import keras

from keras.layers.core import Flatten

from keras.layers.core import Dense

from keras.applications.vgg19 import VGG19

from keras.layers import Input

from keras.models import Model

from keras.regularizers import \*

def get\_model():

aliases = {}

Input\_1 = Input(shape=(3, 224, 224), name='Input\_1')

VGG19\_1\_model = VGG19(include\_top= False, input\_tensor = Input\_1)

VGG19\_1 = VGG19\_1\_model(Input\_1)

aliases['VGG19\_1'] = VGG19\_1\_model.name

num\_layers = len(VGG19\_1\_model.layers)

for i, layer in enumerate(VGG19\_1\_model.layers):

if ((i \* 100) / (num\_layers - 1)) <= (100 - 0):

layer.trainable = False

Flatten\_3 = Flatten(name='Flatten\_3')(VGG19\_1)

Dense\_7 = Dense(name='Dense\_7',output\_dim= 3,activation= 'softmax' )(Flatten\_3)

model = Model([Input\_1],[Dense\_7])

return aliases, model

from keras.optimizers import \*

def get\_optimizer():

return Adam()

def is\_custom\_loss\_function():

return False

def get\_loss\_function():

return 'categorical\_crossentropy'

def get\_batch\_size():

return 32

def get\_num\_epoch():

return 10

def get\_data\_config():

return '{"kfold": 1, "samples": {"validation": 2548, "training": 7646, "split": 2, "test": 2548}, "datasetLoadOption": "full", "shuffle": true, "numPorts": 1, "mapping": {"Label": {"port": "OutputPort0", "type": "Categorical", "options": {}, "shape": ""}, "Filename": {"port": "InputPort0", "type": "Image", "options": {"Normalization": false, "Scaling": 1, "rotation\_range": "30", "Resize": true, "width\_shift\_range": "0.2", "horizontal\_flip": false, "Height": "224", "height\_shift\_range": "0.2", "Width": "224", "shear\_range": "0.2", "vertical\_flip": true, "Augmentation": true, "pretrained": "None"}, "shape": ""}}, "dataset": {"type": "private", "samples": 12744, "name": "train"}}'