

차량비전시스템

Assignment 1

학번 5533082
전공 컴퓨터공학전공
이름 최승환

Q1) 개발환경

Windows에서 Anaconda, VSCode를 이용하여 python 개발환경 구축

→ conda=22.9.0 / python=3.8.16

필요 라이브러리

→ numpy=1.24.2 / matplotlib=3.7.1 / opencv-python=4.7.0.72

Q2.1~6)

- | | | |
|-----------------|-----------------------|-------------------|
| 1) RGB 이미지 출력 | 2) R-channel 출력 | 3) G-channel 출력 |
| 4) B-channel 출력 | 5) Gray scale 이미지로 변환 | 6) Binary 이미지로 변환 |

```
import cv2
import matplotlib.pyplot as plt

origin_img = cv2.imread('./img/duck.jpg')

# RGB 이미지로 변환
RGB_img = cv2.cvtColor(origin_img, cv2.COLOR_BGR2RGB)

# 채널 분리
Red_img, Green_img, Blue_img = cv2.split(RGB_img)

# 그레이스케일 이미지로 변환
Gray_img = cv2.cvtColor(RGB_img, cv2.COLOR_RGB2GRAY)

# 그레이스케일 이미지에 이진화
# ret: 설정 임계값, Binary_img: 결과 이미지
ret, Binary_img = cv2.threshold(Gray_img, 100, 255, cv2.THRESH_BINARY)

plt.figure( figsize=(20,10) )

# RGB
plt.subplot(2, 3, 1)
plt.title("1. RGB image")
plt.axis("off")
plt.imshow(RGB_img)

# R-channel
RGB_img[:, :, 0] = Red_img
RGB_img[:, :, 1] = 0
RGB_img[:, :, 2] = 0
plt.subplot(2, 3, 2)
plt.title("2. Red image")
plt.axis("off")
plt.imshow(RGB_img)
```

```

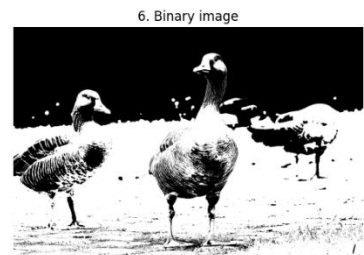
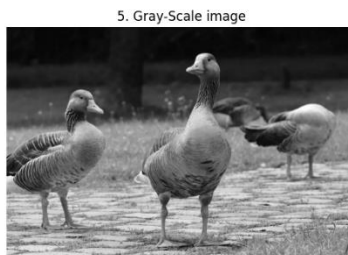
# G-channel
RGB_img[:, :, 0] = 0
RGB_img[:, :, 1] = Green_img
RGB_img[:, :, 2] = 0
plt.subplot(2, 3, 3)
plt.title("3. Green image")
plt.axis("off")
plt.imshow(RGB_img)

# B-channel
RGB_img[:, :, 0] = 0
RGB_img[:, :, 1] = 0
RGB_img[:, :, 2] = Blue_img
plt.subplot(2, 3, 4)
plt.title("4. Blue image")
plt.axis("off")
plt.imshow(RGB_img)

# Gray-Scale
plt.subplot(2, 3, 5)
plt.axis("off")
plt.title("5. Gray-Scale image")
# 컬러맵을 지정하지 않으면 임의로 색상이 지정되어 출력된다.
plt.imshow(Gray_img, cmap='gray')

# Binarization
plt.subplot(2, 3, 6)
plt.axis("off")
plt.title("6. Binary image")
plt.imshow(Binary_img, cmap='gray')
plt.show()

```



Q2. 7~9)

7) 5)번 작업을 수행한 후 출력 이미지에서 최대 화소값 출력

8) 5)번 작업을 수행한 후 출력 이미지에서 최소 화소값 출력

9) 5)번 작업을 수행한 후 전체 화소값의 평균값 출력

```
import numpy as np
print(f"7. 최대 화소값: {np.max(Gray_img)}")
print(f"8. 최소 화소값: {np.min(Gray_img)}")
print(f"9. 평균 화소값: {np.mean(Gray_img)}")
```

```
7. 최대 화소값: 255
8. 최소 화소값: 0
9. 평균 화소값: 105.31127888997396
```

Q3-1) 두 이미지 모두 gray scale로 변환

```
import cv2
import matplotlib.pyplot as plt

pic1 = cv2.imread('./img/duck.jpg')
pic2 = cv2.imread('./img/cat.jpg')
pic1_gray = cv2.cvtColor(pic1, cv2.COLOR_BGR2GRAY)
# 픽셀 resize
pic1_gray = cv2.resize(pic1_gray, dsize=(510, 340), interpolation=cv2.INTER_AREA)
pic2_gray = cv2.cvtColor(pic2, cv2.COLOR_BGR2GRAY)

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

plt.subplot(1,2,1)
plt.title("pic1 gray-scale image")
plt.axis("off")
plt.imshow(pic1_gray, cmap='gray')

plt.subplot(1,2,2)
plt.title("pic2 gray-scale image")
plt.axis("off")
plt.imshow(pic2_gray, cmap='gray')

plt.show()
```

pic1 gray-scale image



pic2 gray-scale image



Q3-2) 두 이미지의 화소값을 모두 더하시오.

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

# Saturation
pic_saturation = cv2.add(pic1_gray, pic2_gray) # 자동 Saturation
plt.subplot(1,2,1)
plt.title("2-a. Saturation")
plt.axis("off")
plt.imshow(pic_saturation, cmap='gray')

# Wrapping
pic_wrapping = pic1_gray + pic2_gray # 자동 Wrapping
plt.subplot(1,2,2)
plt.title("2-b. Wrapping")
plt.axis("off")
plt.imshow(pic_wrapping, cmap='gray')
plt.show()
```

2-a. Saturation



2-b. Wrapping

