به نام خدا

مبانی بینایی کامپیوتر

دکتر محمدی

تمرین صفر

علی عطاریان – ۹۹۵۲۱۴۵۱

سوال ۱) الف)

در هر دو کتابخانه cv2 و matplotlib از دستور imread برای خواندن تصویر و از دستور imshow برای نمایش تصویر استفاده می‌شود.

خروجی cv2:

خروجی matplotlib:

تفاوت این دو آن است که cv2 کانالهای رنگ را به فرمت bgr و matplotlib به فرمت rgb ذخیره می‌کند پس اگر تصویر دریافت شده از matplotlib را با دستور cv2.imshow نمایش بدهیم، خواهیم داشت:

این مشکل را با دستور cv2.cvtColor و ویژگی cv2.COLOR_RGB2BGR برطرف می‌سازیم.

سوال ۱) ب)

خروجی نشان می‌دهد که یک آرایه سه بعدی داریم که بعد اول آن ۶۸۰ درایه، بعد دوم ۱۰۲۰ درایه و بعد سوم آن ۳ درایه دارد. این بدین معناست که یک تصویر ۶۸۰*۱۰۲۰ پیکسلی سه کاناله داریم.

سوال ۱) ج)

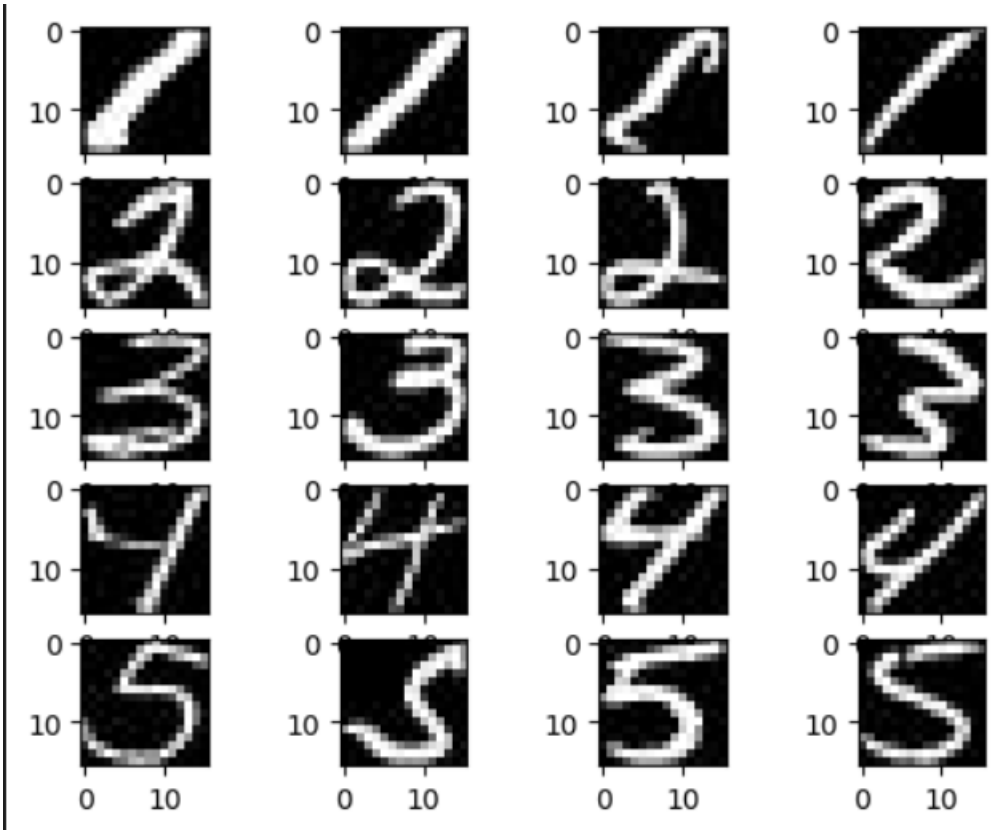
با توجه به اینکه بعد از ساختن پوشه موردنظر، اگر دوباره دستور os.makedirs را اجرا کنیم ارور دریافت می‌کنیم پس از try استفاده می‌کنیم. تصاویر را با استفاده از cv2.cvtColor و آرگومان COLOR_BGR2GRAY سیاه و سفید می‌کنیم. تصاویر را با دستور cv2.imwrite ذخیره می‌کنیم. با استفاده از دستورات open,write,close و تبدیل ابعاد آرایه به رشته حروف، فایل متنی موردنظر را ذخیره می‌کنیم.

سوال ۲)

تابع resize را با استفاده از cv2.resize پیاده‌سازی می‌کنیم. در تابع crop هرکدام از ۵ تصویر را به ۱۱۰۰ تصویر مجزا تبدیل می‌کنیم. در واقع تصویر را با step هایی به اندازه cropsize طی و جدا می‌کنیم. با استفاده از np.sum مطمئن می‌شویم تصویر جدا شده خالی نیست.

تصاویر را می‌خوانیم و با ویژگی cv2.IMREAD_GRAYSCALE مطمئن می‌شویم shape تصویر صحیح است. سپس تابع کراپ کردن را فراخوانی می‌کنیم.

با np.random.randint(1100) یک اینتجر رندوم از ۰ تا ۱۱۰۰ تولید می‌کنیم، همچنین از یک پلات ۴ در ۵ با دستور fig.add_subplot(5,4,k) برای نمایش دادن خروجی استفاده می‌کنیم:



سوال ۳) الف)

در پیاده سازی تابع create_matrix(n) از

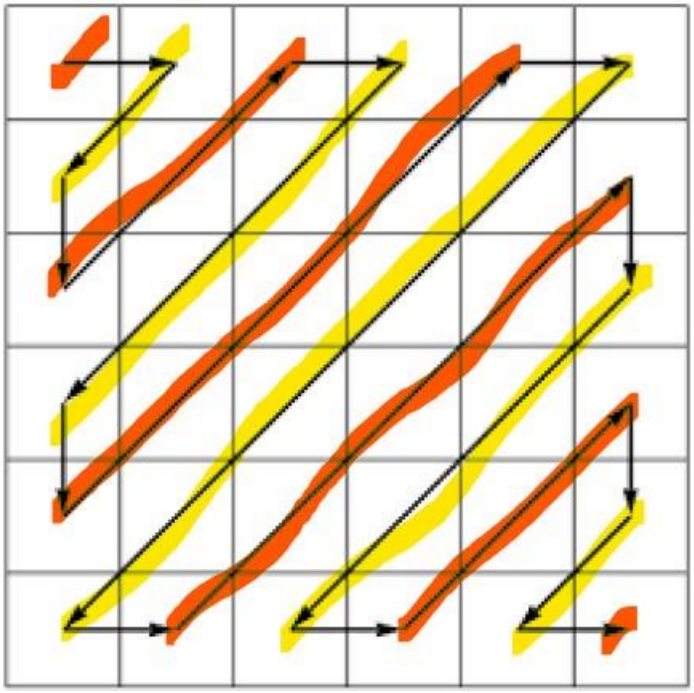
np.random.randint(n,n+100, size=(n, n)) استفاده کردیم که آرایه‌ای به شکل n*n بازمی‌گرداند که درایه‌های آن با اینتجرهای تصادفی در بازه [n,n+100) پر شده‌اند.

```
[[ 62  98  41 106  55  64  90  91]
 [ 38  91  47  24  58  95  63  60]
 [ 57  77  83  13 100  45  17  55]
 [ 69  35   8  20  52  48  94  39]
 [ 23  35  82  86  90  67  77  78]
 [ 99  73  14 101  87  59  64  31]
 [107  48   9  73  15  82  54  40]
 [104  53  23  98  44  42  41  64]]
```

پس از مسطح کردن ماتریکس با دستور matrix.flatten() تعداد هر عدد را در ارقام درایه‌های ماتریس می‌شماریم.

```
{0: 11, 1: 15, 2: 9, 3: 13, 4: 18, 5: 15, 6: 10, 7: 13, 8: 13, 9: 14}
```

سوال ۳) ب)



در این تصویر خط‌های مورب نارنجی و زرد را روی ماتریس می‌بینیم. روی هر خط مورب i+j یکسان است و می‌بینیم خانه‌ای که i+j بیشتری دارد دیرتر پیمایش شده. بین خانه‌هایی که i+j برابر دارند، در خطوط نارنجی خانه‌هایی که در پیمایش عادی دیرتر دیده می‌شوند، در پیمایش زیگزاگی زودتر پیمایش می‌شود و در خطوط زرد برعکس. خطوط زرد خانه‌هایی را شامل می‌شوند که باقیمانده i+j آنها بر ۲ یک می‌باشد و خطوط نارنجی باقیمانده i+j آنها بر ۲ صفر می‌باشد. این منطق در کد پیاده‌سازی شده است. هر [] در solution نشان دهنده خانه‌هایی با i+j یکسان می‌باشد که به ترتیب می‌باشند. اگر i+j%2=0 آنگاه عضو جدید به اول لیست می‌رود و در غیراینصورت به آخر لیست می‌رود. در نهایت درایه‌های این آرایه دو بعدی به ترتیب چاپ می‌شوند:

```
traverse_matrix(matrix)
#print output
```

✓ 15m 9.1s Python

```
62 98 38 57 91 41 106 47 77 69 23 35 83 24 55 64 58 13 8 35 99 107 73 82 20 100 95 90 91 63 45 52 86 14 48 104 53 9 101 90 48 17 60 55 94 67 87 73 23
98 15 59 77 39 78 64 82 44 42 54 31 40 41 64
```

سوال ۴) الف)

با استفاده از np.transpose ماتریس را ترانهاده می‌کنیم و با دستور np.dot ضرب ماتریس‌ها را انجام می‌دهیم و از np.linalg.inv برای ماتریس وارون استفاده می‌کنیم. خروجی:

```
[[0.796875 1.796875 2.796875]
 [4.       5.       6.      ]
 [7.203125 8.203125 9.203125]]
```

سوال ۴) ب)

با استفاده از دو فور تو در تو عمل لغزش را انجام داده و در هر iteration یک آرایه ۳*۳ با جایگاه مناسب از ماتریس B جدا کرده تا ضرب و جمع موردنظر را انجام دهیم و حاصل را در نهایت در ماتریس جواب قرار دهیم.

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

سوال ۵)

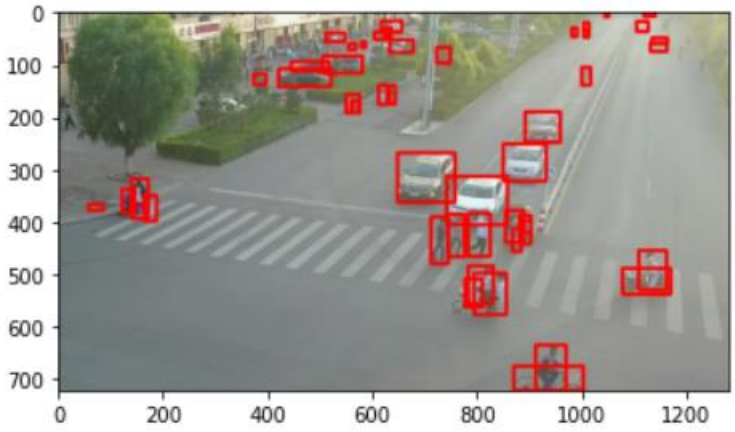
ابعاد تصویر را np.shape، نوع داده را image.dtype، مینیموم را np.min، ماکسیموم را np.max و میانگین را np.mean نشان می‌دهد. همچنین با [:,:,0] تمام درایه‌های مربوط به کانال رنگ آبی نمایش داده می‌شود.

```
shape: (721, 1281, 3)
dtype: uint8
min: 0
max: 255
mean: 139.98719468688319
second dimension zero indices: [[126 113 152 ... 154 154 154]
 [128 128 149 ... 154 153 154]
 [135 156 148 ... 153 153 153]
 ...
 [122 122 123 ... 139 138 139]
 [123 124 124 ... 139 139 139]
 [123 123 123 ... 138 137 137]]
```

آرایه detentions ۴۹ آیتم دارد که هر کدام دو بعدی می‌باشند. درایه اول بُعد اول تا درایه چهارم بعد اول به ترتیب x_min,y_min,width,height را نشان می‌دهند. و بُعد دوم نیز میزان اطمینان از تشخیص خود (امتیاز) را نشان می‌دهد. همچنین classes آرایه‌ای می‌باشد که classid مربوط به هرکدام از ۴۹ شئ را در خود دارد.

```
class: 0, x_min: 912, y_min: 634, width: 58, height: 86, score: 0.9171572327613831
class: 0, x_min: 783, y_min: 383, width: 42, height: 82, score: 0.9002619385719299
class: 0, x_min: 712, y_min: 390, width: 33, height: 89, score: 0.8785350322723389
class: 0, x_min: 1000, y_min: 105, width: 17, height: 34, score: 0.8591136336326599
class: 0, x_min: 744, y_min: 385, width: 32, height: 81, score: 0.8549011945724487
class: 0, x_min: 794, y_min: 497, width: 62, height: 79, score: 0.8065704703330994
class: 0, x_min: 140, y_min: 317, width: 32, height: 76, score: 0.7626901268959045
class: 0, x_min: 164, y_min: 349, width: 24, height: 49, score: 0.6349918246269226
class: 0, x_min: 1044, y_min: 0, width: 8, height: 9, score: 0.611725389957428
class: 0, x_min: 1109, y_min: 454, width: 53, height: 75, score: 0.6056272387504578
class: 0, x_min: 612, y_min: 139, width: 17, height: 37, score: 0.5378798842430115
class: 0, x_min: 981, y_min: 30, width: 10, height: 17, score: 0.48285484313964844
class: 0, x_min: 633, y_min: 141, width: 11, height: 34, score: 0.44777053594589233
class: 0, x_min: 783, y_min: 483, width: 48, height: 72, score: 0.42681318521499634
class: 0, x_min: 854, y_min: 378, width: 35, height: 60, score: 0.33320730924606323
class: 0, x_min: 1004, y_min: 34, width: 8, height: 14, score: 0.3231087327003479
class: 0, x_min: 624, y_min: 30, width: 12, height: 18, score: 0.322853147983551
class: 0, x_min: 881, y_min: 387, width: 22, height: 32, score: 0.3011518716812134
class: 0, x_min: 562, y_min: 169, width: 13, height: 23, score: 0.26651692390441895
class: 0, x_min: 1004, y_min: 20, width: 10, height: 28, score: 0.24979643523693085
class: 1, x_min: 870, y_min: 674, width: 133, height: 46, score: 0.9566846489906311
class: 1, x_min: 868, y_min: 414, width: 16, height: 42, score: 0.6192728877067566
class: 1, x_min: 604, y_min: 38, width: 25, height: 14, score: 0.5917620658874512
class: 1, x_min: 560, y_min: 170, width: 14, height: 22, score: 0.3744150400161743
class: 1, x_min: 982, y_min: 31, width: 9, height: 16, score: 0.37189358472824097
...
class: 11, x_min: 723, y_min: 67, width: 26, height: 30, score: 0.5689176917076111
class: 25, x_min: 511, y_min: 40, width: 37, height: 16, score: 0.44780242443084717
class: 25, x_min: 122, y_min: 337, width: 24, height: 45, score: 0.23393459618091583
class: 28, x_min: 777, y_min: 508, width: 33, height: 54, score: 0.2846028506755829
```

ابزاری برای مستطیل کشیدن روی تصویر cv2.rectangle می‌باشد. آرگومان‌هایی که به این تابع داده‌ایم به ترتیب بدین معنا می‌باشند: تصویر موردنظر، ابتدای مستطیل (چپ بالا)، انتهای مستطیل (راست پایین)، رنگ و ضخامت. در نهایت تصویر را همانند سوالات قبلی ذخیره می‌کنیم و نمایش می‌دهیم.




```
# Check if file has been saved or not
!(ls result.png && echo yes) || echo no
```

```
result.png
yes
```