In [1]:
```
# Import libraries
import os,cv2
import time
import numpy as np
import matplotlib.pyplot as plt

from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
from keras.preprocessing import image
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers import Input
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from keras import callbacks
from keras import backend as K
K.set_image_data_format('channels_last')
from sklearn.metrics import classification_report,confusion_matrix
import itertools
from keras.models import Model
# from tensorflow.keras.applications.resnet import ResNet50
# from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.applications.mobilenet import MobileNet
from tensorflow.keras.applications.inception_v3 import decode_predictions
from tensorflow.keras.applications.vgg19 import VGG19
```

Set path for application

In [2]:
```
data_path = 'D:/Harold/MyDNN/DataSet/Chest_xray_seperate'
data_dir_list = os.listdir(data_path)
print(data_path)
```

```
D:/Harold/MyDNN/DataSet/Chest_xray_seperate
```

Set Image Size and Epocs

In [3]:
```
img_rows=128
img_cols=128
num_channel=3
num_epoch=300
```

Define the number of classes

In [4]:
```
num_classes = 2

img_data_list=[]
```

In [5]:
```
def preprocess_input(x):
    x[:, :, :, 0] -= 103.939
    x[:, :, :, 1] -= 116.779
    x[:, :, :, 2] -= 123.68
    # 'RGB'->'BGR'
    x = x[:, :, :, ::-1]
    return x

def data_preperation():
    for dataset in data_dir_list:
        img_list=os.listdir(data_path+'/'+ dataset)
        print ('Loading the images of dataset-'+'{}\n'.format(dataset))
        for img in img_list:
            img_path = data_path + '/'+ dataset + '/'+ img
            img = image.load_img(img_path, target_size=(224, 224))
            x = image.img_to_array(img)
            x = np.expand_dims(x, axis=0)
            x = preprocess_input(x)
#             print('Input image shape:', x.shape)
            img_data_list.append(x)
        print('Loading Complete')

#     for dataset in data_dir_list:
#         img_list=os.listdir(data_path+'/'+ dataset)
#         print ('Loading the images of dataset-'+'{}\n'.format(dataset))
#         for img in img_list:
#             img_path = data_path + '/'+ dataset + '/'+ img
#             img = image.load_img(img_path, target_size=(224, 224))
#             x = image.img_to_array(img)
#             x = np.expand_dims(x, axis=0)
#             x = preprocess_input(x)
# #             print('Input image shape:', x.shape)
#             img_data_list.append(x)
#         print('Loading Complete')

def display_loss_accuracy(hist):
    train_loss=hist.history['loss']
    val_loss=hist.history['val_loss']
    train_acc=hist.history['accuracy']
    val_acc=hist.history['val_accuracy']
    xc=range(num_epoch)

    plt.figure(1,figsize=(7,5))
    plt.plot(xc,train_loss)
    plt.plot(xc,val_loss)
    plt.xlabel('num of Epochs')
    plt.ylabel('loss')
    plt.title('train_loss vs val_loss')
    plt.grid(True)
    plt.legend(['train','val'])
    #print plt.style.available # use bmh, classic,ggplot for big pictures
    plt.style.use(['classic'])

    plt.figure(2,figsize=(7,5))
    plt.plot(xc,train_acc)
    plt.plot(xc,val_acc)
    plt.xlabel('num of Epochs')
    plt.ylabel('accuracy')
    plt.title('train_acc vs val_acc')
    plt.grid(True)
    plt.legend(['train','val'],loc=4)
    #print plt.style.available # use bmh, classic,ggplot for big pictures
    plt.style.use(['classic'])
```

```
def get_featuremaps(model, layer_idx, X_batch):
    get_activations = K.function([model.layers[0].input, K.learning_phase()],[mode
l.layers[layer_idx].output,])
    activations = get_activations([X_batch,0])
    return activations

def plot_featuremap_activations(activations):
    print (np.shape(activations))
    feature_maps = activations[0][0]
    print (np.shape(feature_maps))
    print (feature_maps.shape)

    fig=plt.figure(figsize=(16,16))
    plt.imshow(feature_maps[:,:,filter_num],cmap='gray')
    plt.savefig("featuremaps-layer-{}".format(layer_num) + "-filternum-{}".format(f
ilter_num)+'.jpg')

    num_of_featuremaps=feature_maps.shape[2]
    fig=plt.figure(figsize=(16,16))
    plt.title("featuremaps-layer-{}".format(layer_num))
    subplot_num=int(np.ceil(np.sqrt(num_of_featuremaps)))
    for i in range(int(num_of_featuremaps)):
        ax = fig.add_subplot(subplot_num, subplot_num, i+1)
        #ax.imshow(output_image[0,:,:,i],interpolation='nearest' ) #to see the firs
t filter
        ax.imshow(feature_maps[:,:,i],cmap='gray')
        plt.xticks([])
        plt.yticks([])
        plt.tight_layout()
    plt.show()
    fig.savefig("featuremaps-layer-{}".format(layer_num) + '.jpg')

# Plotting the confusion matrix
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.figure()
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")
```

```
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()
```

Data Preperation

In [6]:
```
# Calling Data Preperation
data_preperation()
```

```
Loading the images of dataset-NORMAL

Loading Complete
Loading the images of dataset-PNEUMONIA

Loading Complete
```

In [7]:
```
print (len(img_data_list))
img_data = np.array(img_data_list)
#img_data = img_data.astype('float32')
print (img_data.shape)
img_data=np.rollaxis(img_data,1,0)
print (img_data.shape)
img_data=img_data[0]
print (img_data.shape)
```

```
5856
(5856, 1, 224, 224, 3)
(1, 5856, 224, 224, 3)
(5856, 224, 224, 3)
```

Assiging Labels

In [8]:
```
num_of_samples = img_data.shape[0]
labels = np.ones((num_of_samples,),dtype='int64')

labels[0:1582]=0
labels[1583:5856]=1

names = ['normal','pneumonia']
```

Creating clasas labels to one-hot encoding

In [9]:
```
# convert class labels to on-hot encoding
Y = np_utils.to_categorical(labels, num_classes)
```

Split Data set into training and validation set

In [10]:
```
#Shuffle the dataset
x,y = shuffle(img_data,Y, random_state=2)
# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_sta
te=2)
```

Model Definition

Training the classifier alone

In [11]:
```
image_input = Input(shape=(224, 224, 3))
model = VGG19(input_tensor=image_input, include_top=True,weights='imagenet')
model.summary()
output_layer = model(image_input)

# last_layer = model.get_layer('avg_pool').output
x= Flatten(name='flatten')(output_layer)
out = Dense(num_classes, activation='softmax', name='output_layer')(x)
custom_resnet_model = Model(inputs=image_input,outputs= out)
custom_resnet_model.summary()
```

```
Model: "vgg19"
_________________________________________________________________
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_conv4 (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_conv4 (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_conv4 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312
_________________________________________________________________
predictions (Dense)          (None, 1000)              4097000
=================================================================
Total params: 143,667,240
Trainable params: 143,667,240
Non-trainable params: 0
_________________________________________________________________
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

```
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
vgg19 (Functional)           (None, 1000)              143667240
_________________________________________________________________
flatten (Flatten)            (None, 1000)              0
_________________________________________________________________
output_layer (Dense)         (None, 2)                 2002
=================================================================
Total params: 143,669,242
Trainable params: 143,669,242
Non-trainable params: 0
```

In [12]:

```python
for layer in custom_resnet_model.layers[:-1]:
    layer.trainable = False

custom_resnet_model.layers[-1].trainable
```

Out[12]:

```
True
```

In [13]:

```python
custom_resnet_model.compile(loss='categorical_crossentropy',optimizer='adam',metric
s=['accuracy'])
```

In [14]:
```
t = time.time()
hist = custom_resnet_model.fit(X_train, y_train, batch_size=32, epochs=num_epoch, v
erbose=1, validation_data=(X_test, y_test))
print('Training time: %s' % (t - time.time()))
(loss, accuracy) = custom_resnet_model.evaluate(X_test, y_test, batch_size=10, verb
ose=1)
print("[INFO] loss={:.4f}, accuracy: {:.4f}%".format(loss,accuracy * 100))
```

```
Epoch 1/300
147/147 [==============================] - 13s 91ms/step - loss: 0.6473 - accura
cy: 0.7126 - val_loss: 0.6041 - val_accuracy: 0.7338
Epoch 2/300
147/147 [==============================] - 12s 82ms/step - loss: 0.5818 - accura
cy: 0.7289 - val_loss: 0.5561 - val_accuracy: 0.7338
Epoch 3/300
147/147 [==============================] - 12s 83ms/step - loss: 0.5463 - accura
cy: 0.7293 - val_loss: 0.5270 - val_accuracy: 0.7363
Epoch 4/300
147/147 [==============================] - 12s 83ms/step - loss: 0.5232 - accura
cy: 0.7310 - val_loss: 0.5056 - val_accuracy: 0.7406
Epoch 5/300
147/147 [==============================] - 12s 83ms/step - loss: 0.5048 - accura
cy: 0.7348 - val_loss: 0.4873 - val_accuracy: 0.7398
Epoch 6/300
147/147 [==============================] - 12s 83ms/step - loss: 0.4888 - accura
cy: 0.7415 - val_loss: 0.4709 - val_accuracy: 0.7423
Epoch 7/300
147/147 [==============================] - 12s 83ms/step - loss: 0.4744 - accura
cy: 0.7464 - val_loss: 0.4562 - val_accuracy: 0.7483
Epoch 8/300
147/147 [==============================] - 12s 83ms/step - loss: 0.4612 - accura
cy: 0.7536 - val_loss: 0.4426 - val_accuracy: 0.7594
Epoch 9/300
147/147 [==============================] - 12s 84ms/step - loss: 0.4493 - accura
cy: 0.7590 - val_loss: 0.4303 - val_accuracy: 0.7739
Epoch 10/300
147/147 [==============================] - 12s 83ms/step - loss: 0.4384 - accura
cy: 0.7711 - val_loss: 0.4189 - val_accuracy: 0.7833
Epoch 11/300
147/147 [==============================] - 12s 84ms/step - loss: 0.4284 - accura
cy: 0.7786 - val_loss: 0.4084 - val_accuracy: 0.8003
Epoch 12/300
147/147 [==============================] - 12s 84ms/step - loss: 0.4192 - accura
cy: 0.7867 - val_loss: 0.3988 - val_accuracy: 0.8106
Epoch 13/300
147/147 [==============================] - 12s 84ms/step - loss: 0.4108 - accura
cy: 0.7978 - val_loss: 0.3899 - val_accuracy: 0.8242
Epoch 14/300
147/147 [==============================] - 12s 84ms/step - loss: 0.4030 - accura
cy: 0.8081 - val_loss: 0.3816 - val_accuracy: 0.8311
Epoch 15/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3959 - accura
cy: 0.8158 - val_loss: 0.3740 - val_accuracy: 0.8362
Epoch 16/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3893 - accura
cy: 0.8224 - val_loss: 0.3669 - val_accuracy: 0.8413
Epoch 17/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3833 - accura
cy: 0.8313 - val_loss: 0.3603 - val_accuracy: 0.8430
Epoch 18/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3776 - accura
cy: 0.8328 - val_loss: 0.3542 - val_accuracy: 0.8498
Epoch 19/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3723 - accura
cy: 0.8380 - val_loss: 0.3485 - val_accuracy: 0.8549
Epoch 20/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3674 - accura
cy: 0.8401 - val_loss: 0.3431 - val_accuracy: 0.8601
Epoch 21/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3628 - accura
cy: 0.8463 - val_loss: 0.3381 - val_accuracy: 0.8643
```

```
Epoch 22/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3585 - accura
cy: 0.8482 - val_loss: 0.3333 - val_accuracy: 0.8695
Epoch 23/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3544 - accura
cy: 0.8508 - val_loss: 0.3288 - val_accuracy: 0.8729
Epoch 24/300
147/147 [==============================] - 12s 85ms/step - loss: 0.3506 - accura
cy: 0.8538 - val_loss: 0.3245 - val_accuracy: 0.8771
Epoch 25/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3469 - accura
cy: 0.8567 - val_loss: 0.3206 - val_accuracy: 0.8788
Epoch 26/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3436 - accura
cy: 0.8565 - val_loss: 0.3168 - val_accuracy: 0.8848
Epoch 27/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3403 - accura
cy: 0.8587 - val_loss: 0.3132 - val_accuracy: 0.8857
Epoch 28/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3373 - accura
cy: 0.8617 - val_loss: 0.3098 - val_accuracy: 0.8882
Epoch 29/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3344 - accura
cy: 0.8634 - val_loss: 0.3066 - val_accuracy: 0.8882
Epoch 30/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3316 - accura
cy: 0.8640 - val_loss: 0.3035 - val_accuracy: 0.8882
Epoch 31/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3290 - accura
cy: 0.8661 - val_loss: 0.3005 - val_accuracy: 0.8891
Epoch 32/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3264 - accura
cy: 0.8661 - val_loss: 0.2977 - val_accuracy: 0.8891
Epoch 33/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3240 - accura
cy: 0.8678 - val_loss: 0.2951 - val_accuracy: 0.8891
Epoch 34/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3218 - accura
cy: 0.8683 - val_loss: 0.2925 - val_accuracy: 0.8908
Epoch 35/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3195 - accura
cy: 0.8689 - val_loss: 0.2900 - val_accuracy: 0.8933
Epoch 36/300
147/147 [==============================] - 12s 83ms/step - loss: 0.3174 - accura
cy: 0.8706 - val_loss: 0.2876 - val_accuracy: 0.8942
Epoch 37/300
147/147 [==============================] - 12s 83ms/step - loss: 0.3153 - accura
cy: 0.8717 - val_loss: 0.2855 - val_accuracy: 0.8951
Epoch 38/300
147/147 [==============================] - 12s 83ms/step - loss: 0.3134 - accura
cy: 0.8717 - val_loss: 0.2833 - val_accuracy: 0.8951
Epoch 39/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3114 - accura
cy: 0.8715 - val_loss: 0.2812 - val_accuracy: 0.8976
Epoch 40/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3097 - accura
cy: 0.8736 - val_loss: 0.2792 - val_accuracy: 0.9002
Epoch 41/300
147/147 [==============================] - 12s 83ms/step - loss: 0.3078 - accura
cy: 0.8732 - val_loss: 0.2772 - val_accuracy: 0.9002
Epoch 42/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3061 - accura
cy: 0.8736 - val_loss: 0.2754 - val_accuracy: 0.8993
Epoch 43/300
```

```
147/147 [==============================] - 12s 83ms/step - loss: 0.3045 - accura
cy: 0.8747 - val_loss: 0.2735 - val_accuracy: 0.8993
Epoch 44/300
147/147 [==============================] - 12s 84ms/step - loss: 0.3029 - accura
cy: 0.8766 - val_loss: 0.2718 - val_accuracy: 0.9010
Epoch 45/300
147/147 [==============================] - 12s 83ms/step - loss: 0.3013 - accura
cy: 0.8757 - val_loss: 0.2701 - val_accuracy: 0.9027
Epoch 46/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2998 - accura
cy: 0.8757 - val_loss: 0.2685 - val_accuracy: 0.9027
Epoch 47/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2983 - accura
cy: 0.8785 - val_loss: 0.2669 - val_accuracy: 0.9027
Epoch 48/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2969 - accura
cy: 0.8792 - val_loss: 0.2654 - val_accuracy: 0.9027
Epoch 49/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2955 - accura
cy: 0.8792 - val_loss: 0.2639 - val_accuracy: 0.9044
Epoch 50/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2941 - accura
cy: 0.8796 - val_loss: 0.2624 - val_accuracy: 0.9078
Epoch 51/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2928 - accura
cy: 0.8811 - val_loss: 0.2610 - val_accuracy: 0.9078
Epoch 52/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2915 - accura
cy: 0.8815 - val_loss: 0.2597 - val_accuracy: 0.9087
Epoch 53/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2903 - accura
cy: 0.8826 - val_loss: 0.2583 - val_accuracy: 0.9096
Epoch 54/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2890 - accura
cy: 0.8824 - val_loss: 0.2571 - val_accuracy: 0.9096
Epoch 55/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2878 - accura
cy: 0.8822 - val_loss: 0.2557 - val_accuracy: 0.9147
Epoch 56/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2867 - accura
cy: 0.8849 - val_loss: 0.2545 - val_accuracy: 0.9138
Epoch 57/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2855 - accura
cy: 0.8856 - val_loss: 0.2533 - val_accuracy: 0.9147
Epoch 58/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2844 - accura
cy: 0.8858 - val_loss: 0.2522 - val_accuracy: 0.9147
Epoch 59/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2833 - accura
cy: 0.8860 - val_loss: 0.2511 - val_accuracy: 0.9147
Epoch 60/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2823 - accura
cy: 0.8866 - val_loss: 0.2499 - val_accuracy: 0.9147
Epoch 61/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2813 - accura
cy: 0.8871 - val_loss: 0.2489 - val_accuracy: 0.9155
Epoch 62/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2802 - accura
cy: 0.8866 - val_loss: 0.2478 - val_accuracy: 0.9164
Epoch 63/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2793 - accura
cy: 0.8879 - val_loss: 0.2468 - val_accuracy: 0.9164
Epoch 64/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2783 - accura
```

```
cy: 0.8879 - val_loss: 0.2458 - val_accuracy: 0.9164
Epoch 65/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2773 - accura
cy: 0.8871 - val_loss: 0.2449 - val_accuracy: 0.9155
Epoch 66/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2764 - accura
cy: 0.8881 - val_loss: 0.2440 - val_accuracy: 0.9164
Epoch 67/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2755 - accura
cy: 0.8881 - val_loss: 0.2430 - val_accuracy: 0.9155
Epoch 68/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2747 - accura
cy: 0.8898 - val_loss: 0.2420 - val_accuracy: 0.9147
Epoch 69/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2738 - accura
cy: 0.8903 - val_loss: 0.2412 - val_accuracy: 0.9155
Epoch 70/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2729 - accura
cy: 0.8915 - val_loss: 0.2403 - val_accuracy: 0.9155
Epoch 71/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2721 - accura
cy: 0.8907 - val_loss: 0.2395 - val_accuracy: 0.9155
Epoch 72/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2712 - accura
cy: 0.8915 - val_loss: 0.2387 - val_accuracy: 0.9155
Epoch 73/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2704 - accura
cy: 0.8933 - val_loss: 0.2378 - val_accuracy: 0.9155
Epoch 74/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2697 - accura
cy: 0.8928 - val_loss: 0.2370 - val_accuracy: 0.9147
Epoch 75/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2689 - accura
cy: 0.8937 - val_loss: 0.2362 - val_accuracy: 0.9138
Epoch 76/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2682 - accura
cy: 0.8943 - val_loss: 0.2355 - val_accuracy: 0.9155
Epoch 77/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2674 - accura
cy: 0.8937 - val_loss: 0.2347 - val_accuracy: 0.9138
Epoch 78/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2666 - accura
cy: 0.8945 - val_loss: 0.2339 - val_accuracy: 0.9130
Epoch 79/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2659 - accura
cy: 0.8947 - val_loss: 0.2332 - val_accuracy: 0.9130
Epoch 80/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2653 - accura
cy: 0.8950 - val_loss: 0.2326 - val_accuracy: 0.9155
Epoch 81/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2645 - accura
cy: 0.8954 - val_loss: 0.2318 - val_accuracy: 0.9121
Epoch 82/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2638 - accura
cy: 0.8965 - val_loss: 0.2311 - val_accuracy: 0.9130
Epoch 83/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2632 - accura
cy: 0.8965 - val_loss: 0.2304 - val_accuracy: 0.9121
Epoch 84/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2625 - accura
cy: 0.8975 - val_loss: 0.2298 - val_accuracy: 0.9138
Epoch 85/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2619 - accura
cy: 0.8973 - val_loss: 0.2292 - val_accuracy: 0.9147
```

```
Epoch 86/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2612 - accura
cy: 0.8965 - val_loss: 0.2284 - val_accuracy: 0.9155
Epoch 87/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2606 - accura
cy: 0.8988 - val_loss: 0.2278 - val_accuracy: 0.9172
Epoch 88/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2600 - accura
cy: 0.8990 - val_loss: 0.2272 - val_accuracy: 0.9164
Epoch 89/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2594 - accura
cy: 0.9009 - val_loss: 0.2267 - val_accuracy: 0.9164
Epoch 90/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2588 - accura
cy: 0.8999 - val_loss: 0.2261 - val_accuracy: 0.9164
Epoch 91/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2582 - accura
cy: 0.8997 - val_loss: 0.2255 - val_accuracy: 0.9181
Epoch 92/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2576 - accura
cy: 0.9005 - val_loss: 0.2250 - val_accuracy: 0.9181
Epoch 93/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2571 - accura
cy: 0.9007 - val_loss: 0.2244 - val_accuracy: 0.9181
Epoch 94/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2565 - accura
cy: 0.9005 - val_loss: 0.2239 - val_accuracy: 0.9181
Epoch 95/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2560 - accura
cy: 0.9007 - val_loss: 0.2233 - val_accuracy: 0.9181
Epoch 96/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2554 - accura
cy: 0.9026 - val_loss: 0.2227 - val_accuracy: 0.9198
Epoch 97/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2549 - accura
cy: 0.9026 - val_loss: 0.2222 - val_accuracy: 0.9189
Epoch 98/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2544 - accura
cy: 0.9026 - val_loss: 0.2216 - val_accuracy: 0.9189
Epoch 99/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2538 - accura
cy: 0.9039 - val_loss: 0.2212 - val_accuracy: 0.9189
Epoch 100/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2533 - accura
cy: 0.9033 - val_loss: 0.2206 - val_accuracy: 0.9189
Epoch 101/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2528 - accura
cy: 0.9035 - val_loss: 0.2202 - val_accuracy: 0.9198
Epoch 102/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2523 - accura
cy: 0.9039 - val_loss: 0.2196 - val_accuracy: 0.9215
Epoch 103/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2518 - accura
cy: 0.9039 - val_loss: 0.2192 - val_accuracy: 0.9215
Epoch 104/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2514 - accura
cy: 0.9052 - val_loss: 0.2187 - val_accuracy: 0.9198
Epoch 105/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2509 - accura
cy: 0.9048 - val_loss: 0.2183 - val_accuracy: 0.9206
Epoch 106/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2504 - accura
cy: 0.9052 - val_loss: 0.2179 - val_accuracy: 0.9215
Epoch 107/300
```

```
147/147 [==============================] - 12s 84ms/step - loss: 0.2499 - accura
cy: 0.9063 - val_loss: 0.2174 - val_accuracy: 0.9224
Epoch 108/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2495 - accura
cy: 0.9063 - val_loss: 0.2169 - val_accuracy: 0.9224
Epoch 109/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2491 - accura
cy: 0.9061 - val_loss: 0.2165 - val_accuracy: 0.9232
Epoch 110/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2486 - accura
cy: 0.9061 - val_loss: 0.2160 - val_accuracy: 0.9241
Epoch 111/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2481 - accura
cy: 0.9063 - val_loss: 0.2156 - val_accuracy: 0.9241
Epoch 112/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2477 - accura
cy: 0.9067 - val_loss: 0.2152 - val_accuracy: 0.9241
Epoch 113/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2473 - accura
cy: 0.9065 - val_loss: 0.2147 - val_accuracy: 0.9241
Epoch 114/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2469 - accura
cy: 0.9073 - val_loss: 0.2144 - val_accuracy: 0.9241
Epoch 115/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2465 - accura
cy: 0.9065 - val_loss: 0.2140 - val_accuracy: 0.9249
Epoch 116/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2461 - accura
cy: 0.9073 - val_loss: 0.2136 - val_accuracy: 0.9249
Epoch 117/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2457 - accura
cy: 0.9078 - val_loss: 0.2132 - val_accuracy: 0.9241
Epoch 118/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2453 - accura
cy: 0.9078 - val_loss: 0.2127 - val_accuracy: 0.9232
Epoch 119/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2449 - accura
cy: 0.9095 - val_loss: 0.2124 - val_accuracy: 0.9232
Epoch 120/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2445 - accura
cy: 0.9097 - val_loss: 0.2120 - val_accuracy: 0.9232
Epoch 121/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2441 - accura
cy: 0.9088 - val_loss: 0.2117 - val_accuracy: 0.9249
Epoch 122/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2437 - accura
cy: 0.9093 - val_loss: 0.2113 - val_accuracy: 0.9241
Epoch 123/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2433 - accura
cy: 0.9097 - val_loss: 0.2110 - val_accuracy: 0.9249
Epoch 124/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2430 - accura
cy: 0.9097 - val_loss: 0.2106 - val_accuracy: 0.9249
Epoch 125/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2426 - accura
cy: 0.9097 - val_loss: 0.2103 - val_accuracy: 0.9249
Epoch 126/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2423 - accura
cy: 0.9101 - val_loss: 0.2100 - val_accuracy: 0.9249
Epoch 127/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2419 - accura
cy: 0.9101 - val_loss: 0.2096 - val_accuracy: 0.9249
Epoch 128/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2416 - accura
```

```
cy: 0.9097 - val_loss: 0.2093 - val_accuracy: 0.9249
Epoch 129/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2413 - accura
cy: 0.9093 - val_loss: 0.2089 - val_accuracy: 0.9258
Epoch 130/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2409 - accura
cy: 0.9095 - val_loss: 0.2086 - val_accuracy: 0.9249
Epoch 131/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2405 - accura
cy: 0.9103 - val_loss: 0.2083 - val_accuracy: 0.9258
Epoch 132/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2402 - accura
cy: 0.9105 - val_loss: 0.2079 - val_accuracy: 0.9266
Epoch 133/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2399 - accura
cy: 0.9105 - val_loss: 0.2076 - val_accuracy: 0.9275
Epoch 134/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2395 - accura
cy: 0.9101 - val_loss: 0.2074 - val_accuracy: 0.9275
Epoch 135/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2392 - accura
cy: 0.9105 - val_loss: 0.2071 - val_accuracy: 0.9283
Epoch 136/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2389 - accura
cy: 0.9103 - val_loss: 0.2067 - val_accuracy: 0.9275
Epoch 137/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2386 - accura
cy: 0.9101 - val_loss: 0.2064 - val_accuracy: 0.9266
Epoch 138/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2383 - accura
cy: 0.9108 - val_loss: 0.2061 - val_accuracy: 0.9275
Epoch 139/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2380 - accura
cy: 0.9105 - val_loss: 0.2058 - val_accuracy: 0.9275
Epoch 140/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2376 - accura
cy: 0.9103 - val_loss: 0.2056 - val_accuracy: 0.9275
Epoch 141/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2374 - accura
cy: 0.9108 - val_loss: 0.2053 - val_accuracy: 0.9275
Epoch 142/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2371 - accura
cy: 0.9099 - val_loss: 0.2051 - val_accuracy: 0.9283
Epoch 143/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2368 - accura
cy: 0.9114 - val_loss: 0.2047 - val_accuracy: 0.9275
Epoch 144/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2365 - accura
cy: 0.9108 - val_loss: 0.2045 - val_accuracy: 0.9275
Epoch 145/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2362 - accura
cy: 0.9114 - val_loss: 0.2042 - val_accuracy: 0.9283
Epoch 146/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2359 - accura
cy: 0.9118 - val_loss: 0.2039 - val_accuracy: 0.9283
Epoch 147/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2356 - accura
cy: 0.9105 - val_loss: 0.2037 - val_accuracy: 0.9292
Epoch 148/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2353 - accura
cy: 0.9114 - val_loss: 0.2034 - val_accuracy: 0.9275
Epoch 149/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2350 - accura
cy: 0.9114 - val_loss: 0.2032 - val_accuracy: 0.9275
```

```
Epoch 150/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2348 - accura
cy: 0.9116 - val_loss: 0.2029 - val_accuracy: 0.9292
Epoch 151/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2345 - accura
cy: 0.9120 - val_loss: 0.2026 - val_accuracy: 0.9283
Epoch 152/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2343 - accura
cy: 0.9112 - val_loss: 0.2024 - val_accuracy: 0.9292
Epoch 153/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2340 - accura
cy: 0.9118 - val_loss: 0.2022 - val_accuracy: 0.9292
Epoch 154/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2337 - accura
cy: 0.9114 - val_loss: 0.2019 - val_accuracy: 0.9292
Epoch 155/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2334 - accura
cy: 0.9118 - val_loss: 0.2017 - val_accuracy: 0.9292
Epoch 156/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2332 - accura
cy: 0.9125 - val_loss: 0.2015 - val_accuracy: 0.9292
Epoch 157/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2329 - accura
cy: 0.9120 - val_loss: 0.2012 - val_accuracy: 0.9292
Epoch 158/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2326 - accura
cy: 0.9125 - val_loss: 0.2009 - val_accuracy: 0.9292
Epoch 159/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2324 - accura
cy: 0.9123 - val_loss: 0.2008 - val_accuracy: 0.9292
Epoch 160/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2321 - accura
cy: 0.9118 - val_loss: 0.2005 - val_accuracy: 0.9292
Epoch 161/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2319 - accura
cy: 0.9131 - val_loss: 0.2003 - val_accuracy: 0.9292
Epoch 162/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2316 - accura
cy: 0.9135 - val_loss: 0.2001 - val_accuracy: 0.9292
Epoch 163/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2314 - accura
cy: 0.9131 - val_loss: 0.1998 - val_accuracy: 0.9292
Epoch 164/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2311 - accura
cy: 0.9133 - val_loss: 0.1996 - val_accuracy: 0.9292
Epoch 165/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2309 - accura
cy: 0.9133 - val_loss: 0.1994 - val_accuracy: 0.9292
Epoch 166/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2307 - accura
cy: 0.9135 - val_loss: 0.1992 - val_accuracy: 0.9292
Epoch 167/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2304 - accura
cy: 0.9135 - val_loss: 0.1990 - val_accuracy: 0.9292
Epoch 168/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2302 - accura
cy: 0.9140 - val_loss: 0.1987 - val_accuracy: 0.9300
Epoch 169/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2299 - accura
cy: 0.9144 - val_loss: 0.1985 - val_accuracy: 0.9300
Epoch 170/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2297 - accura
cy: 0.9135 - val_loss: 0.1983 - val_accuracy: 0.9292
Epoch 171/300
```

```
147/147 [==============================] - 12s 83ms/step - loss: 0.2295 - accura
cy: 0.9146 - val_loss: 0.1981 - val_accuracy: 0.9283
Epoch 172/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2293 - accura
cy: 0.9144 - val_loss: 0.1979 - val_accuracy: 0.9292
Epoch 173/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2290 - accura
cy: 0.9144 - val_loss: 0.1977 - val_accuracy: 0.9283
Epoch 174/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2288 - accura
cy: 0.9144 - val_loss: 0.1975 - val_accuracy: 0.9292
Epoch 175/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2286 - accura
cy: 0.9150 - val_loss: 0.1973 - val_accuracy: 0.9292
Epoch 176/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2284 - accura
cy: 0.9142 - val_loss: 0.1972 - val_accuracy: 0.9292
Epoch 177/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2282 - accura
cy: 0.9152 - val_loss: 0.1969 - val_accuracy: 0.9292
Epoch 178/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2279 - accura
cy: 0.9148 - val_loss: 0.1967 - val_accuracy: 0.9292
Epoch 179/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2277 - accura
cy: 0.9150 - val_loss: 0.1966 - val_accuracy: 0.9300
Epoch 180/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2275 - accura
cy: 0.9150 - val_loss: 0.1963 - val_accuracy: 0.9292
Epoch 181/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2273 - accura
cy: 0.9142 - val_loss: 0.1961 - val_accuracy: 0.9292
Epoch 182/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2271 - accura
cy: 0.9150 - val_loss: 0.1959 - val_accuracy: 0.9292
Epoch 183/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2269 - accura
cy: 0.9146 - val_loss: 0.1958 - val_accuracy: 0.9292
Epoch 184/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2267 - accura
cy: 0.9150 - val_loss: 0.1956 - val_accuracy: 0.9292
Epoch 185/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2265 - accura
cy: 0.9152 - val_loss: 0.1954 - val_accuracy: 0.9292
Epoch 186/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2262 - accura
cy: 0.9146 - val_loss: 0.1952 - val_accuracy: 0.9300
Epoch 187/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2260 - accura
cy: 0.9148 - val_loss: 0.1950 - val_accuracy: 0.9300
Epoch 188/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2259 - accura
cy: 0.9148 - val_loss: 0.1949 - val_accuracy: 0.9292
Epoch 189/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2257 - accura
cy: 0.9146 - val_loss: 0.1947 - val_accuracy: 0.9292
Epoch 190/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2254 - accura
cy: 0.9155 - val_loss: 0.1945 - val_accuracy: 0.9300
Epoch 191/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2252 - accura
cy: 0.9150 - val_loss: 0.1944 - val_accuracy: 0.9300
Epoch 192/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2251 - accura
```

```
cy: 0.9150 - val_loss: 0.1942 - val_accuracy: 0.9300
Epoch 193/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2249 - accura
cy: 0.9150 - val_loss: 0.1940 - val_accuracy: 0.9300
Epoch 194/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2247 - accura
cy: 0.9157 - val_loss: 0.1939 - val_accuracy: 0.9292
Epoch 195/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2245 - accura
cy: 0.9157 - val_loss: 0.1937 - val_accuracy: 0.9300
Epoch 196/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2243 - accura
cy: 0.9150 - val_loss: 0.1936 - val_accuracy: 0.9300
Epoch 197/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2241 - accura
cy: 0.9150 - val_loss: 0.1934 - val_accuracy: 0.9300
Epoch 198/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2239 - accura
cy: 0.9157 - val_loss: 0.1932 - val_accuracy: 0.9300
Epoch 199/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2237 - accura
cy: 0.9155 - val_loss: 0.1930 - val_accuracy: 0.9300
Epoch 200/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2235 - accura
cy: 0.9155 - val_loss: 0.1929 - val_accuracy: 0.9300
Epoch 201/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2234 - accura
cy: 0.9159 - val_loss: 0.1927 - val_accuracy: 0.9292
Epoch 202/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2232 - accura
cy: 0.9155 - val_loss: 0.1926 - val_accuracy: 0.9292
Epoch 203/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2230 - accura
cy: 0.9155 - val_loss: 0.1924 - val_accuracy: 0.9292
Epoch 204/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2228 - accura
cy: 0.9161 - val_loss: 0.1923 - val_accuracy: 0.9292
Epoch 205/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2226 - accura
cy: 0.9157 - val_loss: 0.1922 - val_accuracy: 0.9292
Epoch 206/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2225 - accura
cy: 0.9161 - val_loss: 0.1920 - val_accuracy: 0.9283
Epoch 207/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2223 - accura
cy: 0.9155 - val_loss: 0.1918 - val_accuracy: 0.9283
Epoch 208/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2221 - accura
cy: 0.9155 - val_loss: 0.1917 - val_accuracy: 0.9283
Epoch 209/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2219 - accura
cy: 0.9161 - val_loss: 0.1915 - val_accuracy: 0.9300
Epoch 210/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2218 - accura
cy: 0.9159 - val_loss: 0.1913 - val_accuracy: 0.9283
Epoch 211/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2216 - accura
cy: 0.9157 - val_loss: 0.1912 - val_accuracy: 0.9292
Epoch 212/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2214 - accura
cy: 0.9155 - val_loss: 0.1911 - val_accuracy: 0.9292
Epoch 213/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2212 - accura
cy: 0.9157 - val_loss: 0.1909 - val_accuracy: 0.9283
```

```
Epoch 214/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2211 - accura
cy: 0.9157 - val_loss: 0.1908 - val_accuracy: 0.9283
Epoch 215/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2209 - accura
cy: 0.9152 - val_loss: 0.1907 - val_accuracy: 0.9283
Epoch 216/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2208 - accura
cy: 0.9155 - val_loss: 0.1906 - val_accuracy: 0.9283
Epoch 217/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2206 - accura
cy: 0.9161 - val_loss: 0.1904 - val_accuracy: 0.9283
Epoch 218/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2204 - accura
cy: 0.9152 - val_loss: 0.1904 - val_accuracy: 0.9292
Epoch 219/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2203 - accura
cy: 0.9163 - val_loss: 0.1902 - val_accuracy: 0.9283
Epoch 220/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2201 - accura
cy: 0.9159 - val_loss: 0.1900 - val_accuracy: 0.9283
Epoch 221/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2199 - accura
cy: 0.9155 - val_loss: 0.1899 - val_accuracy: 0.9283
Epoch 222/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2198 - accura
cy: 0.9150 - val_loss: 0.1897 - val_accuracy: 0.9283
Epoch 223/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2196 - accura
cy: 0.9165 - val_loss: 0.1896 - val_accuracy: 0.9283
Epoch 224/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2195 - accura
cy: 0.9170 - val_loss: 0.1895 - val_accuracy: 0.9283
Epoch 225/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2193 - accura
cy: 0.9163 - val_loss: 0.1894 - val_accuracy: 0.9283
Epoch 226/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2192 - accura
cy: 0.9159 - val_loss: 0.1892 - val_accuracy: 0.9283
Epoch 227/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2190 - accura
cy: 0.9155 - val_loss: 0.1891 - val_accuracy: 0.9283
Epoch 228/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2188 - accura
cy: 0.9165 - val_loss: 0.1890 - val_accuracy: 0.9283
Epoch 229/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2187 - accura
cy: 0.9161 - val_loss: 0.1889 - val_accuracy: 0.9283
Epoch 230/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2185 - accura
cy: 0.9159 - val_loss: 0.1887 - val_accuracy: 0.9283
Epoch 231/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2184 - accura
cy: 0.9165 - val_loss: 0.1886 - val_accuracy: 0.9283
Epoch 232/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2183 - accura
cy: 0.9161 - val_loss: 0.1885 - val_accuracy: 0.9283
Epoch 233/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2181 - accura
cy: 0.9167 - val_loss: 0.1883 - val_accuracy: 0.9283
Epoch 234/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2180 - accura
cy: 0.9161 - val_loss: 0.1882 - val_accuracy: 0.9275
Epoch 235/300
```

```
147/147 [==============================] - 12s 84ms/step - loss: 0.2178 - accura
cy: 0.9163 - val_loss: 0.1881 - val_accuracy: 0.9275
Epoch 236/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2177 - accura
cy: 0.9167 - val_loss: 0.1880 - val_accuracy: 0.9283
Epoch 237/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2175 - accura
cy: 0.9167 - val_loss: 0.1879 - val_accuracy: 0.9275
Epoch 238/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2173 - accura
cy: 0.9167 - val_loss: 0.1878 - val_accuracy: 0.9275
Epoch 239/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2172 - accura
cy: 0.9167 - val_loss: 0.1876 - val_accuracy: 0.9275
Epoch 240/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2171 - accura
cy: 0.9172 - val_loss: 0.1876 - val_accuracy: 0.9275
Epoch 241/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2169 - accura
cy: 0.9165 - val_loss: 0.1874 - val_accuracy: 0.9275
Epoch 242/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2168 - accura
cy: 0.9172 - val_loss: 0.1873 - val_accuracy: 0.9275
Epoch 243/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2167 - accura
cy: 0.9178 - val_loss: 0.1871 - val_accuracy: 0.9283
Epoch 244/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2165 - accura
cy: 0.9174 - val_loss: 0.1870 - val_accuracy: 0.9283
Epoch 245/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2164 - accura
cy: 0.9180 - val_loss: 0.1870 - val_accuracy: 0.9283
Epoch 246/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2162 - accura
cy: 0.9178 - val_loss: 0.1869 - val_accuracy: 0.9283
Epoch 247/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2161 - accura
cy: 0.9176 - val_loss: 0.1867 - val_accuracy: 0.9283
Epoch 248/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2160 - accura
cy: 0.9178 - val_loss: 0.1866 - val_accuracy: 0.9292
Epoch 249/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2158 - accura
cy: 0.9174 - val_loss: 0.1865 - val_accuracy: 0.9283
Epoch 250/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2157 - accura
cy: 0.9178 - val_loss: 0.1863 - val_accuracy: 0.9283
Epoch 251/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2155 - accura
cy: 0.9178 - val_loss: 0.1863 - val_accuracy: 0.9292
Epoch 252/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2154 - accura
cy: 0.9178 - val_loss: 0.1862 - val_accuracy: 0.9292
Epoch 253/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2153 - accura
cy: 0.9180 - val_loss: 0.1861 - val_accuracy: 0.9292
Epoch 254/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2151 - accura
cy: 0.9180 - val_loss: 0.1860 - val_accuracy: 0.9292
Epoch 255/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2150 - accura
cy: 0.9182 - val_loss: 0.1859 - val_accuracy: 0.9292
Epoch 256/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2149 - accura
```

```
cy: 0.9178 - val_loss: 0.1858 - val_accuracy: 0.9292
Epoch 257/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2147 - accura
cy: 0.9182 - val_loss: 0.1857 - val_accuracy: 0.9292
Epoch 258/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2146 - accura
cy: 0.9176 - val_loss: 0.1856 - val_accuracy: 0.9292
Epoch 259/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2145 - accura
cy: 0.9180 - val_loss: 0.1855 - val_accuracy: 0.9292
Epoch 260/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2144 - accura
cy: 0.9178 - val_loss: 0.1854 - val_accuracy: 0.9292
Epoch 261/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2142 - accura
cy: 0.9180 - val_loss: 0.1852 - val_accuracy: 0.9292
Epoch 262/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2141 - accura
cy: 0.9182 - val_loss: 0.1851 - val_accuracy: 0.9292
Epoch 263/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2140 - accura
cy: 0.9182 - val_loss: 0.1851 - val_accuracy: 0.9292
Epoch 264/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2138 - accura
cy: 0.9180 - val_loss: 0.1850 - val_accuracy: 0.9292
Epoch 265/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2138 - accura
cy: 0.9180 - val_loss: 0.1849 - val_accuracy: 0.9292
Epoch 266/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2136 - accura
cy: 0.9182 - val_loss: 0.1848 - val_accuracy: 0.9292
Epoch 267/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2135 - accura
cy: 0.9187 - val_loss: 0.1846 - val_accuracy: 0.9292
Epoch 268/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2134 - accura
cy: 0.9182 - val_loss: 0.1845 - val_accuracy: 0.9292
Epoch 269/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2132 - accura
cy: 0.9187 - val_loss: 0.1844 - val_accuracy: 0.9292
Epoch 270/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2131 - accura
cy: 0.9184 - val_loss: 0.1843 - val_accuracy: 0.9292
Epoch 271/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2130 - accura
cy: 0.9189 - val_loss: 0.1844 - val_accuracy: 0.9292
Epoch 272/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2128 - accura
cy: 0.9187 - val_loss: 0.1841 - val_accuracy: 0.9292
Epoch 273/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2127 - accura
cy: 0.9187 - val_loss: 0.1841 - val_accuracy: 0.9292
Epoch 274/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2126 - accura
cy: 0.9178 - val_loss: 0.1840 - val_accuracy: 0.9292
Epoch 275/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2125 - accura
cy: 0.9182 - val_loss: 0.1840 - val_accuracy: 0.9292
Epoch 276/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2124 - accura
cy: 0.9193 - val_loss: 0.1839 - val_accuracy: 0.9292
Epoch 277/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2123 - accura
cy: 0.9182 - val_loss: 0.1838 - val_accuracy: 0.9292
```

```
Epoch 278/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2121 - accura
cy: 0.9197 - val_loss: 0.1837 - val_accuracy: 0.9292
Epoch 279/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2120 - accura
cy: 0.9187 - val_loss: 0.1836 - val_accuracy: 0.9292
Epoch 280/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2119 - accura
cy: 0.9189 - val_loss: 0.1836 - val_accuracy: 0.9292
Epoch 281/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2118 - accura
cy: 0.9195 - val_loss: 0.1834 - val_accuracy: 0.9292
Epoch 282/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2117 - accura
cy: 0.9189 - val_loss: 0.1834 - val_accuracy: 0.9292
Epoch 283/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2115 - accura
cy: 0.9197 - val_loss: 0.1833 - val_accuracy: 0.9292
Epoch 284/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2115 - accura
cy: 0.9189 - val_loss: 0.1832 - val_accuracy: 0.9292
Epoch 285/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2113 - accura
cy: 0.9197 - val_loss: 0.1830 - val_accuracy: 0.9292
Epoch 286/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2112 - accura
cy: 0.9199 - val_loss: 0.1830 - val_accuracy: 0.9292
Epoch 287/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2111 - accura
cy: 0.9195 - val_loss: 0.1829 - val_accuracy: 0.9292
Epoch 288/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2110 - accura
cy: 0.9195 - val_loss: 0.1828 - val_accuracy: 0.9292
Epoch 289/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2109 - accura
cy: 0.9191 - val_loss: 0.1826 - val_accuracy: 0.9317
Epoch 290/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2108 - accura
cy: 0.9195 - val_loss: 0.1826 - val_accuracy: 0.9309
Epoch 291/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2107 - accura
cy: 0.9202 - val_loss: 0.1824 - val_accuracy: 0.9317
Epoch 292/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2106 - accura
cy: 0.9195 - val_loss: 0.1824 - val_accuracy: 0.9309
Epoch 293/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2104 - accura
cy: 0.9199 - val_loss: 0.1823 - val_accuracy: 0.9309
Epoch 294/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2103 - accura
cy: 0.9199 - val_loss: 0.1822 - val_accuracy: 0.9317
Epoch 295/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2102 - accura
cy: 0.9197 - val_loss: 0.1822 - val_accuracy: 0.9300
Epoch 296/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2101 - accura
cy: 0.9202 - val_loss: 0.1821 - val_accuracy: 0.9309
Epoch 297/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2100 - accura
cy: 0.9202 - val_loss: 0.1820 - val_accuracy: 0.9300
Epoch 298/300
147/147 [==============================] - 12s 84ms/step - loss: 0.2099 - accura
cy: 0.9197 - val_loss: 0.1819 - val_accuracy: 0.9300
Epoch 299/300
```

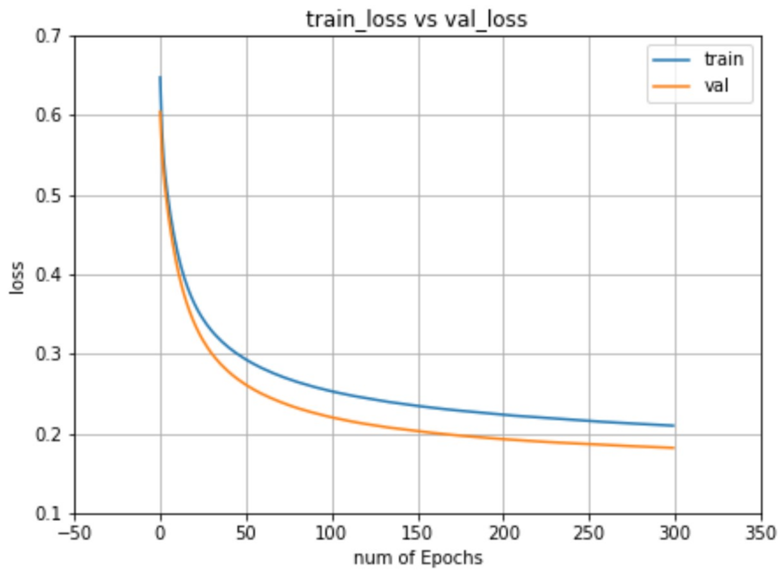
```
147/147 [==============================] - 12s 83ms/step - loss: 0.2098 - accura
cy: 0.9206 - val_loss: 0.1819 - val_accuracy: 0.9300
Epoch 300/300
147/147 [==============================] - 12s 83ms/step - loss: 0.2097 - accura
```

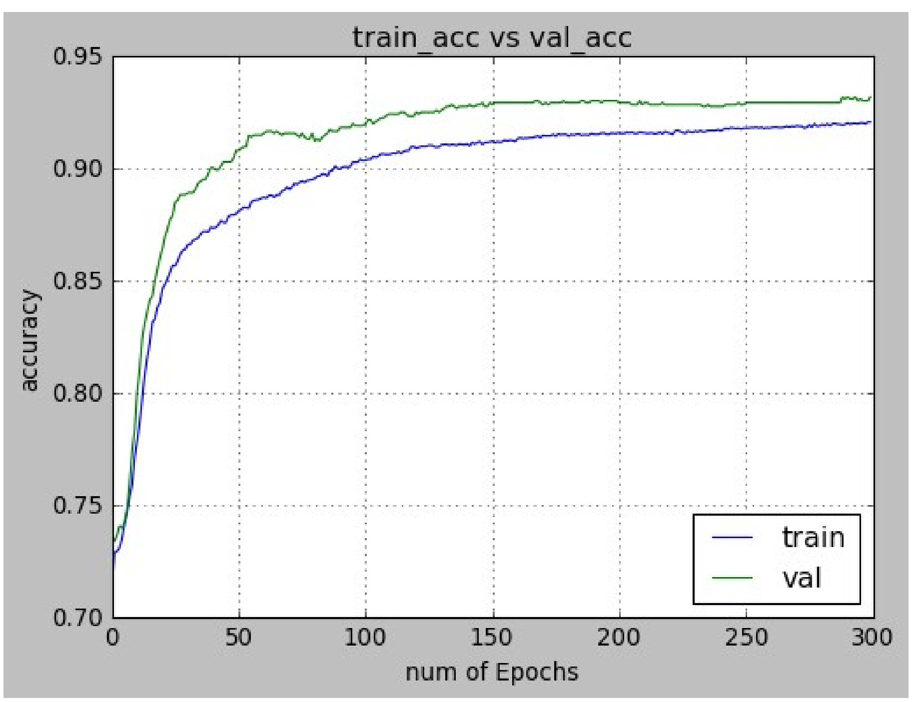
In [15]:
```
(loss, accuracy) = custom_resnet_model.evaluate(X_test, y_test, batch_size=10, verb
ose=1)

print("[INFO] loss={:.4f}, accuracy: {:.4f}%".format(loss,accuracy * 100))
```

```
118/118 [==============================] - 4s 31ms/step - loss: 0.1818 - accurac
y: 0.9317
[INFO] loss=0.1818, accuracy: 93.1741%
```

visualizing losses and accuracy

In [16]:
```
display_loss_accuracy(hist)
```





Evaluating the model

In [17]:
```
score = custom_resnet_model.evaluate(X_test, y_test, verbose=0)
print('Test Loss:', score[0])
print('Test accuracy:', score[1])

test_image = X_test[0:1]
print (test_image.shape)

print(model.predict(test_image))
print(model.predict_classes(test_image))
print(y_test[0:1])
```

```
Test Loss: 0.18175366520881653
Test accuracy: 0.9317406415939331
(1, 224, 224, 3)
[[1.60319269e-05 1.03313741e-05 1.10294837e-04 1.62154553e-04
  1.54952664e-04 3.75904006e-06 1.45744079e-05 1.49391735e-06
  7.57235000e-07 9.59678005e-07 2.06255368e-06 4.56184489e-06
  1.86742704e-06 5.95964275e-06 4.45468061e-07 1.67001065e-06
  4.04327830e-06 2.47253524e-06 4.92232402e-06 1.51862960e-05
  6.89771525e-07 1.77929940e-06 2.35125708e-06 7.93170329e-06
  3.36765743e-06 4.09337645e-07 9.40696225e-07 1.42265390e-06
  3.05310095e-06 1.17761365e-05 1.59667673e-06 4.37783820e-06
  1.06327821e-06 8.20454170e-06 9.17864963e-06 9.03331909e-07
  3.70212774e-06 1.50403264e-06 7.12718975e-06 6.36542381e-06
  1.66181053e-05 1.08479242e-06 2.40343957e-06 1.54460020e-06
  2.00442173e-06 2.86294403e-06 1.11290256e-05 2.00422119e-06
  2.27432247e-06 1.55147627e-05 4.49494037e-05 4.17682313e-05
  5.78564004e-06 6.57222654e-06 1.30594326e-05 4.42919827e-06
  6.07707670e-06 1.76240144e-06 3.19003698e-06 3.83554652e-06
  1.56950391e-05 3.07739174e-05 1.43814004e-05 4.24008431e-06
  4.80574045e-06 3.03465595e-06 3.81626342e-06 7.26074995e-06
  1.51082668e-06 3.48639660e-05 2.34314507e-06 1.89028760e-05
  5.30490979e-06 9.16534020e-07 8.63349101e-07 3.58039392e-06
  6.97309861e-06 3.29716181e-06 6.22108337e-06 2.37078239e-05
  1.79604956e-06 1.88003901e-06 6.68061887e-07 6.39696225e-07
  1.57687123e-06 7.92699552e-07 1.66018594e-06 7.19225063e-05
  1.16075262e-05 5.61878260e-05 1.65175732e-06 4.02936303e-06
  7.42185136e-07 6.60048636e-06 1.61907292e-06 1.24259429e-06
  2.79969026e-06 9.39246661e-07 2.60249891e-07 3.26425106e-06
  5.19502964e-06 1.01820755e-04 1.48046627e-06 1.14670820e-06
  1.17285413e-06 3.59738419e-06 7.95398773e-06 2.11877486e-05
  4.10908797e-05 6.67888344e-06 7.93686286e-06 2.56585947e-04
  1.64501209e-04 3.13655414e-06 7.33983870e-06 2.06123241e-06
  6.07913398e-06 1.79852559e-05 3.00856755e-05 1.47011121e-06
  1.61154605e-06 8.46889907e-06 2.63959046e-05 1.69565046e-05
  4.48179926e-05 5.52515417e-07 1.45818674e-04 3.65918868e-06
  3.45736044e-06 8.83706889e-06 5.66262133e-06 2.56274848e-06
  1.06754724e-05 6.25442590e-06 7.71363011e-06 4.60823600e-07
  1.24716871e-06 1.88098784e-06 3.70481553e-06 2.89821259e-07
  4.86155432e-06 1.07358881e-06 5.60310866e-07 7.11699431e-06
  1.09040229e-05 1.21046605e-05 3.63108484e-05 6.77904245e-06
  1.02750528e-05 9.26039138e-06 2.47772141e-06 2.37317345e-05
  6.35963579e-06 1.03248196e-04 8.50519154e-06 2.92284985e-05
  1.37217421e-05 1.07453570e-05 1.28756401e-05 4.24032150e-05
  6.60260557e-05 5.38223794e-05 2.00800459e-05 6.52082963e-05
  1.41905957e-05 2.18924288e-05 2.63704915e-05 1.47886958e-05
  2.29550624e-05 1.65246020e-04 2.12509694e-05 1.48845153e-04
  3.64316104e-04 6.53028328e-05 2.95670270e-05 1.36863118e-05
  3.05038993e-05 2.69443826e-05 2.86597235e-04 9.67894539e-06
  2.92806981e-05 1.83899338e-05 1.30747612e-05 2.11740717e-05
  1.26354325e-05 6.16543366e-06 7.33729848e-06 1.82500244e-05
  2.38655266e-05 6.79423874e-06 1.64281482e-05 4.80749368e-05
  4.15156173e-05 1.55071593e-05 3.40504812e-06 6.10024144e-05
  8.19202905e-05 1.00648998e-04 3.41478917e-05 1.72595916e-04
  3.97444601e-05 1.52323328e-05 4.09904533e-05 2.49311648e-04
  4.68228754e-05 1.27179910e-05 5.25914584e-06 8.89047624e-06
  5.79922089e-05 3.30010771e-05 9.93528283e-06 1.23883701e-05
  2.54203296e-05 2.39899464e-06 1.29629452e-05 1.00658362e-05
  1.07877640e-05 5.44915783e-05 4.05032961e-06 2.99432686e-05
  7.04572085e-06 4.85013879e-06 5.26463264e-05 8.78974788e-06
  1.70346939e-05 8.65449965e-06 2.41151101e-05 8.95348239e-06
  6.93146831e-06 5.34416904e-05 3.21619918e-05 3.44353175e-05
  1.38240357e-05 2.07384455e-05 3.19933388e-05 1.97944100e-05
  2.83556419e-05 4.37624221e-05 3.52999014e-05 2.55983814e-05
```

```
6.91186960e-06 4.02791102e-06 9.31724135e-05 4.79334740e-05
2.71028125e-06 3.91251087e-05 2.39291243e-04 1.50379956e-05
3.90658279e-05 2.80563072e-05 6.19828643e-05 3.14953795e-04
1.88501126e-05 4.90171915e-05 5.85773414e-05 6.55306349e-06
3.15395009e-05 1.73074965e-04 6.98888762e-05 1.87221922e-05
3.05717117e-06 9.39800793e-06 8.83249811e-07 1.26066780e-05
1.43866546e-05 1.63504828e-05 1.56728438e-05 1.33058356e-05
2.65211952e-06 3.34704418e-06 2.13711610e-05 1.63754260e-06
2.65783683e-06 1.74572615e-05 4.30745359e-07 4.26765610e-07
9.59851036e-07 6.53064376e-07 2.67874753e-07 1.44172338e-06
1.35773098e-06 2.05589931e-05 2.77315303e-05 2.86596423e-05
1.06678133e-04 5.00159695e-05 3.93999653e-06 4.92581739e-06
2.39194242e-06 2.25244366e-06 1.35650680e-06 1.43068519e-06
1.04239589e-05 1.38163966e-06 1.08233712e-06 4.14736860e-06
5.67871621e-06 1.47708170e-06 1.07638755e-06 1.02625172e-05
3.01192330e-07 4.05845321e-06 1.29402486e-06 4.89679724e-06
1.08715381e-06 1.51738433e-07 3.79183393e-06 9.64056198e-07
9.59223257e-07 2.25460576e-06 3.51004473e-06 5.24022153e-06
6.43917974e-06 7.86925448e-05 2.17878078e-05 2.77105391e-05
4.60551428e-06 6.35121660e-06 8.61789886e-06 2.23393499e-06
1.65569384e-06 5.22420578e-07 1.24700455e-06 7.50749734e-07
1.89000912e-06 8.52049823e-07 4.47671567e-07 2.56580406e-05
3.93660093e-06 3.14198078e-05 1.40381508e-06 9.37569496e-07
1.13737178e-05 1.29104010e-05 2.91128958e-06 5.36240520e-07
6.53219445e-07 9.43673058e-07 1.54857444e-05 2.16202648e-06
1.66223672e-05 2.02626197e-05 7.37217476e-07 5.06860488e-07
8.25593725e-06 7.31602222e-06 1.06559571e-06 8.69236317e-07
6.76275022e-06 5.61255376e-07 1.09636824e-06 2.21486835e-06
2.15010868e-06 2.13603948e-06 2.06479558e-06 9.71767577e-07
7.00482497e-06 5.50537266e-07 1.97207919e-05 2.31398899e-05
3.58857591e-07 1.45133945e-05 2.83409986e-06 9.86642954e-06
1.86196041e-06 3.11344843e-06 1.81955670e-06 3.49242464e-06
3.56492887e-06 3.90437981e-06 2.35651009e-06 7.06303081e-06
2.74208014e-06 1.17179320e-06 1.38298685e-06 9.07817957e-06
2.35450852e-06 3.84195664e-06 2.93031712e-06 1.80563461e-06
5.22406435e-06 6.29523413e-07 2.08774236e-06 2.10029157e-06
9.10378503e-07 1.11642912e-04 3.44472428e-05 1.63438386e-07
3.34854803e-06 1.31891895e-04 8.65294423e-05 2.32978695e-04
1.56588953e-06 4.38277821e-05 2.48833006e-04 1.03217273e-04
2.74880085e-06 1.60974594e-06 4.65702396e-05 2.30849311e-02
7.33014778e-04 1.18834767e-04 1.01327016e-04 1.39592771e-06
3.81466930e-06 5.16925274e-06 1.29832115e-04 1.06388634e-05
2.52194877e-06 2.73809768e-04 1.39873764e-05 4.79292031e-03
5.19143214e-05 3.89142697e-05 4.47466085e-03 1.39371434e-04
2.21995961e-05 3.19195569e-05 9.68719760e-05 1.86327510e-04
8.25277675e-05 3.54409538e-04 9.11652314e-05 1.50567948e-05
6.88548462e-05 1.83142811e-06 1.84519828e-04 4.16005096e-05
2.42814622e-06 2.04729076e-05 1.30038425e-05 2.14210749e-02
1.61065123e-04 1.33468784e-04 6.67014765e-03 3.56896897e-03
9.52569826e-06 9.82693746e-05 8.53714810e-05 2.26096327e-05
1.96198773e-04 1.44435267e-04 2.15311899e-04 1.38524896e-03
2.38698385e-06 1.25098726e-04 5.71421522e-04 1.29172535e-04
1.54385270e-05 9.55530936e-07 1.26648942e-04 1.99522168e-04
5.56244107e-04 3.03735051e-05 2.17826164e-05 9.82563506e-05
1.05477535e-04 2.97599612e-03 1.87792193e-05 1.63793471e-03
6.92060257e-06 3.42617254e-03 3.59340484e-05 4.41557750e-05
2.18985573e-04 4.62186872e-04 1.06219784e-04 1.36412127e-04
9.15578767e-06 1.53589044e-05 3.31807852e-04 2.41510843e-05
4.27105397e-06 5.71528333e-04 1.56241562e-02 9.68999302e-06
8.53931488e-05 1.82322896e-04 2.06790908e-04 7.28846586e-04
5.50588702e-05 1.10126266e-05 9.43327996e-06 2.99689254e-05
3.79536823e-06 2.04554981e-05 4.49863728e-05 1.64006371e-04
3.36227131e-05 4.26836086e-05 4.11208486e-04 4.47737966e-06
5.21015818e-06 3.16495389e-05 3.07591603e-04 3.94455128e-05
```

```
1.87475394e-04 1.55559290e-04 9.36624128e-05 1.12943584e-03
1.56399201e-05 2.59460025e-02 9.40578349e-04 5.73464611e-04
1.98426715e-04 3.04019515e-04 1.06723928e-04 2.51947167e-05
4.31178596e-05 3.38464015e-05 2.71007457e-06 1.72321888e-05
1.35378636e-04 4.44797159e-04 6.87745807e-04 3.96852440e-04
6.48850482e-03 1.03746088e-05 7.67391030e-05 2.88501742e-05
1.81053008e-03 5.86810056e-05 6.79024879e-06 7.10307807e-03
2.32559140e-03 1.26185914e-04 7.64987362e-06 1.55565431e-05
6.81042147e-05 7.83646363e-04 3.10840624e-05 8.55247708e-05
2.97965089e-05 6.35390970e-05 2.29425333e-03 5.15807151e-05
1.23254586e-06 6.86643853e-06 5.29439421e-05 4.04479215e-05
2.87437706e-06 1.11579953e-04 8.27618176e-04 2.66439223e-04
5.57875046e-06 4.93174266e-05 9.40265672e-05 1.67889084e-06
3.47784044e-06 1.88498510e-04 3.51324816e-05 4.73838008e-05
8.67693801e-04 2.37871118e-05 1.97406061e-05 2.43600903e-06
4.70135310e-05 4.52559150e-04 3.42104293e-04 2.72877456e-04
1.63843397e-05 4.71878366e-06 3.50007213e-05 2.91325472e-04
5.40781242e-04 1.44803662e-05 2.36022897e-04 1.45900747e-04
7.97500554e-03 2.01737021e-06 2.59742141e-03 1.85973586e-05
7.97404427e-05 9.66398943e-07 1.03528155e-05 1.06674315e-05
1.12890712e-05 1.19831413e-04 8.37275684e-02 1.29231441e-04
1.70508647e-05 2.87992661e-05 4.00853423e-05 8.35987794e-06
5.89391093e-05 1.23447587e-03 2.99343469e-06 7.17704359e-04
1.23988732e-03 2.99835787e-03 4.24182736e-06 6.17826311e-03
2.04555945e-05 6.91490641e-05 1.77771726e-04 1.67307292e-06
1.78289731e-04 1.86381891e-04 3.06895963e-05 3.26779118e-05
4.01882106e-04 4.33941232e-03 8.04528026e-05 1.02534239e-06
2.33694343e-04 1.59130359e-04 1.42672670e-03 2.42684400e-05
4.81151184e-03 3.94198059e-06 3.86178493e-02 3.95126244e-06
2.50421340e-06 2.59493518e-05 2.22984492e-03 1.02647592e-03
2.27303572e-05 9.96093731e-03 3.27270478e-04 9.17604135e-04
8.18816334e-05 4.81259258e-06 5.70895281e-05 1.91486848e-04
2.92165696e-05 5.58253987e-06 1.18716052e-04 4.21267672e-04
9.16961108e-06 1.55119676e-04 7.28833373e-04 5.19975380e-04
1.10635272e-04 1.81569394e-05 5.91371600e-06 4.11449355e-06
4.91292623e-04 4.21108962e-06 2.70571187e-03 8.38333450e-04
1.61232401e-06 2.11122213e-04 8.00290945e-05 1.50961347e-03
1.74654153e-04 5.44562772e-06 2.05236938e-05 1.97415371e-04
2.53793318e-04 1.33682861e-05 3.91371839e-04 6.70430454e-05
4.88777645e-04 2.43980940e-05 4.04712664e-05 1.49456365e-03
6.74199910e-05 2.68967473e-04 1.45328551e-04 6.63232349e-05
2.54017505e-06 1.31187687e-06 2.59787193e-05 1.37970244e-04
2.26249267e-05 3.92745869e-06 1.46454531e-05 1.69466512e-04
8.56776678e-05 1.81699591e-03 1.08780368e-05 4.97994733e-06
1.62041993e-04 5.61861088e-05 5.07855111e-06 5.07659979e-06
1.19888564e-05 3.31579322e-05 1.39716663e-03 6.23712549e-05
9.93385547e-05 6.56091433e-05 7.90486502e-06 2.03377364e-04
3.20675390e-05 5.83116662e-06 4.32214401e-05 2.12390046e-03
6.31318426e-06 1.50280260e-02 9.29719022e-07 6.27027650e-04
3.05942912e-03 5.14462554e-05 7.06916944e-06 4.39039786e-06
6.65783416e-04 1.93485171e-02 9.86861778e-05 4.74226341e-04
6.37904915e-04 1.45694603e-05 2.01414659e-04 1.18259766e-06
1.04375576e-05 5.89896308e-06 1.77756374e-05 1.72302171e-05
6.13324984e-04 1.48846710e-04 3.88127319e-06 1.79723807e-04
1.54201873e-04 1.57496779e-05 2.65244726e-05 3.06731363e-06
9.26564462e-05 6.44701117e-07 1.78780519e-05 1.17702084e-05
1.14719827e-04 5.55849122e-03 7.39918760e-05 2.70640180e-06
1.10150413e-05 1.09544804e-03 2.81733014e-06 2.58291602e-05
1.49531208e-03 5.24916977e-04 8.03333478e-07 4.18251293e-04
6.84956422e-06 3.57130280e-04 1.86888622e-06 9.28397197e-03
1.99775877e-05 2.01856150e-04 5.25498990e-06 4.34743597e-05
1.13690272e-04 3.36795347e-04 2.83585596e-05 1.32007219e-04
4.04204853e-04 8.22601360e-05 1.73680055e-05 1.19544903e-03
5.08963771e-04 5.73876605e-04 2.80326675e-03 5.29872113e-06
```

```
 6.34234020e-05 9.31100640e-03 1.15069088e-05 8.17254022e-06
 1.48944264e-05 3.28881615e-05 3.82455037e-05 4.84449265e-05
 1.18228479e-03 1.01485915e-04 4.90250459e-05 3.71651295e-05
 6.26636393e-05 3.40368933e-05 9.38845915e-06 3.84880186e-05
 1.07918077e-05 5.09490637e-05 4.74274246e-04 5.21298352e-05
 2.33722749e-04 3.92459071e-04 2.30986261e-04 3.18218139e-03
 4.43626894e-04 2.07800214e-04 3.85329186e-04 1.12457656e-05
 5.87678696e-05 2.66920142e-06 1.82133226e-05 8.22053262e-05
 1.84634337e-04 2.62670219e-03 7.81100171e-05 9.39066667e-05
 3.13199591e-04 3.01090273e-04 5.37138731e-05 6.35960241e-05
 1.54895039e-04 1.00662920e-03 3.09729844e-01 1.04010753e-04
 3.59318394e-04 6.47318363e-03 6.84189990e-06 3.21661209e-04
 5.00451870e-06 1.94857948e-05 2.14344686e-06 1.72386081e-06
 6.61506565e-05 3.48080175e-05 1.08291593e-03 1.82282292e-05
 5.96208447e-05 3.63468098e-05 5.12238194e-06 2.63500144e-04
 2.63440365e-04 8.21180292e-04 1.22557358e-05 8.05696982e-06
 7.89780897e-05 5.47358286e-06 5.82633947e-05 6.77239004e-05
 2.65229164e-06 1.03666980e-05 6.36750992e-05 3.99242475e-04
 3.01652439e-02 3.46966772e-06 1.04584224e-05 1.11964437e-05
 1.49201805e-04 4.38907500e-06 2.48686876e-04 1.77349051e-04
 1.41802338e-05 9.16827321e-06 9.48329270e-03 1.37502536e-06
 1.81126670e-04 3.82784900e-04 1.27218125e-04 2.37650711e-05
 5.14602172e-04 3.48264836e-02 1.86458509e-03 1.21830190e-05
 5.58678148e-05 1.00676937e-03 1.91992905e-04 5.15187003e-06
 1.48921254e-05 3.18947241e-05 2.87434268e-05 1.41469782e-05
 1.01306678e-05 1.19616525e-05 2.78161466e-03 8.90330193e-05
 5.31392971e-07 3.21876105e-05 5.89872470e-05 1.69842708e-04
 3.64147127e-05 3.87919572e-04 7.62617565e-05 3.08204239e-04
 8.07548986e-07 1.87526784e-05 1.11185345e-05 7.41962594e-06
 7.28229701e-04 4.49476726e-02 1.08206052e-06 5.29359932e-06
 1.37103416e-04 1.36150236e-04 6.83998451e-07 2.86124792e-04
 1.44213683e-03 1.48466352e-04 9.94783477e-06 5.66250492e-05
 1.71225292e-05 5.13872146e-05 5.65154594e-04 8.92300508e-04
 1.51820318e-03 2.09048912e-02 4.05425817e-05 1.77717749e-02
 2.40463523e-05 1.04626270e-04 4.17706769e-06 3.33191529e-05
 3.05257126e-04 6.22556690e-05 2.06982009e-02 2.15643486e-05
 7.93905114e-04 1.46306981e-03 7.75219349e-04 1.36577047e-03
 2.20662150e-05 1.15992816e-05 4.50448388e-05 1.15255907e-03
 2.22373230e-04 1.55445095e-02 2.91565363e-03 2.61124893e-04
 1.30278995e-05 4.06940380e-05 2.36637215e-03 2.67326622e-03
 8.59521151e-06 7.82948121e-07 8.84162182e-06 2.24221112e-05
 8.16251486e-06 5.58746360e-05 2.65364302e-04 7.93287563e-06
 9.85665883e-06 1.49677522e-04 2.33207305e-04 1.50661537e-04
 1.48810086e-05 1.09221228e-05 3.59100818e-06 2.67928681e-05
 2.73309779e-05 1.55229995e-04 3.31500749e-04 6.38112106e-05
 2.72561429e-05 1.00200941e-05 3.33330419e-04 7.60209659e-05
 1.11031120e-04 5.91472235e-05 3.83362130e-05 1.05969972e-04
 9.11242478e-06 4.64681762e-06 6.08663177e-05 6.93005641e-05
 1.18161861e-05 5.97667213e-06 2.23482130e-06 3.17308659e-05
 2.65495783e-05 4.25018561e-06 1.20645427e-05 7.67836445e-06
 4.83140093e-06 3.99348937e-05 5.04005817e-04 4.72208594e-05
 3.45014887e-05 3.73107127e-06 6.86786598e-06 2.32253387e-05
 3.88046865e-05 4.06940992e-04 3.26363333e-05 3.11813310e-05
 2.15266427e-05 4.23675578e-04 1.70035884e-04 3.32325180e-05
 2.95545105e-05 2.93031007e-05 1.94241984e-05 5.07530967e-05
 4.13134403e-05 4.67299333e-06 6.01951660e-06 4.72595821e-06
 6.86714475e-06 1.28987900e-04 4.10891334e-06 7.82038205e-06
 6.52552353e-06 3.18396305e-05 2.07620882e-03 6.62207458e-06
 1.99378360e-06 1.10867059e-05 5.60765272e-07 3.37390666e-04
 1.75066850e-06 4.20925289e-07 2.73975115e-06 6.46207718e-06
 2.46186028e-06 4.78918480e-07 3.05887738e-06 9.52664777e-07
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-17-83421ec204f5> in <module>
      7
      8 print(model.predict(test_image))
----> 9 print(model.predict_classes(test_image))
```

Testing a new image

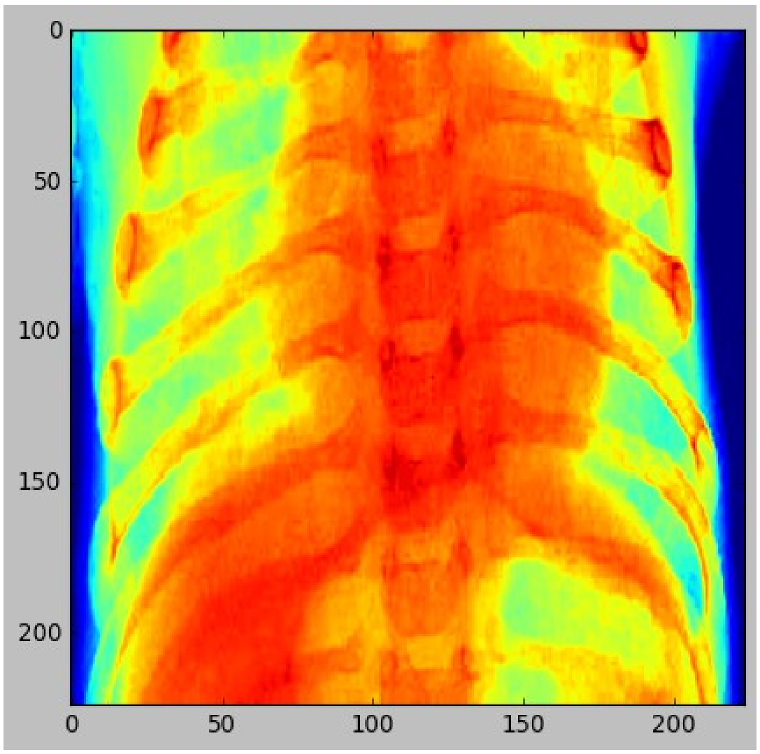
In [18]:
```
test_image_path = 'D:/Harold/MyDNN/DataSet/Chest_xray_seperate/PNEUMONIA/person11_b
acteria_45.jpeg'
test_image = image.load_img(test_image_path, target_size=(224, 224))
x = image.img_to_array(test_image)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)
print (x.shape)


# if num_channel==1:
#     if (K.image_data_format() == 'channels_first'):
#         test_image= np.expand_dims(test_image, axis=0)
#         test_image= np.expand_dims(test_image, axis=0)
#         print (test_image.shape)
#     else:
#         test_image= np.expand_dims(test_image, axis=3)
#         test_image= np.expand_dims(test_image, axis=0)
#         print (test_image.shape)
# else:
#     if (K.image_data_format() == 'channels_first'):
#         test_image=np.rollaxis(test_image,2,0)
#         test_image= np.expand_dims(test_image, axis=0)
#         print (test_image.shape)
#     else:
#         test_image= np.expand_dims(test_image, axis=0)
#         print (test_image.shape)


# Predicting the test image
yhat = custom_resnet_model.predict(x)
print(yhat)
# print(custom_resnet_model.predict_classes(x))
label = decode_predictions(yhat)
# retrieve the most likely result, e.g. highest probability
label = label[0][0]
```

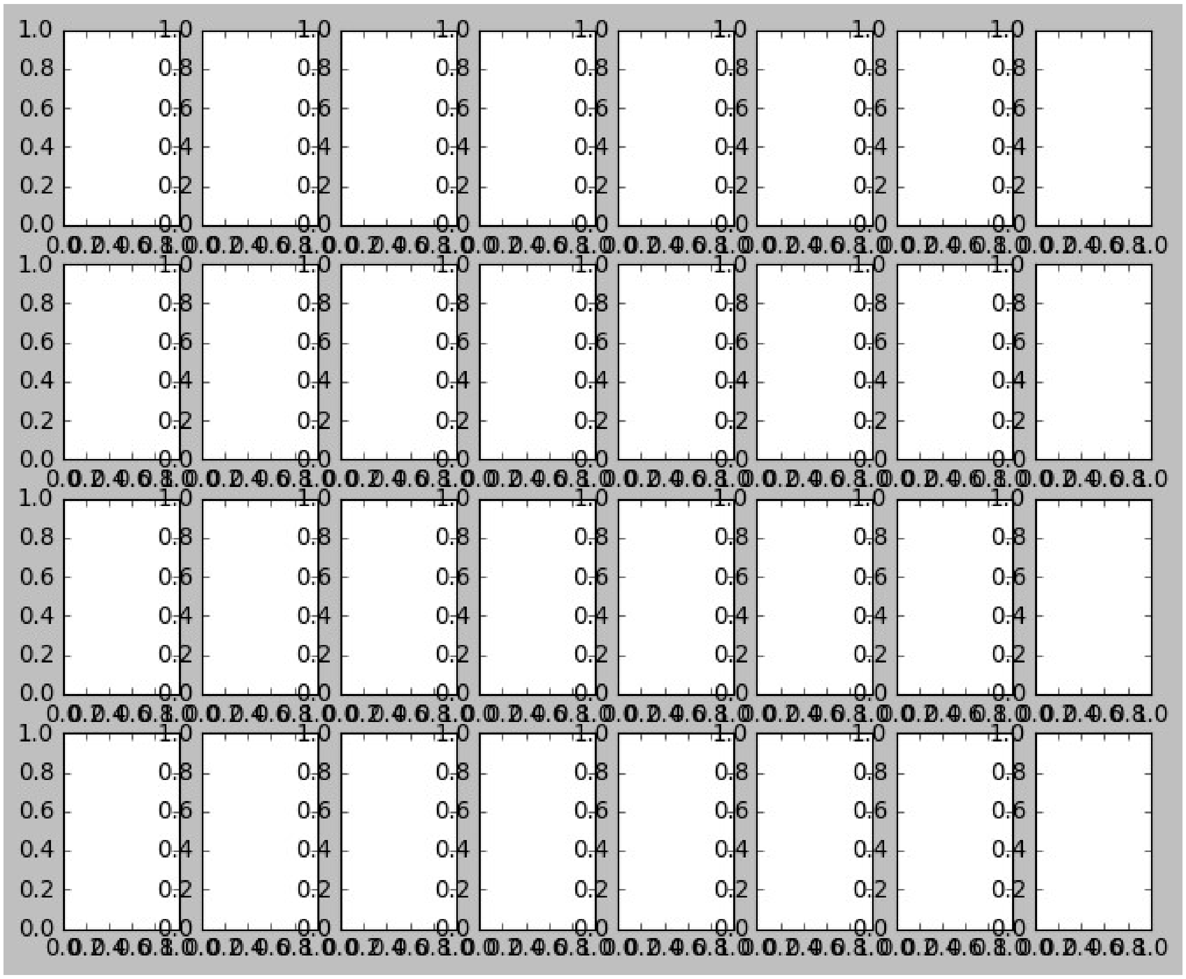
```
(1, 224, 224, 3)
[[0.00989007 0.99010986]]

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-18-505048f79341> in <module>
     30 print(yhat)
     31 # print(custom_resnet_model.predict_classes(x))
---> 32 label = decode_predictions(yhat)
     33 # retrieve the most likely result, e.g. highest probability
     34 label = label[0][0]

D:\Anaconda3\lib\site-packages\tensorflow\python\keras\applications\inception_v
3.py in decode_predictions(preds, top)
    412 @keras_export('keras.applications.inception_v3.decode_predictions')
    413 def decode_predictions(preds, top=5):
--> 414   return imagenet_utils.decode_predictions(preds, top=top)
    415
    416

D:\Anaconda3\lib\site-packages\tensorflow\python\keras\applications\imagenet_uti
ls.py in decode_predictions(preds, top)
    149                      'a batch of predictions '
    150                      '(i.e. a 2D array of shape (samples, 1000)). '
--> 151                      'Found array with shape: ' + str(preds.shape))
    152   if CLASS_INDEX is None:
    153     fpath = data_utils.get_file(

ValueError: `decode_predictions` expects a batch of predictions (i.e. a 2D array
of shape (samples, 1000)). Found array with shape: (1, 2)
```

Visualizing the intermediate layer

In [19]:
```
from keras.models import Model
layer_outputs = [layer.output for layer in model.layers]
activation_model = Model(inputs=custom_resnet_model.input, outputs=layer_outputs)
activations = custom_resnet_model.predict(X_train[10].reshape(1,224,224,3))
print(activations.shape)
def display_activation(activations, col_size, row_size, act_index):
    activation = activations[0, act_index]
    activation_index=1
    fig, ax = plt.subplots(row_size, col_size, figsize=(row_size*2.5,col_size*1))
    for row in range(0,row_size):
        for col in range(0,col_size):
            ax[row][col].imshow(activation[0, :, :, activation_index], cmap='gray')
            activation_index += 1
plt.imshow(test_image)
plt.imshow(X_train[10][:,:,0]);
display_activation(activations, 8, 4, 1)
```

```
(1, 2)

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-19-32e8200fb41b> in <module>
     14 plt.imshow(test_image)
     15 plt.imshow(X_train[10][:,:,0]);
---> 16 display_activation(activations, 8, 4, 1)

<ipython-input-19-32e8200fb41b> in display_activation(activations, col_size, row
_size, act_index)
     10     for row in range(0,row_size):
     11         for col in range(0,col_size):
---> 12             ax[row][col].imshow(activation[0, :, :, activation_index], c
map='gray')
     13             activation_index += 1
     14 plt.imshow(test_image)

IndexError: invalid index to scalar variable.
```





Confusion matrix

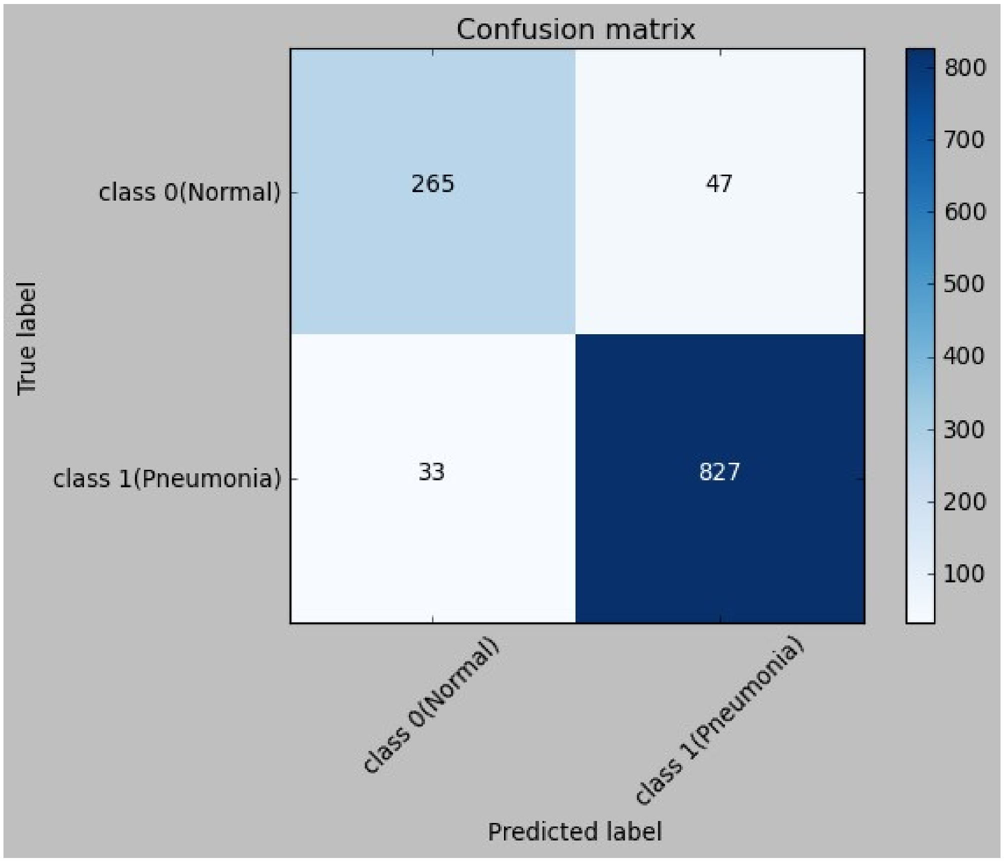
In [20]:
```
Y_pred = custom_resnet_model.predict(X_test)
print(Y_pred)
y_pred = np.argmax(Y_pred, axis=1)
print(y_pred)
#y_pred = model.predict_classes(X_test)
#print(y_pred)
target_names = ['class 0(Normal)', 'class 1(Pneumonia)']
print(classification_report(np.argmax(y_test,axis=1), y_pred,target_names=target_na
mes))
print(confusion_matrix(np.argmax(y_test,axis=1), y_pred))
```

```
[[2.9189095e-03 9.9708110e-01]
 [9.2171538e-01 7.8284606e-02]
 [2.0874552e-01 7.9125452e-01]
 ...
 [1.2248473e-02 9.8775148e-01]
 [2.0469986e-04 9.9979538e-01]
 [9.9027526e-01 9.7247045e-03]]
[1 0 1 ... 1 1 0]
                    precision    recall  f1-score   support

   class 0(Normal)       0.89      0.85      0.87       312
class 1(Pneumonia)       0.95      0.96      0.95       860

          accuracy                           0.93      1172
         macro avg       0.92      0.91      0.91      1172
      weighted avg       0.93      0.93      0.93      1172

[[265  47]
 [ 33 827]]
```

Compute confusion matrix

In [21]:
```
# Compute confusion matrix
cnf_matrix = (confusion_matrix(np.argmax(y_test,axis=1), y_pred))

np.set_printoptions(precision=2)

# Plot non-normalized confusion matrix
plot_confusion_matrix(cnf_matrix, classes=target_names,
                      title='Confusion matrix')
```

```
Confusion matrix, without normalization
[[265  47]
 [ 33 827]]
```



In [ ]:

In [ ]:

In [ ]: