#### 试验内容：OpenCV常用与项目有关模块函数的选择和说明

1. cv2
2. Cv2.imread and cv2.imshow

Sample:

import cv2  
import matplotlib.pyplot as plot

image = cv2.imread('digivisual.png') #load company image  
plot.imshow(image) #show image

1. Change color spaces

Sample code:

import cv2import matplotlib.pyplot as plt

# Reading the image

original = cv2.imread('digivisual.jpg')

fig = plt.figure(figsize=(8, 2))

axarr = fig.subplots(1, 3)

# Change color space

image = cv2.cvtColor(original, cv2.COLOR\_BGR2RGB)

axarr[0].imshow(image)

image = cv2.cvtColor(original, cv2.COLOR\_BGR2GRAY)

axarr[1].imshow(image)

image = cv2.cvtColor(original, cv2.COLOR\_BGR2LAB)

axarr[2].imshow(image)

plt.show()

## 四：resize image

Sample:

import cv2

# Reading digivisual the image

original = cv2.imread('digivisual.jpg')

# Resize

resized = cv2.resize(original, (200, 200))

print(original.shape)

print(resized.shape)

五：Image rotation

cv2.rotate(image, cv2.ROTATE\_90\_CLOCKWISE)

To have more control over the rotation we can use getRotationMatrix2D and warpAffine instead.

六：Edge Detection

Sample code:

import cv2

import matplotlib.pyplot as plt

# Reading the image

dvt1 = cv2.imread('digivisual.jpg')

fig = plt.figure(figsize=(6, 2))

axarr = fig.subplots(1, 2)

axarr[0].imshow(cv2.cvtColor(original, cv2.COLOR\_BGR2RGB))

threshold1 = 50

threshold2 = 200

grey = cv2.cvtColor(dvt1, cv2.COLOR\_BGR2GRAY)

image = cv2.Canny(dvt1, threshold1, threshold2)

axarr[1].imshow(image, cmap='gray')

plt.show(）

七：Smoothing Techniques

cv2.blur()

cv2.medianBlur()

cv2.GaussianBlur()

Sample code:

img = cv2.imread('digivisual.jpg')

Gaussian\_blur = cv2.GaussianBlur(img,(5,5),0)

cv2\_imshow(median\_blur)

cv2\_imshow(img)

八：Morphological Techniques

### Erosion sample code:

mg = cv2.imread('digivisual.jpg')

kernel = np.ones((5,5),np.uint8)

img\_erosion = cv2.erode(img, kernel, iterations=1)

img\_dilation = cv2.dilate(img, kernel, iterations=1)

cv2\_imshow(img)

cv2\_imshow(img\_erosion)

Dilation sample code:

technique removes the boundary pixels but in Dilation. It adds the additional pixels to the input. It is used when the pixels are missing in the image. In general practices, we apply erosion to shrink the image to remove noises, and then by applying the dilation, there will be no loss of pixels.

img = cv2.imread('digivisual.png')

kernel = np.ones((5,5),np.uint8)

img\_dilation = cv2.dilate(img, kernel, iterations=1)

cv2\_imshow(img)

cv2\_imshow(img\_dilation)