Hellenic Mediterranean University

COVID-19 diagnosis using Convolutional Neural Networks

Computational Intelligence Project 2
Project's report

Manos Koutoulakis & Manos Markodimitrakis

Professor: Mr Papadourakis George

Abstract

Background and Objective: Nowadays, humans have been crucially affected by the outbreak of a virus, called COVID-19. The COVID-19 is an infectious disease, started from Wuhan, China in 2019. However, the COVID-19 virus has not yet disappeared. By virtue of COVID-19's spread in 212 countries and territories and accretive numbers of infected cases and death mounting to 76,409,274 and 1,688,256 (as of December 19, 2020), remaining a hazardous threat to the public health system. During the spread of the virus, many tools have been devised to diagnose and monitor the spread of COVID-19. In both medicine and machine learning, significant improvements have been made to produce more accurate diagnostic results. This project renders a response to combat the virus through Neural Networks. Deep Learning (DL) methods have been implemented to reach the most accurate diagnosis, using Convolutional Neural Networks (CNN). The current situation encouraged us to develop a DL model that can be found helpful for radiologists and clinicians in detecting COVID-19 through chest X-rays.

Methods: In this project, we propose a CNN model to automagically detect if the patient has been infected by COVID-19 from chest X-ray images. The proposed implementation is based on a Sequential model and trained on a dataset by collecting normal and COVID-19 chest X-ray images from two different publicly available datasets.

Results: The proposed implementation has been trained and tested on the preprocessed dataset. As the experimental results have shown, our proposed model has reached 96% of the overall accuracy, using 2-class cases (COVID-19, Normal). However, the results of this proposed model are not final. Nevertheless, this study of COVID-19 Diagnosis looks promising for further improvements and additional training data with other critical lung diseases, such as Pneumonia bacterial, Pneumonia viral.

Conclusion: Overall, our proposed model has shown promising results on a small range of cases (COVID-19 and Normal), we could achieve more precise results using additional cases. There can be no doubt that the proposed model could be a useful tool for clinical practitioners and radiologists so as to aid them in the diagnosis of Covid-19 cases.

1 Introduction

In 2019, a novel coronavirus, first infected people in Wuhan, China. The novel coronavirus, alternatively called COVID-19 caused by another virus SARS-CoV-2. In January 2020, the World Health Organization declared an outbreak and in March 2020 declared a pandemic. As of December 2020, more than 76.3 million cases have been confirmed, and more that of 1.7 million deaths have been reported worldwide [1].

The symptoms of COVID-19 range from an asymptomatic patient to a serious illness. Infected patients may usually experience difficulty breathing, speech, sudden fatigue, coughing, loss of taste or smell. Patients remain infectious for up to two weeks, and they can spread the virus even if they do not have symptoms (asymptomatic patient). The virus spreads through the air when people are near each other or via contaminated surfaces. For these reasons, the authorities have proposed precautionary measures to reduce the rapid spread of the virus. More specifically, these measures include social distancing, wearing a face mask in public, ventilation and air-filtering, self-isolation for exposed or asymptomatic people.

Throughout the virus spread, rapid screening of infected patients is a significant improvement in reducing the spread of the virus and so as to the infected patient can be isolated and treated. Currently, the main diagnosis method used for detecting the COVID-19 is real-time reverse transcription polymerase chain reaction (rRT-PCR) [2]. When the test is done, the result can be available within a few hours to 2 days. Additionally, research has been done in alternative methods such as PCR screening method which is based on chest radiography images. Furthermore, a chest radiology image detection system has many advantages over the common method of diagnosis. One of the most significant advantages is the rapid diagnosis of multiple cases with no potential of limited number of testing kits or resources. Additionally, such diagnosis systems are available in every medical research laboratory or hospital, hence, the CNN based diagnosis approach is an appropriate tool and easily available for implementation.

There are many similar approaches because of the tremendous number of patients' data that the research communities have. Many of the approaches have shown optimistic results not only for COVID-19 but also for other diseases (Pneumonia Bacterial, Pneumonia Virus) and they have already been implemented in many hospitals [3]–[5]

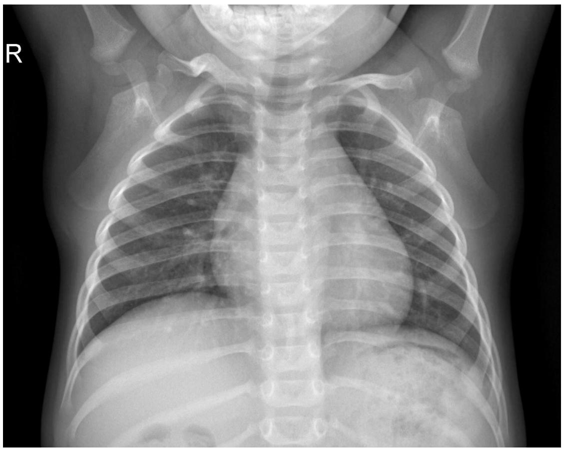
Generally, AI has predominant techniques for automated diagnosis in the medical field, considering the rapid growth of COVID-19 cases, the automatic detection through AI techniques is necessary. Some proposed approaches have shown agreeable results. For example, the deep learning model from Wang and Wong [6] reach 83.5% of accuracy using three classes from a dataset with 13,975 Chest X-ray (CXR) images across 13,870 patient cases. Additionally, Farooq and Hafeez [7] CNN approach achieved 96.23% accuracy using three classes.

In this project, we present a deep learning-based approach to detect COVID-19 infections from CXR images. We propose a deep CNN model to classify a COVID-19. To put it in another way, we built a model that can identify COVID-19 infections from CXR images. The proposed model can be useful to doctors in the triage.

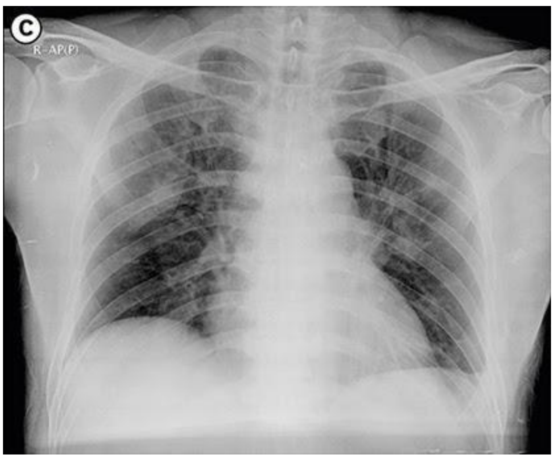
2 Dataset

AI and more specifically Deep Learning is all about data which are provided as incentive in these learning models. Taking into consideration that COVID-19 has radically increased through the data collection age, there are many open-source datasets and databases that are useful for scientific studies. Thus, we collect our dataset from two different open-source repositories. The first public repository is the COVID-19 CXR images are available by Joseph *et al.* in GitHub [8]. These images have been collected from various credible sources such as Radiological Society of North America (RSNA), and Radiopedia which includes COVID-19 cases. The second public repository is the Chest X-ray Image (Pneumonia) from Kaggle collected by Paul Mooney [9]. This dataset consists of 5,863 chest X-ray images both of normal and pneumonia cases. When we collect all the data, we choose randomly from the sources 270 of COVID-19 cases and 120 normal cases. Then, we resize all images to the dimension of 224 × 224 pixels. Table 1 below depicts the summary of the prepared dataset. Additionally, Figure 1. Illustrate samples from Normal CXRs and COVID-19 CXRs that have been collected to train the diagnosis model.

Cases	**No. Of Images**
COVID-19	270
Normal	120



(a) Normal Case



(b) COVID-19 case

3 Tools

3.1 TensorFlow

TensorFlow is an open-source framework for machine learning created by Google. It complements a flexible ecosystem of mathematical and machine learning tools. Deep Learning is a category of machine learning models, making the TensorFlow dominant for our project's purposes.

3.2 Keras

Keras is an Application Programming Interface (API), and it complements the TensorFlow framework. By using Keras, the whole process of data preprocessing and model training is minimized and clear. Keras is commonly used by CERN, NASA, NIH and many more scientific institutes and universities, providing low-level flexibility to implement arbitrary research ideas.

3.3 ScikitLearn.skimage

In general, ScikitLearn is another open-source machine learning library for Python. ScikitLearn is designed to intercommunicate with many numerical Python libraries such as NumPy and SciPy. We have already implemented the TensorFlow library for these purposes, but ScikitLearn has another useful library named Skimage for image manipulation. More specifically, Skimage is a submodule of ScikitLearn for image processing and computer vision. The main purpose of this library was image rescale in order to reduce the number of the parameters in our neural network model.

3.4 NumPy

NumPy is an essential numerical library in order to produce machine learning models. This module is combined in TensorFlow, automating many steps of preprocessing. Furthermore, machine learning models needs mathematical processes to produce results and make prediction

3.5 Matplotlib and Seaborn

Both MatplotLib and Seaborn are open-source Python libraries for data visualization. It is useful to create static, animated, and interactive visualizations in Python. These libraries are mainly used in visualization of results.

4 Methodology

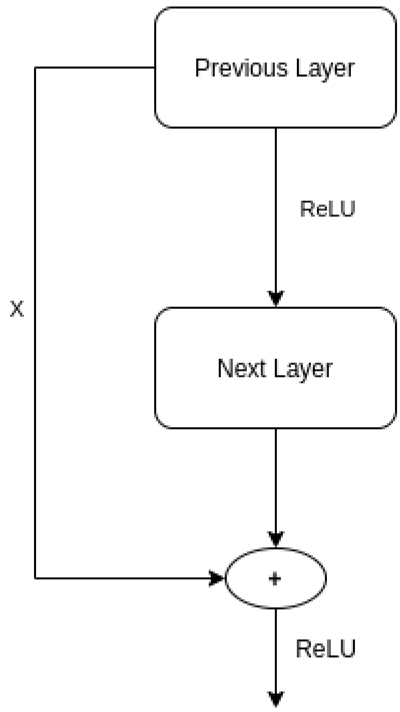
In this section we make an extended explanation of the project methodology for the proposed technique, model architecture and the implementation and training.

4.1 Introduction to Convolutional Neural networks

Convolutional Neural Network (CNN) is a Deep Learning technique comprising multiple convolutions and max-pooling layers stacked together. The stacked layers have local connections for better performance and efficiency. This technique facilitates the process to learn many complex features which a simple neural network cannot learn. In general CNN is a significant technique mainly in computer vision that has many applications in self-driving cars, robotics, medical diagnosis and treatment.

4.1.1 Convolutional Layers

In the convolutional layers the activations from the previous layers are convolved with a set of small filters, usually of size 3 x 3, collected in a tensor W(i,j), where "j" is the filter number and "i" is the layer number. Each filter shares the same weights across the whole input domain; we achieve reduction of the weights that need to be learned. Applying all the convolutional filters at all locations of the input to a convolutional layer produces a tensor of feature maps.



4.1.2 Activation Layer

The output of each convolutional layer is equipped with an activation function to introduce the non-linearity. There are a plethora of activation functions but for deep learning is the Rectified Linear Unit (ReLU) which simply computes the activation through thresholding the input ReLU(z)=max(0,z). To put it differently, if the input is less than zero then ReLU outputs 0, otherwise, the output is equal to the input. Feeding the features maps through an activation function produces new tensors, typically also called feature maps.

4.1.3 Pooling Layers

In CCN, the sequence of convolution layers is followed by a pooling layer in order to reduce the size of an input. Hence, this layer reduces the number of parameters in the network. The most common pooling techniques are Max Pooling which simply outputs the maximum value in the input region and Average Pooling which outputs the average value of the input. Pooling operations take small grid regions as input and produce single numbers for each region.

Other common elements in many modern CNNs include:

- Dropout regularization

A simple idea that gave a huge boost in the performance of CNNs. By averaging several models in an ensemble, one tends to get better performance than when using single models. Dropout is an averaging technique based on a stochastic sampling of neural networks. By randomly removing neurons during training, one ends up using slightly different networks for each batch of training data, and the weights of the trained network are tuned based on optimization of multiple variations of the network.

- Batch normalization

These layers are typically placed after activation layers, producing normalized activation maps by subtracting the mean and dividing by the standard deviation for each training batch. Including batch normalization layers forces the network to periodically change its activations to zero mean and unit standard deviation as the training batch hits these layers, which works as a regularizer for the network, speeds up training, and makes it less dependent on careful parameter initialization.

4.2 Model Building

We have implemented a 12 layers Convolutional Neural Network model consisting of multiple convolutions, activations, pooling, dropout and batch normalization layers. In the following Table 1 we specify each layer with the according output shape and the parameters. For the convolutional input layer, we use images of shape (width, height) 222 x 222 and batch size 32. The amount of output parameters of the first layer is 896. Following another convolutional layer of size 220 x 220 with batch size 64 and 18469 output parameters. Then we use a pooling and a dropout layer in order to reduce overfitting. We reuse this architecture with different parameters two times to enhance our model. Finally, we use a Flatten and a Dense layer followed by a Dropout and a Dense layer in order to result in a fully connected neural network.

Layer (type)	**Output Shape**	**Params #**
Conv2D	222 × 222 × 32	896
Conv2D_1	220 × 220 × 64	18469
MaxPooling2D	110 × 110 × 64	0
Dropout	110 × 110 × 64	0
Conv2D_2	108 × 108 × 64	36928
MaxPooling2D_1	54 × 54 × 64	0
Dropout_1	54 × 54 × 64	0
Conv2D_3	52 × 52 × 128	73856
MaxPooling2D_2	26 × 26 × 128	0
Dropout_2	26 × 26 × 128	0
Flatten	86528	0
Dense	64	5537856
Dropout_3	64	0
Dense_1	1	65
Total Parameters: 5,668,097		
Trainable Parameters: 5,668,097		
Non-Trainable Parameters: 0		

Table 1 *Details of COVID-19 Diagnosis architecture*

5 Results

5.1 Training process

The model was trained on the training dataset for 10 epochs using the early stopping method and resulted in 98.44 % accuracy according to Table 2.

```
Epoch 1/10
1/8 [==>...........................] - ETA: 0s - loss: 0.7509 - accuracy: 0.4375WARNING:tensorflow:From /home/mano
s/anaconda3/envs/tensorEnv/lib/python3.6/site-packages/tensorflow/python/ops/summary_ops_v2.py:1277: stop (from te
nsorflow.python.eager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
8/8 [==============================] - 19s 2s/step - loss: 1.1420 - accuracy: 0.5125 - val_loss: 0.6904 - val_accu
racy: 0.6875
Epoch 2/10
8/8 [==============================] - 18s 2s/step - loss: 0.6777 - accuracy: 0.6125 - val_loss: 0.6652 - val_accu
racy: 0.5781
Epoch 3/10
8/8 [==============================] - 18s 2s/step - loss: 0.6046 - accuracy: 0.6542 - val_loss: 0.4759 - val_accu
racy: 0.9219
Epoch 4/10
8/8 [==============================] - 18s 2s/step - loss: 0.3420 - accuracy: 0.8750 - val_loss: 0.2837 - val_accu
racy: 0.9219
Epoch 5/10
8/8 [==============================] - 18s 2s/step - loss: 0.3849 - accuracy: 0.8125 - val_loss: 0.2239 - val_accu
racy: 0.9844
Epoch 6/10
8/8 [==============================] - 19s 2s/step - loss: 0.2425 - accuracy: 0.8875 - val_loss: 0.1338 - val_accu
racy: 0.9688
Epoch 7/10
8/8 [==============================] - 18s 2s/step - loss: 0.2373 - accuracy: 0.8875 - val_loss: 0.1789 - val_accu
racy: 0.9688
Epoch 8/10
8/8 [==============================] - 18s 2s/step - loss: 0.2479 - accuracy: 0.9042 - val_loss: 0.1806 - val_accu
racy: 0.9844
Epoch 9/10
8/8 [==============================] - 18s 2s/step - loss: 0.1389 - accuracy: 0.9667 - val_loss: 0.0851 - val_accu
racy: 0.9844
Epoch 10/10
8/8 [==============================] - 18s 2s/step - loss: 0.1490 - accuracy: 0.9458 - val_loss: 0.1062 - val_accu
racy: 0.9844
```

Figure 2

***Table 2** Training process*

The validation metrics Epoch Loss and Epoch Accuracy is shown in Table 3 and Table 4 respectively reveals that there is overall good performance in general without the existence of overfitting neither underfitting.

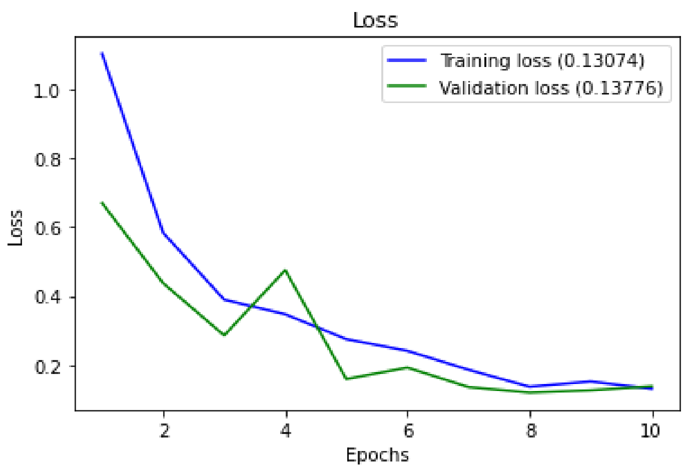


***Figure 3** Epoch Loss*

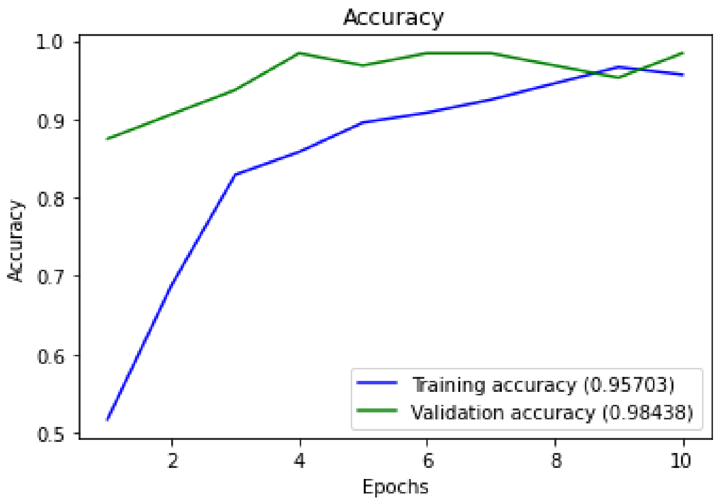


Figure 4 *Epoch Accuracy*

5.2 Confusion Matrix

The confusion matrix in Table 5 eliminates the case of False Positives (0) and that is a desirable result. On the other hand, we should try to eliminate the False Negatives (8) if we want to build a robust and reliable model. Finally, we can observe that we have 54 True Positives and 60 True Negatives results based on the validation dataset.

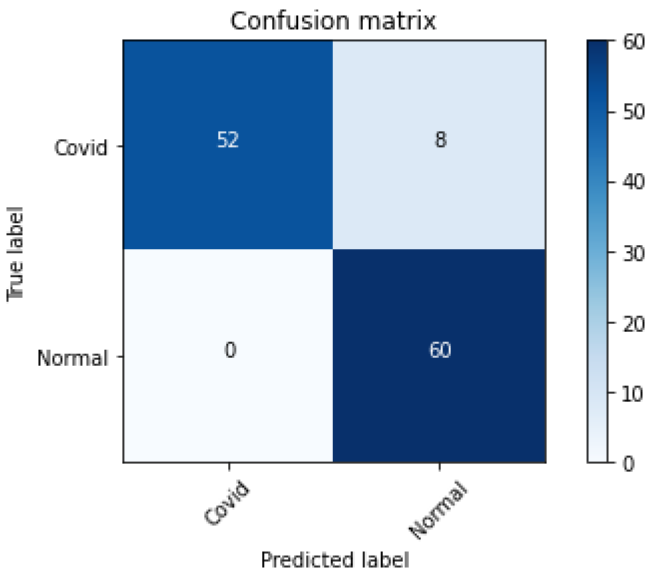


Figure 5 *Confusion Matrix*

6 Conclusions

Undoubtedly, COVID-19 is a notorious disease that cannot be denied. The improvement for more robust diagnosis tools is an essential weapon to bypass unexpected spread. So, the implementation of diagnosis systems varied, but many of them have low efficiency and are deemed vague. In our implementation, we tried to build a convolutional neural network with great performance and low complexity. The accuracy of the implementation was full of promise. Furthermore, the open data are plenty and that could be helpful by adding more cases and knowledge into our neural network.

There are several improvements that could be established. To begin with, the addition of classes is an important incentive. For example, we would like to add more cases from different types of pneumonia, making our model more robust with additional knowledge. Additionally, the implementation with different neural network architectures and the comparison between them is useful for a broad sense of the efficiency and the accuracy of the results.

Taking everything into consideration, Neural Networks is one of the most noteworthy fields in computer science combining a lot of different scientific fields, resolving many serious real-world problems. In medicine, there are many gaps in which neural networks can shed light on.

References

[1] "WHO Coronavirus Disease (COVID-19) Dashboard | WHO Coronavirus Disease (COVID-19) Dashboard." https://covid19.who.int/ (accessed Jan. 26, 2021).

[2] M. Ezhilan, I. Suresh, and N. Nesakumar, "SARS-CoV, MERS-CoV and SARS-CoV-2: A Diagnostic Challenge," *Meas. J. Int. Meas. Confed.*, vol. 168, Jan. 2021, doi: 10.1016/j.measurement.2020.108335.

[3] A. I. Khan, J. L. Shah, and M. Bhat, "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images," *Comput. Methods Programs Biomed.*, vol. 196, Apr. 2020, doi: 10.1016/j.cmpb.2020.105581.

[4] X. Xu *et al.*, "A Deep Learning System to Screen Novel Coronavirus Disease 2019 Pneumonia," *Engineering*, Oct. 2020, doi: 10.1016/j.eng.2020.04.010.

[5] T. Tuncer, S. Dogan, and F. Ozyurt, "An automated Residual Exemplar Local Binary Pattern and iterative ReliefF based corona detection method using lung X-ray image," *Chemom. Intell. Lab. Syst.*, vol. 203, Aug. 2020, doi: 10.1016/j.chemolab.2020.104054.

[6] M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images," *Informatics Med. Unlocked*, vol. 20, Jan. 2020, doi: 10.1016/j.imu.2020.100412.

[7] M. Farooq and A. Hafeez, "COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs," *arXiv*, Mar. 2020, Accessed: Jan. 26, 2021. [Online]. Available: http://arxiv.org/abs/2003.14395.

[8] J. Paul Cohen, P. Morrison, and L. Dao, "COVID-19 Image Data Collection," *arXiv*. arXiv, Mar. 25, 2020.

[9] "Chest X-Ray Images (Pneumonia) | Kaggle." https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia (accessed Jan. 26, 2021).