Prediction based on CT S | Covid-19 Disease Diagno | Validate Anaconda Packa | Desktop/CAPSTONE PRO | Untitled1 - Jupyter Noteb

localhost:8888/notebooks/Desktop/CAPSTONE%20PROJECT/Predicting%20Covid/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 7 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run Code

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import os
from tqdm import tqdm
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
```

In [2]:
```
disease_types=['COVID', 'non-COVID']
data_dir = r'C:\Users\asmaf\Desktop\covid_prediction_input'
train_dir = os.path.join(data_dir)
```

In [3]:
```
train_dir
```

Out[3]:
```
'C:\\Users\\asmaf\\Desktop\\covid_prediction_input'
```

In [4]:
```
train_data = []
for defects_id, sp in enumerate(disease_types):
    for file in os.listdir(os.path.join(train_dir, sp)):
        train_data.append(['{}/{}'.format(sp, file), defects_id, sp])

train = pd.DataFrame(train_data, columns=['File', 'DiseaseID','Disease Type'])
train.head()
```

26°C Mostly clear | Search | ENG IN | 21:47 28-01-2023

Prediction based on CT S | Covid-19 Disease Diagno | Validate Anaconda Packa | Desktop/CAPSTONE PRO | Untitled1 - Jupyter Noteb

localhost:8888/notebooks/Desktop/CAPSTONE%20PROJECT/Predicting%20Covid/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 7 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run Code

Out[4]:

	File	DiseaseID	Disease Type
0	COVID/Covid (1).png	0	COVID
1	COVID/Covid (10).png	0	COVID
2	COVID/Covid (100).png	0	COVID
3	COVID/Covid (1000).png	0	COVID
4	COVID/Covid (1001).png	0	COVID

In [5]:
```
#this seed value to generate random numbers in the program
SEED = 42
train = train.sample(frac=1, random_state=SEED)
# To Reset indices
train.index = np.arange(len(train))
train.head()
```

Out[5]:

	File	DiseaseID	Disease Type
0	COVID/Covid (1227).png	0	COVID
1	COVID/Covid (431).png	0	COVID
2	non-COVID/Non-Covid (322).png	1	non-COVID
3	non-COVID/Non-Covid (379).png	1	non-COVID
4	COVID/Covid (61).png	0	COVID

26°C Mostly clear | Search | ENG IN | 21:47 28-01-2023

Prediction based on CT Sc | Covid-19 Disease Diagnos | Validate Anaconda Packa | Desktop/CAPSTONE PRO | Untitled1 - Jupyter Noteb

localhost:8888/notebooks/Desktop/CAPSTONE%20PROJECT/Predicting%20Covid/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 7 hours ago (autosaved)

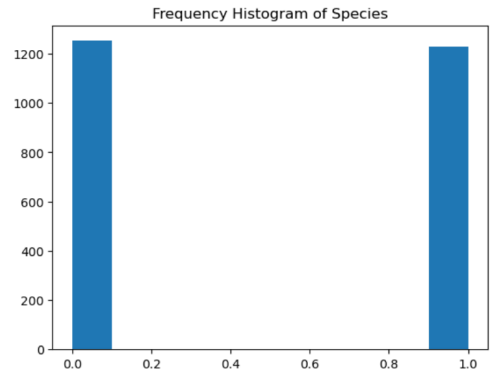
Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [6]:
```
plt.hist(train['DiseaseID'])
plt.title('Frequency Histogram of Species')
plt.figure(figsize=(8, 8))
plt.show()
```



26°C Mostly clear | Search | ENG IN | 21:47 28-01-2023

Prediction based on CT Sc | Covid-19 Disease Diagnos | Validate Anaconda Packa | Desktop/CAPSTONE PRO | Untitled1 - Jupyter Noteb

localhost:8888/notebooks/Desktop/CAPSTONE%20PROJECT/Predicting%20Covid/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 7 hours ago (autosaved)

Logout

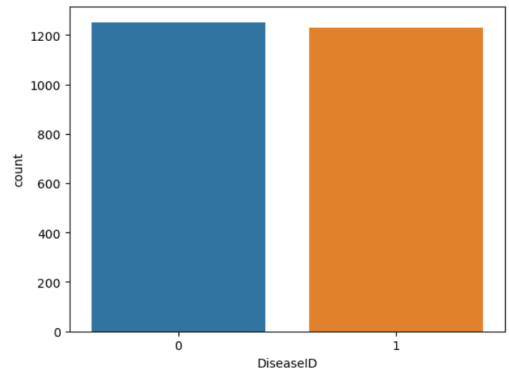
File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [7]:
```
import seaborn as sns
sns.countplot(x ='DiseaseID', data = train)
```

Out[7]: <AxesSubplot: xlabel='DiseaseID', ylabel='count'>



In [8]:

26°C Mostly clear | Search | ENG IN | 21:48 28-01-2023

Prediction based on CT Sc | Covid-19 Disease Diagnos | Validate Anaconda Packa | Desktop/CAPSTONE PRO | Untitled1 - Jupyter Noteb

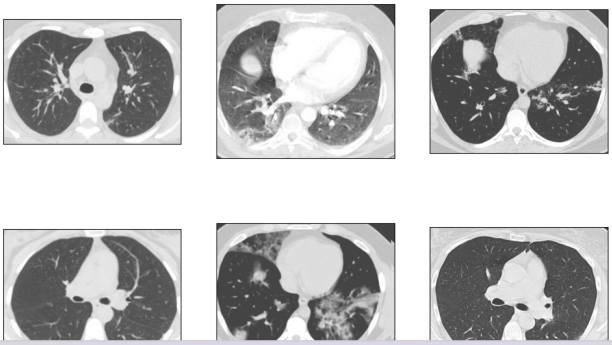
localhost:8888/notebooks/Desktop/CAPSTONE%20PROJECT/Predicting%20Covid/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 7 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

In [9]:
```
import cv2
def plot_defects(defect_types, rows, cols):
    fig, a = plt.subplots(rows, cols, figsize=(12, 12))
    defect_files = train['File'][train['Disease Type'] == defect_types].values
    n = 0
    for i in range(rows):
        for j in range(cols):
            image_path = os.path.join(data_dir, defect_files[n])
            a[i, j].set_xticks([])
            a[i, j].set_yticks([])
            a[i, j].imshow(cv2.imread(image_path))
            n += 1
# Displays first n images of class from training set
plot_defects('COVID', 3, 3)
```



26°C Mostly clear | Search | ENG IN | 21:48 28-01-2023

Prediction based on CT Sc | Covid-19 Disease Diagnos | Validate Anaconda Packa | Desktop/CAPSTONE PRO | Untitled1 - Jupyter Noteb

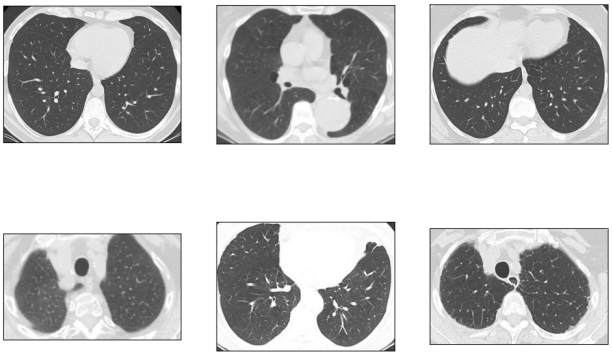
localhost:8888/notebooks/Desktop/CAPSTONE%20PROJECT/Predicting%20Covid/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 7 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

In [10]:
```
def plot_defects(defect_types, rows, cols):
    fig, a = plt.subplots(rows, cols, figsize=(12, 12))
    defect_files = train['File'][train['Disease Type'] == defect_types].values
    n = 0
    for i in range(rows):
        for j in range(cols):
            image_path = os.path.join(data_dir, defect_files[n])
            a[i, j].set_xticks([])
            a[i, j].set_yticks([])
            a[i, j].imshow(cv2.imread(image_path))
            n += 1
# Displays first n images of class from training set
plot_defects('non-COVID', 3, 3)
```



26°C Mostly clear | Search | ENG IN | 21:49 28-01-2023

Prediction based on CT Sc | Covid-19 Disease Diagnos | Validate Anaconda Packa | Desktop/CAPSTONE PRO | Untitled1 - Jupyter Noteb

localhost:8888/notebooks/Desktop/CAPSTONE%20PROJECT/Predicting%20Covid/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 7 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help — Not Trusted — Python 3 (ipykernel)

In [11]:
```
IMAGE_SIZE = 64
def read_image(filepath):
    return cv2.imread(os.path.join(data_dir, filepath))
# Loading a color image is the default flag

# Resize image to target size
def resize_image(image, image_size):
    return cv2.resize(image.copy(), image_size, interpolation=cv2.INTER_AREA)
```

In [12]:
```
X_train = np.zeros((train.shape[0], IMAGE_SIZE, IMAGE_SIZE, 3))
for i, file in tqdm(enumerate(train['File'].values)):
    image = read_image(file)
    if image is not None:
        X_train[i] = resize_image(image, (IMAGE_SIZE, IMAGE_SIZE))

# Normalize the data
X_Train = X_train / 255.
print('Train Shape: {}'.format(X_Train.shape))
```

```
2481it [00:11, 216.08it/s]

Train Shape: (2481, 64, 64, 3)
```

In [16]:
```
from keras.utils.np_utils import to_categorical
from tensorflow.keras.layers import Input
from keras.models import Model,Sequential, load_model
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization, AveragePooling2D, GlobalAveragePooling2D
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input, decode_predictions
```

In [17]:
```
Y_train = train['DiseaseID'].values
Y_train = to_categorical(Y_train, num_classes=2)
print(Y_train.shape)
```

```
(2481, 2)
```

26°C Mostly clear — Search — ENG IN — 21:49 28-01-2023

Prediction based on CT Sc | Covid-19 Disease Diagnos | Validate Anaconda Packa | Desktop/CAPSTONE PRO | Untitled1 - Jupyter Noteb

localhost:8888/notebooks/Desktop/CAPSTONE%20PROJECT/Predicting%20Covid/Untitled1.ipynb

jupyter Untitled1 Last Checkpoint: 7 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help — Not Trusted — Python 3 (ipykernel)

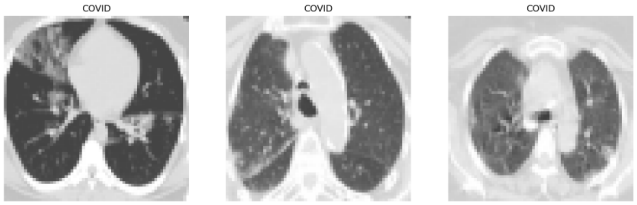
```
(2481, 2)
```

In [18]:
```
#the number of samples used in one forward/backward pass
BATCH_SIZE = 64

# Split the train and validation sets
X_train, X_val, Y_train, Y_val = train_test_split(X_Train, Y_train, test_size=0.2, random_state=SEED)
```

In [19]:
```
print(f'X_train:',X_train.shape)
print(f'X_val:',X_val.shape)
print(f'Y_train:',Y_train.shape)
print(f'Y_val:',Y_val.shape)
```

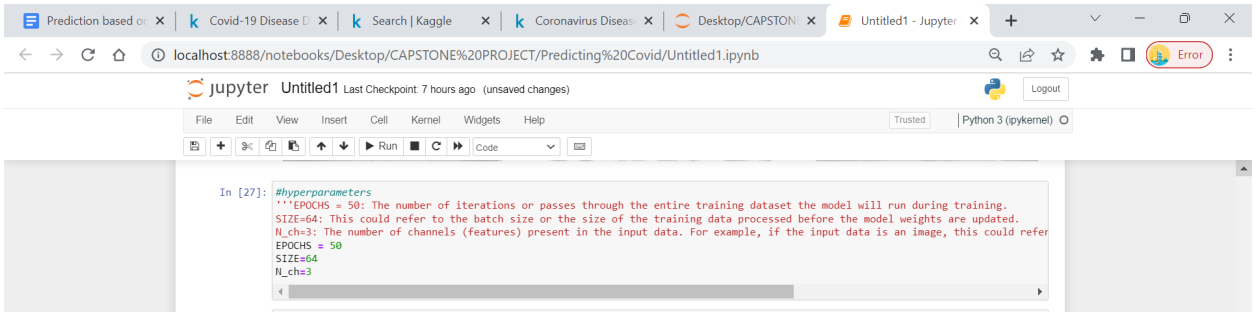
```
X_train: (1984, 64, 64, 3)
X_val: (497, 64, 64, 3)
Y_train: (1984, 2)
Y_val: (497, 2)
```

In [20]:
```
fig, ax = plt.subplots(1, 3, figsize=(15, 15))
for i in range(3):
    ax[i].set_axis_off()
    ax[i].imshow(X_train[i])
    ax[i].set_title(disease_types[np.argmax(Y_train[i])])
```



In [21]:
```
#hyperparameters
```

26°C Mostly clear — Search — ENG IN — 21:51 28-01-2023



In [28]:
```
model = build_resnet50()
annealer = ReduceLROnPlateau(monitor='val_accuracy', factor=0.70, patience=5, verbose=1, min_lr=1e-4)
checkpoint = ModelCheckpoint('model.h5', verbose=1, save_best_only=True)
# Generates batches of image data with data augmentation
datagen = ImageDataGenerator(rotation_range=360, # Degree range for random rotations
                        width_shift_range=0.2, # Range for random horizontal shifts
                        height_shift_range=0.2, # Range for random vertical shifts
                        zoom_range=0.2, # Range for random zoom
                        horizontal_flip=True, # Randomly flip inputs horizontally
                        vertical_flip=True) # Randomly flip inputs vertically

datagen.fit(X_train)
```

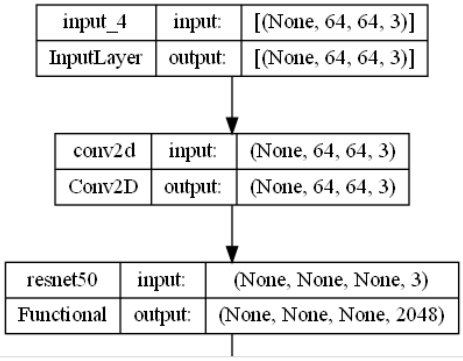
```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 64, 64, 3)]       0

 conv2d (Conv2D)             (None, 64, 64, 3)         84

 resnet50 (Functional)       (None, None, None, 2048)  23587712

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 batch_normalization (BatchN  (None, 2048)             8192
 ormalization)

 dropout (Dropout)           (None, 2048)              0

 dense (Dense)               (None, 256)               524544

 batch_normalization_1 (Batc  (None, 256)              1024
 hNormalization)

 dropout_1 (Dropout)         (None, 256)               0
```

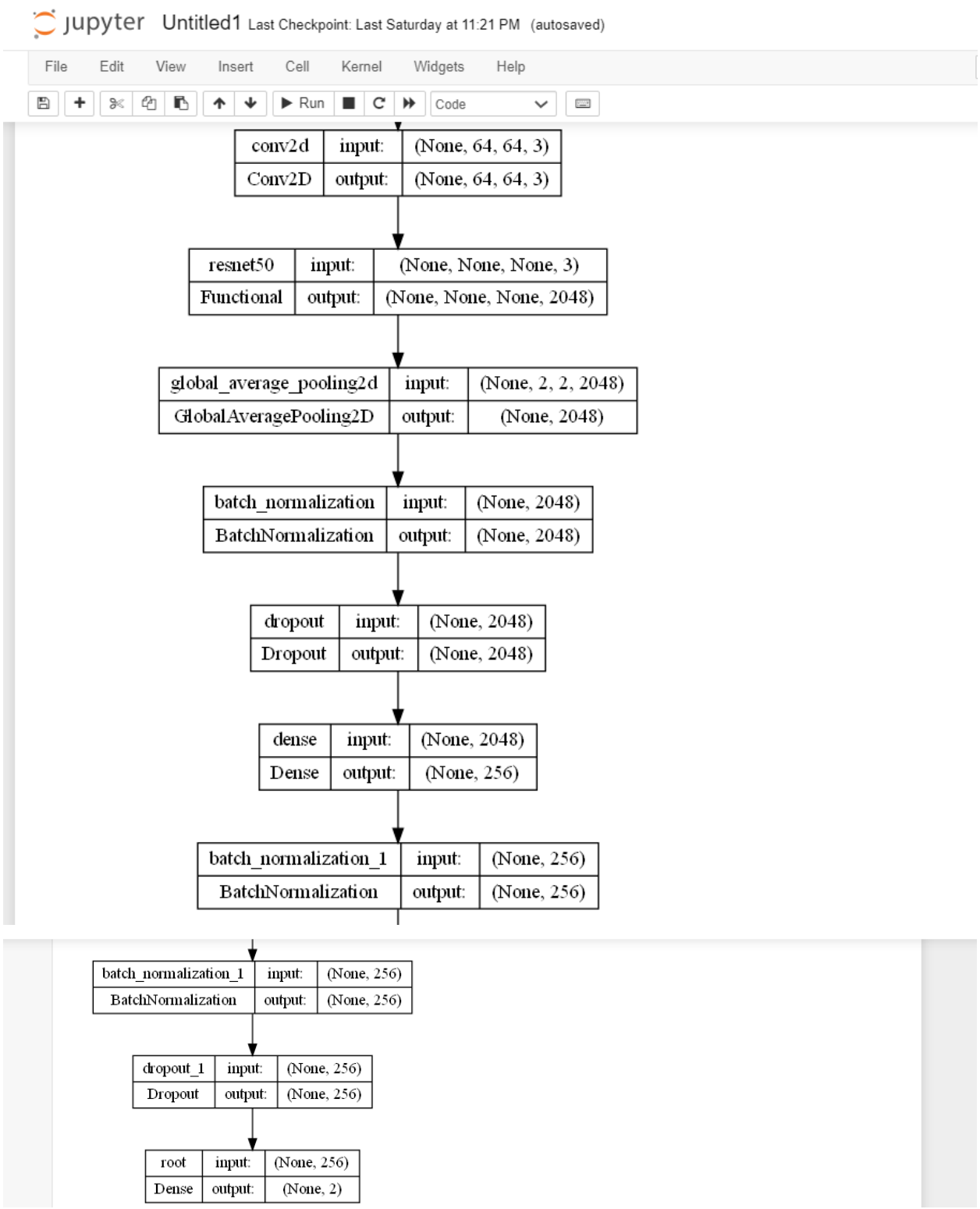
```
dropout_3 (Dropout)          (None, 256)               0

root (Dense)                 (None, 2)                 514

=================================================================
Total params: 24,122,070
Trainable params: 24,064,342
Non-trainable params: 57,728
_________________________________________________________________
```

In [ ]:
```
pip install pydot
```

In [29]:
```
from tensorflow.keras.utils import plot_model
from IPython.display import Image
plot_model(model, to_file='convnet.png', show_shapes=True,show_layer_names=True)
Image(filename='convnet.png')
```

Out[29]:




jupyter Untitled1 Last Checkpoint: Last Saturday at 11:21 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run Code
conv2d | input: | (None, 64, 64, 3)
Conv2D | output: | (None, 64, 64, 3)
resnet50 | input: | (None, None, None, 3)
Functional | output: | (None, None, None, 2048)
global_average_pooling2d | input: | (None, 2, 2, 2048)
GlobalAveragePooling2D | output: | (None, 2048)
batch_normalization | input: | (None, 2048)
BatchNormalization | output: | (None, 2048)
dropout | input: | (None, 2048)
Dropout | output: | (None, 2048)
dense | input: | (None, 2048)
Dense | output: | (None, 256)
batch_normalization_1 | input: | (None, 256)
BatchNormalization | output: | (None, 256)
batch_normalization_1 | input: | (None, 256)
BatchNormalization | output: | (None, 256)
dropout_1 | input: | (None, 256)
Dropout | output: | (None, 256)
root | input: | (None, 256)
Dense | output: | (None, 2)

```
hist = model.fit(datagen.flow(X_train, Y_train, batch_size=BATCH_SIZE),
                 steps_per_epoch=X_train.shape[0] // BATCH_SIZE,
                 epochs=EPOCHS,
                 verbose=1,
                 callbacks=[annealer, checkpoint],
                 validation_data=(X_val, Y_val))
```

```
Epoch 1/40
31/31 [==============================] - ETA: 0s - loss: 1.2102 - accuracy: 0.5509
Epoch 1: val_loss improved from inf to 5.51973, saving model to model.h5
31/31 [==============================] - 182s 5s/step - loss: 1.2102 - accuracy: 0.5509 - val_loss: 5.5197 - val_accuracy: 0.44
87 - lr: 0.0030
Epoch 2/40
31/31 [==============================] - ETA: 0s - loss: 0.9709 - accuracy: 0.6114
Epoch 2: val_loss did not improve from 5.51973
31/31 [==============================] - 170s 6s/step - loss: 0.9709 - accuracy: 0.6114 - val_loss: 7.6335 - val_accuracy: 0.44
87 - lr: 0.0030
Epoch 3/40
31/31 [==============================] - ETA: 0s - loss: 0.8747 - accuracy: 0.6436
Epoch 3: val_loss did not improve from 5.51973
31/31 [==============================] - 164s 5s/step - loss: 0.8747 - accuracy: 0.6436 - val_loss: 17.3597 - val_accuracy: 0.4
487 - lr: 0.0030
Epoch 4/40
31/31 [==============================] - ETA: 0s - loss: 0.7929 - accuracy: 0.6653
Epoch 4: val_loss did not improve from 5.51973
31/31 [==============================] - 101s 3s/step - loss: 0.7929 - accuracy: 0.6653 - val_loss: 6.5023 - val_accuracy: 0.55
13 - lr: 0.0030
Epoch 5/40
31/31 [==============================] - ETA: 0s - loss: 0.7041 - accuracy: 0.7031
Epoch 5: val_loss did not improve from 5.51973
31/31 [==============================] - 107s 3s/step - loss: 0.7041 - accuracy: 0.7031 - val_loss: 19.5301 - val_accuracy: 0.5
513 - lr: 0.0030
Epoch 6/40
31/31 [==============================] - ETA: 0s - loss: 0.6095 - accuracy: 0.7379
Epoch 6: val_loss improved from 5.51973 to 4.26177, saving model to model.h5
31/31 [==============================] - 102s 3s/step - loss: 0.6095 - accuracy: 0.7379 - val_loss: 4.2618 - val_accuracy: 0.55
13 - lr: 0.0030
Epoch 7/40
31/31 [==============================] - ETA: 0s - loss: 0.5701 - accuracy: 0.7666
Epoch 7: val_loss improved from 4.26177 to 1.37565, saving model to model.h5
31/31 [==============================] - 115s 4s/step - loss: 0.5701 - accuracy: 0.7666 - val_loss: 1.3756 - val_accuracy: 0.55
13 - lr: 0.0030
Epoch 8/40
31/31 [==============================] - ETA: 0s - loss: 0.5352 - accuracy: 0.7671
Epoch 8: val_loss did not improve from 1.37565
31/31 [==============================] - 101s 3s/step - loss: 0.5352 - accuracy: 0.7671 - val_loss: 7.9309 - val_accuracy: 0.44
87 - lr: 0.0030
```

```
                epochs=EPOCHS,
                verbose=1,
                callbacks=[annealer, checkpoint],
                validation_data=(X_val, Y_val))
```

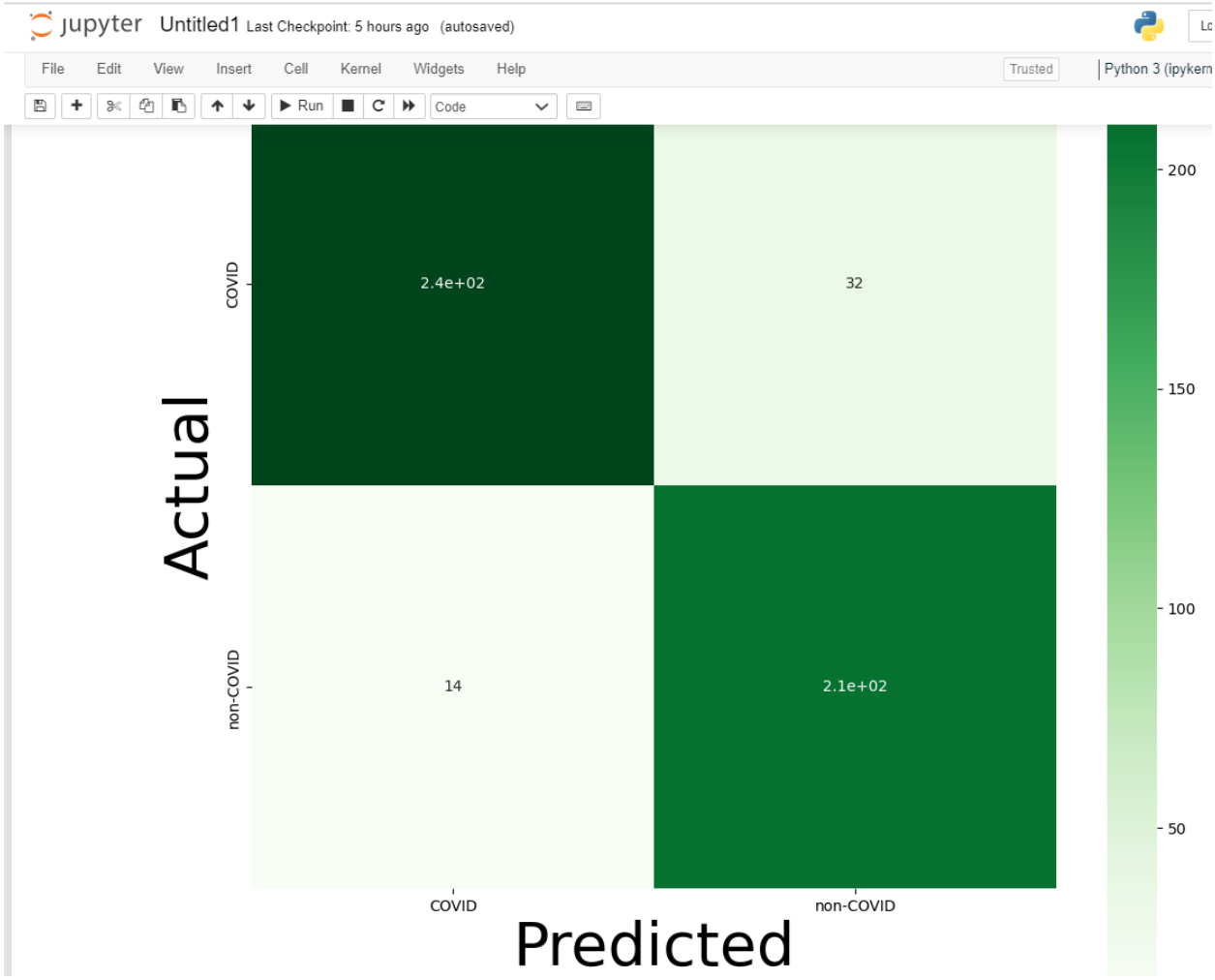
```
31/31 [==============================] - 96s 3s/step - loss: 0.2581 - accuracy: 0.8916 - val_loss: 0.2176 - val_accuracy: 0.
9095 - lr: 3.5295e-04
Epoch 38/40
31/31 [==============================] - ETA: 0s - loss: 0.2679 - accuracy: 0.8992
Epoch 38: ReduceLROnPlateau reducing learning rate to 0.0002470628678565845.

Epoch 38: val_loss did not improve from 0.21326
31/31 [==============================] - 96s 3s/step - loss: 0.2679 - accuracy: 0.8992 - val_loss: 0.2225 - val_accuracy: 0.
8994 - lr: 3.5295e-04
Epoch 39/40
31/31 [==============================] - ETA: 0s - loss: 0.2697 - accuracy: 0.8952
Epoch 39: val_loss did not improve from 0.21326
31/31 [==============================] - 92s 3s/step - loss: 0.2697 - accuracy: 0.8952 - val_loss: 0.2218 - val_accuracy: 0.
8974 - lr: 2.4706e-04
Epoch 40/40
31/31 [==============================] - ETA: 0s - loss: 0.2439 - accuracy: 0.9017
Epoch 40: val_loss did not improve from 0.21326
31/31 [==============================] - 98s 3s/step - loss: 0.2439 - accuracy: 0.9017 - val_loss: 0.2183 - val_accuracy: 0.
9074 - lr: 2.4706e-04
```

In [41]:
```
#model = load_model('../output/kaggle/working/model.h5')
final_loss, final_accuracy = model.evaluate(X_val, Y_val)
print('Final Loss: {}, Final Accuracy: {}'.format(final_loss, final_accuracy))
```

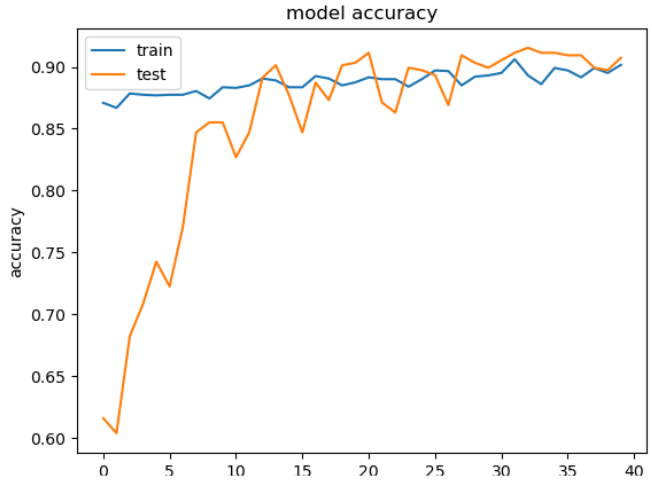
```
16/16 [==============================] - 3s 183ms/step - loss: 0.2183 - accuracy: 0.9074
Final Loss: 0.21825508773326874, Final Accuracy: 0.9074446558952332
```

In [42]:
```
Y_pred = model.predict(X_val)

Y_pred = np.argmax(Y_pred, axis=1)
Y_true = np.argmax(Y_val, axis=1)

cm = confusion_matrix(Y_true, Y_pred)
plt.figure(figsize=(12, 12))
ax = sns.heatmap(cm, cmap=plt.cm.Greens, annot=True, square=True, xticklabels=disease_types, yticklabels=disease_types)
ax.set_ylabel('Actual', fontsize=40)
ax.set_xlabel('Predicted', fontsize=40)
```

```
16/16 [==============================] - 2s 157ms/step
```


jupyter Untitled1 Last Checkpoint: 5 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykern
Run
Code
Actual
COVID
non-COVID
2.4e+02
32
14
2.1e+02
COVID
non-COVID
Predicted
200
150
100
50

In [30]:
```
# accuracy plot
plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# loss plot
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



In [31]:
```
# accuracy plot
plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# loss plot
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```