LUNGS CT-BASED CLASSIFICATION FOR COVID19-VS-NON COVID19

Author:- Bhimarao Dasari
email address:- bd20827@essex.ac.uk

I. Introduction

"Computed tomography or CT" nowadays is the useful manner mainly used for diagnosing imaging tests to create in detailed images of "internal organs" such as "Bones, soft tissues, blood vessels, injuries, damages and so on". Moreover, with the help of CT scan measures different types of "X ray" images from different types of angles around the body. Furthermore, after the Covid 19 pandemic situation arises this can diagnose "Covid 19 patients". On the other hand, due to privacy issues, publicly available "Covid 19 CT datasets" are difficult to get, which problems the research and development of "AI powered diagnosis methods" of Covid 19 based on Computed Tomography's. However, for this issue "UC San Diego and UC Berkeley " build Covid CT datasets that contain "349 Covid 19" images from 216 patients and "463 Non Covid 19" Computed Tomography. Here in this task we can perform binary image classification such as Covid and Non Covid and improve the performance of the results such as "Feature Extraction Transfer Learning and Fine Tuning Transfer Learning" by using appropriate metrics. Moreover, to use "Data Neural Networks" are including "Convolution neural network or CNN, Advanced CNNs, transfer learning". Hence, using Python programming language easily developed the code and fulfill all the requirements as per needed.

II. AIM

The main aim of the assignment is to perform binary image classification such as "Covid vs. Non Covid" and to improve the performance of the results of "Feature Extraction Transfer Learning and Fine Tuning Transfer learning" or shown in Table 1 by using appropriate metrics. Moreover, coding on "Data Neural Networks" including CNN, Advanced CNNs, Transfer Learning to fulfill all the requirements as per needed in this project work.

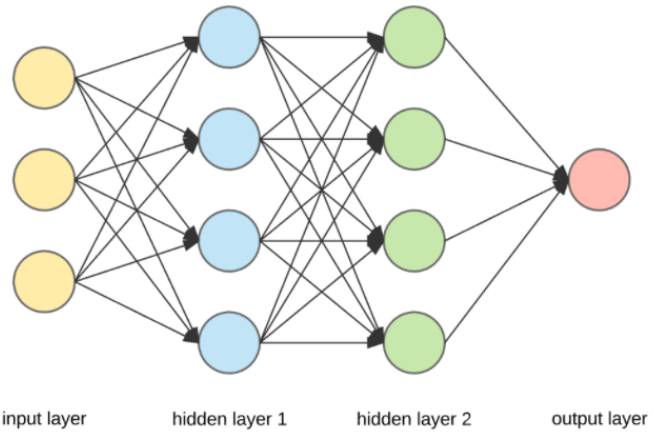

Fig. 1. Figure 1: Neural Network

III. Discussion

Implementing "Feature Extraction Transfer Learning and Fine Tuning Transfer Learning" python codes to improve train accuracy, test accuracy, validation accuracy, precision, Recall, F1 score, AUC. Improve this accuracy with the help of developing the codes into Python and by running the codes in Google Colab. Moreover, with the help of creating the codes in Google Colab and after running the codes help to show the increasing level of accuracy of the given metrics (Han et al. 2021). Use Deep Neural Networks including CNNs, advanced CNNs, and transfer learning helps to easily show the output values with the improvements of metrics. Here CNN is used for analysing visual imagery which is a class of "Deep Neural Networks". Moreover, with the help of this network, mathematical operation of two functions is to be done which produces a "third function" that expresses how the shape of one is modified by another. Here in deep learning CNN, advanced CNN, transfer learning is used or applied for connecting all the networks

and produces shapes where one is modified into another. Different deep neural networks are applied in this python programming for "Image classification and recognition" as of its "High Accuracy" (Hamdy et al. 2021). These networks here follow a "hierarchical model" that works on creating a network and gives out a fully connected layer and preceding the output. Moreover, with the help of the datasets can increase the metrics levels such as "Train Accuracy, Test Accuracy, Validation Accuracy and so on".

```
Epoch 1/25
18/18 [==============================] - 5s 249ms/step - loss: 1.9695 - accuracy: 0.5298 - val_loss: 0.6771 - val_accuracy: 0.5
625
Epoch 2/25
18/18 [==============================] - 5s 262ms/step - loss: 0.6169 - accuracy: 0.6544 - val_loss: 0.6133 - val_accuracy: 0.5
781
Epoch 3/25
18/18 [==============================] - 4s 238ms/step - loss: 0.5222 - accuracy: 0.7333 - val_loss: 0.5088 - val_accuracy: 0.7
188
Epoch 4/25
18/18 [==============================] - 4s 241ms/step - loss: 0.3718 - accuracy: 0.8281 - val_loss: 0.4859 - val_accuracy: 0.7
031
Epoch 5/25
18/18 [==============================] - 4s 237ms/step - loss: 0.2531 - accuracy: 0.9158 - val_loss: 0.5461 - val_accuracy: 0.7
188
Epoch 6/25
18/18 [==============================] - 4s 240ms/step - loss: 0.1652 - accuracy: 0.9544 - val_loss: 0.6316 - val_accuracy: 0.7
188
Epoch 7/25
18/18 [==============================] - 5s 252ms/step - loss: 0.1035 - accuracy: 0.9754 - val_loss: 0.6393 - val_accuracy: 0.7
188
Epoch 8/25
18/18 [==============================] - 4s 240ms/step - loss: 0.0670 - accuracy: 0.9877 - val_loss: 0.7671 - val_accuracy: 0.7
344
Epoch 9/25
18/18 [==============================] - 4s 247ms/step - loss: 0.0489 - accuracy: 0.9895 - val_loss: 0.7469 - val_accuracy: 0.7
344
Epoch 10/25
18/18 [==============================] - 4s 242ms/step - loss: 0.0294 - accuracy: 0.9965 - val_loss: 0.7405 - val_accuracy: 0.7
```

Fig. 2. Figure 2: Check the accuracy

IV. Simulations

Description of the dataset Create code in Python with the help of Jupyter Notebook for easily developing the code and running the code easily. Moreover, running the code in a Jupyter notebook shows the increased level of current accuracy and other metrics. Implement different types of Deep Neural Networks including CNNs, Advanced CNNs, and Transfer Learning to easily get the accuracy of the two datasets (Bouchareb et al. 2021). Moreover, with the application of "Deep Learning" one of the most important parts which helps to do the operation for diagnosing "Covid 19 CT images" from 216 patients and "463 Non Covid 19 CTs". Furthermore, the "Covid CT dataset" has 349 CT images containing from 216 patients whereas the "Non Covid CT dataset" has 397 CT images. However, with the help of these datasets can easily evaluate the code and give an output curve by fulfilling all the aspects.

V. Description of the data set

The train, test, validation or other accuracy and metrics can easily be found with the help of running the codes in Jupyter Notebook. The accuracy level

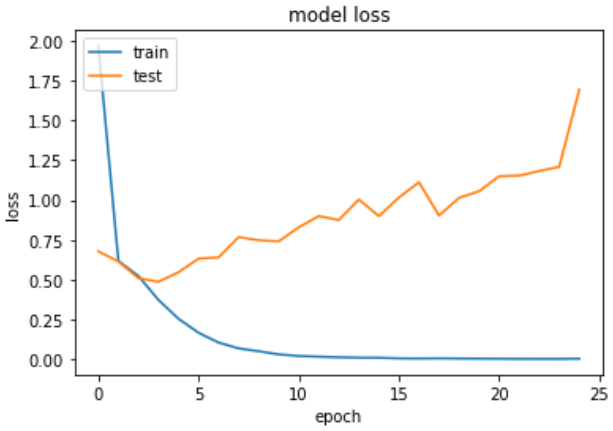

Fig. 3. Figure 3: Model Loss

compared with the given accuracy level is high (Kechagias-Stamatis et al. 2020). There have been two types of datasets provided such as "CTCovid and CTNonCovid". With the help of these datasets can easily find and calculate the accuracy level of train, test and validation accuracy and show the output. Based on these datasets, researchers can easily get the desired output as they are required or obtained or needed (Castiglioni et al. 2020). Here in this work we have to calculate and find the "accuracy rate" of train, test, validation, precision, recall and so on. "Deep learning" is used to execute the operation with the help of "Jupyter Notebook". Moreover, "Neural networks" are used to detect "CNNs, Advanced CNNs, Transfer Learning" one by one, as they can allow only one network to show the output at a time. "Deep learning" by using "Jupyter notebook" helps to show the "Training, testing and validation and others accuracy" result through the help of a "Deep Neural Network". "Neural networks" show the accuracy of an image file by detecting it through creating code in "Jupyter notebook".

A. *Encode the dataset*

"Encoding data" is the process which converts the data into a given sequence such as "Symbols, characters, alphabets and many more". Here encoding is done with the help of different types of code evaluation which keeps the data safe and secure in a folder or into a file. The files or folders are not readable until user can access it. Furthermore, after "Converting and transmitting" the data

through communication mediums such as “Neural networks”, the transmitting image files shows an image data curve. The other process is to be done such as manner where computer receives data which converts data back into the “normal format” (Rehouma et al. 2021). Here, encoding is done with the help of dataset which is reading all the images into a “Jupyter notebook”. However, while encoding the “test, train, and validate dataset” we can create a level under the image files. There are a numerous number of image files are given in the “CT-Covid and CT-NonCovid” folders. However, with the help of given folders can create “levels” and these “levels” are helpful to easily find the image file and read those files very easily. However, with the help of this easily split the data files into a “File, folder and levels”. This separation can help to search out the “files, folders and levels” very easily. Furthermore, to show the split or separation of data files can give comments such as “df.head”. Moreover, as all the images have different “sizes, shapes”, however, to fix the “image size” can give a maximum image size such as 128. Here, evaluates the “train, validates and tests data” files into “x and y axis” (Berenguer et al. 2020). Import “array, Tensorflow, and Keras” in the “Jupyter notebook” to easily show the percentage accuracy in the “data model” where x axis shows the number of epochs and y axis represents the accuracy in the data model. The curve shows different accuracy and the below figures and datasets are show the accuracy for ”train, test and validation and others” datasets and other accuracy and metrics. The curves shows the output values of deep learning by using “Keras, Tensorflow” in the Jupyter notebook.

B. *Result Obtained*

Get the output results of increasing level of “accuracy and other metrics”, first load all the given data files into the folders in “Jupyter Notebook”. Moreover, after that reads all the files for create “image datasets”. The “image dataset files” are converted into characters to easily read all the files easily with the help of “Jupyter notebook encoding process”. Levels are created in “Jupyter notebook” to find or search the image path very easily and to process the image file step by step to take the final output results. Encoding the dataset is for safe the data into a folder and it keeps the data safe from

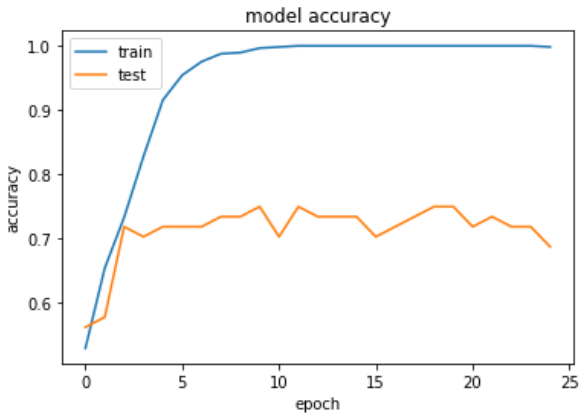

Fig. 4. Figure 4: The x-axis represents the number of epochs and the y-axis represents the percentage accuracy

hackers or unauthorized users (Fridadar et al. 2021). “Deep learning” is done through import “Tensorflow, Keras” in the “Jupyter notebook”. This can help to show the number of epochs “x axis and the y axis”. The “x axis and y axis” are represents the percentage accuracy in the “data model”. Moreover, in the “Jupyter notebook” set all the paths of the given datasets after that merge them into a folder.

```
print("Train Accuracy:\t", acc[-1])
print("Val Accuracy:\t", val_acc[-1])
print("Test Accuracy:\t", test_acc)
print("Precision:\t", precision)
print("Recall:\t\t", recall)
print("F1 Score:\t", f1score)
print("AUC:\t\t", auc)
```

```
Train Accuracy:  0.9982455968856812
Val Accuracy:    0.6875
Test Accuracy:   0.6785714285714286
Precision:       [0.75757576 0.6625    ]
Recall:          [0.48076923 0.88333333]
F1 Score:        [0.58823529 0.75714286]
AUC:             0.6778846153846154
```

Fig. 5. Figure 5: Accuracy and Metrics

C. *Conclusion*

This has been concluded that image processing with the help of deep learning is to be done here by using Jupyter Notebook. Here, in this study work “Deep neural networks” including “CNN, Advanced CNNs, and Transfer Learning” is to be done. For checking the accuracy validation of different matrices can create code in Jupyter Notebook and run the codes very easily. As per the given job card

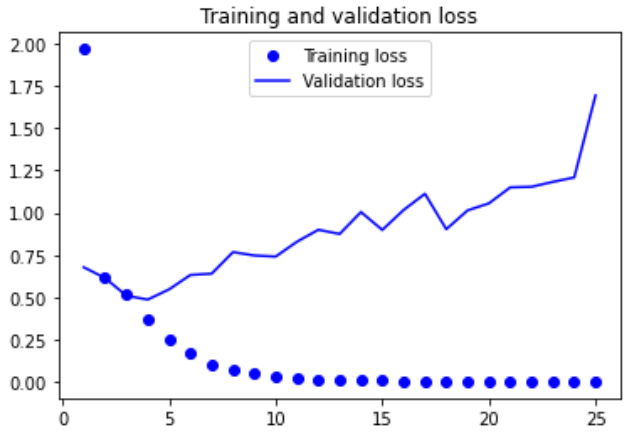

Fig. 6. Figure 7: Training and Validation Loss

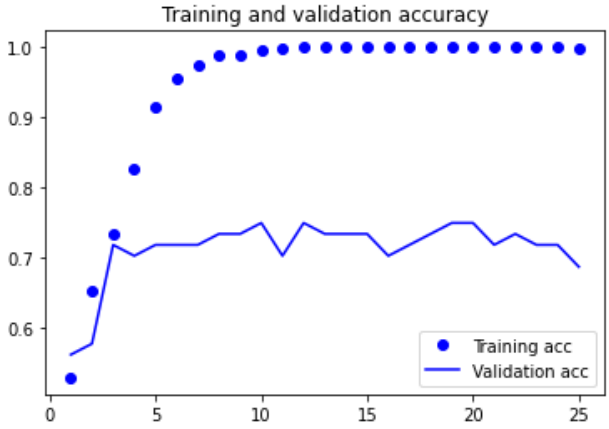

Fig. 7. Figure 6: Training and Validation Accuracy

the accuracy and other metrics are needed to be above compared with them. However, by analysing it the accuracy levels are high compared with them. However, with the help of creating codes in Jupyter Notebook easily shows the result of accuracy which is higher compared with them.

VI. Reference

References

[1] Berenguer, A.D., Sahli, H., Joukovsky, B., Kvasnytsia, M., Dirks, I., Alioscha-Perez, M., Deligiannis, N., Gonidakis, P., Sánchez, S.A., Brahimetaj, R. and Papavasileiou, E., "Explainable-by-design semi-supervised representation learning for COVID-19 diagnosis from CT imaging," arXiv preprint arXiv:2011.11719, 2020.

[2] Bouchareb, Y., Khaniabadi, P.M., Al Kindi, F., Al Dhuhli, H., Shiri, I., Zaidi, H. and Rahmim, A., "Artificial intelligence-driven assessment of radiological images for COVID-19," Computers in biology and medicine, p.104665, 2021.

[3] Castiglioni, I., Ippolito, D., Interlenghi, M., Monti, C.B., Salvatore, C., Schiaffino, S., Polidori, A., Gandola, D., Messa, C. and Sardanelli, F., "Artificial intelligence applied on chest X-ray can aid in the diagnosis of COVID-19 infection: a first experience from Lombardy," Italy. MedRxiv, 2020.

[4] Fridadar, M., Amer, R., Gozes, O., Nassar, J. and Greenspan, H., "COVID-19 in CXR: From detection and severity scoring to patient disease monitoring," IEEE journal of biomedical and health informatics, 2021.

[5] Hamdy, W., Elansary, I., Darwish, A. and Hassanien, A.E., "An Optimized Classification Model for COVID-19 Pandemic Based on Convolutional Neural Networks and Particle Swarm Optimization Algorithm," 2021.

[6] Han, C.H., Kim, M. and Kwak, J.T., "Semi-supervised learning for an improved diagnosis of COVID-19 in CT images," Plos one, 16(4), p.e0249450, 2021.

[7] Kechagias-Stamatis, O., Aouf, N. and Koukos, J., "Deep Learning Fusion for COVID-19 Diagnosis," medRxiv, 2020.

[8] Rehouma, R., Buchert, M. and Chen, Y.P.P., "Machine learning for medical imaging-based COVID-19 detection and diagnosis," International Journal of Intelligent Systems, 2021.