ADITUM
Enrich your Research

International Journal of Interventional Radiology and Imaging

Research Article

Automatic detection of COVID-19 from Chest CT Scans Through Deep Learning

Roberto Rodríguez* and Laura Brito
Institute of Cybernetics, Mathematics & Physics (ICIMAF), Havana 10 400, Cuba

Article Info

Received: April 10, 2025
Accepted: April 20, 2025
Published: April 24, 2025

***Corresponding author:** Roberto Rodríguez, Institute of Cybernetics, Mathematics & Physics (ICIMAF), Havana 10 400, Cuba.

Citation: Rodríguez R, Brito L. (2025) “Automatic detection of COVID-19 from Chest CT Scans Through Deep Learning.” J Clinical Cardiology Interventions, 3(1); DOI: 10.61148/3065-6702/IJIRI/030

Abstract

In order to mitigate the spread of COVID-19 and initiate patient isolation to reduce its effect, early diagnosis was required, with real-time reverse transcriptase-polymerase chain reaction (RT-PCR) swab test being the most commonly used. However, RT-PCR testing was sometimes found to be lengthy and inaccurate and, in many cases of severe complications of COVID-19 pneumonia, chest CT scans was preferred. In this paper, we conducted a study for the detection of COVID-19 from computed tomography images and compared the obtained results using deep learning (DL) with a machine learning algorithm (support vector machine (SVM)). The obtained results using deep learning for COVID-19 prediction were outstanding and promising. However, in the presence of small databases, the accuracy of COVID-19 prediction when employing a machine learning technique (SVM) was slightly higher than that of deep learning. In addition, the run time with SVM was also shorter.

Keywords: Deep learning; supervised learning; convolutional neural networks; support vector machines; training; neural network architectures

Introduction

COVID-19 was a pandemic that spread around the world, killed millions of people, caused severe damage to the economies of all countries, and put the entire healthcare system under enormous pressure. In order to mitigate the spread of COVID-19 and initiate patient isolation to reduce its effect, early diagnosis was required, with real-time reverse transcriptase- polymerase chain reaction (RT-PCR) swab test being the most commonly used [1]. However, RT-PCR diagnosis often requires retesting because it has a high false negative rate and takes many hours to complete [2]. In that sense, for many cases of severe complications of COVID- 19 pneumonia, chest computed tomography proved to be an alternative method to visualize thoracic lesions, being more adequate for the detection of COVID-19 than chest radiography, but at the same time, it is slower, more expensive and not always available, especially in economically underdeveloped countries.

Face to face to COVID-19 stimulated many researchers and scientific institutions in the world to search effective methods and techniques that helped put end to this pandemic. In that direction, the computer vision community did not lag behind and many papers were published to address this disease, using mainly X-ray and CT images [3, 4, 5]; and many researchers proven that chest computed tomography was more effective and sensitive in detecting COVID-19 than RT-PCR tests [6]. Then, based on radiographic changes of COVID-19 in CT images, these studies evidenced that machine learning methods might be able to extract specific features of COVID-19 and provide a clinical diagnosis prior to RT- PCR testing, saving significant time for disease control [7]. On the other hand, COVID-19 remains a global public health challenge due to new immune-evasive SARS-CoV-2 variants continue to emerge. For that reason, any automated system to detect COVID-19 from CT images will always be welcome in the medical field, as automated analysis of biomedical images has been proven to reduce the workload of radiologists and pathologists; in addition to offering accurate and faster diagnoses.

Many methods of machine learning have been proposed for bio-medical image analysis [8, 9], and among them, deep learning techniques have occupied a prominent position [10, 11], since these algorithms extract image features automatically and providing accurate diagnoses. Furthermore, deep learning is known to be classified into supervised and unsupervised learning, with supervised learning giving exceptional results in bio-medical image processing, with performance comparable to that of humans, and sometimes superior [12]. Supervised learning requires a set of real data (ground truth) and prior knowledge about the result to be obtained with that dataset, and a sufficient amount of training samples is required when working with algorithms based on deep learning. In many cases, as is in medical data analysis, specialists lack a suitable large dataset, and when a sufficiently large database is not available to carry out a good training on the neural network, the obtained result may be far from the expected one. Therefore, some traditional machine learning methods should not be discarded, especially those that have proven to be efficient and do not require large databases.

In this paper, we carried out a study for automatic detection of COVID-19 from chest CT scans, and compared the obtained results using deep learning (DL) and support vector machine (SVM). The obtained results by using deep learning for COVID-19 prediction were outstanding and promising. However, in the presence of small databases, the prediction accuracy of COVID-19 when using a machine learning technique (SVM) was very similar to that of deep learning. In addition, the execution time with SVM was also shorter.

The rest of the paper is organized as follows: In section 2, the materials and methods are given, and we slightly outline some theoretical and algorithmic aspects. Here, we will specify on the database used. Section 3 contains the obtained results and discussion. We will describe our conclusions in section 4.

Materials and methods

We collected chest CT scan images containing two classes: COVID and non-COVID from a Hospital selected to admit patients with disease symptom, which were used to train the proposed models. Examples of some of these images is shown in Figure 1.

In our study, each class contains 600 COVID-19 positive individuals and 600 negative individuals (non-COVID-19). We did not use some method to augment the database (e.g., horizontal flipping or gamma correction [4]), since the goal of this research was to test and compare the performance of deep learning against a machine learning technique.

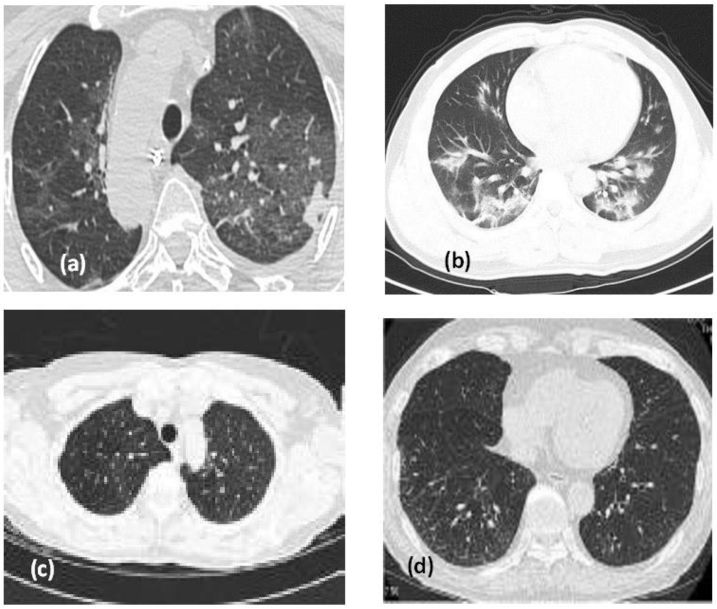


Figure 1: Sample images from CT dataset. Patients with COVID, (a) and (b). Non-COVID, (c) and (d).

We subsequently resized the entire database to a dimension of 100x100 pixels. We resized the database to a dimension of 100x100 pixels, and each image was normalized by dividing the value of each pixel by the maximum possible value.

Proposed Convolutional Neural Network

In Figure 2, we show the workflow of our CNN model, which we focused on detecting COVID or non-COVID features from chest radiographic images and contains multiple blocks, such as convolution layer, pooling layer, activation function and fully connected layers that can extract trainable spatial features adaptively using back-propagation algorithms [13].

Dropout and max-pooling layers do not represent in this architecture, since these do not contain parameters to be trained. However, after the max-pooling layer and first fully connected layer a dropout layer there is. The pooling layer reduces the number of parameters and filters out only useful features; while a fully connected layer creates a combination of one or more layers to convert the features into a one-dimensional matrix or vector.

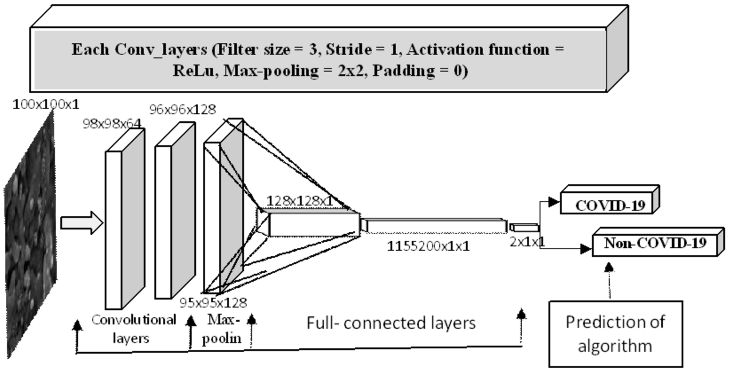


Figure 2: Architecture of CNN model

Tuning of trained models. Transfer learning

Today, what is done is to use a previously trained architecture and apply it to the solution of our practical problem by using transfer learning. Transfer learning is a procedure that trains a model by using knowledge from previous classification task (and in accordance with similar goals) to implement a new one. The transfer learning procedure initializes the previously trained weights to ensure better learning over the new dataset [14]. In this procedure, it becomes necessary to change the output of the last layer of the previously trained model by the number of classes of the new classification task. Here, the interesting being in this procedure is that the model that was previously trained can be retrained on all layers (convolutional layers, pooling layers and fully connected layers) [4].

We implemented different metrics to measure and quantitatively evaluate the performance of the learning process and the predictive power of the models. These were the following: accuracy, recall, F1 score, confusion matrix and precision [15].

Learning mode

A usual problem in training deep learning models is overfitting, which produces a good behavior in its training set (high performance) and poor performance in another dataset. In that sense, we to mitigate this effect we used mini-batch training, which too offers the advantage of guaranteeing faster convergence [16]. We used a mini-batch size equal to 10.

It is important to point out, that setting the initial values should be treated carefully because it might have significant effects on the following learning steps. We randomly initialized the network weights, and the samples were drawn from a uniform distribution in the interval [-l, l], where l was defined as [17],

$$l = \sqrt{\frac{6}{fin+fout}}, \qquad (1)$$

where, fin is the number of input units in the weight's matrix and fout is the number of output units. The idea is that the weights start with small values to avoid saturation and slowdown in network training. However, if they are all set to zeros, the initial outputs from the network become zeros and good-for-nothing for the following steps. If, on the other hand, they are all set to a same constant, they act like just one neuron regardless of the number of nodes and neurons in the network. In any case, learning cannot be adequately improved. For that reason, one should choose different values in a reasonable range, where the choice of the initial values depends on activation functions in the network.

We addressed an L2 regularization technique with weight decay and a coefficient 0.001 on the convolutional layers, which strongly limited the obtaining of large weights. One should use L2 regularization when one is less concerned about creating a space network and one wants to configure lower weights. The lower weight values will typically lead to less overfitting [18].

We used as optimization algorithm the Stochastic gradient descent (SGD), which is a standard procedure widely used to solve optimization problems in neural networks, offering very good results. It works very similarly to Batch/Mini-Batch training, except that the batches are made up of a random set of training elements. Since the neural network is trained each time with a random sample of the entire training set, the error does not decrease gradually. However, it usually decreases [19].

An important hyper-parameter in the training of a neural network is the learning rate, which is a key parameter whose proper adjustment can help to obtain the desired performance, since it determines how much the weights are adjusted in each iteration of the algorithm, as well as serving as a regularization mechanism. In this work, we use an adaptive technique for CNN training, which is called Adaptive moment estimates (Adam). Adam estimates the first (mean) and second (variance) moments to determine the weight corrections [20]. Adam starts with an exponentially decreasing average of past gradients (m),

$$mt = \beta 1\, m(t{-}1) + (1-\beta 1)gt \qquad (2)$$

This average serves a similar purpose as the classical moment update; however, its value is automatically calculated based on the current gradient (gt). The update rule then calculates the second moment (vt):

$$vt = \beta 2 v(t{-}1) + (1-\beta 2)g2 \qquad (3)$$

The mt and vt values are estimates of the first moment of the gradients (the mean) and the second moment (the uncentered variance). However, they will be strongly biased toward zero in the initial training cycles. The first moment's bias is corrected as follows,

$$\hat{m}t = \frac{mt}{1-\beta t}, \qquad (4)$$

1

Similarly, the second moment is also corrected as follows,

$$\hat{v}t = \frac{vt}{1-\beta t}, \qquad (5)$$

These bias-corrected first and second moment estimates are applied to the ultimate Adam update rule, that is,

$$xt = x(t{-}1) - \frac{\propto \hat{m}t\ \hat{m}t}{\sqrt{\hat{v}t+\eta}} \qquad (6)$$

where, α is an initial learning rate which in this case was set to 0.001, η is used to avoid divisions by zero which was assigned the value 10−8 , and β1=0.9, β2=0.999 are other constants. These values were taken for these parameters according to criteria appeared in [20]. Furthermore, this publication ([20]) states that this method is computationally efficient, requires little memory, is invariant to the diagonal scale change of the gradient, and is suitable for problems that are large in terms of data/parameters.

Another regularization mechanism we used was the Dropout. Although dropout works differently than L1 and L2, it accomplishes the same goal, the prevention of overfitting. However, the algorithm performs the task by eliminating neurons and connections, at least temporarily. Unlike L1 and L2, no weight penalty is added. Dropout does not directly seek to train small weights. Dropout works by causing hidden neurons in the neural network to be unavailable during part of the training. Dropping a portion of the neural network allows the remaining trained portion to achieve a good score even without the dropped neurons. This technique decreases the co-adaptation between neurons, resulting in less overfitting [21].

Most neural network frameworks implement dropout as a separate layer. These layers function like a normal, densely connected neural network layer. The only difference is that the dropout layers periodically drop some of their neurons during training. We added dropout layers that made the training process efficient, creating a good relationship between training and model validation accuracy. In our case, two dropout layers were used with the parameter p=0.25, this value being the probability that a selected neuron remains active.

Experimental results and analysis

We developed a Python program to carry out the experiments to make predictions for COVID and non-COVID patients. In addition, we used cross-validation to split the database into training, testing and validation set. We communicated with TensorFlow using Keras, which allowed us to specify the number of hidden layers and create the neural network. Keras is a higher-level abstraction for neural networks that one builds upon TensorFlow.

In order to carry out a quantitative comparison of the results obtained by using deep learning, we implemented -in Python-, a classic and efficient machine learning technique, the support vector machine (SVM). In this case, we used the AutoML system (AutoSklearn) based on Keras, which it is an Automatics Machine Learning that attempts to use machine learning to automate itself. In other words, data is passed to the AutoML application in raw form, and models are automatically generated.

One aspect of great importance in deep learning is the learning rate, and this a crucial concept for backpropagation training. Setting the learning rate can be complex by two aspects: 1) too low a learning rate will usually converge to a reasonable solution; but the process might be prolonged, and 2) too high a learning rate will either fail outright or converge to a higher error than a better learning rate. Common values for learning rate are: 0.1, 0.01 and 0.001. We used a learning rate with a value of 0.001.

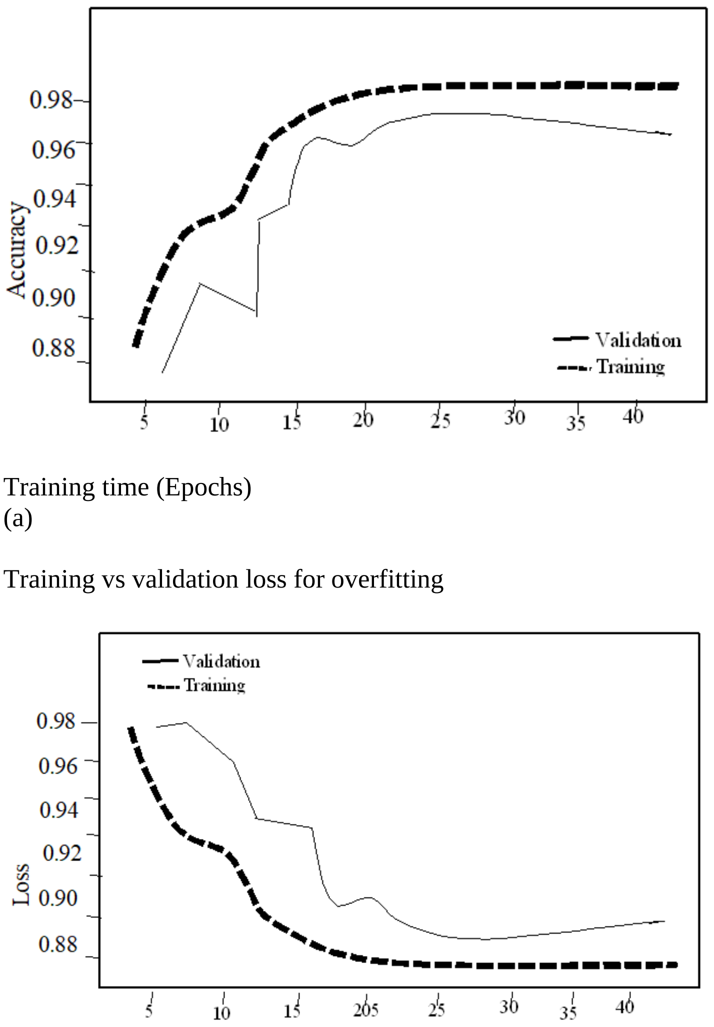
Since it is possible to calculate the gradient for a training set element, where these gradients can also be summed in each batch, updating the weights once per batch, we addressed the mini-batch training technique, which is widely used and often in the 32-64 element range. In this study, we used a batch size of 50 and 100x 100 size input images.

The above parameters made the training process efficient, creating a good relation between training and model validation precision [22]. In this work, we followed a procedure very similar to the one shown in [12].

It is possible to use cross-validation for a variety of purposes in predictive models. For example; generating out-of-sample predictions from a neural network, estimate a good number of epochs to train a neural network for early stopping and evaluate the effectiveness of some hyperparameters, such as activation functions, neuron counts, and layer counts, among other.

However, to try out each of these hyperparameters one will need to run train neural networks with multiple settings for each hyperparameter, and it was possible to note that neural networks often produced somewhat different results when trained multiple times. This is because the neural networks start with random weights. Because of this it is necessary to fit and evaluate a neural network time to ensure that one set of hyperparameters are actually better than another. Bootstrapping can be an effective means of benchmarking (comparing) two sets of hyperparameters

Many times, it can be difficult to determine how many epochs to cycle through to train a neural network. However, overfitting will occur if you train the neural network for too many epochs, and the neural network will not perform well on new data, despite attaining a good accuracy on the training set. Overfitting occurs when a neural network is trained to the point that it begins to memorize rather than generalize. Figure 3 shows the learning curves of proposed model for overfitting, where some fluctuations (random spikes) can be observed in the validation as the epochs advance with training. This is indicative of some overfitting and that the neuron weights were not uniformly adjusted in the validation process.



Training vs. validation error for overfitting

Training time (Epochs)
(a)

Training vs validation loss for overfitting

Training time (Epochs)
(b)

Figure 3. Learning curves of proposed model. (a) Accuracy error, (b) Loss function.

However, despite the lack of uniformity in learning (occurrence of some peaks at certain epochs), these peaks decreased in magnitude as the epochs progressed, indicating an adequate learning process. Here, the important issue is to split the original dataset into several datasets; that is, in a training set, in a validation set and, in a holdout set, which can be construct in several different ways. In many cases, the performance of a model can be evaluated by graphical analysis, which often does not provide accurate evidence by taking a single metric. For this reason, it is necessary to use other evaluation metrics (accuracy, recall, F1 score, etc.) to perform a more in-depth comparison of models.

3.1 A comparison of the obtained results with CNNs and Support Vector Machine

It is known that for tabular data, neural networks often do not perform significantly better that different than other models, such as: Support Vector Machines. In addition, when one applies to relatively low-dimensional tabular data tasks, deep neural networks do not necessarily add significant accuracy over other model types. However, at present most state- of-the-art solutions depend on deep neural networks for video, audio, text and image data.

In this case our database was not very unbalanced as in [12], but we proceeded in the same way. We selected the SVM method to carry out the comparison because it has proven its effectiveness and it was necessary to compare the obtained results with CNN with a classical machine learning method.

In Table I, we show the obtained results from the evaluation metrics for the proposed CNN model, while Table II shows the results of the evaluation metrics using the SVM model.

<table>
<tr><td></td><td colspan="2">Classes</td></tr>
<tr><td>Metrics</td><td><i>COVID-19</i></td><td><i>Non-COVID-19</i></td></tr>
<tr><td>Precision</td><td>0.9233</td><td>0.9261</td></tr>
<tr><td>Recall</td><td>0.9305</td><td>0.9261</td></tr>
<tr><td>F1-score</td><td>0.9268</td><td>0.9261</td></tr>
<tr><td>Confusion Matrix</td><td colspan="2"><table>
<tr><td></td><td></td><td colspan="2">Predicted</td></tr>
<tr><td></td><td></td><td><i>COVID-19</i></td><td><i>Non-COVID-19</i></td></tr>
<tr><td>True</td><td><i>COVID-19</i></td><td><b>241</b></td><td>18</td></tr>
<tr><td></td><td><i>on- COVID-1</i></td><td>20</td><td><b>251</b></td></tr>
</table></td></tr>
<tr><td>Accuracy</td><td colspan="2">0.9283</td></tr>
<tr><td>ing time (min per e</td><td colspan="2">20</td></tr>
</table>

Table I: Results of the evaluation metrics for the proposed CNN model

<table>
<tr><td></td><td colspan="2">Classes</td></tr>
<tr><td>Metrics</td><td><i>COVID-19</i></td><td><i>Non-COVID-19</i></td></tr>
<tr><td>Precision</td><td>0.9264</td><td>0.9340</td></tr>
<tr><td>Recall</td><td>0.9224</td><td>0.9340</td></tr>
<tr><td>F1-score</td><td>0.9243</td><td>0.9340</td></tr>
<tr><td>Confusion Matrix</td><td colspan="2"><table>
<tr><td></td><td></td><td colspan="2">Predicted</td></tr>
<tr><td></td><td></td><td><i>COVID-19</i></td><td><i>Non-COVID-19</i></td></tr>
<tr><td>True</td><td><i>COVID-19</i></td><td><b>214</b></td><td>18</td></tr>
<tr><td></td><td><i>Non-COVID-19</i></td><td>18</td><td><b>255</b></td></tr>
</table></td></tr>
<tr><td>Accuracy</td><td colspan="2">0.9287</td></tr>
<tr><td>Training time (min per epoch)</td><td colspan="2">0.8</td></tr>
</table>

Table II. Results of the evaluation metrics for the SVM model

From Tables I and II and considering the size of the databases of patients with COVID-19 and without COVID-19, we can carry out a deeper analysis of the obtained results. For example, it is evident (as it was pointed out) that when the database is small machine learning models do not learn well, which was in correspondence by the number of false positives and negatives that were classified by the models (see the confusion matrix). It should be kept in mind that the correctly classified samples are those that appear on the diagonal.

Here, the interesting about these results is that when the database is small or very unbalanced, the DL model learns less than the SVM model, which it is similar result we obtained in [12]. Note that the false positives (FP) classified by the DL model were slightly higher, which is not a symptom of inferiority of the DL model, since we are in the presence of small databases. Our interest in this comparison is to analyze that in many cases the most advanced technique in the state of the art is applied blindly without prior study of the data. This often leads to the underestimation of already established machine learning models (as is the case with SVM, for example). On the other hand, it is a fact that the larger the database, the more

the neural network learns, but also the more time is required for training.

It is known that accuracy tends to hide classification errors in database classes with fewer elements, since these classes have little weight compared to other classes in larger databases. For such a reason, one should direct the analysis by taking other metrics to make the study more accurate in validating the performance of a model. For example, by taking the F1 score, which is the harmonic mean between precision and recall, it is observed, in Table II, that there is a tendency to a higher value for the No-COVID class in the SVM model.

In Figures 4. 5 and 6, we show three examples of false positive and false negative patient classification performed by both models in predicting the COVID pandemic

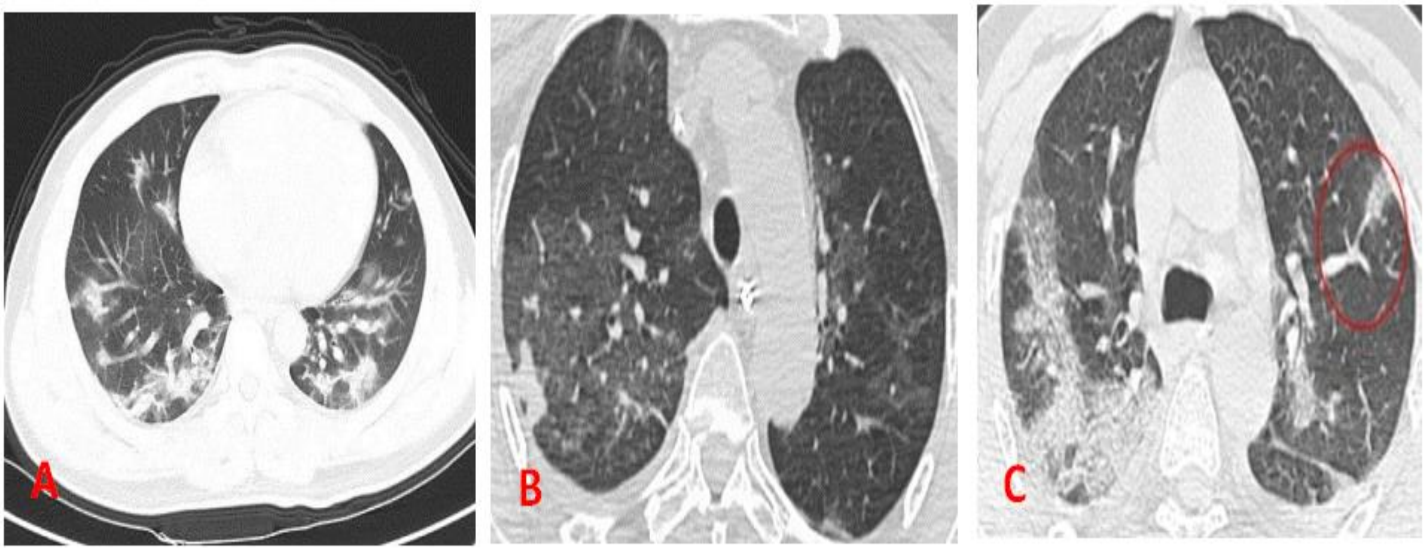


Fig. 4. The three images represent chest CT scans of COVID-19 positive individuals. The classification of the DL model is for (a) negative, for (b) positive and for (c) negative. The classification by the SVM model is for
(a) positive, for (b) positive and for (c) negative.

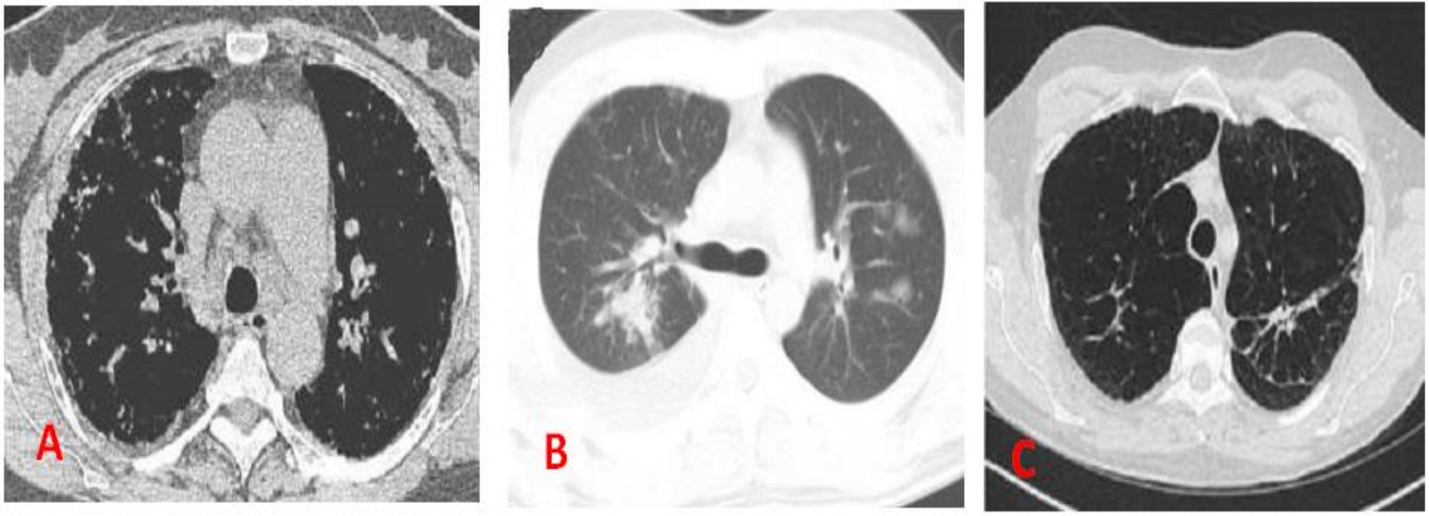


Fig. 5. The three images represent chest CT scans of COVID-19 negative individuals. The classification of the DL model is for (a) positive, for (b) negative and for (c) negative. The classification by the SVM model is for
(a) negative, (b) negative and (c) negative.

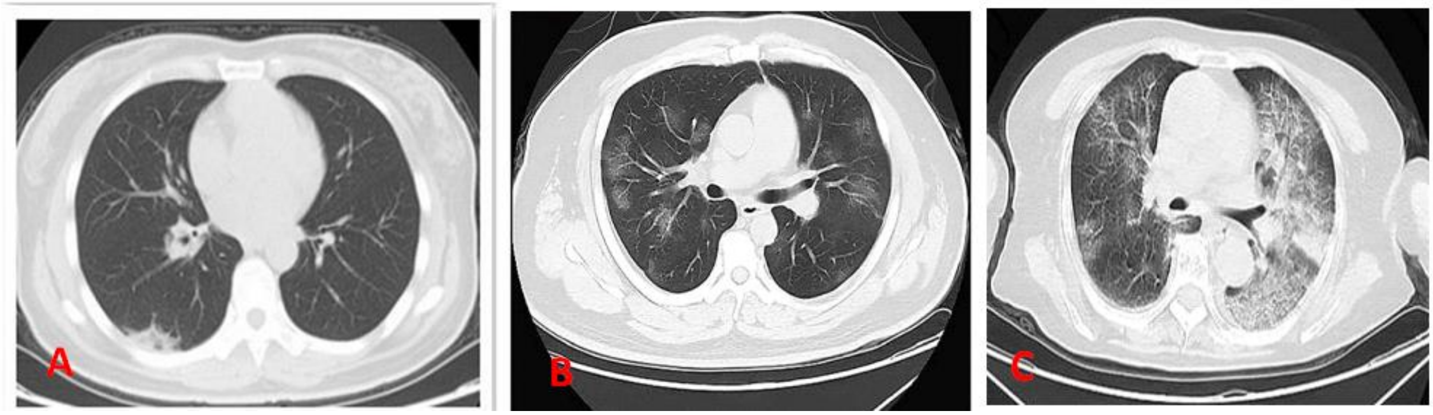


Fig. 6. The three images represent chest CT scans of COVID-19 positive individuals. The classification of the DL model is for (a) positive, (b) positive and (c) positive. The classification by the SVM model was exactly equal.

We do not intend to draw definitive conclusions from these results. Our objective was to know the performance of both models in terms of prediction accuracy according to existing databases of COVID and non-COVID patients. Note that the databases were used as they were without enlarging their size through affine transformations. The obtained results showed that in presence of small databases, established machine learning methods cannot be completely discarded. For example, the training time of the SVM model was much lower than the DL model, the obtained results being very similar.

In many real (non-simulated) applications, the response time of a technique or algorithm is of vital importance. In the case of COVID-19 the waiting time of the RT-PCR test to a patient was crucial because the subsequent implications that this result could have (as was explained in the introduction). For such reason, the need to refine the effectiveness of the predictions of these machine learning models, without discarding the importance and social impact of this methodology in an era where artificial intelligence covers more space.

In the last decade it has become evident that for the good performance of machine learning models, and mainly for deep learning, it is of vital importance to have real and large databases. Although there are numerous numerical methods and transformations that can be used to expand the database [4], which could be effective for work not related to human life, in real situations of medical images associated with a pathology or diagnosis, the most advisable is to have real and large databases.

Conclusions

In this study, we used a neural network model to perform deep learning in order to predict COVID-19 disease from chest CT images that included samples from COVID-19 patients and samples from healthy individuals (without COVID-19).

We quantitatively compared the obtained results using deep learning and SVM, and the evaluation metrics in the prediction of COVID-19 disease were very similar in both models. However, the training time for the SVM model was much shorter.

We do not intend to draw definitive conclusions from these results. Our objective was to know the performance of both models in terms of prediction accuracy according to existing databases of COVID and non-COVID patients. The obtained results showed that in presence of small databases, established machine learning methods cannot be completely discarded.

In the case of COVID-19 the waiting time of the RT-PCR test to a patient was crucial because the subsequent implications that this result could have. For such reason, the need to refine the effectiveness of the predictions of these machine learning models, without discarding the importance and social impact of this methodology in an era where artificial intelligence covers more space

References:

1. Brian Mondeja, Odalys Valdes, Sonia Resik, Ananayla Vizcaino, Emilio Acosta, Adelmo Montalván, Amira Paez, Mayra Mune, Roberto Rodríguez, Juan Valdés, Guelsys Gonzalez, Daisy Sanchez, Viviana Falcón, Yorexis González, Vivian Kourí, The IPK Virology Research Group, Angelina Díaz and María Guzmán: SARS-CoV-2: preliminary study of infectedhuman nasopharyngeal tissue by high resolution microscopy, Virology Journal, 18:149, 2021. https://doi.org/10.1186/s12985-021-01620-1
2. Fang, Z., Zhao, H., Ren, J., Maclellan, C., Xia, Y., Sun, M. and Ren, K.: SC2Net: A Novel Segmentation-based Classification Network for Detection of COVID-19 in Chest X-ray Images, IEEE journal of biomedical and health informatics, [online], 2022. https://doi.org/10.1109/JBHI.2022.3177854
3. Pedro Silvaa, Eduardo Luza, Guilherme Silvab, Gladston Moreiraa, Rodrigo Silvaa, Diego Lucioc and David Menottic: COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis, Informatics in Medicine Unlocked, Vol. 20, 100427, 2020. https://doi.org/10.1016/j.imu.2020.100427
4. Md. K. Islama, Sultana U. Habibaa, Tahsin A. Khana and Farzana Tasnimb: COV- RadNet: A Deep Convolutional Neural Network for Automatic Detection of COVID-19 from Chest X-Rays and CT Scans, Computer Methods and Programs in Biomedicine Update, Vol. 2, 100064, 2022. https://doi.org/10.1016/j.cmpbup.2022.100064
5. H. M. Shyni and E. Chitra: A comparative study of X-ray and CT images in COVID-19 detection using image processing and deep learning techniques, Computer Methods and Programs in Biomedicine Update, Vol. 2, 1000054, 2022. https://doi.org/10.1016/j.cmpbup.2022.100054

6. C. Long, H. Xu, Q. Shen, X. Zhang, B. Fan, C. Wang, H. Li, Diagnosis of the Coronavirus disease (COVID-19): rRT-PCR or CT?, European journal of radiology, Vol. 126, 108961, 2020.
7. Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng and Bo Xu: A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19), European Radiology, Vol. 31: 6096–6104, 2021. https://doi.org/10.1007/s00330-021-07715-1
8. Abeer Aljuaid and· Mohd Anwar: Survey of Supervised Learning for Medical Image Processing, SN Computer Science, Vol. 3:292, 2022. https://doi.org/10.1007/s42979-022-01166-1
9. Eyad Elyan, Pattaramon Vuttipittayamong, Pamela Johnston, Kyle Martin, Kyle McPherson, Carlos F. M. García, Chrisina Jayne and Md. M. K. Sarker: Computer vision and machine learning for medical image analysis: recent advances, challenges, and way forward, Artificial Intelligence Surgery, Vol. 2:2445, 2022. http://dx.doi.org/10.20517/ais.2021.15
10. Geert Litjens, Thijs Kooi, Babak E. Bejnordi, Arnaud A. A. Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken and Clara I. Sánchez: A Survey on Deep Learning in Medical Image Analysis, arXiv:1702.05747v1 [cs.CV], 2017.
11. Kenji Suzuki: Survey on Deep Learning Applications to Medical Image Analysis: Medical Imaging Technology, Vol. 35: 4, 2017.
12. Laura Brito and Roberto Rodríguez: Classification of some epidemics through microscopic images by using deep learning. Comparison, Imaging and Radiation Research, Vol. 6:1, 2023. http://dx.doi.org/10.24294/irr.v6i1.5451
13. A. Jeddi, M. J. Shafiee and A. Wong, A: Simple Fine-tuning Is All You Need: Towards Robust Deep Learning Via Adversarial Fine-tuning, arXiv preprint (2020) arXiv: 2012.13628, 2020.
14. R. Yamashita, M. Nishio, R. Do and K. Togashi: Convolutional neural networks: an overview and application in radiology, Insights Imaging, Vol. 9: 4, 2018. https://doi.org/10.1007/s13244-018-0639-9
15. Borja-Robalino R, Monleón-Getino A, Rodellar J. Standardization of performance metrics for classifiers (Spanish). Revista Ibérica de Sistemas e Tecnologias de Informação, E30: 184-196, 2020.
16. Fernando Berzal: Redes neuronales & Deep Learning, 2018. https://deep-learning.ikor.org
17. Xavier Glorot, and Yoshua Bengio: Understanding the difficulty of training deep feedforward neural networks, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR 9:249-256, 2010.
18. Jeff Heaton: Applications of Deep Neural Networks with Keras, arXiv:2009.05673v5 [cs.LG], 17 May 2022
19. Stephan Mandt, Matthew D. Hoffman and David M. Blei: Stochastic gradient descent as approximate Bayesian inference, Journal of Machine Learning Research, Vol. 18: 1-35, 2017. https://arxiv.org/abs/1704.04289
20. Diederik P. Kingma and Jimmy Ba: Adam: A Method for Stochastic Optimization, Third International Conference for Learning Representations, San Diego, 2015. https://arxiv.org/abs/1412.6980
21. Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever and Ruslan
22. R. Salakhutdinov: Improving neural networks by preventing co-adaptation of feature detectors, 2012. https://arxiv.org/abs/1207.0580
23. T. Mahmud, M. A. Rahman and S. A. Fattah: CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization, Computers in biology and medicine Vol.122:103869, 2020.