

بسم الله الرحمن الرحيم



محمد عرفان زارع زردینی

۹۸۱۴۱۱۴۳۲

تمرین سری صفر درس بینایی ماشین

References:

<https://www.geeksforgeeks.org/create-a-directory-in-python/>

<https://stackoverflow.com/questions/18262293/how-to-open-every-file-in-a-folder>

[https://stackoverflow.com/questions/38598118/difference-between-plt-imshow-and-cv2-](https://stackoverflow.com/questions/38598118/difference-between-plt-imshow-and-cv2-imshow)

[imshow](#)

<https://www.entechin.com/how-to-convert-an-image-to-black-and-white-in-python/#:~:text=If%20you%20want%20to%20transform,the%20image%20to%20grayscale%20mode.>

<https://www.geeksforgeeks.org/python-opencv-cv2-imwrite-method/>

(الف)

با استفاده از دو کتابخانه **matplotlib** و **opencv**، تصویر را خوانده و نمایش می‌دهیم. برای اینکه فرجی **matplotlib** مانند **opencv** شود باید کانال‌های آن را تغییر داد و مانند **opencv** کرد. **opencv** به شکل **bgr** هست ولی **matplotlib** برعکس. علت تفاوت فرجی تصاویر این است که همانطور که گفته شد ترتیب کانال‌ها و نحوه ذخیره سازی در دو کتابخانه متفاوت. این معکوس بودن کانال‌ها سبب همیشه وقتی با **opencv** ذخیره کنیم و با **matplotlib** نمایش دهیم، تصویر به صورت نگاتیو نمایش یابد همچنین برعکس همین کار رو هم اگر بتوانیم، همینگونه هست. برای یکسان سازی از توابع مربوط به **opencv** برای تغییر کانال عکس خوانده شده استفاده می‌شود.

```
conv=cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
```

(ب)

دستور `image.shape` نشان میدهد که خروجی یک آرایه سه بعدی که دو عدد ابتدایی شامل ابعاد تصویر (طول و عرض عکس که تعداد پیکسل هر سطر ستون رو هم نمایندگی میکند) است که اینجا یعنی تصویر ما `1020*680` پیکسل می باشد و بعد سوم که `3` هست نشان دهنده تعداد کانال های رنگی عکس هست که یعنی هر پیکسل مان دارای سه مقدار رنگی قرمز و سبز و آبی هست.

```
(680, 1020, 3)
```

(ج)

فایل های فولدر `Q1` را خوانده عکس ها و نام های شان را ذخیره می نماییم. حال می آییم و فولدری داخل فولدر `Q1` ساخته (با دستوری میسازیم که در صورت عدم وجود فولدر آن را بسازد. حال عکس هایمان که هنگام خواندن و ذخیره سازی به فرمت سیاه سفید ذخیره شده است را ، سایشان (`shape`) را مشخص کرده و در فایل متنی آن را با همان نام عکسمان ذخیره کرده و عکس سیاه سفید متناظر آن را هم در آنجا ذخیره می نماییم.

References:

<https://cloudinary.com/guides/bulk-image-resize/python-image-resize-with-pillow-and-opencv#opencv>

<https://numpy.org/doc/stable/reference/generated/numpy.resize.html>

<https://realpython.com/python-sum-function/>

<https://learnopencv.com/cropping-an-image-using-opencv/#:~:text=There%20is%20no%20specific%20function,And%20it's%20done!>

<https://www.geeksforgeeks.org/python-randint-function/>

در حل سوال ابتدا کتابخانه های مد نظر را افزوده ، برای تابع **resize** از متدی با همین نام که در کتابخانه **cv2** هست استفاده می کنیم. در نوشتن تابع **crop** نیز ابتدا نقاط شروع را مشخص کرده و مقدار **0** می دهیم. سپس طول عکس را با مقدار مشخص شده و **cropsizes** پیموده و عکس را کراپ می نماییم. سپس چک میشه که کل عکس اگر سیاه نبود ذخیره شود در ارایه. همچنین هرگاه طول ارایه به **1100** رسید، ارایه را برگرداند. در قسمت بعد عکس هارا خوانده در مسیری که مشخص شده و در ارایه ای میریزیم. سپس توابع توضیح داده شده را فرا میخوانیم تا عکس های کراپ شده در نهایت ذخیره در ارایه ای شوند و در نهایت روی ارایه نهایی ، حلقه زده و با استفاده از تابع تولید عدد رندوم، **4** عکس کراپ شده از هر عکس اصلی را برداشته و سائز آن را با تابعمان تغییر داده و نمایش می دهیم .

(3)

References:

<https://www.geeksforgeeks.org/python-dictionary/>

https://www.w3schools.com/python/python_dictionary.asp

(الف)

ابتدا کتابخانه **numpy** را اضافه نموده ، سپس تابع **create_matrix** که برای ساخت ماتریس با اعداد رندوم هست را کامل مینماییم. حال در بخش بعد سایز ماتریس $n \times n$ را مشخص میکنیم و از کاربر گرفته و بدین سان ، ارایه مد نظر را می سازیم. سپس در تابع **count_digits** می اییم دیکشنری ساخته و اعداد رو بررسی کرده و تعداد تکرار ارقام را در دیکشنری با کلید رشته رقم، ذخیره می نماییم. و تابع را روی ماتریس صدا میزنیم و چاپ می نماییم خروجی رو.

(ب)

در این بخش ابتدا ایندکس هایی برای پیمایش روی ماتریس تعریف و مقدار اولیه صفر میدهیم. سپس اندازه ماتریس را ذخیره مینماییم تا با بررسی آن به حل پردازیم. الگوریتم سوال از حرکت قطری استفاده مینماید. برای همین ابتدا متغیری برای آن تعریف و مقدار اولیه 2 میدهیم. همچنین **flag** ی برای اینکه مشخص کنیم در کدام قسمت قطر اصلی هستیم در نظر میگیریم. سپس وارد حلقه میشیمو تا زمانی که مقدار طول قطر تعریفی از صفر بیشتره کار های زیر را میکنیم:

مسیر حرکت‌مان به شکل قطری می‌باشد و همچنین هر بار مقدار قطر و جهتش تغییر میکند(زیاد میشه تا به قطر اصلی برسیم و از آنجا کم میشه و همچنین در مسیر حرکت به حالتی که در آن هستیم تصمیم به انجام حرکتی میگیریم.
سپس در اخر هم تابع را با پیمایش گفته شده چاپ میکنیم.

(4)

References:

https://www.tutorialspoint.com/numpy/numpy_inv.htm#:~:text=We%20use%20numpy.,it%20results%20in%20identity%20matrix.

[https://www.javatpoint.com/numpy-dot#:~:text=numpy._,dot\(\)%20in%20Python,vectors%20\(without%20complex%20conjugation\).](https://www.javatpoint.com/numpy-dot#:~:text=numpy._,dot()%20in%20Python,vectors%20(without%20complex%20conjugation).)

<https://www.geeksforgeeks.org/python-numpy-numpy-transpose/>

https://www.tutorialspoint.com/numpy/numpy_inv.htm#:~:text=We%20use%20numpy.,it%20results%20in%20identity%20matrix.

<https://www.geeksforgeeks.org/numpy-add-in-python/>

(الف)

در این بخش با استفاده از متدهای کتابخانه **numpy** و استفاده از متدهایی چون **transpose** و وارون کردن و ضرب و جمع ماتریسی، حاصل را می‌یابیم و چاپ مینماییم.

(ب)

ابتدا برای جلوگیری از شلوغی زیاد و تمیزی بیشتر و خوانایی، تابعی برای محاسبه ضرب ماتریسی دو ورودی تعریف مینماییم تا بدین سان در هر مرحله حلقه، ماتریس **A** و یک ماتریس 3×3 جداسازی شده از ماتریس **B** و فرستادن به تابع محاسبه گر، جواب را یافته و در خانه نظیر فونه اول (بالا) سمت چپ آرایه جدا شده در ماتریس **B**، نظیر ایندکسش در آرایه جدید ذخیره شود. بدین سان پاسخ که آرایه 3×3 هست یافته شده و جواب برگردانده می‌شود. در بخش آخر هم ماتریس‌های **A, B** تعریف شده و توابع روی آن‌ها زده زده شونده و نتیجه چاپ شده است.

(5)

References:

<https://stackoverflow.com/questions/63923800/drawing-bounding-rectangles-around-multiple-objects-in-binary-image-in-python>

<https://www.geeksforgeeks.org/python-statistics-mean-function/>

در ابتدا ، کتابخانه های مورد نیاز اضافه شده اند. سپس در چند بخش بعد تنها دستور پیش نویس شده اجرا و ران شدند. حال برای تصویر مد نظر ،

`shape,dtype,mean,amax,amin` را مناسبه مینماییم.

همچنین `amin,amax` برای کانال اول ارایه را مناسبه مینماییم.

سپس سلول بعدی را ران میکنیم. حال برای تابع `detections` جزئیات موجود را چاپ می نماییم. هر بخش دارای دو عضو بوده که عضو دوم درستی تشخیص را نمایش میدهد. عضو اول شامل 4 بخش هست که که خانه اول آن مقدار `x` در بالا چپ محدوده هست و خانه دوم `y` نقطه بالا چپ هست و دو بخش دیگر نیز عرض و ارتفاع محدوده اند.

در بخش بعد با تابع `visualize`، با استفاده از عکس اصلی و ارایه `detections`، محدوده ها را تعریف می نماییم و روی عکس نمایش می دهیم. بدین سان که حلقه زده و با محدوده های بیان شده ، چهارگوشه محدوده را مشخص کرده و با تابع `cv2.rectangle` روی شکل مشخص نموده ایم. حال تصویر نهایی را با نام مشخص شده ذخیره نموده ایم. و در نهایت ذخیره شدن آن را چک می نماییم.