به نام خدا

امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

گزارش تمرین سری پنجم بینایی کامپیوتر

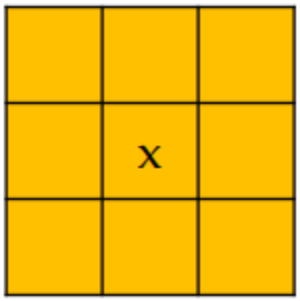
۱)

در این سوال قصد داریم connected components ها را با استفاده از رنگ های مختلف استخراج کنیم.ابتدا تصویر را خوانده ایم سپس تصویر را باینری کرده ایم.
سپس از تابع connectedcomponents موجود در کتابخانه cv2 استفاده کرده ایم به این تابع میتوانیم یک تصویر باینری بدهیم که به ما دو مقدار تعداد لیبل ها یا همان تعداد اجزای متصل را میدهد(num labels) و همچنین ماتریسی که پیکسل های مربوط به هر لیبل را درون خودش دارد(labels_im). سپس به تعداد لیبل هایی که یافتیم رنگ به صورت رندوم میسازیم.
در مرحله بعد به هر یک از لیبل هایی که داریم یک رنگ میدهیم این کار با استفاده از رنگ کردن پیکسل هایی که مربوط به هر component یا لیبل که در مرحله قبل پیدا کردیم انجام میشود.
سپس پلات مربوط به آنرا رسم میکنیم و تعداد را نیز مینویسیم و در نهایت خروجی را نمایش میدهیم.

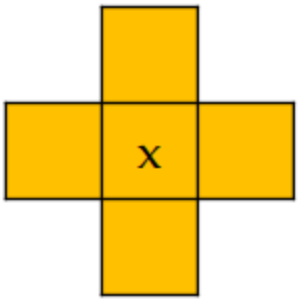
۲) برای الگوریتم های رشد ناحیه میتوانیم هم از استک استفاده کنیم و هم از صف که سودوکد هر دو در جزوه موجود میباشد. من از صف استفاده کردم که بیشتر توضیح داده شد.

ابتدا تصویر را خواندیم و سپس نمایش دادیم.
در تابع segment ابتدا از کپی عکس استفاده کردیم تا عکس اصلی دچار تغییر نشود.سپس یک نقطه را به عنوان نقطه بذر یا seed در نظر گرفتیم.سپس آنرا در صف اضافه کردیم.

سپس برای پیدا کردن نحوه اتصال پیکسل ها دو نوع connectivity – 4,8 را در نظر گرفتیم که لیست مربوط به آنرا همانند این تصویر در جزوه پر کردم.



8-connectivity



4-connectivity

منظور موقعیت مکانی نسبت به عنصر مرکزی است.

در مرحله بعدی شروع به پردازش اعضای وارد شده در صف کردیم و بدین صورت عمل کردیم که اگر اختلاف نواحی متصل طبق یکی از دو روش بالا از مقدار threshold کمتر باشد در نتیجه آن پیکسل نیز میتواند همرنگ بذر شود و همینطوری هر نقطه ای که به صف اضافه میشود میتواند این کار را با نواحی خود طبق روش های بالا انجام دهد.

در قسمت های بعد یک نقطه که مربوط به سمت راست صورت شخص بود را انتخاب کردم.

سپس چندین آستانه قرار دادم و در لیستی تمام حالت هایی که برای صورت شخص با آستانه های متفاوت به وجود می آمد قرار دادم و سپس با هر آستانه در هر دو نوع اتصال خروجی گرفتم.

نتیجه مشخص است هر چقدر مقدار آستانه بیشتر شود نواحی بیشتری قرمز رنگ میشوند و علت این مورد به این است که ما اجازه میدهیم تا اختلاف بیشتری بین نواحی قرمز شده و نقطه بذر برقرار باشد و کار برای قرمز رنگ شدن یعنی همرنگ نقطه بذر شدن است راحت تر میشود.

۳) در این سوال مقدار در هر دو آستانه ۶و۱۰ یکسان شد با عوض کردن با مقادیر دیگر مقادیر متفاوتی خواهیم داشت. اما به صورت کلی هر چقدر واریانس کمتر باشد بهتر است یعنی اگر واریانس ها در مثال دیگری مقادیر متفاوتی به دست می آوردند واریانس کمتر بهتر بود.

۳) تصویر ۵×۵ زیر داریم

۲	۸	۱۲	۱۴	۴
۲	۸	۱۲	۱۴	۴
۲	۸	۱۲	۱۴	۴
۲	۸	۱۲	۱۴	۴
۲	۸	۱۲	۱۴	۴

ابتدا برای threshold = ۶ در نظر می‌گیریم

کلاس اول کمتر از ۶ و کلاس دوم بیشتر از ۶

$$w_1 = \frac{10}{25}$$

میانگین کلاس اول $= \frac{30}{10} = 3$

$$w_2 = \frac{15}{25}$$

میانگین کلاس دوم $= \frac{170}{15} = 11.3$

$$\sigma_1^2 = \frac{(2-3)^2 \times 5 + (4-3)^2 \times 5}{25} = \frac{10}{25} = \frac{2}{5}$$

$$\sigma_2^2 = \frac{(8-11.3)^2 \times 5 + (12-11.3)^2 \times 5 + (14-11.3)^2 \times 5}{25}$$

$$= \frac{10.89 \times 5 + 0.49 \times 5 + 7.29 \times 5}{25} = 3.73$$

$$\sigma_w^2 = (0.4)^2 + (0.6) \times 3.73 = 1.5$$

برای آستانه ۱۰

کلاس ۱ کمتر از ۱۰   کلاس ۲ بیشتر از ۱۰

$$w_1 = \frac{15}{25}, \quad w_2 = \frac{10}{25}$$

میانگین ۱ $= \frac{70}{15} = 4.6$ ، میانگین ۲ : ۱۳

$$\sigma_1^2 = \frac{(2-4.6)^2 \times 5 + (4-4.6)^2 \times 5 + (8-4.6)^2}{25} = 3.73$$

(6.76, 0.36, 11.56)

$$\sigma_2^2 = \frac{(12-13)^2 \times 5 + (14-13)^2 \times 5}{25} = \frac{2}{5}$$

$$\sigma_w^2 = (0.6)(3.73) + (0.4)^2 = 1.5$$

برای هر دو مقدار یکسانی شد و در این مثال تفاوتی ندارند

۴) آستانه گذاری وفقی با استفاده از تابع cv2.adaptiveThreshold انجام میشود که دارای آرگومان های نام برده شده در سوال نیز میباشد.

thresholdType : برای سفید بودن یا نبودن بک گراند

Blocksize : اندازه پنجره ایست که برای محاسبه میانگین استفاده میشود و افزایش آن نشان دهنده سراسری تر شدن آستانه گذاری میباشد.

C : برای شیفت دادن آستانه نسبت به میانگین پنجره

به بررسی هر یک از تصاویر میپردازیم :

تصویر اول ، نوشته های سمت چپ پایین به خوبی واضح نیستند که نشان دهنده استفاده از یک blocksize بزرگ است که در آن آستانه گذاری به صورت سراسری بوده و باعث شده تصویر به خوبی باینری نشود.تصویر کمی روشن است که نشان دهنده c کمتر میباشد و همچنین پس زمینه تصویر چون سفید است از threshold_binary استفاده شده است.

blocksize = 41 , c=5

در تصویر دوم ، باینری شدن تصویر به خوبی انجام شده پس blocksize کمتر سراسری بوده در نتیجه میتوان گفت blocksize = 21 میباشد.خطوط کمی تیره تر و پررنگ تر پس اندازه C بزرگ بوده در این مثال C= 30 بوده و به دلیل سفید بودن بکگراندthreshold_binary داریم.

در تصویر سوم ، باینری شدن تصویر خوب نبوده مثل تصویر ۱ سمت چپ نوشته ها نامعلومند در نتیجه blocksize=41 و همچنین بعضی خطوط باز هم خیلی تیره و پررنگند در نتیجه C=30 و باز هم threshold-binary

در تصویر چهارم ، باینری شدن خوب بوده blocksize = 21. در رابطه با مقدار C ابهام دارم. اگر به صورت کلی باز هم بعضی خطوط پررنگند و نشان دهنده C زیاد هستند اما اگر به نسبت

تصویر ۲ بخواهیم مقایسه کنیم مقدار C ان کمتر است.اما به صورت کلی چون نسبت به میانگین و خطوط متن خیلی تفاوت در پررنگ بودن و غلظت نبوده است میتوانیم بگوییم c= 5 است .و threshold binary داریم.

در تصویر پنجم ، باینری شدن خوب نبوده پس blocksize = 41 همچنین پررنگی خاصی نداشته ایم که از میانگین بیشتر باشد پس c=5 و چون پس زمینه تار است threshold binary inv میباشد.

۵)

تصویر زیر با اعمال reflect padding به صورت زیر میشود(تصویر اصلی مشخص است):

۲۲	۲۲	۲۲	۲۲	۳۳	۲۲	۲۲	۳۳	۲۲	۲۲
۲۲	۲۲	۲۲	۲۲	۳۳	۲۲	۲۲	۳۳	۲۲	۲۲
۲۲	۲۲	۳۳	۳۳	۳۳	۳۳	۳۳	۳۳	۲۲	۲۲
۲۲	۲۲	۲۲	۲۲	۳۳	۲۲	۳۳	۴۴	۲۲	۲۲
۲۲	۲۲	۲۲	۳۳	۴۴	۲۲	۳۳	۲۲	۲۲	۲۲
۲۲	۲۲	۲۲	۴۴	۲۲	۲۲	۴۴	۳۳	۲۲	۲۲
۳۳	۳۳	۲۲	۴۴	۲۲	۴۴	۳۳	۳۳	۲۲	۲۲
۳۳	۳۳	۳۳	۳۳	۳۳	۳۳	۲۲	۳۳	۲۲	۲۲
۳۳	۳۳	۳۳	۴۴	۳۳	۲۲	۴۴	۲۲	۴۴	۴۴
۳۳	۳۳	۳۳	۴۴	۳۳	۲۲	۴۴	۲۲	۴۴	۴۴

$$A \oplus B = \left\{ z \middle| \left(\hat{B}\right)_z \cap A \neq \emptyset \right\}$$

عملیات dilation به این صورت است:

در نتیجه عنصر ساختاری باید ابتدا ۱۸۰ درجه تغییر کند و سپس بر روی تصویر با اندازه بردار Z جا به جا شود.

عملیات erosion به این صورت است و نیازی به تغییر عنصر ساختاری نداریم:

$$A \ominus B = \{z|(B)_z \subseteq A\}$$

عنصر ساختاری بعد از تغییر ۱۸۰ درجه:

1	0	0
1	0	0
1	1	1

در این سوال anchor را عنصر وسطی میگیریم.

این سوال به دلیل اینکه از مقادیر مختلف و غیر باینری استفاده شده است و از سه رنگ است و نمیتوانیم در کل باینری بگیریم با روش ماکسیمم گیری عمل میکنیم.

به این صورت که در یک پنجره ۳*۳ از تصویر که جدا میکنیم در قسمت هایی که نظیر به نظیر در عنصر ساختاری برابر ۱ میشود مقدار ماکسیمم را در نظر میگیریم.

۲۲	۲۲	۲۲
۲۲	۲۲	۲۲
۳۳	۲۲	۲۲

این مقادیر را در نظر میگیریم و ماکسیمم بین اینها ۳۳ میشود در نتیجه به جای ۲۲ مقدار ۳۳ را در نظر میگیریم.

جدول کلی به این صورت میشود:

۳۳	۳۳	۳۳	۳۳	۳۳	۳۳	۳۳	۳۳
۴۴	۴۴	۳۳	۳۳	۳۳	۴۴	۳۳	۳۳
۲۲	۳۳	۴۴	۴۴	۴۴	۴۴	۳۳	۳۳
۳۳	۴۴	۴۴	۴۴	۴۴	۴۴	۴۴	۲۲

۳۳	۴۴	۴۴	۴۴	۴۴	۴۴	۳۳	۳۳
۳۳	۴۴	۳۳	۴۴	۴۴	۳۳	۳۳	۳۳
۳۳	۴۴	۴۴	۴۴	۴۴	۴۴	۴۴	۴۴
۳۳	۴۴	۴۴	۴۴	۴۴	۴۴	۴۴	۴۴

حال به سراغ عملیات erosion میرویم.

به همان صورت که در گسترش داشتیم عمل میکنیم اما این بار مینیمم میگیریم.

جدول آن به این صورت میشود:

۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲
۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲
۲۲	۲۲	۲۲	۲۲	۳۳	۲۲	۲۲	۲۲
۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲
۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲
۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲
۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲	۲۲
۳۳	۳۳	۳۳	۳۳	۲۲	۲۲	۲۲	۲۲

۶)

برای قسمت اول یعنی سمت چپ معادله به این صورت داریم:

در این سوال نیز از reflect padding برای محاسبه استفاده کردم.

$(A \ominus B_1) \wedge (A \ominus B_2) = ?$ (۶)

ابتدا این قسمت را حل می‌کنیم

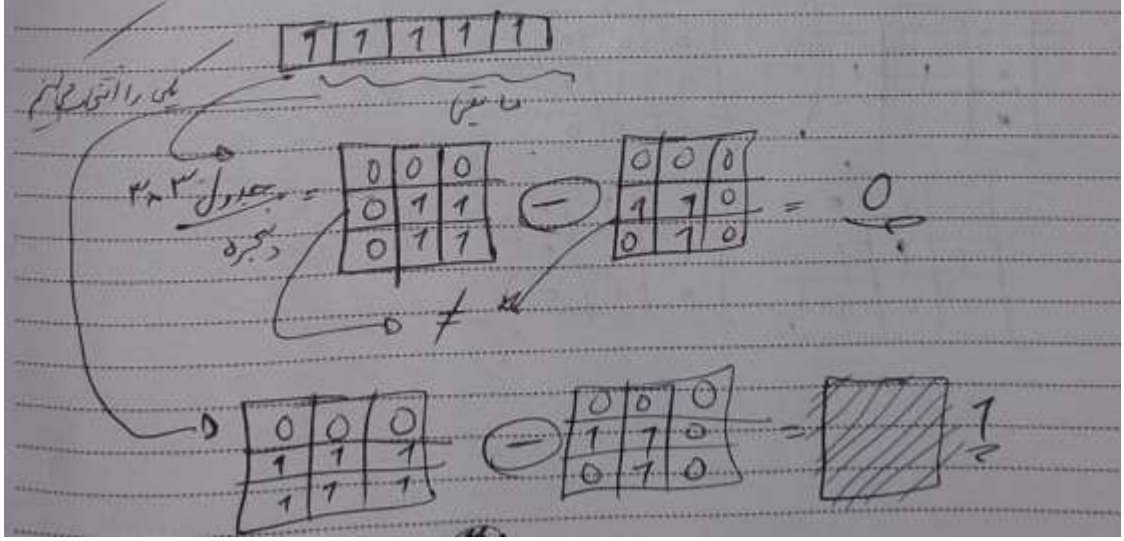
$A \ominus B_1$

سطر اول و آخر کاملا صفر هستند و عنصر برزی $B_1 = 1$

⇐ خروجی در سطر اول و آخر کاملا صفر

⇐ تمام جاهایی از ستون که $= 0$ هستند در خروجی نیز صفر هستند

سطر ۲ تا ۴، سطر ۶ تا ۱۱ مقدار مشابه دارند که بررسی می‌کنیم (منظور از ستون شماره ۵ تا ۹ است) یعنی جاهایی که ۱ داریم را بررسی می‌کنیم.



پس در این قسمت

1	1	1	1	1

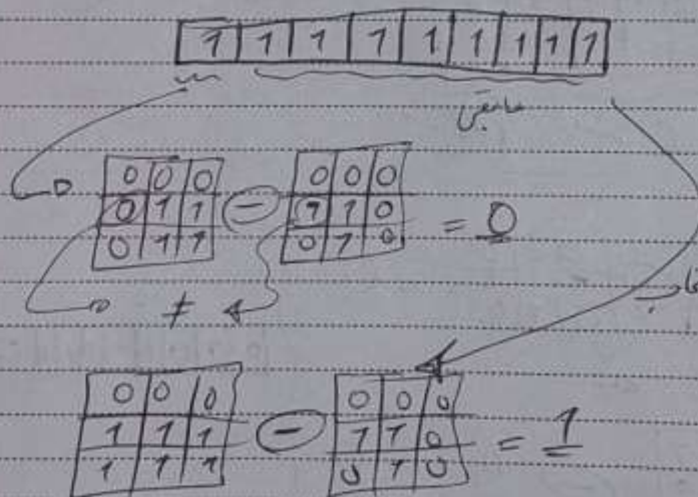
که در سطرهای ۲ تا ۴ و ۹ تا ۱۱ تکرار شده است

خروجی به این صورت می‌شود

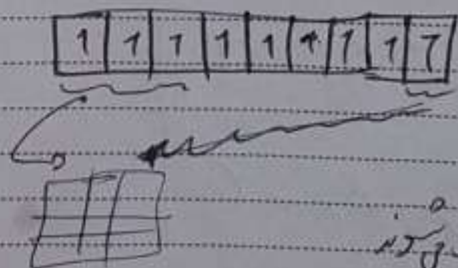
0	1	1	1	1

در این سطرها فقط این ۴ یک را داریم و بقیه جاها صفر می‌شود

حال برای سطرهای ۵ تا ۸ یک سطر را بررسی می‌کنیم و بقیه مشابه می‌شوند

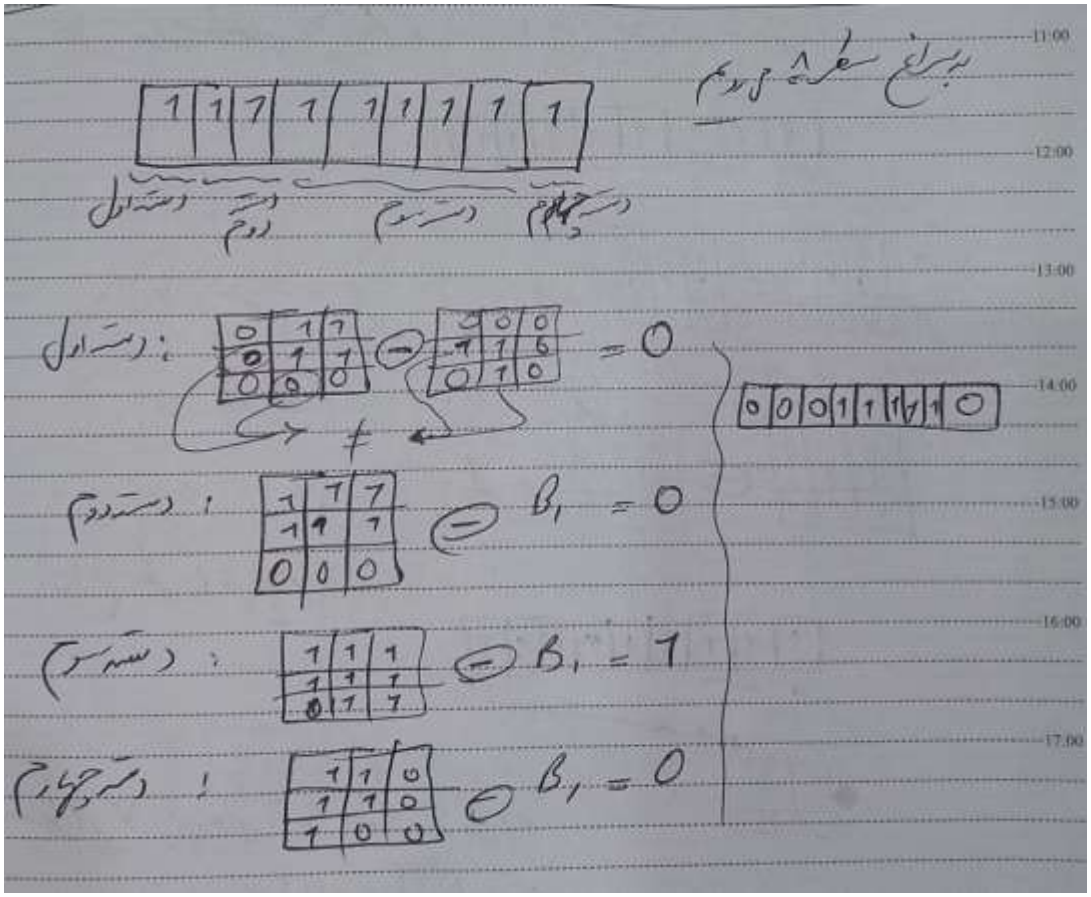


برای سطر ۸ هم داریم،



سطر ۶ و ۷ در نمونه‌های ۳ و ۴ متفاوت می‌شوند

اما نتیجه با سطر ۵ یکسان است



جدول نهایی بخش اول به این صورت میشود:

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	0	0
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

جدول مکمل A به این صورت میشود + reflect padding مربوط به آن:

۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۱	۱	۱	۱
۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۱	۱	۱	۱
۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۱	۱	۱	۱
۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱
۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱
۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱
۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱
۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۱	۱	۱	۱
۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۱	۱	۱	۱
۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱

نتیجه سایش آن با عنصر ساختاری به صورت جدول زیر میشود:

۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۰	۰	۰	۰	۰	۱	۱	۱	۱
۱	۱	۱	۰	۰	۰	۰	۰	۰	۱	۱	۱
۱	۱	۱	۰	۰	۰	۰	۰	۰	۱	۱	۱
۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱
۱	۱	۱	۰	۰	۰	۰	۰	۰	۰	۱	۱
۱	۱	۱	۰	۰	۰	۰	۰	۰	۱	۱	۱
۱	۱	۱	۰	۰	۰	۰	۰	۰	۱	۱	۱
۱	۱	۱	۰	۰	۰	۰	۰	۰	۱	۱	۱

در نهایت اشتراک این دو به این صورت میشود که در سطر دوم و ستون ۹ + سطر ۵ ستون ۱۰ مقادیر آن برابر ۱ میشود و بقیه مقادیر برابر 0 میشوند.

۷)

الف) ابتدا تصویر را میخوانیم به grayscale میبریم و سپس آنرا باینری میکنیم.سپس یک عنصر ساختاری مشخص میکنیم و عملیات opening را انجام میدهیم تا جزئیات تصویر حذف بشوند سپس یکبار دیگر بر روی تصویری که جزئیات آنرا حذف کردیم عملیات closing را اعمال میکنیم تا اجزای مختلف را جدا کنیم.

در قسمت بعد ، از خروجی به دست آمده بعد از closing شروع به پیدا کردن نقاط گوشه میکنیم و تعداد آنها را نیز بدست می آوریم.سپس شرطی میگذاریم که هر آبجکت چه محدوده و مساحتی باید داشته باشد تا به عنوان ماشین شناخته شود.

و در نهایت تعداد ماشین ها را نمایش میدهیم.

ب)

در این قسمت نیز تقریبا عملیات های قسمت قبل را تکرار میکنیم.با این تفاوت که چون تصویر باینری نیست تصویر را ماسک میکنیم.همانطور که اشاره شده از نقطه مرکزی و وسط هر گل استفاده میکنیم تا گل ها را شناسایی کنیم.سپس بر اساس ماسک ایجاد شده عملیات های opening,closing را انجام میدهیم و در نهایت تصویر به دست آمده از closing را برای پیدا کردن نقاط گوشه استفاده میکنیم و سپس شروطی میگذاریم تا به خوبی بتوانیم وسط هر گل را بیابیم.و در نهایت نقاط به دست آمده را رسم میکنیم.

۸)

برای بدست آوردن اسکلت تمام مراحلی که در جزوه گفته شده را مرحله به مرحله طی میکنیم.

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

$$A \ominus kB = \big((A \ominus B) \ominus B\big) \ominus \cdots\big)$$

$$K = max\{k | (A \ominus kB) \neq \emptyset\}$$

$$A = \bigcup_{k=0}^{K} S_k(A) \oplus kB$$

ابتدا تصویر را باینری میکنیم سپس دو عملیات فرسایش و گسترش را انجام میدهیم تا عملیات opening را انجام داده باشیم.سپس تصویر اصلی را از عملگر باز کم میکنیم تا Sk بدست آید.تعداد k نیز نشان دهنده تعداد دفعات تکرار این عملیات است.این مقادیر در params نگهداری میشوند تا بتوانیم از اسکلت تصویر اصلی را نیز بسازیم. در مرحله بعد سه عکس را خواندیم و اسکلت مربوط به هر کدام را دریافت کردیم و اسکلت مربوط به هر کدام را بدست آوردیم و در دایرکتوری images ذخیره کردیم و در نهایت نمایش دادیم.

برای بازسازی عکس طبق رابطه زیر عمل کردیم:

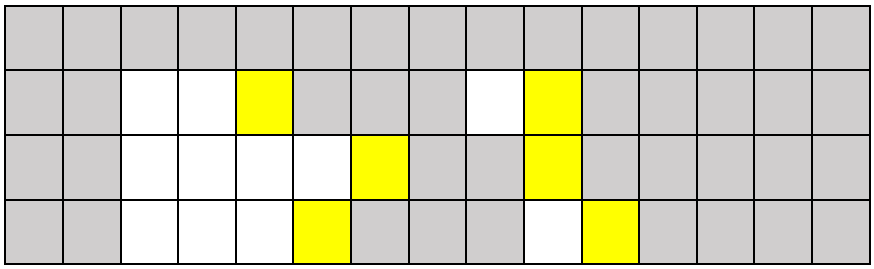
$$A = \bigcup_{k=0}^{K} S_k(A) \oplus kB$$

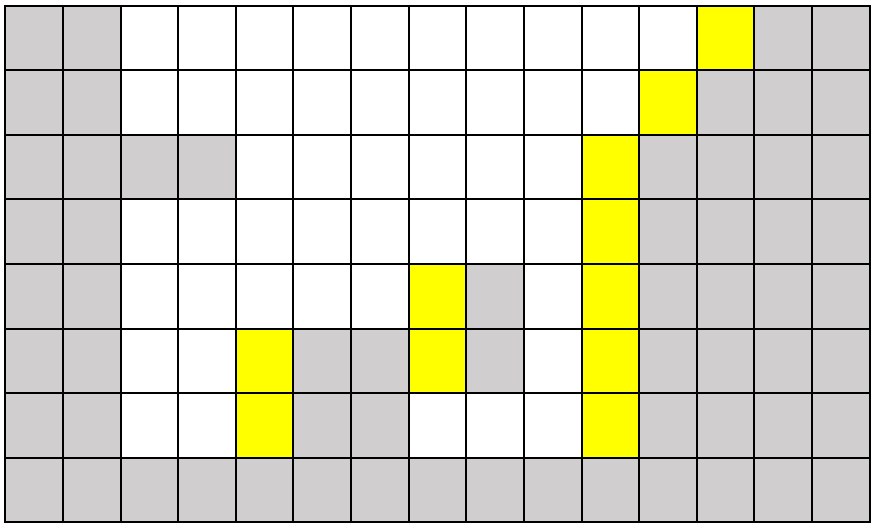
همانطور که گفتیم بر اساس params که در مرحله قبل ذخیره کردیم حال برای بازسازی تصویر استفاده میکنیم و عملیات گسترش را انجام میدهیم تا به نتایج قبل برسیم.

۹)

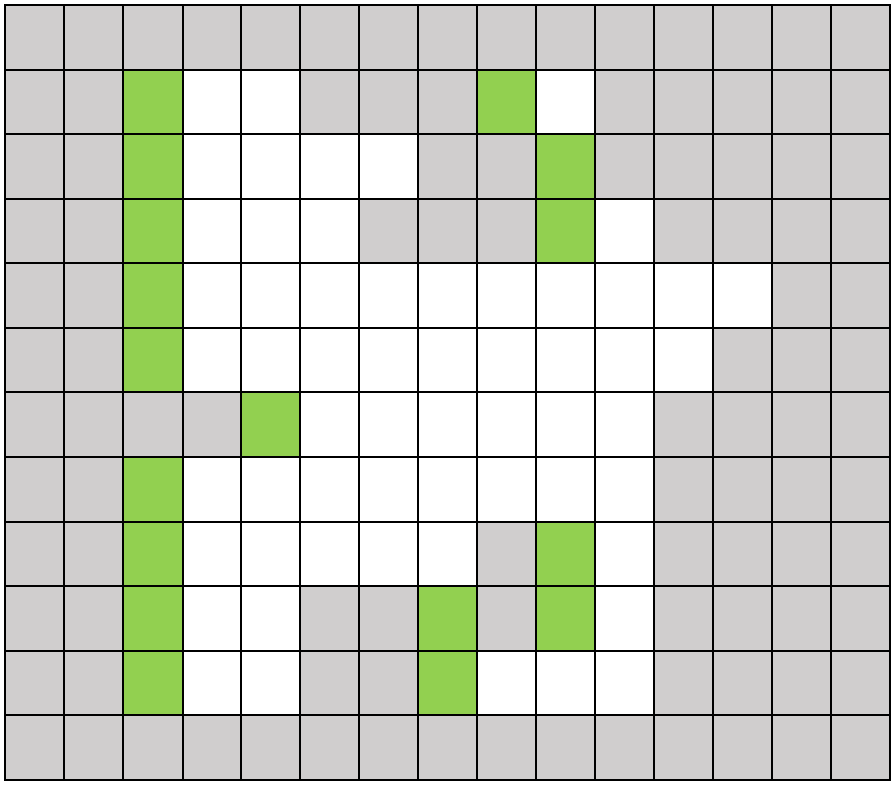
باید عناصر ساختاری در جهات راست چپ بالا پایین را بدست آورده و درنهایت تصویر حاصل هر یک از آنها را با هم اجتماع بگیریم.

مرز سمت راست :

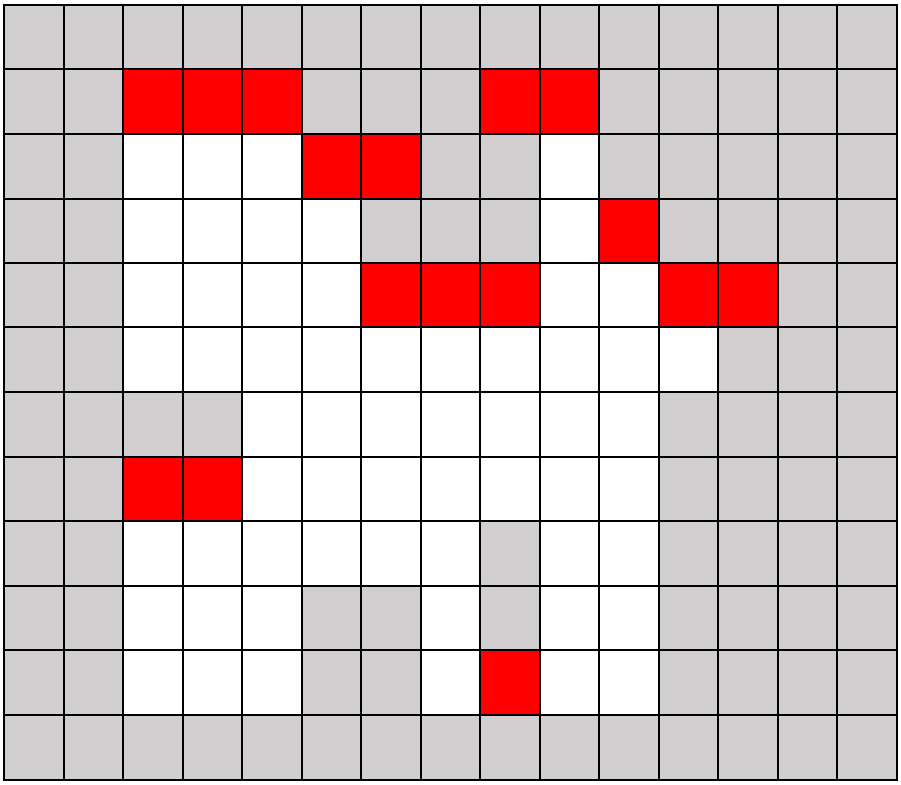




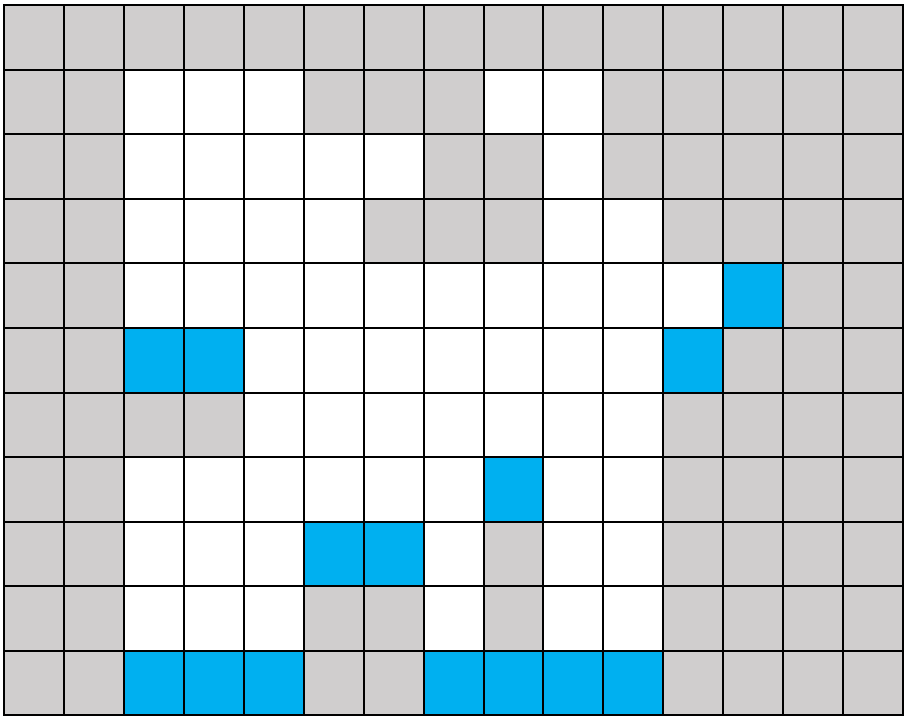
مرز چپ:



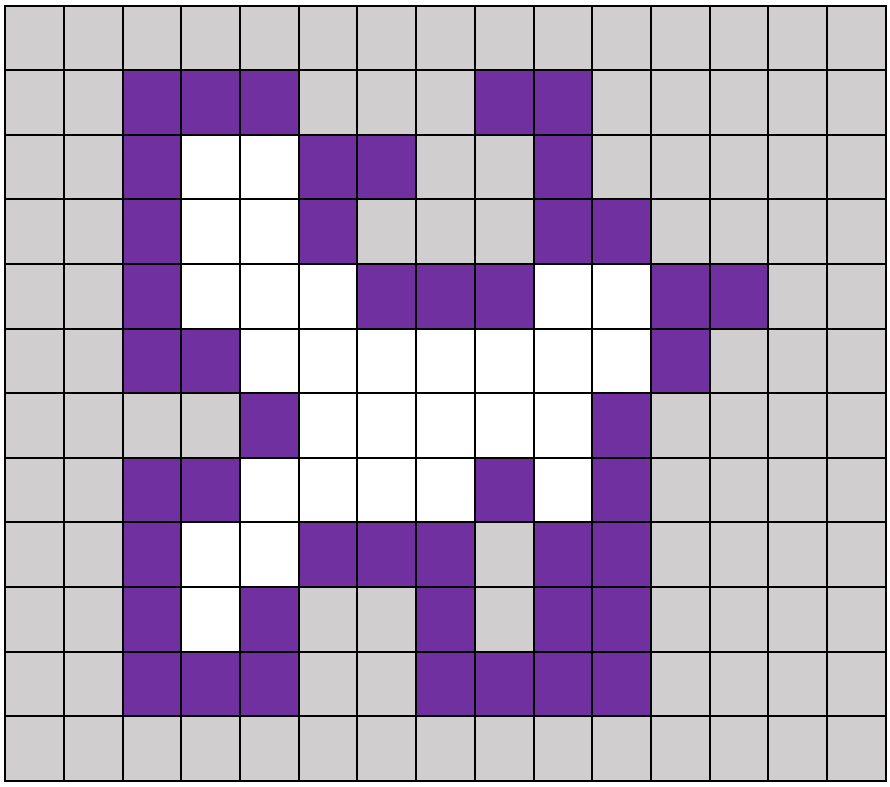
مرز بالا:



مرز پایین:



تصویر نهایی :



عناصر ساختاری نیز به این صورت هستند:

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