04 | 8.36 |
|              | Fold 3 | 0.9180 | 5.53 | 12.09 | 0.9298 | 4.59 | 10.70 |
|              | Fold 4 | 0.9554 | 4.41 | 6.85 | 0.9853 | 3.22 | 4.19 |
|              | Fold 5 | 0.9482 | 5.86 | 8.83 | 0.9708 | 4.57 | 6.14 |
|              | Mean   | 0.9278 | 5.49 | 9.85 | 0.9535 | 4.34 | 7.32 |
| Densenet-161 | Fold 1 | 0.9447 | 5.17 | 9.13 | 0.9599 | 4.20 | 7.10 |
| Huber        | Fold 2 | 0.8990 | 5.93 | 11.10 | 0.9482 | 4.35 | 7.35 |
|              | Fold 3 | 0.9174 | 5.68 | 12.10 | 0.9283 | 4.75 | 10.75 |
|              | Fold 4 | 0.9603 | 3.90 | 6.42 | 0.9856 | 2.95 | 3.89 |
|              | Fold 5 | 0.9487 | 6.49 | 9.83 | 0.9658 | 5.79 | 7.97 |
|              | Mean   | 0.9340 | 5.43 | 9.71 | 0.9576 | 4.41 | 7.41 |
| Inception-v3 | Fold 1 | 0.9412 | 5.24 | 8.86 | 0.9576 | 3.94 | 6.65 |
| MSE          | Fold 2 | 0.8489 | 6.91 | 13.72 | 0.9246 | 4.94 | 8.61 |
|              | Fold 3 | 0.9142 | 5.69 | 12.01 | 0.9247 | 4.73 | 10.67 |
|              | Fold 4 | 0.9565 | 3.96 | 6.52 | 0.9831 | 2.60 | 3.81 |
|              | Fold 5 | 0.9443 | 5.73 | 8.86 | 0.9622 | 4.58 | 6.42 |
|              | Mean   | 0.9210 | 5.51 | 9.99 | 0.9505 | 4.16 | 7.23 |
| Inception-v3 | Fold 1 | 0.9486 | 4.91 | 8.75 | 0.9613 | 4.04 | 6.94 |
| Huber        | Fold 2 | 0.8720 | 6.38 | 12.39 | 0.9293 | 4.84 | 8.43 |
|              | Fold 3 | 0.9264 | 5.36 | 11.15 | 0.9351 | 4.64 | 9.89 |
|              | Fold 4 | 0.9637 | 3.76 | 6.00 | 0.9865 | 2.54 | 3.57 |
|              | Fold 5 | 0.9539 | 5.43 | 8.34 | 0.9738 | 4.23 | 5.87 |
|              | Mean   | 0.9329 | 5.17 | 9.33 | 0.9572 | 4.06 | 6.94 |

Methods

Topical subheadings are allowed. Authors must ensure that their Methods section includes adequate experimental and characterization data necessary for others in the field to reproduce their work.

References

1. Kucirka, L. M., Lauer, S. A., Laeyendecker, O., Boon, D. & Lessler, J. Variation in False-Negative rate of reverse transcriptase polymerase chain Reaction–Based SARS-CoV-2 Tests by time since exposure. *Annals Intern. Medicine* 173, 262–267, DOI: 10.7326/M20-1495 (2020). Publisher: American College of Physicians.

2. Vantaggiato, E. et al. COVID-19 Recognition Using Ensemble-CNNs in Two New Chest X-ray Databases. *Sensors* 21, DOI: 10.3390/s21051742 (2021).
Table 2. Five folds cross-validation results of the second scenario (pretrained X-ray models) with using three CNN architectures ResneXt-50, Densenet-161, Inception-v3 and two loss functions (MSE, Huber).

| Architecture | Fold | PC ↑ | MAE ↓ | RMSE ↓ | PC subj ↑ | MAE subj ↓ | RMSE subj ↓ |
|--------------|------|------|-------|--------|-----------|------------|-------------|
| Resnext-50   | Fold 1 | 0.9453 | 5.24 | 8.77  | 0.9619 | 4.05       | 6.88        |
|              | Fold 2 | 0.8569 | 6.68 | 13.39 | 0.9234 | 4.87       | 8.70        |
|              | Fold 3 | 0.9083 | 6.17 | 12.53 | 0.9196 | 5.23       | 11.17       |
|              | Fold 4 | 0.9589 | 3.97 | 6.32  | 0.9884 | 2.31       | 3.22        |
|              | Fold 5 | 0.9457 | 5.99 | 9.05  | 0.9674 | 4.92       | 6.43        |
|              | Mean   | **0.9230** | **5.61** | **10.01** | **0.9521** | **4.28** | **7.22** |
| Resnext-50   | Fold 1 | 0.9487 | 4.96 | 8.48  | 0.9633 | 3.80       | 6.43        |
|              | Fold 2 | 0.8227 | 6.93 | 15.29 | 0.9147 | 4.96       | 9.25        |
|              | Fold 3 | 0.9191 | 5.35 | 11.84 | 0.9301 | 4.27       | 10.47       |
|              | Fold 4 | 0.9622 | 3.70 | 6.09  | 0.9873 | 2.39       | 3.26        |
|              | Fold 5 | 0.9507 | 5.50 | 8.80  | 0.9703 | 4.34       | 6.30        |
|              | Mean   | **0.9207** | **5.29** | **10.10** | **0.9532** | **3.95** | **7.14**   |
| Densenet-161 | Fold 1 | 0.9438 | 5.15 | 8.65  | 0.9611 | 3.79       | 6.36        |
|              | Fold 2 | 0.8768 | 6.48 | 12.34 | 0.9354 | 4.62       | 8.04        |
|              | Fold 3 | 0.9116 | 5.86 | 12.44 | 0.9259 | 4.73       | 10.89       |
|              | Fold 4 | 0.9576 | 4.11 | 6.65  | 0.9861 | 2.97       | 4.14        |
|              | Fold 5 | 0.9479 | 5.79 | 8.99  | 0.9691 | 4.82       | 6.38        |
|              | Mean   | **0.9275** | **5.48** | **9.81** | **0.9555** | **4.18** | **7.16**   |
| Inception-v3 | Fold 1 | 0.9440 | 4.97 | 8.78  | 0.9569 | 3.91       | 6.84        |
|              | Fold 2 | 0.8938 | 6.14 | 11.36 | 0.9459 | 4.57       | 7.58        |
|              | Fold 3 | 0.9158 | 5.89 | 12.33 | 0.9290 | 4.99       | 10.87       |
|              | Fold 4 | 0.9643 | 3.67 | 5.93  | 0.9893 | 2.33       | 3.16        |
|              | Fold 5 | 0.9526 | 5.48 | 8.71  | 0.9697 | 4.56       | 6.54        |
|              | Mean   | **0.9341** | **5.23** | **9.42** | **0.9582** | **4.07** | **7.00**   |
| Inception-v3 | Fold 1 | 0.9483 | 5.22 | 8.51  | 0.9630 | 3.95       | 6.53        |
|              | Fold 2 | 0.8571 | 6.63 | 13.14 | 0.9258 | 5.01       | 8.57        |
|              | Fold 3 | 0.9111 | 5.89 | 12.28 | 0.9240 | 5.22       | 10.84       |
|              | Fold 4 | 0.9605 | 3.91 | 6.22  | 0.9878 | 2.52       | 3.27        |
|              | Fold 5 | 0.9463 | 6.10 | 9.22  | 0.9667 | 5.02       | 6.74        |
|              | Mean   | **0.9247** | **5.55** | **9.87** | **0.9535** | **4.34** | **7.19**   |

3. Wang, G. et al. A deep-learning pipeline for the diagnosis and discrimination of viral, non-viral and COVID-19 pneumonia from chest X-ray images. Nat. Biomed. Eng. 1–13 (2021). Publisher: Nature Publishing Group.

4. Wang, L., Lin, Z. Q. & Wong, A. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci. Reports 10, 19549, DOI: 10.1038/s41598-020-76550-z (2020). Number: 1 Publisher: Nature Publishing Group.

5. Bougourzi, F., Contino, R., Distante, C. & Taleb-Ahmed, A. CNR-IEMN: A Deep Learning Based Approach To Recognize COVID-19 From CT-SCAN. In 2021 IEEE International Conference on Autonomous Systems (IEEE ICAS) (2021).

6. Lassau, N. et al. Integrating deep learning CT-scan model, biological and clinical variables to predict severity of COVID-19 patients. Nat. Commun. 12, 634, DOI: 10.1038/s41467-020-20657-4 (2021). Number: 1 Publisher: Nature Publishing Group.

7. Zhao, X. et al. D2A U-Net: Automatic Segmentation of COVID-19 Lesions from CT Slices with Dilated Convolution and Dual Attention Mechanism. arXiv:2102.05210 [cs, eess] (2021). ArXiv: 2102.05210.
8. Fan, D.-P. et al. Inf-Net: Automatic COVID-19 Lung Infection Segmentation From CT Images. *IEEE Transactions on Med. Imaging* 39, 2626–2637, DOI: 10.1109/TMI.2020.2996645 (2020). Conference Name: IEEE Transactions on Medical Imaging.

9. Bougourzi, F., Dornaika, F., Mokrani, K., Taleb-Ahmed, A. & Ruichek, Y. Fusion Transformed Deep and Shallow features (FTDS) for Image-Based Facial Expression Recognition. *Expert. Syst. with Appl.* 113459 (2020). Publisher: Elsevier.

10. Paladini, E. et al. Two Ensemble-CNN Approaches for Colorectal Cancer Tissue Type Classification. *J. Imaging* 7, 51, DOI: 10.3390/jimaging7030051 (2021). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

11. Krittanawong, C. et al. Machine learning and deep learning to predict mortality in patients with spontaneous coronary artery dissection. *Sci. Reports* 11, 8992, DOI: 10.1038/s41598-021-88172-0 (2021). Number: 1 Publisher: Nature Publishing Group.

12. Lei, J., Li, J., Li, X. & Qi, X. CT Imaging of the 2019 Novel Coronavirus (2019-nCoV) Pneumonia. *Radiology* 295, 18–18, DOI: 10.1148/radiol.2020200236 (2020). Publisher: Radiological Society of North America.

13. Jalaber, C. et al. Chest CT in COVID-19 pneumonia: A review of current knowledge. *Diagn. Interv. Imaging* 101, 431–437, DOI: 10.1016/j.diii.2020.06.001 (2020).

14. Wang, X. et al. A Weakly-Supervised Framework for COVID-19 Classification and Lesion Localization From Chest CT. *IEEE Transactions on Med. Imaging* 39, 2615–2625, DOI: 10.1109/TMI.2020.2995965 (2020). Conference Name: IEEE Transactions on Medical Imaging.

15. Zheng, C. et al. Deep Learning-based Detection for COVID-19 from Chest CT using Weak Label. *MedRxiv* 13 (2020).

16. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F. (eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, 234–241, DOI: 10.1007/978-3-319-24574-4_28 (Springer International Publishing, Cham, 2015).

17. He, K. et al. Synergistic learning of lung lobe segmentation and hierarchical multi-instance classification for automated severity assessment of COVID-19 in CT images. *Pattern Recognit.* 113, 107828, DOI: 10.1016/j.patcog.2021.107828 (2021).

18. Yao, Q., Xiao, L., Liu, P. & Kevin Zhou, S. Label-Free Segmentation of COVID-19 Lesions in Lung CT. *IEEE Transactions on Med. Imaging* 1–1, DOI: 10.1109/TMI.2021.3066161 (2021). Conference Name: IEEE Transactions on Medical Imaging.

19. Huber, P. J. Robust estimation of a location parameter. In *Breakthroughs in statistics*, 492–518 (Springer, 1992).

20. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, 8026–8037 (2019).

21. Russakovsky, O. et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* 115, 211–252, DOI: 10.1007/s11263-015-0816-y (2015).

22. Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500 (2017).

23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826 (2016).

24. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708 (2017).

Acknowledgements (not compulsory)

Special thanks for Hospitals of Hakim Saidane Biskra and Ziouch Mohamed Tolga (Algeria) for their datasets and for their help estimated Covid-19 infection percentage.

Author contributions statement

**Fares Bougourzi:** Data curation, Formal analysis, Investigation, Resources, Project administration, Software, Validation, Visualization Writing - original draft. **Cosimo Distante:** Data curation, Formal analysis, Investigation, Resources, Software, Project administration, Validation, Supervision, Visualization Writing - original draft. **Ouafi Abdelkrim:** Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing - original
draft. **Fadi Dornaika:** Formal analysis, Investigation, Methodology, Supervision, Writing - original draft. **Abdenour Hadid:** Formal analysis, Investigation, Methodology, Supervision, Writing - original draft. **Abdelmalik Taleb-Ahmed:** Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Supervision, Writing - original draft.
Figure 1

Histogram of slices number for the patients in the proposed PrC-Covid-19 database.
Figure 2

Slices examples which have the Covid-19 infection percentages of 0%, 5%, 18%, 40%, 70% and 80%, respectively. These slices are classified as Normal, Minimal (<10%), Moderate (10-25%) , Extent (25-50%) , Severe (50-75%) and Critical (> 75%), respectively
Figures

Figure 1

Histogram of slices number for the patients in the proposed Prc-Covid-19 database.
Figure 2

Slices examples which have the Covid-19 infection percentages of 0%, 5%, 18%, 40%, 70% and 80%, respectively. These slices are classified as Normal, Minimal (<10%), Moderate (10-25%) , Extent (25-50%) , Severe (50-75%) and Critical (> 75%), respectively
Figure 1

Histogram of slices number for the patients in the proposed Prc-Covid-19 database.
Figure 2

Slices examples which have the Covid-19 infection percentages of 0%, 5%, 18%, 40%, 70% and 80%, respectively. These slices are classified as Normal, Minimal (<10%), Moderate (10-25%) , Extent (25-50%) , Severe (50-75%) and Critical (> 75%), respectively
Figure 1

Histogram of slices number for the patients in the proposed Prc-Covid-19 database.
Figure 2

Slices examples which have the Covid-19 infection percentages of 0%, 5%, 18%, 40%, 70% and 80%, respectively. These slices are classified as Normal, Minimal (<10%), Moderate (10-25%), Extent (25-50%), Severe (50-75%) and Critical (> 75%), respectively.