import cv2

import numpy as np

import matplotlib.pyplot as plt

from google.colab.patches import cv2\_imshow

img1 = cv2.imread('./image1.jpeg', cv2.IMREAD\_GRAYSCALE)

kernel1 = np.array([[1,-1,1],

[-1,1,-1],

[1,-1,1]])

img1\_kernel = cv2.filter2D(img1, -1, kernel1)

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title("Original Image")

plt.imshow(img1, cmap='gray')

plt.subplot(1, 2, 2)

plt.title("2D Convolution")

plt.imshow(img1\_kernel, cmap='gray')

plt.show()

img2 = cv2.imread('./image2.jpeg', cv2.IMREAD\_GRAYSCALE)

kernel2 = np.array([[0, -1, 0],

[-1, 5, -1],

[0, -1, 0]])

img2\_kernel = cv2.filter2D(img2, -1, kernel2)

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title("Original Image")

plt.imshow(img2, cmap='gray')

plt.subplot(1, 2, 2)

plt.title("Sharpen 2D Convolution")

plt.imshow(img2\_kernel, cmap='gray')

plt.show()

img3 = cv2.imread('./image3.jpeg', cv2.IMREAD\_GRAYSCALE)

kernel3 = np.array([[0, -1, 0],

[-1, 5, -1],

[0, -1, 0]])

img3\_kernel = cv2.filter2D(img3, -1, kernel3)

zero\_pad = cv2.copyMakeBorder(img3, 100, 100, 100, 100, cv2.BORDER\_CONSTANT, value=[0, 0, 0])

reflect\_pad = cv2.copyMakeBorder(img3, 100, 100, 100, 100, cv2.BORDER\_REFLECT)

same\_pad = cv2.copyMakeBorder(img3, 1, 1, 1, 1, cv2.BORDER\_REFLECT)

plt.figure(figsize=(20, 12))

plt.subplot(1, 4, 1)

plt.title("Original Image")

plt.imshow(img3, cmap='gray')

plt.subplot(1, 4, 2)

plt.title("Zero pad")

plt.imshow(zero\_pad, cmap='gray')

plt.subplot(1, 4, 3)

plt.title("Reflect pad")

plt.imshow(reflect\_pad, cmap='gray')

plt.subplot(1, 4, 4)

plt.title("Same pad")

plt.imshow(same\_pad, cmap='gray')

plt.show()

img4 = cv2.imread('./image4.jpeg', cv2.IMREAD\_COLOR)

blur\_img = cv2.blur(img4, (5, 5))

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title("Original Image")

img4 = cv2.cvtColor(img4, cv2.COLOR\_BGR2RGB)

plt.imshow(img4)

plt.subplot(1, 2, 2)

plt.title("Gaussian Blur")

blur\_img = cv2.cvtColor(blur\_img, cv2.COLOR\_BGR2RGB)

plt.imshow(blur\_img)

plt.show()

gaussian\_img1 = cv2.GaussianBlur(blur\_img, (5, 5), 0)

gaussian\_img2 = cv2.GaussianBlur(blur\_img, (5, 5), 10)

gaussian\_img3 = cv2.GaussianBlur(blur\_img, (5, 5), 50)

gaussian\_img4 = cv2.GaussianBlur(blur\_img, (5, 5), 100)

plt.figure(figsize=(20, 12))

plt.subplot(1, 4, 1)

plt.title("Sigma 0")

plt.imshow(gaussian\_img1)

plt.subplot(1, 4, 2)

plt.title("Sigma 10")

plt.imshow(gaussian\_img2)

plt.subplot(1, 4, 3)

plt.title("Sigma 50")

plt.imshow(gaussian\_img3)

plt.subplot(1, 4, 4)

plt.title("Sigma 100")

plt.imshow(gaussian\_img4)

plt.show()

img5 = cv2.imread('./image5.jpeg', cv2.IMREAD\_GRAYSCALE)

laplacian\_img = cv2.Laplacian(img5, cv2.CV\_64F)

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title("Original Image")

plt.imshow(img5, cmap='gray')

plt.subplot(1, 2, 2)

plt.title("Laplacian")

plt.imshow(laplacian\_img, cmap='gray')

plt.show()

img6 = cv2.imread('./image6.jpeg', cv2.IMREAD\_GRAYSCALE)

h\_kernel = np.array([[-1,0,1],

[-1,0,1],

[-1,0,1]])

v\_kernel = np.array([[-1,-1,1],

[0,0,0],

[1,1,1]])

img6\_h = cv2.filter2D(img6, -1, h\_kernel)

img6\_v = cv2.filter2D(img6, -1, v\_kernel)

plt.figure(figsize=(20, 12))

plt.subplot(1, 3, 1)

plt.title("Original")

plt.imshow(img6, cmap='gray')

plt.subplot(1, 3, 2)

plt.title("Horizental")

plt.imshow(img6\_h, cmap='gray')

plt.subplot(1, 3, 3)

plt.title("Vertical")

plt.imshow(img6\_v, cmap='gray')

plt.show()

img7 = cv2.imread('./image7.jpg', cv2.IMREAD\_GRAYSCALE)

equalizer\_img = cv2.equalizeHist(img7)

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.title("Original Image")

plt.imshow(img7, cmap='gray')

plt.subplot(1, 2, 2)

plt.title("Histogram equalizer")

plt.imshow(equalizer\_img, cmap='gray')

plt.show()

equalizer\_img1 = cv2.equalizeHist(img7)

equalizer\_img2 = cv2.equalizeHist(equalizer\_img1)

equalizer\_img3 = cv2.equalizeHist(equalizer\_img2)

plt.figure(figsize=(20, 12))

plt.subplot(1, 4, 1)

plt.title("Original")

plt.imshow(img7, cmap='gray')

plt.subplot(1, 4, 2)

plt.title("1st")

plt.imshow(equalizer\_img1, cmap='gray')

plt.subplot(1, 4, 3)

plt.title("2nd")

plt.imshow(equalizer\_img2, cmap='gray')

plt.subplot(1, 4, 4)

plt.title("3rd")

plt.imshow(equalizer\_img3, cmap='gray')

plt.show()



