

working-with-image-in-python-1

June 14, 2024

OpenCv Basics

```
[ ]: !git clone https://github.com/misbah4064/opencvTutorial.git  
%cd opencvTutorial/  
from IPython.display import clear_output  
clear_output()
```

```
[ ]:
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).

```
[51]: import cv2  
from google.colab.patches import cv2_imshow  
#colourful image  
image = cv2.imread("/content/61fcb7bc0c52e.jpg")  
cv2_imshow(image)
```



Image properties

```
[52]: print(image.shape)
```

(1080, 1920, 3)

Converting to Grayscale Image

```
[53]: gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

```
cv2_imshow(gray)
```



Now, Let's See the properties of the image

```
[54]: print(gray.shape)
```

(1080, 1920)

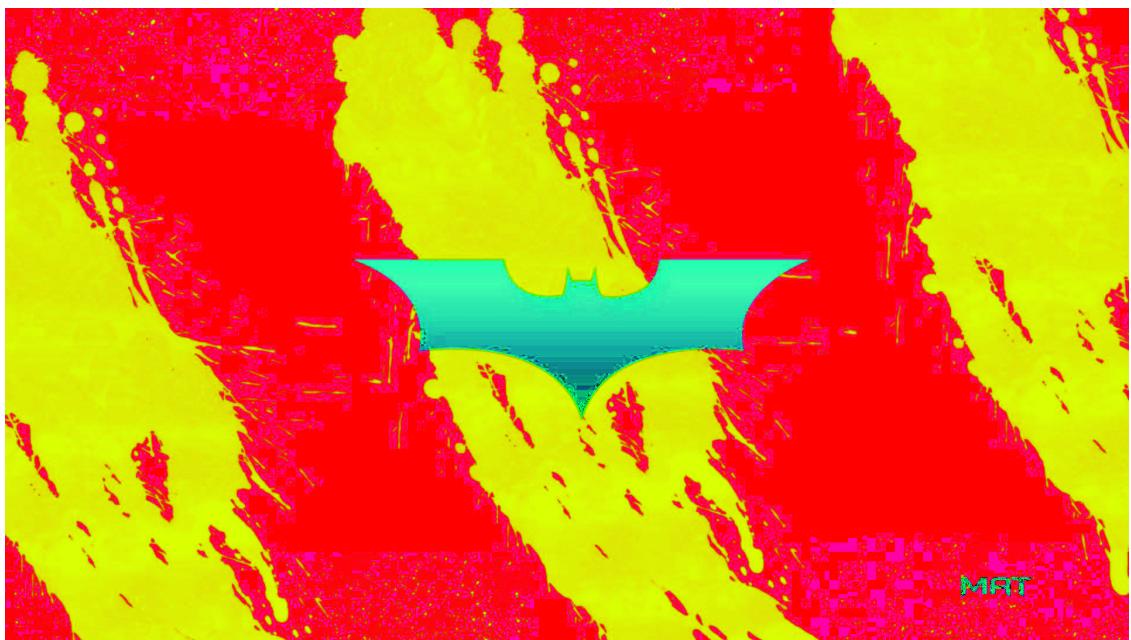
Now, if we compare the properties of the colourful to grayscale image we can see that the number of channels has decreased from 3 to 1.

Converting to HSV image.

```
[55]: # HSV Image
```

```
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```

```
cv2_imshow(hsv)
```



1 Edge Detection

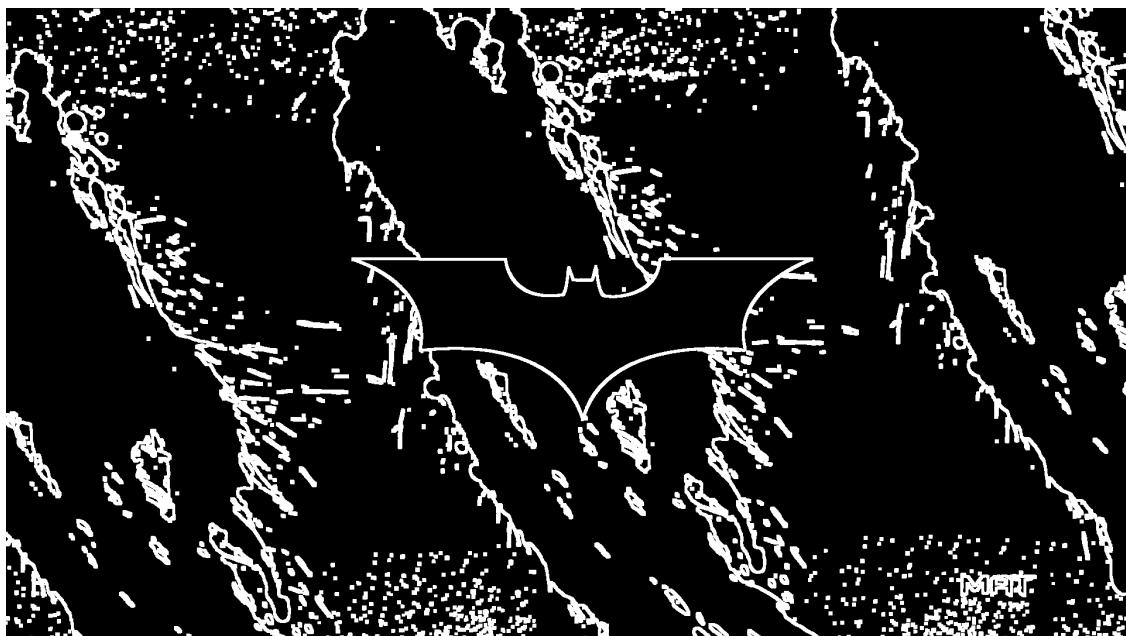
```
[56]: canny_image = cv2.Canny(gray,100,200)  
cv2_imshow(canny_image)
```

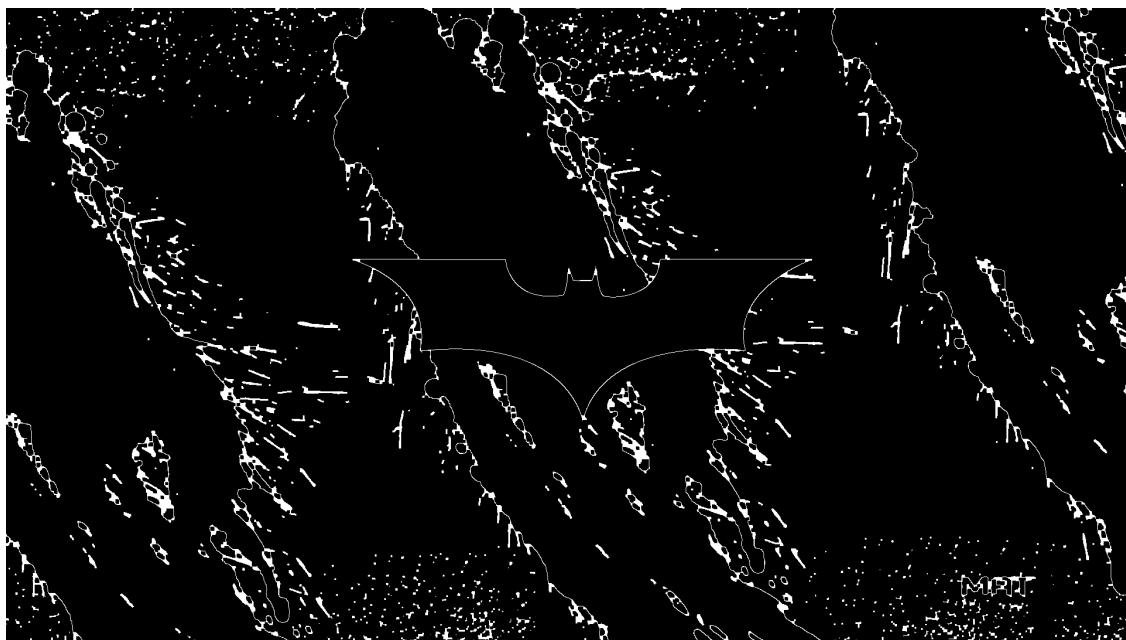


Erosion and Dilation

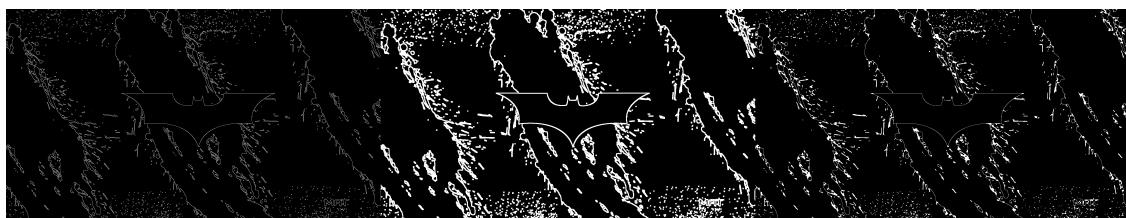
```
[57]: import numpy as np  
kernel = np.ones((2,2),np.uint8)
```

```
[58]: #Dilation  
kernel = np.ones((5,5),np.uint8)  
dilate_image = cv2.dilate(canny_image,kernel,iterations=1)  
  
cv2_imshow(dilate_image)  
  
erode_image = cv2.erode(dilate_image,kernel,iterations=1)  
cv2_imshow(erode_image)
```





```
[59]: display = np.hstack((canny_image,dilate_image,erode_image))
cv2_imshow(display)
```



2 Image Manipulation

```
[60]: lion_image = cv2.imread("/content/opencvTutorial/images/lion.jpg")
cv2_imshow(lion_image)
print(lion_image.shape)
```



(960, 640, 3)

De-Noising

```
[61]: dst = cv2.fastNlMeansDenoisingColored(lion_image, None, 20, 20, 10, 15)
display_n= np.hstack((lion_image,dst))
cv2.imshow(display_n)
```



[]: