|  |
| --- |
| import cv2 2  import matplotlib.pyplot as plt  image\_path = ‘ your path'  image = cv2.imread(image\_path)  ksize = (15, 15) sigmaX = 0  blurred\_image=cv2.GaussianBlur(image,  ksize,sigmaX)  plt.figure(figsize=(10, 5))  plt.subplot(1, 2, 1)  plt.title('Original Image')  plt.imshow(cv2.cvtColor(image,  cv2.COLOR\_BGR2RGB))  plt.axis('off')  plt.subplot(1, 2, 2)  plt.title('Blurred Image')  plt.imshow(cv2.cvtColor(blurred\_image,  cv2.COLOR\_BGR2RGB))  plt.axis('off') plt.show()  blurred\_image\_path = 'path’  cv2.imwrite(blurred\_image\_path,blurred  \_image) |

|  |
| --- |
| import cv2 4  import numpy as np  import matplotlib.pyplot as plt  def rotate\_image(image, angle):  (h,w) = image.shape[:2]  center =(w//2,h//2)  M=cv2.getRotationMatrix2D(center,angle,  1.0)  rotated = cv2.warpAffine(image,M,(w, h))  return rotated  image\_path ='path’  image = cv2.imread(image\_path)  angle =45  rotated\_image=rotate\_image(image,angle)  plt.figure(figsize=(10, 5))  plt.subplot(1, 2, 1)  plt.title('Original Image')  plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))plt.axis('off')  plt.subplot(1, 2, 2)  plt.title('Rotated Image')  plt.imshow(cv2.cvtColor(rotated\_image,  cv2.COLOR\_BGR2RGB))  plt.axis('off') plt.show()  rotated\_image\_path = 'path'  cv2.imwrite(rotated\_image\_path,  rotated\_image) |

|  |
| --- |
| import cv2 1  import matplotlib.pyplot as plt  image\_path = 'path\_to\_your\_image.jpg' colored\_image = cv2.imread(image\_path)  grayscale\_image = cv2.cvtColor(colored  \_image, cv2.COLOR\_BGR2GRAY)  plt.figure(figsize=(10, 5))  plt.subplot(1, 2, 1)  plt.title('Colored Image')  plt.imshow(cv2.cvtColor(colored\_image,  cv2.COLOR\_BGR2RGB))  plt.axis('off')  plt.subplot(1, 2, 2)  plt.title('Grayscale Image')  plt.imshow(grayscale\_image,cmap='gray')  plt.axis('off') plt.show()  grayscale\_image\_path='grayscale\_image. cv2.imwrite(grayscale\_image\_path,  grayscale\_image) |

|  |
| --- |
| import cv2 import numpy as np 5  import matplotlib.pyplot as plt  image\_path=‘path’ image = cv2.imread(image\_path,cv2.IMREAD\_GRAYSCALE)  kernel\_size=(5,5)  kernel=np.ones(kernel\_size,np.uint8)  dilated\_image=cv2.dilate(image,kernel,  iterations=1)  eroded\_image = cv2.erode(image, kernel, iterations=1)plt.figure(figsize=(15, 5))  plt.subplot(1, 3, 1)  plt.title('Original Image')  plt.imshow(image, cmap='gray')  plt.axis('off')  plt.subplot(1, 3, 2)  plt.title('Dilated Image')  plt.imshow(dilated\_image, cmap='gray')  plt.axis('off')  plt.subplot(1, 3, 3)  plt.title('Eroded Image')  plt.imshow(eroded\_image, cmap='gray')  plt.axis('off') plt.show()  dilated\_image\_path = 'path,  eroded\_image\_path =’path,  cv2.imwrite(dilated\_image\_path,dilated\_image)  cv2.imwrite(eroded\_image\_path,eroded\_image) |

|  |
| --- |
| import cv2 3  image\_path = 'path’  image = cv2.imread(image\_path, 0)  edges = cv2.Canny(image, 100, 200)  cv2.imshow('Edges', edges)  cv2.waitKey(0)  cv2.destroyAllWindows() |