**SIIM-FISABIO-RSNA COVID-19 Detection**

## (Identify and localize COVID-19 abnormalities on chest radiographs)

* Jay Bhakhar

1. **Overview**

**- Description**

COVID-19 reasons full-size sickness and death. Like different pneumonias, pulmonary infection with COVID-19 consequences in infection and fluid within the lungs. COVID-19 appears very similar to different viral and bacterial pneumonias on chest radiographs, which makes it difficult to diagnose.

**- Evaluation**

**There are two labels**

* Study-level labels
* Image-level labels

### Study-level labels

Studies inside the take a look at set may incorporate a couple of label. They are as follows:

"negative", "typical", "indeterminate", "atypical"

* **Image-level labels**

Images in the test set may contain more than one object. For each object in a given test image, you must predict a class ID of "opacity", a confidence score, and bounding box in format xmin ymin xmax ymax. If you predict that there are NO objects in a given image, you should predict none 1.0 0 0 1 1, where none is the class ID for "No finding", 1.0 is the confidence, and 0 0 1 1 is a one-pixel bounding box.

## Submission File

The submission file should contain a header and have the following format:

Id,PredictionString

2b95d54e4be65\_study,negative 1 0 0 1 1

2b95d54e4be66\_study,typical 1 0 0 1 1

2b95d54e4be67\_study,indeterminate 1 0 0 1 1 atypical 1 0 0 1 1

2b95d54e4be68\_image,none 1 0 0 1 1

2b95d54e4be69\_image,opacity 0.5 100 100 200 200 opacity 0.7 10 10 20 20

etc.

1. Data

**Data Description**

We are identifying and localizing COVID-19 abnormalities on chest radiographs. This is an object detection and classification problem

For each test image, you will be predicting a bounding box and class for all findings. If you predict that there are no findings, you should create a prediction of "none 1 0 0 1 1" ("none" is the class ID for no finding, and this provides a one-pixel bounding box with a confidence of 1.0).

Further, for each test study, you should make a determination within the following labels:

'Negative for Pneumonia' 'Typical Appearance' 'Indeterminate Appearance' 'Atypical Appearance'

**Data Explorer(128.51 GB)**

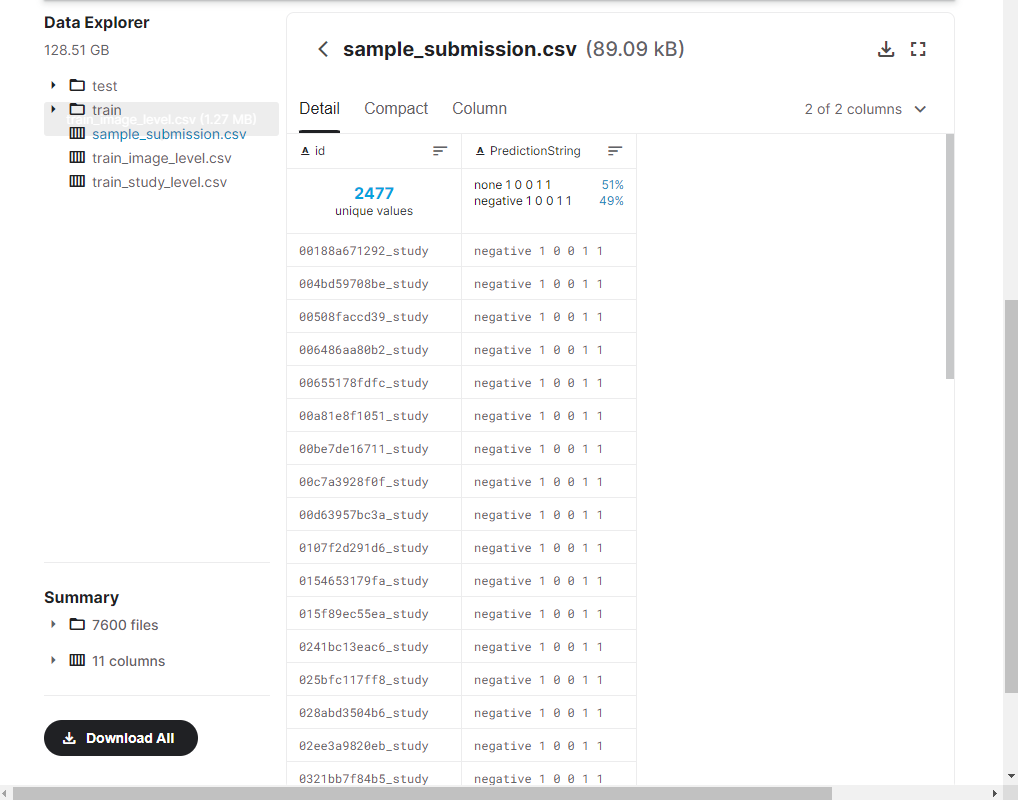
Two main directories test and train. In test directory got 1214 directories. In train directory got 6054 directories and 3 .csv files.

Those are 3 .csv files.

* sample\_submission.csv
* train\_image\_level.csv
* train\_study\_level.csv

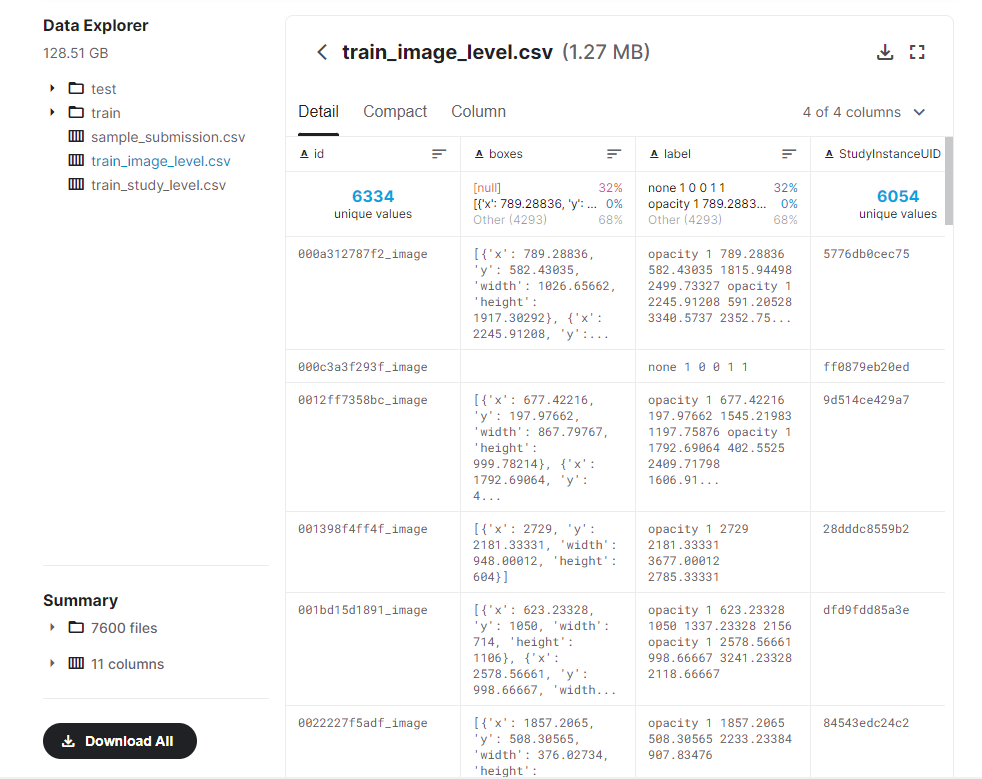
- Details of sample\_submission.csv

* two columns
  + id :- 2477 unique values
  + PredictionString :-
    - none 1 0 0 1 1 (51%)
    - Negative 1 0 0 1 1 (49%)



- Details of train\_image\_level.csv

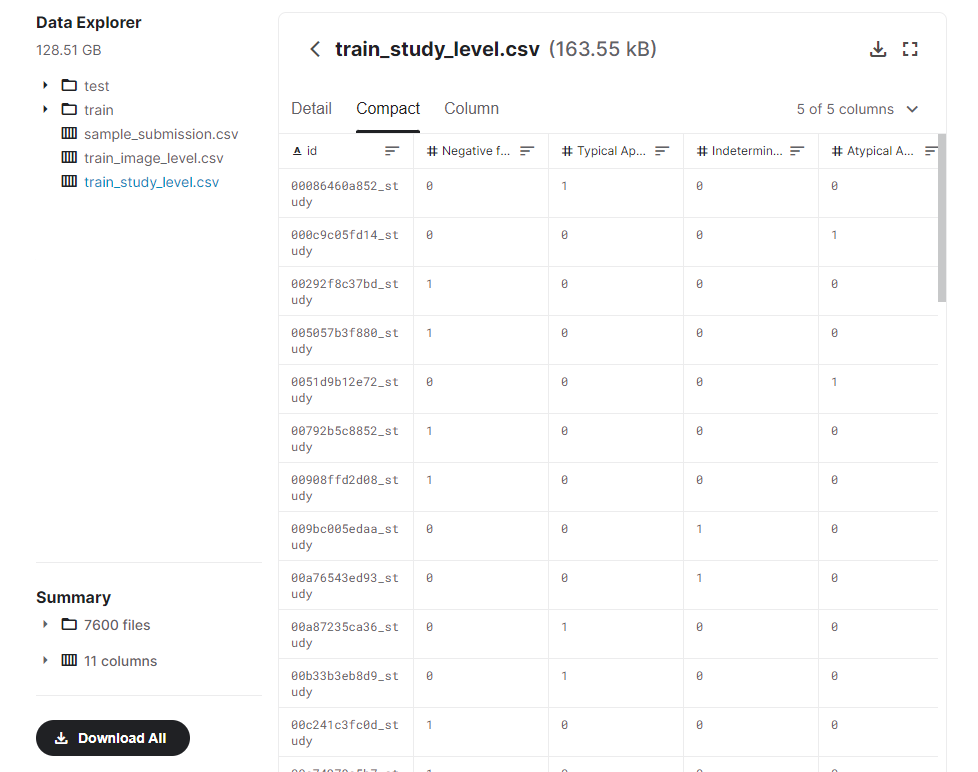
* Four columns
  + Id :- 6334 unique values
  + boxes :- list(json object)
    - keys :- {x,y, width, height}
  + label :- opacity
  + StudyInstanceUID :- 6054 unique values



##### 

##### - Details of train\_study\_level.csv

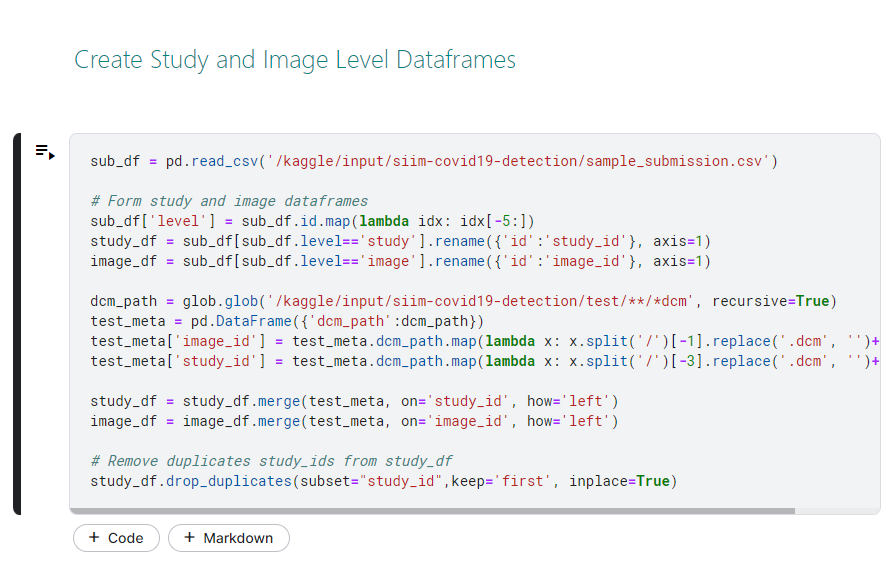
* Five columns
  + Id :- 6054 unique values
  + **Negative for Pneumonia :- boolen type**
  + **Typical Appearance :- boolen type**
  + **Indeterminate Appearance :- boolen type**
  + **Atypical Appearance :- boolen type**



**METHOD OF SOLUTION**

Our data consists of images + .csv files, containing custom information for each radiography.

So first started with import necessary library and create Data frames



Made a sub\_df variable where sample\_submission.csv is read by help of pandas .read\_csv method.

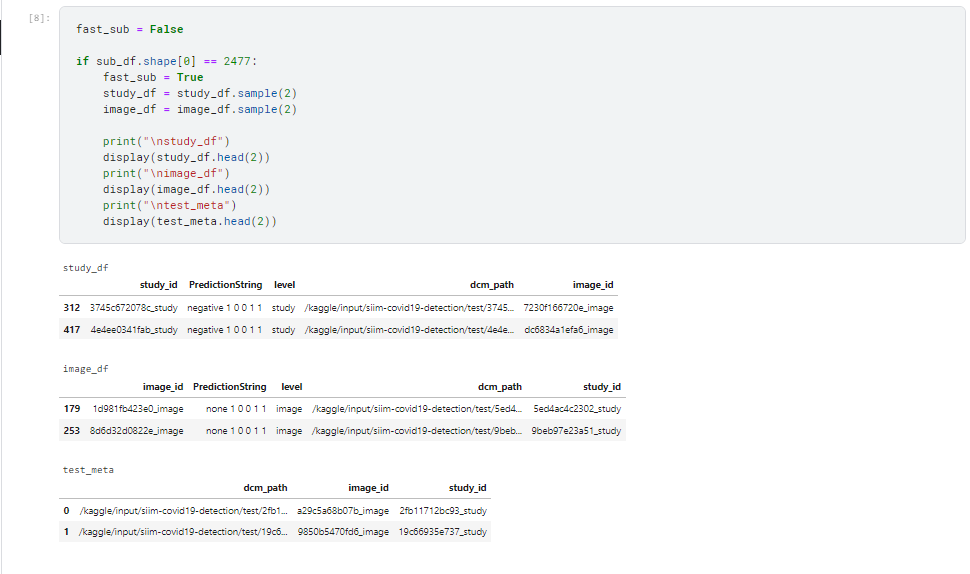
**- what is pandas ?**

Pandas is a software program library written for the Python programming language for information manipulation and evaluation. In particular, it offers information structures and operations for manipulating numerical tables and time series.

After made a form and Removed duplicates study\_ids from study\_df.

**-** Fast or Full Predictions

In case of non-competetion submission commits, we run the notebook with just two images each for image level and study level inference from the public test data.



- What is TensorFlow ?

An open source **artificial intelligence** library for fast numerical computing created and released by means of Google.It is a foundation library that can be used to create Deep Learning fashions directly or by way of using wrapper libraries that simplify the process constructed on top of TensorFlow.

TensorFlow is using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers.

Mainly use :-

Classification, Perception, Understanding, Discovering, Prediction

Creation



Custom Wrapper for Loading TFHub Model trained in TPU

Since the EffNetV2 Classifier models were trained on a TPU with the tfhub.KerasLayer formed with the handle argument as a GCS path, while loading the saved model for inference, the method tries to download the pre-trained weights from the definition of the layer from training i.e a GCS path.

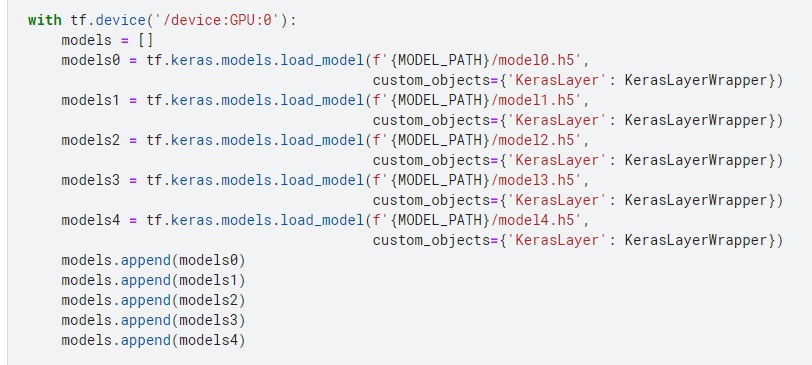
Since, inference notebooks don't have GCS and internet access, it is not possible to load the model without the pretrained weights explicitly loaded from the local directory.

If the models were trained on a GPU, we can use the cache location method to load the pre-trained weights by storing them in a cache folder with the hashed key of the model location, as the folder name. I tried this method here but, it doesn't seem to work as the model was trained with a GCS path defined in the tfhub.KerasLayer and the method kept on hitting the GCS path rather than loading the weights from the cache location. The only solution was to create a wrapper class to correct the handle argument to load the right pretrained weights explicitly from the local directory.

Predict Study Level

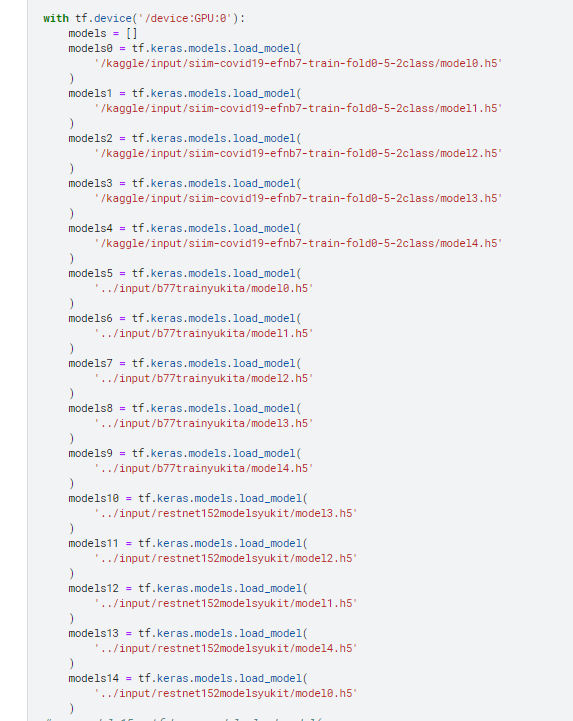
There are three functions build\_decoder, build\_augmenter, build\_dataset for returns decode, augment, dataset.

from TensorFlow used .device method add 4 models in models list

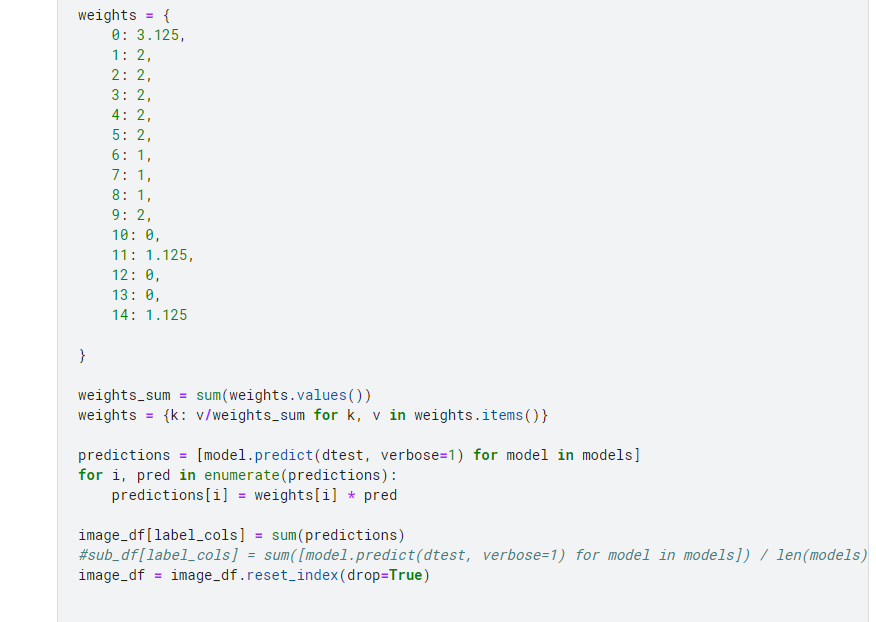


Predict 2Class Image Level

from TensorFlow used .device method add 4 models in models list



Made weights dictionary



For get good results changes weights dictionary.

Predict Image Level

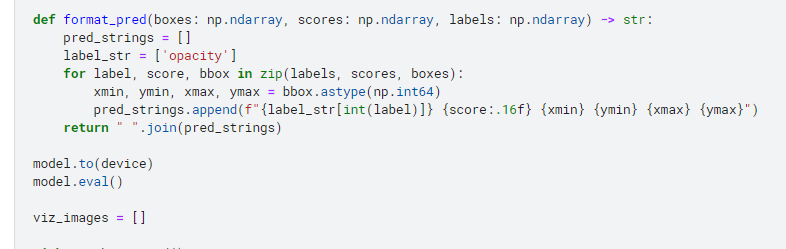


Import cv2 and matplotlib.pyplot to made functions plot\_img, plot\_imgs, draw\_bbox, draw\_bbox\_small

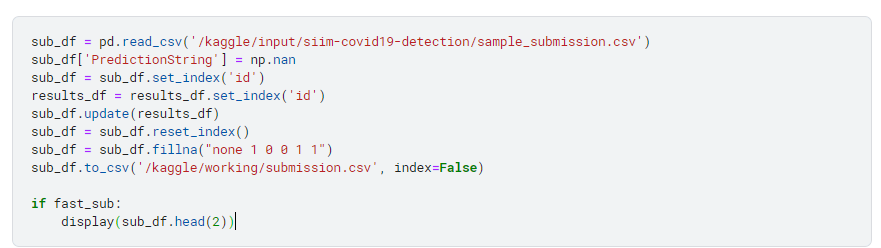
Added config to Config.fromfile(baseline\_cfg\_path)



import weighted\_boxes\_fusion, nms to made format\_pred

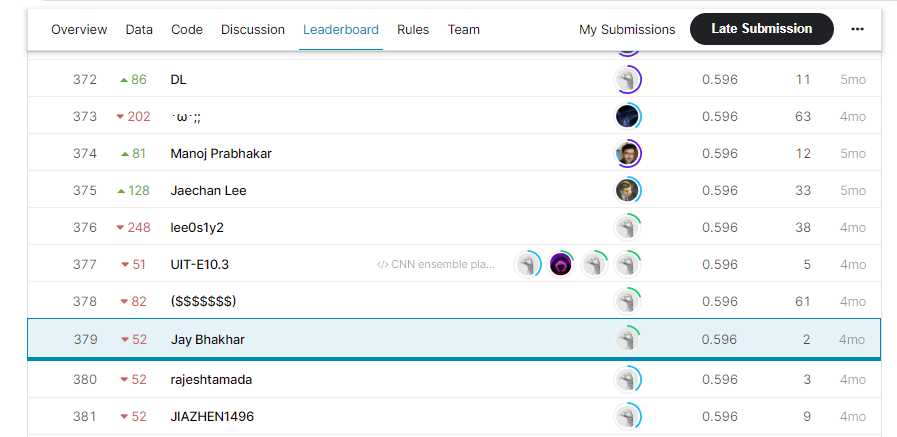


create submission.csv and submit



Leaderboard

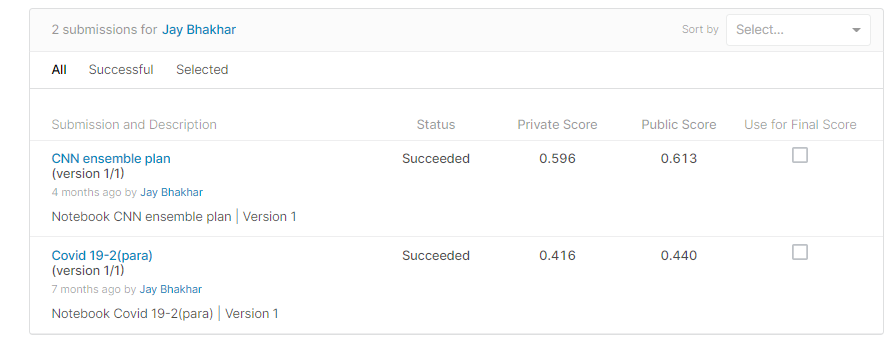
<https://www.kaggle.com/c/siim-covid19-detection/leaderboard>



final notebook:

<https://www.kaggle.com/jaybhakhar/cnn-ensemble-plan/edit/run/70536607>

Submissions :-



**References**

<https://www.tensorflow.org/tutorials>

<https://www.tensorflow.org/hub/api_docs/python/hub/KerasLayer>

https://pandas.pydata.org/docs/