산업 컴퓨터 비전 실제 프로그래밍 과제#1

2024254002 전창수

Homework #1

**아래의 각 알고리즘에 대한 코드와 결과를 넣은 보고서를 제출하세요.**

**1. 히스토그램 평탄화**

사용자로부터 R, G, B 중의 하나의 채널을 입력받고 입력받은 채널에 대한 히스토그램을 그리고 평탄화를 한 후에 다시 그 영상을 출력하시오.  
입력받은 컬러 영상을 HSV 컬러 스페이스로 변경한 후에 V 채널에 대한 평탄화를 한 후에 다시 그 영상을 출력하시오.

SOURCE CODE

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
image = cv2.imread("images/lena.png", cv2.IMREAD\_COLOR)  
img\_eq = image.copy()  
b, g, r = cv2.split(image)  
inverse = cv2.merge((r, g, b))  
  
keyin = input("R, B, G 중 하나를 입력하세요. Color : ")  
  
if keyin == "R":  
 eq = cv2.equalizeHist(r)  
 img\_eq = cv2.merge((b, b, eq))  
elif keyin == "G":  
 eq = cv2.equalizeHist(g)  
 img\_eq = cv2.merge((b, eq, b))  
elif keyin == "B":  
 eq = cv2.equalizeHist(b)  
 img\_eq = cv2.merge((eq, g, g))  
  
hist, bins = np.histogram(eq, 256, [0,255])  
plt.fill(hist)  
plt.xlabel('histogram')  
plt.show()  
  
cv2.imshow("orginal image", image)  
cv2.imshow("equalize color = " + keyin, img\_eq)  
cv2.waitKey()  
cv2.destroyAllWindows()

R

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명인간의 얼굴, 머리장식, 의류, 패션 액세서리이(가) 표시된 사진

자동 생성된 설명

B

텍스트, 스크린샷, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명의류, 인간의 얼굴, 머리장식, 패션 액세서리이(가) 표시된 사진

자동 생성된 설명

G

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명 인간의 얼굴, 의류, 패션 액세서리, 모자이(가) 표시된 사진

자동 생성된 설명

**2. 공간 도메인 필터링**

입력 영상에 대해서 임의의 노이즈를 입히고, Gaussian Filtering, Median Filtering, Bilateral Filtering 알고리즘을 적용해서 결과를 출력하시오.  
각 결과를 입력 영상(노이즈 입히기 전)과 절대값 차이를 취해서 그 결과를 출력하시오.

source

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
image = cv2.imread('images/lena.png').astype(np.float32) / 255  
  
noised = (image + 0.2 \* np.random.rand(\*image.shape).astype(np.float32))  
noised = noised.clip(0, 1)  
plt.imshow(noised[:,:,[2,1,0]])  
plt.show()  
  
gauss\_blur = cv2.GaussianBlur(noised, (7,7),0)  
plt.imshow(gauss\_blur[:,:,[2,1,0]])  
plt.show()  
  
median\_blur = cv2.medianBlur((noised \* 255).astype(np.uint8), 7)  
plt.imshow(median\_blur[:,:,[2,1,0]])  
plt.show()  
  
bilat = cv2.bilateralFilter(noised, 10, 0.3, 10)  
plt.imshow(bilat[:,:,[2,1,0]])  
plt.show()

|  |  |
| --- | --- |
| **인간의 얼굴, 의류, 패션 액세서리, 스크린샷이(가) 표시된 사진  자동 생성된 설명**  **Noise 입힌 후** | **인간의 얼굴, 의류, 스크린샷, 패션 액세서리이(가) 표시된 사진  자동 생성된 설명**  **Gaussian Filtering** |
| **인간의 얼굴, 스크린샷, 의류, 모자이(가) 표시된 사진  자동 생성된 설명**  **Median Filtering** | **인간의 얼굴, 의류, 스크린샷, 패션 액세서리이(가) 표시된 사진  자동 생성된 설명**  **Bilateral Filtering** |

**3. 주파수 도메인 필터링**

DFT를 통해서 입력 영상을 주파수 도메인으로 바꿔서 출력한 후에 사용자로부터 반지름을 두 개 입력 받은 후에 영상의 중심을 원의 중심으로 하는 두 개의 원을 각각 그린 후에 그 두 원 사이의 값을 통과시키고, 나머지 부분은 다 지워버리는 band pass 필터링을 구현하여 출력하시오.

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
mask\_size = int(input('mask size를 입력하세요'))  
  
chk = False  
flag = 1  
while not chk:  
 flag = int(input('1(High-Pass) 또는 2(Low-Pass)를 입력하세요'))  
 if 0 < flag < 3:  
 chk = True  
 else:  
 print('1 or 2 중 하나를 입력하세요')  
  
  
image = cv2.imread('images/lena.png', 0).astype(np.float32) / 255  
fft = cv2.dft(image, flags=cv2.DFT\_COMPLEX\_OUTPUT)  
fft\_shift = np.fft.fftshift(fft, axes=[0, 1])  
  
mask = np.zeros(image.shape, np.uint8)  
center = (image.shape[0]//2, image.shape[1]//2)  
cv2.circle(mask, center, mask\_size, 255, cv2.FILLED)  
  
if flag == 1:  
 mask = cv2.bitwise\_not(mask)  
  
fft\_shift[:, :, 0] \*= mask  
fft\_shift[:, :, 1] \*= mask  
  
fft = np.fft.ifftshift(fft\_shift, axes=[0, 1])  
  
filtered = cv2.idft(fft, flags=cv2.DFT\_SCALE | cv2.DFT\_REAL\_OUTPUT)  
mask\_new = np.dstack((mask, np.zeros((image.shape[0], image.shape[1]), dtype=np.uint8)))  
  
plt.figure(figsize=(15, 8))  
plt.subplot(131)  
plt.axis('off')  
plt.title('original')  
plt.imshow(image, cmap='gray')  
plt.subplot(132)  
plt.axis('off')  
plt.title('no high frequencies')  
plt.imshow(filtered, cmap='gray')  
plt.subplot(133)  
plt.axis('off')  
plt.title('mask')  
plt.imshow(mask, cmap='gray')  
plt.show()

스크린샷, 인간의 얼굴, 여성, 흑백이(가) 표시된 사진

자동 생성된 설명

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
image = cv2.imread('images/lena.png', 0).astype(np.float32) / 255  
  
fft = cv2.dft(image, flags=cv2.DFT\_COMPLEX\_OUTPUT)  
fft\_shift = np.fft.fftshift(fft, axes=[0, 1])  
sz = 25  
mask = np.zeros(fft.shape, np.uint8)  
mask[image.shape[0]//2-sz:image.shape[0]//2+sz,  
 image.shape[1]//2-sz:image.shape[1]//2+sz, :] = 1  
fft\_shift \*= mask  
fft = np.fft.ifftshift(fft\_shift, axes=[0, 1])  
  
filtered = cv2.idft(fft, flags=cv2.DFT\_SCALE | cv2.DFT\_REAL\_OUTPUT)  
mask\_new = np.dstack((mask, np.zeros((image.shape[0], image.shape[1]), dtype=np.uint8)))  
  
plt.figure()  
plt.subplot(131)  
plt.axis('off')  
plt.title('original')  
plt.imshow(image, cmap='gray')  
plt.subplot(132)  
plt.axis('off')  
plt.title('no high frequencies')  
plt.imshow(filtered, cmap='gray')  
plt.subplot(133)  
plt.axis('off')  
plt.title('mask')  
plt.imshow(mask\_new\*255, cmap='gray')  
plt.tight\_layout()  
plt.show()

스크린샷, 인간의 얼굴, 화이트, 여성이(가) 표시된 사진

자동 생성된 설명

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
def create\_bandpass\_filter(rows, cols, inner\_radius, outer\_radius):  
 center\_row, center\_col = rows // 2, cols // 2  
  
 # Create two circles centered at the center of the image with given radii  
 mask\_inner\_circle = np.zeros((rows, cols), dtype=np.uint8)  
 mask\_outer\_circle = np.ones((rows, cols), dtype=np.uint8)  
 cv2.circle(mask\_inner\_circle, (center\_col, center\_row), inner\_radius, 255, -1)  
 cv2.circle(mask\_outer\_circle, (center\_col, center\_row), outer\_radius, 0, -1)  
  
  
# Create a mask with two channels  
 band\_pass\_mask = np.zeros((rows, cols, 2), dtype=np.uint8)  
 band\_pass\_mask[:, :, 0] = mask\_inner\_circle  
 band\_pass\_mask[:, :, 1] = mask\_inner\_circle # Duplicate for the imaginary part  
  
 return band\_pass\_mask  
  
# Read the image  
image = cv2.imread('images/lena.png', cv2.IMREAD\_GRAYSCALE).astype(np.float32) / 255  
  
# Perform Discrete Fourier Transform (DFT)  
fft = cv2.dft(image, flags=cv2.DFT\_COMPLEX\_OUTPUT)  
fft\_shift = np.fft.fftshift(fft, axes=[0, 1])  
  
# Get image dimensions  
rows, cols = image.shape  
  
# Get user input for inner and outer radii  
inner\_radius = int(input("Enter inner radius: "))  
outer\_radius = int(input("Enter outer radius: "))  
  
# Create bandpass filter mask  
band\_pass\_mask = create\_bandpass\_filter(rows, cols, inner\_radius, outer\_radius)  
  
# Apply the mask to the shifted spectrum  
fft\_shift\_filtered = fft\_shift \* band\_pass\_mask  
  
# Perform inverse DFT to obtain the filtered image  
filtered\_shift = np.fft.ifftshift(fft\_shift\_filtered, axes=[0, 1])  
filtered\_image = cv2.idft(filtered\_shift, flags=cv2.DFT\_SCALE | cv2.DFT\_REAL\_OUTPUT)  
  
# Display the original and filtered images  
plt.figure(figsize=(10, 5))  
plt.subplot(121)  
plt.imshow(image, cmap='gray')  
plt.title('Original Image')  
plt.axis('off')  
  
plt.subplot(122)  
plt.imshow(filtered\_image, cmap='gray')  
plt.title('Filtered Image (Band Pass)')  
plt.axis('off')  
  
plt.tight\_layout()  
plt.show()

인간의 얼굴, 스크린샷, 의류, 여성이(가) 표시된 사진

자동 생성된 설명

**4. 모폴로지 필터**

사용자의 입력을 받아서 Otsu 기반의 이진화와 Adaptive 이진화 (median)를 선택해서 그 결과를 출력하고, Erosion, Dilation, Opening, Closing에 대한 선택과 횟수를 입력받아서 해당 결과를 출력하시오.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

인간의 얼굴, 스케치, 일러스트레이션, 포스터이(가) 표시된 사진

자동 생성된 설명

- otsu임계값으로 이진화

텍스트, 인간의 얼굴, 포스터, 그래픽 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 포스터, 그래픽 디자인, 그래픽이(가) 표시된 사진

자동 생성된 설명

텍스트, 그래픽 디자인, 그래픽, 실루엣이(가) 표시된 사진

자동 생성된 설명

- Erosion 1회, Dilation 1회, Closing 2회 실행한 순서