1. Abdelghani ROUINI[1], 2. Messaouda LARBI[1], 3. Derradji BAKRIA[1], 4. Belkacem KORICH[1]

Applied Automation and Industrial Diagnostics Laboratory (LAADI), Faculty of Science and Technology University of Ziane Achour of Djelfa, 17000, Algeria (1)

ORCID: 1. 0000-0003-2133-1776; 2. 0009-0002-4367-6214; 3. 0000-0002-1741-9209; 4. 0009-0005-4904-6864

doi:10.15199/48.2024.04.29

A Neural Network Designed for COVID-19 Detection Using CT Images

***Abstract**. To quickly identify COVID-19 and stop its spread, Computer Tomography (CT) imaging of the chest is a reliable diagnostic method. In this work we implement a deep neural network based on Convolutional Neural Network (CNN), to effectively distinguish between healthy CT images and COVID-19 CT images. The final architecture allows for 91.6% accuracy, which is a 2% improvement over the first one. The result obtained can be very useful in medical diagnostics, particularly in the Covid-19 case. The implemented CNN model takes 2-3 seconds to run on a medium-end laptop without GPU acceleration.*

***Streszczenie.** Aby szybko zidentyfikować COVID-19 i zatrzymać jego rozprzestrzenianie się, niezawodną metodą diagnostyczną jest tomografia komputerowa (CT) klatki piersiowej. W tej pracy wdrażamy głęboką sieć neuronową opartą na konwolucyjnej sieci neuronowej (CNN), aby skutecznie rozróżniać zdrowe obrazy CT od obrazów CT COVID-19. Ostateczna architektura pozwala na dokładność na poziomie 91,6%, co stanowi poprawę o 2% w stosunku do pierwszej. Uzyskany wynik może być bardzo przydatny w diagnostyce medycznej, szczególnie w przypadku Covid-19. Zaimplementowany model CNN działa w ciągu 2-3 sekund na laptopie średniej klasy bez akceleracji GPU. (**Sieć neuronowa zaprojektowana do wykrywania COVID-19 za pomocą obrazów tomografii komputerowej**)*

Keywords: COVID-19, Deep Learning, Neural Network, CT scan.
Słowa kluczowe: COVID-19, głębokie uczenie się, sieć neuronowa, tomografia komputerowa..

Introduction

Since 2019, there has been an outbreak of viral pneumonia, of unknown origin in the Chinese city of Wuhan. The World Health Organization quickly named the potent virus coronavirus 2 (SARS-CoV-2), and the consequent respiratory condition was designated coronavirus disease 2019 (COVID-19). Global statistics confirm over 6 million fatalities and over 541 million confirmed cases so far, with the count continuing to rise [1, 2].

The reverse transcription-polymerase chain reaction (RT-PCR) is a prevalent approach for disease testing. It hinges on the utilization of nasopharyngeal swabs to detect the ribonucleic acid (RNA) of SARS-CoV, a widely used technique [3].

Aside from verification by pathogen laboratories, other valuable techniques for identifying COVID-19 encompass the analysis of clinical traits and the application of Computed Tomography (CT) scans.

CT imaging has been suggested as a crucial alternative method for screening COVID-19, and through this imaging technique, rapid predictions could be made in comparison to RT-PCR. [4, 5].

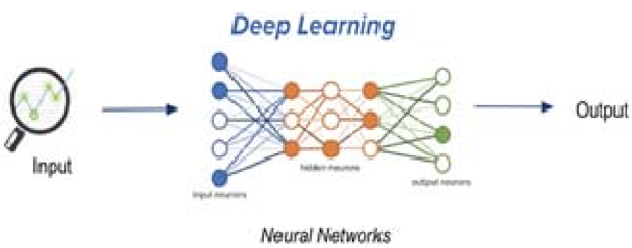


Fig.1. Deep Learning principal

Neural Network for CT images based COVID-19 detection

Artificial Intelligence comprises techniques and resources that empower machines to identify data patterns and employ this inherent structure to engage in problem-solving. Machines strive to comprehend these foundational patterns through various approaches. [6, 7].

Deep Learning [1, 8]. constitutes a subset of Machine Learning. It emphasizes the construction of extensive Neural Network models competent in decision-making. Essentially, Deep Learning involves Neural Networks featuring numerous concealed layers of neurons. Notably, augmenting the number of neural layers has been noted to significantly influence the resultant output quality. [9, 10].

Purposed system

Our proposed system is based on several operations, the entry process where the input is the medical images of the person to be examined, and then the system processes the image to classify it into two categories (Covid, Non-Covid) by giving the probability of each category.

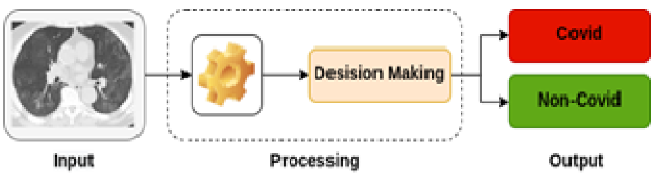


Fig.2. Purposed system

- **Practical part**

In the practical part, we have taken a number of major steps. numbered in the figure 3:

1. Download the dataset
2. Data pre-processing
3. Data spliting
4. Build the model
5. Train the model
6. Evaluate the model
7. If the model performance is bad we will Modify the model architecture and repeat the two steps 4,5.

Prepare and pre-process Dataset

The training set serves as the collection from which our model gains knowledge, while the validation set is employed to assess its performance. The testing set is then utilized to assess the ultimate performance of the model.

To achieve this, we generate two plots (Figure 4): one depicting positive cases of SARS-CoV-2 infection and the other showing non-infected cases with SARS-CoV-2; for this, we used PIL; then we load and pre-process the dataset (resize, convert to gray) by creating a function named data_generator to simplify loading the dataset.

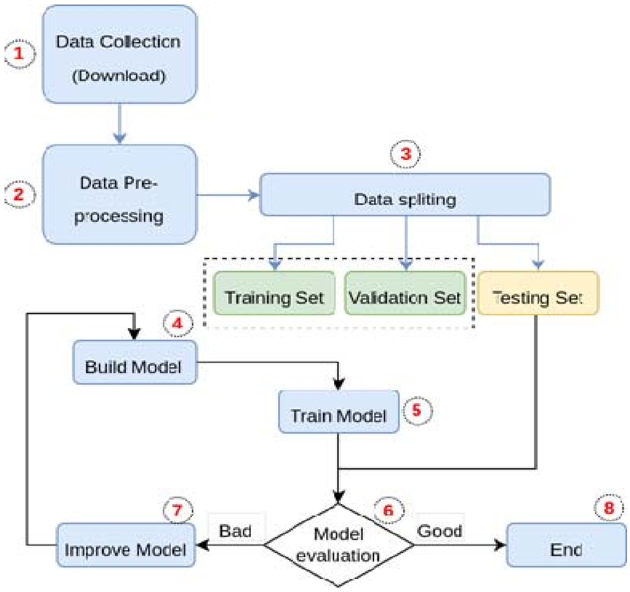
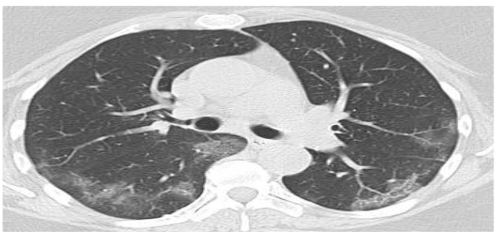


Fig.3. Practical part workflow

a)

```
covid = list(data_dir_train.glob('COVID/*'))
PIL.Image.open(str(covid[0]))
```



b)

```
[13] non_covid = list(data_dir_train.glob('non-COVID/*'))
     PIL.Image.open(str(non_covid[0]))
```

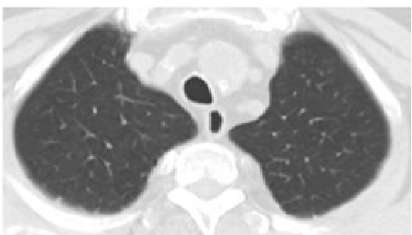


Fig.4. The training set, a) infected case, b) non-infected case

Building machine learning

The process of building a machine learning model passes through three main phases: First the definitions of the model, secondly compile it, and third is the training phase.

Define the model

We have formulated a model for the detection of COVID-19 in CT scan images. In the context of machine learning, the primary objective is image processing, and the deep learning architecture deemed suitable for this particular task is the Convolutional Neural Network (CNN).

Initially, we incorporate an image rescaling layer, commonly referred to as image normalization, as an introductory step. This particular layer is responsible for the rescaling and adjustment of pixel values within a batch of images, thereby transforming them from the initial [0, 255] range to the normalized [0, 1] range.

Subsequently, we introduce three sequential combinations of a Convolutional layer followed by a max-pooling layer. These combinations serve to effectively extract pertinent features from the input data.

Following the feature extraction stage, a flatten layer is introduced to convert the multidimensional volumes produced by the aggregation of feature maps into a singular continuous linear vector. This vector is then employed as input to the fully connected layer, also known as the dense layers, which are responsible for further processing and classification of these features.

Lastly, we incorporate two dense layers in a final step to facilitate the classification of extracted features.

The Figure below shows our model architecture:

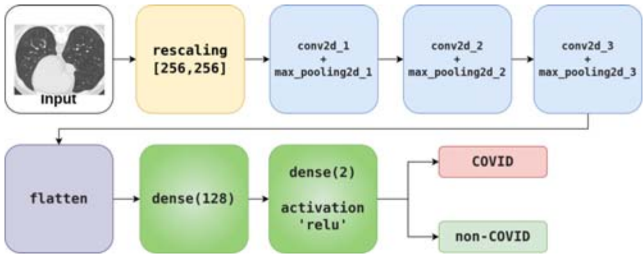


Fig.5. Diagram Architecture of our deep learning model

Compile the model

Prior to preparing the model for the training phase, several additional configurations require attention. These configurations are introduced during the model compilation stage.

First is Loss Function, this metric serves to quantify the model's accuracy during the training process. For this purpose, we have selected the "SparseCategoricalCrossentropy" loss function. The second is the optimizer: The optimizer is responsible for updating the model parameters based on the provided data and its associated loss function. In this context, we have opted for the Adam optimizer. The last is Metrics which are employed to track the progress of both training and testing phases. In this instance, accuracy will be utilized.

Train the model

The last step in building the model is called "training" the neural network. During this process, the values of the weights are adjusted.

To begin, the Convolutional Neural Network (CNN) is initialized with random weights. As the CNN undergoes training, it is provided with an extensive dataset of images, each labeled with its corresponding class (COVID or non-COVID). The CNN processes each image with its initial random values and subsequently contrasts the outcome with the input image's class label.

In situations where the produced output doesn't align with the expected class label (often occurring at the early stages of training) the CNN initiates minor weight adjustments in its neurons. This correction aims to bring the output into agreement with the class label of the image, ensuring accurate classification.

Following 19 epochs of training, we achieve a training accuracy of 0.9614 and a validation accuracy of 0.8909. The learning curve observed during the training process (figure 6) signifies the model's proficient performance on both the training dataset and the validation set.

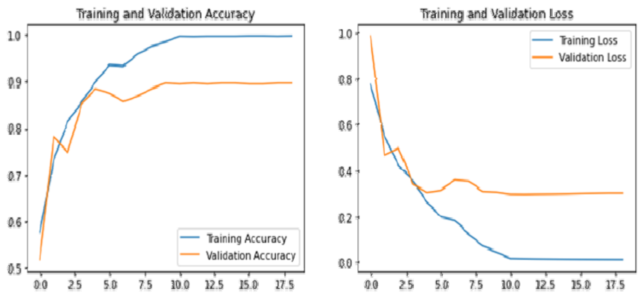


Fig.6 Model1 learning curve

Evaluation

Having concluded the model training, we proceed to assess its accuracy using a distinct test dataset. This test dataset comprises labelled images that were entirely excluded from the training phase.
For each image in this test dataset, it undergoes processing by the CNN model, and the resultant output is contrasted with the actual class label of the respective test image.
The test dataset effectively gauges the predictive capability of the CNN model on unseen data.

Improve the model accuracy

In this part, we will make some modifications to the model architecture to improve the results.

Initially, in the realm of supervised machine learning, a prominent concern arises known as over fitting. This issue emerges due to the presence of over fitting, causing the model to exhibit exceptional performance on the training set, yet displaying subpar fitting capabilities on the testing set. [11, 12]. Among the popular and effective techniques against over fitting in neural networks used to avoid over fitting is Dropout.

The fundamental concept behind dropout involves the intentional random elimination of units and their associated connections within neural networks during the training process. This strategy serves to hinder excessive coadaptation among units.

After adding a dropout layer, the model will be as shown in the following Figure:

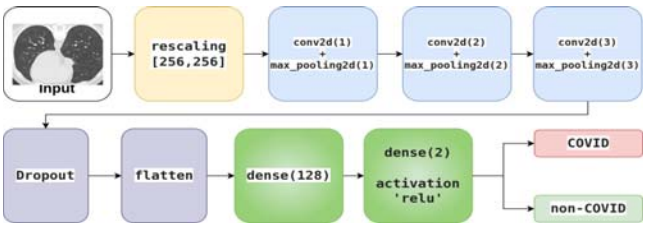


Fig.7. 2nd Model after adding a dropout layer

Now we compile and train the model again, after the training has been completed, we get a training accuracy 0.9487 and a validation accuracy of 0.8949. We note that the validation accuracy increase.

We note that the accuracy of the model decreased by 2% (from 0.8960 to 0.8760). This means that this step does not improve the model. The learning curve depicted in Figure 8 demonstrates the model's performance on both the training dataset and the validation set throughout the training process.

We will try again by adding a batch normalization layer after each convolution layer, Using batch normalization makes the network more stable during training. And accelerating Deep Network Training. The model will look like the Figure 9.

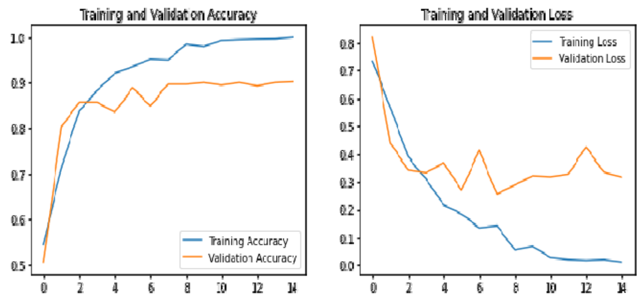


Fig.8. 2nd Model (with dropout layer) learning curve

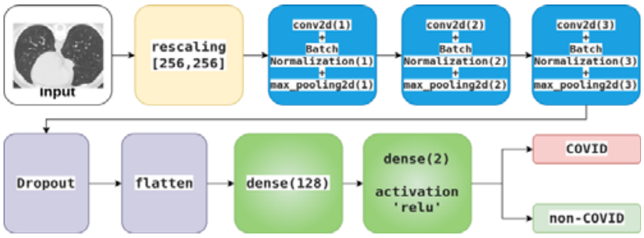


Fig.9. 3rd model after adding batch normalization layers

Upon the completion of the training process, we attain a training accuracy of 0.9988 and a validation accuracy of 0.9494. It is noteworthy that both the training and validation accuracies indicate an improvement in model performance.

We use a test dataset to verify its accuracy again, We note that the accuracy of the model increased by 2% (from 0.8960 to 0.9160). This means a significant improvement in the model's performance.

The learning curve illustrated in Figure 10 illustrates the model's performance across both the training dataset and the validation set over the course of the training process.

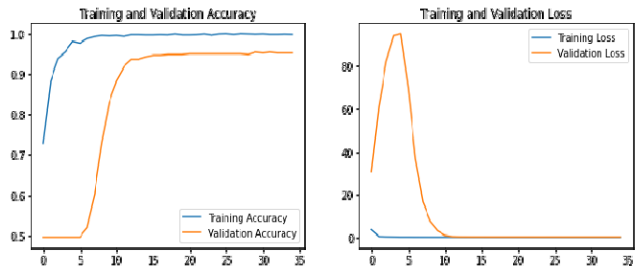


Fig.10 3rd model (with batch normalization layers) learning curve

Results

After we complete the model training, we will create a function that receives a CT image, and then predicts the infection rate of its owner with the Corona virus. The name of the function is make_predaction(), it receives the path of the image and gives the output probability of infection. We will test the function with images from the test set that were not used before in the training process:

```
pr = make_predaction('/content/splited_sarscov2-ctscan-dataset/test/COVID/Covid (834).png',True)
print(class_name[0],': ',pr[0])
print(class_name[1],': ',pr[1])
```

```
Found 1 files belonging to 2 classes.
COVID :  0.9966795236431453
non-COVID :  0.003320476356854742
```

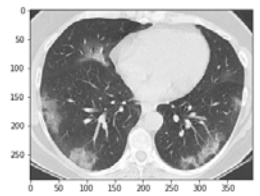


Fig.11. Prediction 1, result 99.66% Covid – 0.33% non-Covid

Fig.12. Prediction 2, result 57.35% Covid – 42.64% non-Covid

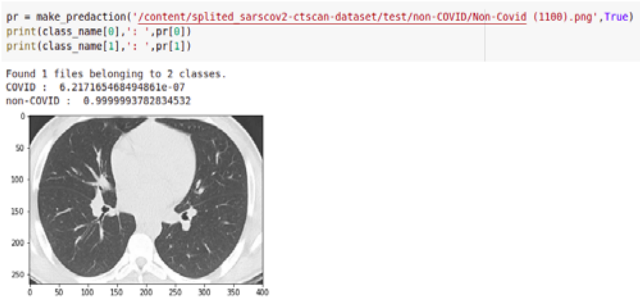


Fig.13. Prediction 3, result ~0.01% Covid – 99.99% non-Covid

Fig.14. Prediction 4, result ~0.01% covid – 99.99% non-COVID

Conclusion

The CNN network models were developed in this study using 2482 CT scans from real patients in hospitals in Sao Paulo, Brazil, to compare the efficiency of the neural network model for rapid prediction of COVID-19 using CT imaging and RT-PCR, which is an important alternative tool for COVID-19 screening. In the dataset used, the CNN network analyses each image with randomly assigned values and then compares them to the class label of the input image. After 19 epochs of training, the model has a training accuracy of 0.9614 and a validation accuracy of 0.8909, indicating that it performs well on both the training and validation datasets.

The dataset test analyzes the prediction performance of the CNN model by completing the model's training and using a test dataset to validate its correctness. The CNN model's performance was 0.896, according to the data. After making some changes to the model architecture to improve the results, the training accuracy was 0.9487 and the validation accuracy was 0.8949. As a result, both training and validation accuracy improve. as a consequence, the model's accuracy reduced by 2% (from 0.8960 to 0.8760). This signifies that this step has no effect on the model.

After each convolution layer, using batch normalization makes the network more robust during training. In addition, Deep Network Training is advancing, with a training accuracy of 0.9988 and a validation accuracy of 0.9494. Both training and validation accuracy improve, resulting in a 2% improvement in model accuracy (from 0.8960 to 0.9160). This indicates a significant improvement in the model's performance, From that the function was tested with images, showing great results

***Authors**: Dr. Abdelghani ROUINI, E-mail :Rouinia@yahoo.fr, Dr.Messaouda LARBI,E-mail:larbi_messaouda@yahoo.fr, Dr . Derradji BAKRIA, E-mail: bakriaderradji@gmail.com, Dr .Belkacem KORICH, E-mail: kor_belkacem@yahoo.fr,*
Applied Automation and Industrial Diagnostics Laboratory (LAADI), Faculty of Science and Technology University of Ziane Achour of Djelfa, 17000, Algeria*)*

REFERENCES

[1] Applications to Combat Novel Coronavirus (COVID-19) Pandemic. SN COMPUT. SCI. 1, 363 (2020).
[2] Beck BR, Shin B, Choi Y, Park S, Kang K. Predicting commercially available antiviral drugs that may act on the novel coronavirus (SARS-CoV-2) through a drug-target interaction deep learning model. Comput Struct Biotechnol J. 2020;18:784–90.
[3] Bernheim, A., Mei, X.Y., Huang, M.Q., Yang, Y., Fayad, Z.A., Zhang, N., Diao, K.Y., Lin, B., Zhu, X.Q., Li, K.W., Li, S.L., Shan, H., Jacobi, A., Chung, M. (2020). Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection. Radiology, 295(3): 685-691.
[4] Tinnathi, S., Sudhavani, G. (2021). An efficient copy move for gery detection using adaptive watershed segmentation with AGSO and hybrid feature extraction. Journal of Visual Communication and Image Representation, 74: 102966.
[5] Wu, Y.H., Gao, S.H., Mei, J., Xu, J., Fan, D.P., Zhang, R.G., Cheng, M.M. (2021). JCS: An explainable COVID-19 diagnosis system by joint classification and segmentation. In IEEE Transactions on Image Processing, IEEE, 30: 3113-3126.
[6] Paules, C.I., Marston, H.D., Fauci, A.S. (2020). Coronavirus infections-more than just the common cold. Jama Network, 323(8): 707-708.
[7] Shi, F., Wang, J., Shi, J., Wu, Z.Y., Wang, Q., Tang, Z.Y., He, K., Shi, Y.H., Shen, D.G. (2020). Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for COVID-19. In IEEE Reviews in Biomedical Engineering, IEEE, 14: 4-15.
[8] Chung, M., Bernheim, A., Mei, X.Y., Zhang, N., Huang, M.Q., Zeng, X.J., Cui, J.F., Xu, W.J., Yang, Y., Fayad, Z.A., Jacobi, A., Li, K.W., Li, S.L., Shan, H. (2020). CT imaging features of 2019 novel coronavirus (2019- nCoV). Radiology, 295(1): 202-207.
[9] Huang, Y.P., Singh, P., Kuo, H.C. (2020). A hybrid fuzzy clustering approach for the recognition and visualization of MRI images of Parkinson's disease. In IEEE Access, IEEE, 8: 25041-25051.
[10] Moftah, H.M., Azar, A.T., Al-Shammari, E.T., Ghali, N.I., Hassanien, A.E., Shoman, M. (2014). Adaptive kmeans clustering algorithm for MR breast image segmentation. Neural Computing and Applications, 24: 1917-1928.
[11] Xu, R., Wunsch, D. (2005). Survey of clustering algorithms. In IEEE Transactions on Neural Networks, IEEE, 16(3): 645- 678.
[12] Zhu H, Guo Q, Li M, Wang C, Fang Z, Wang P, Tan J, Wu S, Xiao Y. Host and infectivity prediction of Wuhan 2019 novel coronavirus using deep learning algorithm. BioRxiv. 2020.