

▼ Contents:

Finding Image Details

Splitting an Image into Individual Channels

Digital Negative

Image Thresholding

Adaptive Thresholding

```
import cv2
from google.colab import files
uploaded=files.upload()
```

No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving flower2.png to flower2.png

▼ *Finding Image Details*

```
img = cv2.imread('flower2.jpg')
from google.colab.patches import cv2_imshow
cv2_imshow(img)
```



▼ *Splitting an Image into Individual Channels*



```
blue, green, red = cv2.split(img) # Split the image into its channels  
cv2_imshow(red) # Display the red channel in the image  
cv2_imshow(blue) # Display the red channel in the image  
cv2_imshow(green) # Display the red channel in the image
```



```
#img_gs = cv2.imread('lotus.jpg', cv2.IMREAD_GRAYSCALE) # Convert image to grayscale  
img_gs = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY ) #Convert image to grayscale  
cv2.imshow(img_gs) # Display the grayscale version of image
```



▼ *Digital Negative*

```
import numpy as np
max_intensity = 255
(h, w) = img_gs.shape
output = img_gs.copy()
for i in np.arange(h):
    for j in np.arange(w):
        a = output.item(i,j)
        b = max_intensity - a
        output.itemset((i,j), b)
cv2.imwrite('images/brightness.jpg', output)
cv2_imshow(output)
```



▼ Image Thresholding

If pixel value is greater than a threshold value, it is assigned one value (may be white), else it is assigned another value (may be black).

`cv2.threshold`: First argument - the source image, which should be a grayscale image. Second argument - the threshold value which is used to classify the pixel values. Third argument - the `maxVal` which represents the value to be given if pixel value is more than (sometimes less than) the threshold value. OpenCV provides different styles of thresholding and it is decided by the fourth parameter of the function. Different types are:

`cv2.THRESH_BINARY` `cv2.THRESH_BINARY_INV` `cv2.THRESH_TRUNC`

```
# from google.colab import files
# uploaded=files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving lotus.jpg to lotus (?) .jpg

```
#img1 = cv2.imread('lotus.jpg', 0)
cv2_imshow(img_gs)
# Perform binary thresholding on the image with T = 125
ret,thresh1 = cv2.threshold(img_gs, 125, 255, cv2.THRESH_BINARY)
#from google.colab.patches import cv2_imshow
```

```
cv2_imshow(thresh1)
```



```
ret,thresh2 = cv2.threshold(img1,125,255,cv2.THRESH_BINARY_INV)
cv2_imshow(thresh2)
```



```
ret,thresh3 = cv2.threshold(img1,125,255,cv2.THRESH_TRUNC)
cv2_imshow(thresh3)
```



```
img2 = cv2.medianBlur(img1,5)
cv2_imshow(img2)
```



▼ Adaptive Thresholding

It is used when image has different lighting conditions in different areas. The threshold for a small regions of the image. So we get different thresholds for different regions of the same image and it gives us better results for images with varying illumination.

cv2.ADAPTIVE_THRESH_MEAN_C : threshold value is the mean of neighbourhood area.

cv2.ADAPTIVE_THRESH_GAUSSIAN_C : threshold value is the weighted sum of neighbourhood values where weights are a gaussian window.

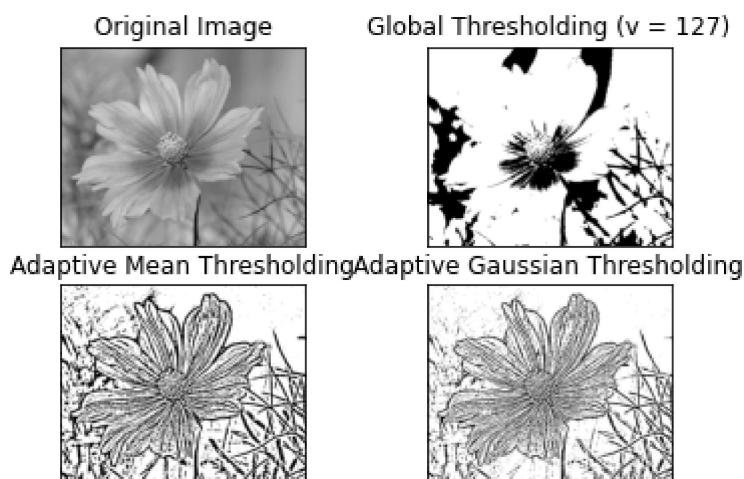
Syntax: cv2.adaptiveThreshold(source, maxVal, adaptiveMethod, thresholdType, blockSize, constant)

```
import numpy as np
from matplotlib import pyplot as plt

ret,th1 = cv2.threshold(img2,127,255,cv2.THRESH_BINARY)
th2 = cv2.adaptiveThreshold(img2,255,cv2.ADAPTIVE_THRESH_MEAN_C,\
    cv2.THRESH_BINARY,11,2)
th3 = cv2.adaptiveThreshold(img2,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,\
    cv2.THRESH_BINARY,11,2)

titles = ['Original Image', 'Global Thresholding (v = 127)',
    'Adaptive Mean Thresholding', 'Adaptive Gaussian Thresholding']
images = [img2, th1, th2, th3]

for i in range(4):
    plt.subplot(2,2,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([],plt.yticks([]))
plt.show()
```



```
uploaded=files.upload()
```


No file chosen

Upload widget is only available when the cell has been

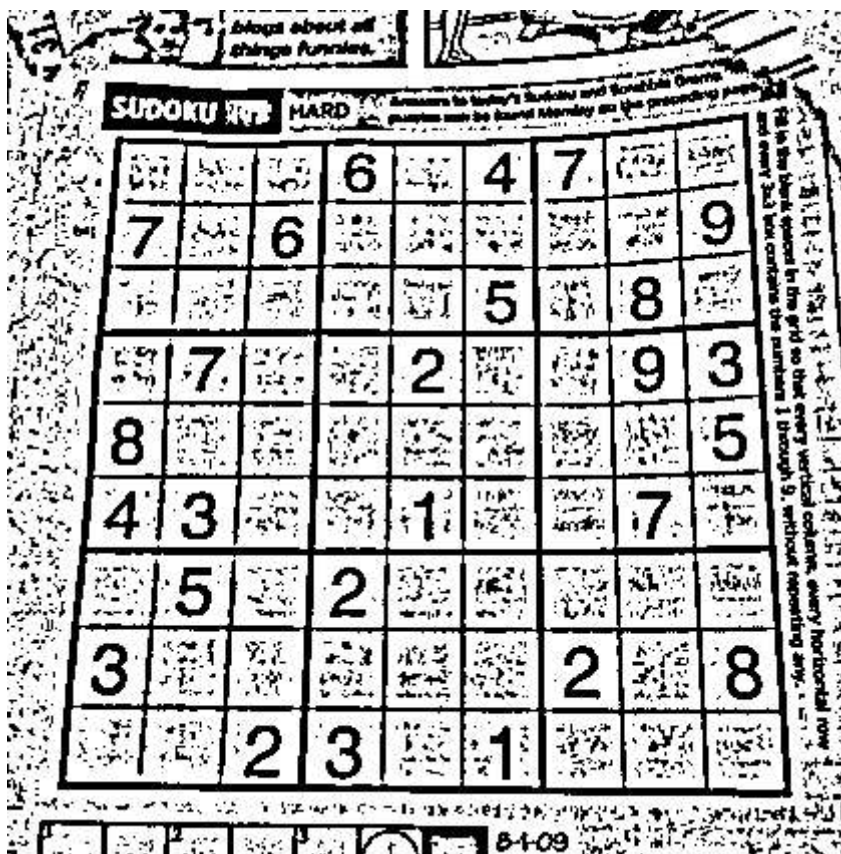
```
img3=cv2.imread('sudoku-original.jpg',0)
cv2_imshow(img3)
```



```
ret,th1 = cv2.threshold(img3,127,255,cv2.THRESH_BINARY)
cv2_imshow(th1)
```



```
th2 = cv2.adaptiveThreshold(img3,255,cv2.ADAPTIVE_THRESH_MEAN_C,\
                             cv2.THRESH_BINARY,11,2)
cv2_imshow(th2)
```



```
th3 = cv2.adaptiveThreshold(img3,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,\
                             cv2.THRESH_BINARY,11,2)
cv2_imshow(th3)
```



