Image Processing Application – Spring 2021

Lecturer: Professor Lee Guee Sang

Student: Le Hoai Duy – 218493

**Homework – Python installation and example**

1. Result:

A picture containing text

Description automatically generated

1. Code and explanation:

# import necessary module: skimage, numpy and matplotlib

from skimage import data, io, filters

import numpy as np

import matplotlib.pyplot as plt

# Load a small section of the image.

image = data.coins()[0:95, 70:370] #crop subimage [y0:y0+h, x0:x0+w]

fig, axes = plt.subplots(ncols=2, nrows=2,

figsize=(24, 12)) #Create a figure and a set of subplots.

ax0, ax1, ax2, ax3 = axes.flat #assign each subplots to variables

ax0.imshow(image, cmap=plt.cm.gray) #show image in gray format at subplot 0

ax0.set\_title('Original', fontsize=24) #set name for subplot 0

ax0.axis('off') #Turn off axes in subplots

# Histogram.

values, bins = np.histogram(image,

bins=np.arange(256)) #Compute the histogram of image

ax1.plot(bins[:-1], values, lw=2, c='k')

ax1.set\_xlim(xmax=256)

ax1.set\_yticks([0, 400])

ax1.set\_aspect(.2)

ax1.set\_title('Histogram', fontsize=24)

# Apply threshold.

from skimage.filters import threshold\_local #import threshold\_local method

bw = threshold\_local(image, 95, offset=-15)

#Compute a threshold mask image based on local pixel neighborhood. #All pixels in the input image higher than the corresponding pixel in the threshold image are considered foreground.

ax2.imshow(bw, cmap=plt.cm.gray)

ax2.set\_title('Adaptive threshold', fontsize=24)

ax2.axis('off')

# Detect edges.

from skimage import feature #import feature class

edges = feature.canny(image, sigma=3,

low\_threshold=10,

high\_threshold=80) #Edge filter an image using the Canny algorithm.

ax3.imshow(edges, cmap=plt.cm.gray)

ax3.set\_title('Edges', fontsize=24)

ax3.axis('off')

# Save result.

fig.savefig("D:/result.png")