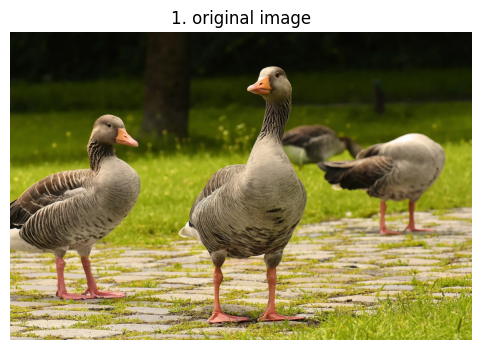
**문제 1) RGB 이미지를 세 개의 채널(R, G, B)로 분리한 뒤 다시 합치는 프로그램을 작성하시오.**



위 원본 이미지를 R/G/B 세 개 채널로 분리하는 방법은 아래 코드와 같다.

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)

# R-channel

RGB\_img[:,:,0] = Red\_img

RGB\_img[:,:,1] = 0

RGB\_img[:,:,2] = 0

plt.subplot(1, 3, 1)

plt.title("2. Red image")

plt.axis("off")

plt.imshow(RGB\_img)



분리된 이미지들을 다시 합치는 방법은 opencv에서 제공하는 메소드인 cv2.merge 함수를 사용하면 된다.

# 채널 병합

merged\_img = cv2.merge((Red\_img, Green\_img, Blue\_img))

# cv2.merge

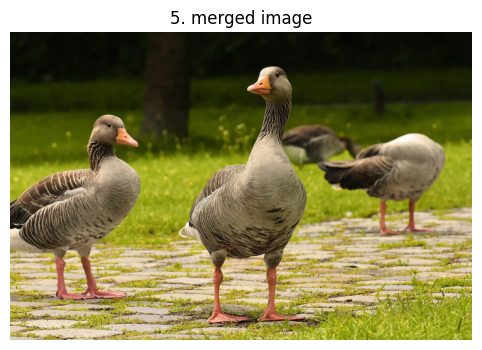
plt.figure( figsize=(8, 4) )

plt.title("5. merged image")

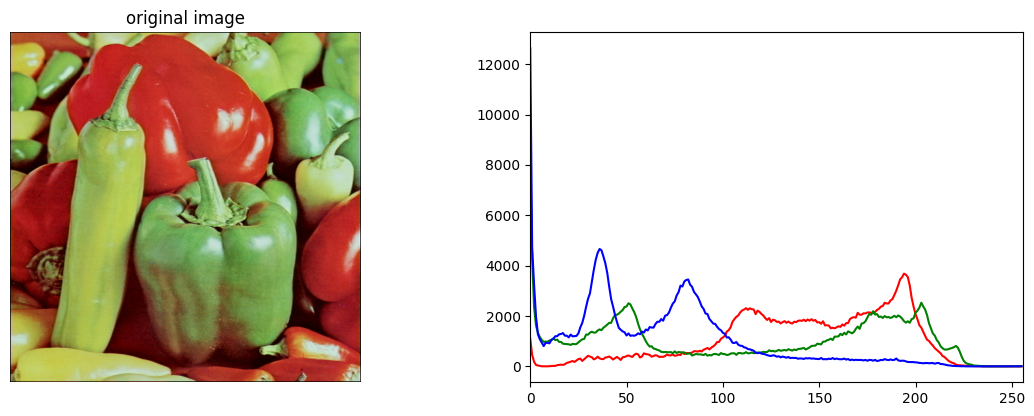
plt.axis("off")

plt.imshow(merged\_img)

plt.show()



**문제 2) peppers.jpg 이미지를 사용하여**



**2-1) R, G, B 각 채널별로 전역 임계값을 구하시오.**

위 원본 이미지의 최적의 전역 임계값을 구하기 위해서 히스토그램을 확인해봐야 한다.

# RGB 채널 나누기

Red\_img, Green\_img, Blue\_img = cv2.split(RGB\_img)

# 히스토그램 출력, 채널은 0으로 표시

plt.subplot(222)

hist = cv2.calcHist([Red\_img], [0], None, [256], [0, 256])

plt.xlim([0, 256])

plt.plot(hist, color = 'r') # R 색상 히스토그램 표현

hist = cv2.calcHist([Green\_img], [0], None, [256], [0, 256])

plt.xlim([0, 256])

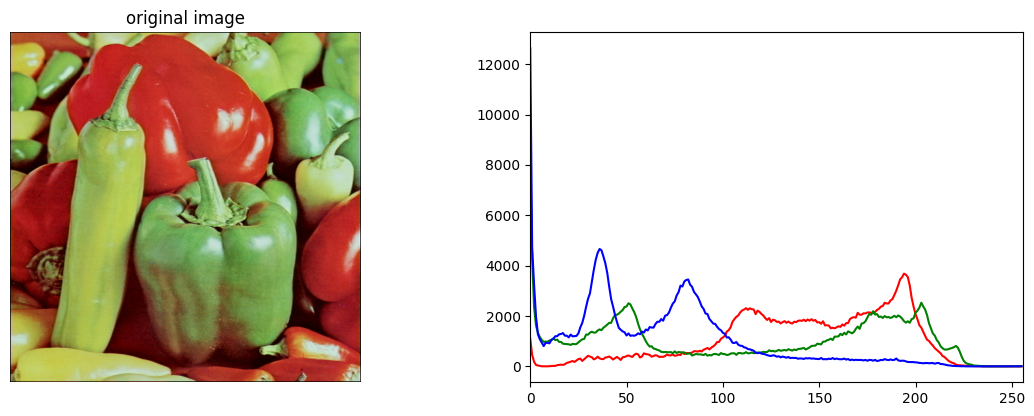
plt.plot(hist, color = 'g') # G 색상 히스토그램 표현

hist = cv2.calcHist([Blue\_img], [0], None, [256], [0, 256])

plt.xlim([0, 256])

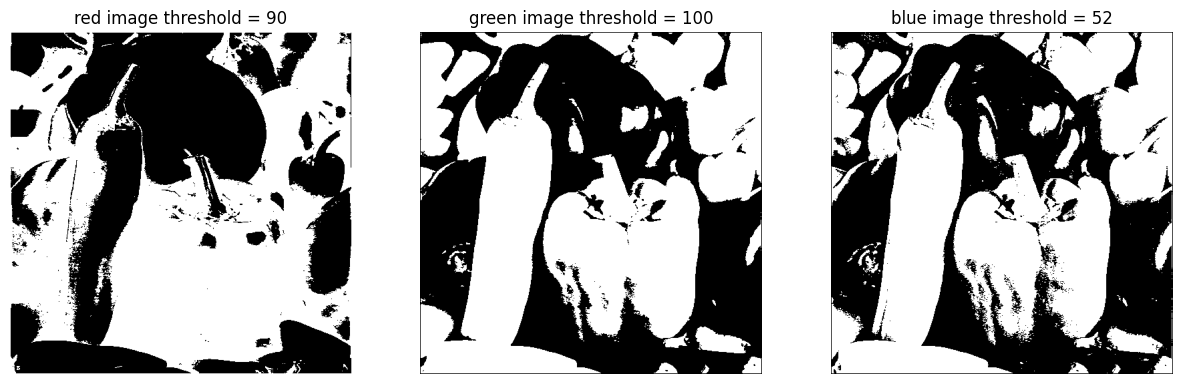
plt.plot(hist, color = 'b') # B 색상 히스토그램 표현

plt.show()



임계값은 전경 영역과 배경 영역의 밝기 차이나 색상 차이를 이용하여 영상의 배경으로부터 전경 영역을 분리할 수 있는 값이기 때문에 위 히스토그램을 보고 최대한의 차이를 나타낼 수 있는 경계를 찾아야 한다.

이 그림의 경우 R-channel : 90, G-channel: 100, B-channel: 52 이 가장 적합하다고 판단된다.



**2-2) gray scale로 변환한 후 전역 임계값을 구하시오.**

# 그레이스케일 이미지로 변환

Gray\_img = cv2.cvtColor(RGB\_img, cv2.COLOR\_RGB2GRAY)

# gray-scale

plt.figure(figsize=(20, 5))

plt.subplot(131)

plt.title("gray-scale image")

plt.axis("off")

plt.imshow(Gray\_img, 'gray')

# gray-scale histogram

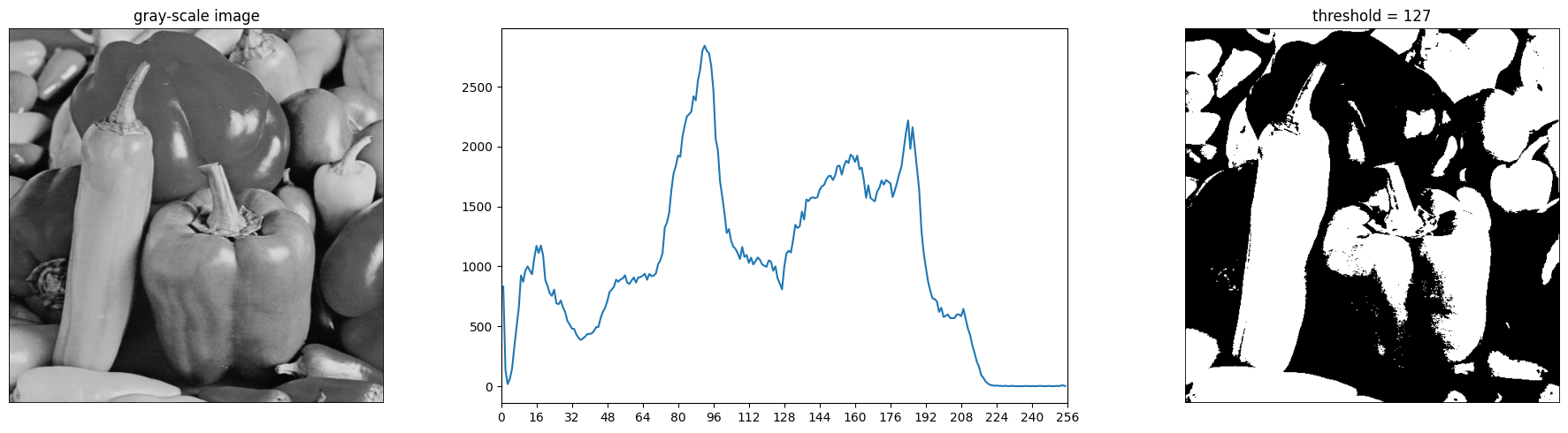
plt.subplot(132)

hist = cv2.calcHist([Gray\_img], [0], None, [256], [0, 256])

plt.xlim([0, 256])

plt.xticks([i\*16 for i in range(17)])

plt.plot(hist)



위 히스토그램을 보면 임계값을 127로 설정했을 때 이진화 결과가 가장 좋았다.

**2-3) 2)에서 얻은 전역 임계값을 기준으로 영상을 이진화 하시오.**

# 그레이스케일 이미지에 이진화

# ret: 설정 임계값, Binary\_img: 결과 이미지

ret, Binary\_img = cv2.threshold(Gray\_img, 127, 255, cv2.THRESH\_BINARY)

plt.subplot(133)

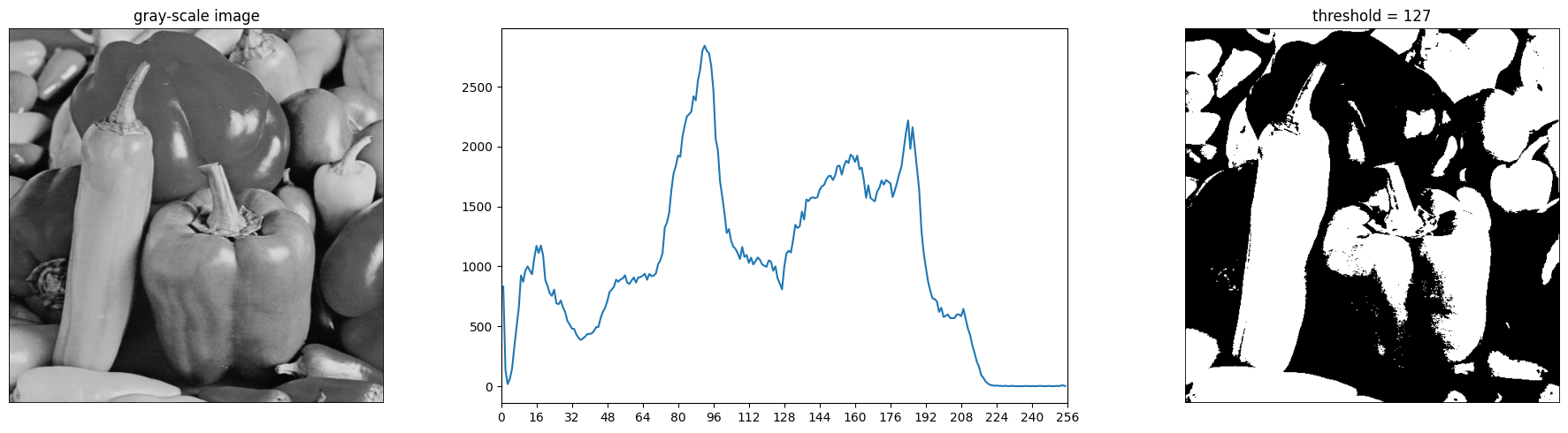
plt.axis("off")

plt.title("threshold = 127")

plt.imshow(Binary\_img, 'gray')

plt.tight\_layout()

plt.show()



**문제 3) peppers.jpg 이미지를 사용하여**

**3-1) 이미지의 블록 개수를 25개로 나눈 후 적응적 임계값을 구하시오.  
3-2) 적응적 임계값을 구한 결과를 기준으로 각 블록을 이진화 한 결과를 나타내시오.**

블록 개수를 25개로 나누고

img = cv2.imread('./img/peppers.jpg')

RGB\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

gray\_img = cv2.cvtColor(RGB\_img, cv2.COLOR\_RGB2GRAY)

# 가로세로 블록의 개수

N=25

# 블록 당 가로와 세로 크기 계산

dimh = np.int32(gray\_img.shape[0] / N)

dimw = np.int32(gray\_img.shape[1] / N)

# 연산에서 제외될 영상 가장자리 크기 계산

dh\_rest = np.int32(gray\_img.shape[0] % N)

dw\_rest = np.int32(gray\_img.shape[1] % N)

# 임계값 적용 후 이진 영상을 담을 기억 장소 생성

mean\_img = np.zeros((N, N)) # 블록의 평균값 저장 배열

median\_img = np.zeros((N, N)) # 블록의 중앙값 저장 배열

maxmin\_mean\_img = np.zeros((N, N)) # 블록의 최대 최소 평균값 저장 배열

binary\_mean\_img = np.zeros((gray\_img.shape[0], gray\_img.shape[1]))

binary\_median\_img = np.zeros((gray\_img.shape[0], gray\_img.shape[1]))

binary\_maxmin\_mean\_img = np.zeros((gray\_img.shape[0], gray\_img.shape[1]))

# 각 블록의 평균값, 중앙값, 최대최소평균값 계산

for h in range(0, img.shape[0] - dh\_rest, dimh):

    for w in range(0, img.shape[1] - dw\_rest, dimw):

        if(h + dimh < img.shape[0] and w + dimw < img.shape[1]):

            mean\_img[np.int32(h/dimh), np.int32(w/dimw)]

= mean\_function(gray\_img, dimh, dimw, h, w)

            median\_img[np.int32(h/dimh), np.int32(w/dimw)]

= medain\_function(gray\_img, dimh, dimw, h, w)

            maxmin\_mean\_img[np.int32(h/dimh), np.int32(w/dimw)]

= maxmin\_mean\_function(gray\_img, dimh, dimw, h, w)

**문제 4) MOT16-1.png 이미지 활용**

1) R/G/B 각 채널별 히스토그램 분포를 나타내시오.

2) gray scale로 변환한 후 히스토그램 분포를 나타내시오

3) 히스토그램 스트레칭을 적용한 결과의 이미지와 스트레칭 적용 후의 히스토그램 분포 그래프를 나타내시오.

4) 히스토그램 평활화를 적용한 결과의 이미지와 평활화 후의 히스토그램 분포 그래프를 나타내시오.

**Q5) Copy of Picture1.png 와 Copy of Picture2.png를 사용하여 연산을 수행하시오.**