needed the newest version of:

keras;

tensorflow:

pillow;

model文件夹：网络模型

model\_weight文件夹：网络模型训练后的权重

samples：提供了几个皮肤癌的图片

prediction.py: 判断程序 黄色区域标出input image和output value

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

import keras

import numpy as np

from model.skin\_resnet import skin\_resnet

from keras.preprocessing import image

from keras.preprocessing.image import transform\_matrix\_offset\_center, apply\_transform

def img\_rotation(x, theta,row\_axis=0, col\_axis=1, channel\_axis=2,

fill\_mode='nearest', cval=0.):

"""modifed from keras random\_rotation

# Arguments

x: Input tensor. Must be 3D.

theta: Rotation range, in degrees.

row\_axis: Index of axis for rows in the input tensor.

col\_axis: Index of axis for columns in the input tensor.

channel\_axis: Index of axis for channels in the input tensor.

fill\_mode: Points outside the boundaries of the input

are filled according to the given mode

(one of `{'constant', 'nearest', 'reflect', 'wrap'}`).

cval: Value used for points outside the boundaries

of the input if `mode='constant'`.

# Returns

Rotated Numpy image tensor.

"""

# theta = np.pi / 180 \* np.random.uniform(-rg, rg)

rotation\_matrix = np.array([[np.cos(theta), -np.sin(theta), 0],

[np.sin(theta), np.cos(theta), 0],

[0, 0, 1]])

h, w = x.shape[row\_axis], x.shape[col\_axis]

transform\_matrix = transform\_matrix\_offset\_center(rotation\_matrix, h, w)

x = apply\_transform(x, transform\_matrix, channel\_axis, fill\_mode, cval)

return x

def img\_norm(img\_array):

"""tensorflow tensor form

"""

img\_array = img\_array.reshape((1,) + img\_array.shape)

# normalization:

for i in range(img\_array.shape[0]):

for k in range(3):

img\_array[i,::,::,k] -= np.mean(img\_array[i,::,::,k])

img\_array[i,::,::,k] /= np.std(img\_array[i,::,::,k]) + 1e-7

return img\_array

def pred\_output(pred = 0):

""" Print the predictions

"""

print ("\*"\*50)

print "皮肤癌相似度：" + str("%.2f") %pred +"%"

print ("\*"\*50)

if \_\_name\_\_ == '\_\_main\_\_':

# parameters:

sz = 224 # resize image into (224,224,3)

theta = [0,-90,90,120,270]

preds = np.ndarray(shape=(5), dtype=float)

model = skin\_resnet() # 导入网络

model.load\_weights("model\_weight/resnet50\_turnable\_152\_d2\_3.h5") # 导入权重

print ("请输入图像所在位置:")

image\_path = raw\_input() # 输入图片

img = image.load\_img(image\_path, target\_size = (sz,sz))

img\_array = image.img\_to\_array(img)

i = 0

for thet in theta:

img\_array2 = img\_rotation(img\_array, thet)

img\_array2 = img\_norm(img\_array2)

results = model.predict(img\_array2) \* 100

preds[i] = results[0,0]

i = i + 1

pred\_f = np.median(preds) # pred\_f是预测结果

pred\_output(pred\_f)