# CS4243 Assignment 1

# ===================

# Name: Tay Yang Shun

# Matric: A0073063M

import sys

import cv2

import cv2.cv as cv

import numpy as np

import random

# Example usage:

# python assg1.py airborne.jpg

IMAGE\_FILE\_NAME = sys.argv[1]

INTENSITY\_RANGE = 256

image = cv2.imread(IMAGE\_FILE\_NAME, cv2.CV\_LOAD\_IMAGE\_GRAYSCALE)

# Convert to 1-d array

flattened\_image = image.reshape(1, -1)[0]

# Calculate intensity frequency histogram

freq = [0] \* 256

for px in flattened\_image:

freq[px] += 1

bin\_size = sum(freq) // INTENSITY\_RANGE

# Calculate cumulative frequency of each intensity

cum\_freq = [0] \* INTENSITY\_RANGE

cum\_freq[0] = freq[0]

for i in range(1, INTENSITY\_RANGE):

cum\_freq[i] = cum\_freq[i-1] + freq[i]

# Generate a list of tuples: (equalized\_intensity, original\_img\_intensity)

freq\_temp = []

for i in range(INTENSITY\_RANGE):

limit = bin\_size \* i

for j, f in list(enumerate(cum\_freq)):

if limit <= f:

freq\_temp.append((i, j))

break

freq\_temp.append((None, 256))

# Generates a dictionary which maps original\_img\_intensity to equalized\_intensity

freq\_map = {}

orig\_intensity\_breakpoints = [f[1] for f in freq\_temp]

for i in range(INTENSITY\_RANGE):

if i in orig\_intensity\_breakpoints:

freq\_map[i] = [f[0] for f in freq\_temp if f[1] == i]

else:

for j in range(INTENSITY\_RANGE):

if freq\_temp[j+1][1] > i:

freq\_map[i] = [freq\_temp[j][0]]

break

# Iterate through original image and replace with new intensity values

for px in np.nditer(image, op\_flags=['readwrite']):

px[...] = random.sample(freq\_map[int(px)], 1)[0]

# Save new histogram equalized image

file\_name, file\_extension = IMAGE\_FILE\_NAME.split('.')

new\_file\_name = file\_name + '-equalized.' + file\_extension

cv2.imwrite(new\_file\_name, image)

print 'Histogram equalized image \'' + new\_file\_name + '\' generated'



