تمرین صفرم

نیکی مجیدی فرد 98522382

سوال 1)

بخش الف)

خواندن تصاویر و نمایش با کمک :matplotlib

خواندن تصویر : plt.imread(image_path)

نشان تصویر : plt.imshow(image)

نتیجه برای ورودی و خروجی با matplotlib:

OpenCV :

خواندن تصویر : cv2.imread(image_path , color_flag)

نشان دادن تصویر: cv2.imshow(text , image)

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

تفاوت این دو کتاب خانه در نمایش دادن تصاویر ، کانالی است که تصویر را با آن میخوانند.

تصاویر رنگی در matplotlibبا فرمت rgb و تصاویر در opencvبا فرمت bgr خوانده می شوند.

```
#show image with opencv read image
plt.imshow(img2)
plt.axis('off')
```

```
(-0.5, 1019.5, 679.5, -0.5)
```

برای رفع مشکل، هنگامی که میخواهیم با opencv عکس را بخوانیم و با matpilotlib، نمایش دهیم ، میتوانیم از

```
(cv2.cvtColor(image,  cv2.COLOR_BGR2RGB)
```

استفاده کنیم .

نتیجه:

```
plt.imshow(cv2.cvtColor(img2 , cv2.COLOR_BGR2RGB))
plt.axis('off')
plt.show()
```

برای از بین رفتن نمودار اطراف عکس از plt.axis('off') استفاده میکنیم.

نمایش عکس در opencv علاوه بر این که در کانال متفاوت رنگی است ، در window جدا گانه پایتون , عکس نمایش داده می شود.

قسمت ب)

```
    width , height , channels= img2.shape
    print('width :', width , ',height:',  height ,',channels:' , channels )
```

```
width : 680 ,height: 1020 ,channels: 3
```

این دستور نمایانگر عرض ، طول و تعداد کانال های تصویر است. در صورتی که عکس رنگی و سه کاناله باشد ،(عکس های رنگی همواره 3 کاناله هستند) یک تاپل سه تای و در صورتی که عکس یک کاناله خوانده شود ، (حالت سیاه سفید و بین 0-255) تاپل خروجی دو تایی طول و عرض است

.

قسمت ج)

ابتدا با کمک os.mkdir یک دایرکتوری در پوشه ی Q1 ساخته ،

با کمک دستوز os.scandir روی همه ی فایل ها یک فور زده و عکس ها را خوانده ،

با کمک دستور cv2.cvtColor(img , cv2.COLOR_BGR2GRAY) ، عکس ها را سیاه سفید کرده و با کمک دستور cv2.imwrite('name' , image)در دیرکتوری بالا ذخیره می کنیم. و در فایل نیز اندازه تصاویر را ذخیره و با اسم قبلی در پوشه جدید ذخیره میشود.

```
# Directory
directory = "new_Q1"

parent_dir = "Q1"

path = os.path.join(parent_dir, directory)

try:
    os.makedirs(path, exist_ok = True)
    print("Directory '%s' created successfully" %directory)
except OSError as error:
    print("Directory '%s' can not be created")
```

✓ 0.0s

```
Directory 'new_Q1' created successfully
```

```
for filename in os.scandir('Q1'):
    if filename.is_file():
        path = filename.path
        img = cv2.imread(path)
        img_gray = cv2.cvtColor(img , cv2.COLOR_BGR2GRAY)
        img_name = path.split('\\')[1].split('.')[0]
        cv2.imwrite(rf'Q1\\new_Q1\\{img_name}_Gray.jpeg' , img_gray)
        file = open(rf'Q1\\new_Q1\\{img_name}_file',"a")
        file.writelines(str(img.shape))
```

سوال 2)

الف) برای resizeکردن تنها از دستور کتابخوانه opencv استفاده میکنیم.

```
resized_image = cv2.resize(image, size,interpolation= cv2.INTER_NEAREST)
```

ب)

در این قسمت لازم است که عکس را با اندازه خواسته شده به تیکه های کوچک تر تقسیم کنیم.

کد کراپ کردن

```
images = []
  M = CropSize[0]
  N = CropSize[1]
  x1 = 0
  y1 = 0
  imgheight= MainSize[0]
  imgwidth= MainSize[1]
  image_copy = image.copy()
  for y in range(0, imgheight, M):
    for x in range(0, imgwidth, N):
        if (imgheight - y) < M or (imgwidth - x) < N:
            break
        y1 = y + M
        x1 = x + N
        if x1 >= imgwidth and y1 >= imgheight:
            x1 = imgwidth - 1
            y1 = imgheight - 1

            tiles = image_copy[y:y+M, x:x+N]

        elif y1 >= imgheight:
            y1 = imgheight - 1


            tiles = image_copy[y:y+M, x:x+N]

        elif x1 >= imgwidth:
            x1 = imgwidth - 1

            tiles = image_copy[y:y+M, x:x+N]

        else:

            tiles = image_copy[y:y+M, x:x+N]
        if np.sum(tiles) != 0:
           images.append(tiles)
```

به همین صورت ، عکس ها را در یک فور و با چند دستور ایف و الس به قسمت های مختلف کراپ کرده و در آرایه می ریزیم.

سپس عکس هایی که مجموع همه ی پیکسل هایشان 0 است را د ر ارایه نمیریزیم.

(کد بر گرفته از خود سایت OpenCV)

https://learnopencv.com/cropping-an-image-using-opencv/

پ) برای انتخاب کردن 5عدد رندوم از

```
np.random.randint(5, size=5)
```

استفاده می کنیم .

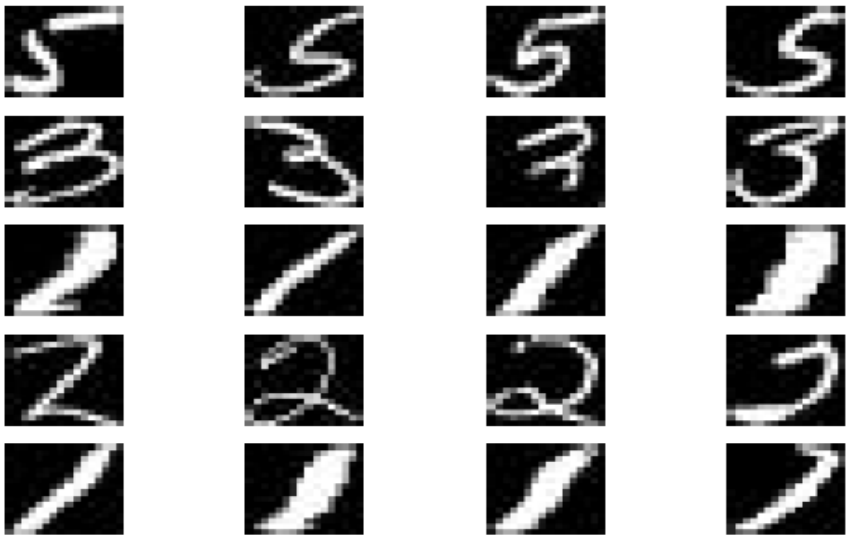
عکس مورد نظر را انتخاب کرده و از لیست درست شده در قسمت قبلی 4 عکس را رندوم انتخاب کرده و در مجموع این 20 عکس را به کمک subplot کنار هم نمایش می دهیم.

```
random_numbers = np.random.randint(5, size=5)
image_arrays = []

for num in random_numbers:
    image = images_path[num]
    cropped_image = All_cropped_images[num]
    indexes = np.random.randint(len(cropped_image), size=4)

    for i in indexes:
        selected_image =  cropped_image[i]
        resized_image = resize(selected_image, (50,50))
        image_arrays.append(resized_image)


fig = plt.figure()
for i in range(20):
    fig.add_subplot(5, 4, i+1)
    plt.imshow(image_arrays[i] , cmap = 'gray')
    plt.axis('off')
```



نتیجه :

سوال 3)

قسمت الف)

برای تعریف کردن یک ارایه با تعداد عناصر رندوم از

```
random.randint(low, high=None, size=None, dtype=int)
```

استفاده می کنیم.

```
matrix = np.random.randint(low=n , high=n+100, size=(n, n))
```

ب) برای شمردن تعداد رقم ها ، ابتدا تمام عدد ها را در ماتریس تبدیل به رقم می کنیم و در آرایه می ریزیم.

سپس با کمک دستور `counter` آن را تبدیل به یک دیکشنری می کند که تعداد عناصر را می شمارد و در انتها دیکشنریی را سورت می کنیم.

```
def return_digits(number):
    list_of_digits = [int(i) for i in str(number)]
    return list_of_digits



def count_digits(matrix):
    """
    In this function you should implement a routine to count the digits in the given matrix.

    Arguments:
        inputs:
            matrix: input matrix
        outputs:
            digits: a dictionary containing digits as its keys and the counted numbers as its value

    """
    list_of_all_digits = []
    for x in np.nditer(matrix):
        list_of_all_digits = return_digits(x) + list_of_all_digits

    counter = collections.Counter(list_of_all_digits)

    ################################################
    ############ YOUR CODES GO HERE ################
    digits = counter
    dictionary1 = sorted(digits)
    sorted_dict = {key:digits[key] for key in dictionary1}
    ################################################

    return sorted_dict
```

نتیجه برای ماتریس :

```
[[ 56 56 8] [ 23 85 74] [ 87 100 81]]
```

```
{0: 2, 1: 2, 2: 2, 4: 2, 5: 2, 6: 4, 7: 1, 8: 2, 9: 1}
```

قسمت ب) همین طور که مشخص است ، در این ماتریس از قطر برعکس شروغ میکند و یک بار به سمت بالا و یک بار درجهت بر عکس می پیماید.

از دستور `np.diognal()`استفاده می کنیم تا قطر ها را در هر زیر ماتریس نمابش دهد. منتها باید قطر فرعی شود پس با کمک فلیپ به راست و چپ آن را تنظیم می کنیم. به طوری که در صورتی که `I`زوج باشد `flipr` و در غیر این صورت `flipud` ، و این عمل برای قسمت بالا مثلثی و پایین مثلثی عینا تکرار می شود.

```python
def traverse_matrix(matrix):
    """
    Traverse the input matrix in the given manner then print result

    Arguments:
        inputs:
            matrix: input matrix
        outputs
    """

    new_matrix = matrix.copy()
    size = matrix.shape[0]
    for i in range(size):
        i= i+1
        submatrix = new_matrix[:i , :i]
        if(i%2 == 0):
            for item in np.fliplr(submatrix).diagonal(): print(item , end =" ")

        else:
            for item in np.flipud(submatrix).diagonal() : print(item , end =" ")

    for i in range(size):
        submatrix = new_matrix[i+1:size , i+1:size]
        if(i%2 == 0):
            for item in np.fliplr(submatrix).diagonal(): print(item , end =" ")

        else:
            for item in np.flipud(submatrix).diagonal() : print(item , end =" ")
```

نتیجه)

```
62 10 48 56 60 97 14 26 58 None
```

سوال 4)

الف) برای محاسبه ی ترنسپورت ماتریس از `np.transpose`

برای ضرب از `np.dot` و برای محاسبه `inverse` ، از تابع `np.linalg.inv()` استفاده می کنیم و عملیات به صورت زیر است.

```
"""
Please define the mentioned matrices and the operations

consider the bellow links:
https://numpy.org/doc/stable/reference/generated/numpy.transpose.html

https://numpy.org/doc/stable/reference/generated/numpy.dot.html

"""
matris_A = np.array([[1,2,3] , [4,5,6] , [7,8,9]])
matris_B = np.array([[1] , [-1] , [1]])

matris_A_transpose = np.transpose(matris_A)

x1 = np.linalg.inv(np.dot( matris_A_transpose , matris_A ))
x2 = np.dot(matris_A_transpose , matris_B)
result = np.dot(x1 , x2) + matris_A
print(result)
```

```
[[0.78125  1.78125  2.78125 ]
 [4.       5.       6.      ]
 [7.171875 8.171875 9.171875]]
```

قسمت ب) همان طور که در صورت سوال گفته شده ، روی ارایه ی بزرگ تر یک فور تو در تو می زنیم و یک فور هم روی ارایه کوچک تر می زنیم. تمام عناصر ارایه کوچک را در همان اسکیل در تمام عناصر ارایه بزرگ تر ضرب و همگی را جمع می کنیم. در صورتی که جمع ایندکس ارایه بزرگ و کوچک بیشتر از اندازه ارایه بزرگ تر شود ، پنجره بیرون بزند ، مقدار آن را در نظر نمی گیریم.

نتیجه کد :

```
def window_sliding(A, B):

  w1 , h1 = B.shape;
  w2, h2 = A.shape;


  result = np.zeros((w2, h2))
  for i in range(w1):
    for j in range(h1):
      x = True
      sum = 0
      for k in range(w2):
        for t in range(h2):
          if(i+k <w1 and j + t < h1):
            sum += B[i+k, j+t]* A[k,t]
          else:
            x = False
      if(x):
        result[i , j]= sum


  return result
```

```
A = np.array([[1,1,1] , [1, -9 , 1] , [1,1,1]])
B = np.array([[1,1,1,1,1], [1,2,2,2,1], [1,2,2,2,1] , [1,2,2,2,1] , [1,1,1,1,1]])
print(window_sliding(A, B))
```

```
[[-7. -5. -7.]
 [-5. -2. -5.]
 [-7. -5. -7.]]
```

منابع)

/https://realpython.com/python-counter

https://numpy.org/doc/stable/reference/generated/numpy.diagonal.html

https://numpy.org/doc/stable/reference/generated/numpy.linalg.inv.html

https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_matplotlib_rgb_brg_image_load_display_save.php#:~:text=There%20is%20a%20difference%20in,convert%20it%20into%20RGB%20mode.&text=If%20we%20did%20not%20switch,we%20get%20rgb%20inverted%20picture.