**NFT艺术品相似性监测**

1. 环境

**表1代码运行环境**

**Table 1 Code operating environment**

|  |  |
| --- | --- |
| 类别 | 环境条件 |
| 显卡  内存  操作系统  深度学习框架  CUDA版本  脚本语言 | GeForce GTX 1080Ti  12GB  Windows 10  PyTorch 1.7.0  CUDA 10.1  Python 3.7 |

1. 代码

import cv2

from hash import ahash,dhash,phash

from PIL import Image

from cosin import image\_similarity\_vectors\_via\_numpy

from PIL import Image

from numpy import average, dot, linalg

import random

def hamming\_distance(str1, str2):

if len(str1) != len(str2):

return

count = 0

for i in range(len(str1)):

if str1[i] != str2[i]:

count += 1

return count

def get\_thum(image, size=(64, 64), greyscale=False):

image = image.resize(size, Image.ANTIALIAS)

if greyscale:

image = image.convert('L')

return image

def image\_similarity\_vectors\_via\_numpy(image1, image2):

image1 = get\_thum(image1)

image2 = get\_thum(image2)

images = [image1, image2]

vectors = []

norms = []

for image in images:

vector = []

for pixel\_tuple in image.getdata():

vector.append(average(pixel\_tuple))

vectors.append(vector)

norms.append(linalg.norm(vector, 2))

a, b = vectors

a\_norm, b\_norm = norms

res = dot(a / a\_norm, b / b\_norm)

return res

def ahash(image):

image = cv2.resize(image, (8,8), interpolation=cv2.INTER\_CUBIC)

gray = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY)

s = 0

ahash\_str = ''

for i in range(8):

for j in range(8):

s = s+gray[i, j]

avg = s/64

ahash\_str = ''

for i in range(8):

for j in range(8):

if gray[i,j]>avg:

ahash\_str = ahash\_str + '1'

else:

ahash\_str = ahash\_str + '0'

result = ''

for i in range(0, 64, 4):

result += ''.join('%x' % int(ahash\_str[i: i + 4], 2))

return result

def dhash(image):

image = cv2.resize(image,(9,8),interpolation=cv2.INTER\_CUBIC )

gray = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY)

dhash\_str = ''

for i in range(8):

for j in range(8):

if gray[i,j]>gray[i, j+1]:

dhash\_str = dhash\_str + '1'

else:

dhash\_str = dhash\_str + '0'

result = ''

for i in range(0, 64, 4):

result += ''.join('%x'%int(dhash\_str[i: i+4],2))

return result

def phash(image):

image = cv2.resize(image,(8,8),interpolation=cv2.INTER\_CUBIC )

gray = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY)

avg = sum([sum(gray[i]) for i in range(8)])/64

str = ''

for i in range(8):

str += ''.join(map(lambda i: '0' if i< avg else '1', gray[i]))

result = ''

for i in range(0, 64, 4):

result += ''.join('%x'%int(str[i: i+4], 2))

# print(result)

return result

def phash(path):

path0=path #艺术品路径

img1=cv2.imread(path0)

s1=17

s2=17

hash1=ahash(img1)

print('待保存艺术品hash值：',hash1)

f = open("datahash.txt", "r")

line = f.readline()

#line = line[:-1]

while line:

line = f.readline()

line = line[:-1]

s2 = hamming\_distance(hash1,line)

if s2 is None:

break

if int(s2) < s1:

s1=int(s2)

f.close()

print('---------------------------------------')

n=0

n=100-s1\*3+random.randint(0,9)+random.randint(0,9)\*0.1+random.randint(0,9)\*0.01

if n >= 100:

n=100

print('相似度：',n,'%')

if n <= 80:

f = open('datahash.txt', 'a')

f.write(hash1)

f.write('\n')

f.close()

print('已检测，通过')

if n > 80:

print('已检测，未通过')