IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115211

Early Detection of Pneumonia in COVID-19 Patients Using CNN Algorithm

Kamakshi D Shanbhag[1], Greeshma G Sail[2], Arshi Prasad[3], Uzma Sulthana[4]

Student, BE, Department of ISE, Atria Institute of Technology, Bangalore, India [1-3]

Assistant Professor, Department of ISE, Atria Institute of Technology, Bangalore, India [4]

Abstract: Neural Networks (NN) are a subset of Machine Learning that is increasingly being employed in pre-processed image analysis. The CNN (Convolutional Neural Network) algorithm is a common NN technique that outperforms ANN in this project. The existing CNN models are Inception V3, ResNet50, MobileNet, and Xception [1], although they have been proven to be less accurate and time expensive. The H5 model is a new CNN model developed in our Project. A model that was originally created for facial detection and differentiation is currently being utilised to detect all objects with greater accuracy, focusing on five zones with variable pixel intensity scheme. The encouragingly high classification accuracy of our proposal implies that it can efficiently automate Pneumonia Detection in COVID-19 patients from radiograph images to provide a fast and reliable evidence of Pneumonia related COVID-19 infection in the lung that can complement existing COVID-19 diagnostics modalities.

Keywords: H5 Convolutional Neural Network model, Convolution Neural Network (CNN) architecture, COVID-19, Severe Acute Respiratory Syndrome corona virus 2 (SARS cov-2), deep learning based chest radiograph classification (DL-CRC), Tensorflow, Haar Cascade Classifiers, different pixel Intensity scheme, facial detection and distinction.

I.INTRODUCTION

We must train a deep learning model that can utilise the robust dataset generated by our suggested technique. We study the modern deep learning architectures suited for classification of normal, COVID-19, and other abnormal instances (e.g., pneumonia) because the problem may be viewed as a classification task of normal, COVID-19, and other abnormal cases (e.g., pneumonia). CNNs are the most powerful deep learning architecture for image classification, compared to other versions of deep learning architectures (e.g., Long Short Term Memory (LSTM), deep belief networks, and so on) and extreme learning machines.

II. EXISTING SYSTEM

Samples are taken from sites where the virus that causes COVID-19 is most likely to be found, such as the back of the nose or mouth, or deep within the lungs. Following the collection of a sample, the virus's RNA is extracted and converted to complementary DNA for testing. The PCR test entails repeatedly copying everything in between and binding sequences on the DNA that are only found in the virus. This technique is done numerous times, with each cycle doubling the target region. When amplification happens, a fluorescent signal is produced, and the test result is considered positive after the signal reaches a certain threshold. If no viral sequence is present, amplification will not occur, resulting in a negative result. Recommendations for testing are regularly updated by the Centres' for Disease Control and Prevention (CDC). [2][3]

III. LIMITATIONS OF THE EXISTING SYSTEM

The existing system for detecting COVID-19 using the above mentioned virus and antibody testing modalities is time-consuming and requires additional resources and approval, which can be a luxury in many developing communities. Hence, at many medical centres', the test kits are often unavailable. Due to the shortage of kits and false-negative rate [4][5] of virus and antibody tests, the authorities in Hubei Province, China momentarily employed radiological scans as a clinical investigation for COVID-19.

IV. SYSTEM REQUIREMENT SPECIFICATION

Software requirements: Anaconda navigator, Jupyter Notebook, Python 3.7 (Programming Language).
Libraries: Matplotlib, Seaborn, Numpy, Pandas, Keras, Tensorflow, Pillow, Open CV and OS.
Hardware Requirements:

- Processor: 64-bit 2.8GHz 8.00 GT/s, i3/i5/i7
- Laptop or PC
- Web Camera or Mobile Camera

- RAM: 8GB or higher

Operating System: Windows 8 or newer, 64-bit mac OS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 6+ and others.

V. EXISTING CNN MODELS

i. Inception V3: Szegedy et al proposed the Inception architecture in 2014. The original architecture was called GoogleLeNet. All the subsequent versions were called Inception Vn (n is the version number). Batch Normalization was added in Inception V2 as an improvement over Inception V1. In InceptionV3 model factorization methods were introduced as an improvement over V2. [6]

ii. ResNet50: In 2015 He et al proposed ResNet - The Residual Networks architecture. It has 50 convolutional layers with skip connections that help in improving the learning accuracy of the model. Also, it uses global averaging pooling instead of fully connected layers thereby reducing the model size.[7]

iii. MobileNet: In 2017 another CNN architecture called MobileNet was proposed by Howard et al. In this separable convolution have been arranged depth-wise and they apply the convolution operation on each color channel separately instead of taking them as a whole. The cost of computation gets reduced in this architecture.

iv. Xception: François Chollet developed Xception in 2017. This model can be considered as an improvised version of Inception as modules of Inception have been replaced with depth wise separable convolutions. This latest and accurate model scores upon speed and accuracy.[8]

VI. H5 MODEL OF CNN

The above CNN models are less responsive and are time consuming during training. Our H5 model created using five regions of a pre-processed image as shown in figure 1. H5 mapping consists of

i. LH Upper
ii. RH Upper
iii. Central
iv. LH Lower
v. RH Lower

Figure 2 shows the imposition of keras H5 model on our assignment. Scientific symbols needs to be suppressed for clarity. An array of right shape needs to be fed into the keras model. The length or number of images you can put into the array is determined by the first position in the shape tuple.

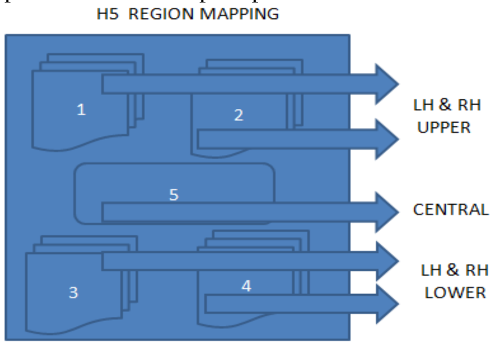
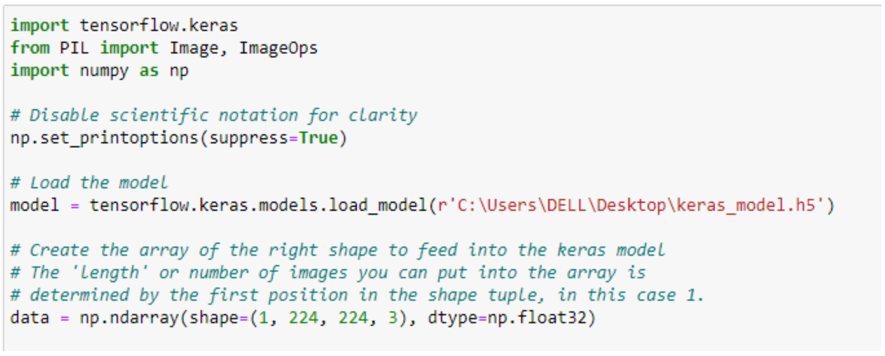


Figure 1 shows H5 region mapping scheme.

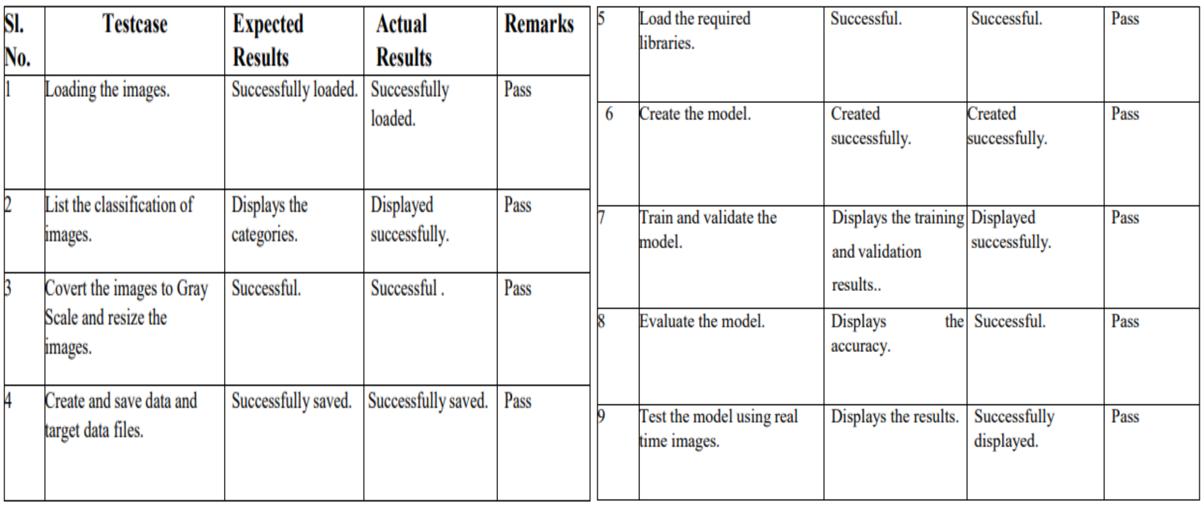


```
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = tensorflow.keras.models.load_model(r'C:\Users\DELL\Desktop\keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
```

Figure 2 shows the imposition of keras H5 model

VII. TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. Figure 3 shows the table containing different test cases and remarks.

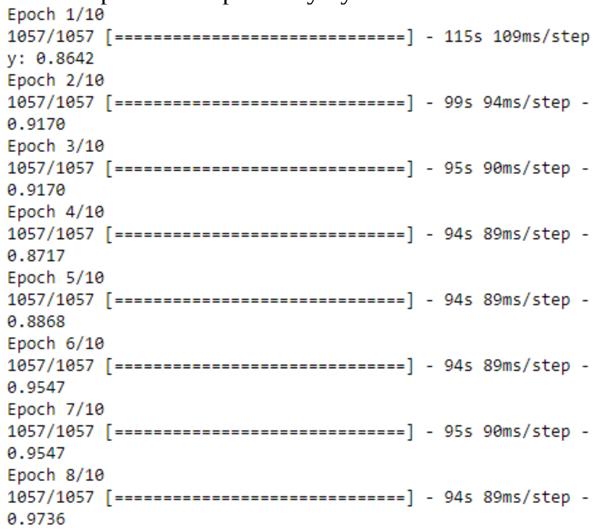


Sl. No.	Testcase	Expected Results	Actual Results	Remarks
1	Loading the images.	Successfully loaded.	Successfully loaded.	Pass
2	List the classification of images.	Displays the categories.	Displayed successfully.	Pass
3	Covert the images to Gray Scale and resize the images.	Successful.	Successful .	Pass
4	Create and save data and target data files.	Successfully saved.	Successfully saved.	Pass
5	Load the required libraries.	Successful.	Successful.	Pass
6	Create the model.	Created successfully.	Created successfully.	Pass
7	Train and validate the model.	Displays the training and validation results..	Displayed successfully.	Pass
8	Evaluate the model.	Displays the accuracy.	Successful.	Pass
9	Test the model using real time images.	Displays the results.	Successfully displayed.	Pass

Figure 3 shows the table containing different test cases and remarks.

VIII. RESULTS

After the train-test split the train images are randomly chosen by few instruction sets with a certain random state set. Real-time images are fed to the system and now the trained system takes decisions that are based on the historical image set. As shown in figure 4 training is done on 1057 samples which are obtained randomly by train test split.[9]Validation is carried out on 265 samples. Epoch is fixed to 10 and 1057 samples are evaluated epoch number of times and average training time noted on each step. One can note from figure 4 that the average time taken for training is 90ms and the accuracy is ever increasing which is appreciable as compared to the above models and RNN. Figure 5 shows the validation losses incurred during training. It can be noted that as the epoch value increase and with the increasing sample size the validation loss decreases. The summation of the training loss and validation loss gives the effective loss of our H5 model which is around 2% which can also be reduced by adopting innovative pre-processing techniques. Figure 6 shows the evaluation of the model or real time testing using images captured by the camera or camcorder. Figure 7 shows the array of the image being processed after normalization. Figure 8 shows the UI screen deployed on the cloud to select image for prediction of pneumonia. Figure 9 and 10 shows the result of a patient's diagnosis as normal and pneumonia positive respectively by the AI bot.



```
Epoch 1/10
1057/1057 [==============================] - 115s 109ms/step
y: 0.8642
Epoch 2/10
1057/1057 [==============================] - 99s 94ms/step -
0.9170
Epoch 3/10
1057/1057 [==============================] - 95s 90ms/step -
0.9170
Epoch 4/10
1057/1057 [==============================] - 94s 89ms/step -
0.8717
Epoch 5/10
1057/1057 [==============================] - 94s 89ms/step -
0.8868
Epoch 6/10
1057/1057 [==============================] - 94s 89ms/step -
0.9547
Epoch 7/10
1057/1057 [==============================] - 95s 90ms/step -
0.9547
Epoch 8/10
1057/1057 [==============================] - 94s 89ms/step -
0.9736
```

Figure 4 shows the process of training on the samples after train test split and time consumed at each step of training.

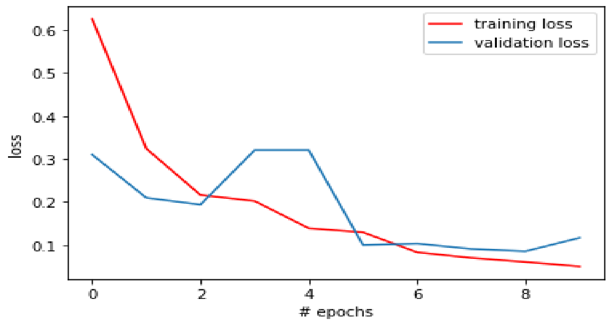
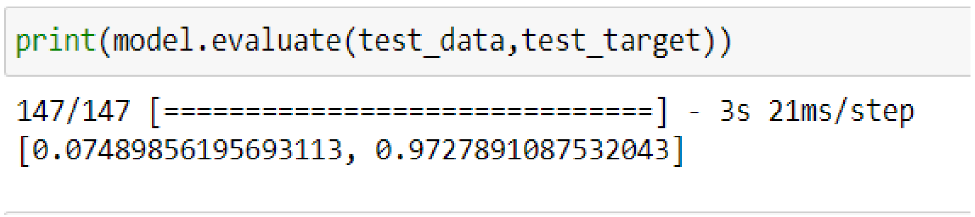
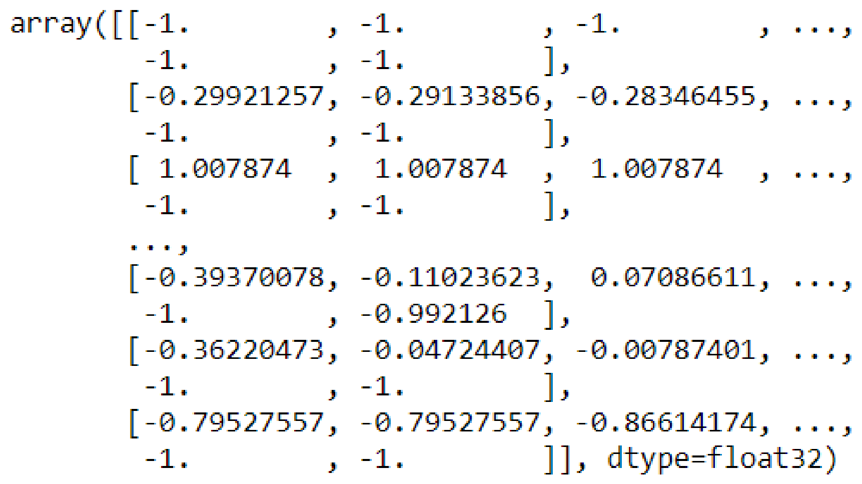


Figure 5 shows the validation losses incurred during training.



```
print(model.evaluate(test_data,test_target))

147/147 [==============================] - 3s 21ms/step
[0.07489856195693113, 0.9727891087532043]
```

Figure 6 shows the evaluation of the model or real time testing using images captured by the camera or camcorder.



```
array([[-1.        , -1.        , -1.        , ...,
        -1.        , -1.        ],
       [-0.29921257, -0.29133856, -0.28346455, ...,
        -1.        , -1.        ],
       [ 1.007874  ,  1.007874  ,  1.007874  , ...,
        -1.        , -1.        ],
       ...,
       [-0.39370078, -0.11023623,  0.07086611, ...,
        -1.        , -0.992126  ],
       [-0.36220473, -0.04724407, -0.00787401, ...,
        -1.        , -1.        ],
       [-0.79527557, -0.79527557, -0.86614174, ...,
        -1.        , -1.        ]], dtype=float32)
```

Figure 7 shows the array of the image being processed after normalization.

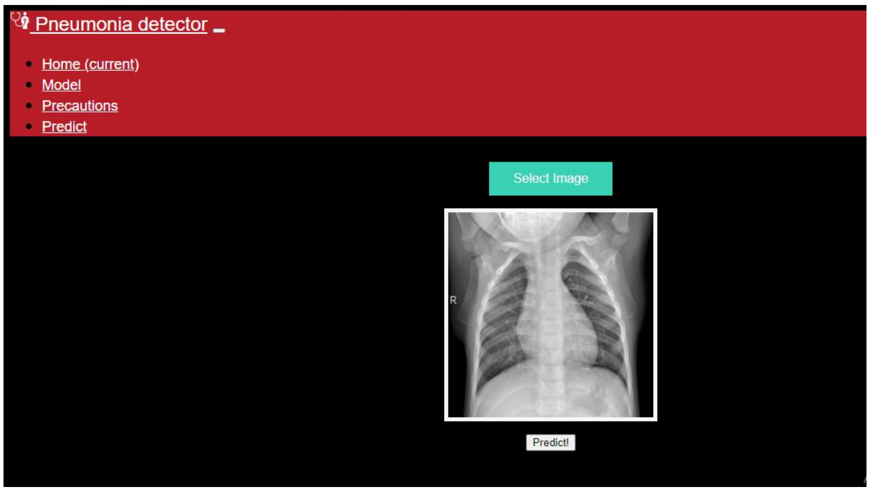


Figure 8 shows the UI screen deployed on the cloud to select image for prediction of pneumonia.

DOI: 10.17148/IJARCCE.2022.115211

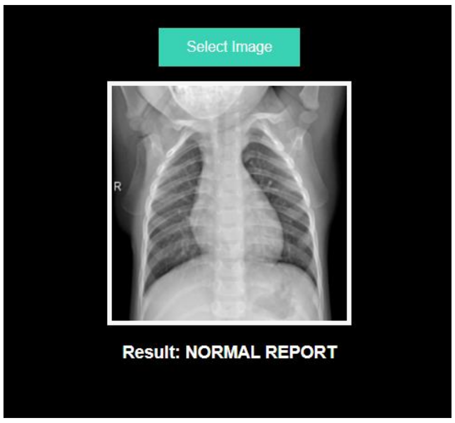


Figure 10 shows the result of a patient's diagnosis.

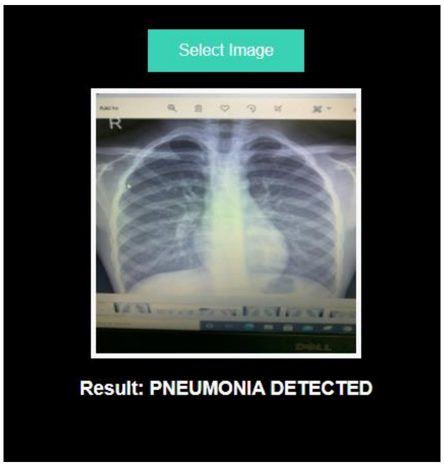


Figure 11 shows the result of a patient's diagnosis as pneumonia positive by the AI bot.

IX. CONCLUSIONS

If Pneumonia caused due to COVID-19 is not detected and treated early enough, it can be fatal. In order to screen a patient's state, RT-PCRs are insufficient. Lung screening is required since this illness attacks the lungs and causes pneumonia in those who are infected. Traditional screening methods have been surpassed by automated models. The CNN algorithm is one example of a detection algorithm for anomalies in the chest. The existing CNN models include Inception V3, ResNet50, MobileNet, and Xception. However, these were discovered to be less precise and time intensive. So we created a CNN at Atria that focuses on five features or regions of the target image. The summation of the training loss and validation loss gives the effective loss of our H5 model which is around 2%. Average time taken for training is 90ms and the accuracy is ever increasing which is appreciable at approx 98%.

REFERENCES

[1] Sadman Sakib, Tharat Tazrin, Mostafa M. Fouda, Zubair MD. Fadlullah, Mohsen Guizani., DL-CRC: Deep Learning-Based Chest Radiograph Classification for COVID-19 Detection.
[2] COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 In X-Ray Images. Authors: Ezz El-Din Hemdan, Marwa A. Shouman, Mohamed Esmail Karar.
[3] P. Chatterjee et al., "The 2019 novel coronavirus disease (Covid-19) pandemic: A review of the current evidence," Indian Journal of Medical Research, Supplement, vol. 151, no. 2–3. Indian Council of Medical Research, pp. 147–159, 2020, doi: 10.4103/ijmr.IJMR_519_20.
[4] G. Pascarella et al., "COVID-19 diagnosis and management: a comprehensive review," Journal of Internal Medicine, 2020, doi: 10.1111/joim.13091.
[5] C. P. West, V. M. Montori, and P. Sampathkumar, "COVID-19 Testing: The Threat of False- Negative Results," Mayo Clinic Proceedings. Elsevier Ltd, 2020, doi: 10.1016/j.mayocp.2020.04.004.
[6] D. Wang et al., "Clinical Characteristics of 138 Hospitalized Patients with 2019 Novel Coronavirus-Infected Pneumonia in Wuhan, China," JAMA - Journal of the American Medical Association, vol. 323, no. 11, pp. 1061–1069, Mar. 2020, doi: 10.1001/jama.2020.1585.

[7] C. Huang et al., “Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China,” The Lancet, vol. 395, no. 10223, pp. 497–506, Feb. 2020, doi: 10.1016/S0140- 6736(20)30183-5.
[8] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, “Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network,” IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1207–1216, May 2016, doi: 10.1109/TMI.2016.2535865.
[9] P. Rajpurkar et al., “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning,” Nov. 2017, [Online]. Available: http://arxiv.org/abs/1711.05225.