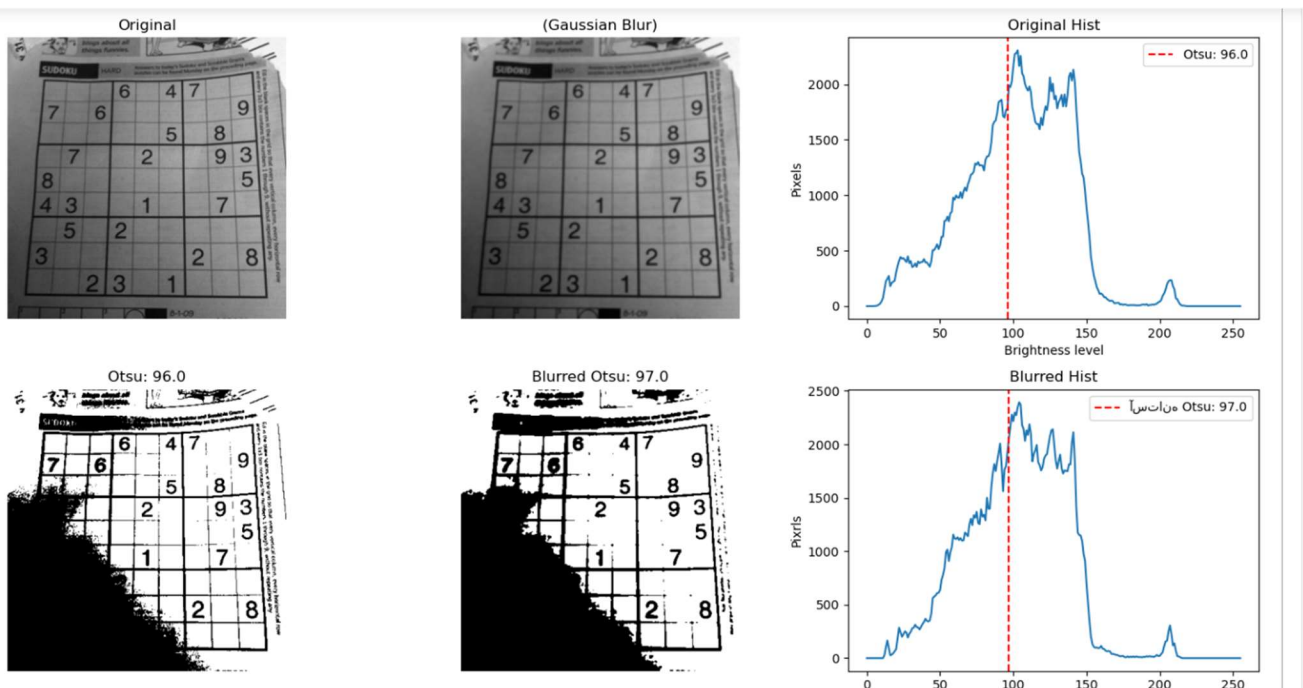


## آستانه گذاری تطبیقی:

در جلسه قبلی با یکی از بهترین روش های آستانه گذاری به نام روش Otsu آشنا شدیم. هر چند که این روش در بسیاری از موارد کاربردی است اما گاهی در استفاده از آن ممکن است با مشکل مواجه شویم. یکی از بزرگترین مشکلات استفاده از این روش زمانی مشخص می شود که نور به صورت مایل به عکس تابیده شده باشد و در توزیع نور شرایط یکسانی مهیا نباشد. در این حالت روش Otsu کاربردی ندارد. به این عکس توجه کنید:



همانطور که مشاهده می کنید در آستانه گذاری برای این تصویر از روش Otsu استفاده شده اما به دلیل اینکه جهت تابش نور به تصویر از چپ به راست بوده در سمت راست تصویر نتوانسته ایم به نتیجه دلخواه برسیم.

آستانه گذاری تطبیقی (Adaptive Thresholding) یک تکنیک قدرتمند در OpenCV برای تبدیل یک تصویر grayscale به یک تصویر باینری (سیاه و سفید) است، به طوری که مقدار آستانه برای هر پیکسل به صورت محلی و بر اساس همسایگی آن پیکسل تعیین می شود. این روش در مقایسه با آستانه گذاری سراسری (Global Thresholding) که از یک مقدار آستانه ثابت برای کل تصویر استفاده می کند، در شرایطی که نورپردازی در تصویر غیریکنواخت است، نتایج بسیار بهتری ارائه می دهد.

در OpenCV، دو نوع اصلی از آستانه گذاری تطبیقی وجود دارد که می توانید از آن ها استفاده کنید:

1. `cv2.adaptiveThreshold()` این تابع اصلی برای انجام آستانه گذاری تطبیقی است.

```
dst = cv2.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C)
```

در این کد:

**src** تصویر ورودی. grayscale.

**maxValue** مقدار پیکسلی که به پیکسل‌هایی اختصاص داده می‌شود که مقدار آن‌ها از آستانه محلی بیشتر است. معمولاً این مقدار 255 (سفید) است.

**adaptiveMethod** روش مورد استفاده برای محاسبه آستانه محلی. دو مقدار رایج برای این پارامتر وجود دارد :

- **cv2.ADAPTIVE\_THRESH\_MEAN\_C** آستانه محلی میانگین مقدار پیکسل‌های همسایگی (تعریف شده توسط **blockSize** منهای مقدار ثابت **C** است).

- **cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C** آستانه محلی میانگین وزنی مقدار پیکسل‌های همسایگی با استفاده از یک پنجره Gaussian منهای مقدار ثابت **C** است. این روش معمولاً نتایج بهتری نسبت به میانگین ساده ارائه می‌دهد زیرا به پیکسل‌های نزدیک‌تر وزن بیشتری می‌دهد.

**thresholdType** نوع آستانه‌گذاری. معمولاً از **cv2.THRESH\_BINARY** استفاده می‌شود، به این معنی که اگر مقدار پیکسل از آستانه محلی بیشتر باشد، به **maxValue** تنظیم می‌شود، در غیر این صورت به 0 تنظیم می‌شود. می‌توانید از **cv2.THRESH\_BINARY\_INV** نیز استفاده کنید که عملکرد معکوس دارد.

**blockSize** اندازه بلوک همسایگی که برای محاسبه آستانه محلی استفاده می‌شود. این مقدار باید یک عدد فرد مثبت باشد. بلوک‌های بزرگتر می‌توانند جزئیات کوچکتر را از بین ببرند، در حالی که بلوک‌های کوچکتر ممکن است به نویز حساس‌تر باشند.

**C** یک مقدار ثابت است که از میانگین یا میانگین وزنی همسایگی کم می‌شود. این پارامتر به شما امکان می‌دهد حساسیت آستانه‌گذاری را تنظیم کنید. مقادیر مثبت **C** منجر به آستانه‌گذاری سخت‌گیرانه‌تر و مقادیر منفی **C** منجر به آستانه‌گذاری ملایم‌تر می‌شوند.

### نحوه عملکرد آستانه‌گذاری تطبیقی:

به جای استفاده از یک مقدار آستانه ثابت برای کل تصویر، آستانه‌گذاری تطبیقی برای هر پیکسل یک مقدار آستانه منحصر به فرد محاسبه می‌کند. این کار با در نظر گرفتن یک ناحیه همسایگی در اطراف پیکسل مورد نظر انجام می‌شود. سپس، بر اساس روش انتخابی **cv2.ADAPTIVE\_THRESH\_MEAN\_C** یا **cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C** میانگین یا میانگین وزنی شدت پیکسل‌های این همسایگی محاسبه شده و مقدار ثابت **C** از آن کم می‌شود. این مقدار به عنوان

آستانه برای پیکسل مرکزی در نظر گرفته می شود. در نهایت، مقدار پیکسل مرکزی با این آستانه مقایسه شده و بر اساس نوع آستانه گذاری `cv2.THRESH_BINARY` یا `cv2.THRESH_BINARY_INV` به مقدار باینری 0 یا `maxValue` تبدیل می شود.

### مزایای آستانه گذاری تطبیقی:

- عملکرد بهتر در نورپردازی غیریکنواخت: این روش به خوبی با تغییرات تدریجی در نورپردازی تصویر سازگار می شود، زیرا آستانه برای هر ناحیه به طور جداگانه محاسبه می شود.
- حفظ جزئیات محلی: در مقایسه با آستانه گذاری سراسری، آستانه گذاری تطبیقی می تواند جزئیات محلی تصویر را بهتر حفظ کند، زیرا آستانه به ویژگی های محلی تصویر حساس است.

### معایب آستانه گذاری تطبیقی:

- پیچیدگی بیشتر: محاسبات بیشتری نسبت به آستانه گذاری سراسری انجام می شود.
- نیاز به تنظیم پارامترها: انتخاب مقادیر مناسب برای `blockSize` و `C` می تواند بر کیفیت نتیجه تأثیر بگذارد و ممکن است نیاز به آزمایش داشته باشد.
- حساسیت به نویز: در برخی موارد، اگر نویز زیادی در تصویر وجود داشته باشد، آستانه گذاری تطبیقی ممکن است نویز را نیز به عنوان بخشی از ساختار محلی در نظر بگیرد و منجر به نتایج نامطلوب شود.

### کاربردهای آستانه گذاری تطبیقی:

آستانه گذاری تطبیقی در بسیاری از کاربردهای پردازش تصویر مفید است، از جمله:

- تشخیص متن در تصاویر با نورپردازی غیریکنواخت: مانند اسناد اسکن شده یا تصاویر گرفته شده در شرایط نوری نامناسب.
- بخش بندی اشیاء در تصاویر با پس زمینه پیچیده: جایی که کنتراست محلی مهم است.
- پردازش تصاویر پزشکی: که ممکن است دارای تغییرات تدریجی در شدت باشند.
- بینایی ماشین صنعتی: برای تشخیص قطعات یا عیوب در شرایط نوری مختلف.

برای انجام عملیات آستانه گذاری تطبیقی می توانیم از قطعه کد زیر استفاده کنیم:

```
# Load the image
img = cv2.imread('images/sudoku.jpg', cv2.IMREAD_GRAYSCALE)

# Apply adaptive thresholding (mean method)
thresh_mean = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
                                     cv2.THRESH_BINARY, 11, 2)

# Apply adaptive thresholding (Gaussian method)
thresh_gauss = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                     cv2.THRESH_BINARY, 11, 2)

# Apply global thresholding (for comparison)
_, thresh_global = cv2.threshold(img, 128, 255, cv2.THRESH_BINARY)
```

در این کد هر دور روش آستانه گذاری بیان شده اند. اما بیایید ببینیم در هر خط کد چه می گذرد

**thresh\_mean = cv2.adaptiveThreshold(...):**

- **cv2.adaptiveThreshold():** این یک تابع از OpenCV است که برای اعمال آستانه گذاری تطبیقی (Adaptive Thresholding) بر روی یک تصویر grayscale استفاده می شود. آستانه گذاری تطبیقی روشی است که در آن مقدار آستانه برای هر پیکسل به صورت محلی و بر اساس همسایگی آن پیکسل تعیین می شود.
- **img:** تصویر grayscale ورودی که در خط قبلی بارگیری شده است.
- **255:** این مقدار، حداکثر مقدار پیکسلی است که به پیکسل هایی اختصاص داده می شود که مقدار شدت آن ها از آستانه محلی بیشتر باشد. در تصاویر باینری (سیاه و سفید)، معمولاً این مقدار 255 (سفید) است.
- **cv2.ADAPTIVE\_THRESH\_MEAN\_C:** این یک روش برای محاسبه آستانه محلی است. در این روش، آستانه برای هر پیکسل به عنوان میانگین مقدار شدت پیکسل های همسایگی آن (که اندازه آن در پارامتر بعدی مشخص می شود) منهای یک مقدار ثابت (که در آخرین پارامتر مشخص می شود) محاسبه می شود.
- **cv2.THRESH\_BINARY:** این نوع آستانه گذاری را مشخص می کند. اگر مقدار شدت پیکسل فعلی از آستانه محلی محاسبه شده بیشتر باشد، مقدار آن به `maxValue` (در اینجا 255) تنظیم می شود؛ در غیر این صورت، مقدار آن به 0 (سیاه) تنظیم می شود.
- **11:** این اندازه بلوک همسایگی است که برای محاسبه آستانه محلی استفاده می شود. در اینجا، یک بلوک 11x11 پیکسلی در نظر گرفته می شود. این مقدار باید یک عدد فرد مثبت باشد.

- **2:** این یک مقدار ثابت است که از میانگین محاسبه شده در روش `cv2.ADAPTIVE_THRESH_MEAN_C` کم می‌شود. این پارامتر به شما امکان می‌دهد حساسیت آستانه‌گذاری را تنظیم کنید.
- نتیجه اعمال آستانه‌گذاری تطبیقی با استفاده از روش میانگین در متغیری با نام `thresh_mean` ذخیره می‌شود. این متغیر حاوی تصویر باینری حاصل خواهد بود.

**`thresh_gauss = cv2.adaptiveThreshold(...):`** این خط نیز دقیقاً مانند خط قبلی عمل می‌کند و آستانه‌گذاری تطبیقی را بر روی تصویر `img` اعمال می‌کند. با این حال، تفاوت در روش محاسبه آستانه محلی است :

- **`cv2.ADAPTIVE_THRESH_GAUSSIAN_C:`** در این روش، به جای میانگین ساده، یک میانگین وزنی از شدت پیکسل‌های همسایگی (با استفاده از یک تابع گوسی) برای محاسبه آستانه محلی استفاده می‌شود. پیکسل‌های نزدیک‌تر به پیکسل مرکزی وزن بیشتری در محاسبه میانگین دارند. این روش معمولاً نتایج بهتری نسبت به روش میانگین ساده ارائه می‌دهد زیرا به توزیع شدت پیکسل‌ها در همسایگی توجه بیشتری دارد.
- سایر پارامترها (`img, 255, cv2.THRESH_BINARY, 11, 2`) همانند خط قبلی هستند و عملکرد مشابهی دارند.
- نتیجه اعمال آستانه‌گذاری تطبیقی با استفاده از روش گوسی در متغیری با نام `thresh_gauss` ذخیره می‌شود.