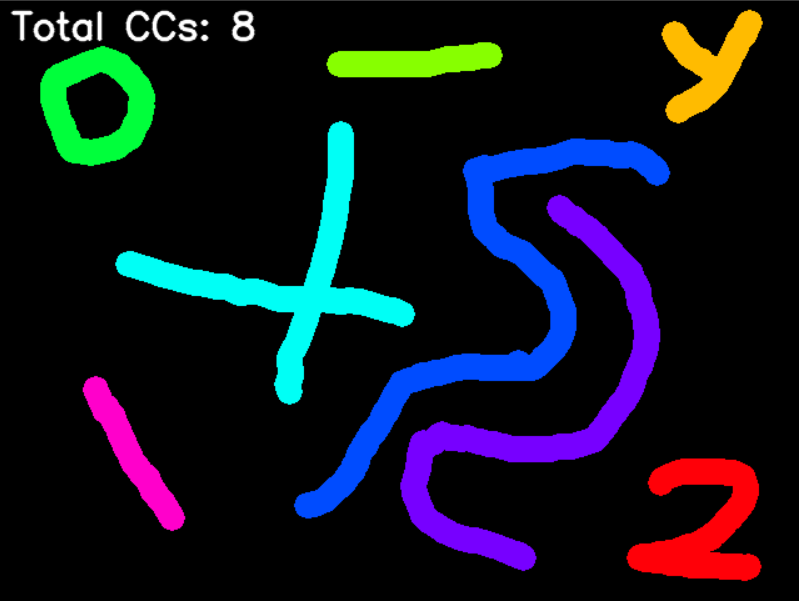
به نام خالق رنگین کمان

ستاره باباجانی – گزارش تمرین سری 5

سوال 1: کد خواسته شده زده شد و برای هر خط آن کامنت مناسب برای درک بهتر آن، قرار داده شد. حال خروجی به دست آمده را در گزارش قرار میدهم:



سوال 2:

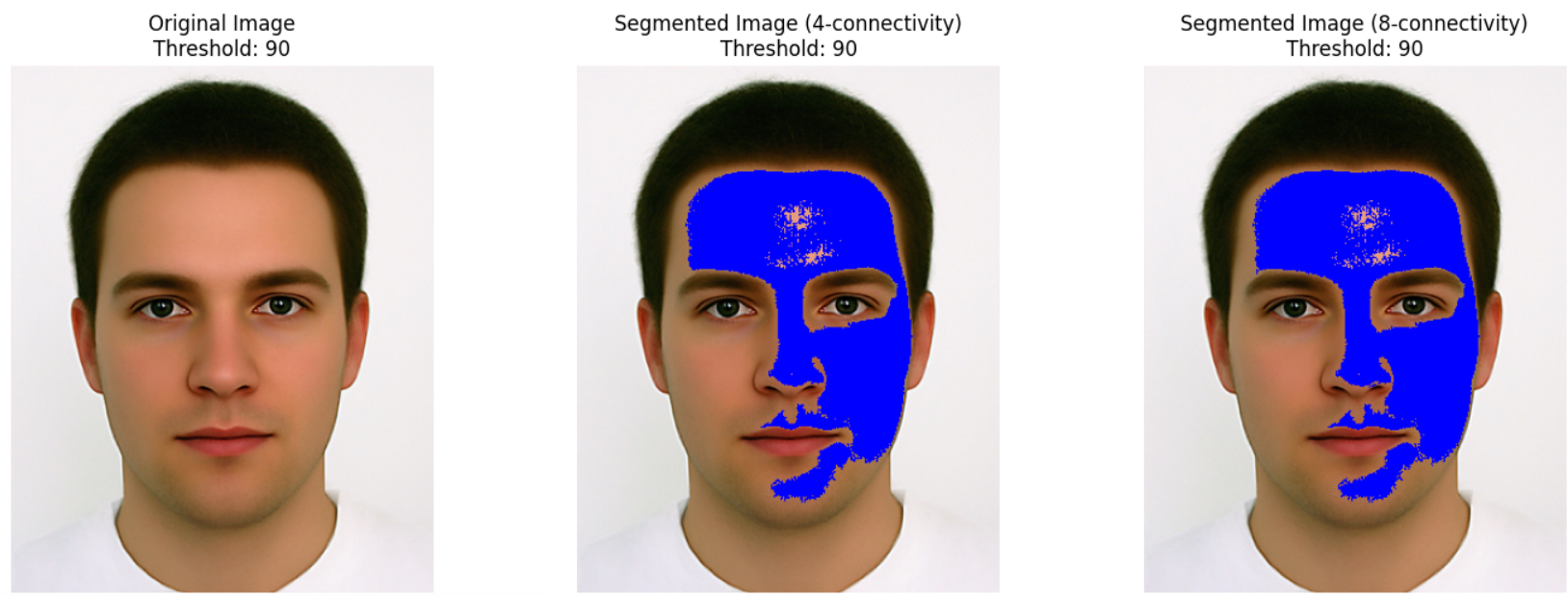
* کنترل آستانه: مقدار آستانه برای کنترل حساسیت الگوریتم در حال رشد منطقه به تفاوت رنگ بسیار مهم است. آستانه‌های پایین‌تر منجر به تقسیم‌بندی محافظه‌کارانه‌تر می‌شوند و مناطقی را که از نظر رنگ بسیار شبیه به نقطه بذر هستند، می‌گیرند. آستانه های بالاتر به الگوریتم اجازه می دهد تا مناطقی با تفاوت رنگ بیشتر را شامل شود.
* 4-Connectivity در مقابل 8-Connectivity: 4-Connectivity محدودتر است، فقط پیکسل های مجاور را می گیرد. این حالت می تواند منجر به مناطق دقیق تر اما بالقوه کوچکتر شود.8-Connectivity فراگیرتراست. این حالت اغلب منجر به ایجاد مناطق تقسیم‌بندی شده بزرگ‌تر، ثبت جزئیات دقیق‌تر و مناطق وسیع‌تر می‌شود.

حال به بررسی نتایج آستانه های مختلف میپردازیم:

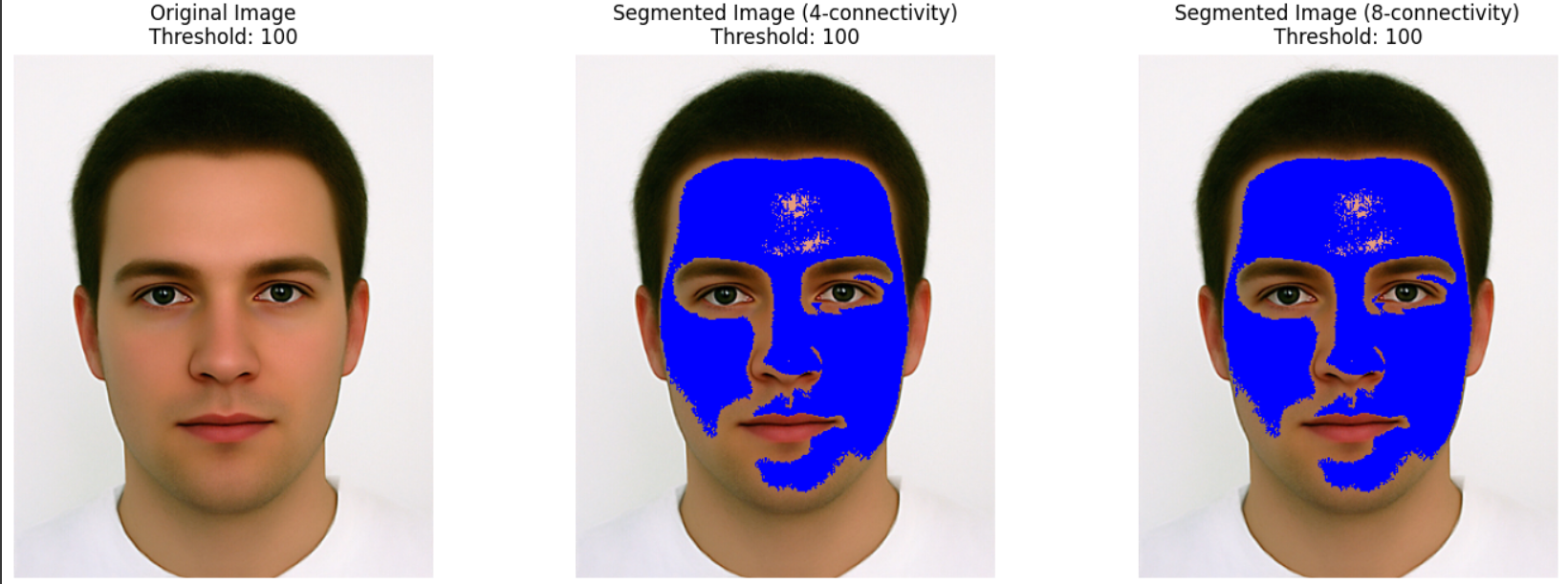
1. آستانه 80:



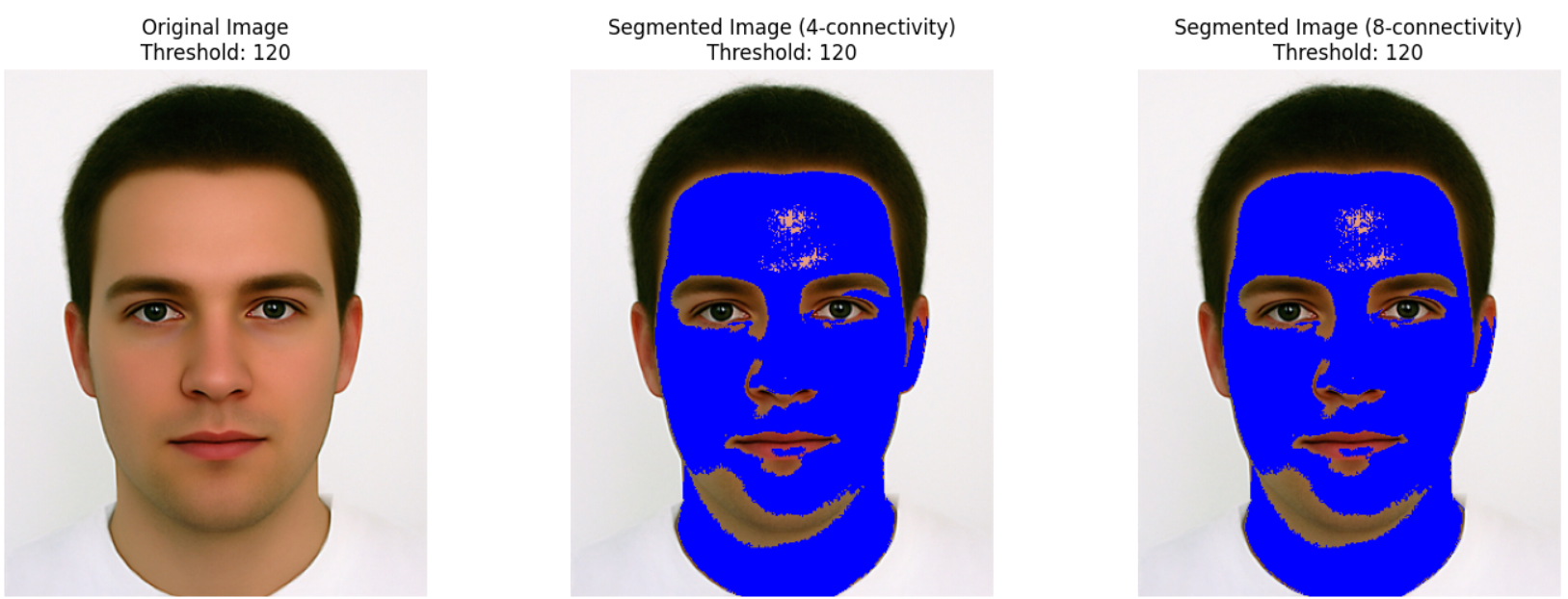
1. آستانه 90:



1. آستانه 100:



1. آستانه 120:



سوال 3: ابتدا بصورت رندوم یک تصویر با شرایط گفته شده ایجاد میکنیم:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 12 | 7 | 9 | 1 |
| 6 | 10 | 5 | 8 | 4 |
| 3 | 14 | 15 | 11 | 7 |
| 13 | 2 | 8 | 5 | 3 |
| 12 | 6 | 9 | 7 | 10 |

سپس مراحل الگوریتم otsu را در پیش میگیریم:

* محاسبه هیستوگرام:

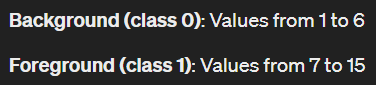


* Probability distribution: تقسیم frequency بر تعداد کل پیکسل ها(25):

P(1) = 0.04, P(2) = 0.08, P(3) = 0.08, P(4) = 0.04, P(5) = 0.08, P(6) = 0.08, P(7) = 0.12, P(8) = 0.08, P(9) = 0.08, P(10) = 0.08, P(11) = 0.04, P(12) = 0.08, P(13) = 0.04, P(14) = 0.04, P(15) = 0.04

* Class Probabilities and Means:

1. حد آستانه 6: میدانیم:



پس خواهیم داشت:

W0 = Sum(P[0:6]) = 0.04 + 0.08 + 0.08 + 0.04 + 0.08 + 0.08 = 0.4

W1 = Sum(P[7:15]) = 1 - W0 = 0.6

پس:

μ0 = (1\*0.04 + 2\*0.08 + 3\*0.08 + 4\*0.04 + 5\*0.08 + 6\*0.08) / 0.4

= (0.04 + 0.16 + 0.24 + 0.16 + 0.4 + 0.48) / 0.4

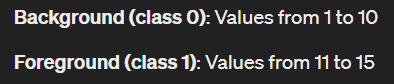
= 3.7

μ1 = (7\*0.12 + 8\*0.08 + 9\*0.08 + 10\*0.08 + 11\*0.04 + 12\*0.08 + 13\*0.04 + 14\*0.04 + 15\*0.04) / 0.6

= (0.84 + 0.64 + 0.72 + 0.8 + 0.44 + 0.96 + 0.52 + 0.56 + 0.6) / 0.6

= 10.8

1. حد آستانه 10: میدانیم:



پس خواهیم داشت:

W0 = Sum(P[0:10]) = 0.04 + 0.08 + 0.08 + 0.04 + 0.08 + 0.08 + 0.12 + 0.08 + 0.08 + 0.08 = 0.76

W1 = Sum(P[11:15]) = 1 - W0 = 0.24

پس:

μ0 = (1\*0.04 + 2\*0.08 + 3\*0.08 + 4\*0.04 + 5\*0.08 + 6\*0.08 + 7\*0.12 + 8\*0.08 + 9\*0.08 + 10\*0.08) / 0.76

= (0.04 + 0.16 + 0.24 + 0.16 + 0.4 + 0.48 + 0.84 + 0.64 + 0.72 + 0.8) / 0.76

= 5.89

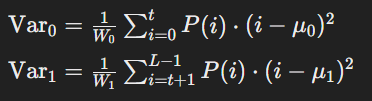
μ1 = (11\*0.04 + 12\*0.08 + 13\*0.04 + 14\*0.04 + 15\*0.04) / 0.24

= (0.44 + 0.96 + 0.52 + 0.56 + 0.6) / 0.24

= 12.83

* Inter-class Variance: اگر داشته باشیم:





آنگاه برای هر حد آستانه خواهیم داشت:

1. حد آستانه 6:

Var0​ = 2.81

Var1​=6.83

* 0.4 \* 2.81 + 0.6 \* 6.83 = 5.222

1. حد آستانه 10:

Var0​=7.45

Var1​=1.81

* 0.76 \* 7.45 + 0.24 \* 1.81 = 6.096

که چون Inter-class Variance برای حد آستانه 6 کمتر است، پس حد آستانه بهتری میباشد.

سوال 4: همان طور که میدانیم آستانه گذاری افقی با cv2.adaptiveThreshold انجام میشود که ورودی های زیر را دارد:

* thresholdType: دو مقدار دارد که بیانگر سفید بودن یا نبودن رنگ پس زمینه است.
* BlockSize: ابعاد پنجره مورد استفاده در محاسبه میانگین. (هر چه بزرگتر باشد، آستانه گذاری بیشتر سراسری میشود.)
* C: مورد استفاده در شیفت دادن آستانه نسبت به میانگین پنجره.

حال به بررسی هر کدام از عکس ها میپردازیم:

1. عکس q4\_1: همان طور که در تصویر مشاهده میشود، نوشته های سمت چپ پایین مشخص نیستند( به خوبی binary نشده اند) پس blockSize بزرگی مثل 41 دارد و چون در مقایسه با بقیه کمی روشن تر است پس احتمالا C آن کوچکتر (مثل 5) دارد. همچنین thresholdType آن THRESH\_BINARY میباشد.
2. عکس q4\_2: همان طور که در تصویر مشاهده میشود، اکثر نواحی به دقت خوبی binary شده اند و واضح اند پس blockSize کوچکی مثل 21 دارد و چون کمی تیره تر از حالت عادی است پس احتمالا C آن بزرگ (مثل 30) است تا از حالت میانگین، تیره تر شود. همچنین thresholdType آن THRESH\_BINARY میباشد.
3. عکس q4\_3: همان طور که در تصویر مشاهده میشود، نوشته های سمت چپ مشخص نیستند( به خوبی binary نشده اند) پس blockSize بزرگی مثل 41 دارد و همچنین خطوط مداد و مورب خیلی پررنگ تر از حالت عادی هستند که نشان دهنده C بزرگ(مثل 30) میباشد. همچنین thresholdType آن THRESH\_BINARY میباشد.
4. عکس q4\_4: همان طور که در تصویر مشاهده میشود، اکثر نواحی به دقت خوبی binary شده اند و واضح اند پس blockSize کوچکی مثل 21 دارد و همچنین در مقایسه با تصویر q4\_2 خطوط مداد و مورب عادی هستند(پررنگ تر یا کم رنگ تر از میانگین نیستند) پس احتمالا C مقدار کوچکی مثل 5 دارد. همچنین thresholdType آن THRESH\_BINARY میباشد.
5. عکس q4\_5: همان طور که در تصویر مشاهده میشود، نوشته های سمت چپ پایین مشخص نیستند( به خوبی binary نشده اند) پس blockSize بزرگی مثل 41 دارد و چون کمی روشن است پس احتمالا C آن کوچکتر (مثل 5) دارد. همچنین thresholdType آن THRESH\_BINARY\_INVمیباشد زیرا پس زمینه سیاه شده است. (روشنایی های بیشتر از آستانه به صفر و کمتر از آستانه به 255، تناظر یافته اند)

سوال 5: ابتدا reflect padding را بر روی تصویر اولیه اعمال میکنیم:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | 22 | 33 | 22 | 22 | 33 | 22 | 22 | 22 | 22 |
| 22 | 22 | 33 | 22 | 22 | 33 | 22 | 22 | 22 | 22 |
| 22 | 22 | 33 | 33 | 33 | 33 | 33 | 33 | 22 | 22 |
| 22 | 22 | 44 | 33 | 22 | 33 | 22 | 22 | 22 | 22 |
| 22 | 22 | 22 | 33 | 22 | 44 | 33 | 22 | 22 | 22 |
| 22 | 22 | 33 | 44 | 22 | 22 | 44 | 22 | 22 | 22 |
| 22 | 22 | 33 | 33 | 44 | 22 | 44 | 22 | 33 | 33 |
| 22 | 22 | 33 | 22 | 33 | 33 | 33 | 33 | 33 | 33 |
| 44 | 44 | 22 | 44 | 22 | 33 | 44 | 33 | 33 | 33 |
| 44 | 44 | 22 | 44 | 22 | 33 | 44 | 33 | 33 | 33 |

در مرحله بعد عملگرهای سایش و گسترش را بر روی تصویر بالا اعمال میکنیم.

برای سایش عنصر ساختاری را بدون تغییر روی خانه به خانه تصویر اعمال میکنیم. تصویر حاصل از عملگر سایش:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | 22 | 33 | 22 | 22 | 33 | 22 | 22 | 22 | 22 |
| 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 |
| 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 |
| 22 | 22 | 22 | 22 | 33 | 22 | 22 | 22 | 22 | 22 |
| 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 |
| 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 |
| 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 33 |
| 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 33 |
| 44 | 22 | 22 | 22 | 22 | 33 | 33 | 33 | 33 | 33 |
| 44 | 44 | 22 | 44 | 22 | 33 | 44 | 33 | 33 | 33 |

برای گسترش باید در مرحله اول structure element را 180 درجه نسبت به مرکزش بچرخانیم و سپس آن را برروی تصویر اعمال کنیم. که حاصل این کار به ما این عنصر ساختاری را میدهد:

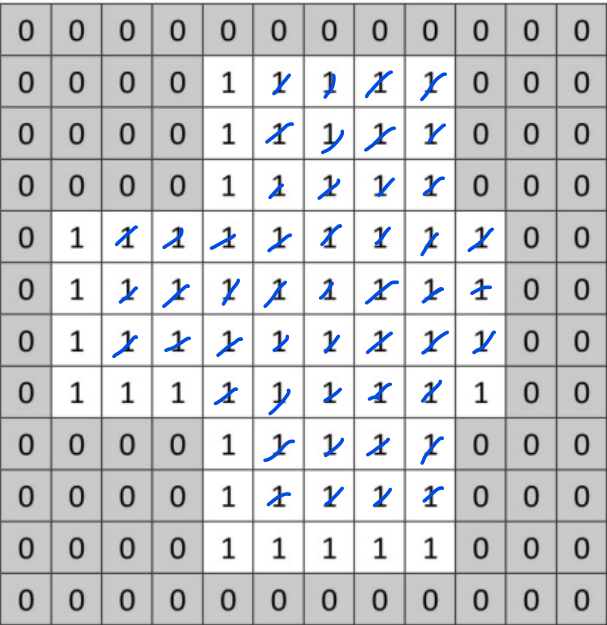
|  |  |  |
| --- | --- | --- |
| 1 | 1 | 1 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |

تصویر حاصل از عملگر گسترش:

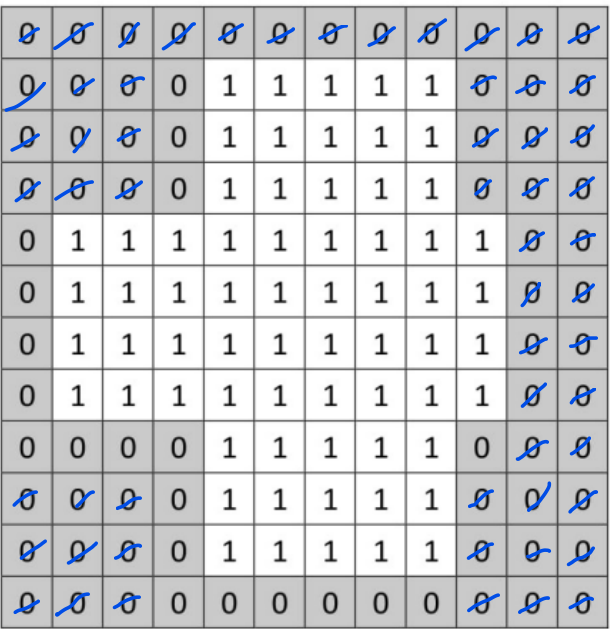
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | 22 | 33 | 22 | 22 | 33 | 22 | 22 | 22 | 22 |
| 22 | 33 | 33 | 33 | 33 | 33 | 33 | 33 | 33 | 22 |
| 22 | 44 | 44 | 44 | 33 | 33 | 33 | 33 | 33 | 22 |
| 22 | 22 | 33 | 44 | 44 | 44 | 44 | 33 | 33 | 22 |
| 22 | 33 | 44 | 44 | 44 | 44 | 44 | 44 | 22 | 22 |
| 22 | 33 | 33 | 44 | 44 | 44 | 44 | 44 | 33 | 22 |
| 22 | 33 | 33 | 33 | 44 | 44 | 33 | 44 | 33 | 33 |
| 22 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 33 | 33 |
| 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 33 | 33 |
| 44 | 44 | 22 | 44 | 22 | 33 | 44 | 33 | 33 | 33 |

سوال 6: ابتدا مرحله به مرحله محاسبات را انجام میدهیم:

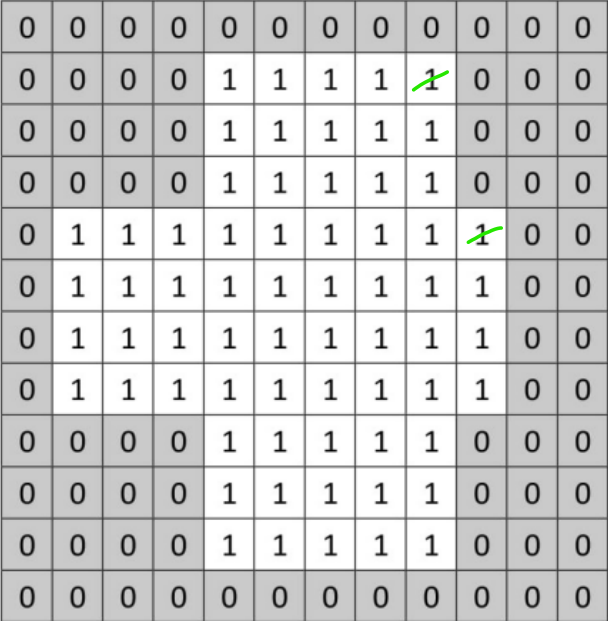
* :



* :



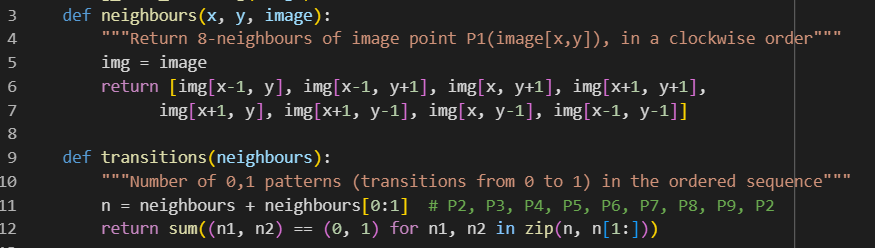
* حال اشتراک آنها را به دست میاوریم:



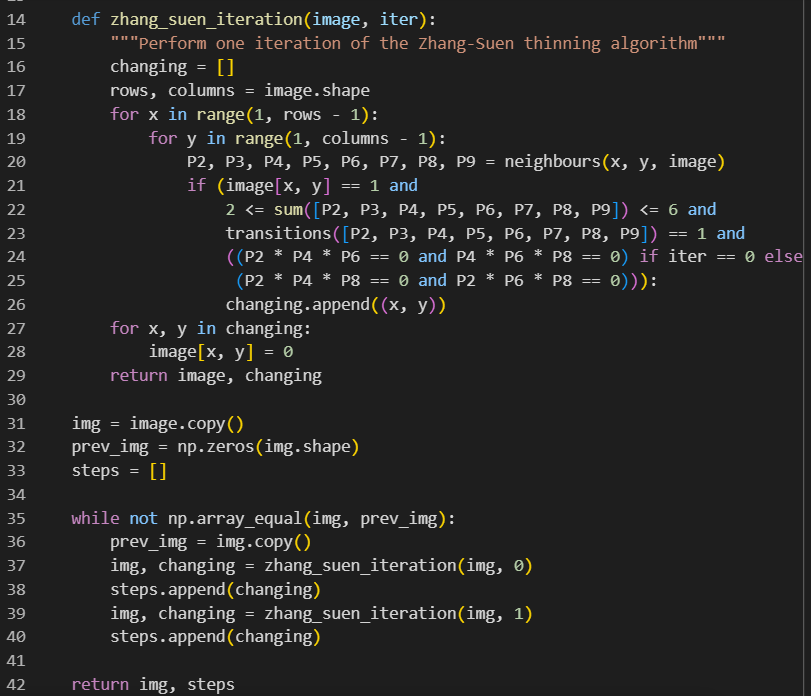
که این دو پیکسل پیدا میشوند(گوشه های سمت راست بالا را پیدا میکند.)

سوال 8: الف) مراحل ساخت اسکلت تصاویر به شرح زیر است:

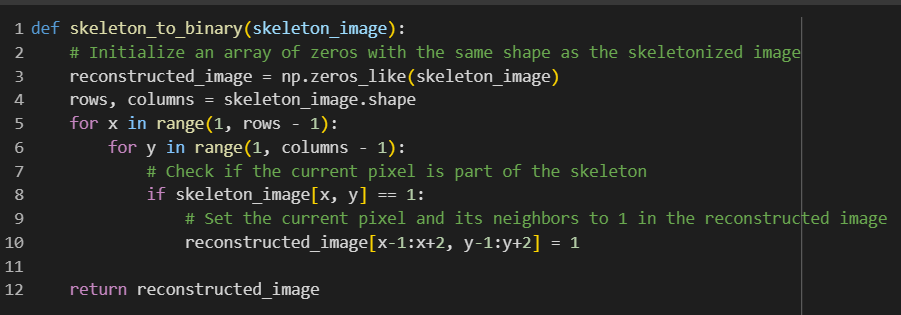
* پیاده سازی توابع neighbours و transitions:



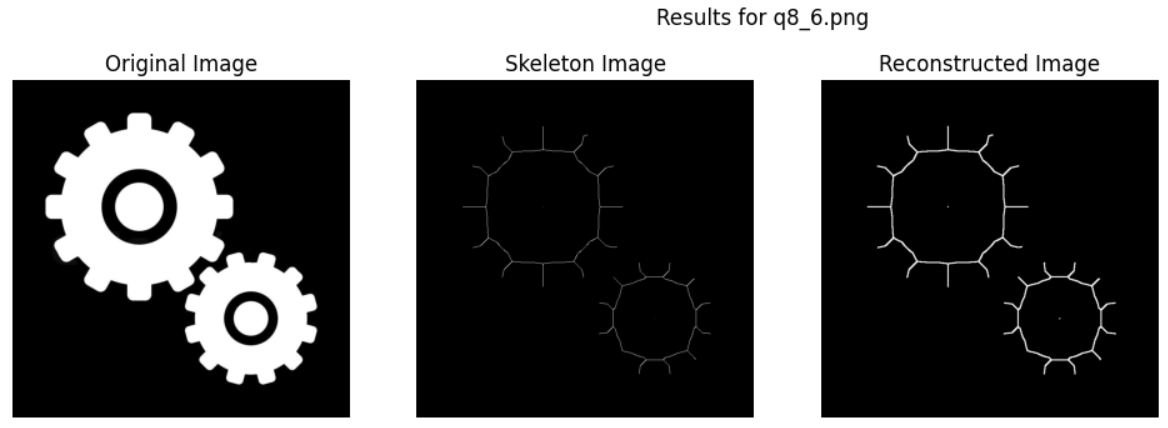
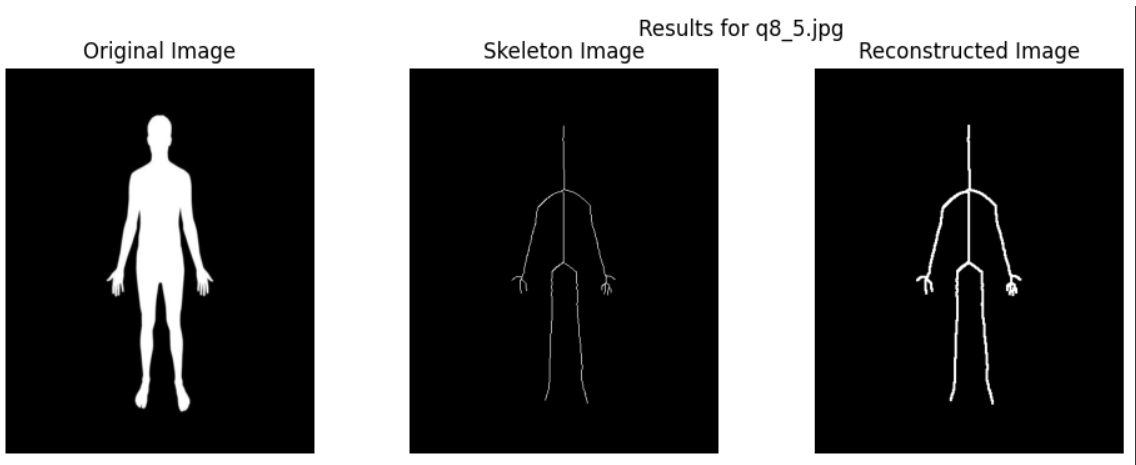
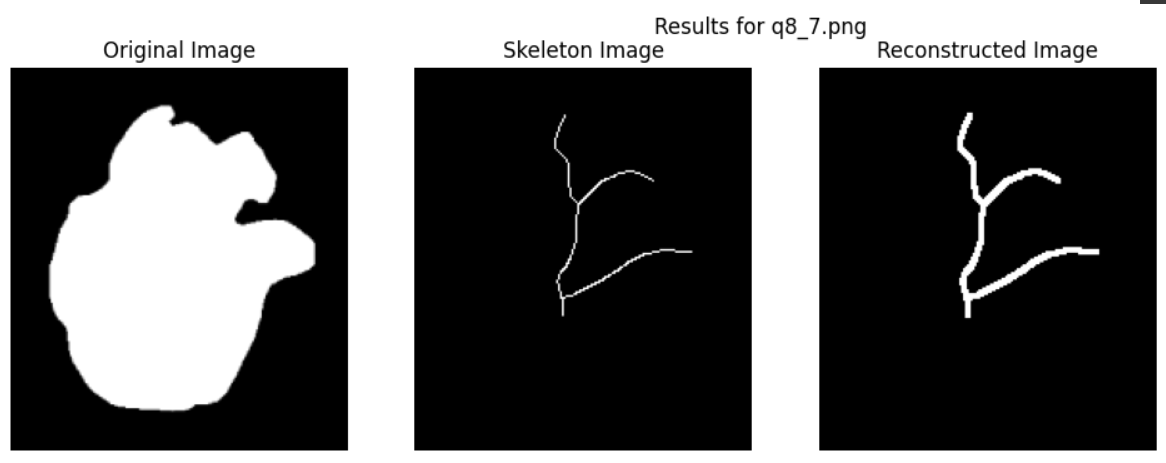
* تابع اصلی:



ب) بازسازی عکس اصلی:



خروجی ها به شرح زیر هستند:

* 
* 
* 

سوال 9: با کمک عناصر ساختاری زیر، مرزهای 4 جهت راست، چپ، پایین، بالا، را بدست می آوریم. سپس بعد از اعمال hit or miss هر یک از عناصر را با هم اجتماع گرفته و نقاط مرزی نهایی را بدست می آوریم:

|  |  |  |
| --- | --- | --- |
| 0 | -1 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | -1 | 0 |

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| -1 | 1 | 0 |
| 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | -1 |
| 0 | 0 | 0 |

* مرز چپ:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* مرز راست:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* مرز بالا:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* مرز پایین:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

تصویر نهایی حاصل از اجتماع 4 حالت مرزی بالا: که خانه های زرد رنگ جواب نهایی بوده و نقاط مرزی تصویر هستند.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |