**BÀI TẬP THỰC HÀNH 01  
MÔN HỌC SÂU (DEEP LEARNING)**

Họ và tên: **Hà Minh Đức**

Mã sinh viên: DTC1954802010077

Lớp: CNTT K18K

**Bài tập 01:**

# 1. Xây dựng chương trình thực hiện phép tích chập (Convolution) trong ảnh.

**Code:**

"""

Created on Wed Mar 1 16:15:13 2023

@author: Admin

"""

# 1. Xây dựng chương trình thực hiện phép tích chập (Convolution) trong ảnh.

import cv2

import numpy as np

image = cv2.imread('../img/img2.jpg')

# Print error message if image is null

if image is None:

print('Could not read image')

# scale

scale\_percent = 60

w = int(image.shape[1] \* scale\_percent / 100)

h = int(image.shape[0] \* scale\_percent / 100)

dim = (w, h)

# resize image

image = cv2.resize(image, dim, interpolation = cv2.INTER\_AREA)

# Apply identity kernel

kernel1 = np.array([[0, 0, 0],

[0, 1, 0],

[0, 0, 0]])

identity = cv2.filter2D(src=image, ddepth=-1, kernel=kernel1)

cv2.imshow('Original', image)

cv2.imshow('Identity', identity)

cv2.waitKey()

cv2.imwrite('../imgExported/identity.jpg', identity)

cv2.destroyAllWindows()

# Apply blurring kernel

kernel2 = np.ones((5, 5), np.float32) / 25

img = cv2.filter2D(src=image, ddepth=-1, kernel=kernel2)

cv2.imshow('Original', image)

cv2.imshow('Kernel Blur', img)

cv2.waitKey()

cv2.imwrite('../imgExported/blur\_kernel.jpg', img)

cv2.destroyAllWindows()

# Laplacian

kernel3 = np.array((

[0, 1, 0],

[1, -4, 1],

[0, 1, 0]), dtype="int")

img = cv2.filter2D(src=image, ddepth=-1, kernel=kernel3)

cv2.imshow('Original', image)

cv2.imshow('Kernel Laplacian', img)

cv2.waitKey()

cv2.imwrite('../imgExported/Laplacian.jpg', img)

cv2.destroyAllWindows()

# Sharpen

kernel4 = np.array((

[0, -1, 0],

[-1, 5, -1],

[0, -1, 0]), dtype="int")

img = cv2.filter2D(src=image, ddepth=-1, kernel=kernel4)

cv2.imshow('Original', image)

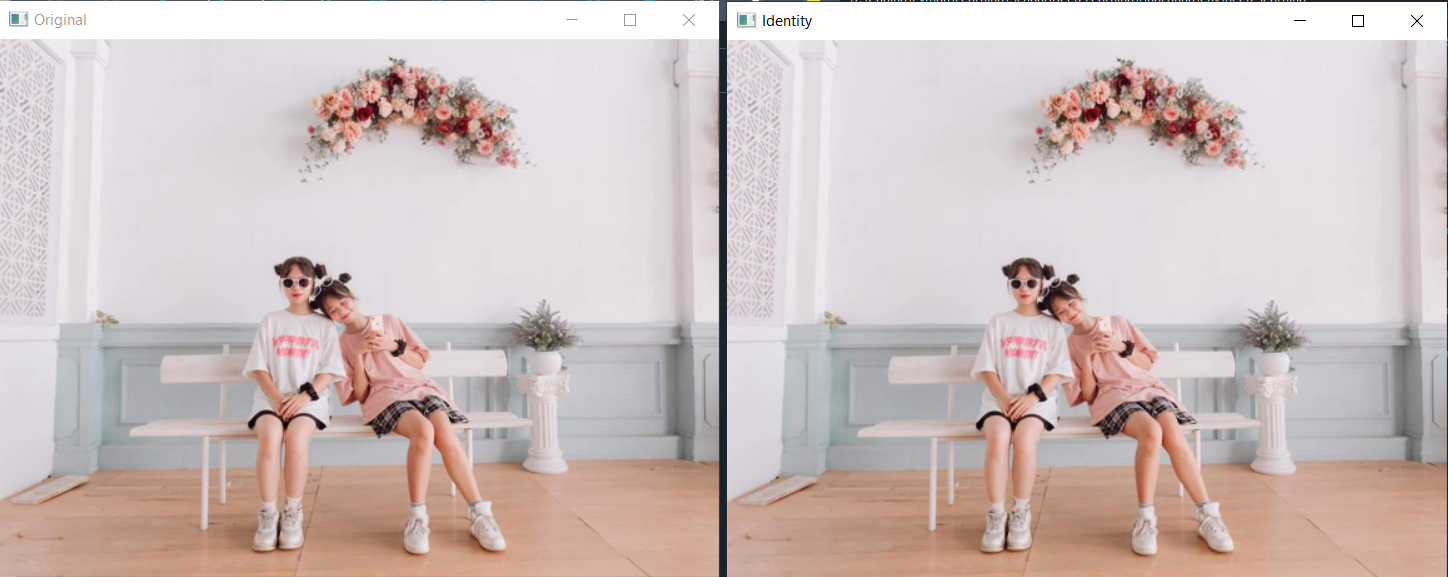
cv2.imshow('Sharpen Laplacian', img)

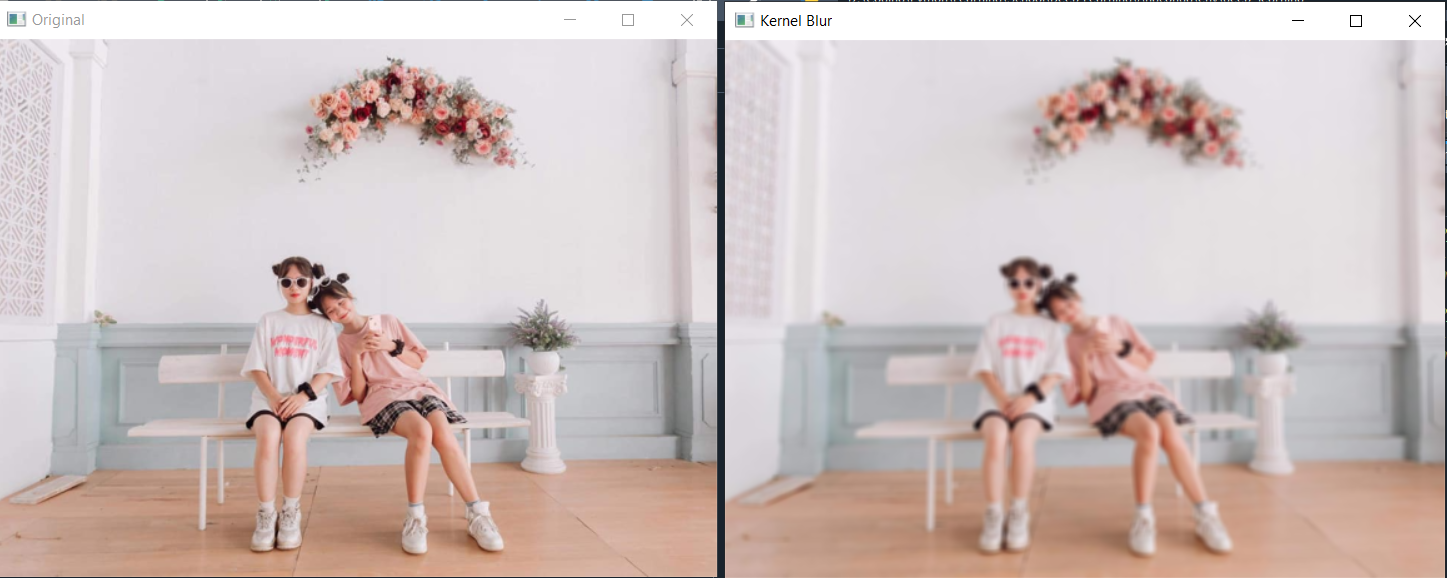
cv2.waitKey()

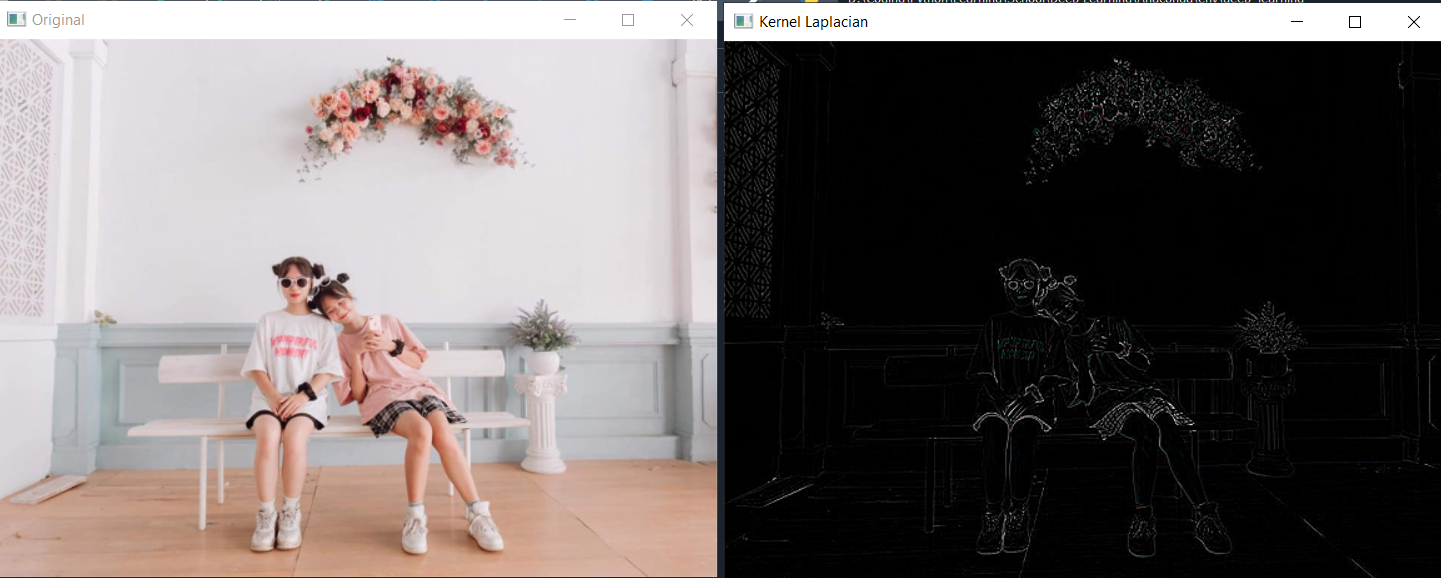
cv2.imwrite('../imgExported/Sharpen.jpg', img)

cv2.destroyAllWindows()

**Kết quả:**









**Bài tập 02:**

# 2. Xây dựng chương trình thực hiện đệm (Padding) trong ảnh.

**Code:**

# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 1 16:40:04 2023

@author: Admin

"""

# 2. Xây dựng chương trình thực hiện đệm (Padding) trong ảnh.

import cv2

import numpy as np

image = cv2.imread('../img/img2.jpg')

# Print error message if image is null

if image is None:

print('Could not read image')

# scale

scale\_percent = 60

w = int(image.shape[1] \* scale\_percent / 100)

h = int(image.shape[0] \* scale\_percent / 100)

dim = (w, h)

# resize image

image = cv2.resize(image, dim, interpolation = cv2.INTER\_AREA)

# using copyMakeBorder(src, top, bottom, left, right, borderType = cv2.BORDER\_CONSTANT, cv2.BORDER\_REFLECT, value = rgb())

imageHasBorder = cv2.copyMakeBorder(image, 10, 10, 10, 10, cv2.BORDER\_CONSTANT, None, value = (0, 255, 255))

cv2.imshow('Original', image)

cv2.imshow('image has border', imageHasBorder)

cv2.waitKey()

cv2.imwrite('../imgExported/imageHasBorder.jpg', imageHasBorder)

cv2.destroyAllWindows()

# using copyMakeBorder - BORDER\_CONSTANT

imageHasBorderRef = cv2.copyMakeBorder(image, 100, 100, 50, 50, cv2.BORDER\_REFLECT)

cv2.imshow('Original', image)

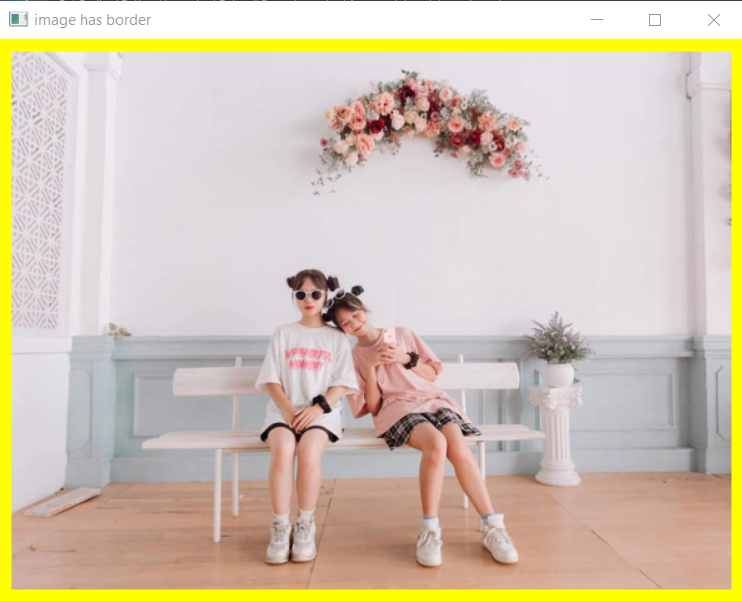
cv2.imshow('image has border reflect', imageHasBorderRef)

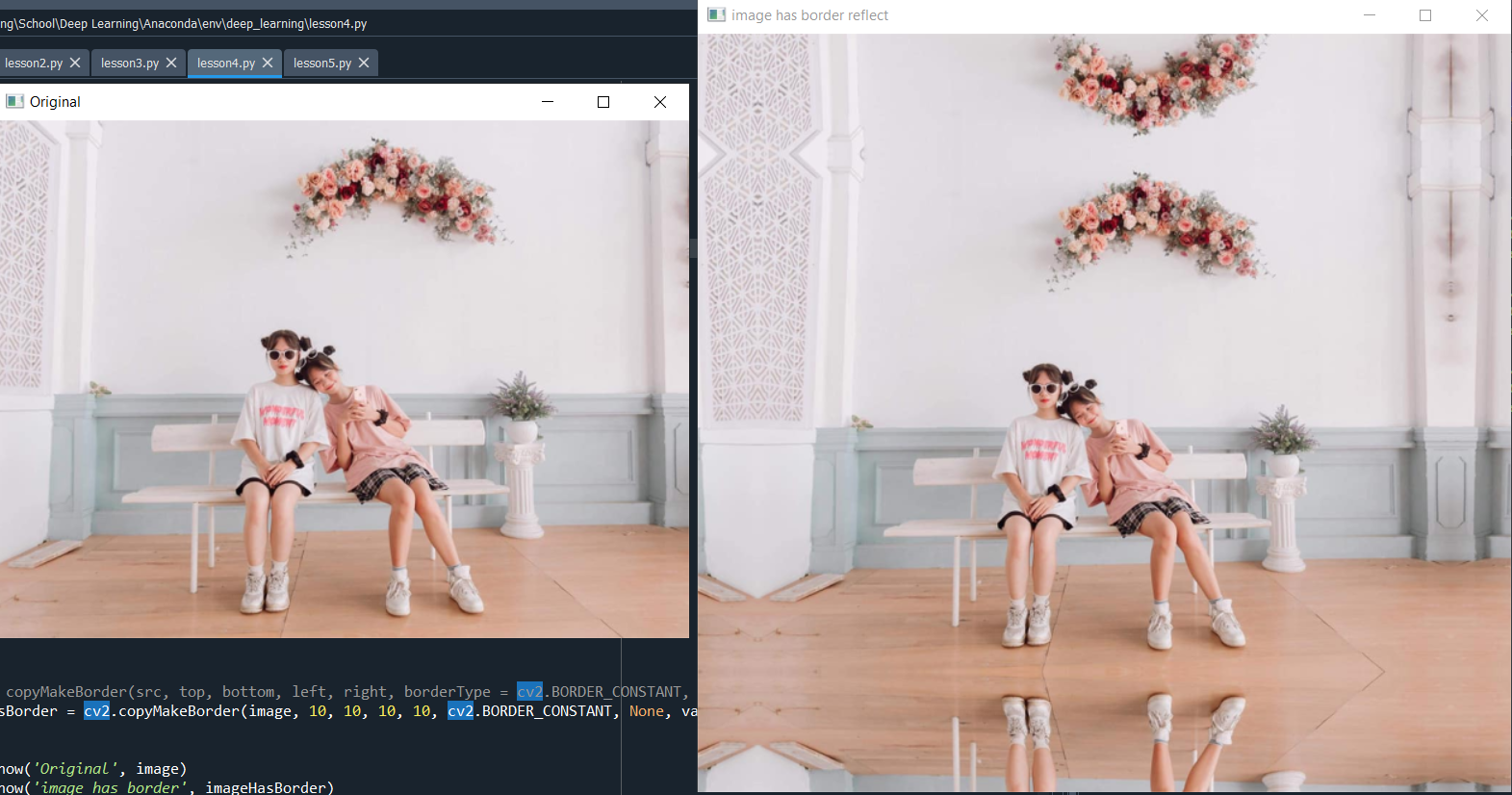
cv2.waitKey()

cv2.imwrite('../imgExported/imageHasBorder.jpg', imageHasBorderRef)

cv2.destroyAllWindows()

**Kết quả:**





**Bài tập 03:**

# 3. Xây dựng chương trình thực hiện sải bước (Stride) trong ảnh.

**Code:**

# -\*- coding: utf-8 -\*-

"""

Created on Wed Mar 1 16:54:49 2023

@author: Admin

"""

# 3. Xây dựng chương trình thực hiện sải bước (Stride) trong ảnh.

import cv2

import numpy as np

# func

def apply\_sliding\_window(img, kernel, padding=0, stride=1):

h, w = img.shape[:2]

img\_p = np.zeros([h+2\*padding, w+2\*padding])

img\_p[padding:padding+h, padding:padding+w] = img

kernel = np.array(kernel)

assert len(kernel.shape) == 2 and kernel.shape[0] == kernel.shape[1] # square kernel

assert kernel.shape[0] % 2 != 0 # kernel size is odd number

k\_size = kernel.shape[0]

k\_half = int(k\_size/2)

y\_pos = [v for idx, v in enumerate(list(range(k\_half, h-k\_half))) if idx % stride == 0]

x\_pos = [v for idx, v in enumerate(list(range(k\_half, w-k\_half))) if idx % stride == 0]

new\_img = np.zeros([len(y\_pos), len(x\_pos)])

for new\_y, y in enumerate(y\_pos):

for new\_x, x in enumerate(x\_pos):

if k\_half == 0:

pixel\_val = img\_p[y, x] \* kernel # element-wise multiply

else:

pixel\_val = np.sum(img\_p[y-k\_half:y-k\_half+k\_size, x-k\_half:x-k\_half+k\_size] \* kernel) # dot product: https://minhng.info/toan-hoc/y-nghia-tich-vo-huong.html

new\_img[new\_y, new\_x] = pixel\_val

return new\_img

def apply\_sliding\_window\_on\_3\_channels(img, kernel, padding=0, stride=1):

layer\_blue = apply\_sliding\_window(img[:,:,0], kernel, padding, stride)

layer\_green = apply\_sliding\_window(img[:,:,1], kernel, padding, stride)

layer\_red = apply\_sliding\_window(img[:,:,2], kernel, padding, stride)

new\_img = np.zeros(list(layer\_blue.shape) + [3])

new\_img[:,:,0], new\_img[:,:,1], new\_img[:,:,2] = layer\_blue, layer\_green, layer\_red

return new\_img

# import image

image = cv2.imread('../img/img2.jpg')

# Print error message if image is null

if image is None:

print('Could not read image')

new\_img = apply\_sliding\_window\_on\_3\_channels(image, kernel=[[1]], padding=0, stride=2)

cv2.imwrite('../imgExported/imgNew.jpg', new\_img)

lighten\_blur\_img = apply\_sliding\_window\_on\_3\_channels(image, kernel=[[0.33, 0.33, 0.33], [0.33, 0.33, 0.33], [0.33, 0.33, 0.33]], padding=1, stride=1)

cv2.imwrite('../imgExported/imagelightenBlur.jpg', lighten\_blur\_img)

print('Success!')

cv2.waitKey()

**Kết quả:**



imagelightenBlur



**Bài tập 04:**

4. Xây dựng chương trình thực hiện gộp (Pooling) trong ảnh

**Code:**

.

…