import os

import os.path

import re

from itertools import islice

import numpy  as np

class rgb\_sim:

   colors = [[0,0,0],

            [255,255, 255],

            [255, 0, 0],

            [246, 232, 9],

            [72, 176, 64],

            [27, 115, 186],

            [53, 118, 84],

            [244, 181, 208],

            [255, 145, 0],

            [177, 125, 85],

            [92, 59, 144],

            [11, 222, 222],

            [228, 0, 130],

            [225, 218, 32],

            [118, 238, 0],

            [17, 168, 226],

            [255, 110, 0],

            [201, 202, 202],

            [255, 249, 177],

            [179, 226, 242],

            [249, 225, 214],

            [186, 149, 195]

    ]

   #创建一个字典，用于存储22个颜色样本

   pic2=open('D:/twinkle/桌面/数学建模华中杯/附件/附件2：图像1颜色列表.txt', mode='r')

   next(pic2)

   #跳过标题

   dic=dict()

   #创建空字典，用于存储目标

   for line in pic2:

      '''按行读取'''

      k,v =line.split(',',1)  # 以','为分隔符分割字符串

      dic[k] = v[0:-1]  #去掉字符串末尾的换行符

   dic=list(dic.values())

   # print(dic)

   rgb\_sum=[]

   #创建空列表

   for i in range(len(dic)):

      rgb=re.findall(r'\d+',dic[i])

      #构建循环，正则表达提取rgb对应数值并添加到列表

      rgb = [int(x) for x in rgb]

      #将列表中的字符串转为数字

      rgb\_sum.append(rgb)

      pic2.close()

      # print (rgb)

   # print (rgb\_sum[1][1])

   # print(pic2.read())

   '''计算三维向量在空间中的距离,衡量样本与目标之间的差异'''

   solver1 =[]

   #创建一个空列表，用于储存结果

   for j in range(len(rgb\_sum)):

      vector2 = np.array(list(rgb\_sum[j]))

      result =[]

      for i in range(len(colors)):

         vector1 = np.array(colors[i])

         op = np.linalg.norm(vector1-vector2)

         #欧式距离范式

         result.append(op)

      solver1.append(result.index(min(result)))

      print (f'目标{j+1}对应的瓷砖颜色应为',result.index(min(result))+1)

      #标号从1开始，索引+1

      #对每一个目标rgb，有一个列表存储其与每个样本color的偏差(空间距离)，求与目标最小偏差对应的样本(索引)

   s = ['序号,瓷砖颜色编号\n']

   solver1 = [ x + 1 for x in solver1]

   for i in range (0,216):

      string = f'{i+1,solver1[i]}'

      string =str(string) + '\n'

      s.append(string)

   # print (type(string))

   # print(string)

   # print(type(s))

   # print(s)

   f1 = open ('D:/twinkle/桌面/数学建模华中杯/附件/ex1.txt',mode ='w+')

   f1.writelines(s)

   f1=f1.readlines()

   for line in f1:

      fileout1.write(line.replace('','(' or ')'))

   print("文件保存成功")

   print ('='\*60)

   #打印分隔符

   '''仅更改目标绝对路径、输出绝对路径与目标个数'''

      #创建一个字典，用于存储22个颜色样本

   pic2=open('D:/twinkle/桌面/数学建模华中杯/附件/附件3：图像2颜色列表.txt', mode='r')

   next(pic2)

   #跳过标题

   dic=dict()

   #创建空字典，用于存储目标

   for line in pic2:

      '''按行读取'''

      k,v =line.split(',',1)  # 以','为分隔符分割字符串

      dic[k] = v[0:-1]  #去掉字符串末尾的换行符

   dic=list(dic.values())

   # print(dic)

   rgb\_sum=[]

   #创建空列表

   for i in range(len(dic)):

      rgb=re.findall(r'\d+',dic[i])

      #构建循环，正则表达提取rgb对应数值并添加到列表

      rgb = [int(x) for x in rgb]

      #将列表中的字符串转为数字

      rgb\_sum.append(rgb)

      pic2.close()

      # print (rgb)

   # print (rgb\_sum[1][1])

   # print(pic2.read())

   '''计算三维向量在空间中的距离,衡量样本与目标之间的差异'''

   solver1 =[]

   #创建一个空列表，用于储存结果

   for j in range(len(rgb\_sum)):

      vector2 = np.array(list(rgb\_sum[j]))

      result =[]

      for i in range(len(colors)):

         vector1 = np.array(colors[i])

         op = np.linalg.norm(vector1-vector2)

         #欧式距离范式

         result.append(op)

      solver1.append(result.index(min(result)))

      print (f'目标{j+1}对应的瓷砖颜色应为',result.index(min(result))+1)

      #标号从1开始，索引+1

      #对每一个目标rgb，有一个列表存储其与每个样本color的偏差(空间距离)，求与目标最小偏差对应的样本(索引)

   s = ['序号,瓷砖颜色编号\n']

   solver1 = [ x + 1 for x in solver1]

   for i in range (0,200):

      #目标个数

      string = f'{i+1,solver1[i]}'

      string =str(string) + '\n'

      s.append(string)

   # print (type(string))

   # print(string)

   # print(type(s))

   # print(s)

   f1 = open ('D:/twinkle/桌面/数学建模华中杯/附件/ex2.txt',mode ='w+')

   f1.writelines(s)

   f1=f1.readlines()

   for line in f1:

      fileout1.write(line.replace('','(' or ')'))

   print("文件保存成功")