ijRASET

International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

www.ijraset.com

Call: ◯ 08813907089 | E-mail ID: ijraset@gmail.com

Automatic Detection of Covid using CT Scans

Nikhil Polkampally[1], S. Srinivas Yadav[2], Janaki Venkata Sai Ram[3], Akshay Kumar Talare[4], Dr. M. Shanker[5]

[1, 2, 3, 4]CSE, Keshav Memorial Institute of Technology, Narayanguda, Hyderabad

[5]Associate Professor, CSE, Keshav Memorial Institute of Technology, Narayanguda, Hyderabad

Abstract: Covid 19 is an infectious disease which is caused by SARS-CoV-2 coronavirus and spreads very quickly from an infected person. There are many ways to detect the presence of covid using classical methods like RT-PCR, Rapid tests, X rays and CT scans. This paper focuses on using CT scans to detect the presence of Covid 19 using deep learning. This method provides a faster, simpler and safer way to diagnose Covid 19. The deep learning model using ResNet50 with additional image preprocessing techniques with a customized dataset detects Covid 19 with an accuracy of 89%. This paper proposes an end-to-end application which provides a deep learning backend to test the CT Image and a Web Front-end for users to diagnose Covid 19 and get the result instantaneously.

Keywords: Covid-19, Deep Learning, ResNet50.

I. INTRODUCTION

In the living hard times of covid pandemic instant detection of covid-19 might be one of the most basic medical necessity. The main purpose of the paper is to provide an accurate deep learning model to detect the presence of Covid 19 in a given CT scan and to provide a user-friendly web page which can be put in real world use with minimal human effort. Existing methods of covid detection takes a huge amount of time and is expensive also. For example people traveling through countries might get delayed at check-in of airports and hence a quick and instant detection could play a vital role in traveling. In this paper we implement a deep learning backend using Resnet 50 convolutional neural network model. And we provide a user-friendly front-end web page to interact with users and upload the CT scans to get the results of covid detection. The developed application can be used to perform rapid and instant detection of COVID-19. This project aims to provide a scalable application for the deep learning models to detect covid using CT scans. This paper is structured as follows: Section 2 discusses literature which gives information about the existing methods and their benefits and drawbacks for each method that can be used for detection of presence of Covid 19. Section 3 discusses the background knowledge that is needed to understand the paper and the tools and techniques used in the paper to implement the proposed goals. Section 4 discusses the implementation details, which briefly depicts how the implementation has been done to achieve the proposed goals. Section 5 gives the results for the deep learning backend accuray with different metrics and the functioning of the frontend webpage. Section 6 concludes about the outcomes of the paper and finally section 7 discusses the future work.

II. LITERATURE SURVEY

One of the standard methods to diagnose Covid 19 is using RT-PCR[1][2] which stands for Reverse Transcription Polymerase Chain Reaction, where a sample is collected form the parts of the body of the infected patient like nose or throat, later the sample is treated will different chemical solutions to remove all extra substances like fats, proteins and other bodily substances and extract only RNA from the sample. The extracted RNA is reverse transcribed to DNA with help of some specific enzyme and then later using additional marker labels to finally detect the virus. As this we can observe here that this method takes a substantial amount of time and effort to detect the Covid 19 virus.

The other method used to detect Covid 19 was by using machine learning by Abdul Salam M, Taha S, Ramadan M [3] where they used a federated learning model with SGD optimizer with X-Ray images. They have achieved an accuracy of 91.6%.

Shanima Akter [4] proposes to study covid classifications using various deep learning pre-trained models like VGG16, VGG19, MobileNetV2, InceptionV3, NFNet, ResNet50, ResNet101, DenseNet, EfficientNetB7, AlexNet, and GoogLeNet on chest X rays and achieved best accuracy with MobileNetV2[5].

Pedro Silva [6] proposes covid detection in CT Images using deep learning model Efficient Net [7] and has performed cross dataset validation to test and has achieved an highest accuracy of 87.68%.

This literature survey reveals that the most of existing work for detecting is using RT-PCR or using chest X Ray to detect with machine learning models. Here we propose for a deep learning backend with a complete fronted webpage for real world use which will help in detecting Covid 19 in faster, simpler and safer way using CT Scans.

III. BACKGROUND

A. Python and Libraries

We used python programming language and its libraries to implement the deep learning backend. Python is a simpler language and easy to use language which provides high level abstractions with most of the functionality as it supports various machine learning and deep learning functions through various libraries such as TensorFlow, Keras and PyTorch.

PyTorch[9] library is a deep learning backend that we used to build a customized model for training. Nibabel[10] package is used to load the datasets which are in NIfTI[11] nii format to numpy arrays. PIL[12] Imaging library is an imaging library that we used to load the images and resize the image from numpy arrays. Sklearn[13] metrics library is used to get different metrics on the training and testing results.

B. Data

The dataset used is a customized dataset which is a merge of 2 similar dataset from two different sources. The first one is from Grand Challenge Covid 19 Dataset [14][15][16] which contains a total 14634 CT scans with their respective masks in nii format with each CT slice of size 512 x 512. The second dataset is from Medical Segmentation Covid 19 Dataset [17] which contains 9 volumes of CT scans of size 630 x 630 in total of 829 slices and 1 volume of CT scans of size 512 x 512 in total of 100 slices. The following gives the information about datasets in tabular form.

Dataset	Volumes	Patients	Slices	Shape
Grand Challenge CT[14][15][16]	199	50	13705	512 x 512
Medical Segmentation Data (volume 1-9)[17]	9	40	829	630 x 630
Medical Segmentation Data (volume 10)[17]	1	40	100	512 x 512

Table 1 : Datasets

C. Google Colab

For all the computation work that is data preparation, preprocessing, model building, data loading and testing and evaluation of model we use Google Colab Pro[18] which provides us a instance of Google Cloud Compute Instance with High RAM of 25 GB and Nvidia Graphics Tesla T4 of 16 GB GPU memory which is required to train over large datasets.

D. Go lang

Go programming language is a new programming language which is mainly used when concurrency is needed and mostly used at web-servers to handle millions of accesses to web servers per second because of its excellent scalability.

IV. IMPLEMENTATION

A. Data

First the given data in .nii format is extracted into images and masks respectively. This is done using a Python Nibabel [10] package, this package helps in converting the nii sclies into numpy arrays which are then converted to PIL [12] Images and Torch tensor for use in later stages .

The CT slices with 630 x 630 shape are resized to 512 x 512 shape to build a dataset with uniform size i.e 512 x 512 using NEAREST interpolation technique which resizes the images with minimal noise for these particular images. The same procedure is done with Masks. After extracting CT Scan Images and corresponding masks which show the regions of covid in the bit mask, the labels for each CT slice is given with the corresponding mask. If a bit is set in the bit mask then the label is given as 1 for covid positive and if no bit is in the bit mask then it is labeled as 0 for covid negative.

After preprocessing the data the dataset looks like the following

Dataset	Total	Covid	Non Covid	Shape
Grand Challenge Dataset[14][15][16]	13705	4851	8854	512 x 512
Medical Segmentation Dataset [17]	929	468	461	512 x 512
Merged Dataset	14634	5319	9315	512 x 512

Table 2 : Dataset info with Covid and Non Covid count

Split	Images
Train	9000
Test	4000
Validation	1634

Table 3 : Train, Test, Validation split

B. Deep Learning Backend

For the deep learning backend we used the ResNet50[19] model because it has greater depth compared to VGG16[20] models and is fairly lighter and more efficient than other deep learning models for this in particular. ResNet50[19] specifically focuses on the problem called overfitting and vanishing gradients which are major drawbacks of other deep learning models which reduces testing accuracy.

The ResNet[19] Model has skip connections which helps to capture the slightest and miniature details from the input data for a long depth as in case of other architectures the model usually misses the information that has been learnt in the initial layers. Image 1 shows how a skip connection in ResNet[19] Neural Network looks like.
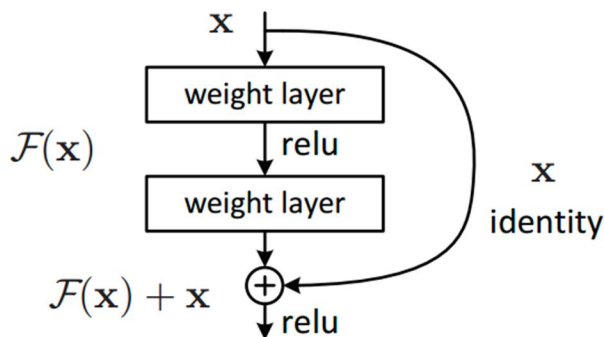

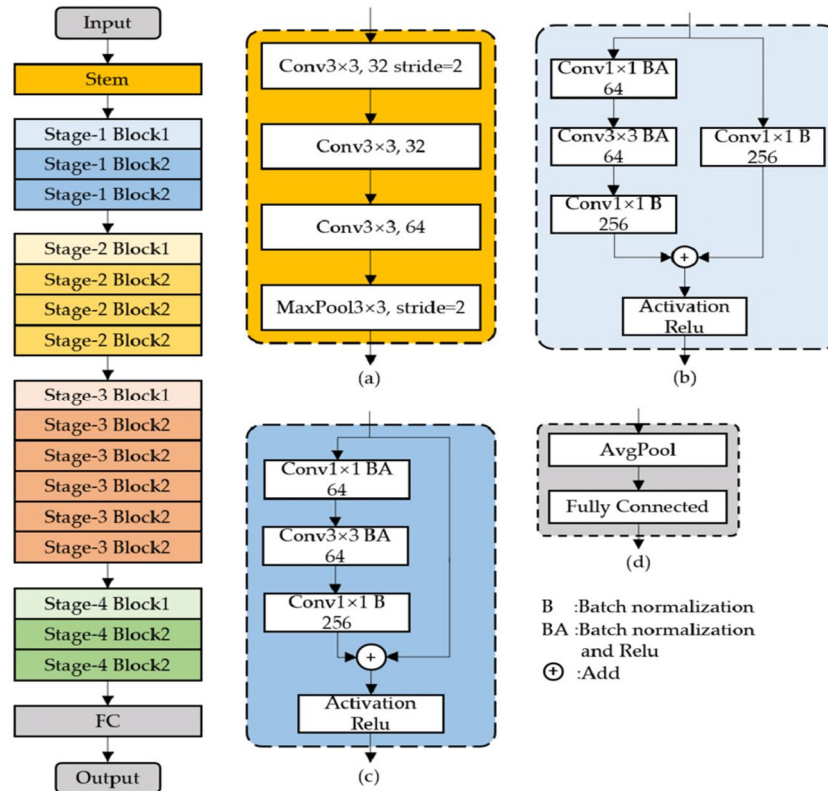Image 1: ResNet Skip Connection [21]

Image 2: ResNet Architecture [22]

Image 2 describes resnet model full blocked architecture with skipped connections.

As mentioned in Table 2 the data is split into 9000, 4000, 1634 for training, testing and validation. The base model of the ResNet model is used from PyTorch build-in models by replacing the input layer with shape of (N, 512, 512) where N is number of Images in input with shape 512 x 512 and output layer with (N, 2) for binary classification.

PyTorch DataLoaders are used to load the data in batches which help in learning efficiently with batch normalization[23] for ResNet[19]. The Batch sizes used are 20, 10, 10 for training, testing and validation respectively.

BinaryCrossEntropy[24] loss function is used to calculate the loss and Adam Optimizer[25] with a learning rate of 0.0001 is used with an LR scheduler called CosineAnnealingLR which decays the learning rate with epochs to backprograte the loss in neural network.

C.   Web Front End

We provide a web-frontend that is built using html and css for designing and Go lang for serving the webpage. Using Go lang as a server enables us to scale the server to a larger server without any hassle due to its excellent scalability, concurrency and load-balancing capabilities.

V.   RESULTS

A.   ResNet50 Model

	Accuracy	Loss
Training	96.92	0.0156
Testing	88.85	0.0273

Table 4 : Training and testing accuracy and loss for the ResNet50 model

Confusion Matrix	Positive	Negative
True	1198	2356
False	193	253

Table 5 : Confusion Matrix

Metric	Value
Accuracy	88.95
Precision	86.13
Recall	82.56
F1 Score	84.31

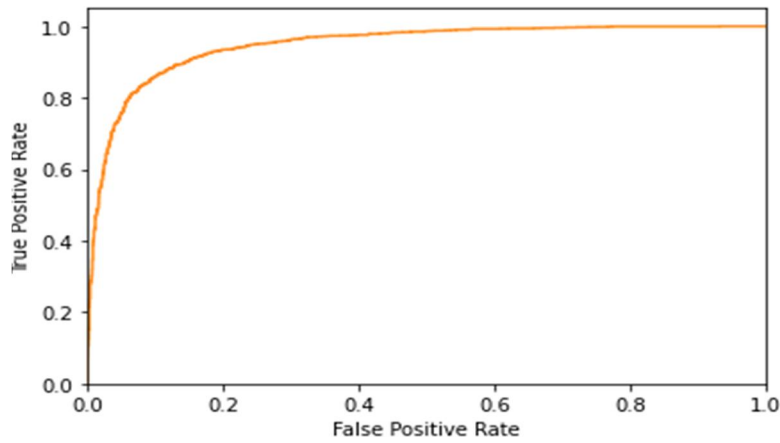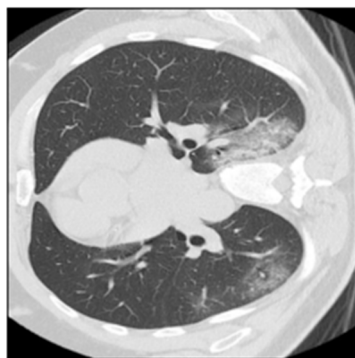Table 6 : Model Evaluation Metrics

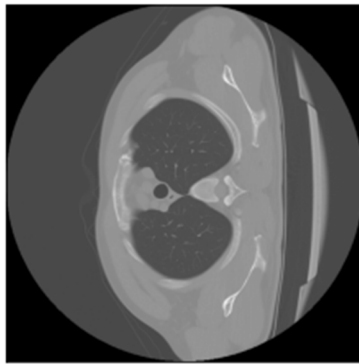Image 3 : ROC : 94.69

B. Test Outputs

Predicted – Covid Positive
Actual – Covid Positive

Image 4

Predicted – Covid Negative
Actual – Covid Negative

Image 5

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
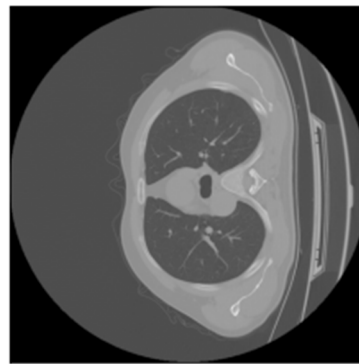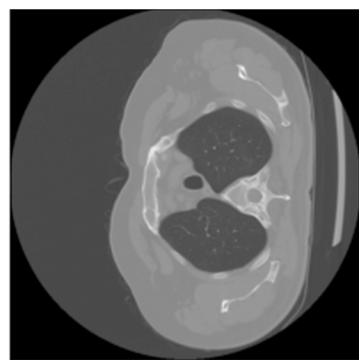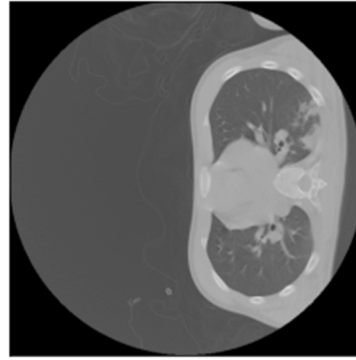*Volume 10 Issue VI June 2022- Available at www.ijraset.com*

Predicted – Covid Negative
Actual – Covid Negative
Image 6

Predicted – Covid Positive
Actual – Covid Negative
Image 7

Predicted – Covid Negative
Actual – Covid Positive
Image 8

Predicted – Covid Positive
Actual – Covid Positive
Image 9

Image 4, 5, 6,7,8,9 shows the actual and predicted output by model from the test dataset.

C. *Web Front-end Outputs*

Automatic Detection of COVID-19 using CT scans

Automatic Detection of COVID-19 using CT scans
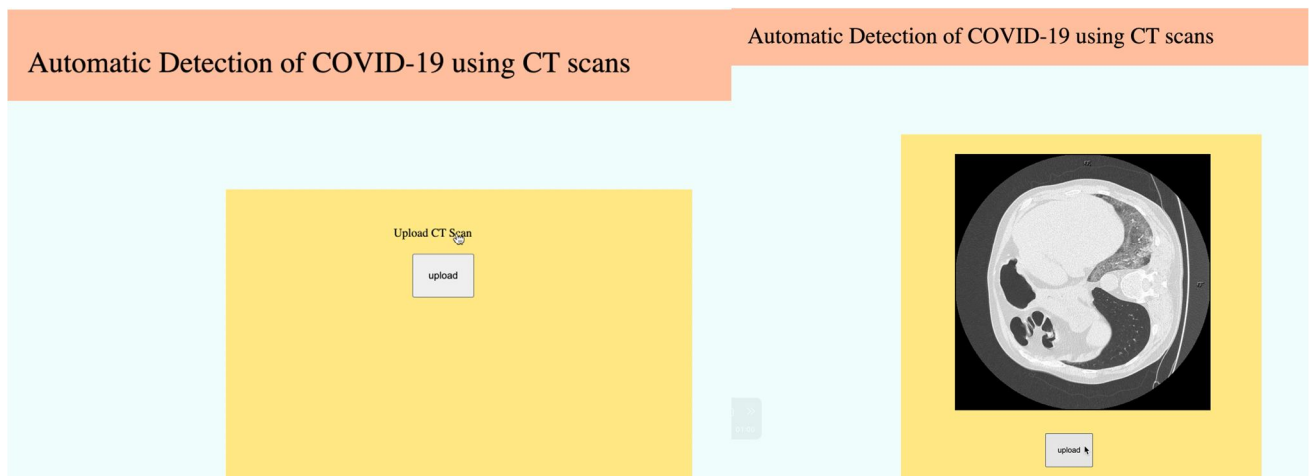
Upload CT Scan

upload

upload

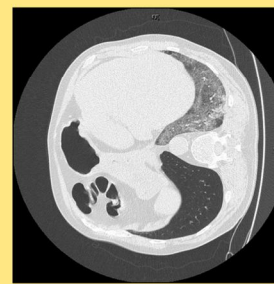Image 10 : Home Page

Image 11 : Upload CT Scan

Automatic Detection of COVID-19 using CT scans

Automatic Detection of COVID-19 using CT scans

Covid Negative

Covid Positive

Image 12 : Covid Negative Result   Image 13 : Covid Positive Result

Image 10 shows the Web Front-end Home page running at Localhost (127.0.0.1:8080) where a user can upload a CT scan using the upload button.

Image 11 shows the selected Image from the Users device and is ready to upload it to the website which will forward it to the deep learning backend.

Image 12 shows the results for Covid Negative CT scan in Green Color.

Image 13 shows the results for Covid Positive CT scan in Red Color.

VI.   CONCLUSION

In the proposed method we implemented the Covid 19 test backend using Deep Learning   which does the classification task with accuracy of 89%, which will do the job predicting a given CT Scan Image for presence of Covid 19 virus. We have implemented Web Front-end which takes the image as input and forwards the image to the deep learning backend and does the classification and displays the result in the   website.   Therefore this website successfully serves a webpage which can be used to check if a person has been infected with covid or not with a single touch of upload in the website of the CT Scan Image and hence the result can be known in   less than a minute. Therefore this method of testing for presence of Covid 19 is instantaneous and very safe.

VII.   FUTURE WORK

This paper as mentioned as does only classification task for COVID CT, for further enhancements we aim to do the following tasks

1) Implement Segmentation backend for COVID which will help us to show where the COVID is actually present in the CT scan.

2) COVID Detection Application for Android and IOS.

VIII.   ACKNOWLEDGEMENTS

Thanks to our mentor Dr. Shankar for constant support and guidance during the project. Thanks to KMIT for providing all the resources and help to complete the project.

REFERENCES

[1] Mo Y, Wan R, Zhang Q. Application of reverse transcription-PCR and real-time PCR in   nanotoxicity research. Methods Mol Biol. 2012;926:99-112. doi:10.1007/978-1-62703-002-1_7

[2] Ai Tang Xiao, Yi Xin Tong, Sheng Zhang, Profile of RT-PCR for SARS-CoV-2: A Preliminary Study From 56 COVID-19 Patients, Clinical Infectious Diseases, Volume 71, Issue 16, 15 October 2020, Pages 2249–2251, https://doi.org/10.1093/cid/ciaa460

[3] Abdul Salam M, Taha S, Ramadan M (2021) COVID-19 detection using federated machine learning. PLOS ONE 16(6): e0252573. https://doi.org/10.1371/journal.pone.0252573

[4] Akter S, Shamrat FMJM, Chakraborty S, Karim A, Azam S. COVID-19 Detection Using Deep Learning Algorithm on Chest X-ray Images. Biology (Basel). 2021 Nov 13;10(11):1174. doi: 10.3390/biology10111174. PMID: 34827167; PMCID: PMC8614951.

[5] Sandler, Mark & Howard, Andrew & Zhu, Menglong & Zhmoginov, Andrey & Chen, Liang-Chieh. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 4510-4520. 10.1109/CVPR.2018.00474.

[6] Pedro Silva, Eduardo Luz, Guilherme Silva, Gladston Moreira, Rodrigo Silva, Diego Lucio, David Menotti,COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis,Informatics in Medicine Unlocked,Volume 20,2020,100427,ISSN 2352-9148,https://doi.org/10.1016/j.imu.2020.100427.

[7] Tan, Mingxing and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." ArXiv abs/1905.11946 (2019): n. Pag.

[8] Abadi, Martín, et al. "{TensorFlow}: A System for {Large-Scale} Machine Learning." 12th USENIX symposium on operating systems design and implementation (OSDI 16). 2016.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: an imperative style, high-performance deep learning library. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, Article 721, 8026–8037.

[10] Nibabel Python library https://nipy.org/nibabel/

[11] Cox, R.W. & Ashburner, John & Breman, Hester & Fissell, Kate & Haselgrove, C. & Holmes, C.J. & Lancaster, J.L. & Rex, D.E. & Smith, S.M. & Woodward, J.B. & Strother, Stephen. (2004). A (sort of) new image data format standard: NiFTI-1. 10th Annual Meeting of the Organization for Human Brain Mapping. 22.

[12] Harshada, Ms & Snehal, & Shitole, Sanjay & Pranaya, Mhatre & Suchita, Kadam & Shweta, Ghanate & Kurle, Darshana. (2016). Python Based Image Processing.

[13] Pedregosa, Fabian & Varoquaux, Gael & Gramfort, Alexandre & Michel, Vincent & Thirion, Bertrand & Grisel, Olivier & Blondel, Mathieu & Prettenhofer, Peter & Weiss, Ron & Dubourg, Vincent & Vanderplas, Jake & Passos, Alexandre & Cournapeau, David & Brucher, Matthieu & Perrot, Matthieu & Duchesnay, Edouard & Louppe, Gilles. (2012). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 12.

[14] Roth H, Xu Z, Diez CT, Jacob RS, Zember J, Molto J, Li W, Xu S, Turkbey B, Turkbey E, Yang D, Harouni A, Rieke N, Hu S, Isensee F, Tang C, Yu Q, Sölter J, Zheng T, Liauchuk V, Zhou Z, Moltz J, Oliveira B, Xia Y, Maier-Hein K, Li Q, Husch A, Zhang L, Kovalev V, Kang L, Hering A, Vilaça J, Flores M, Xu D, Wood B, Linguraru M. Rapid Artificial Intelligence Solutions in a Pandemic - The COVID-19-20 Lung CT Lesion Segmentation Challenge. Res Sq [Preprint]. 2021 Jun 4:rs.3.rs-571332. doi: 10.21203/rs.3.rs-571332/v1.

[15] An P, Xu S, Harmon SA, Turkbey EB, Sanford TH, Amalou A, Kassin M, Varble N, Blain M, Anderson V, Patella F, Carrafiello G, Turkbey BT, Wood BJ (2020). CT Images in Covid-19 [Data set]. The Cancer Imaging Archive. DOI: https://doi.org/10.7937/tcia.2020.gqry-nc81

[16] Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M, Tarbox L, Prior F. The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository, Journal of Digital Imaging, Volume 26, Number 6, December, 2013, pp 1045-1057. DOI: 10.1007/s10278-013-9622-7.

[17] Medical Segmentation http://medicalsegmentation.com/covid19/

[18] Google Colab Pro https://colab.research.google.com/notebooks/pro.ipynb

[19] Amaya Rodríguez, Isabel & Civit-Masot, Javier & Luna-Perejon, Francisco & Duran-Lopez, Lourdes & Rakhlin, Alexander & Nikolenko, Sergey & Kondo, Satoshi & Laiz, Pablo & Vitrià, Jordi & Seguí, Santi & Brandao, Patrick. (2021). ResNet. 10.1007/978-3-030-64340-9_12.

[20] X. Zhang, J. Zou, K. He and J. Sun. "Accelerating Very Deep Convolutional Networks for Classification and Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 38, No. 10, pp. 1943-1955, 1 Oct. 2016. doi: 10.1109/TPAMI.2015.2502579.

[21] Deep Residual Learning for Image Recognition https://arxiv.org/abs/1512.03385v1

[22] Automatic Detection and Classification of Steel Surface Defect Using Deep Convolutional Neural Networks - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-architecture-of-ResNet-50-vd-a-Stem-block-b-Stage1-Block1-c-Stage1-Block2_fig4_349646156 [accessed 13 Jun, 2022]

[23] Ioffe, Sergey & Szegedy, Christian. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

[24] Ruby, Usha & Yendapalli, Vamsidhar. (2020). Binary cross entropy with deep learning technique for Image classification. International Journal of Advanced Trends in Computer Science and Engineering. 9. 10.30534/ijatcse/2020/175942020.

[25] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)