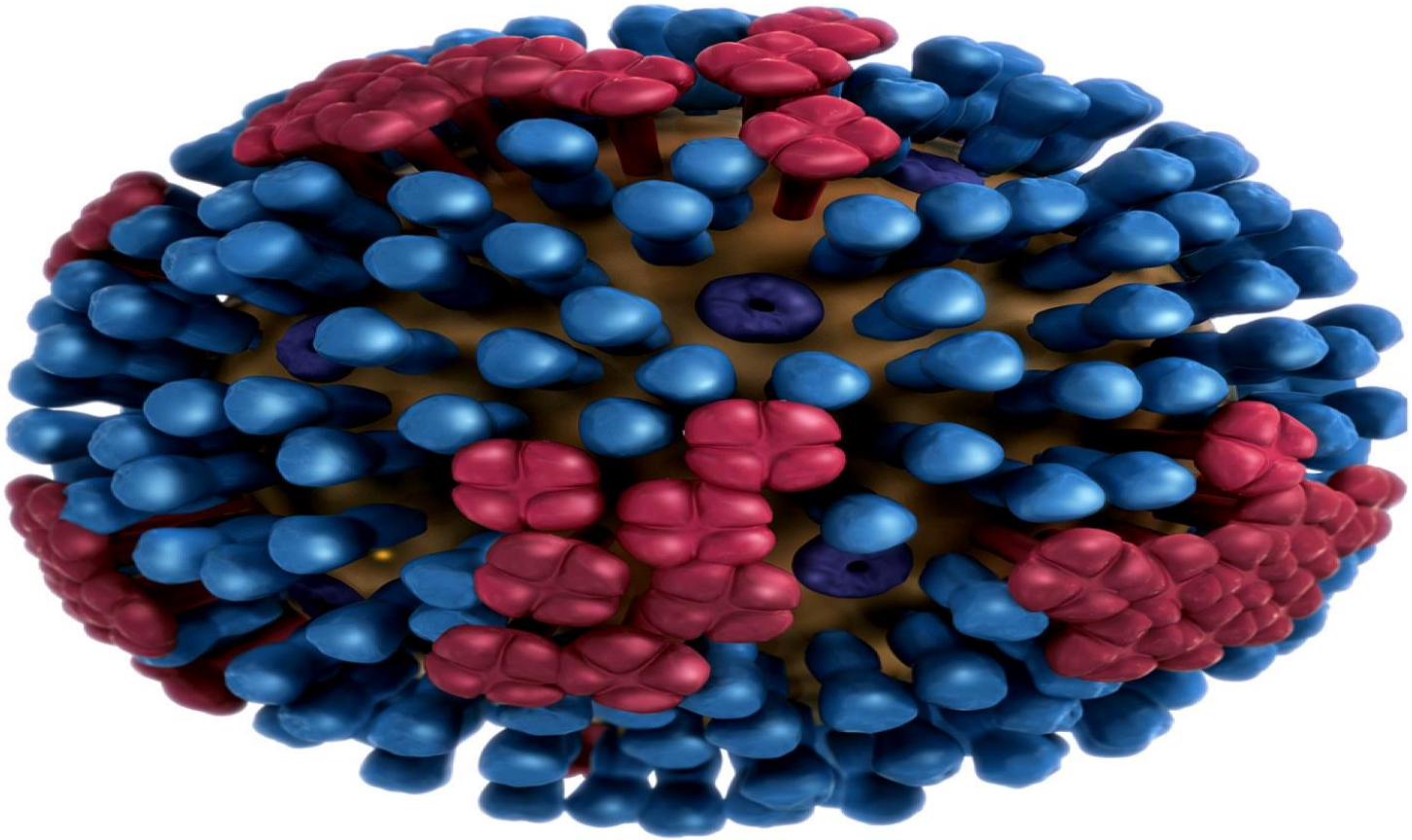

CT Scan Image Classification

Name:- Ashwin Savane

Date:- 4th November 2023

Course:- Full Stack Data Science Program



CT Scan Image Classification

Project Description

- In this project, we aim to develop an AI solution to classify CT scan images as COVID-19 positive(COVID) or negative(Non-COVID).
- The COVID-19 pandemic has highlighted the need for efficient diagnosis, and CT scans have shown promise in aiding detection. Leveraging deep learning techniques, our project focuses on training a model to analyze CT scan images and provide accurate predictions for COVID-19 infection.
- The developed model can assist healthcare providers by offering an additional diagnostic resource, potentially contributing to faster and more accurate patient management during a critical time.

Dataset Description

- This dataset contains 1252 CT scans that are positive for SARS-CoV-2 infection (COVID-19) and 1230 CT scans for patients non-infected by SARS-CoV-2, 2482 CT scans in total.
- These data have been collected from real patients in hospitals from Sao Paulo, Brazil.
- The aim of this dataset is to encourage the research and development of artificial intelligent methods which are able to identify if a person is infected by SARS-CoV-2 through the analysis of his/her CT scans.

Steps to Run this Project

- In order to run this project ,We need to follow this some step:-

- Clone or Download the project from this GitHub repository.
- Download the image data file & save it to the root folder of the project
- After that run Code file in your desired notebook e.g :-Jupyter Notebook, Google Colab & VS code.

Approach to the Problem

1.Data Preprocessing:

- Downloaded the data.zip file and extracted the COVID and Non-COVID directories.
- Loaded the images into list.
- Resized the image into fixed sized of 224 X 224 and then converting images to RGB mode.
- Normalized the pixel values of images.

2.Data Augmentation:

- Applied data augmentation techniques on images using Keras preprocessing “ImageDataGenerator” function.
- Included rotation, shifting, shear, zoom, and flipping to increase data diversity.

3. Data Generator & Splitting the data:

- Combining COVID and Non-COVID data with corresponding labels.
- Splitting the data into Training and validation sets.

4.Model Building:

- We used ResNet50 Model architecture which offers better performance and capacity to capture complex features.
- Adding custom layers like Average pooling 2D layer, Dense Layers
- Frozing the layers of the ResNet base model to avoid overwriting learned features.
- Compiled the model using binary cross-entropy, Adam optimizer and performance metrics as accuracy.

5.Model Training:

- Defined early stopping and model checkpoint callbacks.
- Trained the model using fit method with data generators for training and validation sets.
- Saved the best model.

6.Model Evaluation & Prediction:

- Loaded the best saved model.
- Evaluated model using metrics like accuracy, precision, recall, F1-Score, and confusion matrix on validation set.
- Loaded a test image, preprocessed it and used model to predict the class whether COVID or Non-COVID of the test image.

7.Fine Tuning the best model:

- Firstly, We fine-tuned the model by unfreezing the last 20 layer's of pretrained ResNet50 model & running for the 60 epochs.
- Then again, We adjusted to 80 epochs and by unfreezing last 40 layer's
- Both times the model resulted 77.86% Accuracy i.e almost 18% increase in accuracy of ResNet50 Model.

Last Approach of the Project

Tech Stack Used:

1. Python
2. Google Colab, Jupyter Notebooks & VS code
3. TensorFlow & Keras.
4. Visualization libraries like Matplotlib.Pyplot, and Seaborn.
5. NumPy, Pandas, PIL(Pillow module), and Sklearn.

References:

Scientific Report

ScienceDirect

Keras

TensorFlow

Sklearn