Ton Duc Thang University
Faculty of Information Technology

505060 - Digital Image Processing

Anh H. Vo
vohoanganh@tdtu.edu.vn

August 30, 2020

1 Objectives

In this laboratory, student will install the popular libraries in image processing. In addition, they are able to use the basic functions in image processing based on the simple examples.

2 How to install and set environment variables

- Install python libraries for image processing:

```
sudo apt-get install python-pip
pip install opencv_python-3.0.0-cp34-none-win_amd64.whl
or
pip install opencv-python
pip install python3-numpy python3-scipy python3-matplotlib
pip install pillow
pip install pytesseract
```

- Go to **Window Power-Shell** and run the commands to test installed libraries.

```
import cv2 as cv
import numpy as np
import matplotlib
from scipy.misc import imread, imsave


print(cv.__version__)
```

- Install opencv library for **Android Studio**: *See* this link
- Install opencv library on **Raspberry Pi 3 B+**: *See* this link

3 The basic of functions in Image Processing

3.1 Read, show, save an image

In this section, we introduce the different libraries to read an image from your drive.

Listing 1: Read an image and show on screen with Opencv library

```
import cv2 as cv


img = cv.imread('lena.jpg')
cv2.imshow('First␣example', img)
cv2.waitKey(0)
```

Note: Opencv library:

- To read an image: **cv2.imread('file path')** # numpy.ndarray
- To show an image: **cv2.imshow('Information', image)**
- To save an image: **cv2.imwrite('file name', image)**

Listing 2: Read an image and show on screen with Scipy library

```
from scipy.misc import imread, imshow


img = imread('lena.jpg')
imshow(img)
```

Note: Scipy library:

- To read an image: **scipy.misc.imread('file path')** # numpy.ndarray
- To show an image: **scipy.misc.imshow(image)**
- To save an image: **scipy.misc.imsave('file name', image)**

Listing 3: Read an image and show on screen with Matplotlib library

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

plt.figure()
img = mpimg.imread('lena.jpg')
plt.imshow(img)
plt.title('First example')
plt.show()
```

Note: **matplotlib** library:

- To read an image: **matplotlib.image.imread('path')** # numpy.ndarray
- To show an image: **matplotlib.pyplot.imshow(image)**
- To save an image: **matplotlib.pyplot.savefig('file name')**

Listing 4: Read an image and show on screen with Pillow library

```
from PIL import Image

# PIL.JpegImagePlugin.JpegImageFile
image = Image.open('lena.jpg')
image.show()
```

Note: **Pillow** library:

- To read an image: **PIL.Image.open('file path')**
- To show an image: **image.show()**
- To save an image: **image.save('file name','image type')**
- To convert PIL.Image to numpy.ndarray: **img = np.asarray(image)**
- To convert numpy.ndarray to PIL.Image: **Image.fromarray(img)**

3.2 How to compute the histogram of an image

In the first example, we use **np.histogram** function from **numpy** library to calculate the histogram of an image based on the number of bin (e.g [0, 255])

Listing 5: An example calculate the histogram of an image and show on screen with Opencv and Matplotlib libraries

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

img = cv2.imread('lena.jpg')
## img.ravel() flattened matrix to 1D vector
hist, bin = np.histogram(img.ravel(), 256, [0, 256])

plt.hist(hist)
plt.show()
```

Note: np.histogram function:

- 1D vector
- The number of bin
- The range of bin

In the second example, we could use another function as **cv2.calcHist** function in **Opencv** library instead of **np.histogram** as this previous example

Listing 6: An example uses cv2.calcHist function to calculate the histogram of an input image

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

img = cv2.imread('lena.jpg')
color = ('b','g','r')
for i,col in enumerate(color):
        histr = cv2.calcHist([img],[i],None,[256],[0,256])
        plt.plot(histr,color = col)
        plt.xlim([0,256])
plt.show()
```

In the last example, **Image.histogram** function of **PIL** library could be used to compute the histogram of an image, and **pyplot.bar** is used to show the histogram result on the screen.

Listing 7: An example uses the histogram function of Pillow library

```
import matplotlib.pyplot as plt
from PIL import Image

def getRed(redVal):
    return '#%02x%02x%02x' % (redVal, 0, 0)

image = Image.open("lena.jpg")
histogram = image.histogram()
# Take only the Red counts
l1 = histogram[0:256]
# Take only the Blue counts
l2 = histogram[256:512]

# Take only the Green counts
```

```
l3 = histogram[512:768]
plt.figure(0)
# Red histogram
for i in range(0, 256):
        plt.bar(i, l1[i], color = getRed(i),
                         edgecolor=getRed(i), alpha=0.3)

plt.show()
```

Question:

1. Write your own codes to show the histogram of the blue channel.
2. Write your own codes to show the histogram of the green channel.

3.3 The basic operations on images

3.3.1 Drawing on Images

Listing 8: An example draw on image of Pillow library

```
from PIL import Image, ImageDraw

blank_image = Image.new('RGBA', (400, 300), 'white')
img_draw = ImageDraw.Draw(blank_image)
img_draw.rectangle((70, 50, 270, 200), outline='red',
                                     fill='blue')
img_draw.text((70, 250), 'Hello␣World', fill='green')
blank_image.save('drawn_image.jpg')
```

Note: Pillow library:

- To create an new image: **PIL.Image.new('file name', (w, h), 'white')**
- To draw an image: **PIL.Image.draw(image)**
- To draw a rectange on an image: **image.rectangle((x,y,w,h), outline=color, fill=color)**
- To draw a text on an image: **image.text((x,y),'Information', fill=color)**

Listing 9: An example draw on image of Opencv library

```
import cv2
img = cv2.imread('the_scream.jpg')
blank = np.zeros((200,200,3),np.uint8)
cv2.rectangle(blank,(20,20),(50,50),(255,0,0),3)
```

```
cv2.putText(blank,'Hello world',(20,30),
      cv2.FONT_HERSHEY_SIMPLEX,1,(0,255,0))
cv2.imshow('Hello ',blank)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

> **Note: Opencv** library:
>
> - To create an new image: **np.zeros((height, width, 3), np.uint8)**
> - To draw a rectange on an image: **cv2.rectangle(image, (top, left), (bottom, right), (R, G, B), 3)**
> - To draw a text on an image: **cv2.putText(image, 'Information', (x, y), font, fontscale, (R, G, B), thickness, cv2.LINE_AA)**

3.3.2 Pasting an Image onto Another Image

Listing 10: An example to past one image to another image of of Pillow library

```
from PIL import Image

image = Image.open('unsplash_01.jpg')
logo = Image.open('logo.png')
image_copy = image.copy()
position = ((image_copy.width - logo.width),
            (image_copy.height - logo.height))
image_copy.paste(logo, position)
image_copy.save('pasted_image.jpg')
```

Listing 11: An example to past one image to another image of Opencv library

```
import cv2
img = cv2.imread('the_scream.jpg')
blank = np.zeros((200,200,3),np.uint8)
wb,hb, cb = blank.shape
w,h,c = img.shape
img[10:wb+10,10:hb+10,:] = blank[:,:,:]
cv2.imshow('Hello ',img)
cv2.waitKey(0)
```

4 Exercises

Exercise 1: Write a program to split an input image with RGB space into three images which presented on Red, Green, Blue channels and then save on drive.

Exercise 2: Write a program to save the histogram of images for each channel which is splitted from the previous exercise.

Exercise 3: Write a program to crop a region of input image and save it to another image file by:

- PIL library (*hint:* image.crop((x,y,w,h)) function)
- OpenCV library (*hint:* image[y:y+h, x:x+w] with x, y, are vertical, horizontal value coordinate plane of image and h, w are the width and height of region)

Exercise 4: Write a program to draw a rectangle, circle, eclipse shape on image and put text *Hello world* on an input image by OpenCV library.

- To draw a circle, you need its center coordinates and radius.

  ```
  cv2.circle(img,(447,63), 63, (0,0,255), -1)
  ```

- To draw a rectangle, you need top-left corner and bottom-right corner of rectangle.

  ```
  cv2.rectangle(img,(384,0),(510,128),(0,255,0),3)
  ```

- To draw the ellipse, we need to pass several arguments. One argument is the center location (x,y). Next argument is axes lengths (major axis length, minor axis length). angle is the angle of rotation of ellipse in anti-clockwise direction. startAngle and endAngle denotes the starting and ending of ellipse arc measured in clockwise direction from major axis

  ```
  cv2.ellipse(img,(256,256),(100,50),0,0,180,255,-1)
  ```

- To put text on image:

  ```
  font = cv2.FONT_HERSHEY_SIMPLEX
  cv2.putText(img,'OpenCV',(10,500), font, 4,(255,255,255)
  ,2,cv2.LINE_AA)
  ```

Exercise 5: Write a program to overlay a smaller image on a larger image using OpenCV library.

Exercise 6: Write a program to try on many different glasses for an input of face image.

Exercise 7: Write a program to **add, subtract, multiply** two image and then the results are saved with format *jpg, bmp.* What do you observe?

Exercise 8: Write functions to add two images $f_1(x)$ and $f_2(x)$ based on the following equations:

- $g(x) = (1 - a)f_1(x) + af_2(x)$, where a from [0, 1]

- $g(x) = a.f_1(x) + b.f_2(x) + y$, where a, b from [0, 1] and $y = 0$

Exercise 9: Write a function to compute a mean image from many different images and than it is saved on drive.

5 References

1. Reference 1

2. Reference 2