Problem Statement:

CT Scan Image Classification (SARS -CoV2 Infection)

Data Overview:

The given Dataset contains 1251 CT Scan of SARS-CoV2 infection (Covid19)

And 1229 CT Scan Non-Infected by SRAS-CoV2. Collected from real patients in hospitals .

AIM:

Aim of this project is to Build a Deep Neural Model which is able to identify if a person is infected by COVID-19 or Not through the analysis of His/Her CT scans.

Problem Approach:

- First thing, which comes in mind is which platform or which model to use for the Project.
- As we already instructed to use some specific models (Resnet).
- Collecting the Data (CT Scan Images).
- Carefully Read the given Instructions .
- Now time to understand the different model architecture and there work.
- Major Problem with any deep learning model is the Computational burden. As I don't have any device with higher GPU, required to train deep neural models. So I build and train my model in GOOGLE COLAB a free cloud service provided by google.
- But google colab provide limited RAM and GPU for a limited time session. So cant train models with large number of layers, or try different models in a single session.
- All the CT Scan images is not of the same size. So resize the images is one factor. And images are in grey scale and Resnet model takes 3 channels so further needs image transformation.
- As the data is not enough, there is a chance of overfitting of the model, that's why some image augmentation techniques is used to reduce overfitting.
- Implement the earlystopping function to stop the model if there is no improvements in model performance so it reduces model training time.also implement modelcheckpoint to save the model with its max improvements.

- This is a Medical project for COVID infected and for Non infected. So the first parameter I observe during the model build is the False Negative Rate.
- Train different Resnet model with different batch size and different epochs.
- And plot the model Accuracy and model Loss . By observing the train and validation data and considering model overfitting on the graph further tried with different epoch size and batch size. Larger batch size takes less time to train. Until satisfactory result is achieved.
- As I mention earlier the main parameter to evaluate this project performance in my opinion is the False Negative Rate.
- So plotted the confusion matrix with model validation and predicted data.
- And Finalize the model which gives less False Negative Rate.
- Next step is to deploy the model.
- So now have think for a framework which I can use for deployment of the model.
- I choose HTML and CSS for the front end of my project. And used Flask framework as the backend of my model deployment.
- Deployed the model in local server by creating the virtual python environment and evaluate the performance.

Remarks:

As I observe some CT scan images of covid positive patients is very much similar to the CT scan images of Covid negative patients. Few differences in their pixel values. So that's why some misclassification happen.

The problem of this project is properly addressed and implement all the parameters as instructed. Achieved False Negative: 4 with Final Accuracy: 97%.

Every documents is attached – entire code , saved model, webapp, requirement file,redme file etc ,and uploaded in github.