﻿Python 3.6.13 |Anaconda, Inc.| (default, Feb 23 2021, 12:58:59)

Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

from keras.preprocessing.image import ImageDataGenerator, img\_to\_array,load\_img

from keras.models import Sequential

from keras.layers import Dense

import matplotlib.pyplot as plt

from glob import glob

from keras.applications.vgg16 import VGG16

Using TensorFlow backend.

train\_path="veriseti/egitim/"

test\_path="veriseti/dogrulama/"

img=load\_img(train\_path + "DMO/DME-15307-1.jpeg")

plt.imshow(img)

plt.axes("off")

plt.show()

Traceback (most recent call last):

File "<ipython-input-3-5c9e933879f7>", line 2, in <module>

plt.axes("off")

File "/Users/ahmeturunveren/opt/anaconda3/envs/tf/lib/python3.6/site-packages/matplotlib/pyplot.py", line 966, in axes

return gcf().add\_axes(arg, \*\*kwargs)

File "/Users/ahmeturunveren/opt/anaconda3/envs/tf/lib/python3.6/site-packages/matplotlib/figure.py", line 1236, in add\_axes

if not np.isfinite(rect).all():

TypeError: ufunc 'isfinite' not supported for the input types, and the inputs could not be safely coerced to any supported types according to the casting rule ''safe''

Warning

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

x=img\_to\_array(img)

print(x.shape)

(512, 512, 3)

numberOfClass=len(glob(train\_path+"/\*"))

numberOfClass

Out[6]: 4

vgg=VGG16()

2022-07-05 02:01:05.474971: I tensorflow/core/platform/cpu\_feature\_guard.cc:145] This TensorFlow binary is optimized with Intel(R) MKL-DNN to use the following CPU instructions in performance critical operations: SSE4.1 SSE4.2 AVX AVX2 FMA

To enable them in non-MKL-DNN operations, rebuild TensorFlow with the appropriate compiler flags.

2022-07-05 02:01:05.475767: I tensorflow/core/common\_runtime/process\_util.cc:115] Creating new thread pool with default inter op setting: 4. Tune using inter\_op\_parallelism\_threads for best performance.

Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vgg16\_weights\_tf\_dim\_ordering\_tf\_kernels.h5

553467904/553467096 [==============================] - 894s 2us/step

print(vgg.summary())

Model: "vgg16"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fc1 (Dense) (None, 4096) 102764544

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fc2 (Dense) (None, 4096) 16781312

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

predictions (Dense) (None, 1000) 4097000

=================================================================

Total params: 138,357,544

Trainable params: 138,357,544

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

vgg\_layer\_list=vgg.layers

print(vgg\_layer\_list)

[<keras.engine.input\_layer.InputLayer object at 0x7fdb5ab4cf98>, <keras.layers.convolutional.Conv2D object at 0x7fdb5a7117f0>, <keras.layers.convolutional.Conv2D object at 0x7fdb5a7f6da0>, <keras.layers.pooling.MaxPooling2D object at 0x7fdb5aced940>, <keras.layers.convolutional.Conv2D object at 0x7fdb5aced2b0>, <keras.layers.convolutional.Conv2D object at 0x7fdb5491aa58>, <keras.layers.pooling.MaxPooling2D object at 0x7fdb54930b00>, <keras.layers.convolutional.Conv2D object at 0x7fdb549306d8>, <keras.layers.convolutional.Conv2D object at 0x7fdb5493d390>, <keras.layers.convolutional.Conv2D object at 0x7fdb54955ba8>, <keras.layers.pooling.MaxPooling2D object at 0x7fdb54962a90>, <keras.layers.convolutional.Conv2D object at 0x7fdb54962668>, <keras.layers.convolutional.Conv2D object at 0x7fdb549792e8>, <keras.layers.convolutional.Conv2D object at 0x7fdb54990b38>, <keras.layers.pooling.MaxPooling2D object at 0x7fdb5aba1828>, <keras.layers.convolutional.Conv2D object at 0x7fdb5aba15f8>, <keras.layers.convolutional.Conv2D object at 0x7fdb5abb82b0>, <keras.layers.convolutional.Conv2D object at 0x7fdb5abcdac8>, <keras.layers.pooling.MaxPooling2D object at 0x7fdb432067b8>, <keras.layers.core.Flatten object at 0x7fdb43206588>, <keras.layers.core.Dense object at 0x7fdb43206320>, <keras.layers.core.Dense object at 0x7fdb4322f9e8>, <keras.layers.core.Dense object at 0x7fdb432566a0>]

len(vgg\_layer\_list)

Out[10]: 23

vgg\_layer\_list[-1]

Out[11]: <keras.layers.core.Dense at 0x7fdb432566a0>

model=Sequential()

for i in range(len(vgg\_layer\_list)-1):

model.add(vgg\_layer\_list[i])

print(model.summary())

Model: "sequential\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fc1 (Dense) (None, 4096) 102764544

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fc2 (Dense) (None, 4096) 16781312

=================================================================

Total params: 134,260,544

Trainable params: 134,260,544

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

for layers in model.layers:

layers.trainable=False

model.add(Dense(numberOfClass , activation="softmax"))

print(model.summary())

Model: "sequential\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fc1 (Dense) (None, 4096) 102764544

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fc2 (Dense) (None, 4096) 16781312

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 4) 16388

=================================================================

Total params: 134,276,932

Trainable params: 16,388

Non-trainable params: 134,260,544

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

model.compile(loss="categorical\_crossentropy" , optimizer="rmsprop" , metrics=["accuracy"])

train\_data=ImageDataGenerator().flow\_from\_directory(train\_path,target\_size=(512,512))

test\_data=ImageDataGenerator().flow\_from\_directory(test\_path,target\_size=(512,512))

batch\_size=32

hist=model.fit\_generator(train\_data,

steps\_per\_epoch=1600//batch\_size,

epochs=25,

validation\_data=test\_data,

validation\_steps=800//batch\_size)

Found 4000 images belonging to 4 classes.

Found 1000 images belonging to 4 classes.

Epoch 1/25

Traceback (most recent call last):

File "<ipython-input-17-4dec8d12b821>", line 10, in <module>

validation\_steps=800//batch\_size)

File "/Users/ahmeturunveren/opt/anaconda3/envs/tf/lib/python3.6/site-packages/keras/legacy/interfaces.py", line 91, in wrapper

return func(\*args, \*\*kwargs)

File "/Users/ahmeturunveren/opt/anaconda3/envs/tf/lib/python3.6/site-packages/keras/engine/training.py", line 1732, in fit\_generator

initial\_epoch=initial\_epoch)

File "/Users/ahmeturunveren/opt/anaconda3/envs/tf/lib/python3.6/site-packages/keras/engine/training\_generator.py", line 220, in fit\_generator

reset\_metrics=False)

File "/Users/ahmeturunveren/opt/anaconda3/envs/tf/lib/python3.6/site-packages/keras/engine/training.py", line 1508, in train\_on\_batch

class\_weight=class\_weight)

File "/Users/ahmeturunveren/opt/anaconda3/envs/tf/lib/python3.6/site-packages/keras/engine/training.py", line 579, in \_standardize\_user\_data

exception\_prefix='input')

File "/Users/ahmeturunveren/opt/anaconda3/envs/tf/lib/python3.6/site-packages/keras/engine/training\_utils.py", line 145, in standardize\_input\_data

str(data\_shape))

ValueError: Error when checking input: expected input\_1 to have shape (224, 224, 3) but got array with shape (512, 512, 3)

train\_data=ImageDataGenerator().flow\_from\_directory(train\_path,target\_size=(224,224))

test\_data=ImageDataGenerator().flow\_from\_directory(test\_path,target\_size=(224,224))

batch\_size=32

hist=model.fit\_generator(train\_data,

steps\_per\_epoch=1600//batch\_size,

epochs=25,

validation\_data=test\_data,

validation\_steps=800//batch\_size)

Found 4000 images belonging to 4 classes.

Found 1000 images belonging to 4 classes.

Epoch 1/25

50/50 [==============================] - 1134s 23s/step - loss: 1.1965 - accuracy: 0.6112 - val\_loss: 0.5923 - val\_accuracy: 0.7462

Epoch 2/25

50/50 [==============================] - 1040s 21s/step - loss: 0.7528 - accuracy: 0.7344 - val\_loss: 0.4597 - val\_accuracy: 0.7487

Epoch 3/25

50/50 [==============================] - 1035s 21s/step - loss: 0.5679 - accuracy: 0.7962 - val\_loss: 0.7290 - val\_accuracy: 0.7603

Epoch 4/25

50/50 [==============================] - 1047s 21s/step - loss: 0.5282 - accuracy: 0.8119 - val\_loss: 0.8621 - val\_accuracy: 0.7899

Epoch 5/25

50/50 [==============================] - 1055s 21s/step - loss: 0.4468 - accuracy: 0.8381 - val\_loss: 0.8389 - val\_accuracy: 0.7462

Epoch 6/25

50/50 [==============================] - 1133s 23s/step - loss: 0.4513 - accuracy: 0.8375 - val\_loss: 0.5560 - val\_accuracy: 0.7668

Epoch 7/25

50/50 [==============================] - 1104s 22s/step - loss: 0.4258 - accuracy: 0.8500 - val\_loss: 0.4578 - val\_accuracy: 0.8041

Epoch 8/25

50/50 [==============================] - 1127s 23s/step - loss: 0.3741 - accuracy: 0.8662 - val\_loss: 0.6581 - val\_accuracy: 0.8428

Epoch 9/25

50/50 [==============================] - 1035s 21s/step - loss: 0.3510 - accuracy: 0.8756 - val\_loss: 0.5027 - val\_accuracy: 0.7655

Epoch 10/25

50/50 [==============================] - 1043s 21s/step - loss: 0.3512 - accuracy: 0.8737 - val\_loss: 0.3747 - val\_accuracy: 0.7937

Epoch 11/25

50/50 [==============================] - 1079s 22s/step - loss: 0.2879 - accuracy: 0.8975 - val\_loss: 0.4105 - val\_accuracy: 0.8466

Epoch 12/25

50/50 [==============================] - 1209s 24s/step - loss: 0.3137 - accuracy: 0.8819 - val\_loss: 0.6546 - val\_accuracy: 0.8260

Epoch 13/25

50/50 [==============================] - 1201s 24s/step - loss: 0.2557 - accuracy: 0.9119 - val\_loss: 0.9901 - val\_accuracy: 0.8067

Epoch 14/25

50/50 [==============================] - 1097s 22s/step - loss: 0.2873 - accuracy: 0.8944 - val\_loss: 0.7813 - val\_accuracy: 0.8375

Epoch 15/25

50/50 [==============================] - 1032s 21s/step - loss: 0.2880 - accuracy: 0.9000 - val\_loss: 0.4136 - val\_accuracy: 0.8106

Epoch 16/25

50/50 [==============================] - 1033s 21s/step - loss: 0.2270 - accuracy: 0.9181 - val\_loss: 0.3681 - val\_accuracy: 0.8505

Epoch 17/25

50/50 [==============================] - 1033s 21s/step - loss: 0.2655 - accuracy: 0.9131 - val\_loss: 0.3381 - val\_accuracy: 0.8312

Epoch 18/25

50/50 [==============================] - 1035s 21s/step - loss: 0.2329 - accuracy: 0.9131 - val\_loss: 0.3828 - val\_accuracy: 0.8183

Epoch 19/25

50/50 [==============================] - 1043s 21s/step - loss: 0.2035 - accuracy: 0.9312 - val\_loss: 1.2530 - val\_accuracy: 0.8200

Epoch 20/25

50/50 [==============================] - 1121s 22s/step - loss: 0.2157 - accuracy: 0.9306 - val\_loss: 0.9015 - val\_accuracy: 0.8080

Epoch 21/25

50/50 [==============================] - 1122s 22s/step - loss: 0.2281 - accuracy: 0.9200 - val\_loss: 0.6262 - val\_accuracy: 0.8222

Epoch 22/25

50/50 [==============================] - 1122s 22s/step - loss: 0.1659 - accuracy: 0.9344 - val\_loss: 0.4760 - val\_accuracy: 0.8015

Epoch 23/25

50/50 [==============================] - 1134s 23s/step - loss: 0.2090 - accuracy: 0.9269 - val\_loss: 1.0148 - val\_accuracy: 0.8025

Epoch 24/25

50/50 [==============================] - 1071s 21s/step - loss: 0.1759 - accuracy: 0.9388 - val\_loss: 0.7388 - val\_accuracy: 0.8325

Epoch 25/25

50/50 [==============================] - 1033s 21s/step - loss: 0.1961 - accuracy: 0.9325 - val\_loss: 0.1764 - val\_accuracy: 0.8196