

=====> Capstone Project <=====

Problem Statement

CT Scan Images Classification

Data Overview

This dataset contains 1252 Ct scans that are positive for SARS-CoV-2 infection (COVID-19) and 1230 CT scans for patients no-infected by SARS-CoV-2 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 intelligence 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 Complete This Capstone Project

- 1- Download the Data from this Link. https://drive.google.com/drive/folders/1WOeodRmv1Mw5Cswujp3nUli6ViQWKpo_?usp=sharing
 - 2- the images are in different sizes so you have to take a fixed size on which you have to work.
 - 3- Do data augmentation on it, mention at least 5 args inside it.
 - 4- Train Model on it you only have to use resnet from resnet you can pick any layer model like - ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110, ResNet-152, ResNet-164, ResNet-1202.
- And mention early stopping and modelcheckpoint while training.
- 5- Do Prediction and mention multiple performance metrics.

1. Import the Required Libraries

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#%matplotlib inline
import seaborn as sns
import cv2
import os
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from keras.models import Model
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import Dense, Conv2D, BatchNormalization, GlobalAveragePooling2D
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.utils import plot_model

# Suppress info, warnings and error messages
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

print("All Libraries are Imported Successfully")
```

All Libraries are Imported Successfully

2. Load the Dataset

```
In [3]: disease_types = ['Covid', 'Non-Covid']

train_dir = data_dir = '/content/drive/MyDrive/Colab Notebooks/Capstone Project/Dataset'
train_data = []

for index, sp in enumerate(disease_types):
    for file in os.listdir(os.path.join(train_dir, sp)):
        train_data.append([sp + "/" + file, index, sp])

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

Out[3]:

	File	ID	Disease Type
0	Covid/Covid (726).png	0	Covid
1	Covid/Covid (810).png	0	Covid
2	Covid/Covid (727).png	0	Covid
3	Covid/Covid (757).png	0	Covid
4	Covid/Covid (742).png	0	Covid
...
2476	Non-Covid/Non-Covid (671).png	1	Non-Covid
2477	Non-Covid/Non-Covid (733).png	1	Non-Covid
2478	Non-Covid/Non-Covid (727).png	1	Non-Covid
2479	Non-Covid/Non-Covid (695).png	1	Non-Covid
2480	Non-Covid/Non-Covid (775).png	1	Non-Covid

2481 rows × 3 columns

Setting Up the Dataset and adding column for the Histogram

```
In [4]: Seed = 40

train = train.sample(frac = 1, replace=False, random_state = Seed)

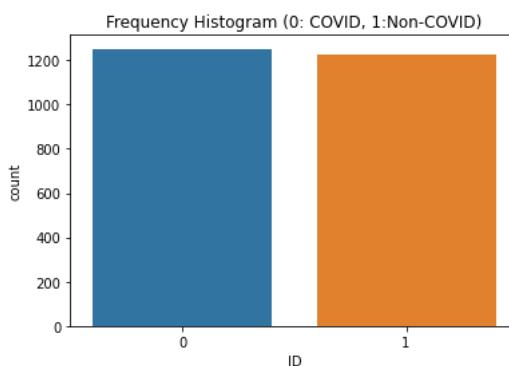
# Reset indices (row numbers)
train = train.reset_index(drop = True)

sns.countplot(x = "ID", data = train).set_title("Frequency Histogram (0: COVID, 1:Non-COVID)")
train
```

Out[4]:

	File	ID	Disease Type
0	Covid/Covid (727).png	0	Covid
1	Covid/Covid (277).png	0	Covid
2	Covid/Covid (29).png	0	Covid
3	Non-Covid/Non-Covid (540).png	1	Non-Covid
4	Covid/Covid (1203).png	0	Covid
...
2476	Non-Covid/Non-Covid (254).png	1	Non-Covid
2477	Non-Covid/Non-Covid (802).png	1	Non-Covid
2478	Non-Covid/Non-Covid (1052).png	1	Non-Covid
2479	Non-Covid/Non-Covid (1034).png	1	Non-Covid
2480	Non-Covid/Non-Covid (87).png	1	Non-Covid

2481 rows × 3 columns



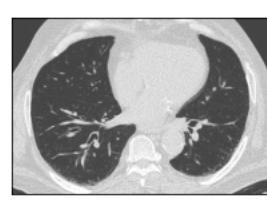
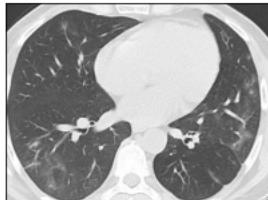
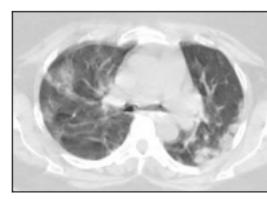
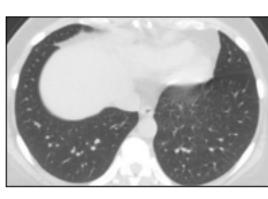
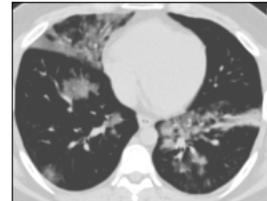
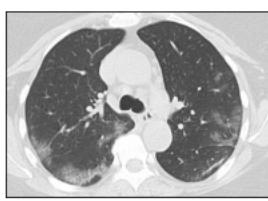
Observation :

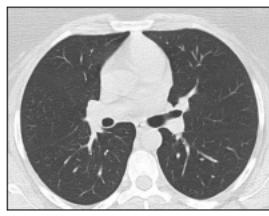
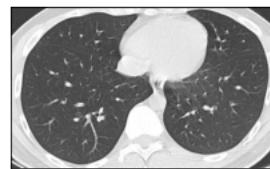
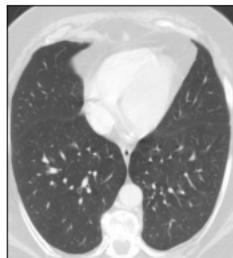
1. We have total 2481 Images in our dataset.
2. Positive are 1250 CT scan images.
3. Negatice are 1230 CT scan images.

Plotting the Images

```
In [5]: from collections.abc import ValuesView
def plot_defects(defect_types, rows, cols):
    fig, ax = plt.subplots(rows, cols, figsize=(12, 12))
    defect_files = train['File'][train['Disease Type'] == defect_types].values
    n = 0
    fig.suptitle(defect_types, fontsize = 22, color = "white")
    for i in range(rows):
        for j in range(cols):
            image_path = os.path.join(data_dir, defect_files[n])
            ax[i, j].set_xticks([])
            ax[i, j].set_yticks([])
            ax[i, j].imshow(cv2.imread(image_path))
            n = n+1

plot_defects('Covid', 3, 3)
plot_defects('Non-Covid', 3, 3)
```





3. Resizing the Images

```
In [6]: IMAGE_SIZE = 224

# OpenCV Function to Load colored image
def read_image(filepath):
    return cv2.imread(os.path.join(data_dir, filepath))

# OpenCV Function to resize an image
def resize_image(image, image_size):
    return cv2.resize(image.copy(), image_size, interpolation = cv2.INTER_AREA)
```

Observations :

Here I am resizing all the images to one size so that my model can train perfectly and do right predictions.

```
In [7]: X_train = np.zeros((train.shape[0], IMAGE_SIZE, IMAGE_SIZE, 3))

for i, file in enumerate(train['File'].values):
    image = read_image(file)
    if image is not None:
        X_train[i] = resize_image(image, (IMAGE_SIZE, IMAGE_SIZE))

X_Train = X_train / 255.0 # Pixel normalization
print('Train Shape:', X_Train.shape)

Y_train = to_categorical(train['ID'].values, num_classes = 2)

print(Y_train)

Train Shape: (2481, 224, 224, 3)
[[1. 0.]
 [1. 0.]
 [1. 0.]
 ...
 [0. 1.]
 [0. 1.]
 [0. 1.]]
```

4. Splitting the Dataset into Train and Test

```
In [8]: # Dataframe split to train and validation set (80% train and 20% validation)
X_train, X_val, Y_train, Y_val = train_test_split(X_Train,
                                                Y_train,
                                                test_size = 0.2, # Percent 20% of the data is using as test set
                                                random_state = Seed)

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, 224, 224, 3)
X_val: (497, 224, 224, 3)
Y_train: (1984, 2)
Y_val: (497, 2)
```

5. Building Model ResNet50

```
In [11]: from keras.layers import Flatten
# Architectural function for ResNet50
def build_ResNet50(IMAGE_SIZE, channels):

    resnet50 = ResNet50(input_shape = [224,224,3],weights = 'imagenet', include_top = False)
    x = Flatten()(resnet50.output)
    x = BatchNormalization()(x)
    x = Dense(64, activation = 'relu')(x)
    x = BatchNormalization()(x)

    output = Dense(2, activation = 'softmax')(x)

    # model
    model = Model(inputs = resnet50.input, outputs = output)

    optimizer = Adam(learning_rate = 0.003, beta_1 = 0.9, beta_2 = 0.999, epsilon = 0.1, decay = 0.0)
    model.compile(loss = 'categorical_crossentropy', # minimize the negative multinomial Log-Likelihood also known as the cross-
                  optimizer = optimizer,
                  metrics = ['accuracy'])
    model.summary()

    return model
```

```
In [ ]: # # Architectural function for DenseNet-169
# def build_ResNet50(IMAGE_SIZE, channels):

#     resnet50 = ResNet50(weights = 'imagenet', include_top = False)

#     input = Input(shape = (IMAGE_SIZE, IMAGE_SIZE, channels))
#     x = Conv2D(3, (3, 3), padding = 'same')(input) # When I am removing this Line it's showing error
#     x = resnet50(x)
#     x = GlobalAveragePooling2D()(x)
#     x = BatchNormalization()(x)
#     x = Dense(64, activation = 'relu')(x)
#     x = BatchNormalization()(x)

#     output = Dense(2, activation = 'softmax')(x)

#     # model
#     model = Model(input, output)

#     optimizer = Adam(Learning_rate = 0.003, beta_1 = 0.9, beta_2 = 0.999, epsilon = 0.1, decay = 0.0)
#     model.compile(loss = 'categorical_crossentropy', # minimize the negative multinomial Log-Likelihood also known as the cross
#                   optimizer = optimizer,
#                   metrics = ['accuracy'])
#     model.summary()

#     return model
```

Observations :

- 1- Here I am training my model from ResNet50.
- 2- I am using Imagenet for weights.
- 3- For my model input size is 224.
- 4- Conv2d layer, Global Average Pooling, Batch Normalization these parameters I am using here.
- 5- My loss will be the Categorical Crossentropy

Data Augmentation

```
In [13]: channels = 3

model = build_ResNet50(IMAGE_SIZE, channels)
annealer = ReduceLROnPlateau(monitor = 'val_accuracy', # Reduce Learning rate when Validation accuracy remains constant
                            factor = 0.70, # Rate by which the Learning rate will decrease
                            patience = 5, # number of epochs without improvement, after which the Learning rate will decrease
                            verbose = 1, # Display messages
                            min_lr = 1e-4 # Lower limit on the Learning rate.
                           )
checkpoint = ModelCheckpoint('model.h5', verbose = 1, save_best_only = True) # Save neural network weights

# 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)

plot_model(model, to_file = 'convnet.png', show_shapes = True, show_layer_names = True)
```

Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[None, 224, 224, 3 0)]		[]
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3) 0		['input_3[0][0]']
conv1_conv (Conv2D)	(None, 112, 112, 64 9472)		['conv1_pad[0][0]']
conv1_bn (BatchNormalization)	(None, 112, 112, 64 256)		['conv1_conv[0][0]']
conv1_relu (Activation)	(None, 112, 112, 64 0)		['conv1_bn[0][0]']
pool1_pad (ZeroPadding2D)	(None, 114, 114, 64 0)		['conv1_relu[0][0]']
pool1_pool (MaxPooling2D)	(None, 56, 56, 64) 0		['pool1_pad[0][0]']
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 64) 4160		['pool1_pool[0][0]']
conv2_block1_1_bn (BatchNormal ization)	(None, 56, 56, 64) 256		['conv2_block1_1_conv[0][0]']
conv2_block1_1_relu (Activatio n)	(None, 56, 56, 64) 0		['conv2_block1_1_bn[0][0]']
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 64) 36928		['conv2_block1_1_relu[0][0]']
conv2_block1_2_bn (BatchNormal ization)	(None, 56, 56, 64) 256		['conv2_block1_2_conv[0][0]']
conv2_block1_2_relu (Activatio n)	(None, 56, 56, 64) 0		['conv2_block1_2_bn[0][0]']
conv2_block1_0_conv (Conv2D)	(None, 56, 56, 256) 16640		['pool1_pool[0][0]']
conv2_block1_3_conv (Conv2D)	(None, 56, 56, 256) 16640		['conv2_block1_2_relu[0][0]']
conv2_block1_0_bn (BatchNormal ization)	(None, 56, 56, 256) 1024		['conv2_block1_0_conv[0][0]']
conv2_block1_3_bn (BatchNormal ization)	(None, 56, 56, 256) 1024		['conv2_block1_3_conv[0][0]']
conv2_block1_add (Add)	(None, 56, 56, 256) 0		['conv2_block1_0_bn[0][0]', 'conv2_block1_3_bn[0][0]']
conv2_block1_out (Activation)	(None, 56, 56, 256) 0		['conv2_block1_add[0][0]']
conv2_block2_1_conv (Conv2D)	(None, 56, 56, 64) 16448		['conv2_block1_out[0][0]']
conv2_block2_1_bn (BatchNormal ization)	(None, 56, 56, 64) 256		['conv2_block2_1_conv[0][0]']
conv2_block2_1_relu (Activatio n)	(None, 56, 56, 64) 0		['conv2_block2_1_bn[0][0]']
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 64) 36928		['conv2_block2_1_relu[0][0]']
conv2_block2_2_bn (BatchNormal ization)	(None, 56, 56, 64) 256		['conv2_block2_2_conv[0][0]']
conv2_block2_2_relu (Activatio n)	(None, 56, 56, 64) 0		['conv2_block2_2_bn[0][0]']
conv2_block2_3_conv (Conv2D)	(None, 56, 56, 256) 16640		['conv2_block2_2_relu[0][0]']
conv2_block2_3_bn (BatchNormal ization)	(None, 56, 56, 256) 1024		['conv2_block2_3_conv[0][0]']
conv2_block2_add (Add)	(None, 56, 56, 256) 0		['conv2_block1_out[0][0]', 'conv2_block2_3_bn[0][0]']
conv2_block2_out (Activation)	(None, 56, 56, 256) 0		['conv2_block2_add[0][0]']
conv2_block3_1_conv (Conv2D)	(None, 56, 56, 64) 16448		['conv2_block2_out[0][0]']
conv2_block3_1_bn (BatchNormal ization)	(None, 56, 56, 64) 256		['conv2_block3_1_conv[0][0]']
conv2_block3_1_relu (Activatio n)	(None, 56, 56, 64) 0		['conv2_block3_1_bn[0][0]']

```

n)

conv2_block3_2_conv (Conv2D)    (None, 56, 56, 64)  36928      ['conv2_block3_1_relu[0][0]']

conv2_block3_2_bn (BatchNormal (None, 56, 56, 64)  256      ['conv2_block3_2_conv[0][0]']

       ization)

conv2_block3_2_relu (Activatio (None, 56, 56, 64)  0      ['conv2_block3_2_bn[0][0]']

n)

conv2_block3_3_conv (Conv2D)    (None, 56, 56, 256) 16640      ['conv2_block3_2_relu[0][0]']

conv2_block3_3_bn (BatchNormal (None, 56, 56, 256) 1024      ['conv2_block3_3_conv[0][0]']

       ization)

conv2_block3_add (Add)         (None, 56, 56, 256)  0      ['conv2_block2_out[0][0]',

        'conv2_block3_3_bn[0][0]']

conv2_block3_out (Activation) (None, 56, 56, 256)  0      ['conv2_block3_add[0][0]']

conv3_block1_1_conv (Conv2D)   (None, 28, 28, 128) 32896      ['conv2_block3_out[0][0]']

conv3_block1_1_bn (BatchNormal (None, 28, 28, 128) 512      ['conv3_block1_1_conv[0][0]']

       ization)

conv3_block1_1_relu (Activatio (None, 28, 28, 128)  0      ['conv3_block1_1_bn[0][0]']

n)

conv3_block1_2_conv (Conv2D)   (None, 28, 28, 128) 147584     ['conv3_block1_1_relu[0][0]']

conv3_block1_2_bn (BatchNormal (None, 28, 28, 128) 512      ['conv3_block1_2_conv[0][0]']

       ization)

conv3_block1_2_relu (Activatio (None, 28, 28, 128)  0      ['conv3_block1_2_bn[0][0]']

n)

conv3_block1_0_conv (Conv2D)   (None, 28, 28, 512) 131584     ['conv2_block3_out[0][0]']

conv3_block1_3_conv (Conv2D)   (None, 28, 28, 512) 66048      ['conv3_block1_2_relu[0][0]']

conv3_block1_0_bn (BatchNormal (None, 28, 28, 512) 2048      ['conv3_block1_0_conv[0][0]']

       ization)

conv3_block1_3_bn (BatchNormal (None, 28, 28, 512) 2048      ['conv3_block1_3_conv[0][0]']

       ization)

conv3_block1_add (Add)         (None, 28, 28, 512)  0      ['conv3_block1_0_bn[0][0]',

        'conv3_block1_3_bn[0][0]']

conv3_block1_out (Activation) (None, 28, 28, 512)  0      ['conv3_block1_add[0][0]']

conv3_block2_1_conv (Conv2D)   (None, 28, 28, 128) 65664      ['conv3_block1_out[0][0]']

conv3_block2_1_bn (BatchNormal (None, 28, 28, 128) 512      ['conv3_block2_1_conv[0][0]']

       ization)

conv3_block2_1_relu (Activatio (None, 28, 28, 128)  0      ['conv3_block2_1_bn[0][0]']

n)

conv3_block2_2_conv (Conv2D)   (None, 28, 28, 128) 147584     ['conv3_block2_1_relu[0][0]']

conv3_block2_2_bn (BatchNormal (None, 28, 28, 128) 512      ['conv3_block2_2_conv[0][0]']

       ization)

conv3_block2_2_relu (Activatio (None, 28, 28, 128)  0      ['conv3_block2_2_bn[0][0]']

n)

conv3_block2_3_conv (Conv2D)   (None, 28, 28, 512) 66048      ['conv3_block2_2_relu[0][0]']

conv3_block2_3_bn (BatchNormal (None, 28, 28, 512) 2048      ['conv3_block2_3_conv[0][0]']

       ization)

conv3_block2_add (Add)         (None, 28, 28, 512)  0      ['conv3_block2_3_conv[0][0]',

        'conv3_block2_3_bn[0][0]']

conv3_block2_out (Activation) (None, 28, 28, 512)  0      ['conv3_block2_add[0][0]']

conv3_block3_1_conv (Conv2D)   (None, 28, 28, 128) 65664      ['conv3_block2_out[0][0]']

conv3_block3_1_bn (BatchNormal (None, 28, 28, 128) 512      ['conv3_block3_1_conv[0][0]']

       ization)

conv3_block3_1_relu (Activatio (None, 28, 28, 128)  0      ['conv3_block3_1_bn[0][0]']

n)

conv3_block3_2_conv (Conv2D)   (None, 28, 28, 128) 147584     ['conv3_block3_1_relu[0][0]']

```

conv3_block3_2_bn (BatchNormal (None, 28, 28, 128) 512 ization)	['conv3_block3_2_conv[0][0]']
conv3_block3_2_relu (Activatio (None, 28, 28, 128) 0 n)	['conv3_block3_2_bn[0][0]']
conv3_block3_3_conv (Conv2D) (None, 28, 28, 512) 66048	['conv3_block3_2_relu[0][0]']
conv3_block3_3_bn (BatchNormal (None, 28, 28, 512) 2048 ization)	['conv3_block3_3_conv[0][0]']
conv3_block3_add (Add) (None, 28, 28, 512) 0	['conv3_block2_out[0][0]', 'conv3_block3_3_bn[0][0]']
conv3_block3_out (Activation) (None, 28, 28, 512) 0	['conv3_block3_add[0][0]']
conv3_block4_1_conv (Conv2D) (None, 28, 28, 128) 65664	['conv3_block3_out[0][0]']
conv3_block4_1_bn (BatchNormal (None, 28, 28, 128) 512 ization)	['conv3_block4_1_conv[0][0]']
conv3_block4_1_relu (Activatio (None, 28, 28, 128) 0 n)	['conv3_block4_1_bn[0][0]']
conv3_block4_2_conv (Conv2D) (None, 28, 28, 128) 147584	['conv3_block4_1_relu[0][0]']
conv3_block4_2_bn (BatchNormal (None, 28, 28, 128) 512 ization)	['conv3_block4_2_conv[0][0]']
conv3_block4_2_relu (Activatio (None, 28, 28, 128) 0 n)	['conv3_block4_2_bn[0][0]']
conv3_block4_3_conv (Conv2D) (None, 28, 28, 512) 66048	['conv3_block4_2_relu[0][0]']
conv3_block4_3_bn (BatchNormal (None, 28, 28, 512) 2048 ization)	['conv3_block4_3_conv[0][0]']
conv3_block4_add (Add) (None, 28, 28, 512) 0	['conv3_block3_out[0][0]', 'conv3_block4_3_bn[0][0]']
conv3_block4_out (Activation) (None, 28, 28, 512) 0	['conv3_block4_add[0][0]']
conv4_block1_1_conv (Conv2D) (None, 14, 14, 256) 131328	['conv3_block4_out[0][0]']
conv4_block1_1_bn (BatchNormal (None, 14, 14, 256) 1024 ization)	['conv4_block1_1_conv[0][0]']
conv4_block1_1_relu (Activatio (None, 14, 14, 256) 0 n)	['conv4_block1_1_bn[0][0]']
conv4_block1_2_conv (Conv2D) (None, 14, 14, 256) 590080	['conv4_block1_1_relu[0][0]']
conv4_block1_2_bn (BatchNormal (None, 14, 14, 256) 1024 ization)	['conv4_block1_2_conv[0][0]']
conv4_block1_2_relu (Activatio (None, 14, 14, 256) 0 n)	['conv4_block1_2_bn[0][0]']
conv4_block1_0_conv (Conv2D) (None, 14, 14, 1024 525312)	['conv3_block4_out[0][0]']
conv4_block1_3_conv (Conv2D) (None, 14, 14, 1024 263168)	['conv4_block1_2_relu[0][0]']
conv4_block1_0_bn (BatchNormal (None, 14, 14, 1024 4096 ization))	['conv4_block1_0_conv[0][0]']
conv4_block1_3_bn (BatchNormal (None, 14, 14, 1024 4096 ization))	['conv4_block1_3_conv[0][0]']
conv4_block1_add (Add) (None, 14, 14, 1024 0)	['conv4_block1_0_bn[0][0]', 'conv4_block1_3_bn[0][0]']
conv4_block1_out (Activation) (None, 14, 14, 1024 0)	['conv4_block1_add[0][0]']
conv4_block2_1_conv (Conv2D) (None, 14, 14, 256) 262400	['conv4_block1_out[0][0]']
conv4_block2_1_bn (BatchNormal (None, 14, 14, 256) 1024 ization)	['conv4_block2_1_conv[0][0]']
conv4_block2_1_relu (Activatio (None, 14, 14, 256) 0 n)	['conv4_block2_1_bn[0][0]']
conv4_block2_2_conv (Conv2D) (None, 14, 14, 256) 590080	['conv4_block2_1_relu[0][0]']

conv4_block2_2_bn (BatchNormal (None, 14, 14, 256) 1024 ization)		['conv4_block2_2_conv[0][0]']
conv4_block2_2_relu (Activatio (None, 14, 14, 256) 0 n)		['conv4_block2_2_bn[0][0]']
conv4_block2_3_conv (Conv2D) (None, 14, 14, 1024 263168)		['conv4_block2_2_relu[0][0]']
conv4_block2_3_bn (BatchNormal (None, 14, 14, 1024 4096 ization))		['conv4_block2_3_conv[0][0]']
conv4_block2_add (Add) (None, 14, 14, 1024 0)		['conv4_block1_out[0][0]', 'conv4_block2_3_bn[0][0]']
conv4_block2_out (Activation) (None, 14, 14, 1024 0)		['conv4_block2_add[0][0]']
conv4_block3_1_conv (Conv2D) (None, 14, 14, 256) 262400		['conv4_block2_out[0][0]']
conv4_block3_1_bn (BatchNormal (None, 14, 14, 256) 1024 ization)		['conv4_block3_1_conv[0][0]']
conv4_block3_1_relu (Activatio (None, 14, 14, 256) 0 n)		['conv4_block3_1_bn[0][0]']
conv4_block3_2_conv (Conv2D) (None, 14, 14, 256) 590080		['conv4_block3_1_relu[0][0]']
conv4_block3_2_bn (BatchNormal (None, 14, 14, 256) 1024 ization)		['conv4_block3_2_conv[0][0]']
conv4_block3_2_relu (Activatio (None, 14, 14, 256) 0 n)		['conv4_block3_2_bn[0][0]']
conv4_block3_3_conv (Conv2D) (None, 14, 14, 1024 263168)		['conv4_block3_2_relu[0][0]']
conv4_block3_3_bn (BatchNormal (None, 14, 14, 1024 4096 ization))		['conv4_block3_3_conv[0][0]']
conv4_block3_add (Add) (None, 14, 14, 1024 0)		['conv4_block2_out[0][0]', 'conv4_block3_3_bn[0][0]']
conv4_block3_out (Activation) (None, 14, 14, 1024 0)		['conv4_block3_add[0][0]']
conv4_block4_1_conv (Conv2D) (None, 14, 14, 256) 262400		['conv4_block3_out[0][0]']
conv4_block4_1_bn (BatchNormal (None, 14, 14, 256) 1024 ization)		['conv4_block4_1_conv[0][0]']
conv4_block4_1_relu (Activatio (None, 14, 14, 256) 0 n)		['conv4_block4_1_bn[0][0]']
conv4_block4_2_conv (Conv2D) (None, 14, 14, 256) 590080		['conv4_block4_1_relu[0][0]']
conv4_block4_2_bn (BatchNormal (None, 14, 14, 256) 1024 ization)		['conv4_block4_2_conv[0][0]']
conv4_block4_2_relu (Activatio (None, 14, 14, 256) 0 n)		['conv4_block4_2_bn[0][0]']
conv4_block4_3_conv (Conv2D) (None, 14, 14, 1024 263168)		['conv4_block4_2_relu[0][0]']
conv4_block4_3_bn (BatchNormal (None, 14, 14, 1024 4096 ization))		['conv4_block4_3_conv[0][0]']
conv4_block4_add (Add) (None, 14, 14, 1024 0)		['conv4_block3_out[0][0]', 'conv4_block4_3_bn[0][0]']
conv4_block4_out (Activation) (None, 14, 14, 1024 0)		['conv4_block4_add[0][0]']
conv4_block5_1_conv (Conv2D) (None, 14, 14, 256) 262400		['conv4_block4_out[0][0]']
conv4_block5_1_bn (BatchNormal (None, 14, 14, 256) 1024 ization)		['conv4_block5_1_conv[0][0]']
conv4_block5_1_relu (Activatio (None, 14, 14, 256) 0 n)		['conv4_block5_1_bn[0][0]']
conv4_block5_2_conv (Conv2D) (None, 14, 14, 256) 590080		['conv4_block5_1_relu[0][0]']
conv4_block5_2_bn (BatchNormal (None, 14, 14, 256) 1024 ization)		['conv4_block5_2_conv[0][0]']

```

        ization)

conv4_block5_2_relu (Activation) (None, 14, 14, 256) 0      ['conv4_block5_2_bn[0][0]']

conv4_block5_3_conv (Conv2D)    (None, 14, 14, 1024) 263168 ['conv4_block5_2_relu[0][0]']

conv4_block5_3_bn (BatchNormal (None, 14, 14, 1024) 4096 ['conv4_block5_3_conv[0][0]']

conv4_block5_add (Add)        (None, 14, 14, 1024) 0      ['conv4_block4_out[0][0]', 'conv4_block5_3_bn[0][0]']

conv4_block5_out (Activation) (None, 14, 14, 1024) 0      ['conv4_block5_add[0][0]']

conv4_block6_1_conv (Conv2D)   (None, 14, 14, 256) 262400 ['conv4_block5_out[0][0]']

conv4_block6_1_bn (BatchNormal (None, 14, 14, 256) 1024 ['conv4_block6_1_conv[0][0]']

conv4_block6_1_relu (Activatio (None, 14, 14, 256) 0      ['conv4_block6_1_bn[0][0]']

conv4_block6_2_conv (Conv2D)   (None, 14, 14, 256) 590080 ['conv4_block6_1_relu[0][0]']

conv4_block6_2_bn (BatchNormal (None, 14, 14, 256) 1024 ['conv4_block6_2_conv[0][0]']

conv4_block6_2_relu (Activatio (None, 14, 14, 256) 0      ['conv4_block6_2_bn[0][0]']

conv4_block6_3_conv (Conv2D)   (None, 14, 14, 1024) 263168 ['conv4_block6_2_relu[0][0]']

conv4_block6_3_bn (BatchNormal (None, 14, 14, 1024) 4096 ['conv4_block6_3_conv[0][0]']

conv4_block6_add (Add)        (None, 14, 14, 1024) 0      ['conv4_block5_out[0][0]', 'conv4_block6_3_bn[0][0]']

conv4_block6_out (Activation) (None, 14, 14, 1024) 0      ['conv4_block6_add[0][0]']

conv5_block1_1_conv (Conv2D)   (None, 7, 7, 512) 524800 ['conv4_block6_out[0][0]']

conv5_block1_1_bn (BatchNormal (None, 7, 7, 512) 2048 ['conv5_block1_1_conv[0][0]']

conv5_block1_1_relu (Activatio (None, 7, 7, 512) 0      ['conv5_block1_1_bn[0][0]']

conv5_block1_2_conv (Conv2D)   (None, 7, 7, 512) 2359808 ['conv5_block1_1_relu[0][0]']

conv5_block1_2_bn (BatchNormal (None, 7, 7, 512) 2048 ['conv5_block1_2_conv[0][0]']

conv5_block1_2_relu (Activatio (None, 7, 7, 512) 0      ['conv5_block1_2_bn[0][0]']

conv5_block1_0_conv (Conv2D)   (None, 7, 7, 2048) 2099200 ['conv4_block6_out[0][0]']

conv5_block1_3_conv (Conv2D)   (None, 7, 7, 2048) 1050624 ['conv5_block1_2_relu[0][0]']

conv5_block1_0_bn (BatchNormal (None, 7, 7, 2048) 8192 ['conv5_block1_0_conv[0][0]']

conv5_block1_3_bn (BatchNormal (None, 7, 7, 2048) 8192 ['conv5_block1_3_conv[0][0]']

conv5_block1_add (Add)        (None, 7, 7, 2048) 0      ['conv5_block1_0_bn[0][0]', 'conv5_block1_3_bn[0][0]']

conv5_block1_out (Activation) (None, 7, 7, 2048) 0      ['conv5_block1_add[0][0]']

conv5_block2_1_conv (Conv2D)   (None, 7, 7, 512) 1049088 ['conv5_block1_out[0][0]']

conv5_block2_1_bn (BatchNormal (None, 7, 7, 512) 2048 ['conv5_block2_1_conv[0][0]']

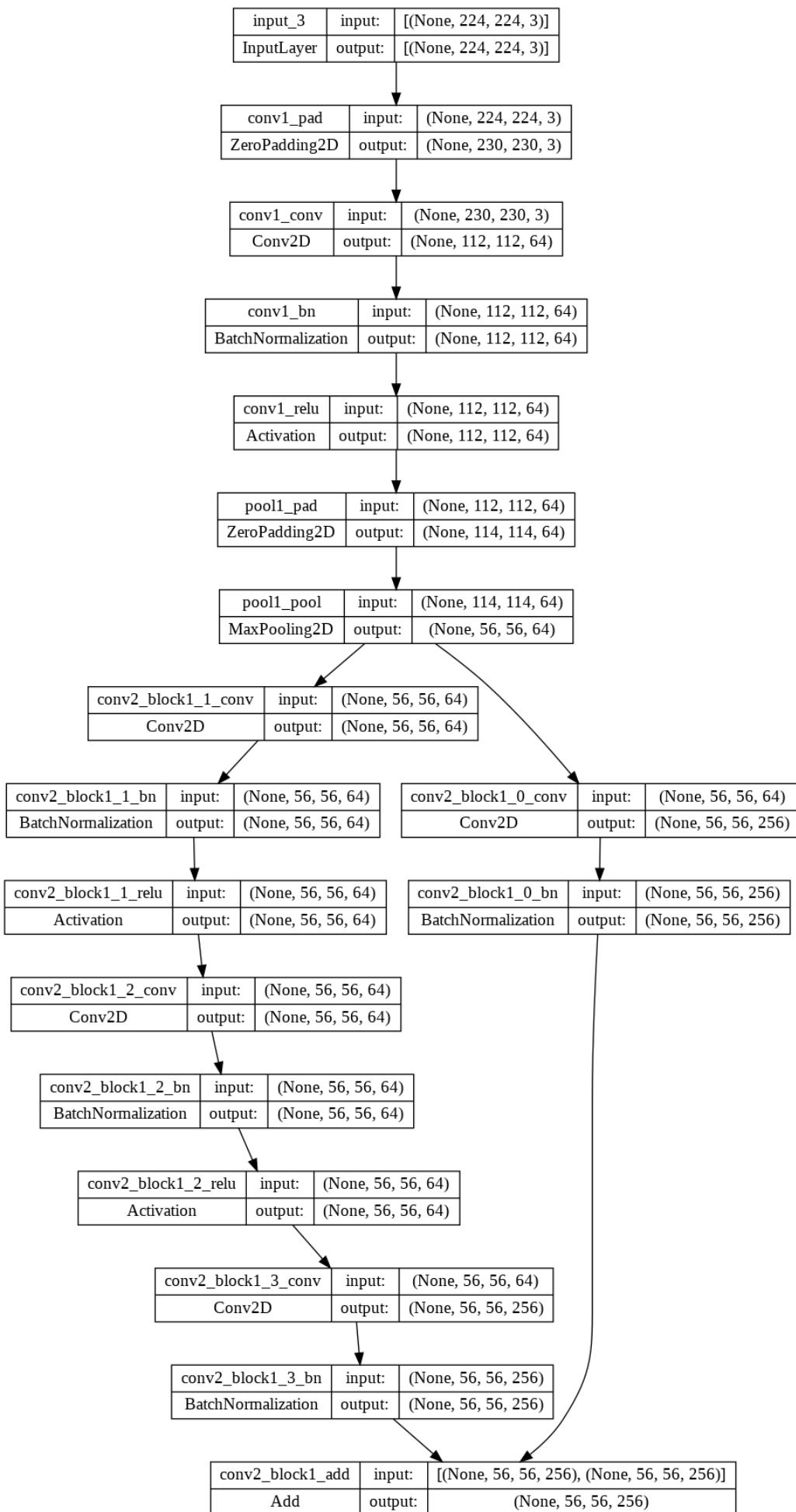
conv5_block2_1_relu (Activatio (None, 7, 7, 512) 0      ['conv5_block2_1_bn[0][0]']

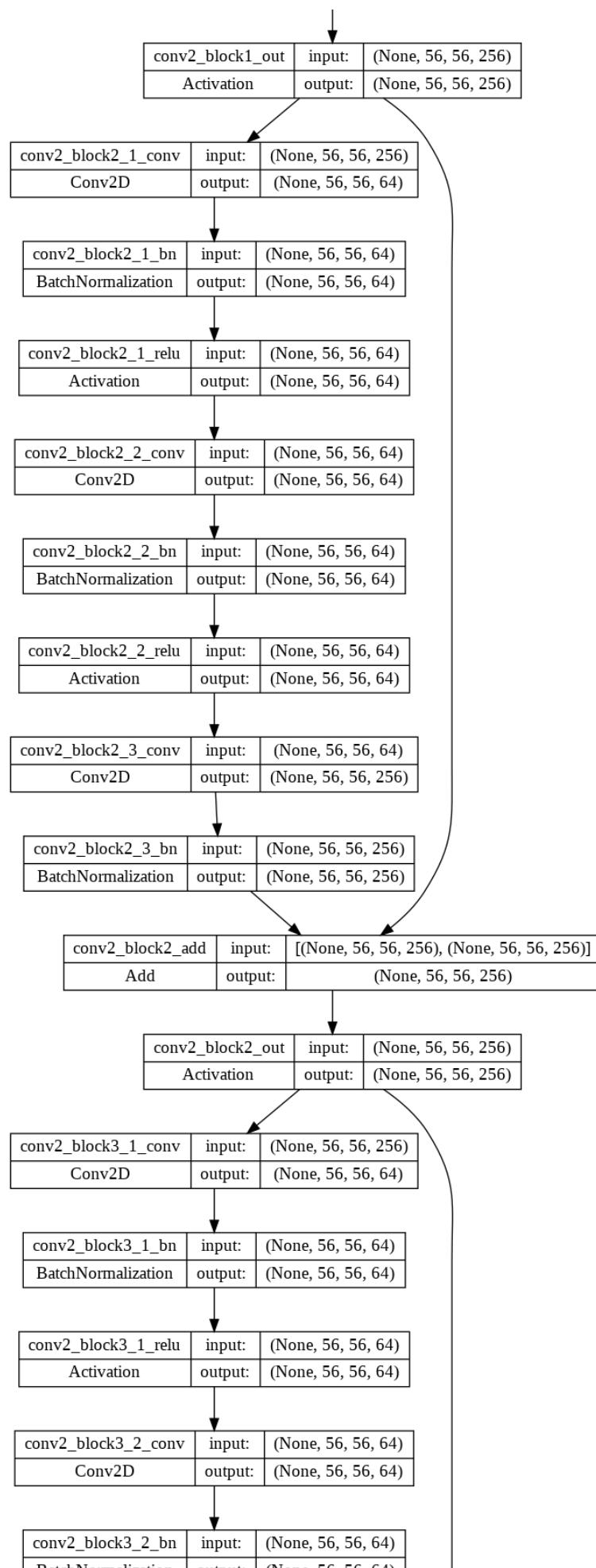
conv5_block2_2_conv (Conv2D)   (None, 7, 7, 512) 2359808 ['conv5_block2_1_relu[0][0]']

```

conv5_block2_2_bn (BatchNormal ization)	(None, 7, 7, 512)	2048	['conv5_block2_2_conv[0][0]']
conv5_block2_2_relu (Activatio n)	(None, 7, 7, 512)	0	['conv5_block2_2_bn[0][0]']
conv5_block2_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	['conv5_block2_2_relu[0][0]']
conv5_block2_3_bn (BatchNormal ization)	(None, 7, 7, 2048)	8192	['conv5_block2_3_conv[0][0]']
conv5_block2_add (Add)	(None, 7, 7, 2048)	0	['conv5_block1_out[0][0]', 'conv5_block2_3_bn[0][0]']
conv5_block2_out (Activation)	(None, 7, 7, 2048)	0	['conv5_block2_add[0][0]']
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 512)	1049088	['conv5_block2_out[0][0]']
conv5_block3_1_bn (BatchNormal ization)	(None, 7, 7, 512)	2048	['conv5_block3_1_conv[0][0]']
conv5_block3_1_relu (Activatio n)	(None, 7, 7, 512)	0	['conv5_block3_1_bn[0][0]']
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 512)	2359808	['conv5_block3_1_relu[0][0]']
conv5_block3_2_bn (BatchNormal ization)	(None, 7, 7, 512)	2048	['conv5_block3_2_conv[0][0]']
conv5_block3_2_relu (Activatio n)	(None, 7, 7, 512)	0	['conv5_block3_2_bn[0][0]']
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	['conv5_block3_2_relu[0][0]']
conv5_block3_3_bn (BatchNormal ization)	(None, 7, 7, 2048)	8192	['conv5_block3_3_conv[0][0]']
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	['conv5_block2_out[0][0]', 'conv5_block3_3_bn[0][0]']
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	['conv5_block3_add[0][0]']
flatten_1 (Flatten)	(None, 100352)	0	['conv5_block3_out[0][0]']
batch_normalization_2 (BatchNo rmalization)	(None, 100352)	401408	['flatten_1[0][0]']
dense_2 (Dense)	(None, 64)	6422592	['batch_normalization_2[0][0]']
batch_normalization_3 (BatchNo rmalization)	(None, 64)	256	['dense_2[0][0]']
dense_3 (Dense)	(None, 2)	130	['batch_normalization_3[0][0]']
<hr/>			
Total params:	30,412,098		
Trainable params:	30,158,146		
Non-trainable params:	253,952		

Out[13]:





```
In [14]: BATCH_SIZE = 8
EPOCHS = 50
```

```
# Fit of the model that conv2_block3_2 will input: (None, 56, 56, 64)
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=[TqdmCallback(checkpoint)],
                  validation_data = (X_val, Y_val))
Conv2D | output: (None, 56, 56, 256)
```

conv2_block3_bn	input:	(None, 56, 56, 256)
BatchNormalization	output:	(None, 56, 56, 256)

conv2_block3_add	input:	[(None, 56, 56, 256), (None, 56, 56, 256)]
Add	output:	(None, 56, 56, 256)

conv2_block3_out	input:	(None, 56, 56, 256)
Activation	output:	(None, 56, 56, 256)

conv3_block1_1_conv	input:	(None, 56, 56, 256)
Conv2D	output:	(None, 28, 28, 128)

conv3_block1_1_bn	input:	(None, 28, 28, 128)
BatchNormalization	output:	(None, 28, 28, 128)

conv3_block1_0_conv	input:	(None, 56, 56, 256)
Conv2D	output:	(None, 28, 28, 512)

conv3_block1_1_relu	input:	(None, 28, 28, 128)
Activation	output:	(None, 28, 28, 128)

conv3_block1_0_bn	input:	(None, 28, 28, 512)
BatchNormalization	output:	(None, 28, 28, 512)

conv3_block1_2_conv	input:	(None, 28, 28, 128)
Conv2D	output:	(None, 28, 28, 128)

conv3_block1_2_bn	input:	(None, 28, 28, 128)
BatchNormalization	output:	(None, 28, 28, 128)

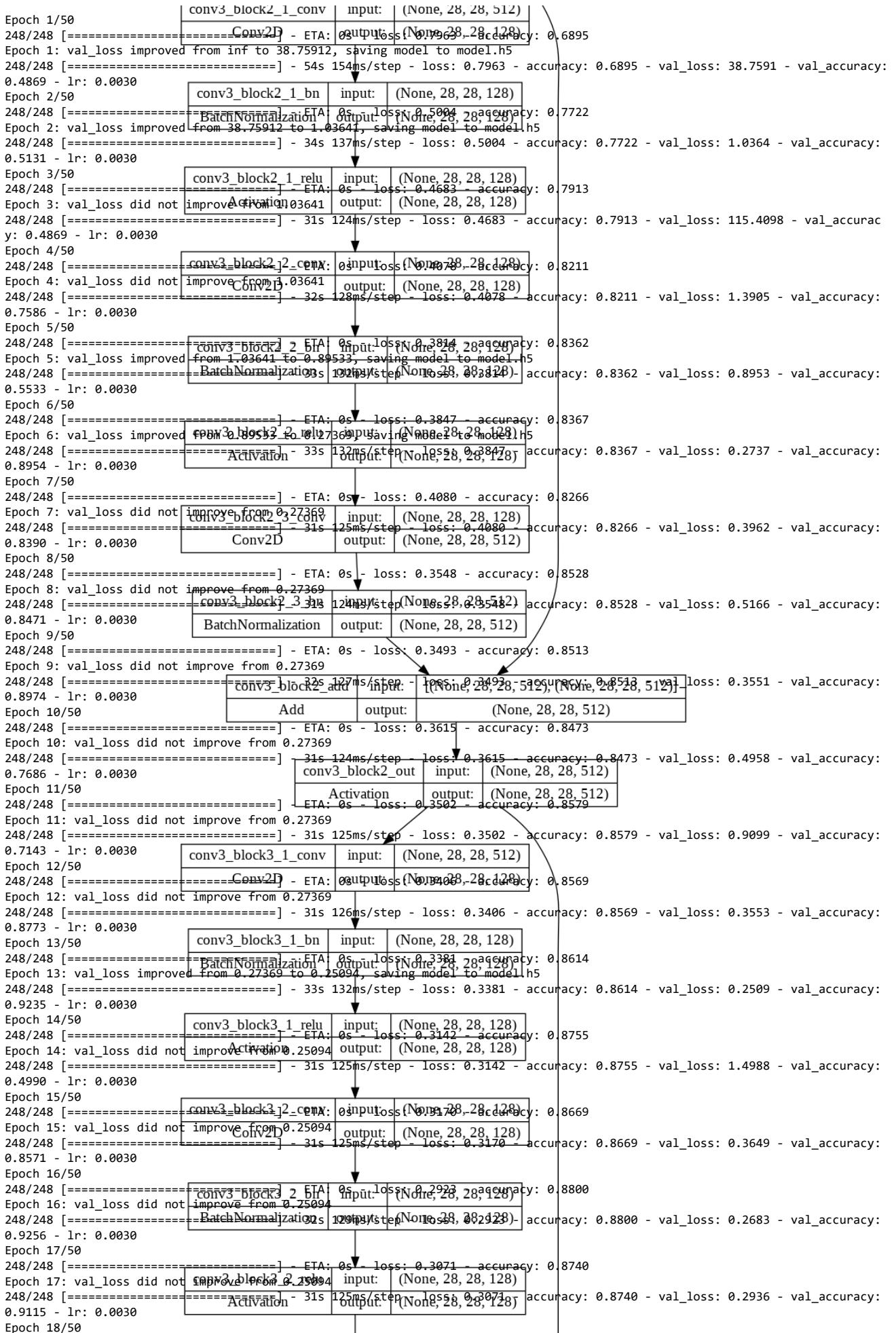
conv3_block1_2_relu	input:	(None, 28, 28, 128)
Activation	output:	(None, 28, 28, 128)

conv3_block1_3_conv	input:	(None, 28, 28, 128)
Conv2D	output:	(None, 28, 28, 512)

conv3_block1_3_bn	input:	(None, 28, 28, 512)
BatchNormalization	output:	(None, 28, 28, 512)

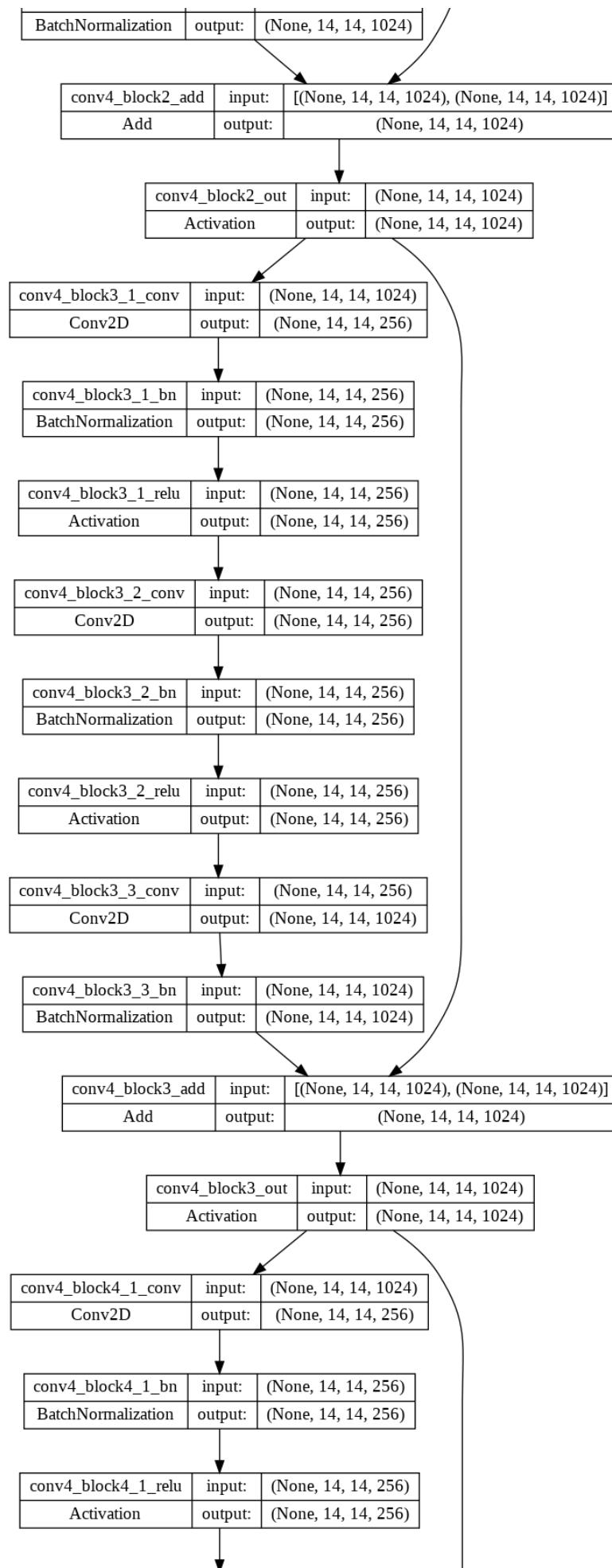
conv3_block1_add	input:	[(None, 28, 28, 512), (None, 28, 28, 512)]
Add	output:	(None, 28, 28, 512)

conv3_block1_out	input:	(None, 28, 28, 512)
Activation	output:	(None, 28, 28, 512)



248/248 [=====] - ETA: 0s - loss: 0.2916 - accuracy: 0.8795
 Epoch 18: val_loss did not improve from 0.25094 to 0.234429, saving model to model.h5
 248/248 [=====] - 31s 124ms/step - loss: 0.2916 - accuracy: 0.8795 - val_loss: 0.3963 - val_accuracy: 0.8793 - lr: 0.0030
 Conv2D input: (None, 28, 28, 128) output: (None, 28, 28, 512)
 Epoch 19/50
 248/248 [=====] - ETA: 0s - loss: 0.3132 - accuracy: 0.8750
 Epoch 19: val_loss improved from 0.25094 to 0.234429, saving model to model.h5
 248/248 [=====] - 31s 124ms/step - loss: 0.2916 - accuracy: 0.8750 - val_loss: 0.2343 - val_accuracy: 0.9155 - lr: 0.0030
 BatchNormalization input: (None, 28, 28, 512)
 Epoch 20/50
 248/248 [=====] - ETA: 0s - loss: 0.2951 - accuracy: 0.8831
 Epoch 20: val_loss did not improve from 0.23429
 248/248 [=====] - 31s 124ms/step - loss: 0.2951 - accuracy: 0.8831 - val_loss: 0.2396 - val_accuracy: 0.9235 - lr: 0.0030
 Add input: (None, 28, 28, 512) output: (None, 28, 28, 512)
 Epoch 21/50
 248/248 [=====] - ETA: 0s - loss: 0.3029 - accuracy: 0.8795
 Epoch 21: val_loss did not improve from 0.23429
 248/248 [=====] - 31s 124ms/step - loss: 0.3029 - accuracy: 0.8795 - val_loss: 0.2707 - val_accuracy: 0.9396 - lr: 0.0030
 conv3_block3_out input: (None, 28, 28, 512)
 Epoch 22/50
 Activation input: (None, 28, 28, 512)
 248/248 [=====] - ETA: 0s - loss: 0.2703 - accuracy: 0.8881
 Epoch 22: val_loss did not improve from 0.23429
 248/248 [=====] - 31s 124ms/step - loss: 0.2703 - accuracy: 0.8881 - val_loss: 0.3674 - val_accuracy: 0.8652 - lr: 0.0030
 conv3_block4_1_conv input: (None, 28, 28, 512)
 Epoch 23/50
 Conv2D input: (None, 28, 28, 128) output: (None, 28, 28, 128)
 248/248 [=====] - ETA: 0s - loss: 0.2951 - accuracy: 0.8816
 Epoch 23: val_loss did not improve from 0.23429
 248/248 [=====] - 31s 133ms/step - loss: 0.2921 - accuracy: 0.8816 - val_loss: 0.3202 - val_accuracy: 0.8934 - lr: 0.0030
 conv3_block4_1_bn input: (None, 28, 28, 128)
 Epoch 24/50
 BatchNormalization input: (None, 28, 28, 128) output: (None, 28, 28, 128)
 248/248 [=====] - ETA: 0s - loss: 0.2495 - accuracy: 0.8952
 Epoch 24: val_loss did not improve from 0.23429
 248/248 [=====] - 31s 124ms/step - loss: 0.2495 - accuracy: 0.8952 - val_loss: 0.2928 - val_accuracy: 0.9155 - lr: 0.0030
 conv3_block4_1_relu input: (None, 28, 28, 128)
 Epoch 25/50
 Activation input: (None, 28, 28, 128) output: (None, 28, 28, 128)
 248/248 [=====] - ETA: 0s - loss: 0.2786 - accuracy: 0.8896
 Epoch 25: val_loss improved from 0.23429 to 0.189356, saving model to model.h5
 248/248 [=====] - 32s 131ms/step - loss: 0.2786 - accuracy: 0.8896 - val_loss: 0.1836 - val_accuracy: 0.9376 - lr: 0.0030
 conv3_block4_2_conv input: (None, 28, 28, 128)
 Epoch 26/50
 Conv2D input: (None, 28, 28, 128) output: (None, 28, 28, 128)
 248/248 [=====] - ETA: 0s - loss: 0.2451 - accuracy: 0.8921
 Epoch 26: ReduceLROnPlateau reducing learning rate to 0.00210000018853983.
 Epoch 26: val_loss did not improve from 0.18356
 248/248 [=====] - 31s 125ms/step - loss: 0.2739 - accuracy: 0.8921 - val_loss: 0.3601 - val_accuracy: 0.9034 - lr: 0.0030
 conv3_block4_2_bn input: (None, 28, 28, 128)
 Epoch 27/50
 BatchNormalization input: (None, 28, 28, 128) output: (None, 28, 28, 128)
 248/248 [=====] - ETA: 0s - loss: 0.2451 - accuracy: 0.8977
 Epoch 27: val_loss improved from 0.18356 to 0.17992, saving model to model.h5
 248/248 [=====] - 32s 130ms/step - loss: 0.2507 - accuracy: 0.8977 - val_loss: 0.1799 - val_accuracy: 0.9296 - lr: 0.0021
 conv3_block4_2_relu input: (None, 28, 28, 128)
 248/248 [=====] - ETA: 0s - loss: 0.2451 - accuracy: 0.9017
 Epoch 28/50
 Activation input: (None, 28, 28, 128) output: (None, 28, 28, 128)
 248/248 [=====] - 31s 124ms/step - loss: 0.2451 - accuracy: 0.9017 - val_loss: 0.6782 - val_accuracy: 0.8189 - lr: 0.0021
 conv3_block4_3_conv input: (None, 28, 28, 128)
 248/248 [=====] - ETA: 0s - loss: 0.2550 - accuracy: 0.8926
 Epoch 29/50
 Conv2D input: (None, 28, 28, 128) output: (None, 28, 28, 512)
 248/248 [=====] - 31s 124ms/step - loss: 0.2550 - accuracy: 0.8926 - val_loss: 0.2261 - val_accuracy: 0.9095 - lr: 0.0021
 conv3_block4_3_relu input: (None, 28, 28, 512)
 248/248 [=====] - ETA: 0s - loss: 0.2550 - accuracy: 0.9098
 Epoch 30/50
 Activation input: (None, 28, 28, 512) output: (None, 28, 28, 512)
 248/248 [=====] - 32s 128ms/step - loss: 0.2236 - accuracy: 0.9098 - val_loss: 0.4999 - val_accuracy: 0.8169 - lr: 0.0021
 conv3_block4_3_bn input: (None, 28, 28, 512)
 Epoch 31/50
 BatchNormalization input: (None, 28, 28, 512) output: (None, 28, 28, 512)
 248/248 [=====] - 34s 136ms/step - loss: 0.2164 - accuracy: 0.9158 - val_loss: 0.1409 - val_accuracy: 0.9497 - lr: 0.0021
 conv3_block4_3_relu input: (None, 28, 28, 512)
 248/248 [=====] - ETA: 0s - loss: 0.2164 - accuracy: 0.9158
 Epoch 32/50
 Activation input: (None, 28, 28, 512) output: (None, 28, 28, 512)
 248/248 [=====] - 31s 124ms/step - loss: 0.2640 - accuracy: 0.8992
 Epoch 32: val_loss did not improve from 0.14091
 248/248 [=====] - 31s 124ms/step - loss: 0.2640 - accuracy: 0.8992 - val_loss: 0.3677 - val_accuracy: 0.8290 - lr: 0.0021
 conv3_block4_4_conv input: (None, 28, 28, 512)
 Epoch 33/50
 Conv2D input: (None, 28, 28, 512) output: (None, 14, 14, 256)
 248/248 [=====] - ETA: 0s - loss: 0.2363 - accuracy: 0.9062
 Epoch 33: val_loss did not improve from 0.14091
 248/248 [=====] - 31s 123ms/step - loss: 0.2363 - accuracy: 0.9062 - val_loss: 0.3476 - val_accuracy: 0.8692 - lr: 0.0021
 conv3_block4_4_relu input: (None, 14, 14, 256)
 248/248 [=====] - ETA: 0s - loss: 0.2382 - accuracy: 0.9078
 Epoch 34/50
 Activation input: (None, 14, 14, 256) output: (None, 14, 14, 1024)
 248/248 [=====] - ETA: 0s - loss: 0.2382 - accuracy: 0.9078
 Epoch 34: val_loss did not improve from 0.14091
 248/248 [=====] - 31s 124ms/step - loss: 0.2382 - accuracy: 0.9078 - val_loss: 0.3476 - val_accuracy: 0.9175 - lr: 0.0021
 conv4_block1_1_bn input: (None, 14, 14, 1024)
 BatchNormalization input: (None, 14, 14, 1024) output: (None, 14, 14, 1024)

Epoch 35/50
248/248 [=====] - ETA: 0s - loss: 0.2249 - accuracy: 0.9083
Epoch 35: val_loss did not improve from 0.14991
248/248 [=====conv4_block1_1_relu input: 31s 124ms/step, 256bs: 0.2249_accuracy: 0.9083 output: val1_Noss: 14.1964024_val_accuracy: 0.9175 - lr: 0.0021 Activation output: (None, 14, 14, 256) BatchNormalization output: (None, 14, 14, 1024)
Epoch 36/50
248/248 [=====] - ETA: 0s - loss: 0.2620 - accuracy: 0.9042
Epoch 36: ReduceLROnPlateau reducing learning rate to 0.001469999475851653.
Epoch 36: val_loss did not improve from 0.14991
248/248 [=====Conv2D - 32s 128ms/step(Noss: 14.0126206) accuracy: 0.9042 - val_loss: 0.2446 - val_accuracy: 0.8954 - lr: 0.0021 Activation output: (None, 14, 14, 256)
Epoch 37/50
248/248 [=====] - ETA: 0s - loss: 0.2133 - accuracy: 0.9204
Epoch 37: val_loss did not improve from 0.14991
248/248 [=====BatchNormalization - 31s 125ms/step(Noss: 14.02133) accuracy: 0.9204 - val_loss: 0.1885 - val_accuracy: 0.9457 - lr: 0.0015 Activation output: (None, 14, 14, 256)
Epoch 38/50
248/248 [=====] - ETA: 0s - loss: 0.2125 - accuracy: 0.9173
Epoch 38: val_loss improved from 0.14991 to 0.12711, saving model to model.h5
248/248 [=====Conv4_Block1_2_ReLu input: (None, 14, 14, 256) - 34s 136ms/step - loss: 0.2125 - accuracy: 0.9173 - val_loss: 0.1271 - val_accuracy: 0.9598 - lr: 0.0015 Activation output: (None, 14, 14, 256)
Epoch 39/50
248/248 [=====] - ETA: 0s - loss: 0.2249 - accuracy: 0.9189
Epoch 39: val_loss did not improve from 0.12711
248/248 [=====Conv4_Block1_3_Bn input: (None, 14, 14, 1024) - 31s 128ms/step loss: 0.2249 accuracy: 0.9189 - val_loss: 0.2699 - val_accuracy: 0.8873 - lr: 0.0015 Conv2D output: (None, 14, 14, 1024)
Epoch 40/50
248/248 [=====] - ETA: 0s - loss: 0.2063 - accuracy: 0.9219
Epoch 40: val_loss did not improve from 0.12711
248/248 [=====Conv4_Block1_3_Bn input: (None, 14, 14, 1024) - 31s 125ms/step loss: 0.2063 - accuracy: 0.9219 - val_loss: 0.4892 - val_accuracy: 0.8471 - lr: 0.0015 Activation output: (None, 14, 14, 1024)
Epoch 41/50
248/248 [=====BatchNormalization - 31s 125ms/step - loss: 0.2036 - accuracy: 0.9274 Activation output: (None, 14, 14, 1024)
Epoch 41: val_loss did not improve from 0.12711
248/248 [=====Conv4_Block1_Add input: [(None, 14, 14, 1024), (None, 14, 14, 1024)] - 31s 125ms/step - loss: 0.2036 - accuracy: 0.9274 - val_loss: 0.1292 - val_accuracy: 0.9557 - lr: 0.0015 Activation output: (None, 14, 14, 1024)
Epoch 42/50
248/248 [=====] - ETA: 0s - loss: 0.1929 - accuracy: 0.9294 - val_loss: 0.1202 - val_accuracy: 0.9718 - lr: 0.0015 Activation output: (None, 14, 14, 1024)
Epoch 43/50
248/248 [=====Activation - 31s 124ms/step - loss: 0.1791 - accuracy: 0.9315 - val_loss: 0.1473 - val_accuracy: 0.9638 - lr: 0.0015 Activation output: (None, 14, 14, 1024)
Epoch 44/50
248/248 [=====Conv2D - 31s 124ms/step - loss: 0.2035 - accuracy: 0.9294 - val_loss: 0.2336 - val_accuracy: 0.9155 - lr: 0.0015 Activation output: (None, 14, 14, 1024)
Epoch 45/50
248/248 [=====Conv4_Block2_1_Conv input: (None, 14, 14, 1024) - 31s 124ms/step - loss: 0.2035 - accuracy: 0.9294 - val_loss: 0.2336 - val_accuracy: 0.9234 Activation output: (None, 14, 14, 256)
Epoch 45: val_loss did not improve from 0.12022 Activation output: (None, 14, 14, 256)
248/248 [=====BatchNormalization - 32s 128ms/step - loss: 0.2053 - accuracy: 0.9234 - val_loss: 0.1789 - val_accuracy: 0.9457 - lr: 0.0015 Activation output: (None, 14, 14, 256)
Epoch 46/50
248/248 [=====Conv4_Block2_1_Bn input: (None, 14, 14, 256) - 31s 125ms/step - loss: 0.2053 - accuracy: 0.9234 - val_loss: 0.1789 - val_accuracy: 0.9457 - lr: 0.0015 Activation output: (None, 14, 14, 256)
Epoch 46: val_loss did not improve from 0.12022 Activation output: (None, 14, 14, 256)
248/248 [=====Conv4_Block2_1_Bn input: (None, 14, 14, 256) - 32s 129ms/step - loss: 0.1666 - accuracy: 0.9400 - val_loss: 0.1379 - val_accuracy: 0.9497 - lr: 0.0015 Activation output: (None, 14, 14, 256)
Epoch 47/50
248/248 [=====Conv2D - 32s 129ms/step - loss: 0.1888 - accuracy: 0.9209 - val_loss: 0.8486 - val_accuracy: 0.7586 - lr: 0.0015 Activation output: (None, 14, 14, 256)
Epoch 48/50
248/248 [=====Conv4_Block2_2_Bn input: (None, 14, 14, 256) - 32s 130ms/step - loss: 0.1888 - accuracy: 0.9209 - val_loss: 0.8486 - val_accuracy: 0.7586 - lr: 0.0015 Activation output: (None, 14, 14, 256)
248/248 [=====BatchNormalization - 32s 129ms/step - loss: 0.1602 - accuracy: 0.9415 Activation output: (None, 14, 14, 256)
Epoch 48: val_loss did not improve from 0.12022 Activation output: (None, 14, 14, 256)
248/248 [=====Conv2D - 32s 129ms/step - loss: 0.1602 - accuracy: 0.9415 - val_loss: 0.1266 - val_accuracy: 0.9678 - lr: 0.0010 Activation output: (None, 14, 14, 256)
Epoch 49/50
248/248 [=====Conv4_Block2_2_RelU input: (None, 14, 14, 256) - 31s 125ms/step - loss: 0.1623 - accuracy: 0.9430 Activation output: (None, 14, 14, 256)
Epoch 49: val_loss did not improve from 0.12022 Activation output: (None, 14, 14, 256)
248/248 [=====Conv2D - 31s 125ms/step - loss: 0.1623 - accuracy: 0.9430 - val_loss: 0.1352 - val_accuracy: 0.9517 - lr: 0.0010 Activation output: (None, 14, 14, 256)
Epoch 50/50
248/248 [=====Conv4_Block2_3_CoNs input: (None, 14, 14, 256) - 32s 128ms/step - loss: 0.1413 - accuracy: 0.9511 - val_loss: 0.2463 - val_accuracy: 0.9135 - lr: 0.0010 Activation output: (None, 14, 14, 256)
248/248 [=====Conv2D - 32s 128ms/step - loss: 0.1413 - accuracy: 0.9511 - val_loss: 0.2463 - val_accuracy: 0.9135 - lr: 0.0010 Activation output: (None, 14, 14, 256)
248/248 [=====Conv4_Block2_3_Bn input: (None, 14, 14, 1024) - 32s 128ms/step - loss: 0.1413 - accuracy: 0.9511 - val_loss: 0.2463 - val_accuracy: 0.9135 - lr: 0.0010 Activation output: (None, 14, 14, 1024)



```
In [15]: 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, square = True, tick1_14_256 = disease_types, yticklabels = disease_types)
ax.set_ylabel('Actual', fontsize = 40)
ax.set_xlabel('Predicted', fontsize = 40)

TP = cm[1][1]
print(f"True Positive: {TP}")

FN = cm[1][0]
print(f"False Negative: {FN}")

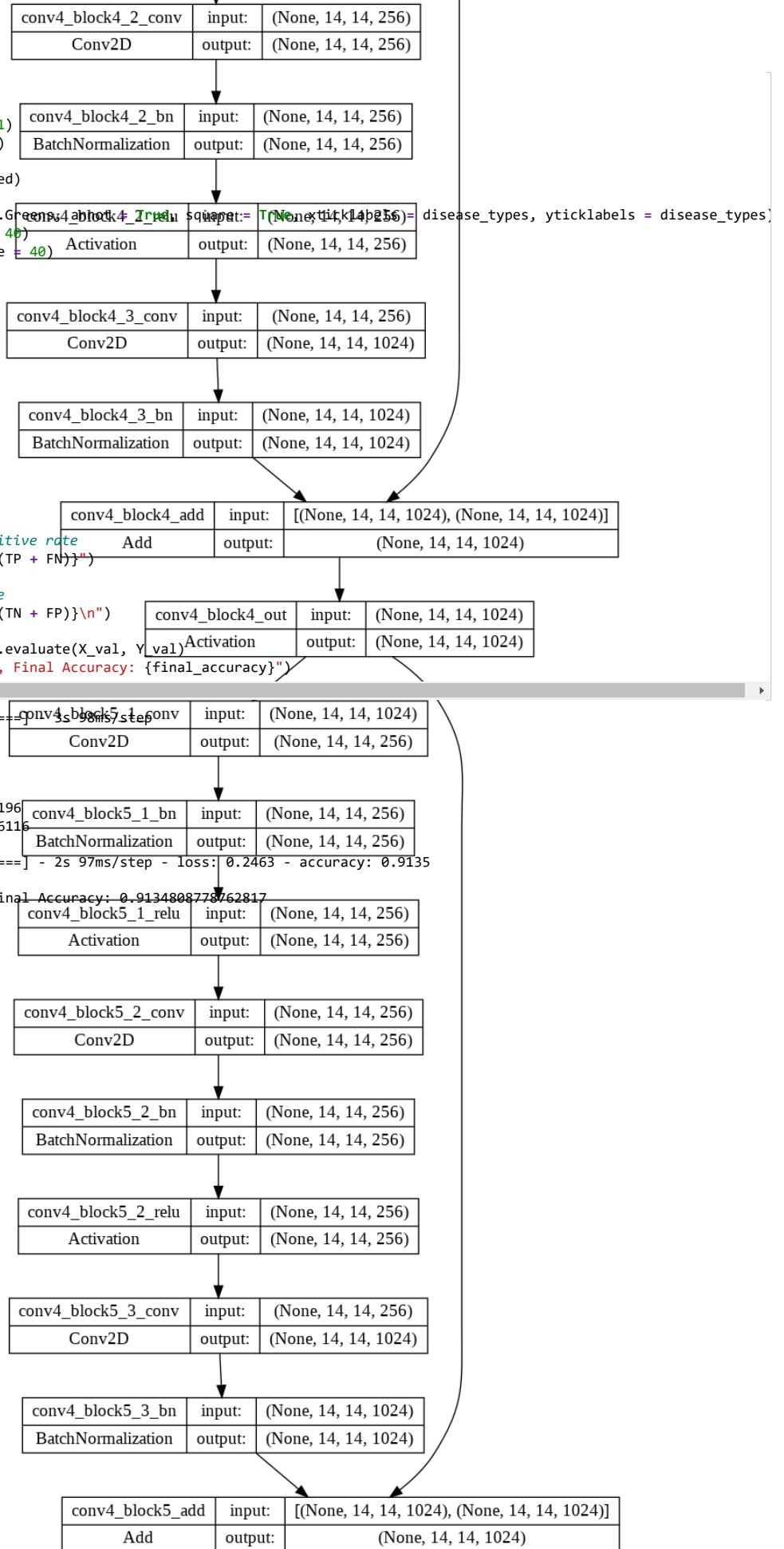
TN = cm[0][0]
print(f"True Negative: {TN}")

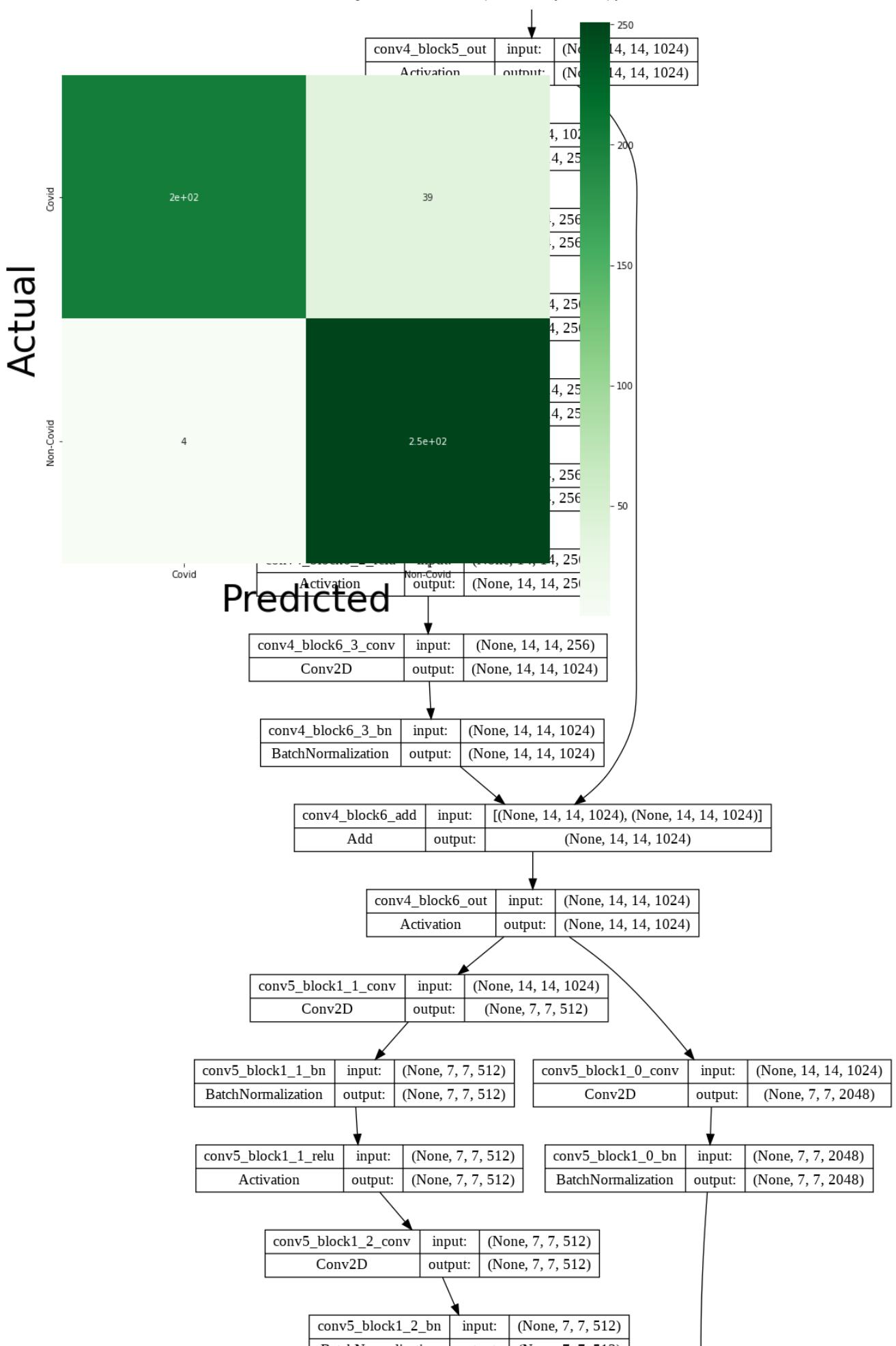
FP = cm[0][1]
print(f"False Positive: {FP}")

# Sensitivity, recall, or true positive rate
print(f"True Positive Rate: {TP / (TP + FN)}")

# Specificity or true negative rate
print(f"True Negative Rate: {TN / (TN + FP)}\n")

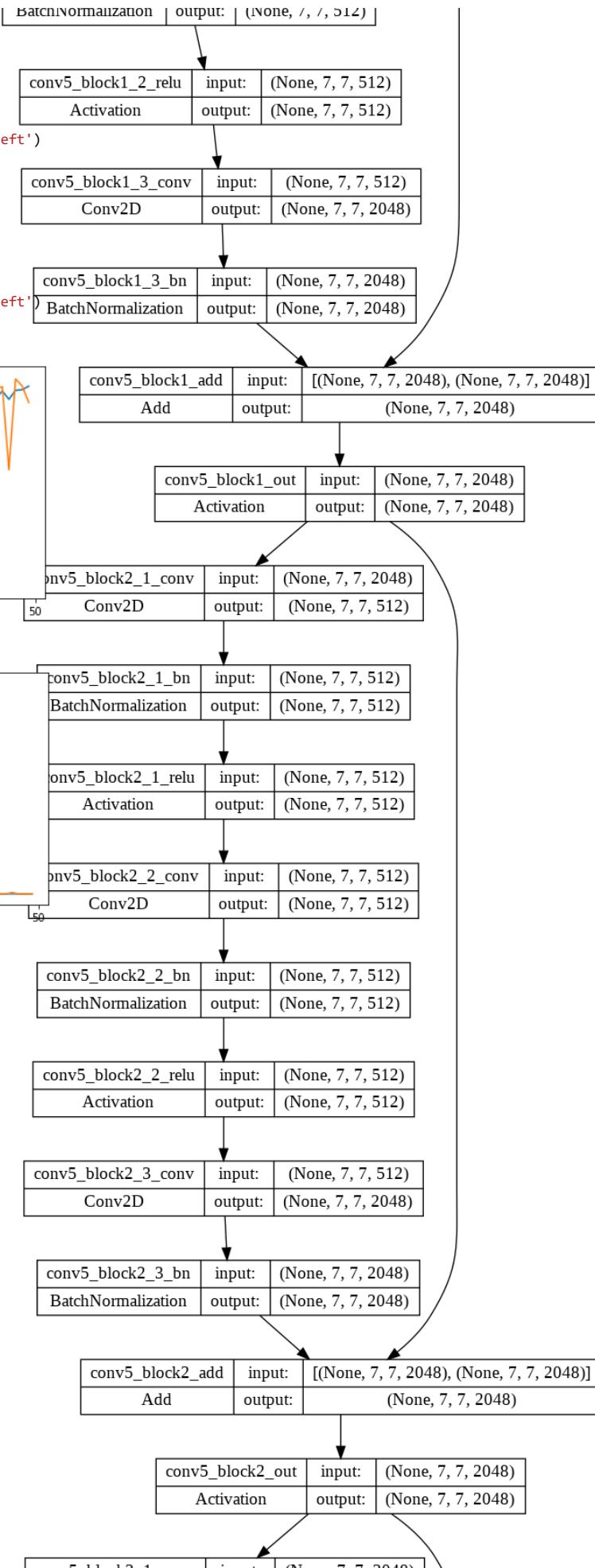
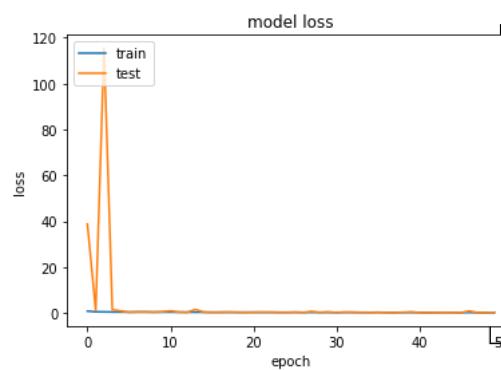
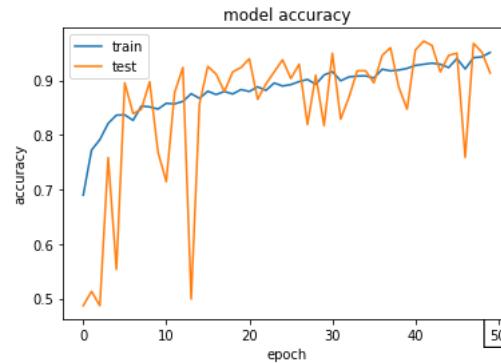
final_loss, final_accuracy = model.evaluate(X_val, Y_val)
print(f"\nFinal Loss: {final_loss}, Final Accuracy: {final_accuracy}")
```





```
In [16]: # 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 [18]: from keras.preprocessing import image
from tensorflow.keras.preprocessing import image

img = image.load_img('/content/drive/MyDrive/Colab Notebooks/Capstone Project/Dataset/Covid/Covid (1015).png', grayscale = False)
show_img = image.load_img('/content/drive/MyDrive/Colab Notebooks/Capstone Project/Dataset/Covid/Covid (1011).png', grayscale = False)
disease_class = ['Covid-19', 'Non Covid-19']
x = image.img_to_array(img)
x = np.expand_dims(x, axis = 0)
x /= 255

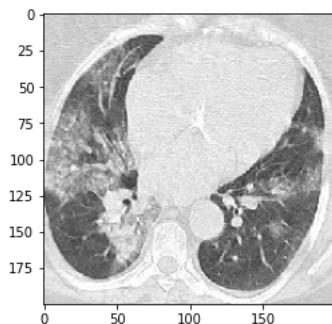
custom = model.predict(x)
print(custom[0])

plt.imshow(show_img)
plt.show()

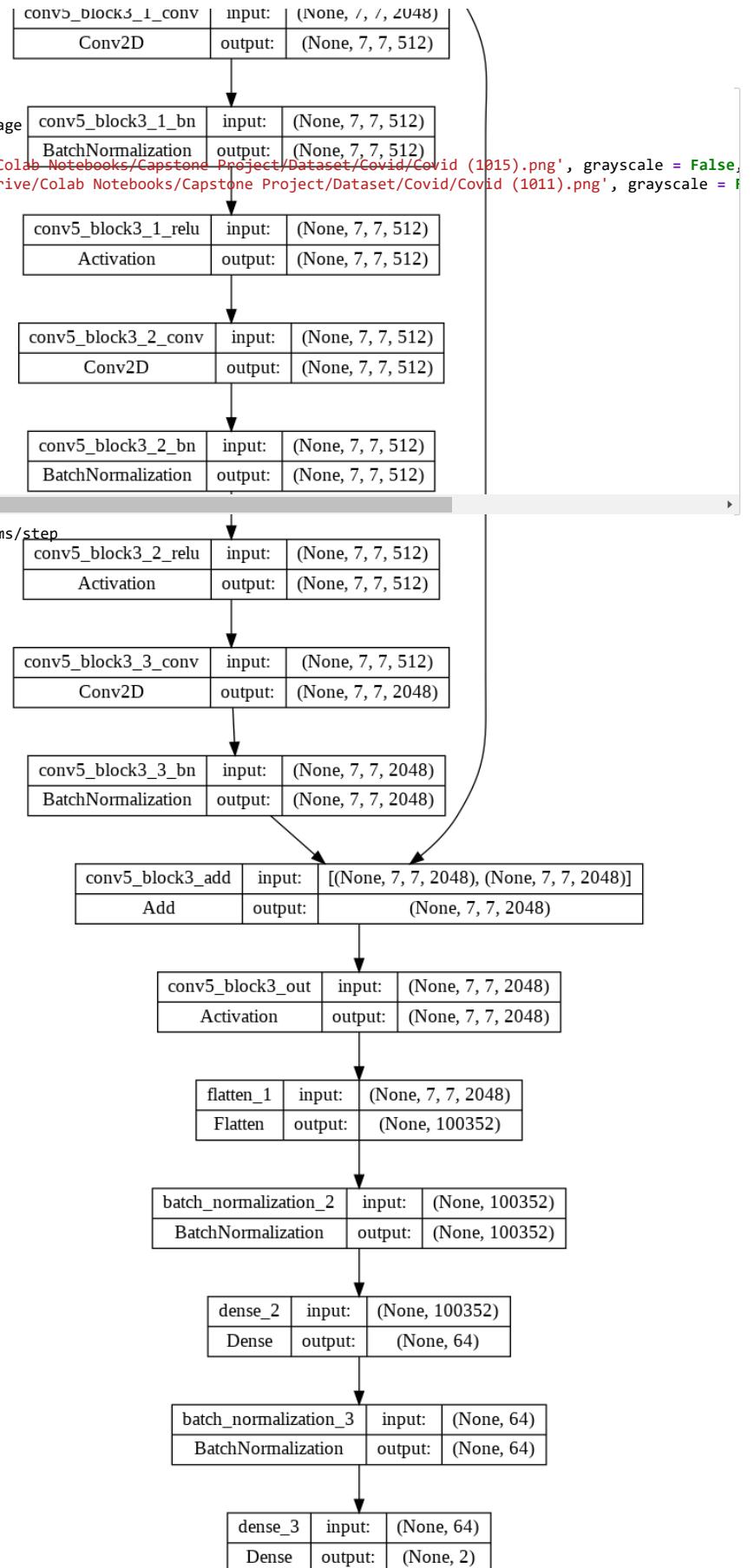
a = custom[0]
ind = np.argmax(a)

print('Prediction:', disease_class[ind])
```

1/1 [=====] - 1s 914ms/step
[0.9855278 0.01447222]



Prediction: Covid-19



Test 2

In [22]:

```
from keras.preprocessing import image
from tensorflow.keras.preprocessing import image

img = image.load_img('/content/drive/MyDrive/Colab Notebooks/Capstone Project/Dataset/Non-Covid/Non-Covid (1017).png', grayscale=True)
show_img = image.load_img('/content/drive/MyDrive/Colab Notebooks/Capstone Project/Dataset/Non-Covid/Non-Covid (1017).png', grayscale=True)
disease_class = ['Covid-19', 'Non Covid-19']
x = image.img_to_array(img)
x = np.expand_dims(x, axis = 0)
x /= 255

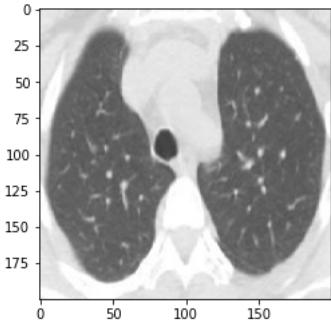
custom = model.predict(x)
print(custom[0])

plt.imshow(show_img)
plt.show()

a = custom[0]
ind = np.argmax(a)

print('Prediction:', disease_class[ind])
```

1/1 [=====] - 0s 23ms/step
[1.2685193e-05 9.9998736e-01]



Prediction: Non Covid-19

Test 3

```
In [23]: from keras.preprocessing import image
from tensorflow.keras.preprocessing import image

img = image.load_img('/content/drive/MyDrive/Colab Notebooks/Capstone Project/Dataset/Non-Covid/Non-Covid (1027).png', grayscale=True)
show_img = image.load_img('/content/drive/MyDrive/Colab Notebooks/Capstone Project/Dataset/Non-Covid/Non-Covid (1017).png', grayscale=True)
disease_class = ['Covid-19', 'Non Covid-19']
x = image.img_to_array(img)
x = np.expand_dims(x, axis = 0)
x /= 255

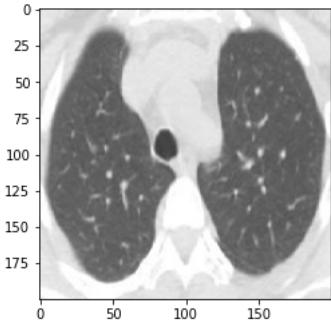
custom = model.predict(x)
print(custom[0])

plt.imshow(show_img)
plt.show()

a = custom[0]
ind = np.argmax(a)

print('Prediction:', disease_class[ind])
```

1/1 [=====] - 0s 22ms/step
[6.330032e-05 9.999367e-01]



Prediction: Non Covid-19

As we can see all the points are predicted correctly.

The Capstone Project has been completed Successfully.