A Computer Aided Medical Classification System of COVID-19 CT Lung Scans using Convolution Neural Networks

Miri Weiss Cohen, Oriel Gilo Liron David

Braude College of Engineering,
, miri@braude.ac.il, orielgilo@gmail.com, lirondavid.work@gmail.com

Corresponding author: Miri Weiss Cohen, miri@braude.ac.il

Abstract. The CT scan is an important diagnostic procedure for COVID-19. CT images tend to reveal similar features in most COVID-19 cases, including ground-glass opacity in the early stages and pulmonary consolidation in the later stages. The work presented here demonstrates the feasibility of developing a classification system for Covid-19 CT lung scans, which would assist doctors in distinguishing COVID-19 from healthy cases. Three architectures, ResN50, Inception-V3, and Xception, were trained, validated, and tested to achieve the highest accuracy. Furthermore, the influence of hyper-parameters on the accuracy of each model was evaluated, searching for minimal loss values. With this system, classification of CT scans can be performed.

Keywords: Lung CT-Scans, COVID-19, Convolution Neural Network (CNN)
DOI: https://doi.org/10.14733/cadaps.2022.522-533

1 INTRODUCTION

COVID-19, an infection disease with a high risk of virulence and a rapid spread, has shocked the world by affecting billions of people both from a health and economic perspective. As of March 15, 2021, there have been 118,754,336 confirmed cases of COVID-19, including 2,634,370 deaths, according to the World Health Organization. [21]. Clinical symptoms, epidemiological history, and positive CT images, as well as pathogenic testing, are used to diagnose COVID-19. There are respiratory symptoms, fever, cough, dyspnea, and pneumonia associated with COVID-19. [13]. A chest CT scan and pathogenic testing were positive even in asymptomatic infected families, so these symptoms are nonspecific. Once a person is identified as a PUI (Person Under Investigation), lower respiratory specimens, such as bronchoalveolar lavage, tracheal aspirate, or sputum, will be collected for pathogenic testing. Tests such as reverse transcription polymerase chain reaction (RT-PCR) are the gold standard (the best available under reasonable conditions), but they require significant time and are associated with false-negative results. The nucleic acid detection rate is low (between 30-50%) and does not look promising for the disease in the early stages.[9]. It has been questioned whether RT-PCR

is sensitive enough to detect the disease at its earliest onset. Fang et al. [20] and [7] compared the sensitivity of non-contrast chest CT to RT-PCR in detecting COVID-19. Serial RT-PCR testing eventually confirmed the diagnosis of COVID-19 infection. According to the researchers, non-contrast chest CT has a sensitivity of 98 percent, and RT-PCR sensitivity is 71 percent. Diagnostic imaging such as X-rays and CTs are important

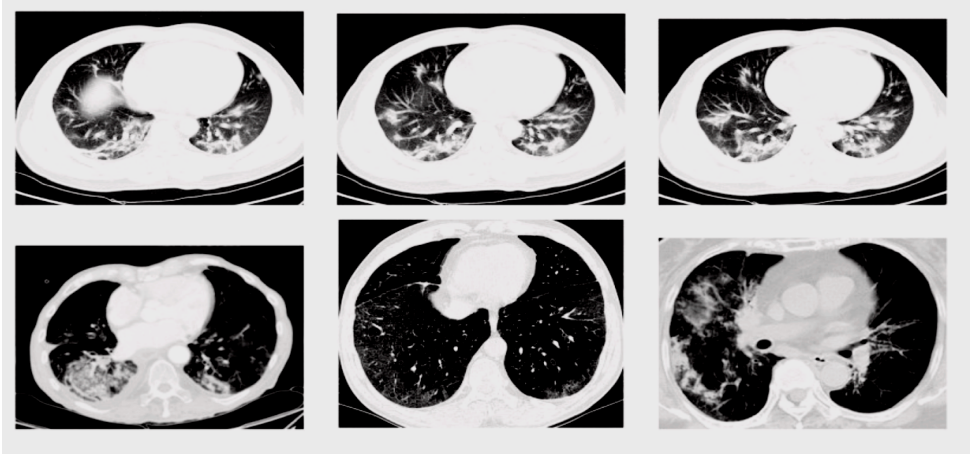


Figure 1: CT scans COVID-19 (top) and normal (bottom) [5].

for COVID-19. [11]. On CT images, patients with COVID-19 typically display ground-glass opacity early on and pulmonary consolidation later on [14]. Some patients have a rounded morphology and a peripheral lung distribution. CT images can help identify potential cases of viral pneumonia earlier, but they overlap with images of other inflammatory and infectious diseases of the lungs. There are instances when radiologists have difficulty distinguishing COVID-19 from viral pneumonia, leading to the complexity of an automated medical classification system [12]. In Wang et al. [20] A CNN method was used to compare two skilled radiologists compared to two trained radiologists. A radiologists' accuracy of 55.6 percent and sensitivity of 0.72 was achieved while using AI automated methods, the optimum threshold probability reached a sensitivity of 0.88 with an accuracy of 89.5 percent in the internal validation, and a sensitivity of 0.83 with a 79.3% in the external testing. Figure 1 depicts various CT scans of six cases, with the top showing COVID-19 and the bottom healthy lungs. The use of X-ray and CT imaging for diagnosing COVID-19 can be implemented in rural areas and emerging countries, where the sources of scans are more readily available. Our developed system will demonstrate the feasibility of establishing an early screening model to help doctors distinguish COVID-19 from healthy cases by using deep learning techniques. In an attempt to achieve the highest possible accuracy in an automated classification system, three Convolutional Neural Network architectures were trained, tested, and validated. Three hyperparameters were studied and tested in order to determine whether they affected the learning system's results. Learning rate, batch size and number of epochs were evaluated to determine the best values that resulted in the lowest loss rate.

2 PREVIOUS WORK

Simpson et al. [15] states that outpatient CT screening for COVID-19 disease 19 (COVID-19) pneumonia is currently not recommended by the majority of radiology societies. In their paper, they outlined at least three characteristics of COVID-19: a ground-glass appearance, striking peripheral distribution with the pleura, and usually more than one independent point of infection. The models were then optimized based on these findings. Xu et al. [22], This study examined whether AI can be used to early screen COVID-19 patients

using computed tomography (CT) images, and what the diagnostic accuracy was using machine learning. 618 CT samples were obtained in the study, 219 from 110 COVID-19 patients. The diversity of the data was ensured by collecting samples at least two days apart, and every patient had a RT-PCR test conducted. 224 patients with influenza-A viral pneumonia and 175 healthy people were also collected and tested. In the study, segmentation techniques were used in combination with a ResNet-18-based classification model and various relative distancings. The system was able to distinguish COVID-19 and healthy cases with an overall accuracy rate of 86.7%. Furthermore, Gosez et al.[9] studied two types of cases: first, a 3D Volume Analysis of nodules and focal opacity; and second, a 2D Slice Analysis of CT scan slices to detect and localize diffuse opacity (GGO). Both aspects of the research have been described among the most significant features for COVID-19. Among the advantages of using 2D scans, the authors highlight the commonality of 2D scans compared with 3D scans, as well as the ease of finding appropriate pre-trained networks as well as their success in resolving limited data scenarios. By extending the pre-processing stage of the data, a different approach was taken by Wang et al. [19]. A three-phased architecture was proposed: ROI extraction using a computer vision model, followed by a random selection of ROIs for training. Second, train the modified Inception model named M-Inception to extract features, followed by training the fully connected network with a classification model. According to the external testing dataset, accuracy was 79.33%, sensitivity was 0.83, and specificity was 0.67. Another interesting aspect was the comparison between the human as well as the computerized Radiologist classification system. An average of 55.8% accuracy as well as 0.71 sensitivity and 0.51 specificity was achieved by radiologists.

A total of 777 CT images of 88 patients diagnosed with COVID-19 were collected by Song et al. [16] from hospitals in two provinces in China. Additionally, 505 images were obtained from patients with bacteria pneumonia, and 708 images from healthy individuals. The authors in their work extracted the lungs' central regions and filled out the lung segmentation with the lung itself to prevent noises caused by different lung contours. Next, two convolutional neural networks, ResNet50 and Details Relation Extraction neural network (DRE-Net), were used to extract the top-K details from the CT images and obtain image-level predictions. The image-level predictions were aggregated to arrive at patient-level diagnoses. The accuracy is near 0.97 at the image level and about 0.99 for the patient level, but implementation was done on the Tianhe-2 supercomputer (a 33.86-petaflops computer located at the National Supercomputer Center in Guangzhou, China).

3 CONVOLUTIONAL NEURAL NETWORKS (CNN)

Deep learning implements supervised learning for classification, which is similar to the way the human brain processes data and creates patterns for decision-making [10, 23]. Convolutional neural networks (CNNs) are deep graph networks that are trained and tested by training with labeled input data. Here we use CT scans of lungs as the input to a CNN. The layers in the CNN are composed of several filters that extract local features. In the forward pass, each input image passes through multiple layers that combine convolution layers, activation layers, pooling, fully connected layers, and a softmax statistical function that classifies the object with probabilistic values. Back-propagation regulates weight changes in relation to a loss value to optimize the net [2]. Using the COVID-19 data-set [6], we trained three different CNN architectures. The three architectures are ResN50, Inception V3 and Xception. They are discussed in a more detail in the following section.

Residual networks -ResNet50 ResNet made a revolution in the CNN architectures by introducing residual learning in CNNs and devised an efficient methodology for training deep networks [8]. ResNet introduced residual learning into CNNs and devised an efficient methodology. ResNet has less failures on image classification and gains a 28 percent improvement in image recognition [17]. In image recognition and localization tasks, ResNet 50's representational depth is of central importance, and a residual block is shown to be able to train networks with extremely deep representational depth. The residual learning method is based on combining layers, thus enabling the training of a network with more than 900 layers using stochastic gradient descent optimization. This residual learning transfers the outputs from previous layers to the outcomes of stacked

layers. This is achieved by defining $F(x) = H(x) - x$ which can be re-framed into $H(x) = F(x) + x$, where $F(x)$ and x represent the stacked non-linear layers and the identity function (input=output) respectively and $H(x)$ is interference mapping. For each residual function F, there is a stack of 3 layers $Conv1x1$, $Conv3x3$, $Conv1x1$, where the $Conv1x1$, layer is responsible for decreasing and then increasing (restoring) dimensions, leaving the $Conv3x3$, layer a bottleneck with smaller input/output dimensions.Using this residual function offers a solution for creating a deep network that contains 50/101/152 layers.

Inception V3 For this model, several optimization techniques have been proposed to loosen the constraints for easier model adaptation [18]. Inception v3 networks are configured so that the output size equals the input size, and variations of the reduction method are used in the architecture. Inception blocks are gradually constructed to reduce grid sizes where applicable. A quick overview of the Inception structure is provided by the following steps:

Step 1: Convolutional factorization optimizes computational efficiency by reducing the number of parameters involved in a network. It also maintains an assessment of the network efficiency. Step 2: The use of smaller convolutions instead of larger convolutions accelerates training. Step 3: Asymmetric convolutions such as $Conv3x3$, could be replaced by a $Conv1x3$, followed by a $Conv3x1$, . If a $Conv3x3$, is replaced by a $Conv2x2$, , t would consist of slightly more parameters compared to the asymmetric convolution proposed. Step 4: The auxiliary classifier is a small CNN inserted between layers during training, and the loss is added to the main network loss. In GoogleNet, auxiliary classifiers were used for a deeper network, whereas an auxiliary classifier in Inception-v3 functions as a regularizer. Step 5: Grid size reduction is done by pooling operations.

Xception architecture is based entirely on depth-wise separable convolution layers and stands for "Extreme Inception". A depth-wise separable convolution is also referred to as a "separable convolution", since it works on the same principles as the standard convolution. This is a modified inception block that has been made wider and has been replaced with the original spatial dimensions. ($Conv1x1$, , $Conv5x5$, and $Conv3x3$,) with a single dimension $Conv3x3$, followed by a $Conv1x1$, to regulate computational complexity [4]. In Xception, spatial correlations in the feature maps of convolutional neural networks are decoupled from cross-channel correlations to make the network computationally efficient [10].The output is spatially transformed n times by raising the dimensional embedding of the output by $Conv1x1$, where n represents a width defining the cardinality. In conventional CNN architectures, convolutional operation uses only one transformation segment, inception block uses three transformation segments, In Xception, the number of transformation segments is equal to the number of feature maps. It is true that Xception's transformation strategy does not reduce the number of parameters, but it makes learning more efficient and results in improved performance.

Due to the fact that the pandemic is a new disease, and there is a lack of quality data-sets, we use transfer learning to extract knowledge from one or more applications within this field of study and increase the learning capabilities of our system. This method is faster and more reliable than training a network from scratch with randomly initialized weights.

4 HYPER-PARAMETER OPTIMIZATION

In our developed system, we used the COVID-19-COV-2 CT-Scan dataset [6]. It consists of 2482 CT scans, of which 1252 correspond to patients with COVID-19-COV-2, and 1230 correspond to patients not diagnosed. For the learning process, we divided the images into three different sets: the training set, the validation set, and the test set. Our results indicated that 71.4 percent of the training data, 14.3 percent of the validation data, and 14.3 percent of the test data yielded the best results. The training phase focused on the influence of hyperparameters on the accuracy of each model, searching for minimal loss values. Learning rate and Epoch size were evaluated, while batch size was fixed at 32. The learning rate ranges tested were 0.0005, 0.00005, and 0.000005, while the epoch sizes were 20, 40, and 60. As batch size is hardware-dependent, we chose an adaptive batch size local GPU use [1].

While training deep networks, it is useful to reduce the learning rate as the training progresses. This can

be accomplished through predefined learning rate schedules or adaptive learning rate methods. We used the schedule learning rate for this system development [3]. When the optimization model performance reaches a plateau, the learning rate is reduced by a factor every few epochs. This callback is designed to reduce the learning rate after the model stops improving and allows for the fine-tuning of weights.
Since Covid-19 is a new disease and there were no quality data sets, the transfer learning (TL) method was employed. This method is much faster and more reliable than training a network from scratch with randomly initialized weights. These tested parameters are defined to simplify the presentation of the results:

- tp = model prediction was COVID-19, and the CT scan was classified as COVID-19
- tn = prediction was NORMAL lungs, and the CT scan was classified as NORMAL lungs
- fp = prediction was COVID-19, and the CT scan was classified as NORMAL lungs
- fn = prediction was NORMAL lungs, and the CT scan was classified as COVID-19
- Accuracy rate prediction $Accuracy = (t_p + t_n)/(t_p + t_n + f_p + f_p)$
- Rate of correctly classifying a CT scan as a COVID-19 $Prescision = (t_p)/(t_p + f_p)$
- Rate of correctly classifying a CT scan as a COVID-19 $Recall = (t_p)/(t_p + f_n)$
- Rate correctly classifying the CT scan of a Normal lungs $Sensitivity = (t_n)/(t_n + f_p)$

The most important aspect of COVID-19 detection is alerting those infected rather than informing healthy people if they are negative for the virus. Hence, we will focus on the Recall measure which represents the number of verified patients out of all those being infected with the virus.

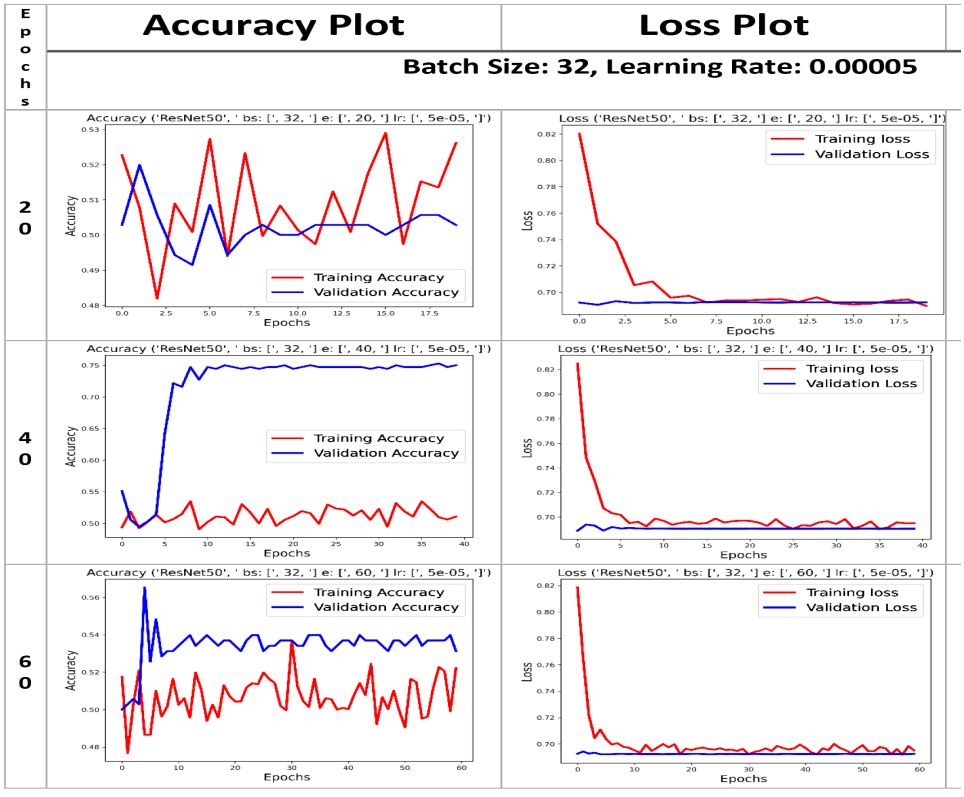


Figure 2: ResNet-50, training and validation accuracy and loss LR=5e-5 Epochs=20,40,60 and BS= 32.

4.1 ResNet-50

Figure 2 depicts one case of ResNet50 architecture, analysing 0.000005 learning rate for a batch size of 32. All the experiments were performed at a static learning rate and batch size, while epoch size was varied continuously during each case of the experiment. In terms of loss, changing the number of Epochs is shown and its effect on system performance. The 20-epochs graph is very volatile in terms of having big change spikes in the graph, meaning unstable. The Accuracy metric indicates that the results for 20 epochs seem acceptable, but they are not stable compared to other epoch sizes. A 5e-4 learning rate may cause some changes in the graph. For the 20 epochs the results:
Accuracy: 53.1% Precision: 58.6% Recall/Sensitivity: 22.9% Specificity 83.6% Train accuracy: 52.78
For the 40 epochs the results:
Accuracy: 55.4% Precision: 53.3% Recall/Sensitivity: 95.9% Specificity 15.2% Train accuracy: 51.06
For the 60 epochs the results:
Accuracy: 55.9% Precision: 53.4% Recall/Sensitivity: 94.1% Specificity 16.9% Train accuracy: 52.21
Overall, we can see that the 40-epochs graph gave us the best results with 93.85% Recall.

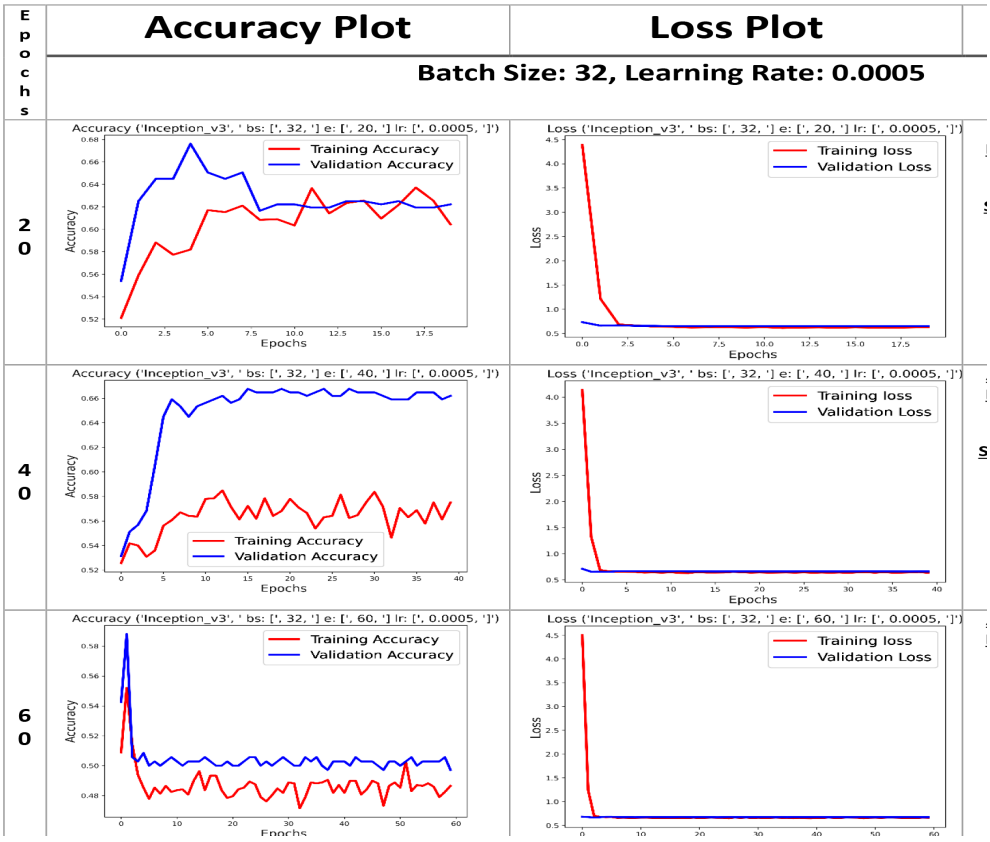


Figure 3: Inception-V3: Training and validation accuracy and loss graphs LR=5e-4, Epochs=20,40,60, BS=32.

4.2 Inception-V3

Figure 3 depicts a case of Inception-V3 architecture, examining the effects of epochs on a 5e-4 learning rate (batch-size = 32). The following results were obtained for 20 epochs:
Accuracy: 64.1% Precision: 71.4% Recall/Sensitivity: 50.4% Specificity 79.7% Train accuracy: 60.4
For the 40 epochs the results:
Accuracy: 58.98% Precision: 55.5% Recall/Sensitivity: 93.8% Specificity 23.4% Train accuracy: 57.5
For the 60 epochs the results:
Accuracy: 50.3% Precision: 50.4% Recall/Sensitivity: 100% Specificity 0% Train accuracy: 48.2
In terms of loss, changing the number of Epochs did not have any effect on the results. Looking at accuracy, we can see that 20 Epochs provided good results, but did not stabilize due to the small number of Epochs. As a result of the relatively high learning rate (5e-4), we can detect volatility in the optimization process that influences the learning process and results. In the graph of the 60-epoch training process we can see that it evolved to a local minima which gives results that are similar to that of a coin flip. Our overall best result was obtained with the 40-epochs diagram, which provided a 93.85% Recall.

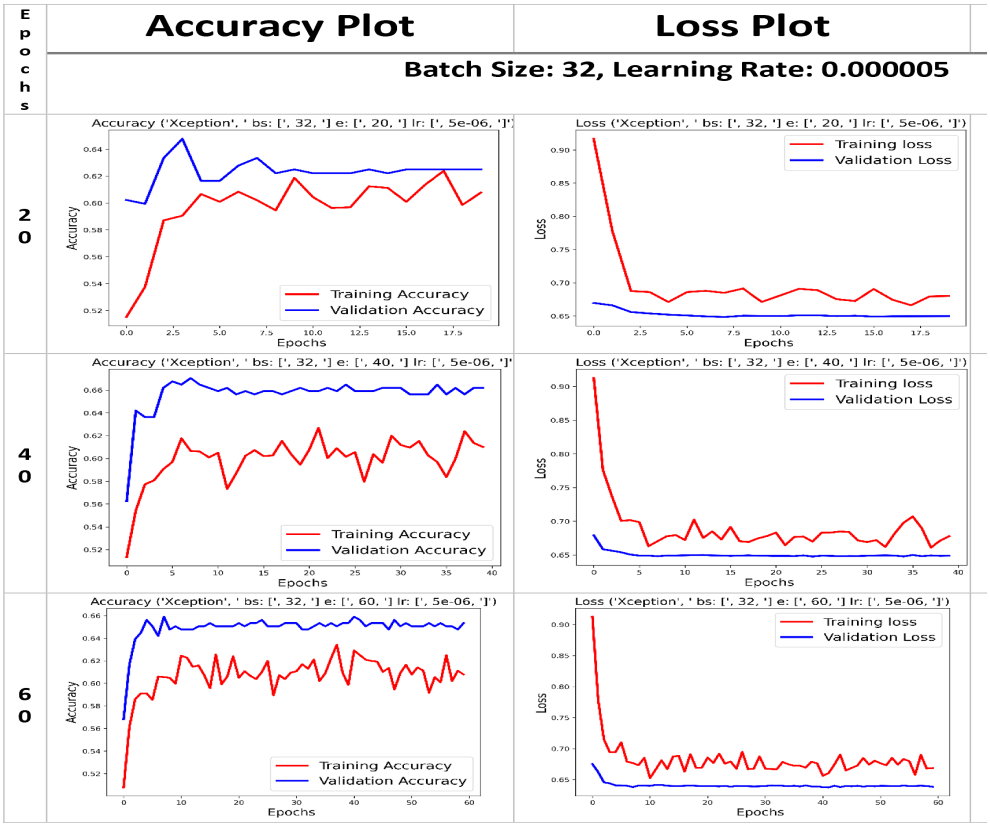


Figure 4: Xception: Training and validation accuracy and loss graphs, LR=5e-5 Epochs=20,40,60 and BS=32.

In Figure 5 all simulations performed in the hyperpapameretes optimization procedure are depicted. The graphs in each case represent the optimization in terms of accuracy, the red graph represents the training process, and the blue graph represents the validation.

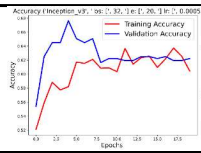
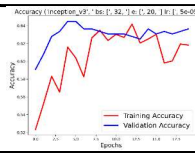
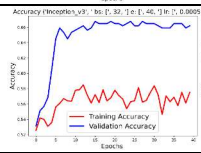
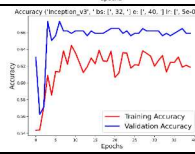
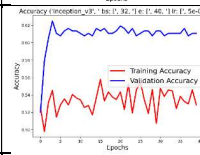
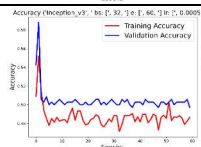
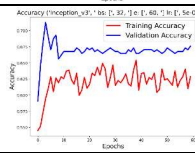
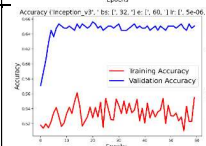
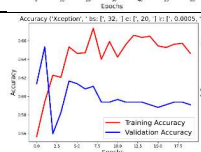
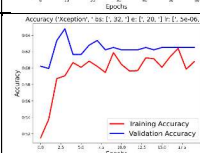
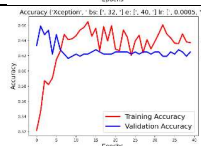
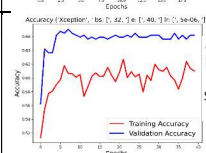
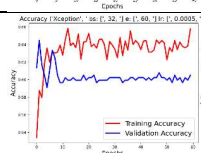
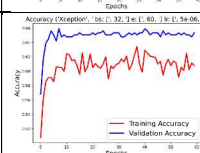
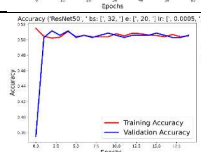
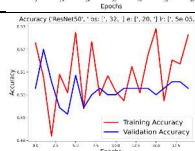
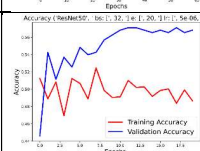
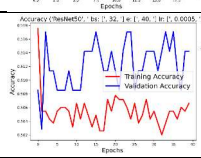
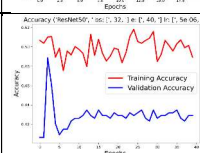
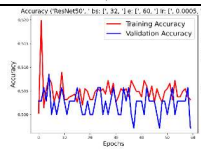
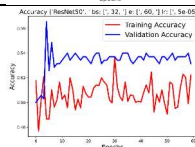
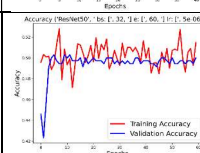
Learning rate / Architecture	0.0005	0.00005	0.000005
Inception_v3 20 Epoches	 Accuracy: 64.8876% Precision: 71.4285% Recall: 50.2793% Specificity 79.6610% F1-score: 59.0164	 Accuracy: 62.3595% Precision: 70.2702% Recall: 43.5754% Specificity 81.3559% F1-score: 53.7931	 Accuracy: 62.0786% Precision: 64.4736% Recall: 54.7486% Specificity 69.4915% F1-score: 59.2145
Inception_v3 40 Epoches	 Accuracy: 58.9887% Precision: 55.4455% Recall: 93.8547% Specificity 23.7288% F1-score: 69.7095	 Accuracy: 64.8876% Precision: 71.4285% Recall: 50.2793% Specificity 79.661% F1-score: 59.0164	 Accuracy: 58.7078% Precision: 58.1632% Recall: 63.6871% Specificity 53.6723% F1-score: 60.8
Inception_v3 60 Epoches	 Accuracy: 50.2809% Precision: 50.2809% Recall: 100.0% Specificity 0.0% F1-score: 66.9159	 Accuracy: 68.5393% Precision: 74.1% Recall: 57.5419% Specificity 79.661% F1-score: 64.7798	 Accuracy: 64.0449% Precision: 64.5714% Recall: 63.1285% Specificity 64.9717% F1-score: 63.8418
Xception 20 Epoches	 Accuracy: 64.3258% Precision: 62.745% Recall: 71.5083% Specificity 57.0621% F1-score: 66.8407	 Accuracy: 66.573% Precision: 66.8539% Recall: 66.48044% Specificity 66.6666% F1-score: 66.6666	 Accuracy: 62.3595% Precision: 68.2926% Recall: 46.9273% Specificity 77.9661% F1-score: 55.6291
Xception 40 Epoches	 Accuracy: 65.7303% Precision: 71.7557% Recall: 52.5139% Specificity 79.096% F1-score: 60.6451	 Accuracy: 64.6067% Precision: 76.2376% Recall: 43.0167% Specificity 86.4406% F1-score: 55.0	 Accuracy: 68.5393% Precision: 72.4832% Recall: 60.3352% Specificity 76.8361% F1-score: 65.8536
Xception 60 Epoches	 Accuracy: 61.7977% Precision: 70.8737% Recall: 40.7821% Specificity 83.0508% F1-score: 51.773	 Accuracy: 67.1348% Precision: 65.6565% Recall: 72.6257% Specificity 61.5819% F1-score: 68.9655	 Accuracy: 66.8539% Precision: 70.7483% Recall: 58.1005% Specificity 75.7062% F1-score: 63.8036
ResNet50 20 Epoches	 Accuracy: 50.2809% Precision: 50.2809% Recall: 100.0% Specificity 0.0% F1-score: 66.9158	 Accuracy: 53.09% Precision: 58.5714% Recall: 22.905% Specificity 83.6158% F1-score: 32.9317	 Accuracy: 56.7415% Precision: 59.2592% Recall: 44.6927% Specificity 68.9265% F1-score: 50.9554
ResNet50 40 Epoches	 Accuracy: 50.8427% Precision: 50.578% Recall: 97.7653% Specificity 3.3898% F1-score: 66.6666	 Accuracy: 55.618% Precision: 53.271% Recall: 95.5307% Specificity 15.2542% F1-score: 68.4	 Accuracy: 42.4157% Precision: 45.1852% Recall: 68.1564% Specificity 16.3842% F1-score: 54.343
ResNet50 60 Epoches	 Accuracy: 52.8089% Precision: 54.7826% Recall: 35.1955% Specificity 70.6214% F1-score: 42.8571	 Accuracy: 55.8989% Precision: 53.481% Recall: 94.4134% Specificity 16.9491% F1-score: 68.2828	 Accuracy: 46.3483% Precision: 44.2307% Recall: 25.6983% Specificity 67.2316% F1-score: 32.5088

Figure 5: A summary of all accuracy results from all cases.

4.3 Xception

Figure 4 depicts a case of Xeception architecture, examining the influence of epochs on a 0.000005 learning rate (batch-size = 32).
For the 20 epochs the results:
Accuracy:62.3% Precision:68.29% Recall/Sensitivity:46.97% Specificity 77.9% Train accuracy: 60.478
For the 40 epochs the results:
Accuracy: 68.6% Precision: 72.5% Recall/Sensitivity: 60.35% Specificity 76.87% Train accuracy: 61.1
For the 60 epochs the results:
Accuracy: 66.8% Precision: 70.74% Recall/Sensitivity: 58.1% Specificity 75.6% Train accuracy: 60.78
Although the Validation Accuracy improved by a small fraction by decreasing the learning rate from 5e-5 to 5e-6, the Training Accuracy declined by a fraction. Overall performance in all measurements decreased, except in 40 epochs. Figure 6 illustrates the combined results of the three best architectures: Xception, Inception-V3 and ResNet50. The Xception is marked with red for the training and blue for the validation; the hyper-parameters chosen for this architecture were 5e-06 learning rate, 32 batch sizes, and 40 epochs.
Inception-V3 is marked with orange for training and cyan for validation. The hyper-parameters selected for this architecture were 5e-05 learning rate, batch size of 32 and 60 epochs. ResNet50 is marked in yellow for

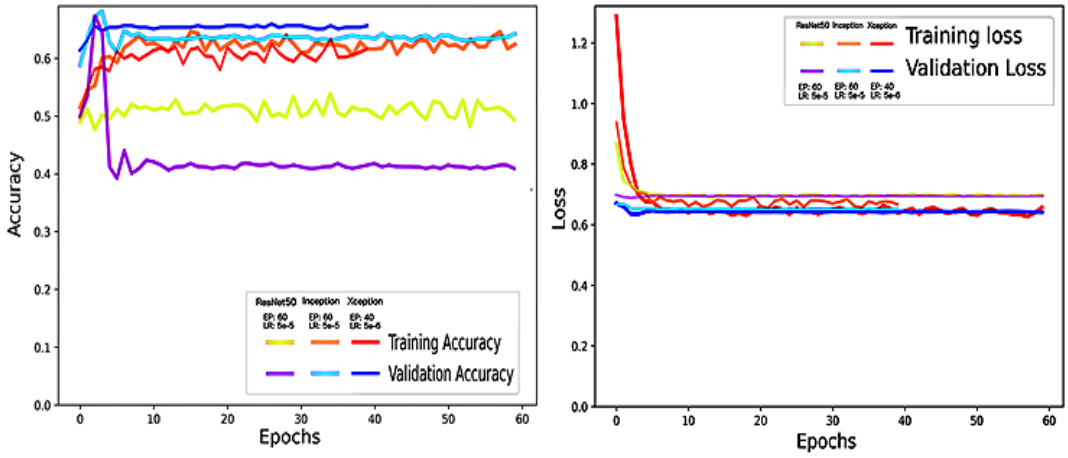


Figure 6: Combined training and validation accuracy and loss graphs for the 3 architectures: Xception, Inception-V3 and ResNet50.

the training and purple for the validation. The chosen parameters for this architecture were Learning rate of 5e-05, batch size of 32 and 60 epochs. In spite of the fact that the Xception graph was shorter than the other 2, it still had the best Validation Accuracy rate and the lowest Validation Loss rate.

5 SYSTEM IMPLEMENTATION

This system is very user friendly, allowing anyone (even less qualified) to use it. The user uploads CT scans into the system by dragging and dropping a CT scan image into the designated area. Once the CT scan image is uploaded, the model makes a prediction and presents the result in the Main window as seen in Figure 7. This results page contains the assessment with colored rectangles indicating the findings: red for detection of COVID-19, and green for normal lung detection. The second line contains the sensitivity of the result, which represents the probability that a diagnostic test will identify patients who have the specific disease. A value

closer to 1 indicates illness, while a value closer to 0 indicates normality. The result can be saved to a file for further study.

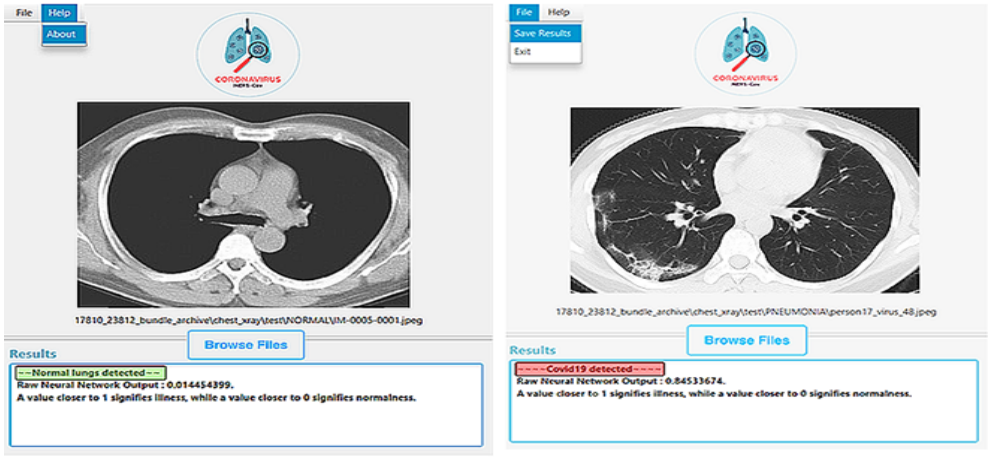


Figure 7: Output format for the classification task, normal CT lung scan (left) and COVID-19 (right).

6 CONCLUSIONS

Using CNN, we demonstrated the feasibility of developing a computer-aided classification system for COVID-19 CT lung scans, which would assist physicians in identifying COVID-19 from healthy cases. A series of CNN architectures with different learning rates, epochs, and batch sizes were tested on the same data-set in order to understand the impact of each hyperparameter. Based on these findings, we can conclude that learning rate volatility did not affect accuracy by the same epoch size. Thus, the Xception architecture proved to be the optimal one. For example: epoch size = 40, learning rate = 0.000005, batch size = 32.

ORCID

Miri Weiss Cohen, http://orcid.org/0000-0001-5250-1016

REFERENCES

[1] Borgert, J.; Schmidt, J.D.; Schmale, I.; Bontus, C.; Gleich, B.; David, B.; Weizenecker, J.; Jockram, J.; Lauruschkat, C.; Mende, O.; et al.: Perspectives on clinical magnetic particle imaging. Biomedical Engineering/Biomedizinische Technik, 58(6), 551–556, 2013. http://doi.org//10.1515/bmt-2012-0064.

[2] Cheng, J.; Wang, P.s.; Li, G.; Hu, Q.h.; Lu, H.q.: Recent advances in efficient computation of deep convolutional neural networks. Frontiers of Information Technology & Electronic Engineering, 19(1), 64–77, 2018. http://doi.org//10.1631/FITEE.1700789.

[3] Chin, W.S.; Zhuang, Y.; Juan, Y.C.; Lin, C.J.: A learning-rate schedule for stochastic gradient methods to matrix factorization. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, 442–455. Springer, 2015. http://doi.org//10.1007/978-3-319-18038-0_35.

[4] Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, 1251–1258, 2017. http://doi.org//10.1007/978-3-319-18038-0_35.

[5] COVID19-DB: http://medicalsegmentation.com/covid19/, 2020.

[6] Dataset, S.C..C.S.: https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset, 2020.

[7] Fang, Y.; Zhang, H.; Xie, J.; Lin, M.; Ying, L.; Pang, P.; Ji, W.: Sensitivity of chest ct for covid-19: comparison to rt-pcr. Radiology, 296(2), E115–E117, 2020. http://doi.org//10.1148/radiol.2020200432.

[8] Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.A.; Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231, 2018.

[9] Gozes, O.; Frid-Adar, M.; Greenspan, H.; Browning, P.D.; Zhang, H.; Ji, W.; Bernheim, A.; Siegel, E.: Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning ct image analysis. arXiv preprint arXiv:2003.05037, 2020.

[10] Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S.: A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review, 53(8), 5455–5516, 2020. http://doi.org//10.1007/s10462-020-09825-6.

[11] Long, C.; Xu, H.; Shen, Q.; Zhang, X.; Fan, B.; Wang, C.; Zeng, B.; Li, Z.; Li, X.; Li, H.: Diagnosis of the coronavirus disease (covid-19): rrt-pcr or ct? European journal of radiology, 126, 108961, 2020. http://doi.org//10.1016/j.ejrad.2020.108961.

[12] Luz, E.; Silva, P.L.; Silva, R.; Moreira, G.: Towards an efficient deep learning model for covid-19 patterns detection in x-ray images. arXiv preprint arXiv:2004.05717, 2020.

[13] Menni, C.; Valdes, A.M.; Freidin, M.B.; Sudre, C.H.; Nguyen, L.H.; Drew, D.A.; Ganesh, S.; Varsavsky, T.; Cardoso, M.J.; Moustafa, J.S.E.S.; et al.: Real-time tracking of self-reported symptoms to predict potential covid-19. Nature medicine, 26(7), 1037–1040, 2020. http://doi.org//10.1038/s41591-020-0916-2.

[14] Poggiali, E.; Dacrema, A.; Bastoni, D.; Tinelli, V.; Demichele, E.; Mateo Ramos, P.; Marcianò, T.; Silva, M.; Vercelli: Can lung us help critical care clinicians in the early diagnosis of novel coronavirus (covid-19) pneumonia. Radiology, 295(3), 2020. http://doi.org//10.1148/radiol.2020200847.

[15] Simpson, S.; Kay, F.U.; Abbara, S.; Bhalla, S.; Chung, J.H.; Chung, M.; Henry, T.S.; Kanne, J.P.; Kligerman, S.; Ko, J.P.; et al.: Radiological society of north america expert consensus document on reporting chest ct findings related to covid-19: endorsed by the society of thoracic radiology, the american college of radiology, and rsna. Radiology: Cardiothoracic Imaging, 2(2), e200152, 2020. http://doi.org//10.1148/ryct.2020200152.

[16] Song, Y.; Zheng, S.; Li, L.; Zhang, X.; Zhang, X.; Huang, Z.; Chen, J.; Zhao, H.; Jie, Y.; Wang, R.; et al.: Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images. MedRxiv, 2020.

[17] Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, 2017.

[18] 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. http://doi.org//10.1109/CVPR.2016.308.

[19] 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. Scientific Reports, 10(1), 1–12, 2020. http://doi.org//10.1038/s41598-020-76550-z.

[20] Wang, S.; Kang, B.; Ma, J.; Zeng, X.; Xiao, M.; Guo, J.; Cai, M.; Yang, J.; Li, Y.; Meng, X.; et al.: A

deep learning algorithm using ct images to screen for corona virus disease (covid-19). European radiology, 1–9, 2021. http://doi.org//10.1038/s41598-020-76550-z.

[21] WHO: https://www.who.int/emergencies/diseases/novel-coronavirus-2019, 2020. Accessed: 2020-03-15.

[22] Xu, X.; Jiang, X.; Ma, C.; Du, P.; Li, X.; Lv, S.; Yu, L.; Ni, Q.; Chen, Y.; Su, J.; et al.: A deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering, 6(10), 1122–1129, 2020. http://doi.org//10.1016/j.eng.2020.04.010.

[23] Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K.: Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9(4), 611–629, 2018. http://doi.org//10.1007/s13244-018-0639-9.