**Image Enhancement**

Source code Available on GitHub :

<https://github.com/ratchanonp/imageprocessing/blob/main/Assignment_2/assignment_2.ipynb>

1. **Gamma Correction**

Source Code:

import cv2

import numpy as np

import matplotlib.pyplot as plt

def gamma\_correction(img, c, gamma):

img = img.astype(np.float32)

img = c \* img \*\* gamma

# Normalize the image

img = img / np.max(img) \* 255

return img.astype(np.uint8)

# %% [markdown]

# # Image Enhancements with Gamma correction

#

# %%

img\_path = "./assignment2\_image1.jpg"

img = cv2.imread(img\_path, cv2.IMREAD\_GRAYSCALE)

c\_vals = [1, 2, 3]

gamma\_vals = [0.25, 0.5, 1, 2]

row, col = len(c\_vals), len(gamma\_vals)

plt.figure(figsize=(20, 16))

# Apply gamma correction

for c in range(len(c\_vals)):

for gamma in range(len(gamma\_vals)):

c\_val, gamma\_val = c\_vals[c], gamma\_vals[gamma]

img\_corrected = gamma\_correction(img, c\_val, gamma\_val)

plt.subplot(row, col, c \* col + gamma + 1)

plt.imshow(img\_corrected, cmap="gray")

plt.axis("off")

plt.title("c: {}, gamma: {}".format(c\_val, gamma\_val))

plt.savefig("./assignment2\_image1\_gamma\_correction.png")

plt.show()

ผลลัพธ์ที่ได้

A close-up of several images of a toothbrush

Description automatically generated

1. Global Histogram Equalization

Source Code:

import cv2

import matplotlib.pyplot as plt

def globalHistogramEqualiztion(img):

# Calculate the histogram

hist = cv2.calcHist([img], [0], None, [256], [0, 256])

# Calculate the cumulative sum

cdf = hist.cumsum()

# Normalize the cdf

cdf\_normalized = cdf \* hist.max() / cdf.max()

# Calculate the new pixel values

img\_new = cdf\_normalized[img]

return img\_new

enchanced\_img = globalHistogramEqualiztion(img)

plt.figure(figsize=(10, 5))

# Plot the image and subplot histogram

plt.subplot(1, 2, 1)

plt.imshow(img, cmap="gray")

plt.axis("off")

plt.title("Original Image")

plt.subplot(1, 2, 2)

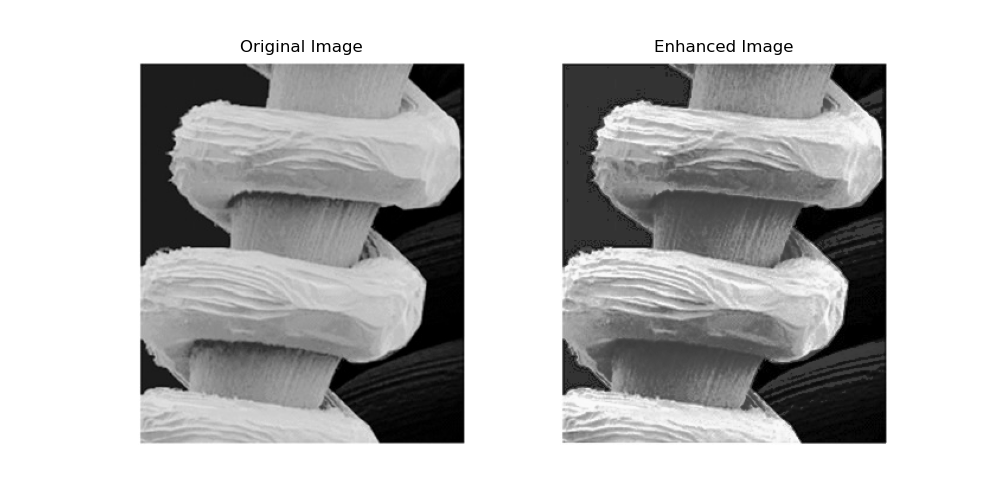
plt.imshow(enchanced\_img, cmap="gray")

plt.axis("off")

plt.title("Enhanced Image")

plt.savefig("./assignment2\_image1\_global\_histogram\_equalization.png")

plt.show()



1. Local Histogram Equalization

Source Code:

def local\_enhancement(img, E, k0, k1, k2, neighborhood\_size):

# Calculate the mean intensity and standard deviation of the entire image.

global\_mean\_intensity, global\_standard\_deviation = cv2.meanStdDev(img)

# Calculate the local mean intensity and local standard deviation.

local\_mean\_intensity = cv2.blur(img, (neighborhood\_size, neighborhood\_size))

local\_mean\_squared\_intensity = cv2.blur(np.square(img), (neighborhood\_size, neighborhood\_size))

local\_standard\_deviation = np.sqrt(local\_mean\_squared\_intensity - np.square(local\_mean\_intensity))

# Select the candidate pixels.

candidate\_pixels = []

for x in range(img.shape[0]):

for y in range(img.shape[1]):

mean\_condition = local\_mean\_intensity[x, y] <= k0 \* global\_mean\_intensity

std\_condition = k1 \* global\_standard\_deviation <= local\_standard\_deviation[x, y] <= k2 \* global\_standard\_deviation

if mean\_condition and std\_condition:

candidate\_pixels.append((x, y))

# Enhance the candidate pixels.

enhanced\_img = img.copy()

for x, y in candidate\_pixels:

# Enhance the candidate pixel.

enhanced\_img[x, y] = min(max(img[x, y] \* E, 0), 255)

return enhanced\_img

neighbourhood\_sizes = [3, 7, 11]

row, col = 1, len(neighbourhood\_sizes)

plt.figure(figsize=(20, 16))

E = 6

k0 = 0.125

k1 = 0.01

k2 = 0.125

for i in range(len(neighbourhood\_sizes)):

neighbourhood\_size = neighbourhood\_sizes[i]

enhanced\_img = local\_enhancement(

img,

E=E,

k0=k0,

k1=k1, k2=k2,

neighborhood\_size=neighbourhood\_size

)

plt.subplot(row, col, i + 1)

plt.imshow(enhanced\_img, cmap="gray")

plt.axis("off")

plt.title(f"Neighborhood Size: {neighbourhood\_size} x {neighbourhood\_size}")

# Add text at bottom

plt.figtext(0.5, 0.275, f"Parameters: E:{E}\n k0:{k0} k1:{k1} k2:{k2}", ha="center", fontsize=16)

plt.savefig("./assignment2\_image1\_local\_enhancement.png")

plt.show()

Several images of a rope

Description automatically generated

ค่า k0 k1 k2 ที่ใช้คือ

K0 = 0.175 K1 = 0.01 และ k2 = 0.175

ที่เลือกใช้ค่าดังกล่าว เนื่องจากต้องการให้ Candidate pixel ที่จะทำการ enhance นั้นเป็น pixel บริเวณสิดำ

**Method ที่ควรใช้คือ**

Local Histogram Equalization เนื่องจากภาพที่เราต้องการ Enhancement นั้นส่วนที่ต้องการ Enhancement เป็นเฉพาะส่วนที่มืดบริเวณเล็กๆ และไม่ต้องการที่จะสูญเสีย Contrast บริเวณสีขาวที่ชัดอยู่แล้วไป โดยเราสามารถกำหนดค่า Parameter เพื่อให้ Algorithm นั้นไป Enhancement เฉพาะส่วนที่เราต้องการได้ และยังรักษา Contrast ของภาพโดยรวมเอาไว้