به نام خدا

مبانی بینایی کامپیوتر

دکتر محمدی

تمرین چهار

علی عطاریان – ۹۹۵۲۱۴۵۱

سوال ۱) الف)

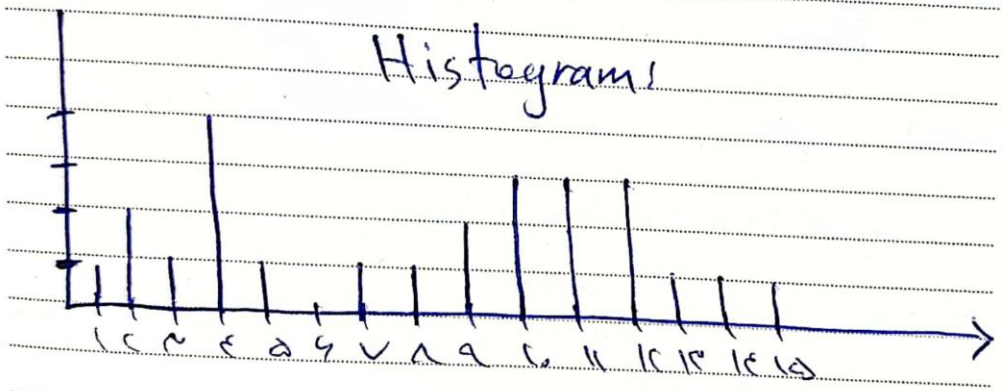
۱) الف)

شنبه
آذر
۱۳ جمادی الاول ۱۴۴۳
18 Dec 2021
۲۷

عدد	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
تعداد	۱	۲	۱	۴	۱	۵	۱	۱	۲	۳	۳	۳

عدد	۱۳	۱۴	۱۵
تعداد	۱	۱	۱

Histogram:



$$\text{Avg: } \frac{1\times1+2\times2+3\times1+4\times4+5\times1+\cdots+15\times1}{25} = 8,12$$

Median = 9

Mode = 4

$$\text{Variance: } \frac{(1-8,12)^2+2(2-8,12)^2+(3-8,12)^2+4(4-8,12)^2+\cdots+(15-8,12)^2}{25}$$

$$= 14,7425$$

Scanned with CamScanner

سوال ۱) ب)

۱) ب) سطح آستانه ۹,۵، $w_1 = 13$

$\sigma_1 = 6,43$

$w_2 = 12 \Rightarrow \sigma_w^2 \approx 114,55$

$\sigma_2 = 2,35$

سطح آستانه ۱۱,۵، $w_1 = 19$

$\sigma_1 = 11,71$

$w_2 = 6 \Rightarrow \sigma_w^2 \approx 230,43$

$\sigma_2 = 1,33$

پس سطح آستانه ۹,۵ بهتر است

Scanned with CamScanner

سوال ۲) الف)

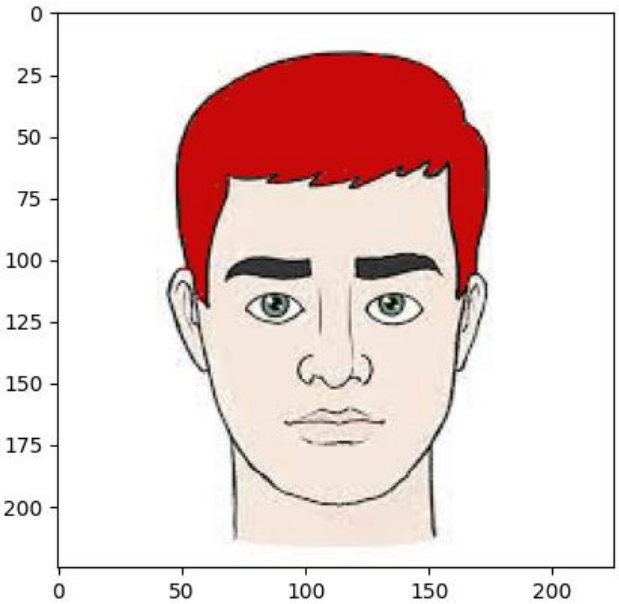
مشاهده می‌کنیم در تصاویری که هیستوگرام نسبتا متعادلی دارند (شبیه به توزیع نرمال) هر دو الگوریتم با دقت یکسان عمل می‌کنند اما در تصاویری که در هیستوگرام آنها یک یا دو پیک بزرگ وجود دارد دقت الگوریتم G.otsu بیشتر است. همانطور که در متن اشاره شده است سرعت G.otsu بسیار بیشتر از الگوریتم اتسوی بهینه است. دلیل آن این است که الگوریتم اتسو به ازای تک تک ۲۵۵ حالت حلقه می‌زند اما در G.otsu در هر سطح بدون محاسبه از اول و فقط با محاسبه یک تغییر قابل انجام است.

سوال ۲) ب)

خیر، کمینه‌کردن واریانس درون‌کلاسی یعنی می‌خواهیم پیکسل‌های درون هرکلاس بیشترین شباهت را به یکدیگر داشته باشند و بیشینه کردن واریانس بین‌کلاسی یعنی می‌خواهیم دو کلاس مدنظر بیشترین تفاوت را با هم داشته باشند. حال ممکن است دو کلاس واریانس درون‌کلاسی‌شان بسیار کم باشد و به هم شبیه باشند اما در عین حال واریانس بین‌کلاسی‌شان بیشینه نباشد زیرا این دو کلاس به یکدیگر نیز شبیه هستند. در الگوریتم اتسو هدف کمینه کردن واریانس درون‌کلاسی است، نه بیشینه کردن واریانس بین‌کلاسی.

سوال ۳)

یک تابع کمکی برای تشخیص در ناحیه بودن یا نبودن پیکسل موردبررسی به نام is_same_seg می‌نویسیم که بر اساس آستانه زدن روی جمع تفاوت سه کانال رنگی با پیکسل seed عمل می‌کند. حال در تابع اصلی یک dfs پیاده‌سازی می‌کنیم و هرکدام از پیکسل های در صف را با تابع is_same_seg می‌سنجیم. برای حد آستانه با آزمون و خطا به عدد ۶۵ می‌رسیم.



سوال۴) الف)

تصویر با reflect padding:

60	60	70	60	60	70	60	60	60	60
60	60	70	60	60	70	60	60	60	60
60	60	70	70	70	70	70	70	60	60
60	60	70	60	70	70	70	70	70	70
80	80	60	80	60	70	80	70	70	70
60	60	70	70	60	70	60	60	60	60
60	60	70	80	60	80	70	60	60	60
70	70	60	80	60	60	80	60	60	60
60	60	70	70	80	60	80	60	70	70
60	60	70	70	80	60	80	60	70	70

برای سایش لنگر را روی پیکسل موردنظر قرارداده و مینیمم مقدارهایی که با ۱ متناظرند را به‌جای پیکسل قرار می‌دهیم:

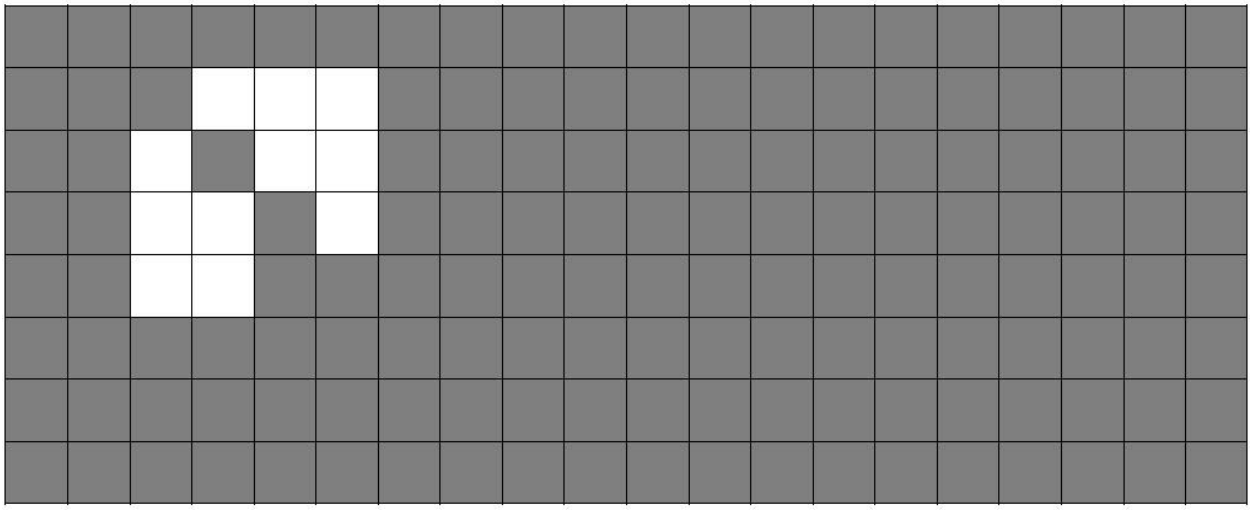
60	60	60	60	60	60	60	60
60	60	60	60	60	60	60	60
60	60	70	60	70	70	60	60
60	60	60	60	60	70	70	70
60	60	60	60	60	70	60	60
60	60	60	60	60	60	60	60
60	60	60	60	60	60	60	60
60	60	60	60	60	60	60	60

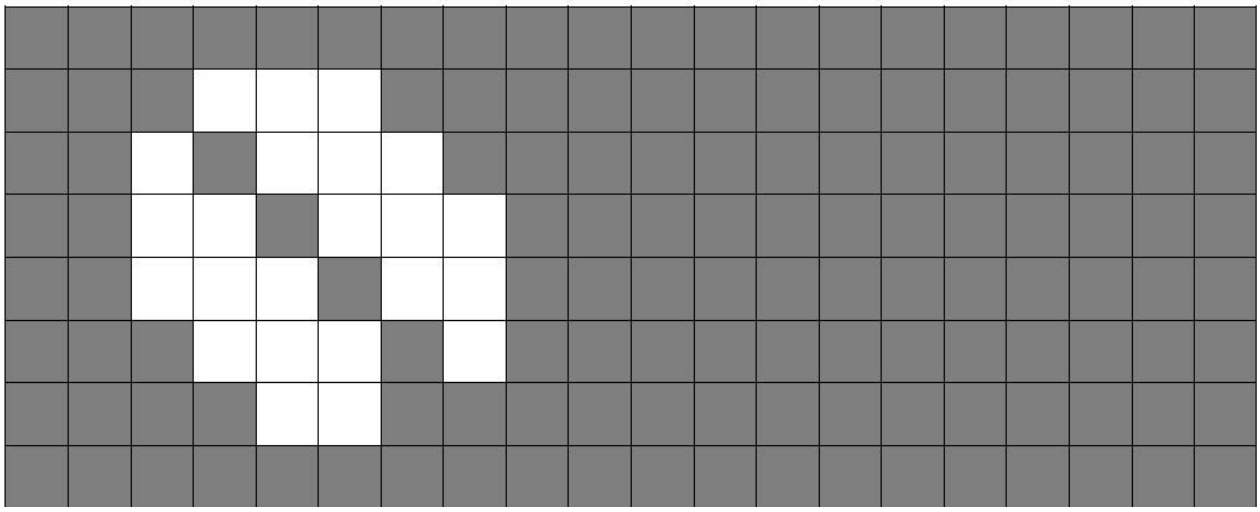
برای گسترش ابتدا عنصر ساختاری را نسبت به لنگر ۱۸۰ درجه می‌چرخانیم سپس ماکسیمم مقدارهایی که با ۱ متناظرند را به‌جای پیکسل موردنظر قرار می‌دهیم:

70	70	70	70	70	70	70	70
70	70	70	70	70	70	70	70
80	80	80	80	80	80	80	70
70	80	70	70	80	70	70	70
70	80	80	80	80	80	80	60
70	80	80	80	80	80	80	60
70	80	80	80	80	80	80	70
70	70	80	80	80	80	80	70

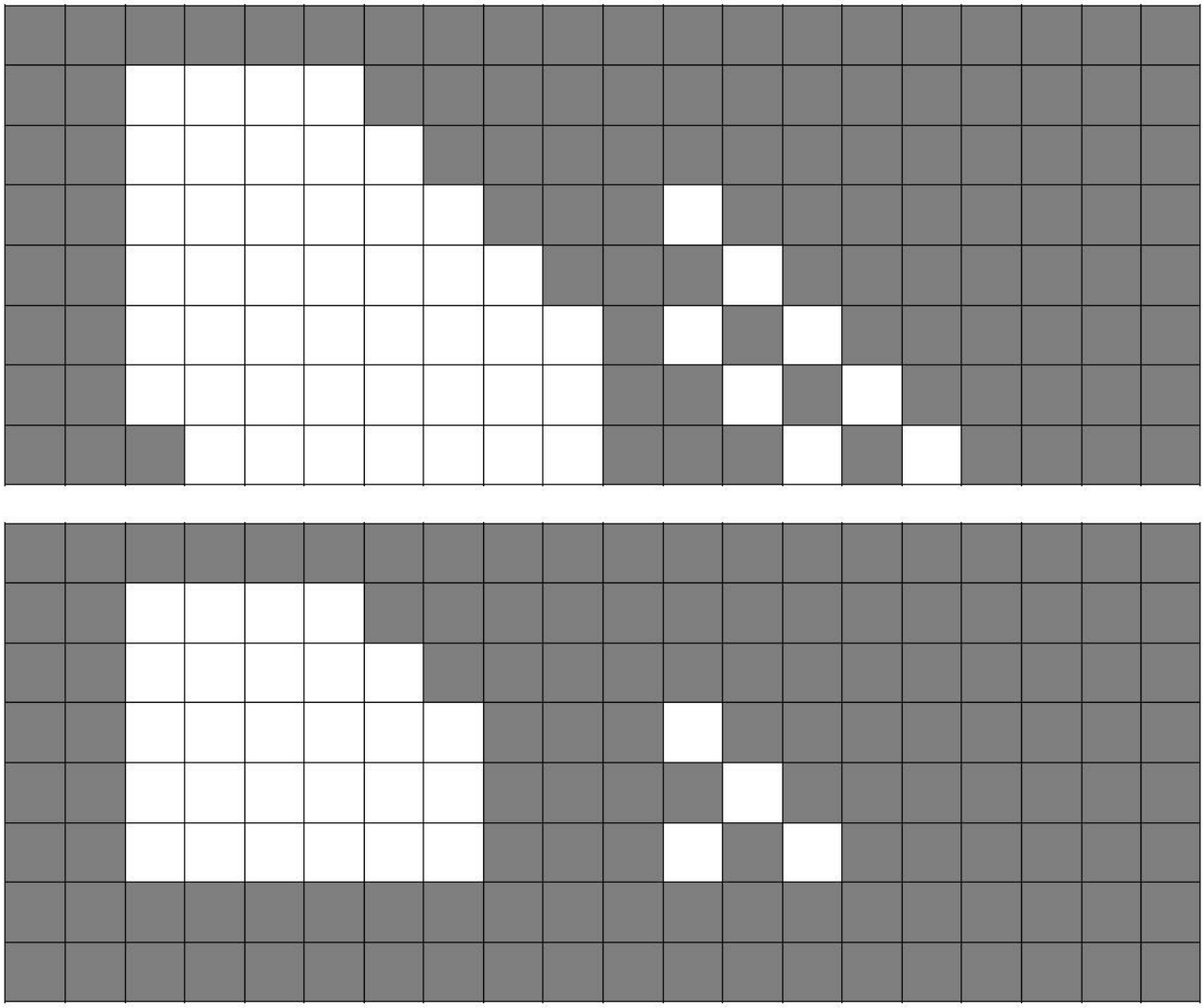
سوال ۴) ب)

برای عملگر باز ابتدا سایش سپس گسترش انجام می‌دهیم (برای ستون و ردیف های اول و آخر تصویر constant padding=grat در نظر گرفتیم:



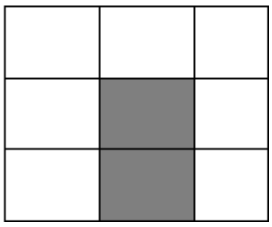


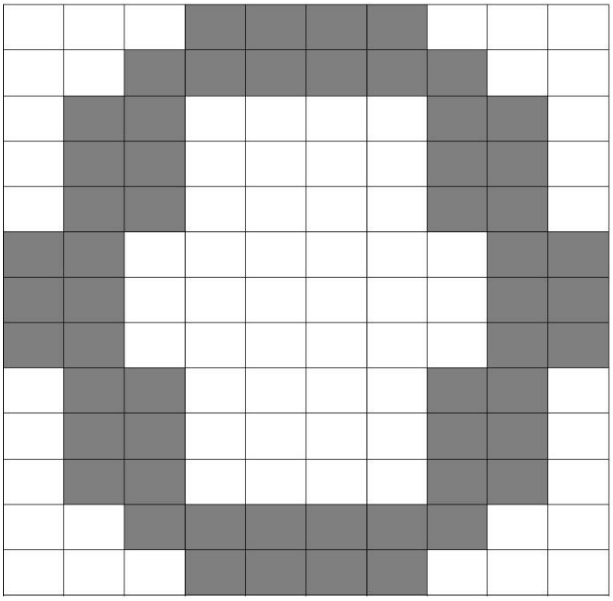
برای عملگر بسته ابتدا گسترش سپس سایش انجام می‌دهیم(باز هم constant padding=gray):



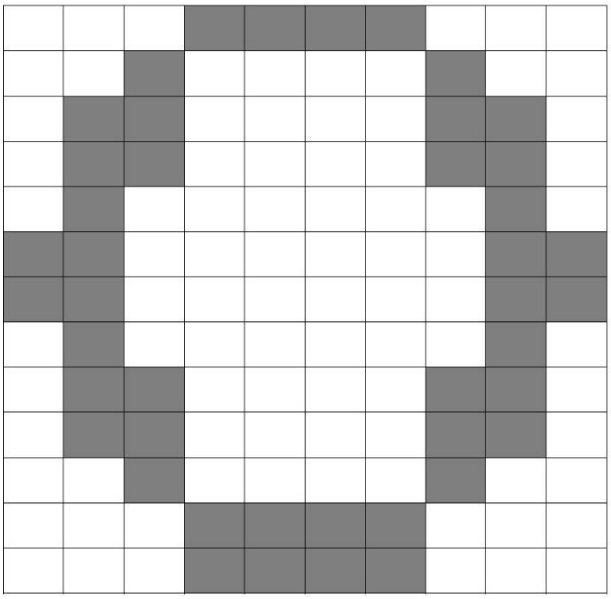
سوال ۵) الف)

با توجه به اینکه عملگرهای گسترش و افزایش ساختار خود صفر را تحت تاثیر قرار می‌دهند و می‌دانیم عملگر بسته معمولا برای از بین بردن حفره‌های کوچک استفاده می‌شود اما ما اتفاقا می‌خواهیم حفره درون صفر به درستی دیده شود پس از عملگر باز استفاده می‌کنیم. با کمی آزمون و خطا متوجه می‌شویم عنصر ساختاری روبرو بهترین گزینه است. رنگ خاکستری را پیکسل با ارزش ۱ در نظر گرفتیم و رنگ سفید را پیکسل با ارزش صفر.





پس از سایش و گسترش



پس از سایش

سوال ۵) ب)

مرز راست

0	0	0
0	1_+	1_-
0	0	0

مرز پایین

0	0	0
0	1_+	0
0	1_-	0

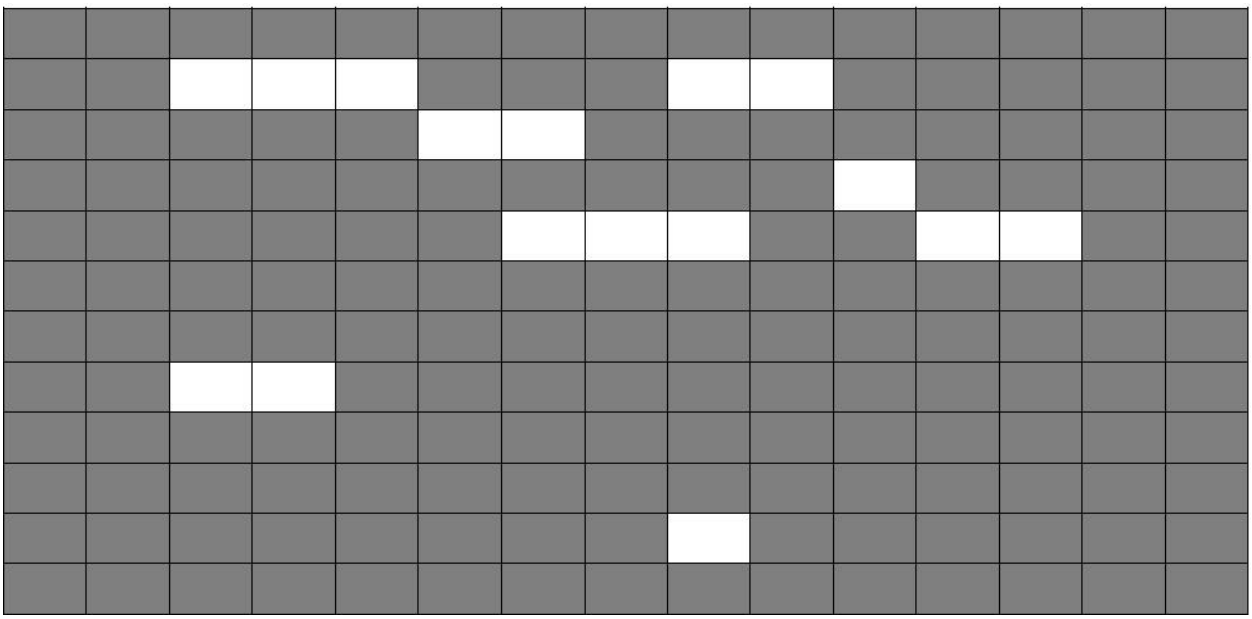
مرز چپ

0	0	0
1_-	1_+	0
0	0	0

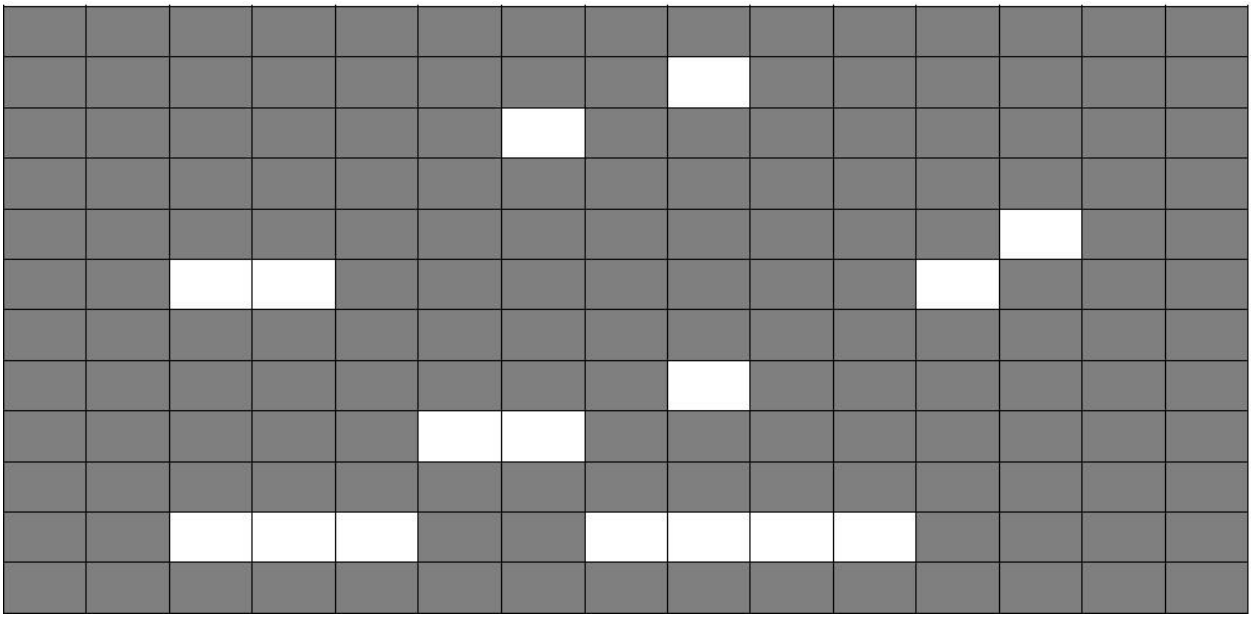
مرز بالا

0	1_-	0
0	1_+	0
0	0	0

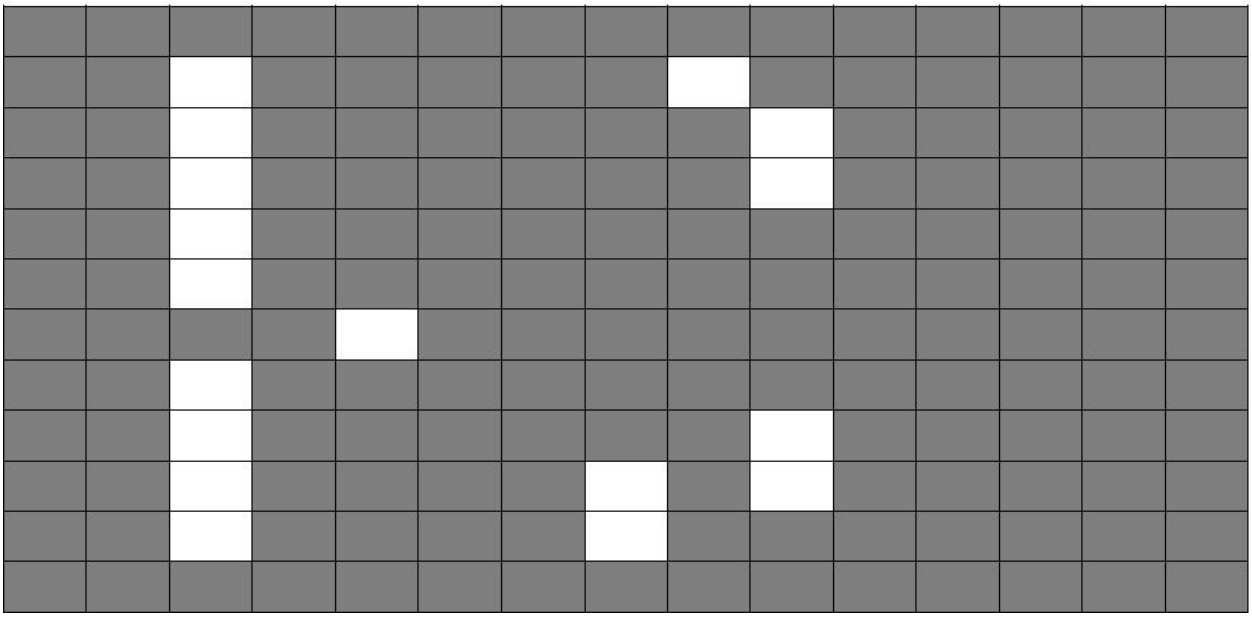
یافتن مرز بالا:



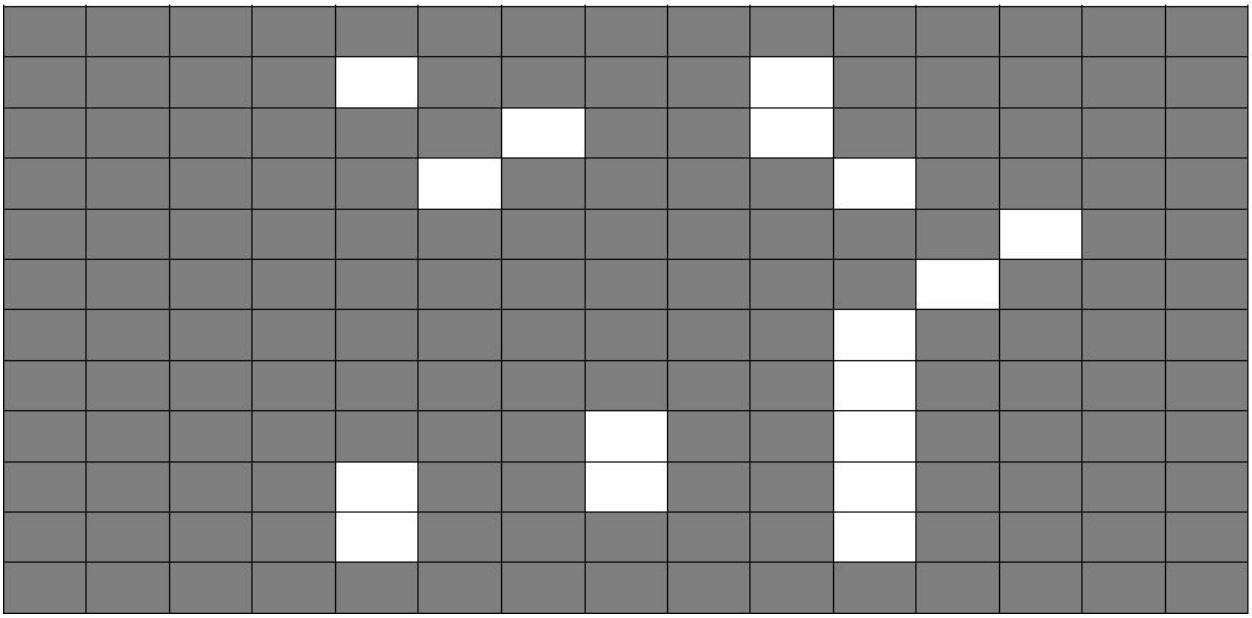
یافتن مرز پایین:



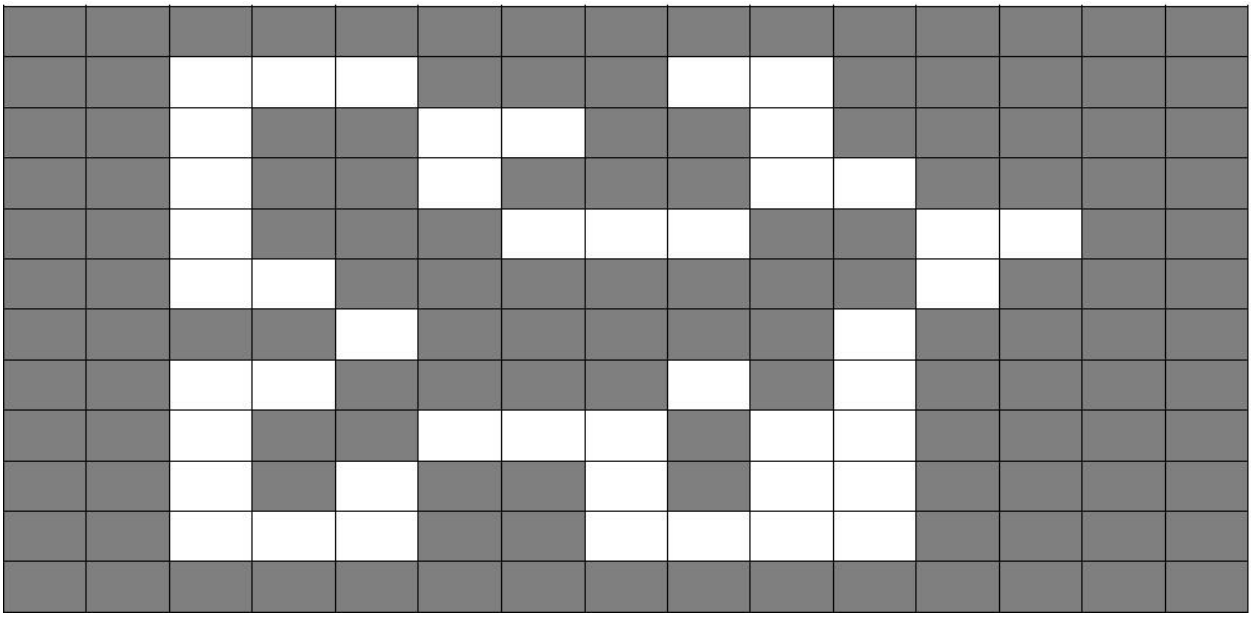
یافتن مرز چپ:



یافتن مرز راست:



اجتماع ۴ مورد بالا:



سوال ۶)

$$\texttt{dst}(x,y) = \max_{(x',y'):\, \texttt{element}(x',y') \neq 0} \texttt{src}(x+x', y+y')$$

برای گسترش باید از فرمول روبرو استفاده کنیم

بنابراین پنجره را در کرنل ضرب می‌کنیم و ماکسیمم را برمی‌گردانیم تا همسایگی‌هایی که متناظر با صفر در کرنل هستند در ماکسیمم گیری اثر نداشته باشند.

برای سایش بایستی مینیمم بگیریم اما نمی‌توان از روش قبل استفاده کرد زیرا همیشه مینیمم صفر می‌شود بنابراین باید حلقه فور بزنیم و اگر همسایگی متناظر با صفر در کرنل نبود، در مینیمم‌گیری شرکت کند.

برای عملگر باز ابتدا سایش سپس گسترش و برای عملگر بسته ابتدا گسترش و سپس سایش انجام می‌شود.

توابع آماده نیز به ترتیب cv2.dilate, cv2.erode,

cv2.morphologyEx(src, cv2.MORPH_OPEN, kernel),
cv2.morphologyEx(src, cv2.MORPH_CLOSE, kernel)

می‌باشند.

البته باید دقت داشت که چون پس زمینه سفید است و شیء در تصویر سیاه، پس برای مثال عملگر گسترش باعث کوچک شدن شیء می‌شود زیرا در حال گسترش ناحیه سفید است. این موضوع درباره باقی عملگر ها نیز صدق می‌کند.

نتایج:

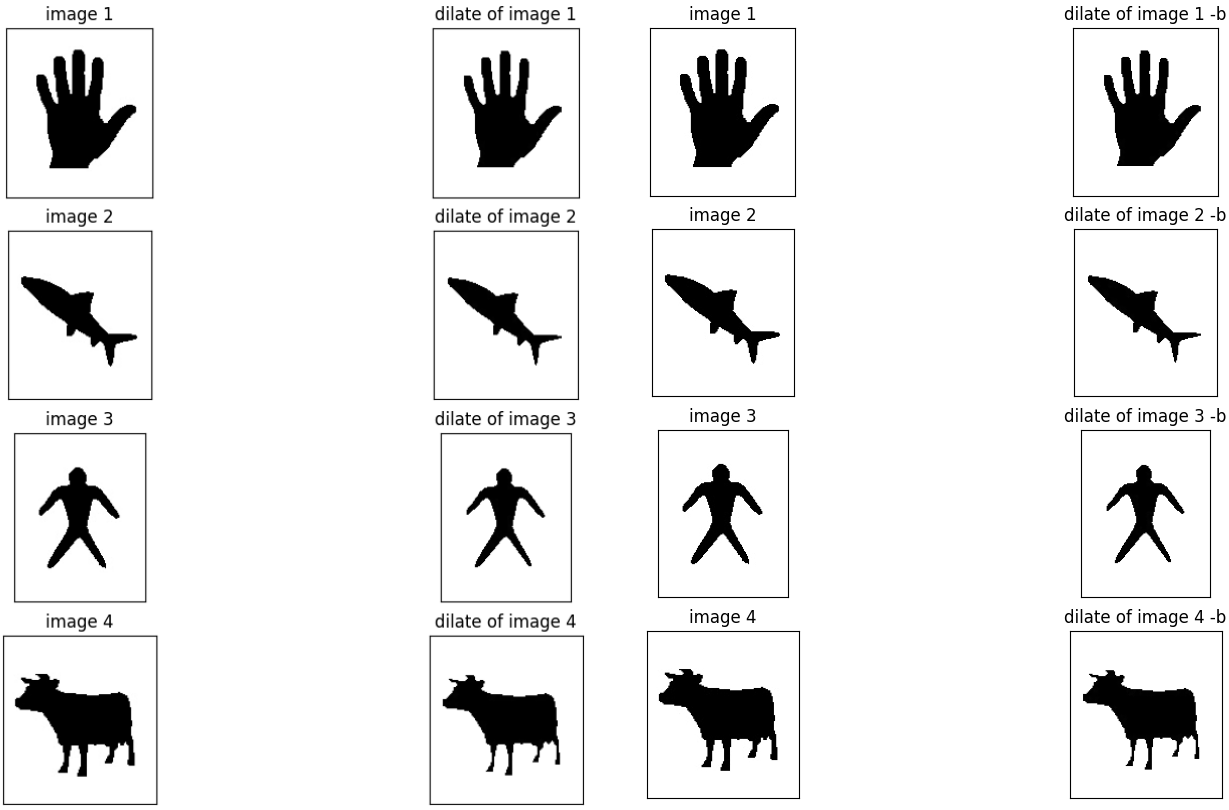


image 1
dilate of image 1
image 1
dilate of image 1 -b
image 2
dilate of image 2
image 2
dilate of image 2 -b
image 3
dilate of image 3
image 3
dilate of image 3 -b
image 4
dilate of image 4
image 4
dilate of image 4 -b

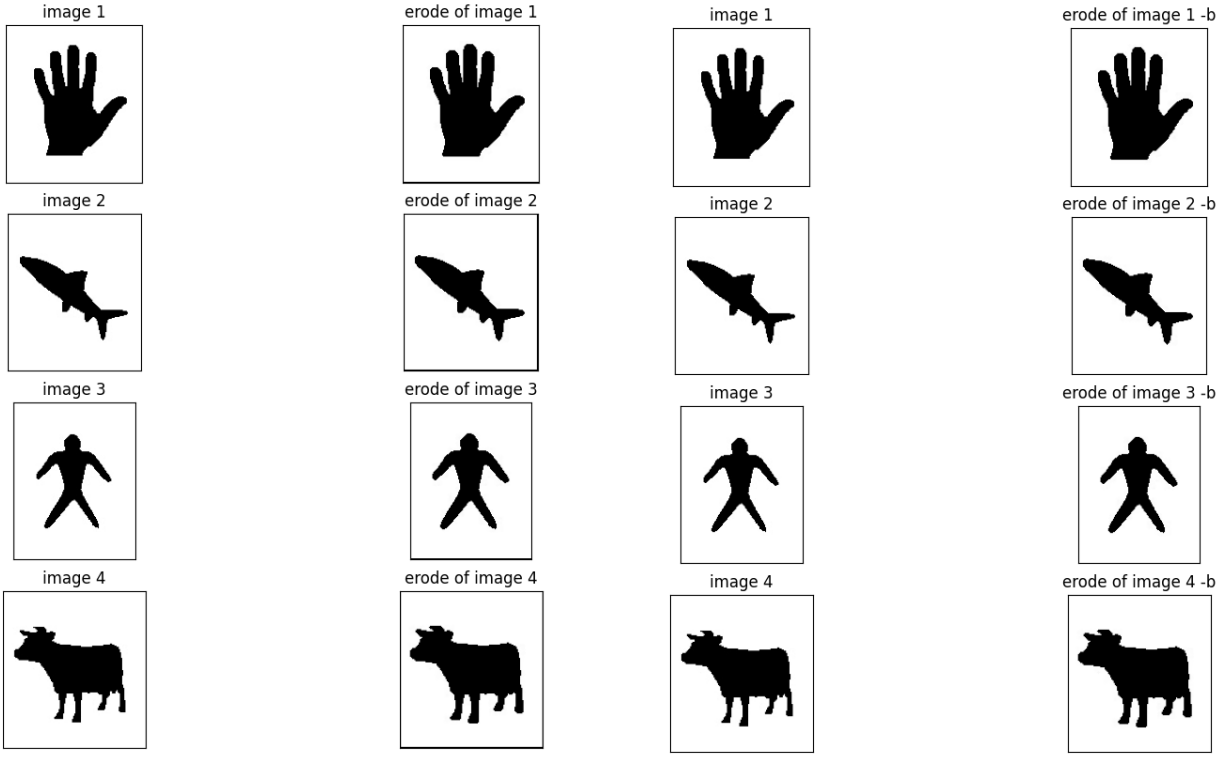


image 1
erode of image 1
image 1
erode of image 1 -b
image 2
erode of image 2
image 2
erode of image 2 -b
image 3
erode of image 3
image 3
erode of image 3 -b
image 4
erode of image 4
image 4
erode of image 4 -b

image 1

image 2

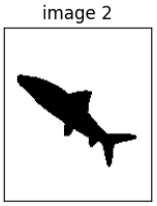


image 3

image 4

open of image 1

open of image 2

open of image 3

open of image 4

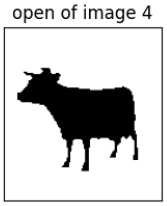
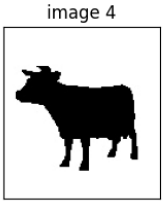


image 1

image 2

image 3

image 4



open of image 1 -b

open of image 2 -b

open of image 3 -b

open of image 4 -b

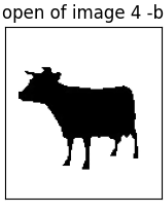


image 1

image 2

image 3

image 4

close of image 1

close of image 2

close of image 3

close of image 4

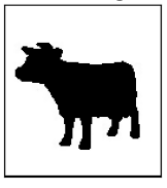
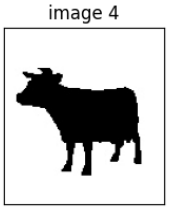


image 1

image 2

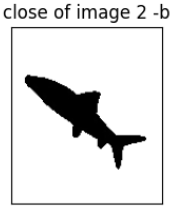
image 3

image 4



close of image 1 -b

close of image 2 -b



close of image 3 -b

close of image 4 -b