

ب) حالت کسینوس، توجه به حد آستانه 12 می توان محاسب کرد که:

1	2	3	4	5	6	7	8	9
13.04	13.04	13.04	9.958	9.958	2.749	0.79	0.79	0.79

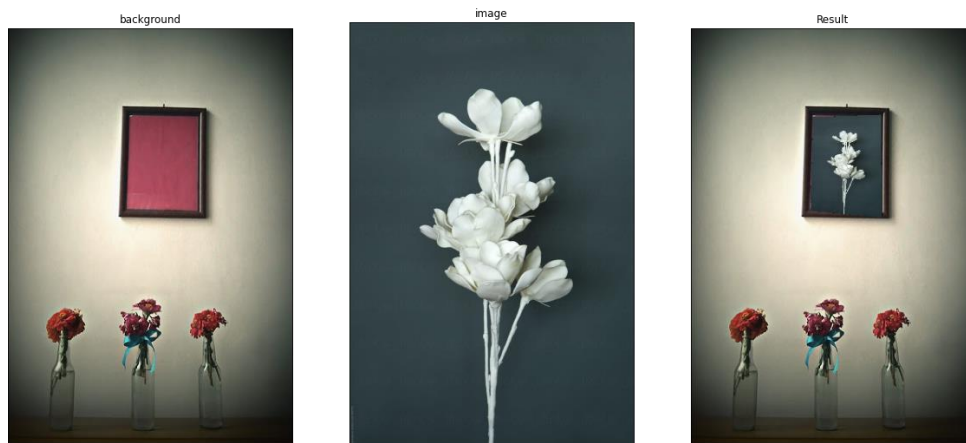
10	11	12	13	14	15
0.79	0.79	0.79	4.639	9.735	12.034

در مجموع باید 15 بار محاسبات برای بازه 1 تا 15 انجام شود که همان بازه شدت روشنایی ها است.

همانطور که در جدول بالا مشخص است در بازه ی 7 تا 12 پایین ترین مقدار واریانس داریم پس میتوان یکی از این مقادیر را انتخاب کرد به عنوان حد آستانه (زیرا پیسکلی با این شدت روشنایی ها وجود ندارد پس فرقی نمیکند)

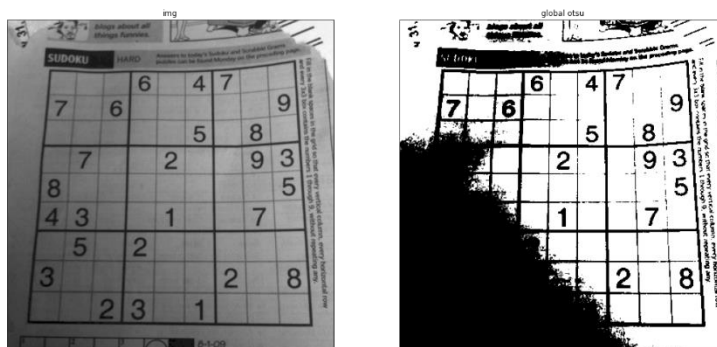
پاسخ سوال دوم

هدف از این سوال انتقال تصویر یک گل به داخل یک قاب بود. ابتدا 4 نقطه کلیدی از قاب که در واقع 4 گوشه آن نیز محسوب می شود را به صورت دستی انتخاب کردیم و مختصات آن ها را یافتیم. در مرحله بعد نقاط متناظر با این 4 نقطه کلیدی که 4 گوشه تصویر گل می باشد را بوسیله یک تبدیل perspective به قاب انتقال دادیم.



پاسخ سوال سوم

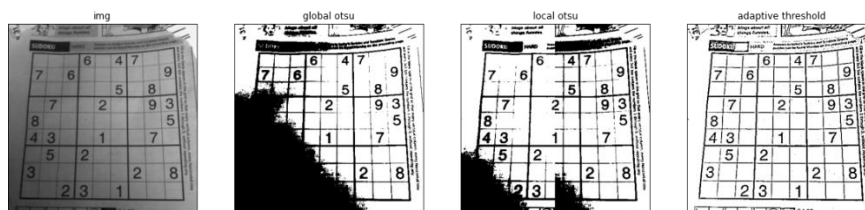
در این پیاده سازی تلاش کردیم تا با ۳ روش زیر یک تصویر را به صورت باینری در بیاوریم.
- روش اول که global otsu با استفاده از الگوریتم های گفته شده و وارپانس به باینری کردن پرداختیم.



- در روش دوم تصویر را به ۴ ناحیه اصلی تقسیم کردیم و برای هر ناحیه به صورت جداگانه روش قسمت قبل را تکرار کردیم.



نتیجه بسیار بهتر شد ولی همچنان ایرادات مشهودی داشت پس به سراغ روش دیگری رفتیم.
-آستانه گذاری وفقی: بدین صورت که برای هر پیکسل بسته به پیکسل های همسایه مشخص میکردیم پیکسل روشن است یا خیر.



مقایسه خروجی ها:

همانطور که در تصویر بالا نیز مشخص است خروجی OpenCV بسیار واضح تر است که به دلیل این است در این الگوریتم بلوک های اطراف هر پیکسل را برای آستانه گذاری انتخاب میکند که نتیجه ی بهتری از حالتی که ما کل تصویر را بررسی میکنیم دارد.

src

عکس ورودی

dst.

عکس خروجی

maxValue

متغیری از نوع دو نشان دهنده مقداری است که اگر مقدار پیکسل بیشتر از مقدار آستانه باشد ، باید داده شود.

adaptiveMethod

متغیری که نشانگر دو روش آستانه گذاری است:

ADAPTIVE_THRESH_MEAN_C: مقدار آستانه میانگین محلی است.

ADAPTIVE_THRESH_GAUSSIAN_C: مقدار آستانه مجموع وزنی مقادیر محله است که در آن وزنها یک

پنجره گاوسی است.

thresholdType

متغیری از نوع عدد صحیح که نشان دهنده نوع آستانه مورد استفاده است.

blockSize

متغیری از نوع عدد صحیح که نشان دهنده اندازه محله پیکسل است برای محاسبه مقدار آستانه استفاده می شود.

C

یک متغیر از نوع دو نشان دهنده ثابت استفاده شده در هر دو روش (تفکیک شده از میانگین یا میانگین گاوسی).

پاسخ سوال چهارم

در این سوال ابتدا نقاط کلیدی را با توابع گفته شده پیدا کردم در طی مسیر انجام سوال فیلتر هایی برای گوشه ها در نظر گرفتم که نتیجه ی بهتری حاصل شود.

سپس حول هر یک از این نقاط کلیدی پنجره هایی با اندازه ی ذکر شده در کد در نظر گرفتم به عنوان یک توصیفگر ساده.

بعد این پنجره ها را دو به دو به معنی به ازای هر پنجره از عکس 1 با تمام پنجره های عکس 2 مقایسه میکنیم با استفاده از توابع **opencv** که نتیجه یک امتیاز برمیگرداند، بیشترین امتیازی که حاصل شد را به عنوان نقطه متناظر در نظر میگیریم.

سپس به ازای مقادیر رندوم بین این نقاط خط های رنگی میکشیم.



مزایا و معایب NCC

NCC ساده ترین اما غیر موثرترین رویکرد است زیرا در روشنایی خطی و تغییرات مقیاس، ثابت نیست عملکرد آن. از لحاظ زمانی خیلی بهینه نیست که دلیل شده در موارد **real-time** از آن استفاده نشود. بیشتر برای بدست آوردن نقاط متراکم متناظر از NCC استفاده می شود.

مقایسه ایی با الگوریتم SIFT در یکی از [مقالات](#):

Table 5. Time performance of our model and the existing model of space resection.

TLS Image Resolution	1000 × 512	2000 × 512	5000 × 512
Existing model	74.8 s	140.6 s	328.1 s
Our model	29.9 s	56.1 s	120.3 s
SIFT	5.0 s	10.0 s	23.2 s
NCC	>1 h	>3 h	>6 h
SIFT + NCC	84.7 s	167.3 s	320.0 s