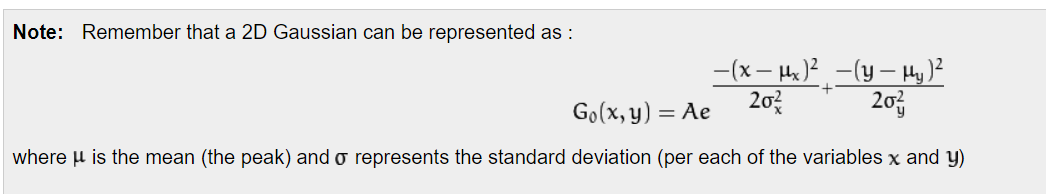
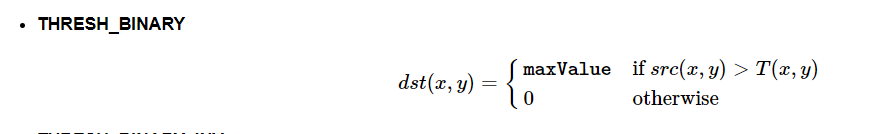
1. **GaussianBlur**,

sử dụng phương pháp 2D convolution (image filtering) sử dụng kernel Gassian

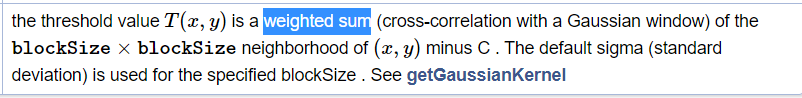
-> tác dụng là để lọc nhiễu.

cv2.GaussianBlur(img, (9, 9), 0) parameter: src, kernel size, độ lệch theo x



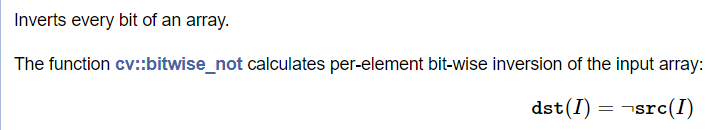
1. **Threshold**

adaptiveMethod: cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C The threshold value is a gaussian-weighted sum of the neighbourhood values minus the constant C.

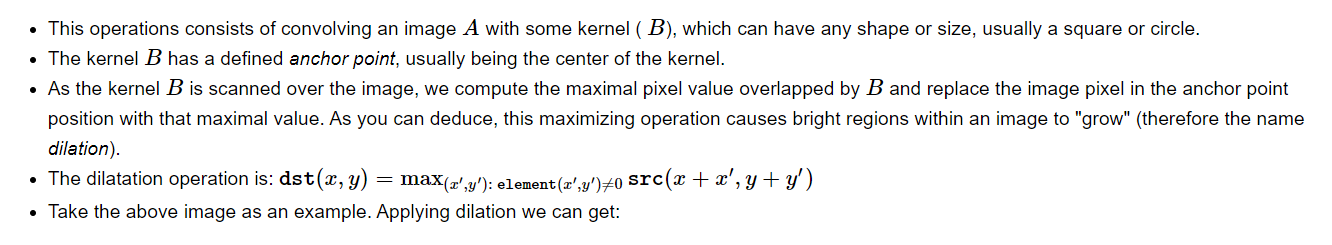


1. **bitwise\_not**

Đảo ngược giá trị max value thành 0 và ngược lại



1. **dilate**



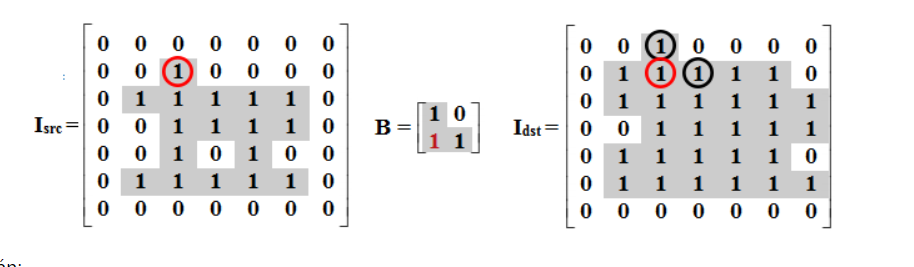
Kernel đc sử dung:

[ 0 1 0

1 1 1

0 1 0 ]

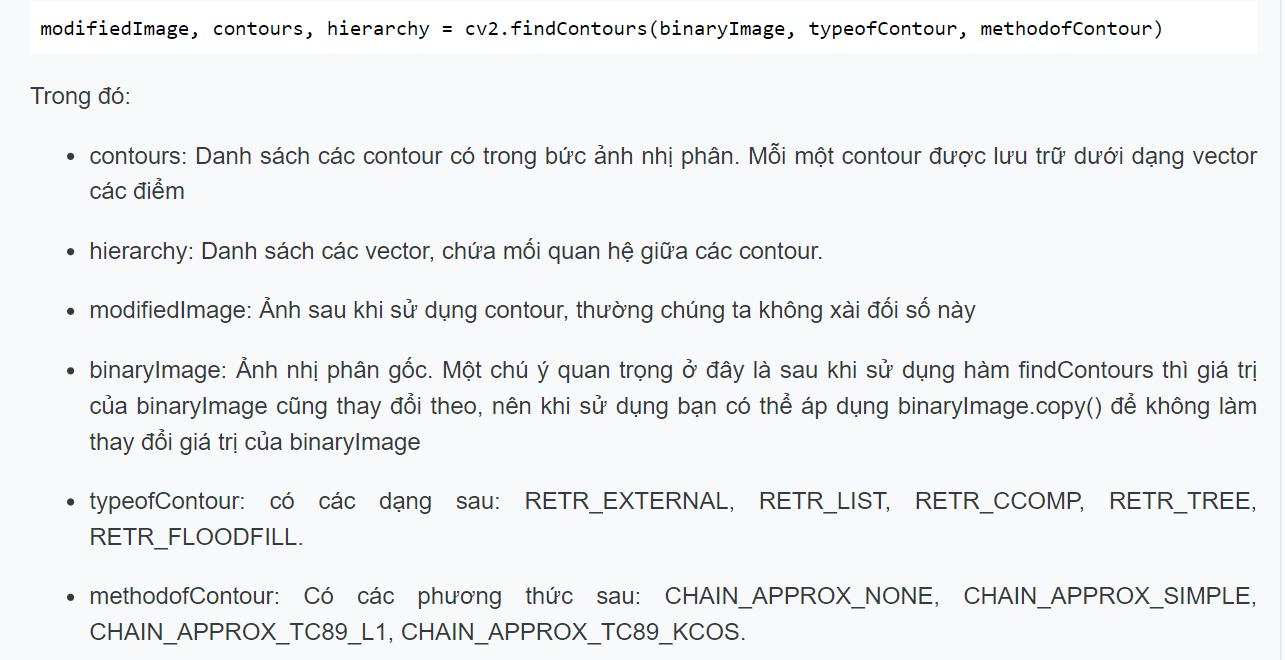
Ví dụ



**II Find lagest polygon**

4. Find contour

contours, h = cv2.findContours(img.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

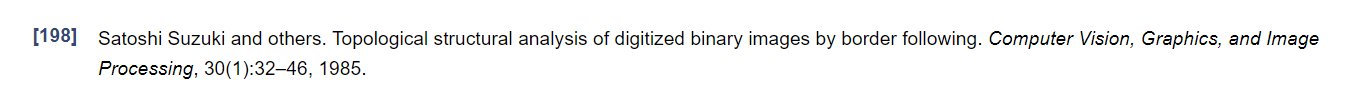
contour tập hợp các điểm liên tục tạo thành một đường cong, không có khoảng hở trong đường cong đó. Bắt buộc phải nhị phân hóa bức ảnh. Tìm các đối tượng màu trắng trên nền đen (do vậy mà ở trên dùng bitwise\_not)

sort contour theo diện tích: contours = sorted(contours, key=cv2.contourArea, reverse=True)

thứ tự giảm dần.

contour lớn nhất chính là đường viền của ma trận sudoku.

Tài liệu về find contour



**III Warping**

Hình dạng ban đầu -> hình dạng cần đổi -> cần tới ma trận chuyển đổi (dựa vào hàm getPerspectiveTranform

