```
1   # This Python 3 environment comes with many helpful analytics libraries installed
2   # It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
3   # For example, here's several helpful packages to load
4
5   import numpy as np # linear algebra
6   import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
7
8   # Input data files are available in the read-only "../input/" directory
9   # For example, running this (by clicking run or pressing Shift+Enter) will list all files und
10
11  import os
12  '''
13  for dirname, _, filenames in os.walk('/kaggle/input'):
14      for filename in filenames:
15          print(os.path.join(dirname, filename))
16  #'''
17
18  # You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as
19  # You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the
```

```
"\nfor dirname, _, filenames in os.walk('/kaggle/input'):\n    for filename in filenames:\n        print(o.
```

```
1
2   import numpy as np
3   import pandas as pd
4   import pickle
5   import numpy as np
6   import random
7   import time
8   import os
9   #os.environ["OPENCV_IO_MAX_IMAGE_PIXELS"] = pow(2,40).__str__()
10  import cv2
11  from tqdm import tqdm
```

● ×

```
19   from tensorflow.keras.models import Model, load_model
20   from tensorflow.keras.initializers import glorot_uniform
21   from tensorflow.keras.utils import plot_model
22   from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint, Lea
23   from IPython.display import display
24   from tensorflow.keras import backend as K
25   import matplotlib.pyplot as plt
26   import matplotlib.image as mpimg
27   from sklearn.model_selection import train_test_split
28   from keras import optimizers
29   #from sklearn.metrics import classification_report, confusion_matrix
30   import sklearn
31   import seaborn as sn
32   from keras.callbacks import CSVLogger, LambdaCallback
33   from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
1   train_file = "../input/covidxct/train_COVIDx_CT-2A.txt"
2   train_df_original = pd.read_csv(train_file, delimiter = " ", header=None)
3   train_df_original.columns = ['filename', 'class', 'xmin', 'ymin', 'xmax', 'ymax']
4   train_df_original.head()
```

	filename	class	xmin	ymin	xmax	ymax
0	NCP_96_1328_0032.png	2	9	94	512	405
1	NCP_96_1328_0035.png	2	10	106	512	405
2	NCP_96_1328_0036.png	2	10	105	512	406

```
 7         #train_df['class'][i] = set(train_df['class'][i])
 8
 9    print(f"train_df['filename'][0] = {train_df['filename'][0]}")
10    train_df.tail()
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame


See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm


/opt/conda/lib/python3.7/site-packages/IPython/core/interactiveshell.py:3437: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame


See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
  exec(code_obj, self.user_global_ns, self.user_ns)
train_df['filename'][0] = ../input/covidxct/2A_images/NCP_96_1328_0032.png
```

	filename	class
143773	../input/covidxct/2A_images/HUST-Patient97-028...	2
143774	../input/covidxct/2A_images/HUST-Patient97-028...	2
143775	../input/covidxct/2A_images/HUST-Patient97-029...	2
143776	../input/covidxct/2A_images/HUST-Patient97-029...	2
143777	../input/covidxct/2A_images/HUST-Patient97-029...	2

0	NCP_1013_2577_0000.png	2	0	170	512	404
1	NCP_1013_2577_0001.png	2	0	170	512	405
2	NCP_1013_2577_0002.png	2	0	166	508	408
3	NCP_1013_2577_0003.png	2	0	163	507	408
4	NCP_1013_2577_0004.png	2	0	159	507	409

```
1  val_df = val_df_original[['filename', 'class']]
2
3  for i in range(val_df.shape[0]):
4      filename = val_df['filename'][i]
5      file_full_path = f"../input/covidxct/2A_images/{filename}"
6      val_df['filename'][i] = file_full_path
7      #val_df['class'][i] = str(val_df['class'][i])
8
9  print(f"val_df['filename'][0] = {val_df['filename'][0]}")
```

—   —

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
  """"Entry point for launching an IPython kernel.
```

```
1  test_file = "../input/covidxct/test_COVIDx_CT-2A.txt"
2  test_df_original = pd.read_csv(test_file, delimiter = " ", header=None)
3  test_df_original.columns = ['filename', 'class', 'xmin', 'ymin', 'xmax', 'ymax']
4  test_df_original.head()
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
  exec(code_obj, self.user_global_ns, self.user_ns)
test_df['filename'][0] = ../input/covidxct/2A_images/NCP_341_1834_0022.png
```

	filename	class
25653	../input/covidxct/2A_images/radiopaedia_29_864...	2

```
1  train_generator = train_datagen.flow_from_dataframe(
```