Department of Computer Engineering
Academic Year 2023-2024

COVID 19 Detection through CT Scans

Deep Learning Laboratory

By

Mohammed Raza 60004200019
Neel Kothari 60004200034

Guide:

Prof. Ruhina Karani

Lecturer, DJSCE

I. INTRODUCTION

In March 2020, the World Health Organization (WHO) officially declared the outbreak of COVID-19, the disease caused by SARS-CoV-2, a pandemic. COVID-19 is highly infectious and can potentially evolve to fatal acute respiratory distress syndrome (ARDS). Early detection and diagnosis is a critical factor to control the COVID-19 spreading. The most common screening method to detect it is the reverse-transcription polymerase chain reaction (RT-PCR) testing. However, it is a laborious method and some studies reported its low sensitivity in early stages [1]. Chest scans such as X-rays and Computer tomography (CT) scans have been used to identify morphological patterns of lung lesions linked to the COVID-19. However, the accuracy of the diagnosis of COVID-19 by Chest scans strongly depends on experts [2] and Deep learning techniques have been studied as a tool to automate and help with the diagnosis [3–8].

A computed tomography scan, or CT scan, produces detailed images of organs, bones, soft tissues and blood vessels. CT images allow physicians to identify internal structures and see their shape, size, density and texture. Different from conventional X-Rays, CT scans produce a set of slices of a given region of the body without overlaying the different body structures. Thus, CT scans give a much more detailed picture of the patient's condition than the conventional X-Rays. This detailed information can be used to determine whether there is a medical problem as well as the extent and exact location of the problem. For these reasons, a number of deep learning based methodologies have been recently proposed for COVID-19 screening in CT scans [9–14].

The main bottleneck for the realization of a study such as the ones cited above is the lack of good quality comprehensive data sets. Possibly the first attempt to create such a data set was the so-called COVID-CT dataset [15] which consists of images mined from research papers. Different versions of this dataset were used in Refs. [9–12]. For its most updated version, the highest reported accuracy, F1-score, and AUC were 86%, 85%, and 94% [9], respectively. More recently, Soares et al. [14] made another set of CT scans publicly available. It consists of 2482 CT scans taken from hospitals in the city of Sao ¯ Paulo, Brazil. They have reported an accuracy, sensitivity, and positive predictive value of 97.38%, 95.53%, and 99.16%, respectively.

These two datasets are, to date, the biggest publicly available datasets. It can be seen that the difference in the best results obtained in each of them is significant which raises two questions: (i) Are the discrepancies in the results due to the differences in the datasets? (ii) Does a model trained in one dataset have good performance when tested with the other? This work aims to answer these two questions.

Another drawback of the best performing techniques is their immense number of parameters which directly influence their footprint and latency. Improving these two metrics allows the model to be more easily embedded in mobile applications and to be less of a burden on the server if provided as a web-service receiving an enormous number of requests per second. In addition, having a more compact baseline model allows the exploitation of higher resolution inputs without making the computational cost prohibitively high. Broadly speaking, the computational cost is an important factor in the accessibility and availability of the technology to the public.

Thus, the main goals of this work are: (i) to propose a high-quality yet compact deep-learnign model for the screening of COVID-19 in CT scans and (ii) to address, for the first time, the aforementioned questions regarding the two biggest datasets, and a (iii) proposal of a voting based evaluation approach.

To produce an efficient model we exploit and extend the EfficientNet Family of deep artificial neural networks along with a data augmentation process and transfer learning. Following previous evaluation protocols [9,14], state-of-the-art results are presented for the COVID-CT dataset (accuracy of 87.60%) and the SARS-CoV-2 CT-scan dataset (accuracy of 98.99%).

The voting based approach showed promising results for the Covid-19 detection in CT images. The remainder of this work is organized as follows. Section 2 present the details of COVID-CT [15] and SARS-CoV-2 CT-scan [14] datasets. The methodology is described in Section 3 and the experiments along with the results in Section 4. Finally, Section 5 presents the conclusion of this work.

II. DATASET

The SARS-CoV-2 CT-scan dataset [14] consists of 2482 CT scans from 120 patients, with 1252 CT scans of 60 patients infected by SARS-CoV-2 rom males (32) and females (28), and 1230 CT scan images of 60 non-infected patients by SARS-CoV-2 from males (30) and females (30), but presenting other pulmonary diseases. Data was collected from hospitals of S˜ ao Paulo, Brazil. In this dataset the images consist of digital scans of the printed CT exams and they have no standard regarding image size (the dimensions of the smallest image in the dataset are 104 × 153 while the largest images are 484 × 416), Fig. 1 shows some examples. This dataset also lacks standardization regarding the contrast of the images, as can be seen in Fig. 2. For method evaluation, the protocol presented in Ref. [14] proposes to randomly divide the dataset in training (80%) and test (20%) partitions. The dataset is available at https://www.kaggle.com/plamenedua rdo/sarscov2-ctscan-dataset.

III. METHODOLOGY

A) Working

Testing of all people for SARS-CoV-2, including those who have no symptoms, who show symptoms of infection such as trouble breathing, fever, sore throat or loss of the sense of smell and taste, and who may have been exposed to the virus will help prevent the spread of COVID-19 by identifying people who are in need of care in a timely fashion. A positive test early in the course of the illness enables individuals to isolate themselves – reducing the chances that they will infect others and allowing them to seek treatment earlier, reducing disease severity and the risk of long-term disability, or death. Testing of people who have been in contact with others who have a documented infection is also important. A negative test doesn't mean you're in the clear; you could become infectious later. Therefore, even if you test negative, you need to continue to protect yourself and others by washing your hands frequently, physically distancing, and wearing a face mask. A positive test makes it clear that you must isolate yourself, and that others with whom you have been in contact since the time of your exposure should also get tested.

Since it is recognized that half of all SARS-CoV-2 infections are transmitted by people who are not showing any symptoms, identifying infected individuals while they are pre symptomatic, as well as those who are asymptomatic, will play a major role in stopping the pandemic.

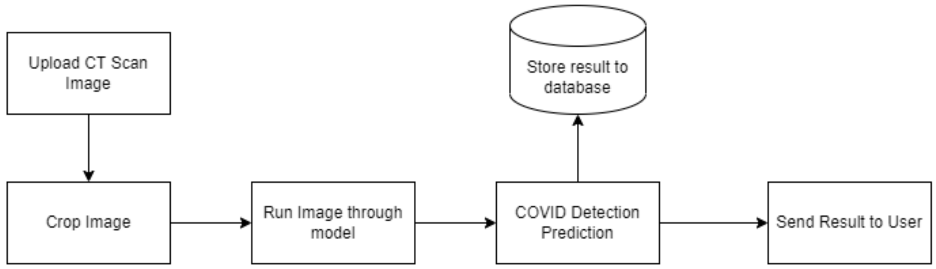


Fig 1. Working Flow

The user uploads an image of the CT scan. The image is cropped as per the requirements. The cropped image is then passed to the Machine Learning model built on CNN. The trained model then predicts on the image given by the user and the prediction is sent to the user as well as it is stored in the database for help in future training.The model developed in this project can be used by hospitals to detect COVID 19 accurately by the CT Scan image of the patient. The hospital will use the data from the model as well as add new samples to the database which will help in increasing the training and testing set for the model, resulting in more accuracy of the model for predicting results. The model will help the people to detect COVID 19 at an early stage and cure it as soon as possible as detection of COVID 19 is the only way to cure it.

B) Algorithm Used

Deep learning is a subset of machine learning, which is a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Convolutional Neural Networks (CNN)

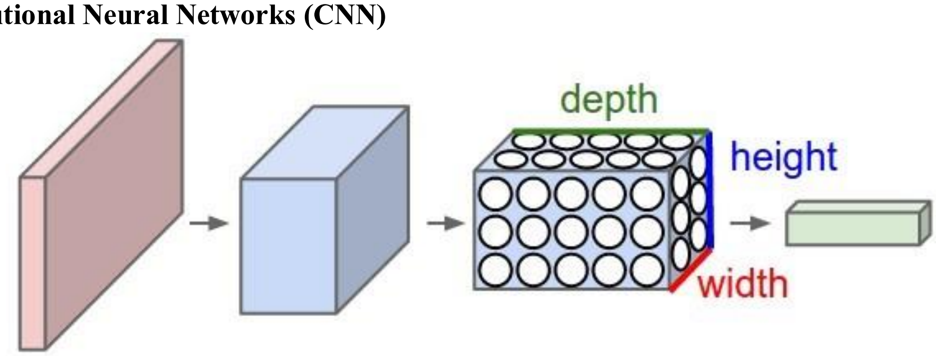


Fig 2. CNN Working Flow

Convolutional Neural Networks are very similar to ordinary Neural Networks they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g., SVM/SoftMax) on the last (fully connected) layer and all the tips/tricks developed for learning regular Neural Networks still apply.

Neural Networks receive an input (a single vector) and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer, and where neurons in a single layer function completely independently and do not share any connections. The last fully connected layer is called the "output layer" and in classification settings it represents the class scores. Regular Neural Nets don't scale well to full images.

Convolutional Neural Networks take advantage of the fact that the input consists of images, and they constrain the architecture in a more sensible way. Unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. For example, the input images in CIFAR-10 are an input volume of activations, and the volume has dimensions 32x32x3 (width, height, depth respectively). The neurons in a layer will only be connected to a small region of the layer before it, instead of all the neurons in a fully connected manner. Moreover, the final output layer would for CIFAR-10 have dimensions 1x1x10, because by the end of the ConvNet architecture we will reduce the full image into a single vector of class scores, arranged along the depth dimension.

A simple ConvNet is a sequence of layers, and every layer of a ConvNet transforms one volume of activations to another through a differentiable function. We use three main types of layers to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully Connected Layer (exactly as seen in regular Neural Networks). We will stack these layers to form a full ConvNet architecture.

Batch Normalization

Batch normalization is a technique for training very deep neural networks that normalizes the contributions to a layer for every mini-batch. This has the impact of settling the learning process and drastically decreasing the number of training epochs required to train deep neural networks. Batch normalization gives a rich method of parametrizing any deep neural network. The reparameterization fundamentally decreases the issue of planning updates across numerous layers.

Working of Batch Normalization

It does this scaling the output of the layer, explicitly by normalizing the activations of each input variable per mini-batch, for example, the enactments of a node from the last layer. By brightening the inputs to each layer, it would make a stride towards accomplishing the fixed distributions of inputs that would evacuate the ill impacts of the internal covariate shift. Normalizing the activations of the earlier layer implies that presumptions the ensuing layer makes about the spread and distribution of inputs during the weight update won't change, in any event not significantly. This has the impact of stabilizing and accelerating the preparation training procedure of deep neural networks. For little, smaller mini batches that don't contain an agent distribution of models from the training dataset, the distinctions in the normalized inputs among training and inference (utilizing the model subsequent to training) can bring about perceptible contrasts in execution performance. This can be tended to with a change of the technique called Batch Renormalization that makes the appraisals of the variable mean and standard deviation increasingly stable across mini-batches. In the batch normalized network, the mean and variances remain moderately steady all through the network. For an unnormalized network, they seem to develop exponentially with profundity.

Dropout

Dropout means to drop out units that are covered up and noticeable in a neural network. Dropout is a staggeringly in vogue method to overcome overfitting in neural networks. Dropout is a method where randomly selected neurons are dropped during training. They are "dropped-out" arbitrarily. This infers that their contribution to the activation of downstream neurons is transiently evacuated on the forward pass and any weight refreshes are not applied to the neuron on the backward pass. If neurons are haphazardly dropped out of the network during training, that other neuron will have to step in and handle the portrayal required to make predictions for the missing neurons. This is believed to bring about various independent internal representations being learned by the network.

The feed-forward operation of a standard neural network

$$
\begin{aligned}
z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\mathbf{y}^l + b_i^{(l+1)}, \\
y_i^{(l+1)} &= f(z_i^{(l+1)}),
\end{aligned}
$$

With dropout, the feed-forward operation becomes

$$
\begin{aligned}
r_j^{(l)} &\sim \text{Bernoulli}(p), \\
\widetilde{\mathbf{y}}^{(l)} &= \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \\
z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\widetilde{\mathbf{y}}^l + b_i^{(l+1)}, \\
y_i^{(l+1)} &= f(z_i^{(l+1)}).
\end{aligned}
$$

Fig. 3 Neural Network Operations

Implementing dropout in Deep Neural Networks

1.Generally, utilize a small dropout value of 20%-50% of neurons with 20% providing a great beginning point. A probability too low has an insignificant impact and worth too high outcomes in under-learning by the system.

2.You are going to show signs of improvement execution when dropout is utilized on a larger network, allowing the model a greater amount of a chance to learn free portrayals.

3.Use dropout on approaching (obvious) just as concealed units. The utilization of dropout at each layer of the system has demonstrated great outcomes.

Adam Optimization

Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first order and second-order moments. The method is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters". Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm.

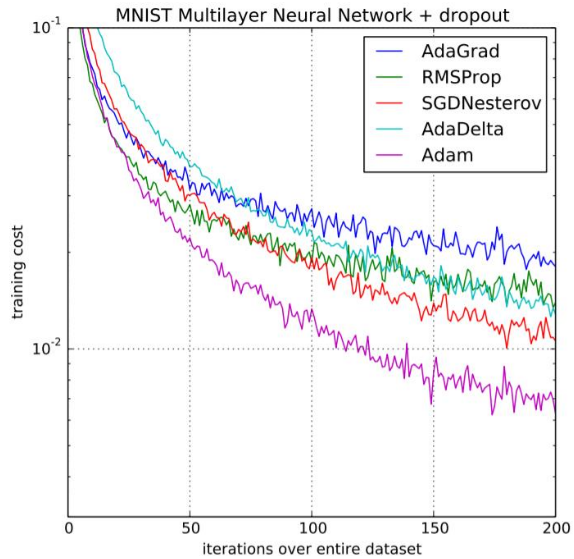


Fig 4. Adam Optimizer

IV. RESULTS & EXPERIMENTATION

1) Model 1 (Basic CNN)

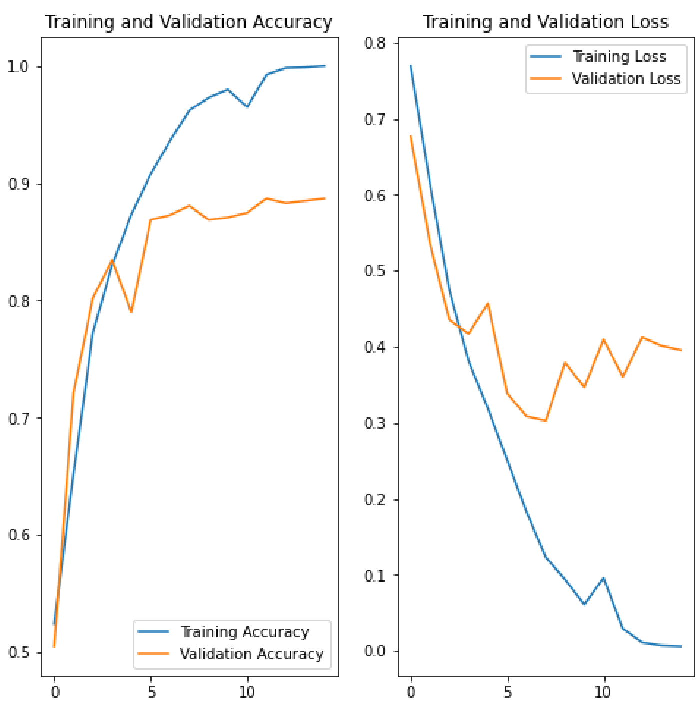
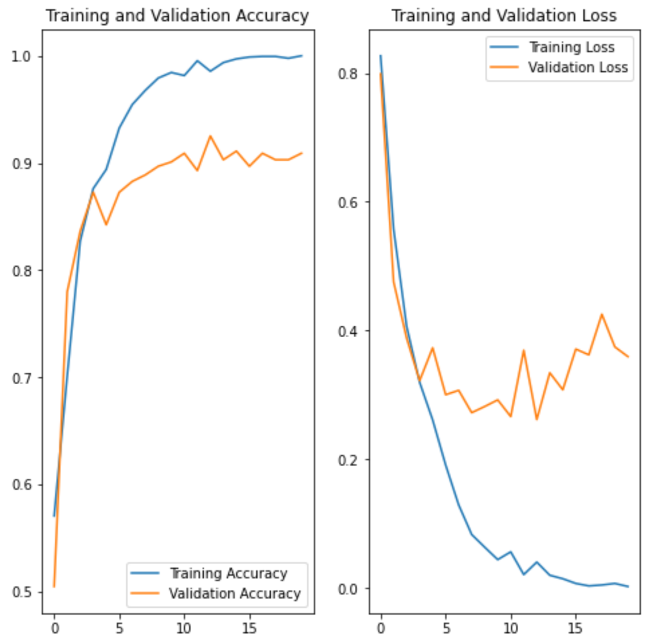


Fig 5. Model 1 Accuracy and Loss Graph

```
Training Accuracy:  0.9619815945625305
Validation Accuracy:  0.8808080554008484
```

2) **Model 2 (Dropout + CNN)**




```
Training Accuracy:  0.9855991005897522
Validation Accuracy:  0.9252524971961975
```

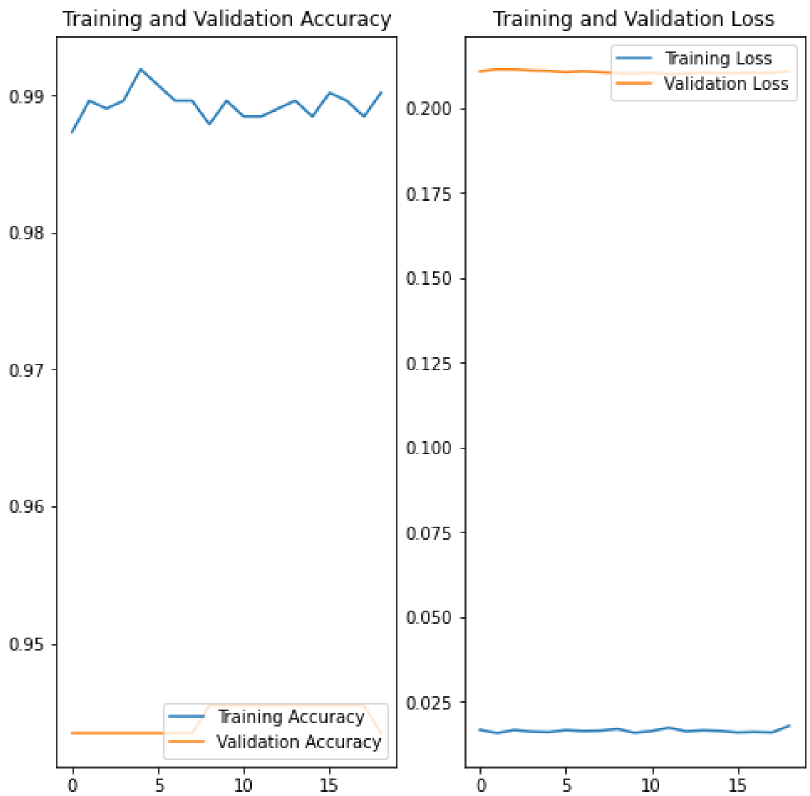
Fig 6. Model 2 Accuracy and Loss Graph

3) **Model 3 (Batch Normalization (Size 32) + CNN)**

Fig 7. Model 3 Accuracy and Loss Graph

Training Accuracy: 0.9873272180557251
Validation Accuracy: 0.9454545378684998

4) Model 4 (Batch Normalization (Size 64) + CNN)



Training Accuracy: 0.9884792566299438
Validation Accuracy: 0.9454545378684998

Fig 8. Model 4 Accuracy and Loss Graph

Larger Batch size shows almost same accuracy, but training took lesser time. Training accuracy 98.8% and Validation Accuracy 94.5% . Last model is used for evaluation on test dataset, and it shows accuracy of 92%, which can be considered as good.

V. CONCLUSION

Image datasets require spatial manipulation and learning of data. Convolutional networks work by applying functions over spatial regions; thus, they learn the local spatial data. The neighbourhood over which the function is applied is defined by the kernel size and the function itself is the kernel. The kernels are specialized matrices that can identify a specific feature in an input image. Thus, applying the kernel helps the model "identify" certain features in the image. These kernels start off with basic features such as edges and curves, the output of these kernels is worked upon by other kernels which combine them to identify even more complex features in the input image. Now to identify which kernels are suitable for the current data, deep learning neural networks are used. Deep learning networks continuously fine tune the kernels for each CNN layer during each iteration using the labelled training data. After the model is trained, the parameter for each filter is saved as weights and can be used to uniquely identify certain features in any input image.

CNNs apply many filter parameters which takes long training times. To overcome this, strategies such as Dropout, Batch Normalization and Larger batch sizes can be used. Dropout works by shutting down some neurons in each iteration thus reducing the number of trainable parameters while the other methods attempt to speed up the mathematical calculations.

From the above results, incremental addition of dropout and normalization strategies improved the model's accuracy by substantial steps during training which accumulated to a ~2-6% increase from 96% To 98.8% in training and 88% to 94.5% in validation accuracy and lower training times.

VI. REFERENCES

[1] Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W, Tao Q, Sun Z, Xia L. Correlation of chest ct and rt-pcr testing in coronavirus disease 2019 (covid-19) in China: a report of 1014 cases. Radiology 2020:200642.
[2] Ng M-Y, Lee EY, Yang J, Yang F, Li X, Wang H, Lui MM-s, Lo CS-Y, Leung B, Khong P-L, et al. Imaging profile of the covid-19 infection: radiologic findings and literature review. Radiology: Cardiothoracic Imaging 2020;2(1):e200034.
[3] Hemdan EE-D, Shouman MA, Karar ME. Covidx-net: a framework of deep learning classifiers to diagnose covid-19 in x-ray images. 2020. p. 11055. arXiv preprint arXiv:2003.
[4] Farooq M, Hafeez A. Covid-resnet: a deep learning framework for screening of covid19 from radiographs. 2020. p. 14395. arXiv preprint arXiv:2003.
[5] Li T, Han Z, Wei B, Zheng Y, Hong Y, Cong J. Robust screening of covid-19 from chest x-ray via discriminative cost-sensitive learning. 2020. p. 12592. ArXiv abs/ 2004.
[6] Abbas A, Abdelsamea M, Gaber M. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. medRxiv; 2020. https://doi.org/ 10.1101/2020.03.30.20047456.
[7] Wang L, Wong A. Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images. 2020. arXiv preprint arXiv:2003.09871.
[8] Luz E, Silva PL, Silva R, Silva L, Moreira G, Menotti D. Towards an effective and efficient deep learning model for covid-19 patterns detection in x-ray images. 2020. arXiv:2004.05717.
[9] He X, Yang X, Zhang S, Zhao J, Zhang Y, Xing E, Xie P. Sample-efficient deep learning for covid-19 diagnosis based on ct scans. medRxiv; 2020.
[10] Mobiny A, Cicalese PA, Zare S, Yuan P, Abavisani M, Wu CC, Ahuja J, de Groot PM, Van Nguyen H. Radiologist-level covid-19 detection using ct scans with detail-oriented capsule networks. 2020. arXiv preprint arXiv:2004.07407.
[11] Polsinelli M, Cinque L, Placidi G. A light cnn for detecting covid-19 from ct scans of the chest. 2020. p. 12837. arXiv preprint arXiv:2004.
[12] Amyar A, Modzelewski R, Ruan S. Multi-task deep learning based ct imaging analysis for covid-19: classification and segmentation. medRxiv; 2020.
[13] Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J, Cai M, Yang J, Li Y, Meng X, Xu B. A deep learning algorithm using ct images to screen for corona virus disease (covid19). medRxiv; 2020. https://doi.org/10.1101/2020.02.14.20023028.

[14] Soares E, Angelov P, Biaso S, Froes MH, Abe DK. Sars-cov-2 ct-scan dataset: a large dataset of real patients ct scans for sars-cov-2 identification. medRxiv; 2020.
[15] Zhao J, Zhang Y, He X, Xie P. Covid-ct-dataset: a ct scan dataset about covid-19. 2020. arXiv preprint arXiv:2003.13865.