بسم الله الرحمن الرحیم

محمد عرفان زارع زردینی

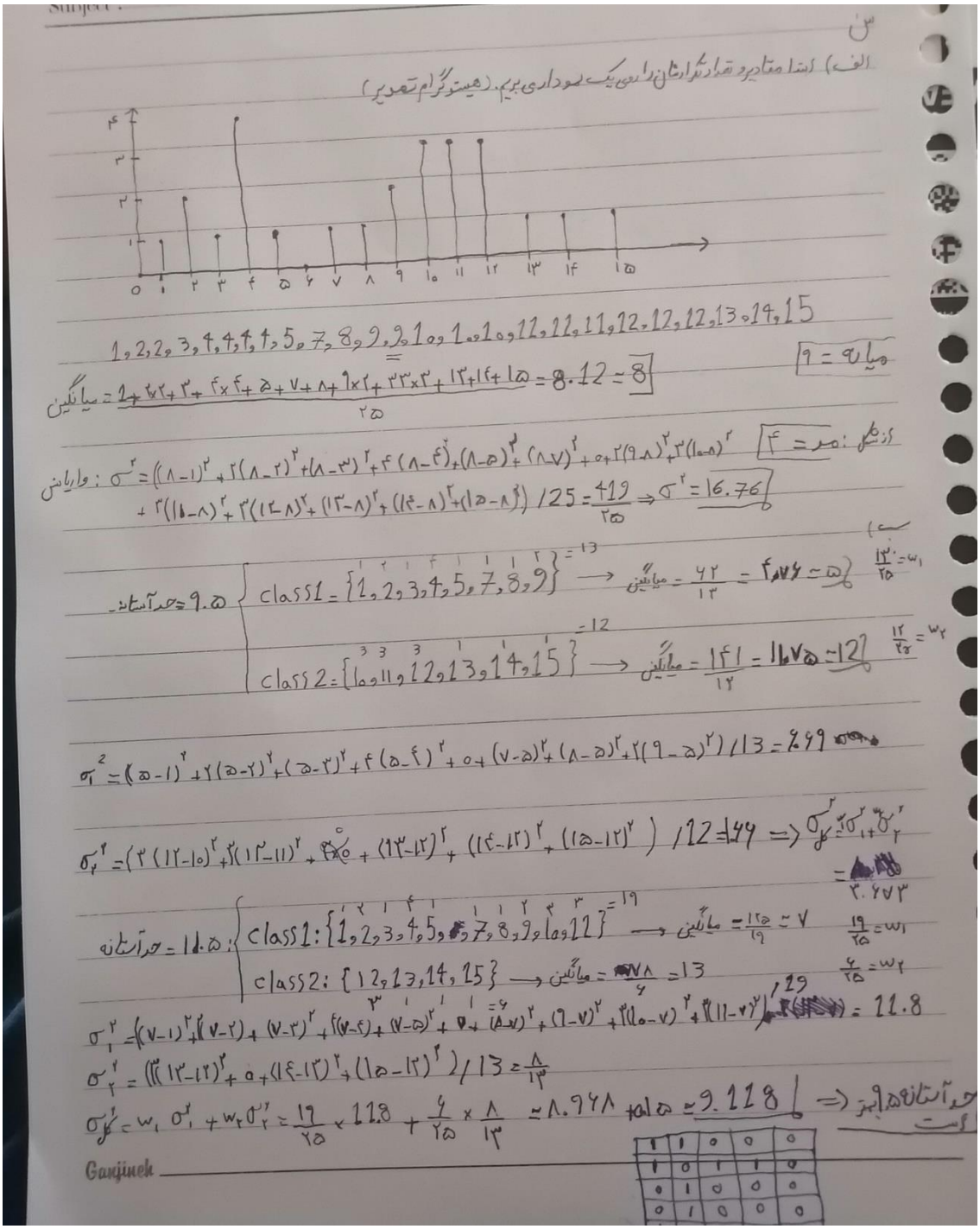
۹۸۴۱۱۱۴۳۲

تمرین سری چهارم درس بینایی ماشین

سوال 1:

Refrences:

Class slides



سوال 2:

Refrences:

(PDF) Improvement in the Between-Class Variance Based on Lognormal Distribution for Accurate Image Segmentation (researchgate.net)

الف)

با اینکه هردو روش برای تقسیم بندی تصاویر و بافت های مختلف تصویر استفاده می گردد ، از نظر سرعت روش otsu guassian به دلیل استفاده از پارامتر بیشتر، کندتر است .همچنین این روش برای تکرارپذیری نیاز به چندین بار اجرا دارد که سرعت آن را کاهش میدهد . روش اتسو نیاز به تکرارپذیری ندارد.از لحاظ دقت ولی روش otsu guassianبرای اصلاح نویز های تصویر ، دقت بیشتری در تقسیم بندی تصاویردارد .روش اتسو به دلیل عدم استفاده از فیلتر گاوسین ، به نتایج نامطلوب تری منتهی میشود. پس از لحاظ سرعت اتسو بهتر است و از لحاظ دقت otsu guassian دقیق تر است.در واقع طبق نتایج مشاهده شده می توان به این نتایج رسید .

ب)

بله، طبق روابط زیر که مربوط به واریانس بین کلاسی و درون کلاسی است میتوان به این نتیجه رسید. براساس این روابط ،کمینه کردن واریانس درون کلاسی ، سبب بیشینه شدن واریانس بین کلاسی می شود.

$$\sigma^2_{Between-Class} = P_o(\mu_o - \mu_T)^2 + P_b(\mu_b - \mu_T)^2$$

$$\sigma^2_{Within-Class} = P_o\sigma_o^2 + P_b\sigma_b^2$$

اگر بخواهیم توضیح بیشتری بدهیم و دلیل آن را بررسی کنیم، هدف از تقسیم بندی تصویر به دو کلاس، جداسازی تمامی نقاط به دو دسته مجزا و متمایز کردنشان است.پس وقتی واریانس داخلی کمینه شود ، نقاط هر کلاس در نزدیکی هم قرار میگیرند و تفاوت دو کلاس زیاد می شود. درواقع اگر واریانس داخلی کلاس ها کم شود،

داده های هر کلاس به شکل متمایز تری از هم جدا شده و در نتیجه واریانس بین کلاس ها بیشتر می شود .پس کاهش واریانس داخلی هر کلاس در این روش و الگوریتم باعث بیشینه شدن واریانس بین کلاس ها می شود .پس معادل با بیشینه شدن واریانس بین کلاس ها است.

سوال3:

Refrences:

https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html

Class slides

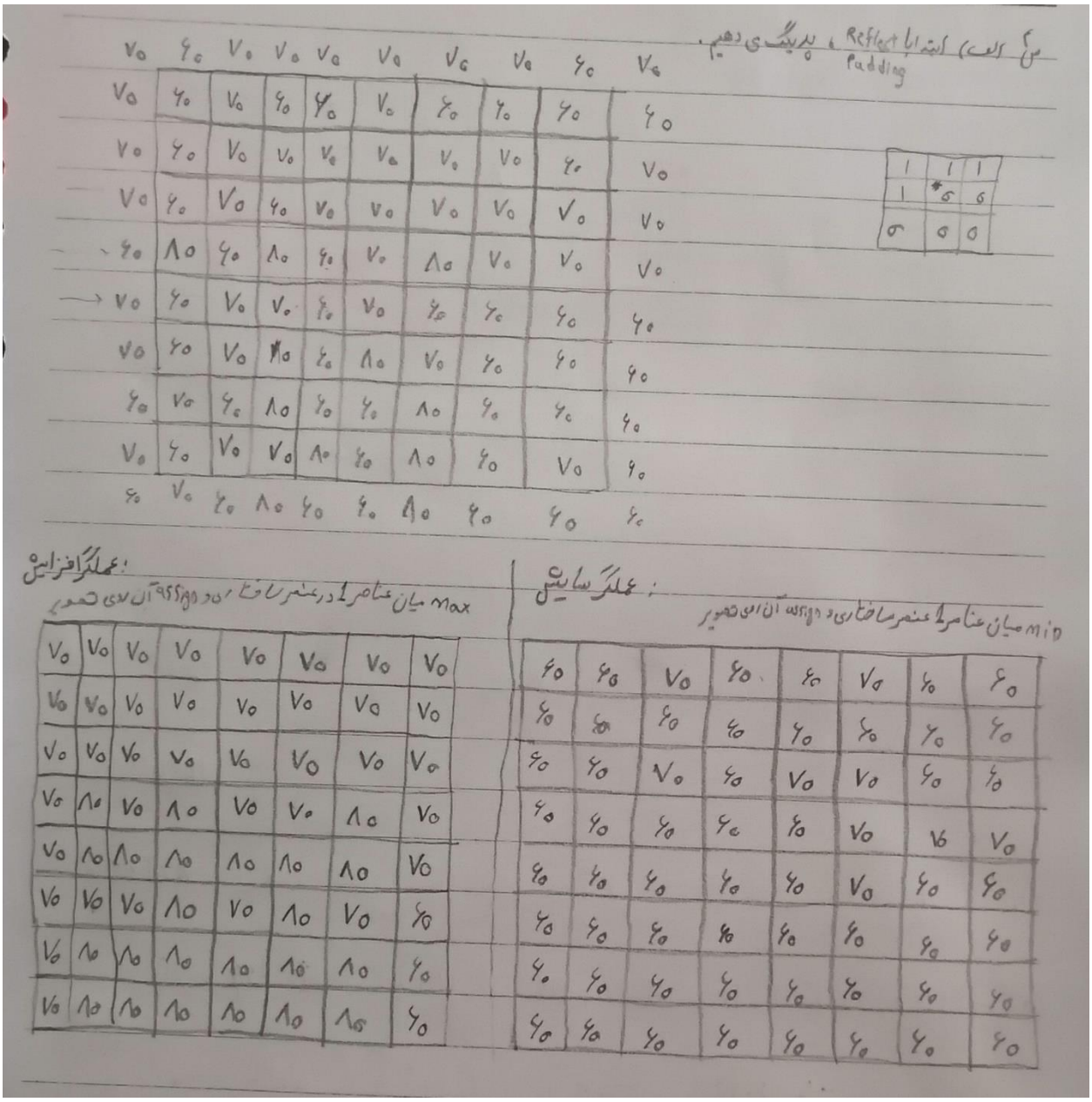
در حل سوال ابتدا تصویر را سیاه سفید کرده و آن را بوسیله تابع های اماده اتسو باینری کرده(در تابع threshold ما عکس منبع و مقدار ترشلد و بیشینه مقدار را تعیین کردیم. همچنین نوع پیاده سازی و ترشولد را تعیین نمودیم در ورودی آخر تابعمان)، سپس مطابق روش درون اسلاید ها ، نقطه seed را مقدار داده و صف را تعیین میکنیم .سپس تا هنگامی که صف خالی نشده ْیک پیکسل را از صف خارج کرده و برای همسایه های ۴تایی متصل به آنْ درصورتی که flag صفر باشد ، وارد صفشان کرده ورنگ خاکستری به آن ها داده و flag را یک می کنیم.

سوال4:

Refrences:

Class slides

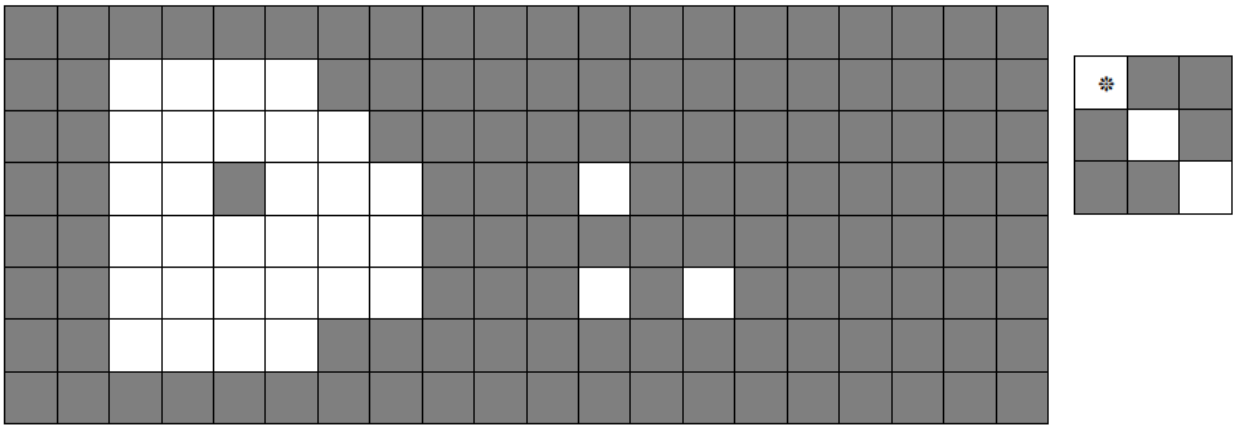
الف)



ب)

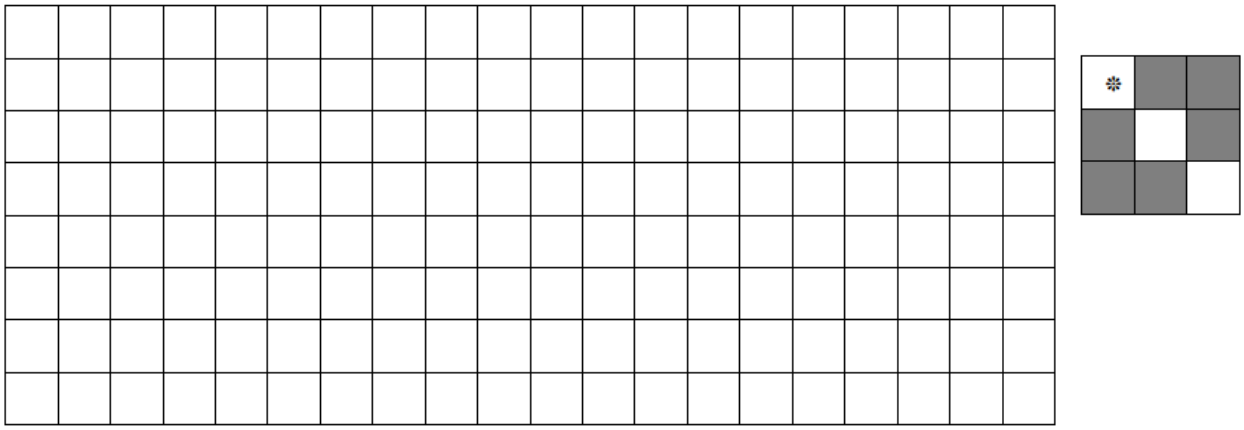
در حل این سوال این نکته را در نظر میگیریم که در عملگر باز ، اول سایش روی تصویر زده و سپس روی نتیجه گسترش میزنیم. در عملگر بسته، اول روی تصویر گسترش میزنیم و بعد سایش روی نتیجه اولیه میزنیم.در سایش ،کرنل باید منطبق باشد وگرنه صفر است خروجی . در افزایش ولی کرنل فقط باید یک خانش منطبق شود تا خروجی 1 شود.

شکل تصویر ورودی هر دو بدین گونه است.



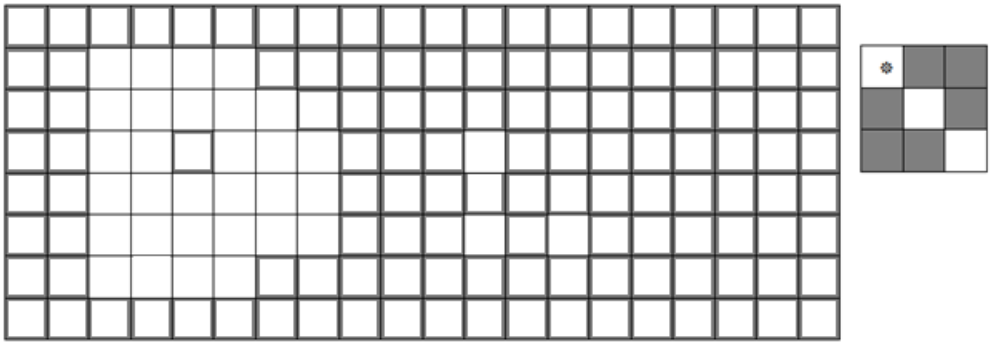
عملگر باز:

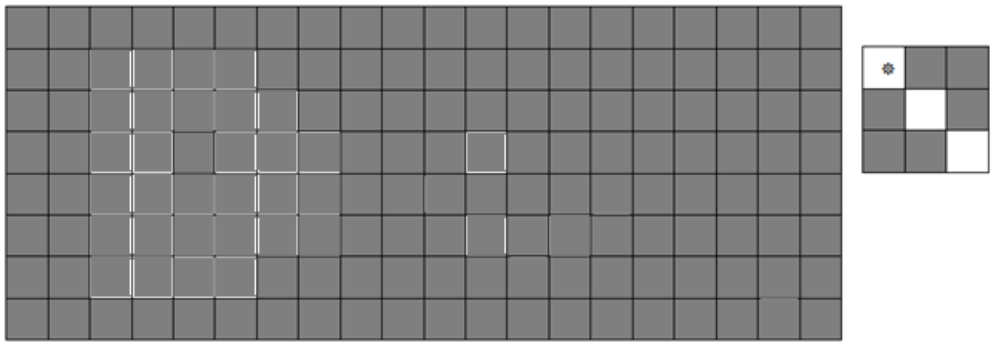
حال ابتدا عملگر باز را انجام میدهیم. با پیاده سازی سایش روی تصویر خروجی تصویری تماما سیاه می شود چون هیچ کدوم منطبق نیست. سپس روی نتیجه نهایی، گسترش زده که درنتیجه تصویر تماما سفید می شود .



عملگر بسته :

ابتدا روی تصویر با توجه به توضیحات اولیه ونکاتش وقتی گسترش بزنیم تصویر تماما سفید می شود. در نهایت وقتی سایش بزنیم با توجه به عدم انتطباق، همش سیاه می شود.



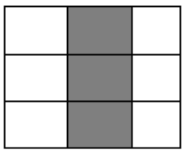


سوال5:

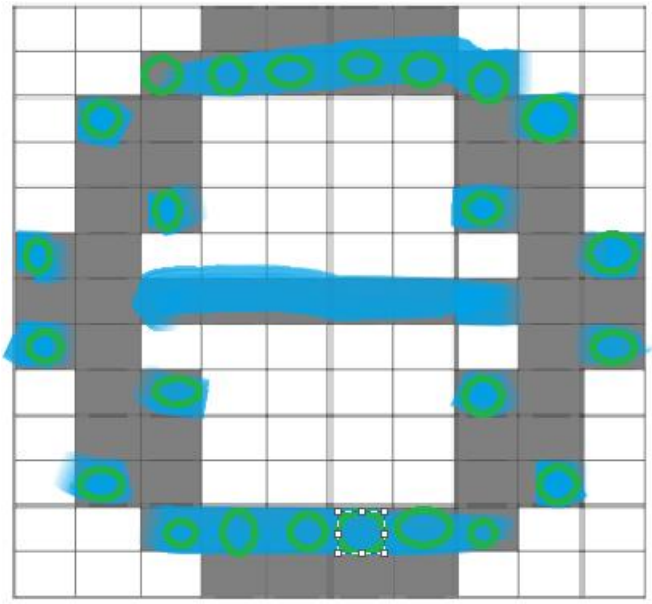
Refrences:

Class slides

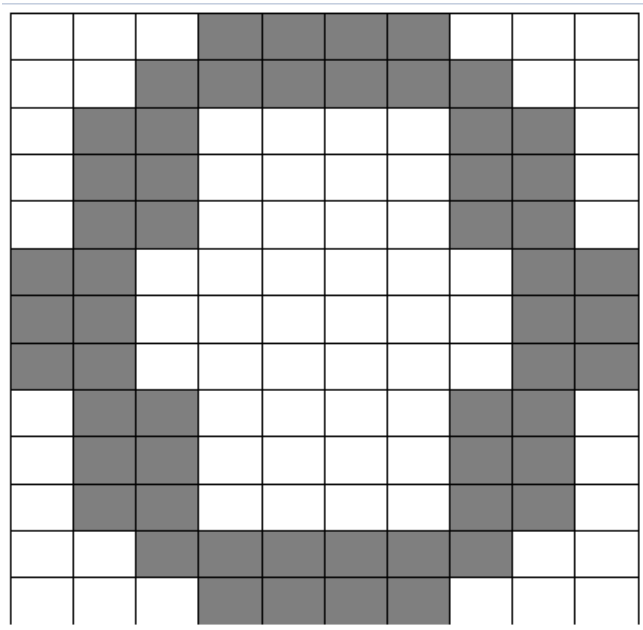
الف)



برای حذف بخش بیان شده،استفاده از عنصر ساختاری بهتر است ، زیرا خط وسط ازنظر افقی دارای پیوستگی بوده وتنها نقطه ایراد آن بخش عمودی آن است.اگر یک فیلتر سایش زده و نقاط مشخص شده در تصویر را حذف کینیم،سپس یک افزایش روی آن بزنیم ، خانه هایی که نقطه سبز روی آن است بازیابی می شوند. البته تمام نقاطی که ناخواسته حذف شده اند بازیابی شده و فقط خط وسط برنمیگردد و در نتیجه ، عبارت حاصل بدون خط مدنظر است.(تبدیل به صفر می شود.)



تصویر نهایی می شود:



ب)

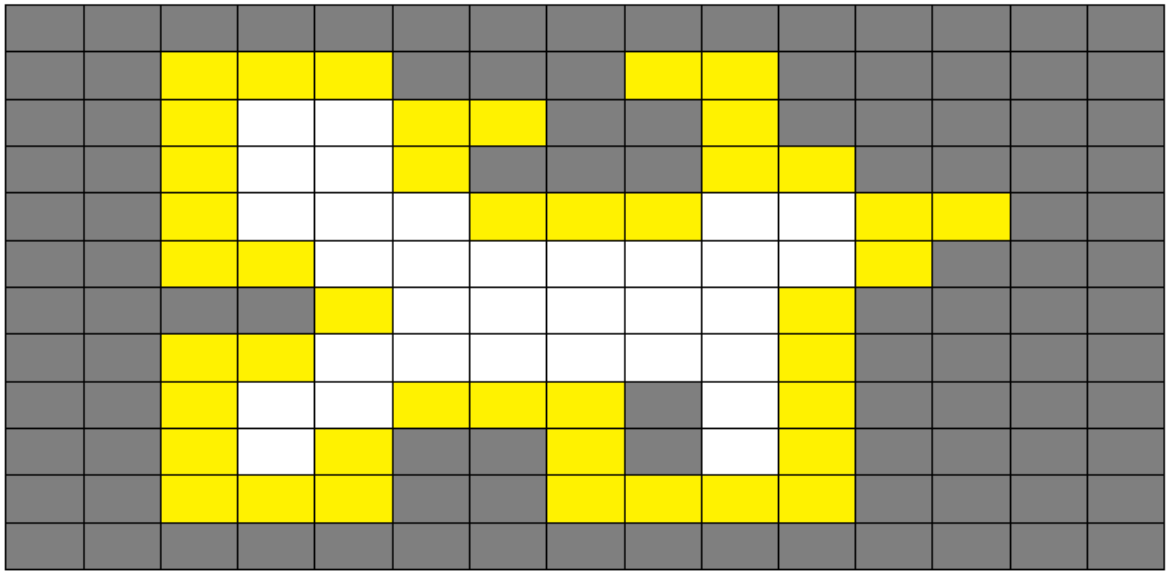
0	0	0
0	1	0
0	-1	0

0	-1	0
0	1	0
0	0	0

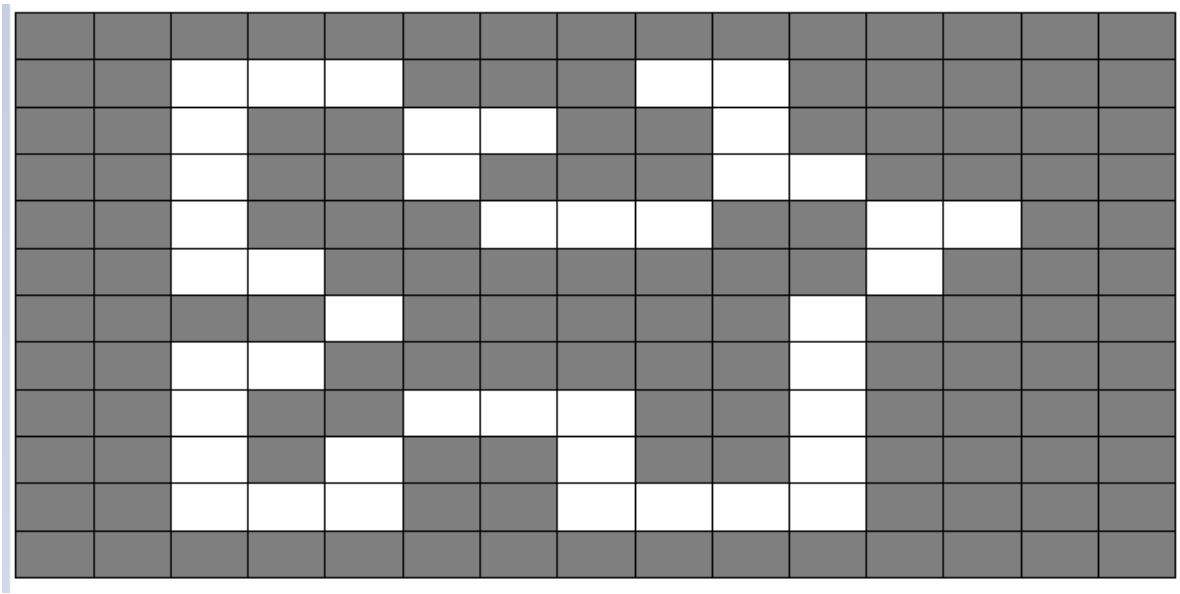
0	0	0
0	1	-1
0	0	0

0	0	0
-1	1	0
0	0	0

برای یافتن مرز های تصویر، بوسیله hit or mis باید از عناصر ساختاری بالا استفاده شود.(به ترتیب برای یافتن جهت لبه (از سمت چپ به راست) چپ، راست، بالا و پایین است).با or کردن فیاترها روی تصویر ، نقاط مرز بدست می ایند که نتیجه آن می شود شکل زیر.در شکل زیر همه خانه های رنگ شده(زرد رنگ) ،سفید یا همان 1 می شوند و بقیه خانه ها (سیاه یا سفید)تبدیل به سیاه یا همان صفر می شوند .(در عکس نهایی کامل مشخص شده است .)



نتیجه نهایی شکل زیر است:



سوال6:

Refrences:

Class slides

https://www.geeksforgeeks.org/python-opencv-cv2-erode-method/

https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html

https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html

الف)

در پیاده سازی این قسمت هر کدام از توابع را جداگانه پیاده سازی می نماییم.

افزایش: ابتدا با اتسو تصویر را باینری کرده سپس با تابع پدینگ نوشته شده ، بدان پدینگ میدهیم.حال یک حلقه روی تصویر زده و برای هر پیکسل ،خودش و 8 همسایه اش را با کرنلش به تابع dilate_utilization میفرستیم. در این تابع یک fکه همان پرچم است گذاشته ، سپس روی 9 پیکسل تصویر و کرنل حرکت کرده و هرجا کرنل1 بود و تصویر متناظر در آن نقطه صفر بود(مشکی) f را یک کند و اگر f صفر بود یعنی شرط را برقرار

میکند و 255 (سفید) برمیگرداند در غیر آن صورت هم صفر(مشکی) برمیگردد.f برای آن است که در پنجره روی تصویر ،یکی از یک ها مطابقت داشت شرط برقراره.

سایش: مشابه تابع قبلی است ولی در e_utilization متفاوت است.(مقدار اولیه f را 1میگذاریم و روی تصویر حرکت میکنیم، هرجا که کرنل یک بود و توصیر مشکی نبود ، f را صفر میکنیم . ادامه آن مانند تابع قبلی است.

تابع open:

در این تابع اول با توابعی که نوشته شده درگذشته ، روی تابع سایش زده و سپس روی نتیجه افزایش میزنیم و خروجی را بر می گردانیم .

تابع close:

در این تابع ابتدا روی تصویر گسترش زده و سپس سایش میزنیم و نتیجه را برمیگردانیم.

ب)

در این قسمت با استفاده از توابع اماده، خروجی های قسمتت های قبل و خواسته های آن را برآورده مینماییم.

توابع اماده شامل cv2.erode,cv2.dilate,cv2.morphologyEx می باشد.در تابع گسترش و سایش ، ورودی، کرنل و تایپ آن ،iteration مشخص می شود.در تابع مورفولوژی نوع باز و بسته بودن آن و کرنل و نوع آن ، مشخص می شود.