

## 영상처리 과제 #4

마감기한: 10월 11일 일요일 23:59까지

학과	전자정보통신공학과	과목명	영상처리(001)
학번	18010697	이름	김해리

1.

### M Code

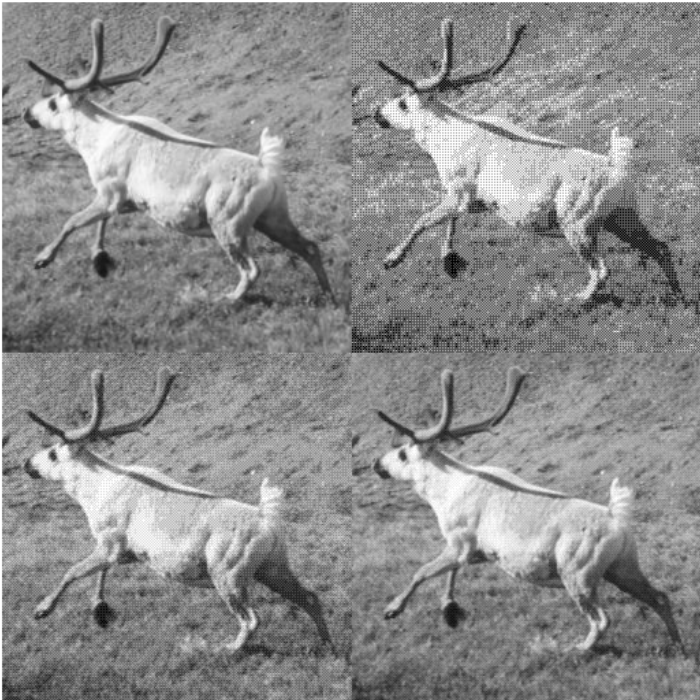
편집기 - C:\MATLAB\_ImageProcessing\HW4\_1.m

```

4 % 강의자료를 참조하여, caribou.tif 영상에 대해
5 % 해당 영상이 4, 8, 16개의 출력을 갖도록 디더링하세요.
6 % (16개의 출력을 갖도록 할 땐 디더링 매트릭스 D = [0 10; 15 5]를 사용할 것)
7
8 im = imread("sample_images/caribou.tif");
9 D4 = [0 56; 84 28]; r4 = repmat(D4, 128, 128);
10 D8 = [0 24; 36 12]; r8 = repmat(D8, 128, 128);
11 D16 = [0 10; 15 5]; r16 = repmat(D16, 128, 128);
12
13 im = double(im);
14
15 q = floor(im/85);
16 im4 = q + (im - 85 * q > r4); % 4개 출력 디더링
17 im4 = uint8(im4*85);
18
19 q = floor(im/37);
20 im8 = q + (im - 37 * q > r8); % 8개 출력 디더링
21 im8 = uint8(im8*37);
22
23 q = floor(im/16);
24 im16 = q + (im - 16 * q > r16); % 16개 출력 디더링
25 im16 = uint8(im16*16);
26
27 im = uint8(im);
28 imshow([im im4; im8 im16]);

```

### 처리 결과



(1,1): 원본,      (1,2): 4개 출력

(2,1): 8개 출력,      (2,2): 16개 출력

2.

## M Code

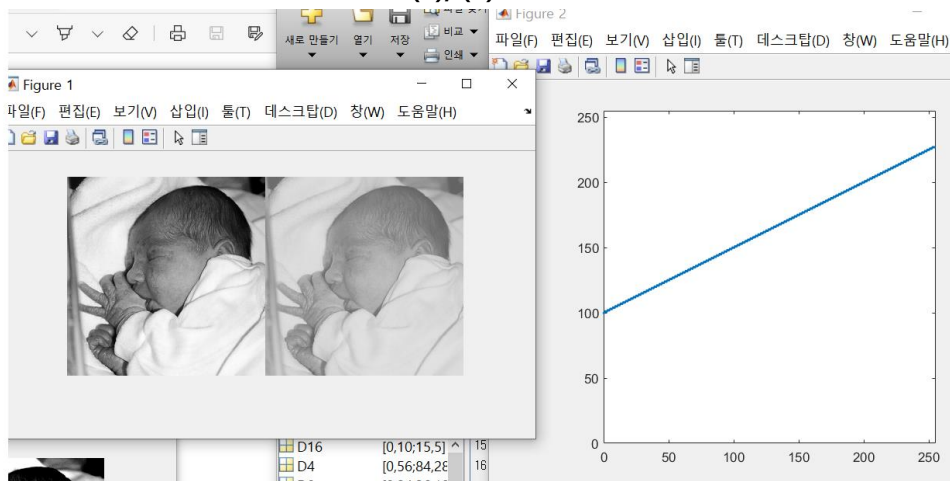
```
im = imread("sample_images/newborn.tif");

% 2.
% 강의자료를 참조하여, newborn.tif 영상에 대해
% 제시된 문제를 해결하세요.

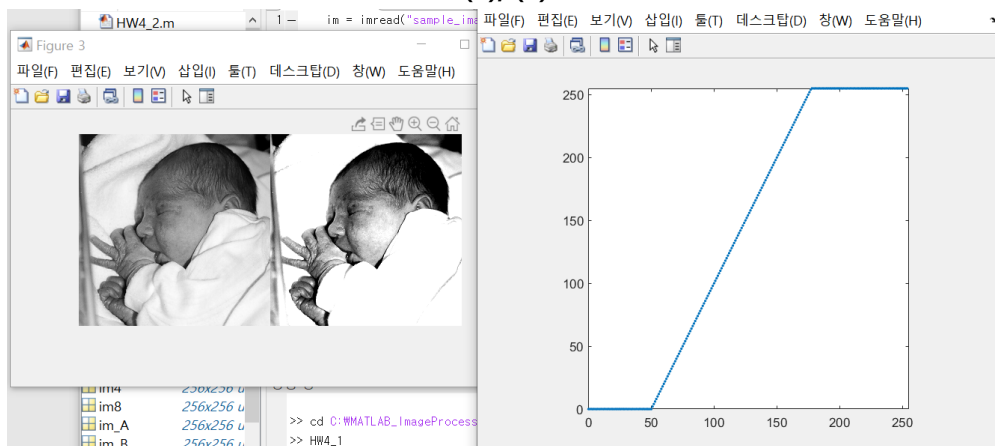
% (1) 모든 화소의 밝기 값을 2로 나누고 100을 더한 영상을 생성하고 디스플레이하세요.
im_A = imadd(imdivide(im, 2), 100);
figure, imshow([im, im_A]);
% (2) 1에서 밝기값이 어떻게 변화되는지를 아래 코드를 사용해 확인하세요.
figure; plot(im(:), im_A(:), '.'); axis equal; axis([0 255 0 255]); % a는 원영상, b는 밝기값이 변경된 영상
% (3) 모든 화소의 밝기값에서 50을 빼고, 2를 곱한 영상을 생성하고 디스플레이하세요.
im_B = immultiply(imsubtract(im, 50), 2);
figure, imshow([im, im_B]);
% (4) (3)에서 밝기값이 어떻게 변화되는가를 아래의 코드를 사용해 확인하세요.
figure; plot(im(:), im_B(:), '.'); axis equal; axis([0 255 0 255]);
```

## 처리결과

(1), (2)



(3), (4)

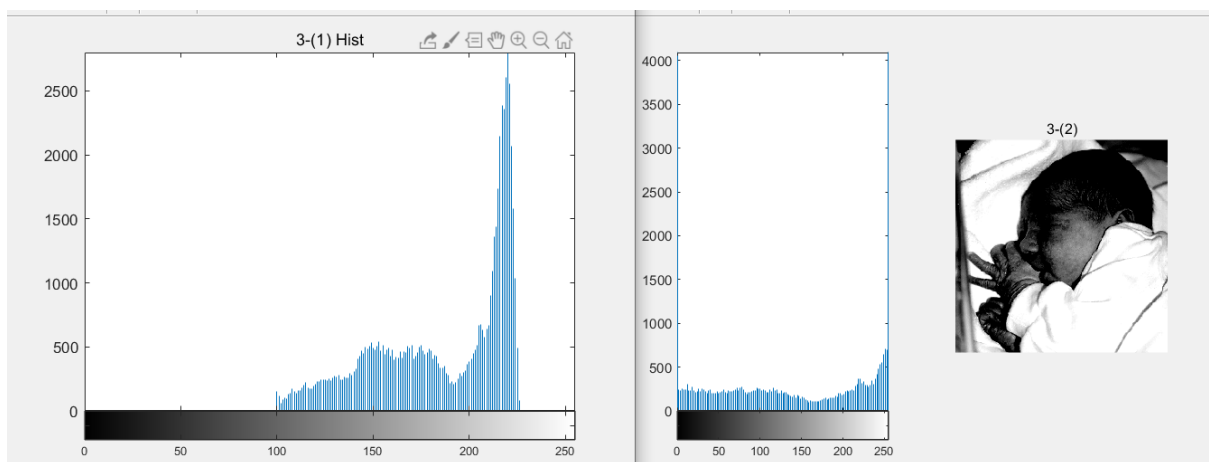


### 3.

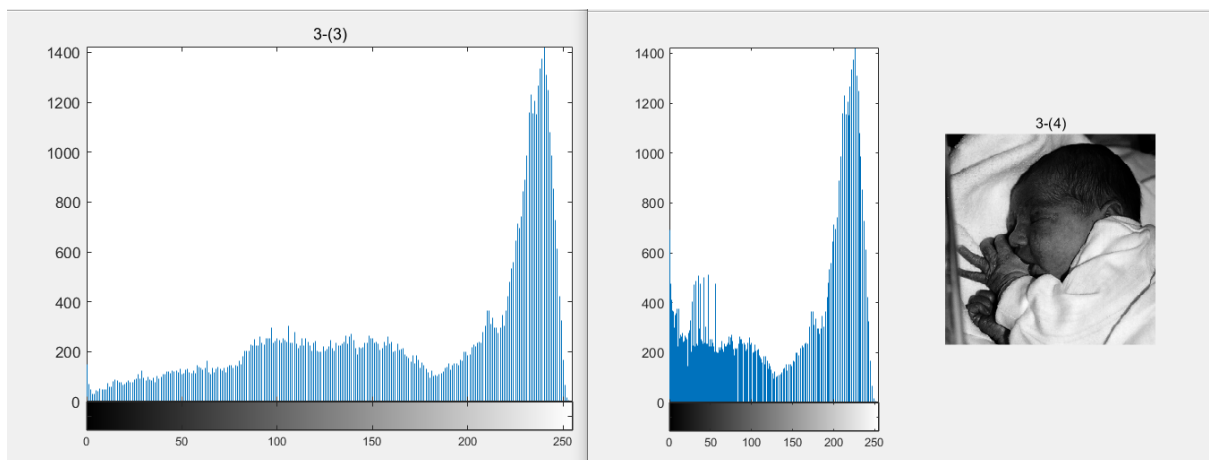
```
% =====
% 3.
% newborn.tif 영상에 대해 아래의 문제를 해결하세요.

% (1) 앞선 2-(1)의 결과로 획득된 영상 b의 histogram을 그려보세요.
figure(5), imhist(imA); axis tight; title("3-(1) Hist");

% (2) Histogram을 보면, 해당 영상에 포함된 화소들의 밝기값이 100~228 사이에 존재함을 확인할 수 있는데,
% imadjust(im, [x y], [z w]);를 사용하여
% 100~228 사이의 밝기값을 0~255 사이의 밝기값으로 변경한 결과 영상과 histogram을 디스플레이하세요.
imA_adj = imadjust(im, [100/255, 228/255], [0, 1]);
figure(6),
subplot(1,2,1); imhist(imA_adj); axis tight;
subplot(1,2,2); imshow(imA_adj); axis tight;
title("3-(2)");
```



```
36 % (3) newborn.tif 영상을 읽어서 histogram을 그려보세요. (4장 강의자료 10쪽 참조)
37 figure(7), imhist(im); title("3-(3)");
38 % (4) Histogram을 보면, 해당 영상은 밝기값 220~250 사이에 많은 화소들이 존재함을 확인할 수 있는데,
39 % imadjust(im, [0 1], [0 1], gamma);를 사용하여 histogram에서 어두운 쪽은 줄이고(shrinking), 밝은 쪽은 늘
40 % 리려고(stretching) 하는 경우에 대해 적절한 gamma 값을 찾고, 이를 실행하여 결과 영상의 histogram을 디스플레이하세요.
41 im_adj = imadjust(im, [0, 1], [0, 1], 2); % 원하는 처리를 위해선 gamma가 1보다 커야 함
42 figure(8),
43 subplot(1,2,1); imhist(im_adj); axis tight;
44 subplot(1,2,2); imshow(im_adj); axis tight;
45 title("3-(4)");
```



#### 4.

```

편집기 - C:\MATLAB_ImageProcessing\HW4_4.m
HW4_4.m x histpwl.m x +
1 % 4. histpwl 함수에 대해 아래의 문제를 해결하세요.
2 % (1) histpwl 함수를 작성하여 histpwl.m 파일로 저장하세요.
3 % |
4
5 % (2) tire.tif 영상을 t로 읽어서 histpwl 함수를 실행한 후,
6 % 결과 영상 th1을 디스플레이하세요.
7 t = imread('sample_images/tire.tif');
8 th1=histpwl(t, [0 0.3 0.8 1.0], [0 0.6 0.7 1.0]);
9
10 % (3) histpwl 함수를 사용하여 (2)번 결과인 th1을 원영상인 t와 유사하게 역변경하려고 합니다.
11 % v1과 v2를 적절하게 선택하고 이를 실행한 후 th2와 t를 디스플레이하여 두 영상이 유사한지 확인하세요.
12
13 % sol:
14 % (0, 0), (0.3, 0.6), (0.8, 0.7), (1, 1)
15 % x, y 좌표 바꿔주면 됩니다.
16 v1 = [0 0.6 0.7 1];
17 v2 = [0 0.3 0.8 1];
18 th2=histpwl(th1, v1, v2);
19 figure, imshow([t th2]);
20
21 figure, subplot(1, 2, 1); imhist(t); title("t의 histogram");
22 subplot(1, 2, 2); imhist(th2); title("th2의 histogram");

```

```

HW4_4.m x histpwl.m x +
function out = histpwl(im, a, b)
classChanged = 0;

if(~isa(im, 'double'))
classChanged = 1;
im = im2double(im);
end

if (length(a) ~= length(b))
error("Vectors A and B must be of equal size");
end

N = length(a);
out = zeros(size(im));
for i = 1:N-1
pix = find(im>a(i) & im<a(i+1));
out(pix) = (im(pix) - a(i)) * (b(i+1)-b(i))/(a(i+1)-a(i)) + b(i);
end

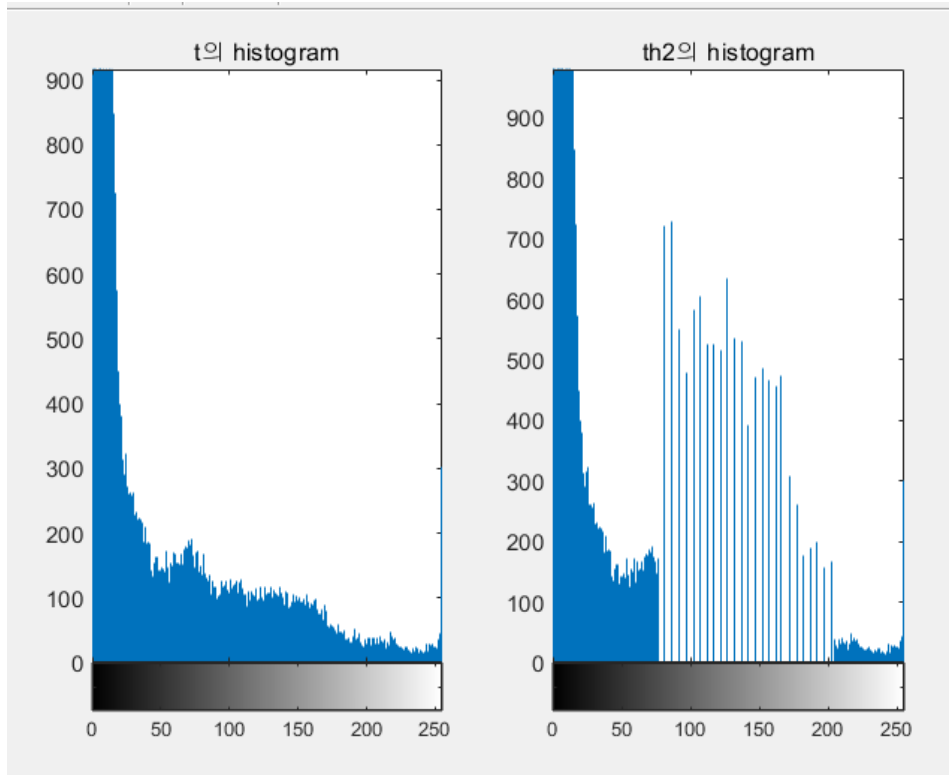
pix = find(im == a(N));
out(pix) = b(N);

if(classChanged == 1)
out= im2uint8(out);
end

```

<- histpwl.m



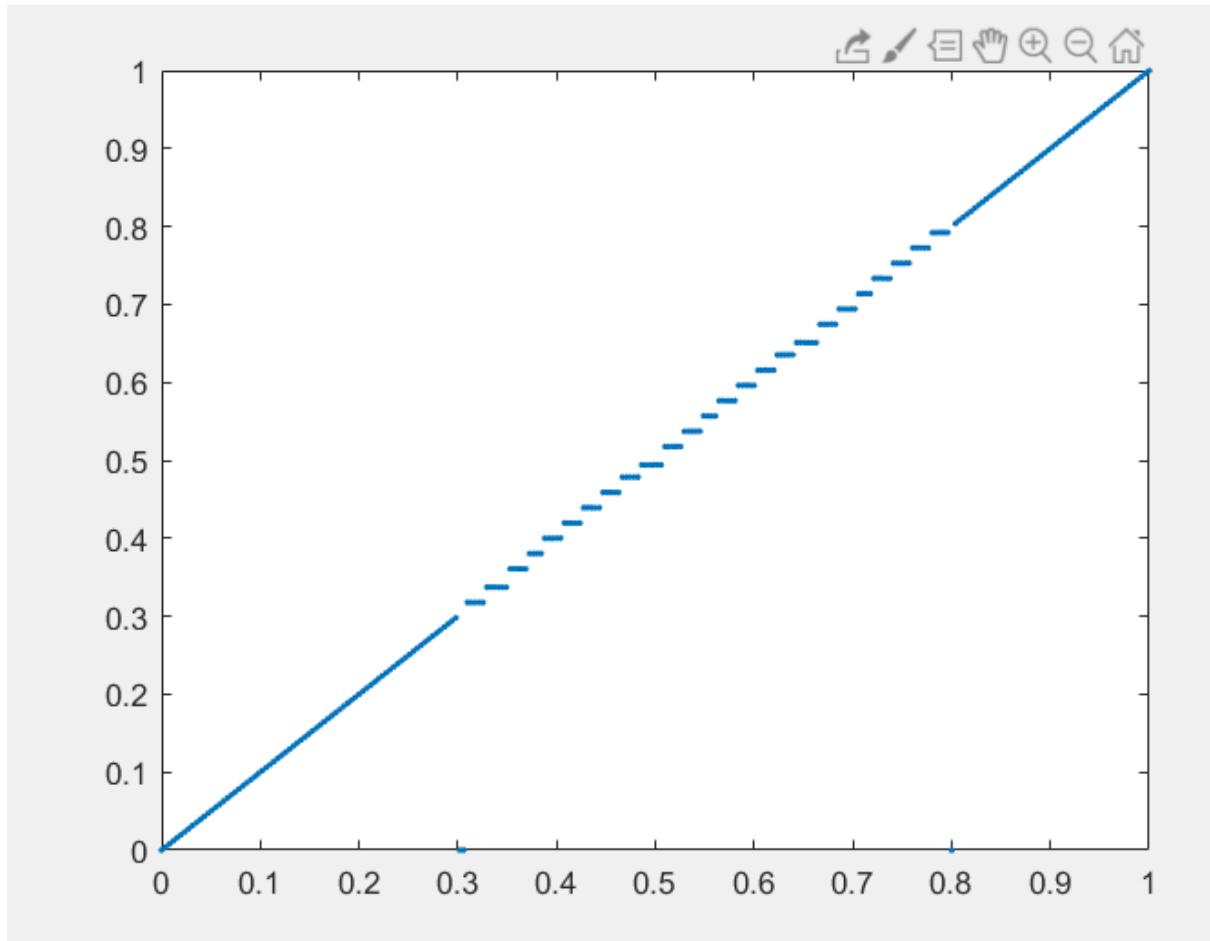


분포는 얼추 맞지만 밝기 값들이 일정 간격으로 그룹으로 묶여 하나의 밝기 값으로 합쳐진 모습을 볼 수 있었습니다.

#### (추가)

제대로 복구가 된 건지 확인을 하기 위해 앞서 2번에서 밝기 값의 변화를 관찰할 때 썼던 코드를 다시 사용해봤습니다.

```
>> HW4_4
>> figure, plot(double(t(:))/255, double(th2(:))/255, 'r.')
>>
```



매핑은 제대로 되었지만 중간 대역에서 정보 손실이 다소 발생한 것을 관찰할 수 있었습니다.