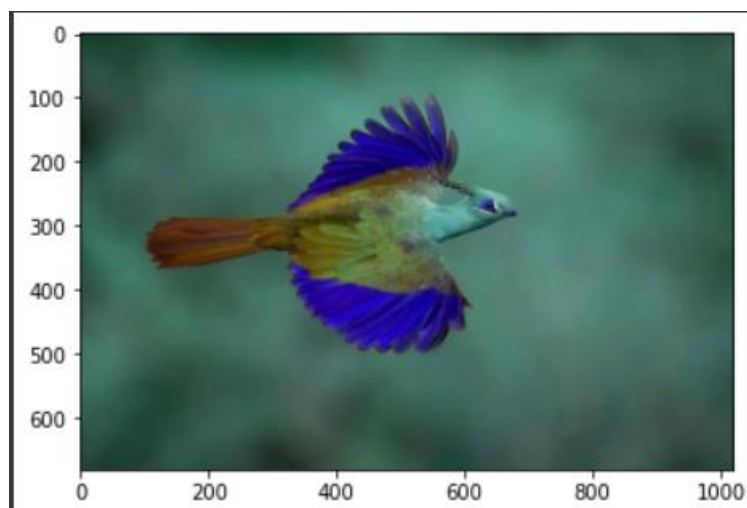


1) الف) با استفاده از `cv2.imread` عکس را خوانده و با `cv2.imshow()` نمایش می‌دهیم و خروجی زیر را دریافت می‌کنیم.



در ادامه با کتابخانه `matplotlib` تصویر را نمایش می‌دهیم.

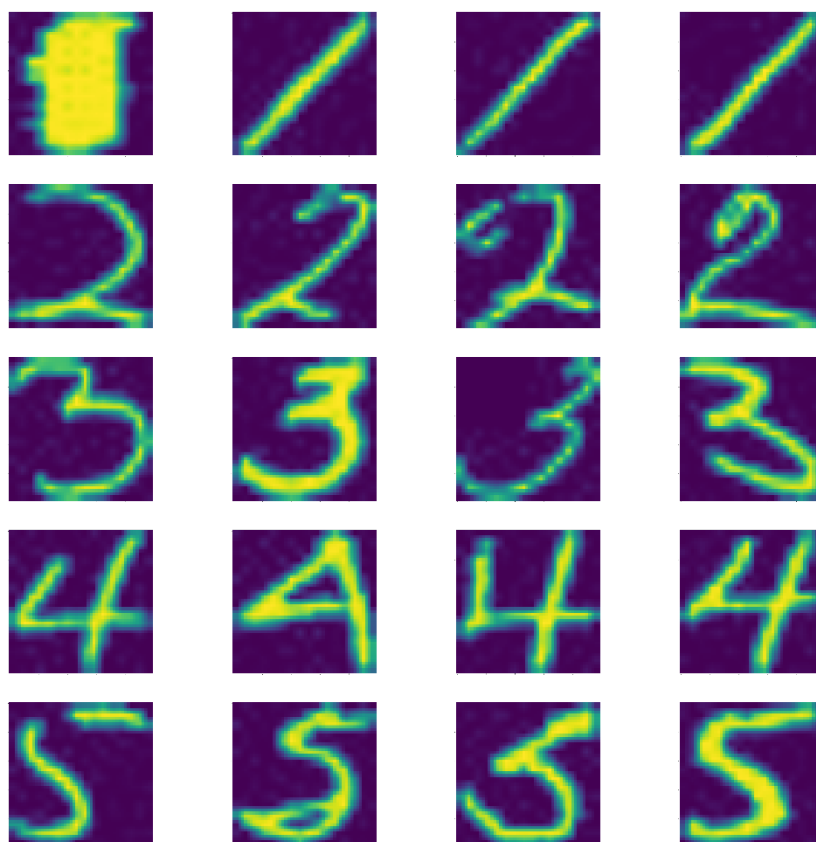


تفاوت دو عکس در رنگ آن هاست. برا اینکه رنگ این دو عکس یکی شود چون می‌دانیم که `cv2` رنگ‌ها را `bgr` سیو میکند با استفاده از تابع `cvtColor` در ارگومان ورودی دوم آن `cv2.COLOR_RGB2BGR` می‌دهیم و خروجی مانند عکس اول می‌شود.

ب) در این قسمت خروجی ما طول و عرض و تعداد کانال‌های عکس است. که برابر با `(680,1020,3)` است.

(ج) در این قسمت بعد از ساخت پوشه، فایل هایی که پسوند آن ها jpeg. است را خوانده و بعد با cvtColor به آرگومان دوم آن cv2.COLOR_BGR2GRAY را می دهیم تا سیاه و سفید شود عکس. سپس این عکس را با استفاده از تابع imwrite سیو میکنیم. برای ابعاد عکس هم از shape استفاده کرده و در فایلی که با open() میسازیم سیو میکنیم.

(2) در تابع resize با استفاده از تابع اماده cv2.resize() اندازه عکس را تغییر میدهیم. در crop هر بار یک ردیف را که کراپ میکنیم به سطر بعدی میرویم. در قسمت بعد عکس هارا خوانده و باید resize کنیم تا در assert به مشکل نخوریم. سپس درقسمت بعد همه ی 5 عکس را کراپ میکنیم و در لیست All_cropped_images سیو میکنیم. در ادامه با استفاده از figure چندین عکس بدست آمده (20تا) را درکنار هم قرار میدهیم. بعد از آن ابتدا یک عدد رندوم انتخاب کرده و در لیست عکس ها، عکس متناظر را resize کرده و در fig اضافه میکنیم.



(3) در قسمت create_matrix با استفاده از np.random.randint() که ورودی اول آن ابتدا و ورودی دوم آن انتهای بازه ای که اعداد رندوم میخواهیم قرار دهیم و ورودی سوم آن طول و عرض ماتریس امان است. در قسمت بعد ورودی را گرفته و ماتریس را میسازیم و آن را پرینت می کنیم.

```
3
[[49 19 50]
 [25 37 32]
 [85 18  4]]
{'4': 2, '9': 2, '1': 2, '5': 3, '0': 1, '2': 2, '3': 2, '7': 1, '8': 2}
```

در تابع `count_digits` روی تک تک خانه های ماتریس رفته و آن را به استرینگ کست میکنیم تا تک تک ارقام اعداد را داشته باشیم و در دیکشنریمان که تعداد هر عدد را سیو کرده ایم را نگه میداریم.

در تابع `traverse_matrix` هر بار که به نصفه ی ماتریس برسیم، مقدار `z` را زیاد میکنیم. در غیر این صورت اگر به سمت پایین حرکت میگردیم، `i++` و `z` میشود و اگر به سمت بالا حرکت کنیم `++j` , `--i` میشود. و تا زمانی ادامه میدهیم که تعداد خانه های دیده شده به اندازه ی توان 2 طول ماتریس باشد. خروجی :

```
[(0, 0), (0, 1), (1, 0), (2, 0), (1, 1), (0, 2), (0, 3), (1, 2), (2, 1), (3, 0), (4, 0), (3, 1), (2, 2), (1, 3), (0, 4), (0, 5), (1, 4), (2, 3), (3, 2), (4, 1), (5, 0), (5, 1), (4, 2), (3, 3), (2, 4), (1, 5), (2, 5), (3, 4), (4, 3), (5, 2), (5, 3), (4, 4), (3, 5), (4, 5), (5, 4), (5, 5)]
```

4) در این سوال با استفاده از تابع `dot` دو ماتریس را در هر ضرب کرده . `T` را با استفاده از `transpose` حساب میکنیم و در آخر با استفاده از `inv` تابع را `reverse` میکنیم. خروجی:

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

در قسمت بعد ابتدا یک ماتریس جواب میسازیم و در ادامه ماتری 3 در 3 ای داریم که نگه دارنده ی مقدار ضرب ماتریس های رو هم قرار گرفته و ضر درایه های آن است. بعد با `sum` جمع درایه ها را سیو میکنیم. خروجی:

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

(5) در این قسمت به ترتیب shape , dtype, mean, max,min قسمتی از ماتریس در هر یک از کانال ها را پرینت میکنیم. خروجی:

```
(721, 1281, 3)
uint8
139.98719468688319
255
0
[[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]]
#####
[[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]] [[147 136 175 ... 191 191 191]
 [149 149 172 ... 191 190 191]
 [156 177 171 ... 190 190 190]
 ...
 [121 121 122 ... 142 141 142]
 [122 123 123 ... 142 142 142]
 [122 122 122 ... 141 140 140]] [[144 132 171 ... 189 189 187]
 [146 146 168 ... 189 188 187]
 [153 174 167 ... 188 188 186]
 ...
 [107 107 108 ... 140 139 140]
 [108 109 109 ... 140 140 140]
 [108 108 108 ... 139 138 138]]
```

در قسمت بعد به مستطیل امان ابتدا عکس مورد نظر بعد موقعیت بالا چپ مستطیل و بعد هم موقعیت پایین راست را میدهیم. خروجی:

