

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
```

```
# Read the image
image1 = cv2.imread('day.jpg')
cv2_imshow(image1)
```



```
# Read the image
image2 = cv2.imread('night.jpg')
cv2_imshow(image2)
```



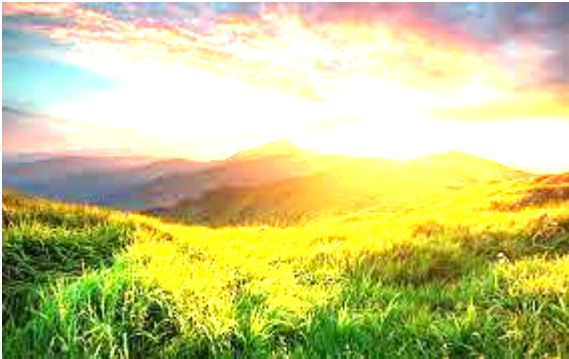
```
dst = cv2.addWeighted(image1, 0.7, image1, 0.6, 0)
cv2_imshow(dst)
```



```
# image1.shape
image2.shape
```



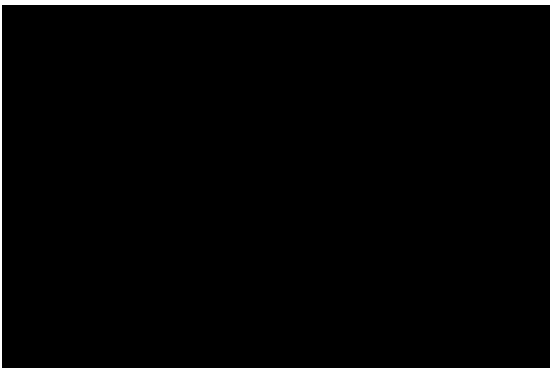
```
# Applying OpenCV addition on images  
fimg = cv2.add(image1, image1)  
cv2_imshow(fimg)
```



```
# Applying OpenCV addition on images  
fimg = cv2.add(image2, image2)  
cv2_imshow(fimg)
```



```
# Applying OpenCV addition on images  
fimg = cv2.subtract(image2, image2)  
cv2_imshow(fimg)
```



```
fimg = cv2.multiply(image1, 1.5)  
cv2_imshow(fimg)  
cv2_imshow(image1)
```





```
fimg = cv2.divide(image1, 1.5)
cv2_imshow(fimg)
cv2_imshow(image1)
```



```
rows, cols, channel = image1.shape
M = np.float32([[1, 0, 100], [0, 1, 50]])
dst = cv2.warpAffine(image1, M, (cols, rows))
dst1 = cv2.warpAffine(image1, M,(10,0))
cv2_imshow( dst)
cv2_imshow( dst1)
```





```
# Horizontal flip
rows, cols, channel = image1.shape
M = np.float32([[1, 0, 0], [0, -1, rows], [0, 0, 1]])
reflected_img = cv2.warpPerspective(image1, M, (int(cols), int(rows)))
cv2_imshow( image1)
print("\n-----\n")
cv2_imshow( reflected_img)
```





```
# Vertical flip
rows, cols, channel = image1.shape
M = np.float32([[-1, 0, cols], [0, 1, 0], [0, 0, 1]])
reflected_img = cv2.warpPerspective(image1, M, (int(cols), int(rows)))
cv2_imshow( image1)
print("\n-----\n")
cv2_imshow( reflected_img)
```



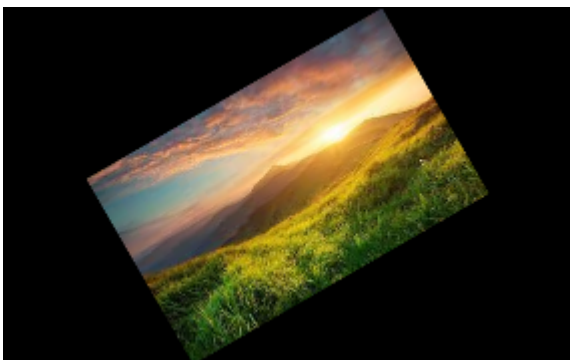
```
cv2.imshow('reflected_img')
```





```
# Image Rotation
rows, cols, channel = image1.shape
M = np.float32([[1, 0, 0], [0, -1, rows], [0, 0, 1]])
img_rotation = cv2.warpAffine(image1, cv2.getRotationMatrix2D((cols/2, rows/2), 30
cv2.imshow( image1)
print("\n-----\n")
cv2.imshow( img_rotation)
```





```
# Image Scaling : Shrinking
rows, cols, channel = image1.shape
img_shrunked = cv2.resize(image2, (250, 200),
                           interpolation=cv2.INTER_AREA)
cv2_imshow(img_shrunked)
img_enlarged = cv2.resize(image2, None,
                           fx=1.5, fy=1.5,
                           interpolation=cv2.INTER_CUBIC)
print("\n-----\n")
cv2_imshow(img_enlarged)
```



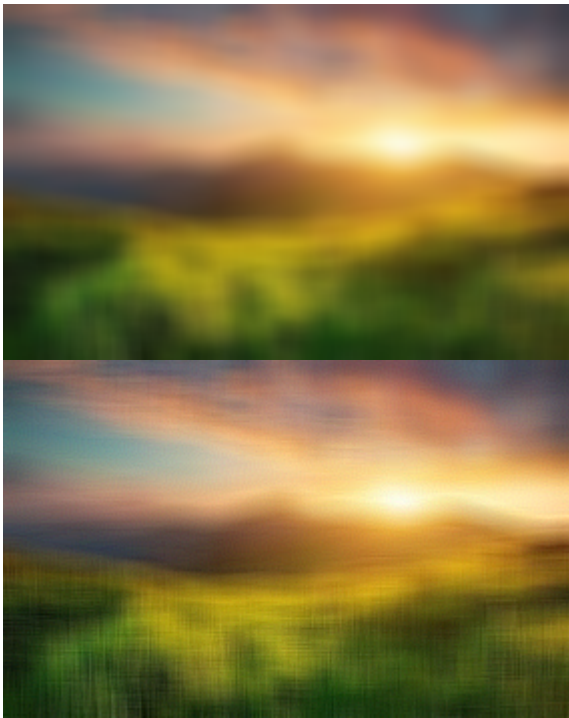


```
# Image Cropping
cropped_img = image1[100:300, 100:300]
cv2_imshow( cropped_img)
```



```
bluring = cv2.blur(image1, (15,15))
cv2_imshow(bluring)
```

```
sharpen_kernel = np.array([[ -1, -1, -1], [ -1, 9, -1], [ -1, -1, -1]])
sharpen = cv2.filter2D(blurimg, -1, sharpen_kernel)
cv2_imshow(sharpen)
```



```
from google.colab.patches import cv2_imshow

# Read the original image
img = cv2.imread('day.jpg')
# Display original image
cv2_imshow(img)
cv2.waitKey(0)

# Convert to grayscale
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Blur the image for better edge detection
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)

# Sobel Edge Detection
sobelx = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=0, ksize=5) # Sobel
sobely = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=0, dy=1, ksize=5) # Sobel
sobelxy = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5) # Comb
# Display Sobel Edge Detection Images
print("/nSobel X")
cv2_imshow(sobelx)
cv2.waitKey(0)

print("/nSobel Y")
cv2_imshow(sobely)
cv2.waitKey(0)

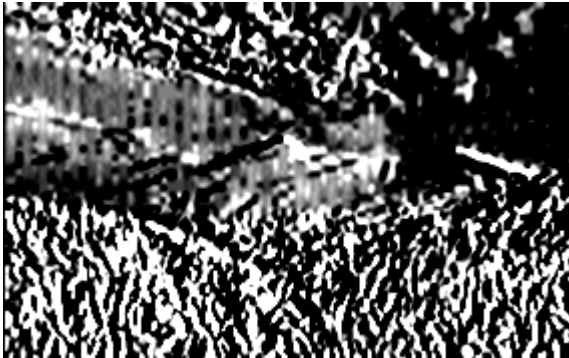
print("/nSobel X Y using Sobel() function")
cv2_imshow(sobelxy)
cv2.waitKey(0)
```

```
# Canny Edge Detection
edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200) # Canny Edge D
print("/nCanny Edge Detection")
# Display Canny Edge Detection Image
cv2_imshow(edges)
cv2.waitKey(0)

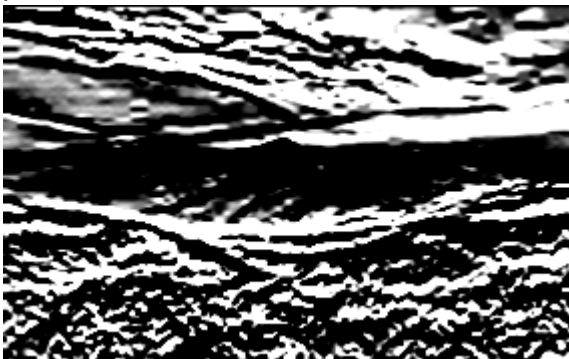
cv2.destroyAllWindows()
```



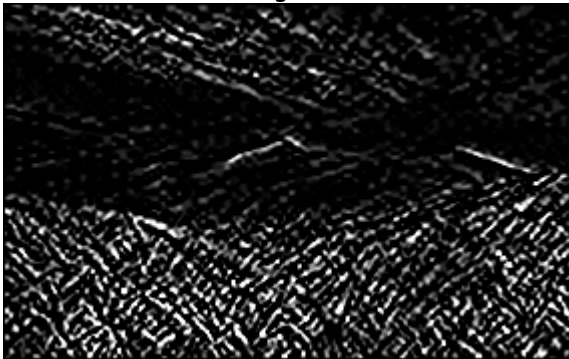
/nSobel X



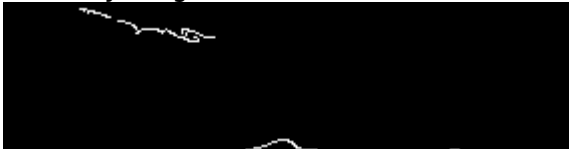
/nSobel Y



/nSobel X Y using Sobel() function



/nCanny Edge Detection





```
from google.colab.patches import cv2_imshow

# Read the original image
img = cv2.imread('night.jpg')
# Display original image
cv2_imshow(img)
cv2.waitKey(0)

# Convert to grayscale
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Blur the image for better edge detection
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)

# Sobel Edge Detection
sobelx = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=0, ksize=5) # Sobel
sobely = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=0, dy=1, ksize=5) # Sobel
sobelxy = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5) # Comb
# Display Sobel Edge Detection Images
print("/nSobel X")
cv2_imshow(sobelx)
cv2.waitKey(0)

print("/nSobel Y")
cv2_imshow(sobely)
cv2.waitKey(0)

print("/nSobel X Y using Sobel() function")
cv2_imshow(sobelxy)
cv2.waitKey(0)

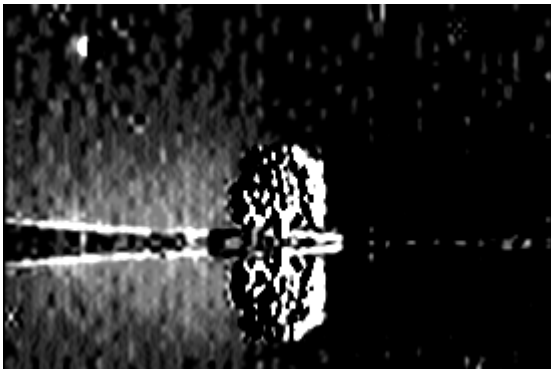
# Canny Edge Detection
edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200) # Canny Edge D
print("/nCanny Edge Detection")
# Display Canny Edge Detection Image
cv2_imshow(edges)
cv2.waitKey(0)

cv2.destroyAllWindows()
```





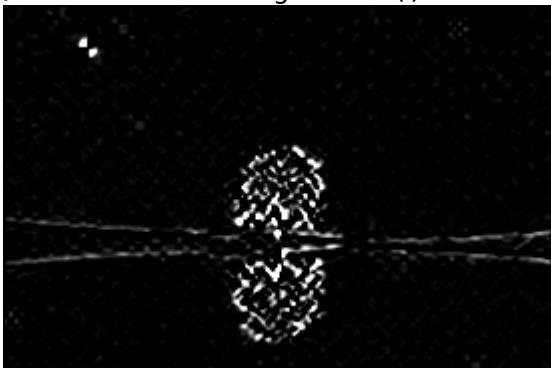
/nSobel X



/nSobel Y



/nSobel X Y using Sobel() function



/nCanny Edge Detection



```
from google.colab.patches import cv2_imshow
```

```
# Read the original image
```

```
img = cv2.imread('girl.jpg')
# Display original image
cv2_imshow(img)
cv2.waitKey(0)

# Convert to grayscale
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Blur the image for better edge detection
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)

# Sobel Edge Detection
sobelx = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=0, ksize=5) # Sobel
sobely = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=0, dy=1, ksize=5) # Sobel
sobelxy = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5) # Comb
# Display Sobel Edge Detection Images
print("/nSobel X")
cv2_imshow(sobelx)
cv2.waitKey(0)

print("/nSobel Y")
cv2_imshow(sobely)
cv2.waitKey(0)

print("/nSobel X Y using Sobel() function")
cv2_imshow(sobelxy)
cv2.waitKey(0)

# Canny Edge Detection
edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200) # Canny Edge D
print("/nCanny Edge Detection")
# Display Canny Edge Detection Image
cv2_imshow(edges)
cv2.waitKey(0)

cv2.destroyAllWindows()
```



/nSobel X





/nSobel Y



/nSobel X Y using Sobel() function



/nCanny Edge Detection





[Colab paid products](#) - [Cancel contracts here](#)