

In [1]:

```
!pip install opencv-python
```

Requirement already satisfied: opencv-python in c:\miniconda\lib\site-packages (4.5.4.60)
Requirement already satisfied: numpy>=1.19.3 in c:\miniconda\lib\site-packages (from open
cv-python) (1.20.3)

In [2]:

```
!pip install keras_applications
```

Collecting keras_applications
 Downloading Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
Requirement already satisfied: numpy>=1.9.1 in c:\miniconda\lib\site-packages (from keras
_applications) (1.20.3)
Requirement already satisfied: h5py in c:\miniconda\lib\site-packages (from keras_applica
tions) (3.6.0)
Installing collected packages: keras-applications
Successfully installed keras-applications-1.0.8

In [3]:

```
import pandas as pd
import numpy as np
import os
import shutil
import glob
import matplotlib.pyplot as plt

import cv2
```

Check meta Data

In [4]:

```
# covid data
covid_imgs = pd.read_csv("C:\IE 7615\Project\Covid_detection_CNN-master\COVID-19_Radiogra  
phy_Dataset\metadata.csv\metadata.csv")
covid_imgs.head(2)
```

Out[4]:

Unnamed: 0	file_name	image_data_grayscale	file_format	image_shape	label
0	0 COVID-1.png	[[185 40 0 ... 0 24 131]\n [75 14 0 ...	PNG	(299, 299)	COVID
1	1 COVID-10.png	[[135 133 131 ... 24 21 19]\n [132 130 129 ...	PNG	(299, 299)	COVID

In [5]:

```
# opacity data
opacity_images = pd.read_csv("C:\IE 7615\Project\Covid_detection_CNN-master\COVID-19_Radi  
ography_Dataset\metadata.csv\Lung_opacity metadata.csv")
opacity_images.head(2)
```

Out[5]:

Unnamed: 0	file_name	image_data_grayscale	file_format	image_shape	label
0	3616 Lung_Opacity-1.png	[[0 0 0 ... 0 0 0]\n [0 0 0 ...	PNG	(299, 299)	Lung_Opacity
1	3617 Lung_Opacity-10.png	[[12 13 13 ... 188 186 168]\n [13 14 14 ...	PNG	(299, 299)	Lung_Opacity

In [6]:

```
# normal data
```

```
normal_images = pd.read_csv("C:\\IE 7615\\Project\\Covid_detection_CNN-master\\COVID-19_Radiography_Dataset\\metadata.csv\\Normal_metadata.csv")
normal_images.head(2)
```

Out[6]:

Unnamed: 0		file_name	image_data_grayscale	file_format	image_shape	label
0	9628	Normal-1.png	[[0 0 0 ... 0 0 0]\n [0 0 0 ... 0 0 0]\n [0 0 ...	PNG	(299, 299)	Normal
1	9629	Normal-10.png	[[10 9 9 ... 10 9 5]\n [10 9 9 ...	PNG	(299, 299)	Normal

In [7]:

```
# pneumonia data
pneumonia_images = pd.read_csv("C:\\IE 7615\\Project\\Covid_detection_CNN-master\\COVID-19_Radiography_Dataset\\metadata.csv\\Viral_Pneumonia_metadata.csv")
pneumonia_images.head(2)
```

Out[7]:

Unnamed: 0		file_name	image_data_grayscale	file_format	image_shape	label
0	19820	Viral Pneumonia-1.png	[[1 1 2 ... 146 79 62]\n [1 1 2 ...	PNG	(299, 299)	Viral Pneumonia
1	19821	Viral Pneumonia-10.png	[[0 3 9 ... 86 89 87]\n [13 19 22 ... 91 92...	PNG	(299, 299)	Viral Pneumonia

images preprocessing

In [8]:

```
# Working with images
old_root_dir = "C:/IE 7615/Project/Covid_detection_CNN-master/COVID-19_Radiography_Dataset/"
imgs = ['COVID', 'Lung Opacity', 'Normal', 'Viral Pneumonia']
NEW_DIR = "C:/IE 7615/Project/all_image/"
```

In [9]:

```
if not os.path.exists(NEW_DIR):
    os.mkdir(NEW_DIR)

for i in imgs:
    org_dir = os.path.join(old_root_dir, i+"/")
    print(i)
    for imgfile in glob.iglob(os.path.join(org_dir, "*.png")):
        shutil.copy(imgfile, NEW_DIR)

else:
    print("Already Exist")
```

Already Exist

visualize the number of Images in each categories

In [10]:

```
# Visualize the number of images in each categories
counter = {'COVID':0, 'Lung Opacity':0, 'Normal':0, 'Viral Pneumonia':0}
for image in imgs:
    for count in glob.iglob(NEW_DIR+image+"*"):
        counter[image] += 1

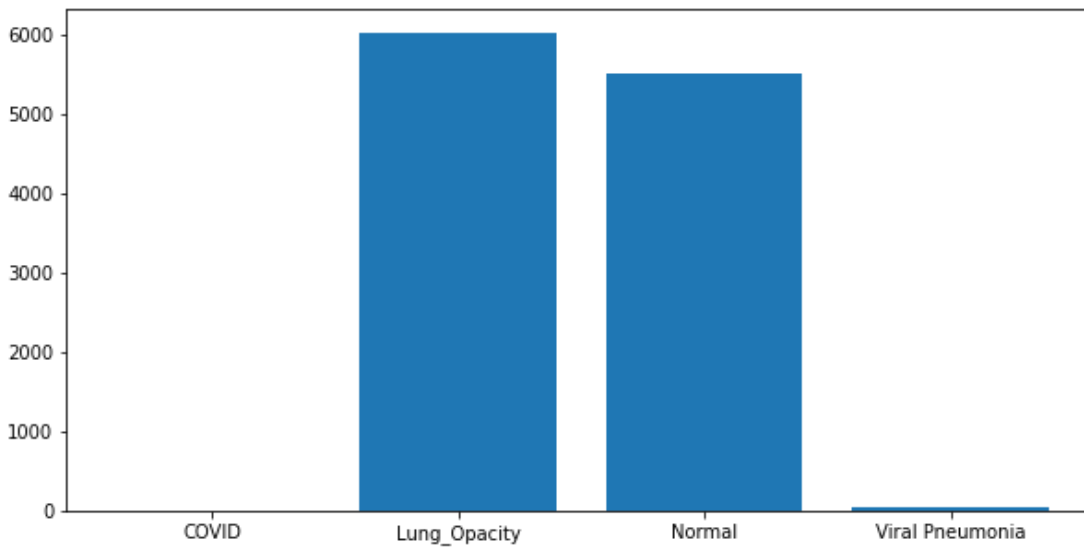
counter
```

Out[10]:

{'COVID': 8, 'Lung Opacity': 6012, 'Normal': 5492, 'Viral Pneumonia': 45}

In [11]:

```
plt.figure(figsize=(10,5))
plt.bar(x = counter.keys(), height= counter.values())
plt.show()
```



train test split

In [12]:

```
# create new folder storing covid and normal Radiography and split them into train and test and validation dataset
if not os.path.exists(NEW_DIR+"train_test_split/"):

    os.makedirs(NEW_DIR+"train_test_split/")

    os.makedirs(NEW_DIR+"train_test_split/train/Normal")
    os.makedirs(NEW_DIR+"train_test_split/train/Covid")

    os.makedirs(NEW_DIR+"train_test_split/test/Normal")
    os.makedirs(NEW_DIR+"train_test_split/test/Covid")

    os.makedirs(NEW_DIR+"train_test_split/validation/Normal")
    os.makedirs(NEW_DIR+"train_test_split/validation/Covid")

    # Train Data
    for i in np.random.choice(replace= False , size= 3000 , a = glob.glob(NEW_DIR+imgs[0]+"*")) ):
        shutil.copy(i , NEW_DIR+"train_test_split/train/Covid" )
        os.remove(i)

    for i in np.random.choice(replace= False , size= 3900 , a = glob.glob(NEW_DIR+imgs[2]+"*")) ):
        shutil.copy(i , NEW_DIR+"train_test_split/train/Normal" )
        os.remove(i)

    for i in np.random.choice(replace= False , size= 900 , a = glob.glob(NEW_DIR+imgs[3]+"*")) ):
        shutil.copy(i , NEW_DIR+"train_test_split/train/Covid" )
        os.remove(i)

    # Validation Data
    for i in np.random.choice(replace= False , size= 308 , a = glob.glob(NEW_DIR+imgs[0]+"*")) ):
        shutil.copy(i , NEW_DIR+"train_test_split/validation/Covid" )
        os.remove(i)

    for i in np.random.choice(replace= False , size= 500 , a = glob.glob(NEW_DIR+imgs[2]+"*")) ):
        shutil.copy(i , NEW_DIR+"train_test_split/validation/Normal" )
        os.remove(i)
```

```

*) ):
    shutil.copy(i , NEW_DIR+"train_test_split/validation/Normal" )
    os.remove(i)

    for i in np.random.choice(replace= False , size= 200 , a = glob.glob(NEW_DIR+imgs[3]+"
*) ):
        shutil.copy(i , NEW_DIR+"train_test_split/validation/Covid" )
        os.remove(i)

    # Test Data
    for i in np.random.choice(replace= False , size= 300 , a = glob.glob(NEW_DIR+imgs[0]+"
*) ):
        shutil.copy(i , NEW_DIR+"train_test_split/test/Covid" )
        os.remove(i)

    for i in np.random.choice(replace= False , size= 300 , a = glob.glob(NEW_DIR+imgs[2]+"
*) ):
        shutil.copy(i , NEW_DIR+"train_test_split/test/Normal" )
        os.remove(i)

    for i in np.random.choice(replace= False , size= 200 , a = glob.glob(NEW_DIR+imgs[3]+"
*) ):
        shutil.copy(i , NEW_DIR+"train_test_split/test/Covid" )
        os.remove(i)

```

In [16]:

```

train_path = "C:/IE 7615/Project/all_image/train_test_split/train"
valid_path = "C:/IE 7615/Project/all_image/train_test_split/validation"
test_path = "C:/IE 7615/Project/all_image/train_test_split/test"

```

In [27]:

```

from keras.preprocessing.image import ImageDataGenerator
from keras.applications import vgg16
from tensorflow.keras.applications import resnet
from tensorflow.keras.applications import inception_v3
from keras.models import Model
from keras.layers import Dense, MaxPool2D, Conv2D
import keras

```

In [18]:

```

# vgg train data
vgg_train_data_gen = ImageDataGenerator(preprocessing_function= vgg16.preprocess_input ,
rescale= 1./255, zoom_range= 0.2, horizontal_flip= True, shear_range= 0.2)
vgg_train = vgg_train_data_gen.flow_from_directory(directory= train_path , target_size=(2
24,224))

# vgg val data
vgg_validation_data_gen = ImageDataGenerator(preprocessing_function= vgg16.preprocess_inp
ut , rescale= 1./255 )
vgg_valid = vgg_validation_data_gen.flow_from_directory(directory= valid_path , target_si
ze=(224,224))

# vgg test data
vgg_test_data_gen = ImageDataGenerator(preprocessing_function= vgg16.preprocess_input ,
rescale= 1./255 )
vgg_test = vgg_test_data_gen.flow_from_directory(directory= test_path , target_size=(224,
224), shuffle= False)

```

Found 7800 images belonging to 2 classes.
Found 1008 images belonging to 2 classes.
Found 800 images belonging to 2 classes.

In [19]:

```

# resnet train data
resnet_train_data_gen = ImageDataGenerator(dtype = 'float32', preprocessing_function= res
net.preprocess_input)
resnet_train= resnet_train_data_gen.flow_from_directory(directory= train_path , target_s

```

```
ize=(224,224))
```

```
# resnet val data
```

```
resnet_validation_data_gen = ImageDataGenerator(dtype = 'float32', preprocessing_function  
= resnet.preprocess_input)  
resnet_valid = resnet_validation_data_gen.flow_from_directory(directory= valid_path , tar  
get_size=(224,224))
```

```
# resnet test data
```

```
resnet_test_data_gen = ImageDataGenerator(dtype = 'float32', preprocessing_function= resn  
et.preprocess_input)  
resnet_test = resnet_test_data_gen.flow_from_directory(directory= test_path , target_size  
=(224,224), shuffle= False)
```

Found 7800 images belonging to 2 classes.

Found 1008 images belonging to 2 classes.

Found 800 images belonging to 2 classes.

In [20]:

```
# Inception train data
```

```
Inc_train_data_gen = ImageDataGenerator(dtype = 'float32', preprocessing_function= incept  
ion_v3.preprocess_input)  
Inc_train = Inc_train_data_gen.flow_from_directory(directory= train_path , target_size=(2  
24,224))
```

```
# Inception val data
```

```
Inc_validation_data_gen = ImageDataGenerator(dtype = 'float32', preprocessing_function= i  
nception_v3.preprocess_input)  
Inc_valid = Inc_validation_data_gen.flow_from_directory(directory= valid_path , target_si  
ze=(224,224))
```

```
# Inception test data
```

```
Inc_test_data_gen = ImageDataGenerator(dtype = 'float32', preprocessing_function= incept  
ion_v3.preprocess_input)  
Inc_test = Inc_test_data_gen.flow_from_directory(directory= test_path , target_size=(224,  
224), shuffle= False)
```

Found 7800 images belonging to 2 classes.

Found 1008 images belonging to 2 classes.

Found 800 images belonging to 2 classes.

In [21]:

```
class_type = {0:'Covid', 1 : 'Normal'}
```

In [22]:

```
t_img , label = vgg_train.next()
```

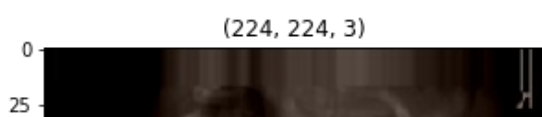
In [23]:

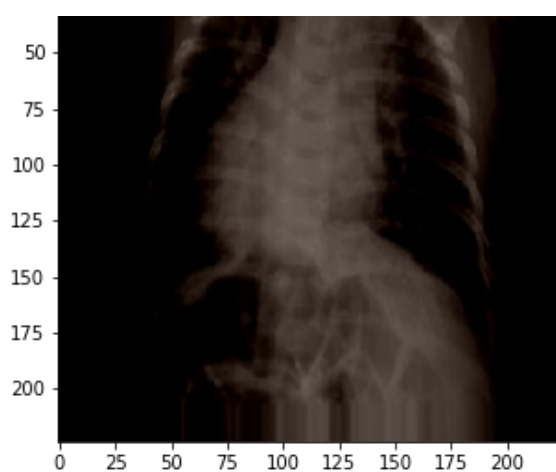
```
def plotImages(img_arr, label):  
  
    for im, l in zip(img_arr,label) :  
        plt.figure(figsize= (5,5))  
        plt.imshow(im, cmap = 'gray')  
        plt.title(im.shape)  
        plt.axis = False  
        plt.show()
```

In [24]:

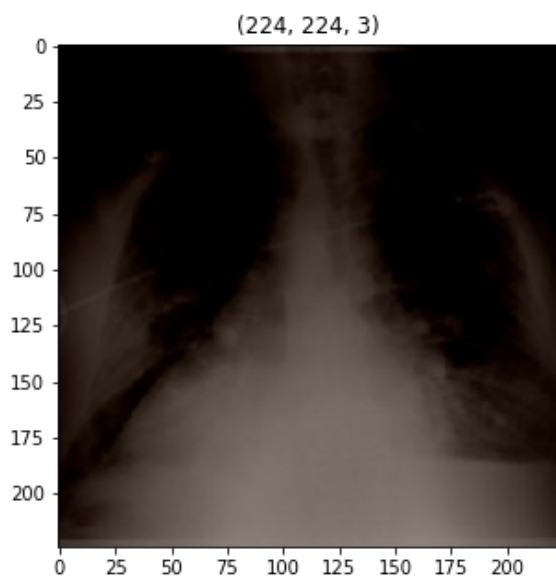
```
plotImages(t_img, label)
```

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0.
.255] for integers).

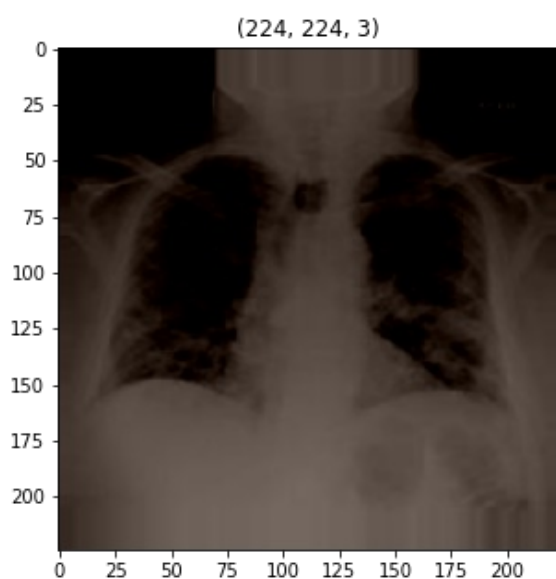




Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

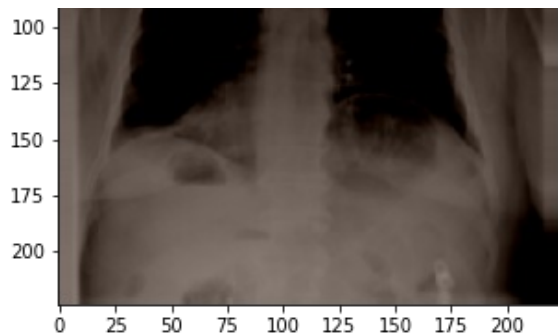


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

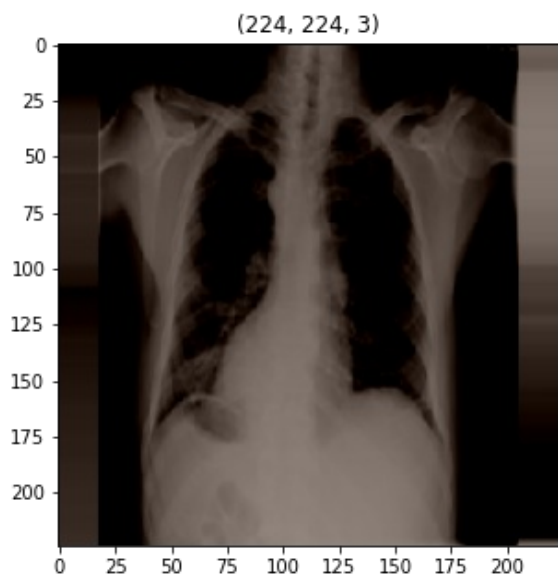


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

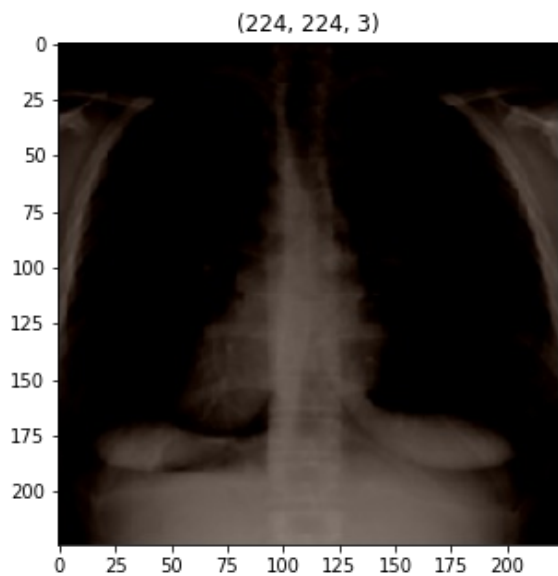




Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

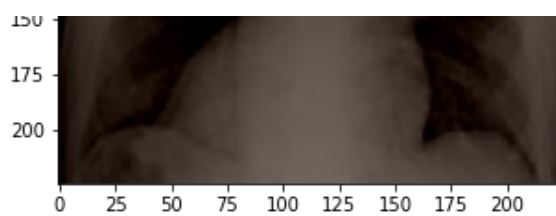


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

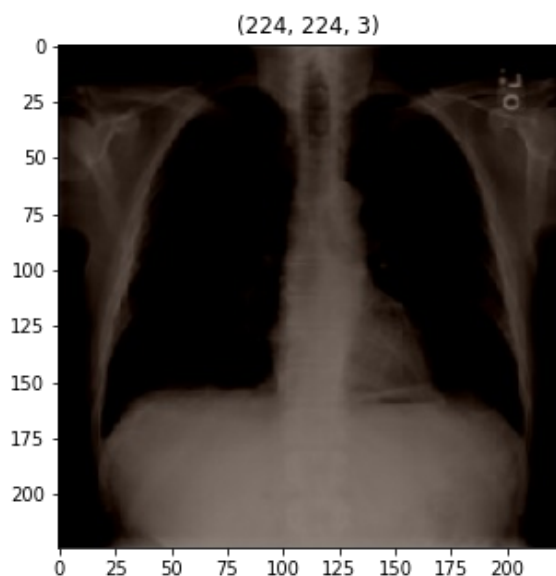


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

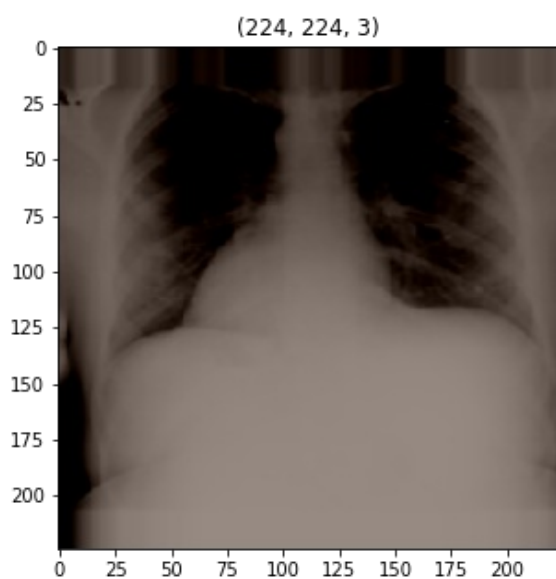




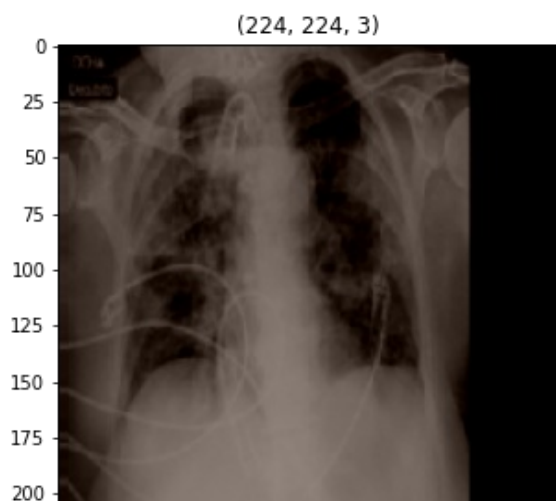
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

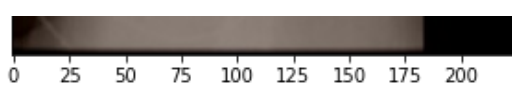


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

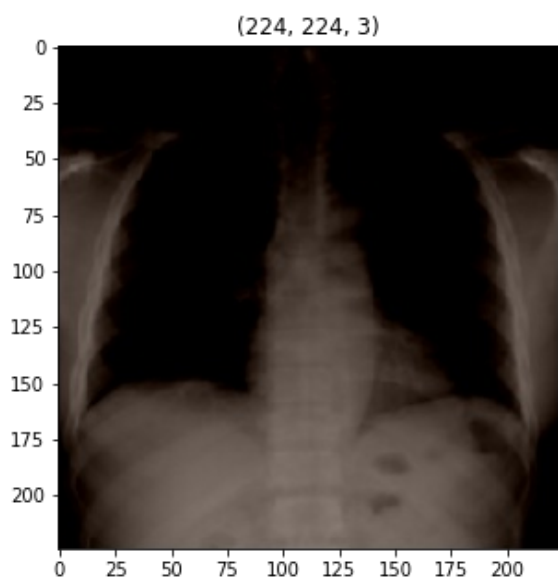


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

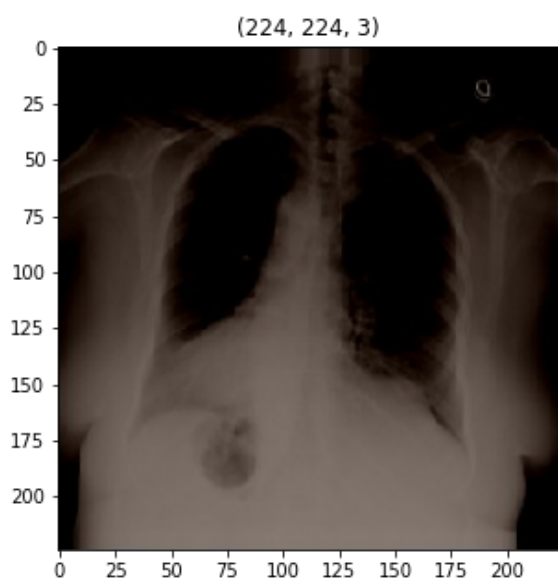




Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

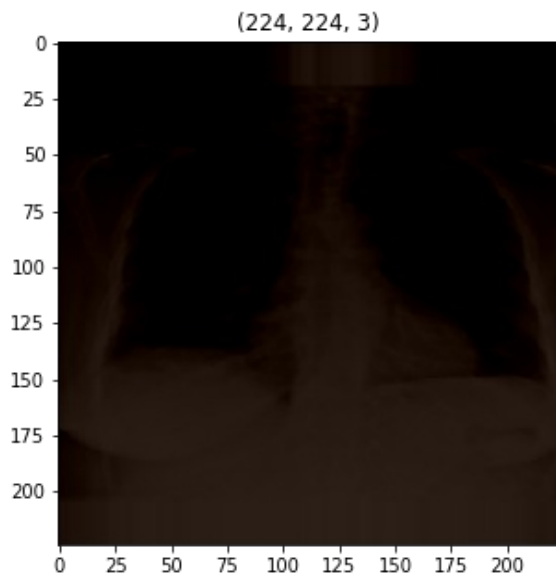


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

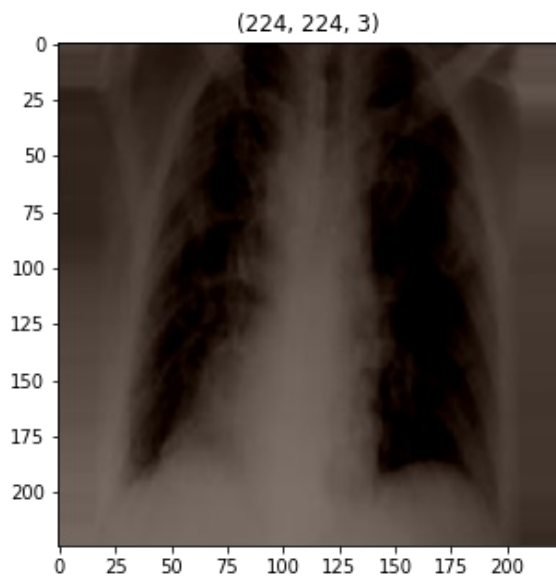


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

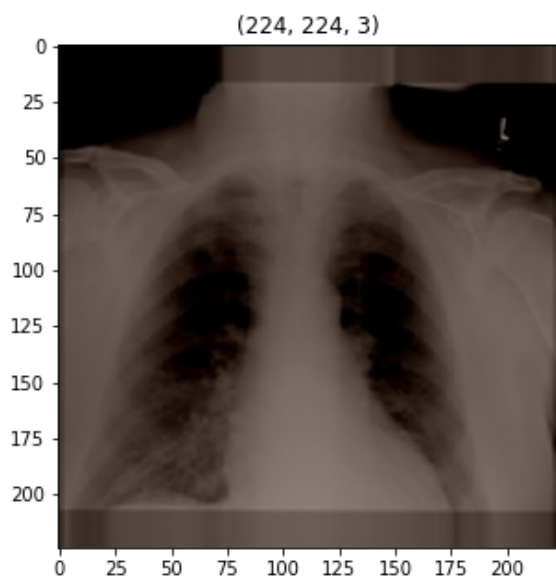
.255] for integers).



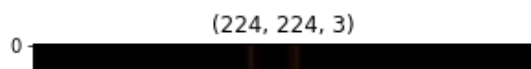
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

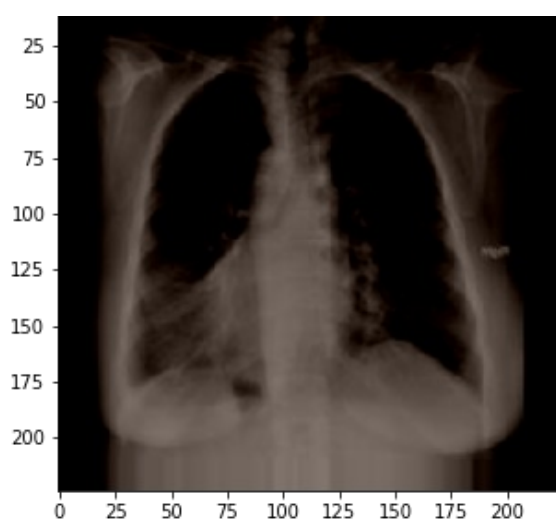


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

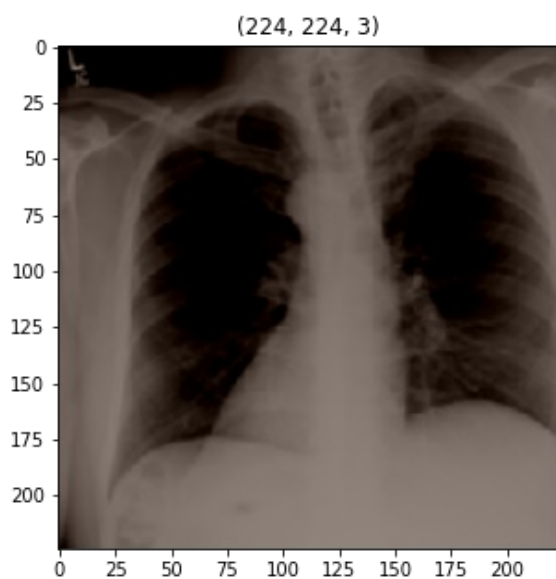




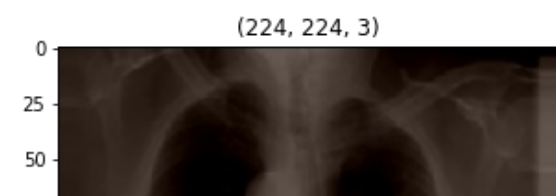
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

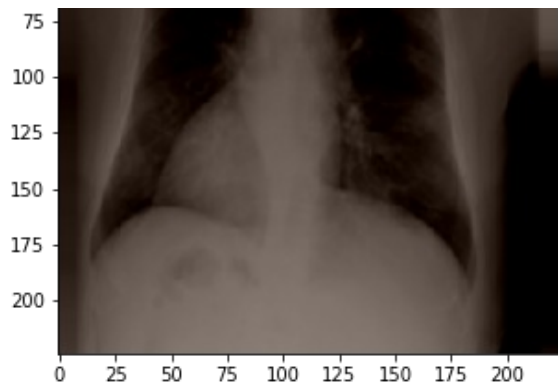


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).





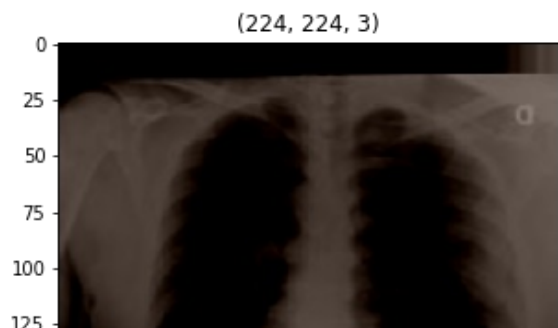
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

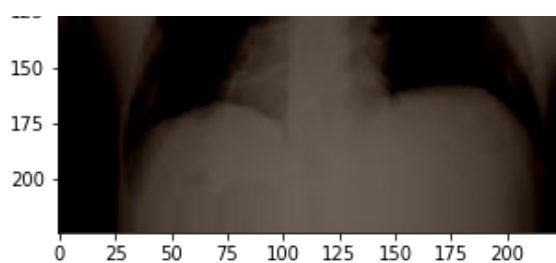


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

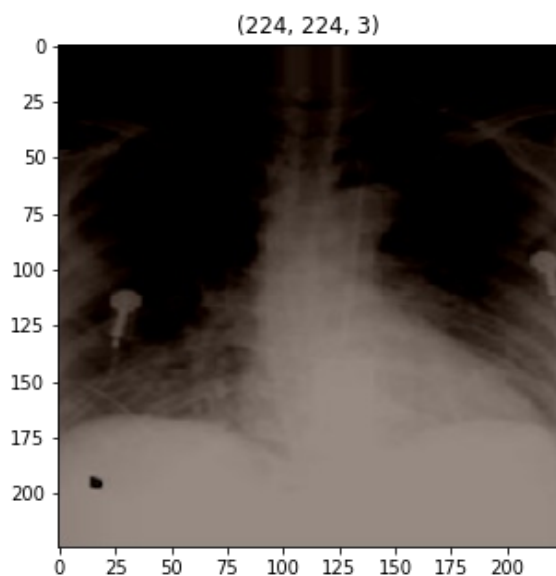


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

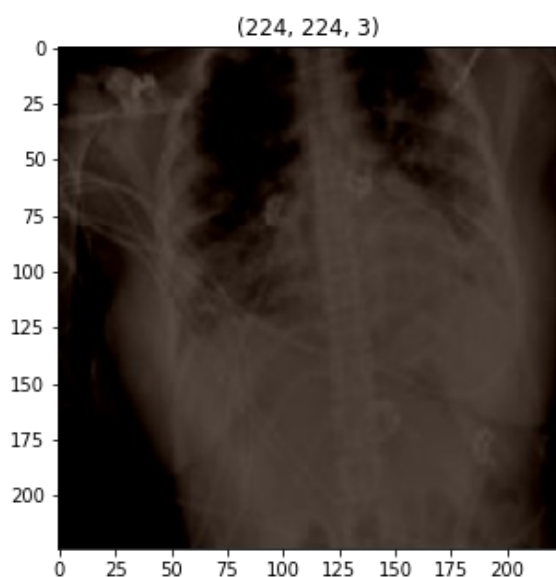




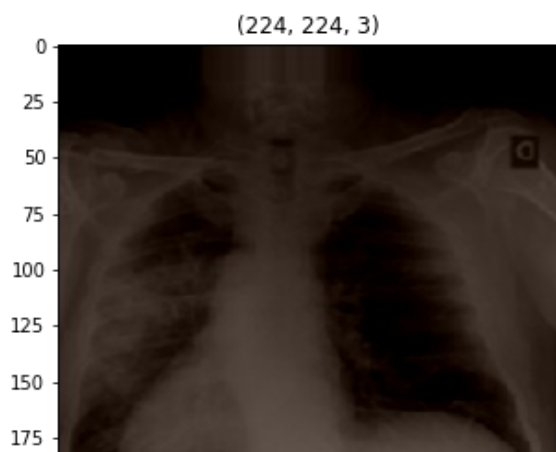
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

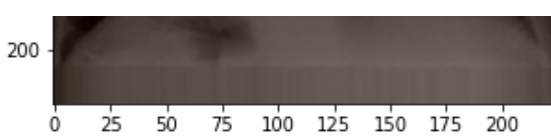


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

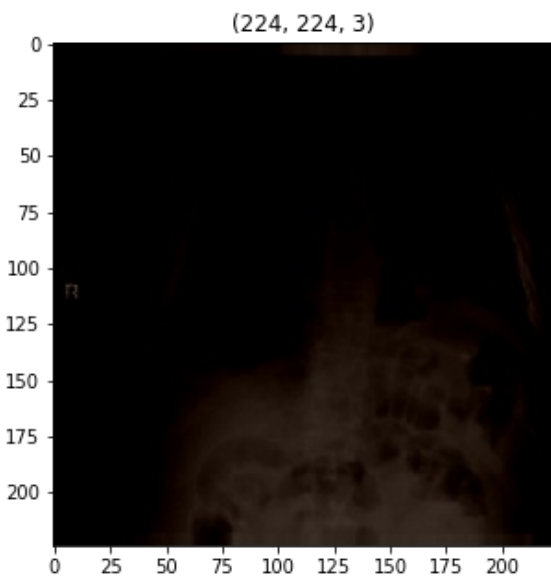


Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

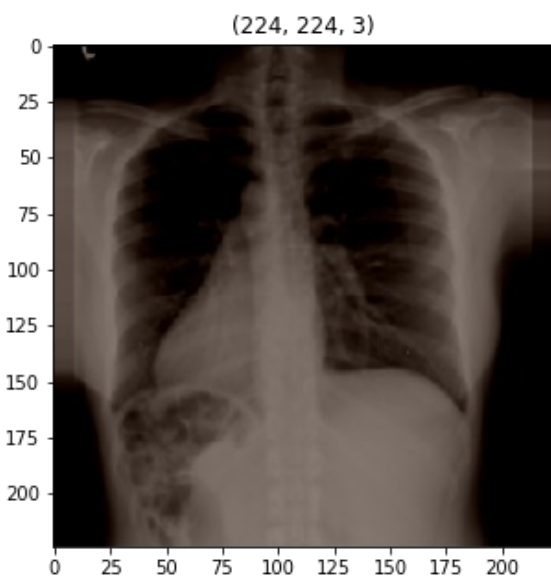




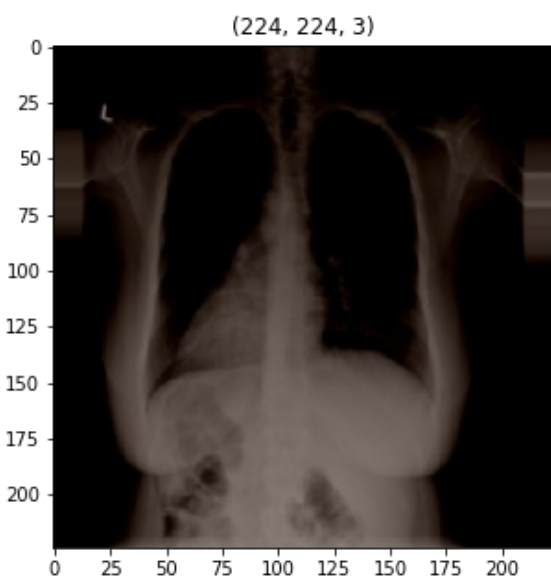
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



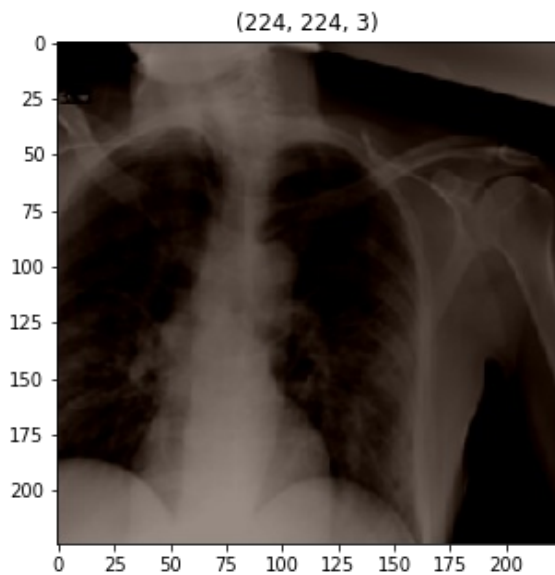
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



In [32]:

```
import keras
```

```

from keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.resnet50 import ResNet50
from keras.layers import Flatten , Dense, Dropout , MaxPool2D, GlobalAveragePooling2D, BatchNormalization
from tensorflow.keras.applications.inception_v3 import InceptionV3

```

In [33]:

```

# VGG16
vgg = VGG16( input_shape=(224,224,3), include_top= False) # include_top will consider the new weights
# resnet
resnet = ResNet50(weights="imagenet", include_top=False, input_shape=(224,224,3))
# Inception
Inception = InceptionV3(weights='imagenet', include_top=False, input_shape=(224,224,3))

```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87916544/87910968 [=====] - 1s 0us/step
87924736/87910968 [=====] - 1s 0us/step

In [34]:

```

for layer in vgg.layers:          # Dont Train the parameters again
    layer.trainable = False
for layer in resnet.layers:      # Dont Train the parameters again
    layer.trainable = False
for layer in Inception.layers:  # Dont Train the parameters again
    layer.trainable = False

```

In [35]:

```

# VGG16
x = Flatten()(vgg.output)
#x = Dropout(rate=.25)(x)
#x = Dense(units=64, activation='relu')(x)
#x = Dropout(rate=.25)(x)
x = Dense(units=2 , activation='sigmoid', name = 'predictions' )(x)

vgg_model = Model(vgg.input, x)

```

In [36]:

```

# resnet
x = resnet.output
x = Flatten()(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(units = 2, activation="sigmoid", name = 'predictions')(x)

resnet_model = Model(inputs = resnet.input, outputs = predictions)

```

In [37]:

```

# Inception
x = Inception.output
x = GlobalAveragePooling2D()(x)
x = Dense(256, activation='relu')(x)
predictions = Dense(2, activation='sigmoid', name = 'predictions')(x)

Inc_model = Model(inputs=Inception.input, outputs=predictions)

```

In [38]:

```

# VGG16 summary
vgg_model.summary()

```

Model: "model"

Layer (type)	Output Shape	Param #
=====		

input_9 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
predictions (Dense)	(None, 2)	50178

```

=====
Total params: 14,764,866
Trainable params: 50,178
Non-trainable params: 14,714,688

```

In [39]:

```

# resnet summary
resnet_model.summary()

```

Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
input_10 (InputLayer)	[(None, 224, 224, 3)]	0	[]
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	['input_10[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 [0]']
ization) (None, 56, 56, 64) 256 ['conv2_block1_1_conv[0]

conv2_block1_1_relu (Activatio [0]']
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 [0]']
ization) (None, 56, 56, 64) 256 ['conv2_block1_2_conv[0]

conv2_block1_2_relu (Activatio [0]']
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 [0]']
ization) (None, 56, 56, 256) 1024 ['conv2_block1_0_conv[0]

```

conv2_block1_3_bn (BatchNormal [0]'] ization)	(None, 56, 56, 256)	1024	['conv2_block1_3_conv[0]
conv2_block1_add (Add) 0]'],	(None, 56, 56, 256)	0	['conv2_block1_0_bn[0] 'conv2_block1_3_bn[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 [0]'] ization)	(None, 56, 56, 64)	256	['conv2_block2_1_conv[0]
conv2_block2_1_relu (Activatio]'] n)	(None, 56, 56, 64)	0	['conv2_block2_1_bn[0][0]
conv2_block2_2_conv (Conv2D) [0]']	(None, 56, 56, 64)	36928	['conv2_block2_1_relu[0]
conv2_block2_2_bn (BatchNormal [0]'] ization)	(None, 56, 56, 64)	256	['conv2_block2_2_conv[0]
conv2_block2_2_relu (Activatio]'] n)	(None, 56, 56, 64)	0	['conv2_block2_2_bn[0][0]
conv2_block2_3_conv (Conv2D) [0]']	(None, 56, 56, 256)	16640	['conv2_block2_2_relu[0]
conv2_block2_3_bn (BatchNormal [0]'] ization)	(None, 56, 56, 256)	1024	['conv2_block2_3_conv[0]
conv2_block2_add (Add)]'],	(None, 56, 56, 256)	0	['conv2_block1_out[0][0] 'conv2_block2_3_bn[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] ']
conv2_block3_2_conv (Conv2D)	(None, 56, 56, 64)	36928	['conv2_block3_1_relu[0] [0]']
conv2_block3_2_bn (BatchNormal [0]'] ization)	(None, 56, 56, 64)	256	['conv2_block3_2_conv[0] [0]']
conv2_block3_2_relu (Activatio n)	(None, 56, 56, 64)	0	['conv2_block3_2_bn[0][0] ']
conv2_block3_3_conv (Conv2D)	(None, 56, 56, 256)	16640	['conv2_block3_2_relu[0] [0]']
conv2_block3_3_bn (BatchNormal [0]'] ization)	(None, 56, 56, 256)	1024	['conv2_block3_3_conv[0] [0]']
conv2_block3_add (Add) '],' [0]']	(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 [0]'] ization)	(None, 28, 28, 128)	512	['conv3_block1_1_conv[0] [0]']
conv3_block1_1_relu (Activatio n)	(None, 28, 28, 128)	0	['conv3_block1_1_bn[0][0] ']
conv3_block1_2_conv (Conv2D) [0]']	(None, 28, 28, 128)	147584	['conv3_block1_1_relu[0] [0]']
conv3_block1_2_bn (BatchNormal [0]'] ization)	(None, 28, 28, 128)	512	['conv3_block1_2_conv[0] [0]']

conv3_block1_2_relu (Activation)	(None, 28, 28, 128)	0	['conv3_block1_2_bn[0][0]
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 (BatchNormalization)	(None, 28, 28, 512)	2048	['conv3_block1_0_conv[0][0]']
conv3_block1_3_bn (BatchNormalization)	(None, 28, 28, 512)	2048	['conv3_block1_3_conv[0][0]']
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 (BatchNormalization)	(None, 28, 28, 128)	512	['conv3_block2_1_conv[0][0]']
conv3_block2_1_relu (Activation)	(None, 28, 28, 128)	0	['conv3_block2_1_bn[0][0]']
conv3_block2_2_conv (Conv2D)	(None, 28, 28, 128)	147584	['conv3_block2_1_relu[0][0]']
conv3_block2_2_bn (BatchNormalization)	(None, 28, 28, 128)	512	['conv3_block2_2_conv[0][0]']
conv3_block2_2_relu (Activation)	(None, 28, 28, 128)	0	['conv3_block2_2_bn[0][0]']
conv3_block2_3_conv (Conv2D)	(None, 28, 28, 512)	66048	['conv3_block2_2_relu[0][0]']

conv3_block2_3_bn (BatchNormal [0]'] ization)	(None, 28, 28, 512)	2048	['conv3_block2_3_conv[0]
conv3_block2_add (Add)]', [0]']	(None, 28, 28, 512)	0	['conv3_block1_out[0][0] 'conv3_block2_3_bn[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 [0]'] ization)	(None, 28, 28, 128)	512	['conv3_block3_1_conv[0]
conv3_block3_1_relu (Activatio]'] n)	(None, 28, 28, 128)	0	['conv3_block3_1_bn[0][0]
conv3_block3_2_conv (Conv2D) [0]']	(None, 28, 28, 128)	147584	['conv3_block3_1_relu[0]
conv3_block3_2_bn (BatchNormal [0]'] ization)	(None, 28, 28, 128)	512	['conv3_block3_2_conv[0]
conv3_block3_2_relu (Activatio]'] n)	(None, 28, 28, 128)	0	['conv3_block3_2_bn[0][0]
conv3_block3_3_conv (Conv2D) [0]']	(None, 28, 28, 512)	66048	['conv3_block3_2_relu[0]
conv3_block3_3_bn (BatchNormal [0]'] ization)	(None, 28, 28, 512)	2048	['conv3_block3_3_conv[0]
conv3_block3_add (Add)]', [0]']	(None, 28, 28, 512)	0	['conv3_block2_out[0][0] 'conv3_block3_3_bn[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 [0]'] ization)	(None, 28, 28, 128)	512	['conv3_block4_1_conv[0]
conv3_block4_1_relu (Activatio]'] n)	(None, 28, 28, 128)	0	['conv3_block4_1_bn[0][0]
conv3_block4_2_conv (Conv2D)	(None, 28, 28, 128)	147584	['conv3_block4_1_relu[0]
conv3_block4_2_bn (BatchNormal [0]'] ization)	(None, 28, 28, 128)	512	['conv3_block4_2_conv[0]
conv3_block4_2_relu (Activatio]'] n)	(None, 28, 28, 128)	0	['conv3_block4_2_bn[0][0]
conv3_block4_3_conv (Conv2D)	(None, 28, 28, 512)	66048	['conv3_block4_2_relu[0]
conv3_block4_3_bn (BatchNormal [0]'] ization)	(None, 28, 28, 512)	2048	['conv3_block4_3_conv[0]
conv3_block4_add (Add)]', [0]']	(None, 28, 28, 512)	0	['conv3_block3_out[0][0] 'conv3_block4_3_bn[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 [0]'] ization)	(None, 14, 14, 256)	1024	['conv4_block1_1_conv[0]
conv4_block1_1_relu (Activatio]'] n)	(None, 14, 14, 256)	0	['conv4_block1_1_bn[0][0]
conv4_block1_2_conv (Conv2D)	(None, 14, 14, 256)	590080	['conv4_block1_1_relu[0]
conv4_block1_2_bn (BatchNormal [0]']	(None, 14, 14, 256)	1024	['conv4_block1_2_conv[0]

```

ization)

conv4_block1_2_relu (Activation) (None, 14, 14, 256) 0 ['conv4_block1_2_bn[0][0]']
n)

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 ['conv4_block1_0_conv[0][0]']
ization)
)

conv4_block1_3_bn (BatchNormal (None, 14, 14, 1024) 4096 ['conv4_block1_3_conv[0][0]']
ization)
)

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 ['conv4_block2_1_conv[0][0]']
ization)

conv4_block2_1_relu (Activation) (None, 14, 14, 256) 0 ['conv4_block2_1_bn[0][0]']
n)

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 ['conv4_block2_2_conv[0][0]']
ization)

```



```

conv4_block2_2_relu (Activation) (None, 14, 14, 256) 0 ['conv4_block2_2_bn[0][0]']
n)

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 ['conv4_block2_3_conv[0][0]']
ization)
)

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 ['conv4_block3_1_conv[0][0]']
ization)

conv4_block3_1_relu (Activation) (None, 14, 14, 256) 0 ['conv4_block3_1_bn[0][0]']
n)

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 ['conv4_block3_2_conv[0][0]']
ization)

conv4_block3_2_relu (Activation) (None, 14, 14, 256) 0 ['conv4_block3_2_bn[0][0]']
n)

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 ['conv4_block3_3_conv[0][0]']
ization)
)

```

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	['conv4_block4_1_conv[0]
[0]']		
ization)		
conv4_block4_1_relu (Activatio	(None, 14, 14, 256) 0	['conv4_block4_1_bn[0][0
']		
n)		
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	['conv4_block4_2_conv[0]
[0]']		
ization)		
conv4_block4_2_relu (Activatio	(None, 14, 14, 256) 0	['conv4_block4_2_bn[0][0
']		
n)		
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	['conv4_block4_3_conv[0]
[0]'])	
ization)		
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 [0]'] ization)	(None, 14, 14, 256)	1024	['conv4_block5_1_conv[0]
conv4_block5_1_relu (Activatio]'] n)	(None, 14, 14, 256)	0	['conv4_block5_1_bn[0][0]
conv4_block5_2_conv (Conv2D)	(None, 14, 14, 256)	590080	['conv4_block5_1_relu[0]
conv4_block5_2_bn (BatchNormal [0]'] ization)	(None, 14, 14, 256)	1024	['conv4_block5_2_conv[0]
conv4_block5_2_relu (Activatio]'] n)	(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]
)		
conv4_block5_3_bn (BatchNormal [0]'] ization)	(None, 14, 14, 1024	4096	['conv4_block5_3_conv[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 [0]'] ization)	(None, 14, 14, 256)	1024	['conv4_block6_1_conv[0]
conv4_block6_1_relu (Activatio]'] n)	(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 [0]'] ization)	(None, 14, 14, 256)	1024	['conv4_block6_2_conv[0]
conv4_block6_2_relu (Activatio]'] n)	(None, 14, 14, 256)	0	['conv4_block6_2_bn[0][0]
conv4_block6_3_conv (Conv2D) [0]'])	(None, 14, 14, 1024	263168	['conv4_block6_2_relu[0]
conv4_block6_3_bn (BatchNormal [0]'] ization)	(None, 14, 14, 1024	4096	['conv4_block6_3_conv[0]
conv4_block6_add (Add)]', [0]'])	(None, 14, 14, 1024	0	['conv4_block5_out[0][0] 'conv4_block6_3_bn[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 [0]'] ization)	(None, 7, 7, 512)	2048	['conv5_block1_1_conv[0]
conv5_block1_1_relu (Activatio]'] n)	(None, 7, 7, 512)	0	['conv5_block1_1_bn[0][0]
conv5_block1_2_conv (Conv2D) [0]']	(None, 7, 7, 512)	2359808	['conv5_block1_1_relu[0]
conv5_block1_2_bn (BatchNormal [0]'] ization)	(None, 7, 7, 512)	2048	['conv5_block1_2_conv[0]
conv5_block1_2_relu (Activatio]'] n)	(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 ization)	(None, 7, 7, 2048)	8192	['conv5_block1_0_conv[0][0]']
conv5_block1_3_bn (BatchNormal ization)	(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 ization)	(None, 7, 7, 512)	2048	['conv5_block2_1_conv[0][0]']
conv5_block2_1_relu (Activatio n)	(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]']

[0]']				'conv5_block2_3_bn[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	(None, 7, 7, 512)	2048		['conv5_block3_1_conv[0]
[0]']				
ization)				
conv5_block3_1_relu (Activatio	(None, 7, 7, 512)	0		['conv5_block3_1_bn[0][0
']				
n)				
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 512)	2359808		['conv5_block3_1_relu[0]
[0]']				
conv5_block3_2_bn (BatchNormal	(None, 7, 7, 512)	2048		['conv5_block3_2_conv[0]
[0]']				
ization)				
conv5_block3_2_relu (Activatio	(None, 7, 7, 512)	0		['conv5_block3_2_bn[0][0
']				
n)				
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624		['conv5_block3_2_relu[0]
[0]']				
conv5_block3_3_bn (BatchNormal	(None, 7, 7, 2048)	8192		['conv5_block3_3_conv[0]
[0]']				
ization)				
conv5_block3_add (Add)	(None, 7, 7, 2048)	0		['conv5_block2_out[0][0
'],				
[0]']				'conv5_block3_3_bn[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
']				
dense (Dense)	(None, 256)	25690368		['flatten_1[0][0]']
dropout (Dropout)	(None, 256)	0		['dense[0][0]']

predictions (Dense) (None, 2) 514 ['dropout[0][0]']

=====
=====
Total params: 49,278,594
Trainable params: 25,690,882
Non-trainable params: 23,587,712

In [40]:

```
# Inception summary  
Inc_model.summary()
```

Model: "model_2"

Layer (type)	Output Shape	Param #	Connected to
=====			
input_11 (InputLayer)	[(None, 224, 224, 3)]	0	[]
conv2d (Conv2D)	(None, 111, 111, 32)	864	['input_11[0][0]']
batch_normalization (BatchNorm alization)	(None, 111, 111, 32)	96	['conv2d[0][0]']
activation (Activation) [0]']	(None, 111, 111, 32)	0	['batch_normalization[0] [0]']
conv2d_1 (Conv2D)	(None, 109, 109, 32)	9216	['activation[0][0]']
batch_normalization_1 (BatchNo rmalization)	(None, 109, 109, 32)	96	['conv2d_1[0][0]']
activation_1 (Activation) 0][0]']	(None, 109, 109, 32)	0	['batch_normalization_1[0][0]']
conv2d_2 (Conv2D)	(None, 109, 109, 64)	18432	['activation_1[0][0]']

batch_normalization_2 (BatchNormalization)	(None, 109, 109, 64)	192	['conv2d_2[0][0]']
activation_2 (Activation)	(None, 109, 109, 64)	0	['batch_normalization_2[0][0]']
max_pooling2d (MaxPooling2D)	(None, 54, 54, 64)	0	['activation_2[0][0]']
conv2d_3 (Conv2D)	(None, 54, 54, 80)	5120	['max_pooling2d[0][0]']
batch_normalization_3 (BatchNormalization)	(None, 54, 54, 80)	240	['conv2d_3[0][0]']
activation_3 (Activation)	(None, 54, 54, 80)	0	['batch_normalization_3[0][0]']
conv2d_4 (Conv2D)	(None, 52, 52, 192)	138240	['activation_3[0][0]']
batch_normalization_4 (BatchNormalization)	(None, 52, 52, 192)	576	['conv2d_4[0][0]']
activation_4 (Activation)	(None, 52, 52, 192)	0	['batch_normalization_4[0][0]']
max_pooling2d_1 (MaxPooling2D)	(None, 25, 25, 192)	0	['activation_4[0][0]']
conv2d_8 (Conv2D)	(None, 25, 25, 64)	12288	['max_pooling2d_1[0][0]']
batch_normalization_8 (BatchNormalization)	(None, 25, 25, 64)	192	['conv2d_8[0][0]']
activation_8 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_8[0][0]']
conv2d_6 (Conv2D)	(None, 25, 25, 48)	9216	['max_pooling2d_1[0][0]']
conv2d_9 (Conv2D)	(None, 25, 25, 96)	55296	['activation_8[0][0]']
batch_normalization_6 (BatchNormalization)	(None, 25, 25, 48)	144	['conv2d_6[0][0]']

rmalization)				
batch_normalization_9 (BatchNo rmalization)	(None, 25, 25, 96)	288	['conv2d_9[0][0]']	
activation_6 (Activation) 0][0]']	(None, 25, 25, 48)	0	['batch_normalization_6[0][0]']	
activation_9 (Activation) 0][0]']	(None, 25, 25, 96)	0	['batch_normalization_9[0][0]']	
average_pooling2d (AveragePool ing2D)	(None, 25, 25, 192)	0	['max_pooling2d_1[0][0]']	
conv2d_5 (Conv2D)]	(None, 25, 25, 64)	12288	['max_pooling2d_1[0][0]']	
conv2d_7 (Conv2D)	(None, 25, 25, 64)	76800	['activation_6[0][0]']	
conv2d_10 (Conv2D)	(None, 25, 25, 96)	82944	['activation_9[0][0]']	
conv2d_11 (Conv2D)]']	(None, 25, 25, 32)	6144	['average_pooling2d[0][0]']	
batch_normalization_5 (BatchNo rmalization)	(None, 25, 25, 64)	192	['conv2d_5[0][0]']	
batch_normalization_7 (BatchNo rmalization)	(None, 25, 25, 64)	192	['conv2d_7[0][0]']	
batch_normalization_10 (BatchN ormalization)	(None, 25, 25, 96)	288	['conv2d_10[0][0]']	
batch_normalization_11 (BatchN ormalization)	(None, 25, 25, 32)	96	['conv2d_11[0][0]']	
activation_5 (Activation) 0][0]']	(None, 25, 25, 64)	0	['batch_normalization_5[0][0]']	
activation_7 (Activation) 0][0]']	(None, 25, 25, 64)	0	['batch_normalization_7[0][0]']	

activation_10 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_10[0][0]']
activation_11 (Activation)	(None, 25, 25, 32)	0	['batch_normalization_11[0][0]']
mixed0 (Concatenate)	(None, 25, 25, 256)	0	['activation_5[0][0]', 'activation_7[0][0]', 'activation_10[0][0]', 'activation_11[0][0]']
conv2d_15 (Conv2D)	(None, 25, 25, 64)	16384	['mixed0[0][0]']
batch_normalization_15 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_15[0][0]']
activation_15 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_15[0][0]']
conv2d_13 (Conv2D)	(None, 25, 25, 48)	12288	['mixed0[0][0]']
conv2d_16 (Conv2D)	(None, 25, 25, 96)	55296	['activation_15[0][0]']
batch_normalization_13 (Batch Normalization)	(None, 25, 25, 48)	144	['conv2d_13[0][0]']
batch_normalization_16 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_16[0][0]']
activation_13 (Activation)	(None, 25, 25, 48)	0	['batch_normalization_13[0][0]']
activation_16 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_16[0][0]']
average_pooling2d_1 (Average Pooling2D)	(None, 25, 25, 256)	0	['mixed0[0][0]']
conv2d_12 (Conv2D)	(None, 25, 25, 64)	16384	['mixed0[0][0]']
conv2d_14 (Conv2D)	(None, 25, 25, 64)	76800	['activation_13[0][0]']

conv2d_17 (Conv2D)	(None, 25, 25, 96)	82944	['activation_16[0][0]']
conv2d_18 (Conv2D)	(None, 25, 25, 64)	16384	['average_pooling2d_1[0][0]']
batch_normalization_12 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_12[0][0]']
batch_normalization_14 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_14[0][0]']
batch_normalization_17 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_17[0][0]']
batch_normalization_18 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_18[0][0]']
activation_12 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_12[0][0]']
activation_14 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_14[0][0]']
activation_17 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_17[0][0]']
activation_18 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_18[0][0]']
mixed1 (Concatenate)	(None, 25, 25, 288)	0	['activation_12[0][0]', 'activation_14[0][0]', 'activation_17[0][0]', 'activation_18[0][0]']
conv2d_22 (Conv2D)	(None, 25, 25, 64)	18432	['mixed1[0][0]']
batch_normalization_22 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_22[0][0]']
activation_22 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_22[0][0]']

conv2d_20 (Conv2D)	(None, 25, 25, 48)	13824	['mixed1[0][0]']
conv2d_23 (Conv2D)	(None, 25, 25, 96)	55296	['activation_22[0][0]']
batch_normalization_20 (Batch Normalization)	(None, 25, 25, 48)	144	['conv2d_20[0][0]']
batch_normalization_23 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_23[0][0]']
activation_20 (Activation)	(None, 25, 25, 48)	0	['batch_normalization_20[0][0]']
activation_23 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_23[0][0]']
average_pooling2d_2 (Average Pooling2D)	(None, 25, 25, 288)	0	['mixed1[0][0]']
conv2d_19 (Conv2D)	(None, 25, 25, 64)	18432	['mixed1[0][0]']
conv2d_21 (Conv2D)	(None, 25, 25, 64)	76800	['activation_20[0][0]']
conv2d_24 (Conv2D)	(None, 25, 25, 96)	82944	['activation_23[0][0]']
conv2d_25 (Conv2D)	(None, 25, 25, 64)	18432	['average_pooling2d_2[0][0]']
batch_normalization_19 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_19[0][0]']
batch_normalization_21 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_21[0][0]']
batch_normalization_24 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_24[0][0]']
batch_normalization_25 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_25[0][0]']

ormalization)				
activation_19 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_19[0][0]']	
activation_21 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_21[0][0]']	
activation_24 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_24[0][0]']	
activation_25 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_25[0][0]']	
mixed2 (Concatenate)	(None, 25, 25, 288)	0	['activation_19[0][0]', 'activation_21[0][0]', 'activation_24[0][0]', 'activation_25[0][0]']	
conv2d_27 (Conv2D)	(None, 25, 25, 64)	18432	['mixed2[0][0]']	
batch_normalization_27 (BatchN	(None, 25, 25, 64)	192	['conv2d_27[0][0]']	
ormalization)				
activation_27 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_27[0][0]']	
conv2d_28 (Conv2D)	(None, 25, 25, 96)	55296	['activation_27[0][0]']	
batch_normalization_28 (BatchN	(None, 25, 25, 96)	288	['conv2d_28[0][0]']	
ormalization)				
activation_28 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_28[0][0]']	
conv2d_26 (Conv2D)	(None, 12, 12, 384)	995328	['mixed2[0][0]']	
conv2d_29 (Conv2D)	(None, 12, 12, 96)	82944	['activation_28[0][0]']	
batch_normalization_26 (BatchN	(None, 12, 12, 384)	1152	['conv2d_26[0][0]']	
ormalization)				

batch_normalization_29 (Batch Normalization)	(None, 12, 12, 96)	288	['conv2d_29[0][0]']
activation_26 (Activation)	(None, 12, 12, 384)	0	['batch_normalization_26[0][0]']
activation_29 (Activation)	(None, 12, 12, 96)	0	['batch_normalization_29[0][0]']
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 288)	0	['mixed2[0][0]']
mixed3 (Concatenate)	(None, 12, 12, 768)	0	['activation_26[0][0]', , 'activation_29[0][0]', 'max_pooling2d_2[0][0]']
conv2d_34 (Conv2D)	(None, 12, 12, 128)	98304	['mixed3[0][0]']
batch_normalization_34 (Batch Normalization)	(None, 12, 12, 128)	384	['conv2d_34[0][0]']
activation_34 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_34[0][0]']
conv2d_35 (Conv2D)	(None, 12, 12, 128)	114688	['activation_34[0][0]']
batch_normalization_35 (Batch Normalization)	(None, 12, 12, 128)	384	['conv2d_35[0][0]']
activation_35 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_35[0][0]']
conv2d_31 (Conv2D)	(None, 12, 12, 128)	98304	['mixed3[0][0]']
conv2d_36 (Conv2D)	(None, 12, 12, 128)	114688	['activation_35[0][0]']
batch_normalization_31 (Batch Normalization)	(None, 12, 12, 128)	384	['conv2d_31[0][0]']
batch_normalization_36 (Batch Normalization)	(None, 12, 12, 128)	384	['conv2d_36[0][0]']

activation_31 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_31[0][0]']
activation_36 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_36[0][0]']
conv2d_32 (Conv2D)	(None, 12, 12, 128)	114688	['activation_31[0][0]']
conv2d_37 (Conv2D)	(None, 12, 12, 128)	114688	['activation_36[0][0]']
batch_normalization_32 (Batch Normalization)	(None, 12, 12, 128)	384	['conv2d_32[0][0]']
batch_normalization_37 (Batch Normalization)	(None, 12, 12, 128)	384	['conv2d_37[0][0]']
activation_32 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_32[0][0]']
activation_37 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_37[0][0]']
average_pooling2d_3 (Average Pooling2D)	(None, 12, 12, 768)	0	['mixed3[0][0]']
conv2d_30 (Conv2D)	(None, 12, 12, 192)	147456	['mixed3[0][0]']
conv2d_33 (Conv2D)	(None, 12, 12, 192)	172032	['activation_32[0][0]']
conv2d_38 (Conv2D)	(None, 12, 12, 192)	172032	['activation_37[0][0]']
conv2d_39 (Conv2D)	(None, 12, 12, 192)	147456	['average_pooling2d_3[0][0]']
batch_normalization_30 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_30[0][0]']
batch_normalization_33 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_33[0][0]']

batch_normalization_38 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_38[0][0]']
batch_normalization_39 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_39[0][0]']
activation_30 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_30[0][0]']
activation_33 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_33[0][0]']
activation_38 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_38[0][0]']
activation_39 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_39[0][0]']
mixed4 (Concatenate)	(None, 12, 12, 768)	0	['activation_30[0][0]', 'activation_33[0][0]', 'activation_38[0][0]', 'activation_39[0][0]']
conv2d_44 (Conv2D)	(None, 12, 12, 160)	122880	['mixed4[0][0]']
batch_normalization_44 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_44[0][0]']
activation_44 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_44[0][0]']
conv2d_45 (Conv2D)	(None, 12, 12, 160)	179200	['activation_44[0][0]']
batch_normalization_45 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_45[0][0]']
activation_45 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_45[0][0]']
conv2d_41 (Conv2D)	(None, 12, 12, 160)	122880	['mixed4[0][0]']
conv2d_46 (Conv2D)	(None, 12, 12, 160)	179200	['activation_45[0][0]']

batch_normalization_41 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_41[0][0]']
batch_normalization_46 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_46[0][0]']
activation_41 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_41[0][0]']
activation_46 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_46[0][0]']
conv2d_42 (Conv2D)	(None, 12, 12, 160)	179200	['activation_41[0][0]']
conv2d_47 (Conv2D)	(None, 12, 12, 160)	179200	['activation_46[0][0]']
batch_normalization_42 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_42[0][0]']
batch_normalization_47 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_47[0][0]']
activation_42 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_42[0][0]']
activation_47 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_47[0][0]']
average_pooling2d_4 (Average Pooling2D)	(None, 12, 12, 768)	0	['mixed4[0][0]']
conv2d_40 (Conv2D)	(None, 12, 12, 192)	147456	['mixed4[0][0]']
conv2d_43 (Conv2D)	(None, 12, 12, 192)	215040	['activation_42[0][0]']
conv2d_48 (Conv2D)	(None, 12, 12, 192)	215040	['activation_47[0][0]']
conv2d_49 (Conv2D)	(None, 12, 12, 192)	147456	['average_pooling2d_4[0][0]']

batch_normalization_40 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_40[0][0]']
batch_normalization_43 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_43[0][0]']
batch_normalization_48 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_48[0][0]']
batch_normalization_49 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_49[0][0]']
activation_40 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_40[0][0]']
activation_43 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_43[0][0]']
activation_48 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_48[0][0]']
activation_49 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_49[0][0]']
mixed5 (Concatenate)	(None, 12, 12, 768)	0	['activation_40[0][0]', 'activation_43[0][0]', 'activation_48[0][0]', 'activation_49[0][0]']
conv2d_54 (Conv2D)	(None, 12, 12, 160)	122880	['mixed5[0][0]']
batch_normalization_54 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_54[0][0]']
activation_54 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_54[0][0]']
conv2d_55 (Conv2D)	(None, 12, 12, 160)	179200	['activation_54[0][0]']
batch_normalization_55 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_55[0][0]']

activation_55 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_55[0][0]']
conv2d_51 (Conv2D)	(None, 12, 12, 160)	122880	['mixed5[0][0]']
conv2d_56 (Conv2D)	(None, 12, 12, 160)	179200	['activation_55[0][0]']
batch_normalization_51 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_51[0][0]']
batch_normalization_56 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_56[0][0]']
activation_51 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_51[0][0]']
activation_56 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_56[0][0]']
conv2d_52 (Conv2D)	(None, 12, 12, 160)	179200	['activation_51[0][0]']
conv2d_57 (Conv2D)	(None, 12, 12, 160)	179200	['activation_56[0][0]']
batch_normalization_52 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_52[0][0]']
batch_normalization_57 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_57[0][0]']
activation_52 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_52[0][0]']
activation_57 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_57[0][0]']
average_pooling2d_5 (Average Pooling2D)	(None, 12, 12, 768)	0	['mixed5[0][0]']
conv2d_50 (Conv2D)	(None, 12, 12, 192)	147456	['mixed5[0][0]']

conv2d_53 (Conv2D)	(None, 12, 12, 192)	215040	['activation_52[0][0]']
conv2d_58 (Conv2D)	(None, 12, 12, 192)	215040	['activation_57[0][0]']
conv2d_59 (Conv2D)	(None, 12, 12, 192)	147456	['average_pooling2d_5[0][0]']
batch_normalization_50 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_50[0][0]']
batch_normalization_53 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_53[0][0]']
batch_normalization_58 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_58[0][0]']
batch_normalization_59 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_59[0][0]']
activation_50 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_50[0][0]']
activation_53 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_53[0][0]']
activation_58 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_58[0][0]']
activation_59 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_59[0][0]']
mixed6 (Concatenate)	(None, 12, 12, 768)	0	['activation_50[0][0]', 'activation_53[0][0]', 'activation_58[0][0]', 'activation_59[0][0]']
conv2d_64 (Conv2D)	(None, 12, 12, 192)	147456	['mixed6[0][0]']
batch_normalization_64 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_64[0][0]']

activation_64 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_64[0][0]']
conv2d_65 (Conv2D)	(None, 12, 12, 192)	258048	['activation_64[0][0]']
batch_normalization_65 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_65[0][0]']
activation_65 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_65[0][0]']
conv2d_61 (Conv2D)	(None, 12, 12, 192)	147456	['mixed6[0][0]']
conv2d_66 (Conv2D)	(None, 12, 12, 192)	258048	['activation_65[0][0]']
batch_normalization_61 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_61[0][0]']
batch_normalization_66 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_66[0][0]']
activation_61 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_61[0][0]']
activation_66 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_66[0][0]']
conv2d_62 (Conv2D)	(None, 12, 12, 192)	258048	['activation_61[0][0]']
conv2d_67 (Conv2D)	(None, 12, 12, 192)	258048	['activation_66[0][0]']
batch_normalization_62 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_62[0][0]']
batch_normalization_67 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_67[0][0]']
activation_62 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_62[0][0]']
activation_67 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_67[0][0]']

average_pooling2d_6 (AveragePooling2D)	(None, 12, 12, 768)	0	['mixed6[0][0]']
conv2d_60 (Conv2D)	(None, 12, 12, 192)	147456	['mixed6[0][0]']
conv2d_63 (Conv2D)	(None, 12, 12, 192)	258048	['activation_62[0][0]']
conv2d_68 (Conv2D)	(None, 12, 12, 192)	258048	['activation_67[0][0]']
conv2d_69 (Conv2D)	(None, 12, 12, 192)	147456	['average_pooling2d_6[0][0]']
batch_normalization_60 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_60[0][0]']
batch_normalization_63 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_63[0][0]']
batch_normalization_68 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_68[0][0]']
batch_normalization_69 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_69[0][0]']
activation_60 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_60[0][0]']
activation_63 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_63[0][0]']
activation_68 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_68[0][0]']
activation_69 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_69[0][0]']
mixed7 (Concatenate)	(None, 12, 12, 768)	0	['activation_60[0][0]', 'activation_63[0][0]', 'activation_68[0][0]', 'activation_69[0][0]']

conv2d_72 (Conv2D)	(None, 12, 12, 192)	147456	['mixed7[0][0]']
batch_normalization_72 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_72[0][0]']
activation_72 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_72[0][0]']
conv2d_73 (Conv2D)	(None, 12, 12, 192)	258048	['activation_72[0][0]']
batch_normalization_73 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_73[0][0]']
activation_73 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_73[0][0]']
conv2d_70 (Conv2D)	(None, 12, 12, 192)	147456	['mixed7[0][0]']
conv2d_74 (Conv2D)	(None, 12, 12, 192)	258048	['activation_73[0][0]']
batch_normalization_70 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_70[0][0]']
batch_normalization_74 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_74[0][0]']
activation_70 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_70[0][0]']
activation_74 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_74[0][0]']
conv2d_71 (Conv2D)	(None, 5, 5, 320)	552960	['activation_70[0][0]']
conv2d_75 (Conv2D)	(None, 5, 5, 192)	331776	['activation_74[0][0]']
batch_normalization_71 (Batch Normalization)	(None, 5, 5, 320)	960	['conv2d_71[0][0]']

batch_normalization_75 (Batch Normalization)	(None, 5, 5, 192)	576	['conv2d_75[0][0]']
activation_71 (Activation)	(None, 5, 5, 320)	0	['batch_normalization_71[0][0]']
activation_75 (Activation)	(None, 5, 5, 192)	0	['batch_normalization_75[0][0]']
max_pooling2d_3 (MaxPooling2D)	(None, 5, 5, 768)	0	['mixed7[0][0]']
mixed8 (Concatenate)	(None, 5, 5, 1280)	0	['activation_71[0][0]', 'activation_75[0][0]', 'max_pooling2d_3[0][0]']
conv2d_80 (Conv2D)	(None, 5, 5, 448)	573440	['mixed8[0][0]']
batch_normalization_80 (Batch Normalization)	(None, 5, 5, 448)	1344	['conv2d_80[0][0]']
activation_80 (Activation)	(None, 5, 5, 448)	0	['batch_normalization_80[0][0]']
conv2d_77 (Conv2D)	(None, 5, 5, 384)	491520	['mixed8[0][0]']
conv2d_81 (Conv2D)	(None, 5, 5, 384)	1548288	['activation_80[0][0]']
batch_normalization_77 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_77[0][0]']
batch_normalization_81 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_81[0][0]']
activation_77 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_77[0][0]']
activation_81 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_81[0][0]']
conv2d_78 (Conv2D)	(None, 5, 5, 384)	442368	['activation_77[0][0]']

conv2d_79 (Conv2D)	(None, 5, 5, 384)	442368	['activation_77[0][0]']
conv2d_82 (Conv2D)	(None, 5, 5, 384)	442368	['activation_81[0][0]']
conv2d_83 (Conv2D)	(None, 5, 5, 384)	442368	['activation_81[0][0]']
average_pooling2d_7 (AveragePooling2D)	(None, 5, 5, 1280)	0	['mixed8[0][0]']
conv2d_76 (Conv2D)	(None, 5, 5, 320)	409600	['mixed8[0][0]']
batch_normalization_78 (BatchNormalization)	(None, 5, 5, 384)	1152	['conv2d_78[0][0]']
batch_normalization_79 (BatchNormalization)	(None, 5, 5, 384)	1152	['conv2d_79[0][0]']
batch_normalization_82 (BatchNormalization)	(None, 5, 5, 384)	1152	['conv2d_82[0][0]']
batch_normalization_83 (BatchNormalization)	(None, 5, 5, 384)	1152	['conv2d_83[0][0]']
conv2d_84 (Conv2D)	(None, 5, 5, 192)	245760	['average_pooling2d_7[0][0]']
batch_normalization_76 (BatchNormalization)	(None, 5, 5, 320)	960	['conv2d_76[0][0]']
activation_78 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_78[0][0]']
activation_79 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_79[0][0]']
activation_82 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_82[0][0]']
activation_83 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_83[0][0]']

batch_normalization_84 (Batch Normalization)	(None, 5, 5, 192)	576	['conv2d_84[0][0]']
activation_76 (Activation)	(None, 5, 5, 320)	0	['batch_normalization_76[0][0]']
mixed9_0 (Concatenate)	(None, 5, 5, 768)	0	['activation_78[0][0]', 'activation_79[0][0]']
concatenate (Concatenate)	(None, 5, 5, 768)	0	['activation_82[0][0]', 'activation_83[0][0]']
activation_84 (Activation)	(None, 5, 5, 192)	0	['batch_normalization_84[0][0]']
mixed9 (Concatenate)	(None, 5, 5, 2048)	0	['activation_76[0][0]', 'mixed9_0[0][0]', 'concatenate[0][0]', 'activation_84[0][0]']
conv2d_89 (Conv2D)	(None, 5, 5, 448)	917504	['mixed9[0][0]']
batch_normalization_89 (Batch Normalization)	(None, 5, 5, 448)	1344	['conv2d_89[0][0]']
activation_89 (Activation)	(None, 5, 5, 448)	0	['batch_normalization_89[0][0]']
conv2d_86 (Conv2D)	(None, 5, 5, 384)	786432	['mixed9[0][0]']
conv2d_90 (Conv2D)	(None, 5, 5, 384)	1548288	['activation_89[0][0]']
batch_normalization_86 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_86[0][0]']
batch_normalization_90 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_90[0][0]']
activation_86 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_86[0][0]']

activation_90 (Activation) [0][0]']	(None, 5, 5, 384)	0	['batch_normalization_90
conv2d_87 (Conv2D)	(None, 5, 5, 384)	442368	['activation_86[0][0]']
conv2d_88 (Conv2D)	(None, 5, 5, 384)	442368	['activation_86[0][0]']
conv2d_91 (Conv2D)	(None, 5, 5, 384)	442368	['activation_90[0][0]']
conv2d_92 (Conv2D)	(None, 5, 5, 384)	442368	['activation_90[0][0]']
average_pooling2d_8 (AveragePo oling2D)	(None, 5, 5, 2048)	0	['mixed9[0][0]']
conv2d_85 (Conv2D)	(None, 5, 5, 320)	655360	['mixed9[0][0]']
batch_normalization_87 (BatchN ormalization)	(None, 5, 5, 384)	1152	['conv2d_87[0][0]']
batch_normalization_88 (BatchN ormalization)	(None, 5, 5, 384)	1152	['conv2d_88[0][0]']
batch_normalization_91 (BatchN ormalization)	(None, 5, 5, 384)	1152	['conv2d_91[0][0]']
batch_normalization_92 (BatchN ormalization)	(None, 5, 5, 384)	1152	['conv2d_92[0][0]']
conv2d_93 (Conv2D) [0]']	(None, 5, 5, 192)	393216	['average_pooling2d_8[0]
batch_normalization_85 (BatchN ormalization)	(None, 5, 5, 320)	960	['conv2d_85[0][0]']
activation_87 (Activation) [0][0]']	(None, 5, 5, 384)	0	['batch_normalization_87
activation_88 (Activation) [0][0]']	(None, 5, 5, 384)	0	['batch_normalization_88

activation_91 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_91[0][0]']
activation_92 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_92[0][0]']
batch_normalization_93 (Batch Normalization)	(None, 5, 5, 192)	576	['conv2d_93[0][0]']
activation_85 (Activation)	(None, 5, 5, 320)	0	['batch_normalization_85[0][0]']
mixed9_1 (Concatenate)	(None, 5, 5, 768)	0	['activation_87[0][0]', 'activation_88[0][0]']
concatenate_1 (Concatenate)	(None, 5, 5, 768)	0	['activation_91[0][0]', 'activation_92[0][0]']
activation_93 (Activation)	(None, 5, 5, 192)	0	['batch_normalization_93[0][0]']
mixed10 (Concatenate)	(None, 5, 5, 2048)	0	['activation_85[0][0]', 'mixed9_1[0][0]', 'concatenate_1[0][0]', 'activation_93[0][0]']
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	['mixed10[0][0]']
dense_1 (Dense)	(None, 256)	524544	['global_average_pooling2d[0][0]']
predictions (Dense)	(None, 2)	514	['dense_1[0][0]']

```

=====
Total params: 22,327,842
Trainable params: 525,058
Non-trainable params: 21,802,784

```

In [41]:

```
vgg_model.compile(optimizer='adam', loss = 'categorical_crossentropy', metrics=['accuracy'])
```

In [42]:

```
resnet_model.compile(optimizer='adam', loss = 'categorical_crossentropy', metrics=['accuracy'])
```

In [43]:

```
Inc_model.compile(optimizer='adam', loss = 'categorical_crossentropy', metrics=['accuracy'])
```

Training Process

In [44]:

```
vgg_hist = vgg_model.fit_generator(vgg_train, steps_per_epoch= 10, epochs= 8, validation_data= vgg_valid , validation_steps= 32)
```

C:\Users\JOHNYU~1\AppData\Local\Temp\ipykernel_21208\3806944101.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

```
vgg_hist = vgg_model.fit_generator(vgg_train, steps_per_epoch= 10, epochs= 8, validation_data= vgg_valid , validation_steps= 32)
```

Epoch 1/8

10/10 [=====] - 191s 21s/step - loss: 0.8253 - accuracy: 0.6406
- val_loss: 0.5481 - val_accuracy: 0.7698

Epoch 2/8

10/10 [=====] - 192s 21s/step - loss: 0.6379 - accuracy: 0.7188
- val_loss: 0.4216 - val_accuracy: 0.8264

Epoch 3/8

10/10 [=====] - 190s 21s/step - loss: 0.4887 - accuracy: 0.8125
- val_loss: 0.3625 - val_accuracy: 0.8413

Epoch 4/8

10/10 [=====] - 192s 21s/step - loss: 0.4031 - accuracy: 0.8094
- val_loss: 0.3737 - val_accuracy: 0.8343

Epoch 5/8

10/10 [=====] - 191s 21s/step - loss: 0.4031 - accuracy: 0.8219
- val_loss: 0.3213 - val_accuracy: 0.8730

Epoch 6/8

10/10 [=====] - 192s 21s/step - loss: 0.4314 - accuracy: 0.8062
- val_loss: 0.4875 - val_accuracy: 0.7738

Epoch 7/8

10/10 [=====] - 196s 21s/step - loss: 0.3705 - accuracy: 0.8219
- val_loss: 0.3177 - val_accuracy: 0.8740

Epoch 8/8

10/10 [=====] - 190s 21s/step - loss: 0.3326 - accuracy: 0.8406
- val_loss: 0.2779 - val_accuracy: 0.8938

In [45]:

```
resnet_hist = resnet_model.fit_generator(resnet_train, steps_per_epoch= 10, epochs= 8, validation_data= resnet_valid , validation_steps= 32)
```

C:\Users\JOHNYU~1\AppData\Local\Temp\ipykernel_21208\530071546.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

```
resnet_hist = resnet_model.fit_generator(resnet_train, steps_per_epoch= 10, epochs= 8, validation_data= resnet_valid , validation_steps= 32)
```

Epoch 1/8

10/10 [=====] - 90s 9s/step - loss: 24.5943 - accuracy: 0.6156 -
val_loss: 2.9157 - val_accuracy: 0.8581

Epoch 2/8

10/10 [=====] - 85s 9s/step - loss: 6.6117 - accuracy: 0.8062 -
val_loss: 1.4771 - val_accuracy: 0.9177

Epoch 3/8

10/10 [=====] - 80s 9s/step - loss: 3.2617 - accuracy: 0.8250 -

```
val_loss: 1.7947 - val_accuracy: 0.8829
Epoch 4/8
10/10 [=====] - 80s 9s/step - loss: 1.4316 - accuracy: 0.8719 -
val_loss: 0.3509 - val_accuracy: 0.9375
Epoch 5/8
10/10 [=====] - 80s 9s/step - loss: 0.4608 - accuracy: 0.9000 -
val_loss: 0.2815 - val_accuracy: 0.9355
Epoch 6/8
10/10 [=====] - 82s 9s/step - loss: 0.3672 - accuracy: 0.8938 -
val_loss: 0.3002 - val_accuracy: 0.9385
Epoch 7/8
10/10 [=====] - 82s 9s/step - loss: 0.3096 - accuracy: 0.8969 -
val_loss: 0.2126 - val_accuracy: 0.9484
Epoch 8/8
10/10 [=====] - 82s 9s/step - loss: 0.2048 - accuracy: 0.9219 -
val_loss: 0.1341 - val_accuracy: 0.9623
```

In [46]:

```
Inc_hist = Inc_model.fit_generator(Inc_train, steps_per_epoch= 10, epochs= 8, validation
_data= Inc_valid , validation_steps= 32)
```

```
C:\Users\JOHNYU~1\AppData\Local\Temp\ipykernel_21208\4004650067.py:1: UserWarning: `Model
.fit_generator` is deprecated and will be removed in a future version. Please use `Model.
fit`, which supports generators.
```

```
Inc_hist = Inc_model.fit_generator(Inc_train, steps_per_epoch= 10, epochs= 8, validatio
n_data= Inc_valid , validation_steps= 32)
```

```
Epoch 1/8
10/10 [=====] - 52s 5s/step - loss: 2.2979 - accuracy: 0.4812 -
val_loss: 1.4602 - val_accuracy: 0.4960
Epoch 2/8
10/10 [=====] - 46s 5s/step - loss: 0.9773 - accuracy: 0.5353 -
val_loss: 0.6402 - val_accuracy: 0.6329
Epoch 3/8
10/10 [=====] - 47s 5s/step - loss: 0.5480 - accuracy: 0.7312 -
val_loss: 0.5166 - val_accuracy: 0.7569
Epoch 4/8
10/10 [=====] - 46s 5s/step - loss: 0.4706 - accuracy: 0.7906 -
val_loss: 0.5085 - val_accuracy: 0.7599
Epoch 5/8
10/10 [=====] - 46s 5s/step - loss: 0.4398 - accuracy: 0.7906 -
val_loss: 0.4534 - val_accuracy: 0.7877
Epoch 6/8
10/10 [=====] - 45s 5s/step - loss: 0.4621 - accuracy: 0.8031 -
val_loss: 0.4446 - val_accuracy: 0.7798
Epoch 7/8
10/10 [=====] - 44s 5s/step - loss: 0.4245 - accuracy: 0.7969 -
val_loss: 0.4396 - val_accuracy: 0.8075
Epoch 8/8
10/10 [=====] - 45s 5s/step - loss: 0.4301 - accuracy: 0.8188 -
val_loss: 0.4339 - val_accuracy: 0.7996
```

Model performamce

In [47]:

```
# vgg
vgg_h = vgg_hist.history
vgg_h.keys()
```

Out[47]:

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

In [48]:

```
# resnet
resnet_h = resnet_hist.history
resnet_h.keys()
```

Out [48]:

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

In [49]:

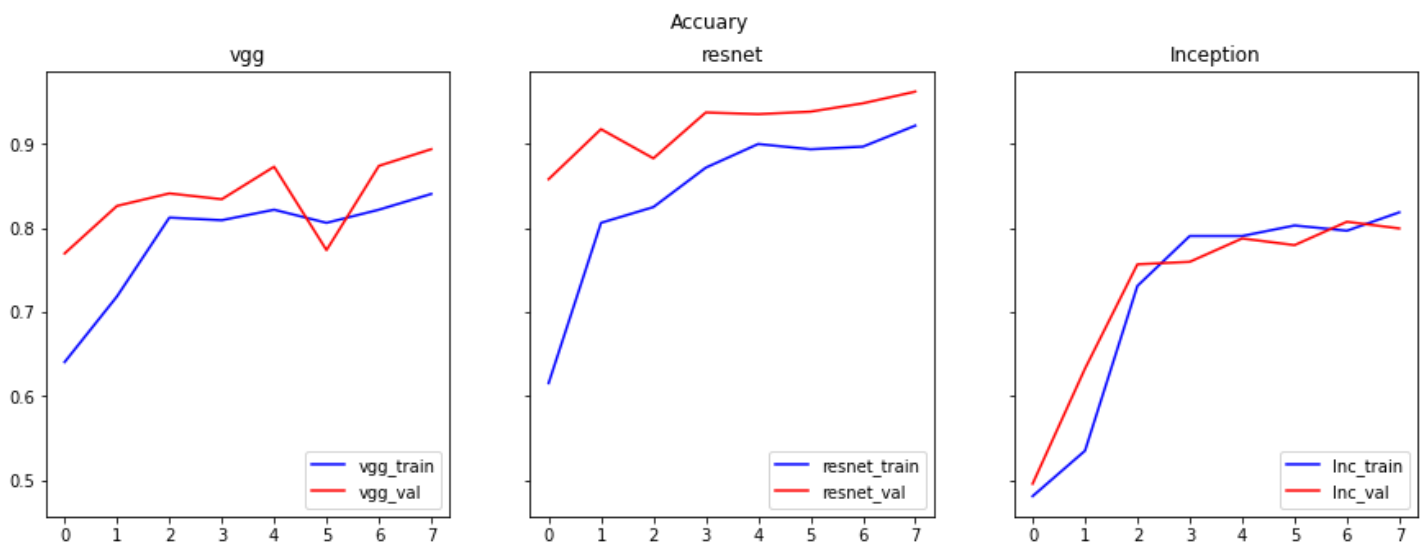
```
# Inception
Inc_h = Inc_hist.history
Inc_h.keys()
```

Out [49]:

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

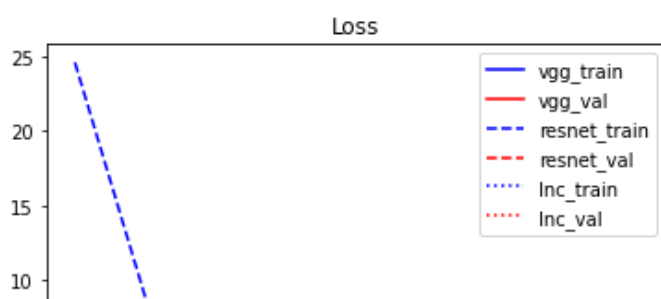
In [50]:

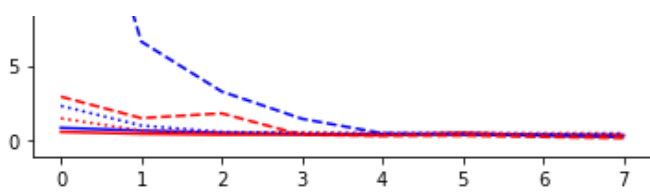
```
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15,5), sharey = True)
fig.suptitle('Accuary')
ax1.plot(vgg_h['accuracy'], c='blue', label = 'vgg_train')
ax1.plot(vgg_h['val_accuracy'], c = "red", label = 'vgg_val')
ax2.plot(resnet_h['accuracy'], c='blue', label = 'resnet_train')
ax2.plot(resnet_h['val_accuracy'], c = "red", label = 'resnet_val')
ax3.plot(Inc_h['accuracy'], c='blue', label = 'Inc_train')
ax3.plot(Inc_h['val_accuracy'], c = "red", label = 'Inc_val')
ax1.title.set_text("vgg")
ax2.title.set_text("resnet")
ax3.title.set_text("Inception")
ax1.legend(loc="lower right")
ax2.legend(loc="lower right")
ax3.legend(loc="lower right")
plt.show()
```



In [51]:

```
plt.plot(vgg_h['loss'], c='blue', label = 'vgg_train')
plt.plot(vgg_h['val_loss'], c = "red", label = 'vgg_val')
plt.plot(resnet_h['loss'], c='blue', linestyle = '--', label = 'resnet_train')
plt.plot(resnet_h['val_loss'], linestyle = '--', c = "red", label = 'resnet_val')
plt.plot(Inc_h['loss'], c='blue', linestyle = ':', label = 'Inc_train')
plt.plot(Inc_h['val_loss'], linestyle = ':', c = "red", label = 'Inc_val')
plt.title("Loss")
plt.legend(loc="upper right")
plt.show()
```





In [52]:

```
vgg_model.evaluate_generator(generator= vgg_test , verbose=1) # Returns [loss , accuracy]
```

C:\Users\JOHNYU~1\AppData\Local\Temp\ipykernel_21208\265580602.py:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.

```
vgg_model.evaluate_generator(generator= vgg_test , verbose=1) # Returns [loss , accuracy]
```

25/25 [=====] - 113s 4s/step - loss: 0.3000 - accuracy: 0.9025

Out[52]:

```
[0.3000353276729584, 0.9024999737739563]
```

In [53]:

```
resnet_model.evaluate_generator(generator= resnet_test , verbose=1) # Returns [loss , accuracy]
```

C:\Users\JOHNYU~1\AppData\Local\Temp\ipykernel_21208\1414072780.py:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.

```
resnet_model.evaluate_generator(generator= resnet_test , verbose=1) # Returns [loss , accuracy]
```

25/25 [=====] - 47s 2s/step - loss: 0.1602 - accuracy: 0.9450

Out[53]:

```
[0.16023865342140198, 0.9449999928474426]
```

In [54]:

```
Inc_model.evaluate_generator(generator= Inc_test , verbose=1) # Returns [loss , accuracy]
```

C:\Users\JOHNYU~1\AppData\Local\Temp\ipykernel_21208\3240097718.py:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.

```
Inc_model.evaluate_generator(generator= Inc_test , verbose=1) # Returns [loss , accuracy]
```

25/25 [=====] - 27s 1s/step - loss: 0.4469 - accuracy: 0.7788

Out[54]:

```
[0.4469171166419983, 0.7787500023841858]
```

In [55]:

```
from keras.preprocessing import image

def get_img_array(img_path):
    """
    Input : Takes in image path as input
    Output : Gives out Pre-Processed image
    """
    path = img_path
    img = image.load_img(path, target_size=(224,224,3))
    img = image.img_to_array(img)/255
    img = np.expand_dims(img , axis= 0 )

    return img
```

In [56]:


```
path = "C:/IE 7615/Project/all_image/train_test_split/test/Covid/COVID-165.png"
```

In [57]:

```
#predictions
img = get_img_array(path)

vgg_res = class_type[np.argmax(vgg_model.predict(img))]
print(vgg_res)

print(f"The chances of image being Covid is : {vgg_model.predict(img)[0][0]*100} percent"
)
print(f"The chances of image being Normal is : {vgg_model.predict(img)[0][1]*100} percent
")
```

Covid

The chances of image being Covid is : 64.62266445159912 percent

The chances of image being Normal is : 20.76798677444458 percent

In [58]:

```
#predictions
img = get_img_array(path)

resnet_res = class_type[np.argmax(resnet_model.predict(img))]
print(resnet_res)

print(f"The chances of image being Covid is : {resnet_model.predict(img)[0][0]*100} perce
nt")
print(f"The chances of image being Normal is : {resnet_model.predict(img)[0][1]*100} perc
ent")
```

Covid

The chances of image being Covid is : 94.17915940284729 percent

The chances of image being Normal is : 20.51371932029724 percent

In [59]:

```
#predictions
img = get_img_array(path)

Inc_res = class_type[np.argmax(Inc_model.predict(img))]
print(Inc_res)

print(f"The chances of image being Covid is : {Inc_model.predict(img)[0][0]*100} percent"
)
print(f"The chances of image being Normal is : {Inc_model.predict(img)[0][1]*100} percent
")
```

Covid

The chances of image being Covid is : 69.99164819717407 percent

The chances of image being Normal is : 51.29391551017761 percent

In [60]:

```
path_delta = "C:/IE 7615/Project/Delta Variant/1638755207.jpg"
```

In [61]:

```
#predictions
img = get_img_array(path_delta)

vgg_res = class_type[np.argmax(vgg_model.predict(img))]
print(vgg_res)

print(f"The chances of image being Covid is : {vgg_model.predict(img)[0][0]*100} percent"
)
print(f"The chances of image being Normal is : {vgg_model.predict(img)[0][1]*100} percent
")
```

Covid

The chances of image being Covid is : 79.10464406013489 percent

The chances of image being Normal is : 22.61694073677063 percent

In [62]:

```
#predictions
img = get_img_array(path_delta)

resnet_res = class_type[np.argmax(resnet_model.predict(img))]
print(resnet_res)

print(f"The chances of image being Covid is : {resnet_model.predict(img)[0][0]*100} percent")
print(f"The chances of image being Normal is : {resnet_model.predict(img)[0][1]*100} percent")
```

Covid

The chances of image being Covid is : 95.23797035217285 percent

The chances of image being Normal is : 9.606027603149414 percent

In [63]:

```
#predictions
img = get_img_array(path_delta)

Inc_res = class_type[np.argmax(Inc_model.predict(img))]
print(Inc_res)

print(f"The chances of image being Covid is : {Inc_model.predict(img)[0][0]*100} percent")
print(f"The chances of image being Normal is : {Inc_model.predict(img)[0][1]*100} percent")
```

Covid

The chances of image being Covid is : 83.30746293067932 percent

The chances of image being Normal is : 37.862685322761536 percent

In []: