In [1]:
```
from keras.layers import Input, Lambda, Dense, Flatten
from keras.models import Model
from keras.applications.vgg16 import VGG16

import numpy as np
from glob import glob
```

In [2]:
```
size_of_image= [224, 224]

train_path = 'C:\\Users\\User\\Documents\\EDT Project\\Dataset\\chest_xray\\train'

valid_path = 'C:\\Users\\User\\Documents\\EDT Project\\Dataset\\chest_xray\\test'
```

In [3]:
```
vgg = VGG16(input_shape=size_of_image + [3], weights='imagenet', include_top=False)
```

In [4]:
```
for layer in vgg.layers:
    layer.trainable = False
```

In [5]:
```
folders_present = glob('C:\\Users\\User\\Documents\\EDT Project\\Dataset\\chest_xray\\train\\*')
print(folders_present)
```

```
['C:\\Users\\User\\Documents\\EDT Project\\Dataset\\chest_xray\\train\\NORMAL', 'C:\\Users\\User\\Documents\\EDT Projec
t\\Dataset\\chest_xray\\train\\PNEUMONIA']
```

In [6]:
```
x = Flatten()(vgg.output)
prediction = Dense(len(folders_present), activation='softmax')(x)
```

In [7]:
```
model = Model(inputs=vgg.input, outputs=prediction)
```

In [8]:
```
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (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
_________________________________________________________________
dense (Dense)                (None, 2)                 50178
=================================================================
Total params: 14,764,866
Trainable params: 50,178
Non-trainable params: 14,714,688
_________________________________________________________________
```

In [9]:
```
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
```

In [10]:
```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

In [11]:
```
training_set = train_datagen.flow_from_directory('C:\\Users\\User\\Documents\\EDT Project\\Dataset\\chest_xray\\train',
                                                 target_size = (224, 224),
                                                 batch_size = 32,
                                                 class_mode = 'categorical')
print(len(training_set))
```

```
Found 5216 images belonging to 2 classes.
163
```

In [12]:
```
test_set = test_datagen.flow_from_directory('C:\\Users\\User\\Documents\\EDT Project\\Dataset\\chest_xray\\test',
                                            target_size = (224, 224),
                                            batch_size = 32,
                                            class_mode = 'categorical')
print(len(test_set))
```

```
Found 624 images belonging to 2 classes.
20
```

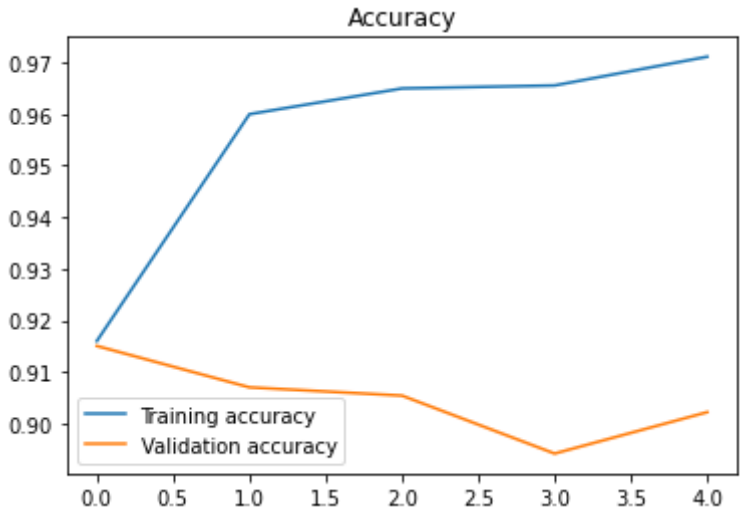
In [13]:
```
r = model.fit_generator(
  training_set,
  validation_data=test_set,
  epochs=5,
  steps_per_epoch=len(training_set),
  validation_steps=len(test_set)
)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\training.py:1972: UserWarning: `Model.fit_generator` is depreca
ted and will be removed in a future version. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '

Epoch 1/5
163/163 [==============================] - 3076s 19s/step - loss: 0.2201 - accuracy: 0.9160 - val_loss: 0.2328 - val_ac
curacy: 0.9151
Epoch 2/5
163/163 [==============================] - 2989s 18s/step - loss: 0.1075 - accuracy: 0.9599 - val_loss: 0.2933 - val_ac
curacy: 0.9071
Epoch 3/5
163/163 [==============================] - 2966s 18s/step - loss: 0.0979 - accuracy: 0.9649 - val_loss: 0.3013 - val_ac
curacy: 0.9054
Epoch 4/5
163/163 [==============================] - 2961s 18s/step - loss: 0.0936 - accuracy: 0.9655 - val_loss: 0.3656 - val_ac
curacy: 0.8942
Epoch 5/5
163/163 [==============================] - 2575s 16s/step - loss: 0.0719 - accuracy: 0.9711 - val_loss: 0.3451 - val_ac
curacy: 0.9022
```

In [14]:
```
import tensorflow as tf
from keras.models import load_model
model.save('model_vgg16.h5')
```

In [15]:
```
import matplotlib.pyplot as plt
plt.plot(r.history['accuracy'], label='Training accuracy')
plt.plot(r.history['val_accuracy'], label='Validation accuracy')
plt.legend()
plt.title("Accuracy")
plt.show()
```



In [16]:
```
plt.plot(r.history['loss'], label='Training loss')
plt.plot(r.history['val_loss'], label='Validation loss')
plt.legend()
plt.title("Loss")
plt.show()
```

In [53]:
```
from keras.models import load_model
from keras.preprocessing import image
from keras.applications.vgg16 import preprocess_input

import numpy as np


x_ray = 'C:\\Users\\User\\Documents\\EDT Project\\Dataset\\chest_xray\\val\\PNEUMONIA\\person1954_bacteria_4886.jpeg'
img = image.load_img(x_ray,target_size = (224,224))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
img_data = preprocess_input(x)

saved_model = load_model('C:\\Users\\User\\Documents\\EDT Project\\Lung Disease Detection\\model_vgg16.h5')
predict_class = saved_model.predict(img_data)
```

In [54]:
```
import matplotlib.pyplot as plt
import matplotlib.image as img

img = img.imread(x_ray)
imgplot = plt.imshow(img,cmap='gray')
plt.show()

print()
print("\t\t",predict_class)

if(predict_class[0][0] > predict_class[0][1]):
  print("\t\tPerson is Normal")
else:
  print("\tPerson is having Pneumonia Disease")
```


```
        [[0.2856061 0.7143939]]
Person is having Pneumonia Disease
```