```
# 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'
```