

سوال اول

در فایل Q1.ipynb کد این سوال زده شده است هم چنین راه حل تشریحی آن به صورت زیر است در صفحه بعد مشاهده میکنید.:

میانگین: ۸.۱۲

میانه: ۹

مد: ۴

واریانس: ۱۶.۷۴۵۶

هیستوگرام نیز در صفحه بعد نوشته شده است

و بین ۹.۵ و ۱۱.۵ سطح استانه 9.5 بهتر است چون اگر فرمول `otsu` را بران اعمال کنیم مقدارش کمتر میشود.

مراد ①

الف

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

شماره

برای محاسبه هستون اول جدول:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	1	4	1	0	1	1	2	3	3	3	1	1	1

برای محاسبه هستون دوم جدول:

$$\frac{12+13+5+4+9+11+7+10+10+1+8+11+3+2+2+9+12+4+4+4+10+11+12+15+14}{25}$$

پایه:

پایه برابر با 9 است

$$= \frac{203}{25} = 8.12$$

1 2 2 3 4 4 4 4 5 7 8 9 10 11 12 13

موله: عدد 4 سیزدهمین تکرار دارد پس mod برابر با 4 است

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n} = \frac{(1-8.12)^2 + 2(2-8.12)^2 + (3-8.12)^2 + 4(4-8.12)^2 + (5-8.12)^2 + (7-8.12)^2}{25} + \frac{(8-8.12)^2 + 2(9-8.12)^2 + 3(8.12-10)^2 + 3(8.12-11)^2 + 3(8.12-12)^2 + (13-8.12)^2 + (14-8.12)^2 + (15-8.12)^2}{25}$$

$$= 16.7456$$

ب) استاندارد

برای سطح استانه 9.5:

$$s^2 = s_1^2 + s_2^2 =$$

1, 2, 2, 3, 4, 4, 4, 4, 5, 7, 8, 9, 9, 10, 10, 10, 11, 11, 11, 12, 12, 12, 13, 14, 15

$$s^2 = 13 \times 6.639 + 12 \times 2.3541 = 2.3541$$

$$= 114.55769$$

1, 2, 2, 3, 4, 4, 4, 4, 5, 7, 8, 9, 9, 10, 10, 10, 11, 11, 11, 12, 12, 12, 13, 14, 15

$$s^2 = 19 \times 11.717 + 6 \times 1.333 = 230.621$$

برای سطح استانه 9.5 است

سوال دوم

الف) مقایسه‌ای از نظر سرعت و دقت بین روش‌های اتسو و گاوسی اتسو:





$$\sigma_B^2(t) = \sigma^2 - \sigma_w^2(t) \quad \text{روش گاوسی اتسو از لحاظ سرعت بهتر عمل می‌کند زیرا طبق فرمول روبه رو}$$

همان ابتدا واریانس کل تصویر را حساب می‌کنیم و در ادامه واریانس فقط برای پیکسل‌هایی که کمتر از t هستند حساب می‌کنیم ولی در روش اتسو باید واریانس برای دو دسته حساب می‌کردیم که باعث میشد کندتر باشد هم چنین از نظر دقت نیز روش گاوسی اتسو نیز بهتر است زیرا اگر هیستوگرام ما دو حالت باشد یعنی مثلاً به تعداد زیلدی پیکسل بین ۰ تا ۵۰ داشته باشیم و بعد به تعداد زیادی پیکسل بین ۲۰۰ تا ۲۵۰ داشته باشیم و در بین این دو تعداد کمی پیکسل داشته باشیم چون در روش اتسو فقط می‌خواهیم جمع واریانس‌های دو دسته مینیم شود در این حالت **threshold** خوبی به ما نمی‌دهد ولی در روش گاوسی اتسو چون می‌خواهد اختلاف واریانس‌ها ماکسیمم شود **threshold** بهتری می‌دهد.

ب) خیر در الگوریتم اتسو می‌خواهیم جمع واریانس‌های دو دسته را مینیم کنیم البته تعداد پیکسل در هر دو دسته را نیز تاثیر می‌دهیم فرمول آن به صورت زیر است:

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2$$

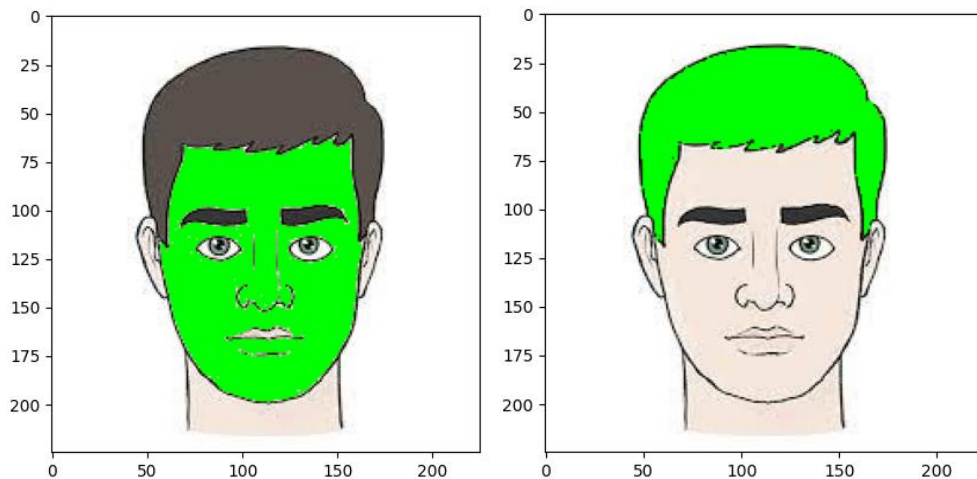
همان مثالی که در قسمت بالا گفتم اگر هیستوگرام دو حالت داشته باشیم به **threshold** مناسبی نمی‌رسیم ولی اگر ماکسیمم واریانس بین کلاسی را محاسبه کنیم به جواب بهتری می‌رسیم

Original image	Histogram	thresholding value	
		Otsu	G.Otsu
 Walkbridge		0	124
 Womandarkhair		0	119

الان در دوتصویر بالا برای اینکه جمع واریانس ها مینیمم شود مقدار * مینیمم را میدهد که در این صورت کل تصویر ما سفید میشود ولی اگر ماکسیمم اختلاف بین کلاسی را در نظر بگیریم مقدار خیلی بهتری را به ما میدهد.

سوال سوم

در ابتدا یک نقطه seed را انتخاب میکنیم (من هم برای مو امتحان کردم و هم برای صورت) و در ادامه یک stack تعریف میکنیم به صورتی تا زمانی که این استک خالی نشده است هر سری یک نقطه از آن pop می کنیم و نقاط همسایه آن را به شرطی که میانگین رنگ سه کانال هایش با هم با این نقطه که pop کردیم و هم نقطه seed کمتر از یک threshold (بر اساس ازمون و خطا این مقادیر را به دست اوردم) بود آن را به استک اضافه میکنیم در واقع روش محلی و سراسری بودن را با هم ترکیب کردم و پس از آن که به استک اضافه کردیم یک ارایه دو بعدی visit نیز تعریف کردم که در ابتدا همه مقادیر آن false است و هر نقطه که وارد استک میشود مقدار آن نقطه را در ارایه visit برابر با true میکنم. آن نقطه را که وارد استک شده است به رنگ سبز در می آوریم تا بتوانیم در تصویر segment نشان دهیم نتایج برای دو قسمت سر و مو به صورت زیر است:



سوال چهارم

الف) جواب آن به صورت زیر است:

در صفحه بعد جواب هست.

(4) الف

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

1	1	1
1	1	1
1	1	1

اول عملگر \max (از پیش)
برای عملگر \max در تصویر ورودی، مقدار \max را در هر پیکسل قرار می‌دهیم.

0	0	0
0	0	1
1	1	1

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

نتیجه

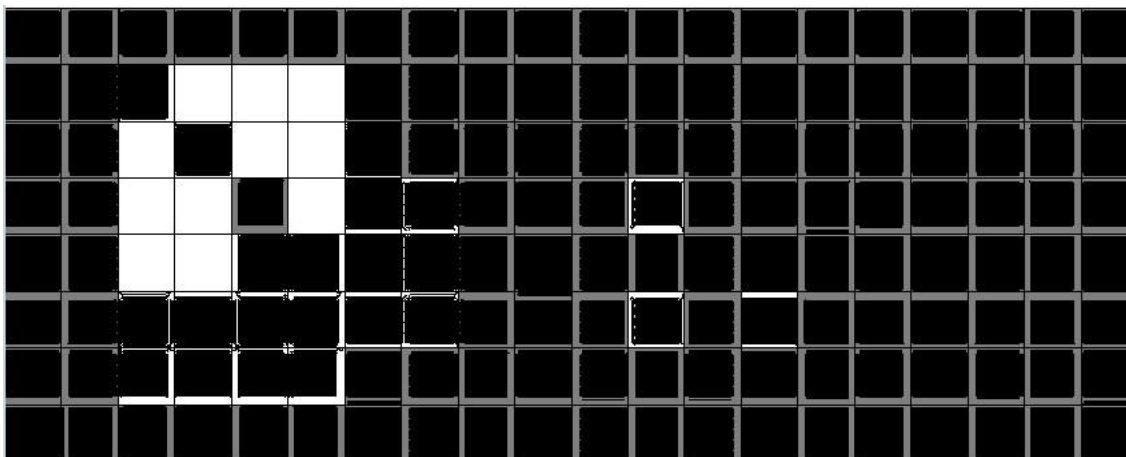
برای عملگر \min (از پیش)
برای عملگر \min در تصویر ورودی، مقدار \min را در هر پیکسل قرار می‌دهیم.

1	1	1
1	0	0
0	0	0

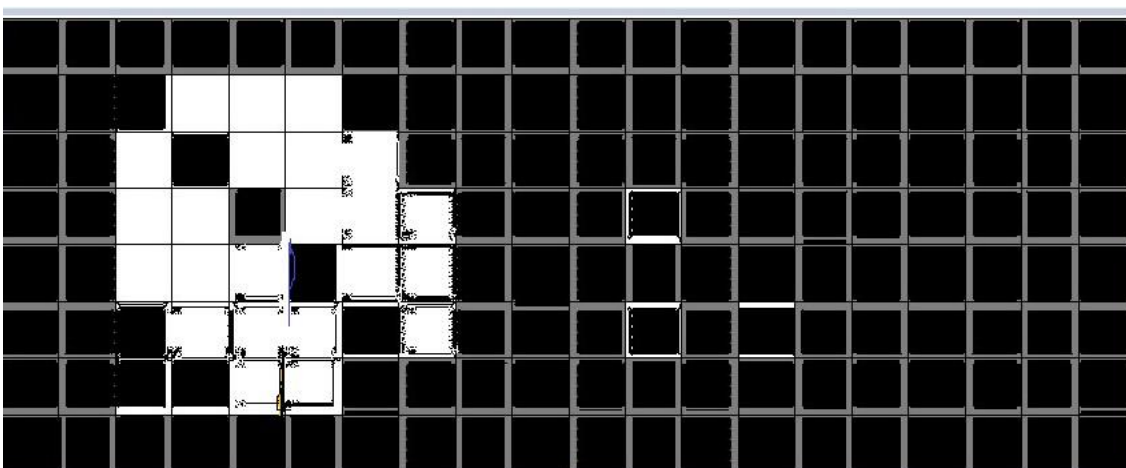
60	60	70	60	60	70	60	60
60	60	60	60	60	60	60	60
60	60	70	60	70	70	70	60
60	60	60	60	60	70	70	70
60	60	60	60	60	70	60	60
60	60	60	60	60	60	60	60
60	60	60	60	60	60	60	60
60	60	60	60	60	60	60	60

ب) جواب عملگر باز به صورت زیر است:

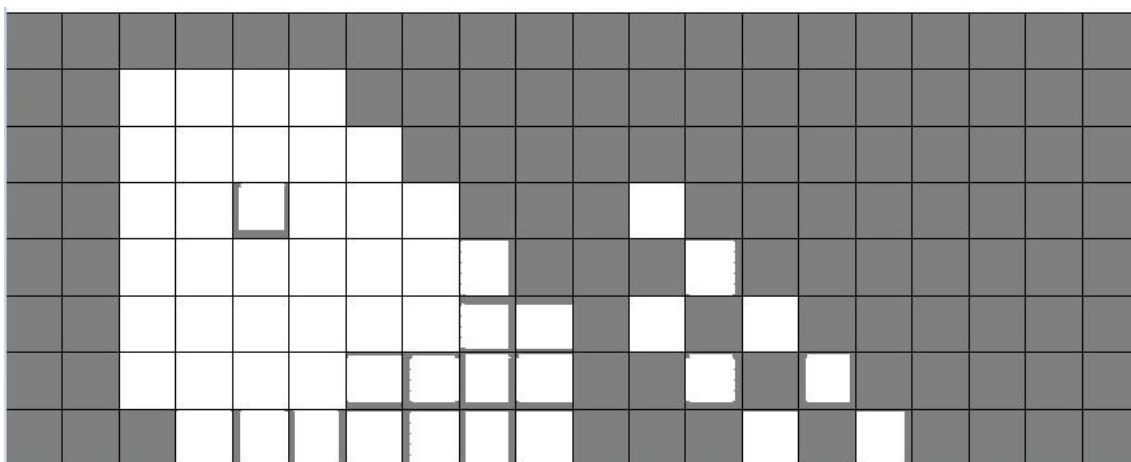
اول عملگر \max در ابتدا باید \max کنیم که به صورت زیر جواب میشود:



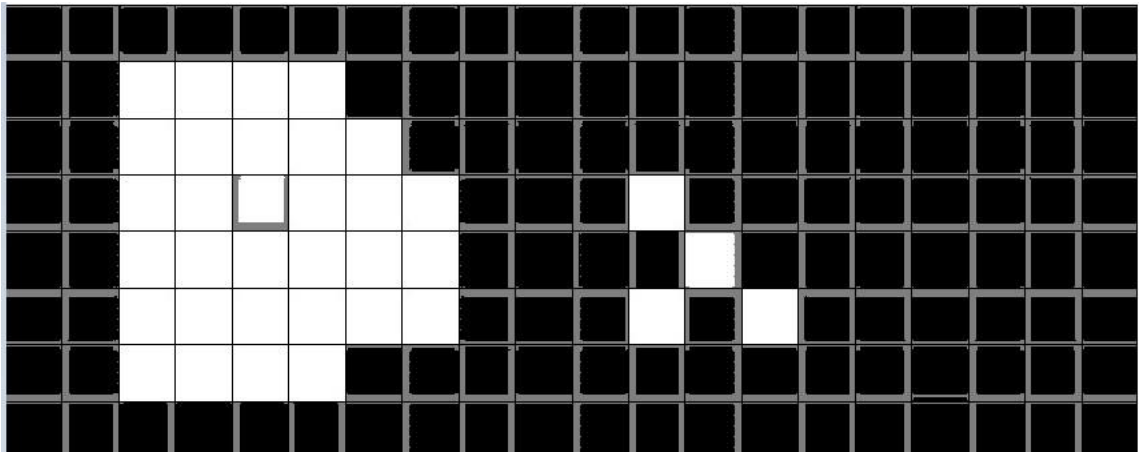
حالا باید عملگر گسترش را انجام بدهیم که شکل ان به صورت زیر میشود:



برای عملگر بسته نیز ابتدا باید گسترش بزنییم که به صورت زیر میشود نقاط خاکستری همان سیاه هستند::

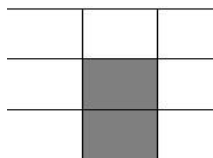


حالا عملگر سایش را میزنیم:

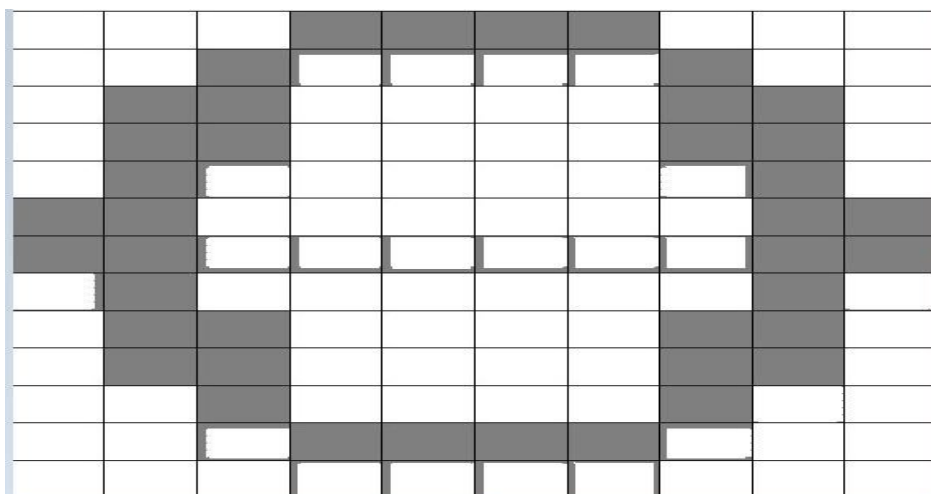


سوال پنجم

الف) در این سوال من پیکسل های سیاه رو به عنوان پیکسل بالاتر یعنی ۱ در نظر گرفتم و پیکسل سفید را پیکسل پایین تر یعنی ۰ در نظر گرفتم اگر بخواهیم فقط خط وسط را حذف کنیم کافی است عملگر سایش را انتخاب کنیم و عنصر ساختاری ما شکل پایین سمت چپ باشه یعنی:



و پس از اینکه شکل را با این عنصر ساختاری عملگر سایش زدیم به صورت زیر میشود که شبیه عدد ۰ است.



ب) با استفاده از ۴ کرنل زیر:

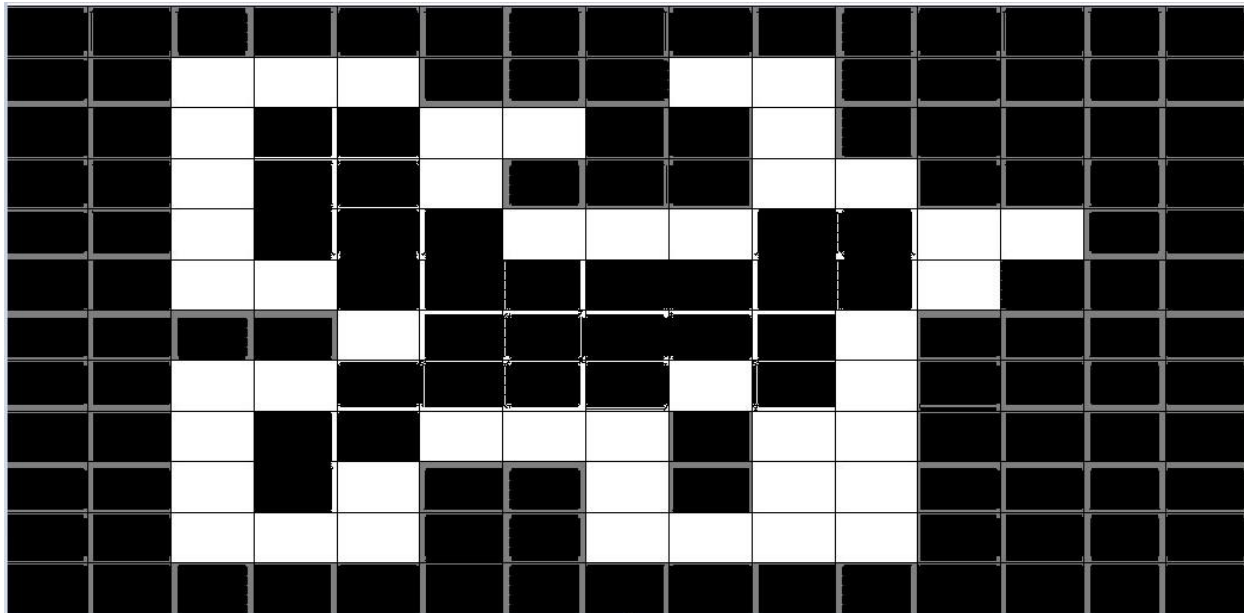
0	0	0
-1	1	0
0	0	0

0	-1	0
0	1	0
0	0	0

0	0	0
0	1	0
0	-1	0

0	0	0
0	1	-1
0	0	0

و مرز تصویر به صورت زیر میشود خانه هایی که سفید هستند مرز تصویر هستند.



سوال ششم

الف) برای عملگر dilate ابتدا می اییم کرنل را بر روی تصویر میگذاریم و مقادیری که بر روی کرنل برابر با ۱ هستند را با هم مقایسه میکنیم و ماکسیمم مقدار را برابر با پیکسل مرکزی که کرنل رو عکس قرار دارد می گذاریم برای عملگر erode نیز همین گونه عمل میکنیم فقط مقدار مینیمم را برابر با پیکسل مرکزی میگذاریم برای عملگر باز و بسته نیز طبق فرمول جزوه که به صورت زیر است عمل میکنیم:

عملگر باز:

$$A \circ B = (A \ominus B) \oplus B$$

عملگر بسته:

ر ی ر

$$A \cdot B = (A \oplus B) \ominus B$$

قسمت ب) نتایج مانند قسمت الف میشود و تصاویر نیز موجود است.

ج) در ابتدا یک وایل زدم برای این هست که تمام دایره های ممکن که مرکز ان ها می تواند اسکلتن باشد را پیدا کنیم

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

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

تو کد هر سری تصویر را منهای تصویر عملگر open میکنیم و با حاصل های قبلی or میکنیم و بعد image را برابر با تصویر erode شده قرار میدهیم

```

while cv2.countNonZero(image)!=0:
    i = i+1

    # apply erosion operator
    eroded = cv2.erode(image, kernel)

    # apply opening operator
    open_ = cv2.dilate(eroded, kernel)

    # apply difference operator between input image and opened image
    diff = cv2.subtract(image, open_)

    # apply OR operator on the new image and previous image
    res = cv2.bitwise_or(res, diff)

    # update input image
    image = eroded

```

و در نهایت اسکلتن تصاویر به صورت زیر میشود:

