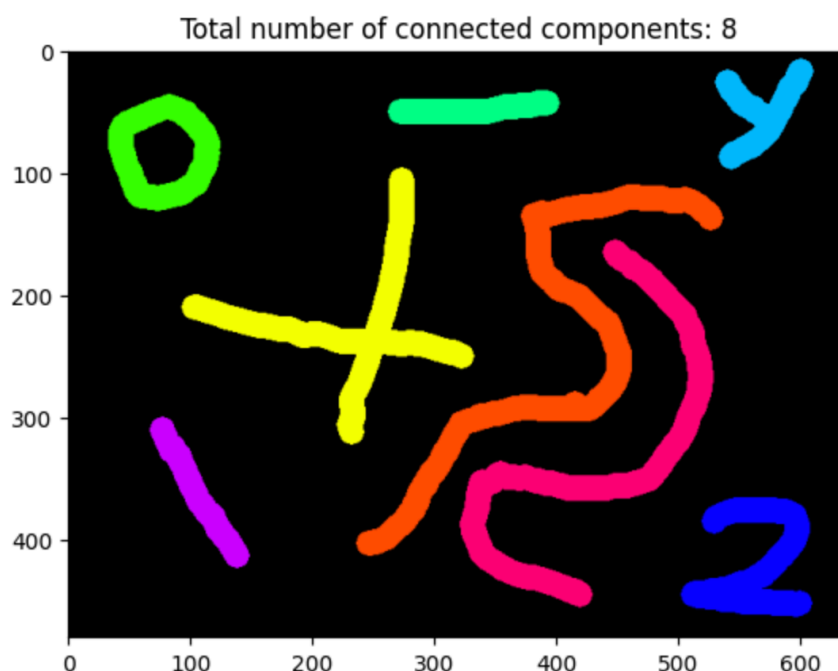


۱.

گزارش کد:

ابتدا تصویر را بارگذاری می‌کنیم. اگر تصویر باینری نباشد، از آستانه‌گذاری برای تبدیل آن به باینری استفاده می‌کنیم.  
از تابع `connectedComponents` برای یافتن و برچسب‌گذاری اجزای متصل استفاده می‌کنیم. این تابع تعداد برچسب‌ها و تصویر برچسب‌گذاری شده را برمی‌گرداند.  
هر برچسب به یک رنگ در فضای رنگ HSV تبدیل می‌شود و سپس به BGR برای نمایش تبدیل می‌شود.  
حال کافیست تصویر برچسب‌گذاری شده با رنگ‌های متفاوت را نمایش دهیم.

تصویر خروجی:



۲.

گزارش کد:

ابتدا تصویر با استفاده از `cv2.imread` بارگذاری شده و سپس با استفاده از `cv2.cvtColor` به فرمت RGB تبدیل می‌شود تا رنگها به درستی نمایش داده شوند. تابع `plot_image` از کتابخانه `matplotlib` برای نمایش تصویر استفاده می‌کند. در الگوریتم رشد منطقه، ابتدا تصویری برای بخش‌بندی کردن تهیه می‌شود. نقطه‌ی شروع و مقدار رنگ آن نقطه مشخص می‌شود. یک ماتریس باینری برای ذخیره‌ی مناطق بخش‌بندی شده ایجاد می‌شود و نقطه‌ی شروع به عنوان اولین نقطه در لیست پردازش قرار می‌گیرد. در حالی که لیست پردازش شامل نقاطی است، الگوریتم هر نقطه را پردازش کرده و همسایه‌های آن را بر اساس حالت همسایگی (4 یا 8) بررسی می‌کند. اگر مقدار رنگ همسایه در محدوده‌ی آستانه باشد، آن همسایه به منطقه‌ی بخش‌بندی شده اضافه می‌شود و به لیست پردازش افزوده می‌شود. در نهایت، رنگ منطقه‌ی بخش‌بندی شده به رنگ دلخواه (قرمز) تغییر داده می‌شود. تعریف نقطه شروع و آستانه:

نقطه شروع و مقدار آستانه برای الگوریتم بخش‌بندی مشخص می‌شود. این مقادیر باید بر اساس تصویر مورد نظر تنظیم شوند. اجرای الگوریتم بخش‌بندی: تابع `segment` برای هر دو حالت همسایگی (4 و 8) اجرا می‌شود تا دو تصویر بخش‌بندی شده با همسایگی‌های مختلف تولید شود.

نمایش نتایج و مقایسه:

۱. آستانه ۳۰:

همسایگی ۴:

در این تصویر، نواحی قرمز فقط در محدوده‌ی نزدیک به نقطه‌ی شروع و در نواحی‌ای که شدت رنگی مشابه دارند رشد کرده است. این نتیجه به دلیل استفاده از همسایگی ۴ است که فقط پیکسل‌های بالا، پایین، چپ و راست نقطه‌ی جاری را بررسی می‌کند. در نتیجه، نواحی قرمز به صورت مقطع و گسسته گسترش یافته‌اند.

همسایگی ۸:

در این تصویر، نواحی قرمز به دلیل استفاده از همسایگی ۸ که علاوه بر پیکسل‌های همسایه‌ی ۴، پیکسل‌های مورب را نیز شامل می‌شود، گسترش بیشتری داشته‌اند. این باعث شده است که مناطق بیشتری به هم متصل شوند و ناحیه‌ی قرمز بزرگتر و پیوسته‌تری ایجاد شود.

۲. آستانه ۵۰:

همسایگی ۴:

در این تصویر، نواحی قرمز به طور چشمگیری بزرگتر شده‌اند. این افزایش به دلیل افزایش آستانه است که اجازه می‌دهد پیکسل‌هایی با تفاوت رنگی بیشتری نسبت به نقطه‌ی شروع به منطقه‌ی بخش‌بندی شده افزوده شوند. استفاده از همسایگی ۴ باعث شده که این مناطق با گسترش کمتری نسبت به همسایگی ۸ پیوسته شوند.

همسایگی ۸:

در این تصویر، نواحی قرمز به طور قابل توجهی گسترش یافته و تقریباً کل صورت را پوشش داد‌اند. افزایش آستانه باعث شده که پیکسل‌های بیشتری به منطقه افزوده شوند. همسایگی ۸ با پوشش دادن پیکسل‌های مورب، مناطق بزرگتر و پیوسته‌تری ایجاد کرده است.

### نتیجه‌گیری:

افزایش آستانه باعث می‌شود که نواحی بزرگتری بخش‌بندی شوند زیرا پیکسل‌های بیشتری به دلیل تفاوت رنگی کمتر نسبت به نقطه‌ی شروع پذیرفته می‌شوند. استفاده از همسایگی ۸ نواحی پیوسته‌تر و بزرگتری نسبت به همسایگی ۴ ایجاد می‌کند.

threshold 30

4-connectivity

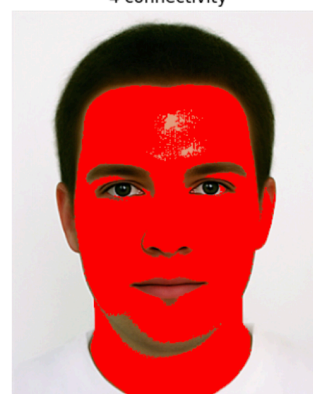


8-connectivity



threshold 50

4-connectivity



8-connectivity



## 1. Random Matrix

$$\begin{bmatrix} 3 & 7 & 12 & 5 & 10 \\ 2 & 6 & 8 & 9 & 14 \\ 1 & 4 & 11 & 13 & 15 \\ 5 & 6 & 9 & 7 & 8 \\ 3 & 10 & 2 & 12 & 4 \end{bmatrix}$$

## 2. Histogram

Value:            1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
 Frequency: 1 2 2 2 2 2 2 2 2 2 1 2 1 1 1

## 3. Calculate the Probability Distribution

$P(i) = \text{Frequency}(i) / \text{Total number of pixels}$   
 Total pixels = 25

$p(1)=.04$ ,  $p(2)=.08$ ,  $p(3)=.08$ ,  $p(4)=.08$ ,  $p(5)=.08$ ,  $p(6)=.08$ ,  $p(7)=.08$ ,  
 $p(8)=.08$ ,  $p(9)=.08$ ,  $p(10)=.08$ ,  $p(11)=.04$ ,  $p(12)=.08$ ,  $p(13)=.04$ ,  
 $p(14)=.04$ ,  $p(15)=.04$

## 4. Compute Class Probabilities and Means for Thresholds 6 and 10

○ Threshold  $t = 6$

**Class 0 (Below threshold):**

Values: 1, 2, 3, 4, 5, 6

Probability  $w_0(6)$ :

$$w_0(6) = P(1) + P(2) + P(3) + P(4) + P(5) + P(6) = 0.04 + 0.08 + 0.08 + 0.08 + 0.08 + 0.08 = 0.44$$

Mean  $\mu_0(6)$ :

$$\begin{aligned}\mu_0(6) &= (1 * 0.04 + 2 * 0.08 + 3 * 0.08 + 4 * 0.08 + 5 * 0.08 + 6 * 0.08) / w_0(6) \\ &= (0.04 + 0.16 + 0.24 + 0.32 + 0.40 + 0.48) / 0.44 = 1.64 / 0.44 = 3.73\end{aligned}$$

**Class 1 (Above threshold):**

Values: 7, 8, 9, 10, 11, 12, 13, 14, 15

Probability  $w_1(6)$ :

$$w_1(6) = 1 - w_0(6) = 1 - 0.44 = 0.56$$

Mean  $\mu_1(6)$ :

$$\begin{aligned}\mu_1(6) &= (7 * 0.08 + 8 * 0.08 + 9 * 0.08 + 10 * 0.08 + 11 * 0.04 + 12 * 0.08 + 13 * 0.04 + 14 * 0.04 + 15 * 0.04) / w_1(6) \\ &= (0.56 + 0.64 + 0.72 + 0.80 + 0.44 + 0.96 + 0.52 + 0.56 + 0.60) / 0.56 = 5.80 / 0.56 = 10.36\end{aligned}$$

○ Threshold  $t=10$

**Class 0 (Below threshold):**

Values: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Probability  $w_0(10)$ :

$$\begin{aligned}w_0(10) &= P(1) + P(2) + P(3) + P(4) + P(5) + P(6) + P(7) + P(8) + P(9) + P(10) \\ &= 0.04 + 0.08 + 0.08 + 0.08 + 0.08 + 0.08 + 0.08 + 0.08 + 0.08 + 0.08 = 0.76\end{aligned}$$

Mean  $\mu_0(10)$ :

$$\begin{aligned}\mu_0(10) &= (1 * 0.04 + 2 * 0.08 + 3 * 0.08 + 4 * 0.08 + 5 * 0.08 + 6 * 0.08 + 7 * 0.08 + 8 * 0.08 + 9 * 0.08 + 10 * 0.08) / w_0(10) \\ &= (0.04 + 0.16 + 0.24 + 0.32 + 0.40 + 0.48 + 0.56 + 0.64 + 0.72 + 0.80) / 0.76 = 4.36 / 0.76 = 5.74\end{aligned}$$

**Class 1 (Above threshold):**

Values: 11, 12, 13, 14, 15

Probability  $w_1(10)$ :

$$w_1(10) = 1 - w_0(10) = 1 - 0.76 = 0.24$$

Mean  $\mu_1(10)$ :

$$\begin{aligned}\mu_1(10) &= (11 * 0.04 + 12 * 0.08 + 13 * 0.04 + 14 * 0.04 + 15 * 0.04)/w_1(10) \\ &= (0.44 + 0.96 + 0.52 + 0.56 + 0.60)/0.24 = 3.08/0.24 = 12.83\end{aligned}$$

## 5: Calculate Intra-class Variance

**For  $t=6$ :**

**Variance of Class 0:**

$$\sigma_0^2(6) = \frac{\sum_{i=1}^6 (i - \mu_0(6))^2 \cdot P(i)}{w_0(6)}$$

$$\sigma_0^2(6) = \frac{(1-3.73)^2 \cdot 0.04 + (2-3.73)^2 \cdot 0.08 + (3-3.73)^2 \cdot 0.08 + (4-3.73)^2 \cdot 0.08 + (5-3.73)^2 \cdot 0.08 + (6-3.73)^2 \cdot 0.08}{0.44}$$

$$\sigma_0^2(6) = \frac{6.80 \cdot 0.04 + 2.99 \cdot 0.08 + 0.53 \cdot 0.08 + 0.07 \cdot 0.08 + 1.62 \cdot 0.08 + 5.15 \cdot 0.08}{0.44}$$

$$\sigma_0^2(6) = \frac{0.27 + 0.24 + 0.04 + 0.01 + 0.13 + 0.41}{0.44}$$

$$\sigma_0^2(6) = \frac{1.10}{0.44} = 2.50$$

**and we do the same for Class 1:**

$$\sigma_1^2(6) = \frac{\sum_{i=7}^{15} (i - \mu_1(6))^2 \cdot P(i)}{w_1(6)}$$

$$\sigma_1^2(6) = \frac{0.90 + 0.45 + 0.15 + 0.01 + 0.02 + 0.22 + 0.28 + 0.53 + 0.86}{0.56}$$

$$\sigma_1^2(6) = \frac{3.42}{0.56} = 6.11$$

**Intra-class variance for  $t = 6$**

$$\sigma_w^2(6) = 0.44 \cdot 2.50 + 0.56 \cdot 6.11 = 1.10 + 3.42 = 4.52$$

**For  $t = 10$ :**

**Variance of Class 0:**

$$\sigma_0^2(10) = \frac{\sum_{i=1}^{10} (i - \mu_0(10))^2 \cdot P(i)}{w_0(10)}$$

$$\sigma_0^2(10) = \frac{0.90 + 1.12 + 0.60 + 0.24 + 0.04 + 0.01 + 0.13 + 0.41 + 0.85 + 1.46}{0.76}$$

$$\sigma_0^2(10) = \frac{5.76}{0.76} = 7.58$$

**Variance of Class 1:**

$$\sigma_1^2(10) = \frac{\sum_{i=11}^{15} (i - \mu_1(10))^2 \cdot P(i)}{w_1(10)}$$

$$\sigma_1^2(10) = \frac{0.13 + 0.06 + 0.002 + 0.05 + 0.19}{0.24}$$

$$\sigma_1^2(10) = \frac{0.43}{0.24} = 1.79$$

**Intra-class variance for  $t = 10$ :**

$$\sigma_w^2(10) = 0.76 \cdot 7.58 + 0.24 \cdot 1.79 = 5.76 + 0.43 = 6.19$$

**The threshold level  $t=6$  has a lower intra-class variance compared to  $t=10$ . Therefore, threshold level  $t=6$  is better.**

C: [5, 30]

blockSize: [21, 41]

thresholdType: [THRESH\_BINARY, THRESH\_BINARY\_INV]

## تصویر اول:

با توجه به این که این تصویر نسبت به سایرین روشن تر است، احتمالاً C کوچکتری (۵) دارد. از آن جا که نوشته های تصویر در برخی نواحی کمی محو به نظر می رسد و تصویر کاملاً باینری نشده است پس از blockSize بزرگتر (۴۱) استفاده شده است.

## تصویر دوم:

با توجه به این که این تصویر نسبت به سایرین تیره تر است، احتمالاً C بزرگتری (۳۰) دارد. از آن جا که تصویر کنتراست بالایی دارد و به خوبی باینری شده است، از blockSize کوچکتری (۲۱) استفاده کرده است.

## تصویر سوم:

خطوط این تصویر کمی bold به نظر می رسد، پس احتمالاً C بزرگتری (۳۰) دارد. همچنین مانند تصویر اول، این تصویر نیز در برخی نواحی کمی محو به نظر می رسد و تصویر کاملاً باینری نشده است پس از blockSize بزرگتر (۴۱) استفاده شده است.

## تصویر چهارم:

این تصویر دارای ظاهری بسیار باینری است که نشان دهنده اندازه بلوک ۲۱ است. خطوط نه خیلی پررنگ و نه خیلی نازک هستند، بنابراین مقدار  $C=5$  مناسب است. این مقدار تضمین می کند که آستانه تقریباً میانگین پنجره است و تصویر را متعادل و واضح نشان می دهد.

## تصویر پنجم:

پس زمینه تصویر سیاه است، در حالی که در عکس اصلی پس زمینه سفید است. این نشان می دهد که نوع آستانه استفاده شده، برخلاف ۴ تصویر قبلی THRESH\_BINARY\_INV است، که تصویر باینری را معکوس می کند به طوری که پس زمینه سیاه و پیش زمینه سفید می شود. قسمت پایین سمت چپ تصویر به خوبی باینری نیست، که نشان دهنده استفاده از اندازه بلوک ۴۱ است. علاوه بر این، مقدار  $C=5$  تضمین می کند که آستانه تقریباً میانگین پنجره است که به حفظ جزئیات تصویر کمک می کند.



۵.

۱. اعمال reflect padding:

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

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

۲. اعمال عملگر سایش: اعمال عنصر ساختاری بدون تغییر.

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

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

۳. اعمال عملگر گسترش: اعمال عنصر ساختاری پس از چرخش ۱۸۰ درجه.

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

۶. همانطور که در شکل آخر پیداست، تنها دو پیکسل در اشتراک عبارات وجود دارند که گوشه‌های بالا و سمت راست تصویر را نشان می‌دهند. لذا می‌توان گفت کارکرد این عملیات، تشخیص گوشه‌های سمت راست بالای تصویر می‌باشد.

$$(A \ominus B_1)$$

[illegible]

$$(A^c \ominus B_2)$$

[illegible]

$$(A \ominus B_1) \cap (A^c \ominus B_2)$$

[illegible]

۷.

۸.

گزارش کد:

مرحله ۱: بارگذاری و دودویی‌سازی تصویر  
در این مرحله، تصویر را بارگذاری کرده و به یک تصویر دودویی تبدیل می‌کنیم. این کار با استفاده از کتابخانه OpenCV انجام می‌شود.

مرحله ۲: نمایش تصویر  
تابعی برای نمایش تصویر با استفاده از matplotlib داریم.

مرحله ۳: استخراج همسایگان  
تابعی برای استخراج همسایگان ۸ تایی یک پیکسل در تصویر تعریف می‌کنیم.  
استخراج همسایگان: این تابع ۸ همسایه پیکسل را به ترتیب ساعتگرد باز می‌گرداند.

مرحله ۴: شمارش انتقالات (ترنزیشن) ۱-۰  
تابعی برای شمارش تعداد انتقالات ۱-۰ در دنباله همسایگان.

مرحله ۵: نازک‌سازی به روش Zhang-Suen  
الگوریتم نازک‌سازی Zhang-Suen را پیاده‌سازی می‌کنیم.  
مراحل اجرای الگوریتم نازک‌سازی Zhang-Suen:  
مرحله 1:

انتخاب پیکسل‌های کاندیدا: در این مرحله، هر پیکسل که مقدار آن 1 است و در اطراف آن بین 2 تا 6 پیکسل با مقدار 1 وجود دارد، به عنوان کاندیدا انتخاب می‌شود.  
تعداد تغییرات 0 به 1: تعداد تغییرات از 0 به 1 در لیست همسایگان 8-تایی پیکسل مورد نظر باید برابر با 1 باشد.

شرایط حذف پیکسل: سه شرط دیگر برای حذف پیکسل وجود دارد:  
مقدار پیکسل‌های P4، P2، و P6 نباید همزمان برابر با 1 باشد.  
مقدار پیکسل‌های P6، P4، و P8 نباید همزمان برابر با 1 باشد.  
مرحله 2:

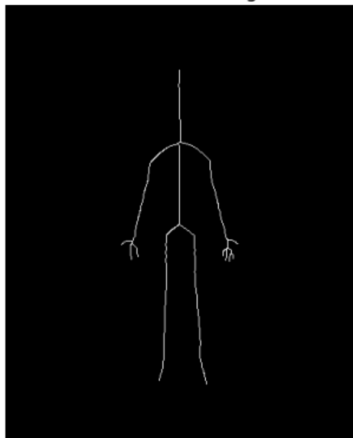
انتخاب پیکسل‌های کاندیدا: مشابه مرحله 1 است.  
تعداد تغییرات 0 به 1: مشابه مرحله 1 است.

شرایط حذف پیکسل: سه شرط برای حذف پیکسل:  
مقدار پیکسل‌های P4، P2 و P8 نباید همزمان برابر با 1 باشد.  
مقدار پیکسل‌های P6، P2 و P8 نباید همزمان برابر با 1 باشد.

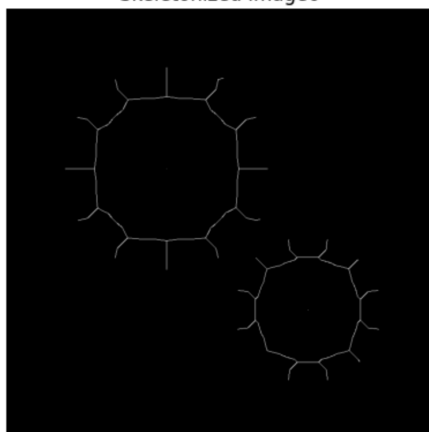
مرحله ۶: بازسازی تصویر  
برای بازسازی تصویر از اسکلت، از عملگر گسترش (dilation) استفاده می‌کنیم.

خروجی‌ها:

Skeletonized Image5



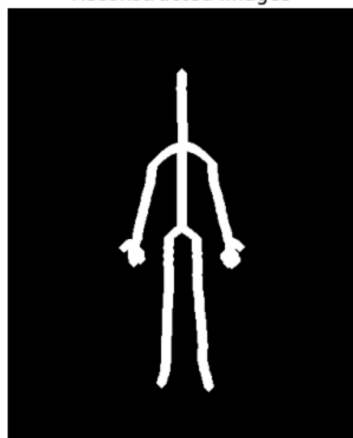
Skeletonized Image6



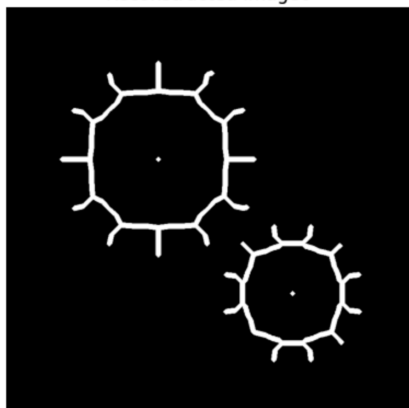
Skeletonized Image7



Reconstructed Image5



Reconstructed Image6

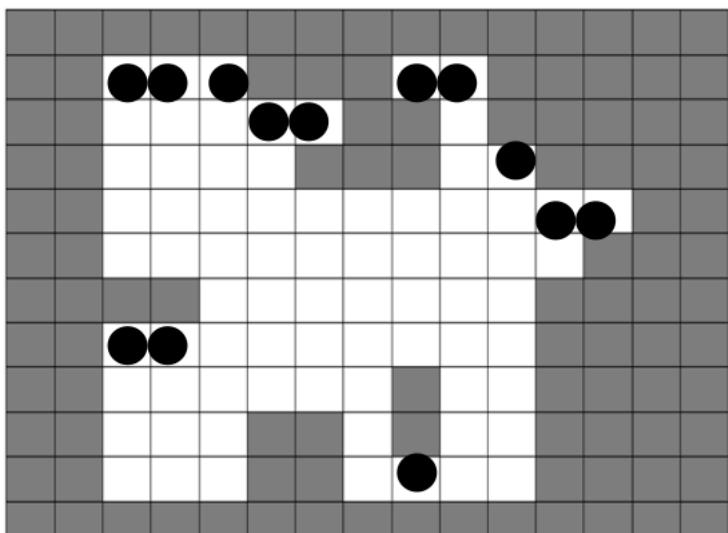


Reconstructed Image7



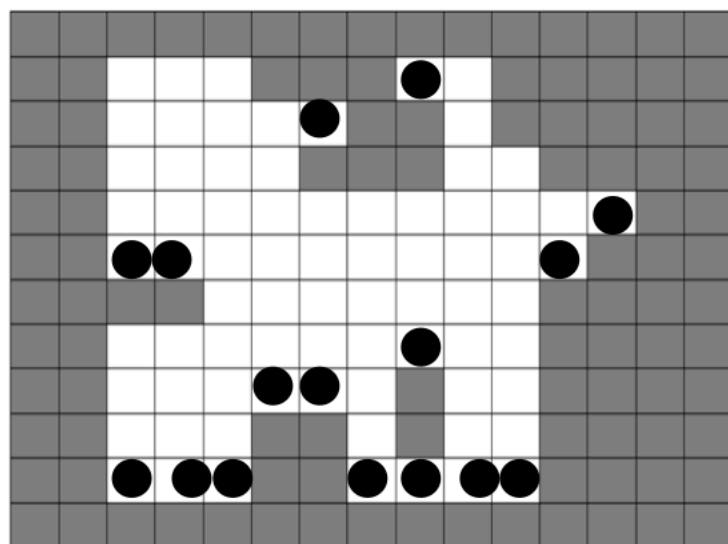
۹.

مرز بالا:



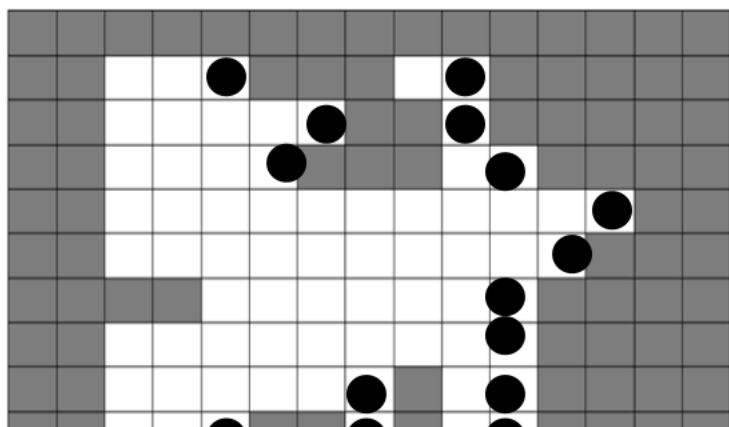
|   |    |   |
|---|----|---|
| 0 | -1 | 0 |
| 0 | 1  | 0 |
| 0 | 0  | 0 |

مرز پایین:



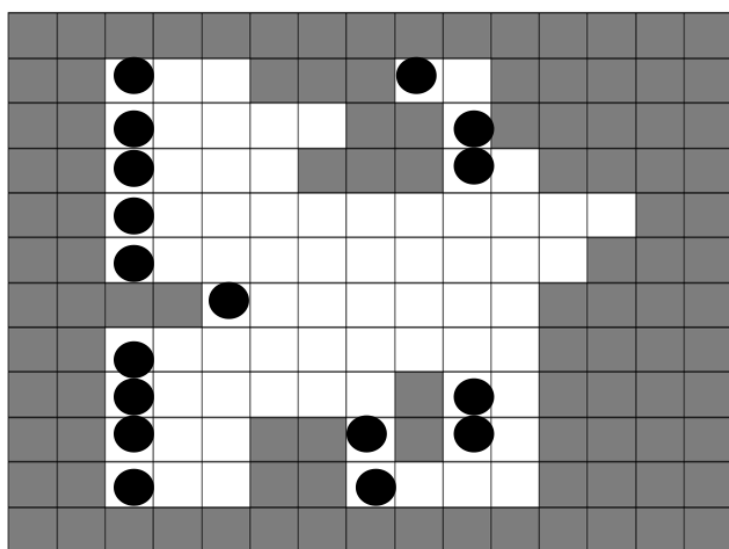
|   |    |   |
|---|----|---|
| 0 | 0  | 0 |
| 0 | 1  | 0 |
| 0 | -1 | 0 |

مرز راست:



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

مرز چپ:



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

اگر هر چهار عملگر بالا را بر روی تصویر اعمال کنیم، تمامی مرزهای تصویر کاور خواهد شد:

