1 - NN코드

% 먼저 scale 변수에 값을 입력하고 스크립트를 실행할 것 (이미지 스케일 배수 입력)

%original\_iamge : 제공받은 1/4축소 이미지

%compare\_image : 제공받은 원본 이미지

%scaled\_image : 계산해낸 출력 이미지

scale = 4;

original\_image=imread('C:\Users\jh242\Desktop\전자공학과\디지털영상처리\Set5\_원본\_resized\small\[크기변환]woman.png'); %오리지널 이미지 읽어들이기

compare\_image=imread('C:\Users\jh242\Desktop\전자공학과\디지털영상처리\Set5\_원본\_resized\원본\woman.png'); %PSNR 계산을 위한 비교 이미지 읽어들이기

[height,width,c]=size(original\_image); %이미지 높이, 너비 읽기

new\_height=scale\*height; %스케일된 높이 읽기

new\_width=scale\*width; %스케일된 너비 읽기

scaled\_image=[new\_height,new\_width,c]; %스케일된 이미지 크기 지정

for k=1:c

for i=0:new\_height-1 %NN 적용

for j=0:new\_width-1

scaled\_image((i+1),(j+1),k)=original\_image(1+floor(i/scale),1+floor(j/scale),k);

end

end

end

peaksnr = psnr(uint8(scaled\_image),compare\_image) %PSNR 출력

figure %오리지널 이미지 출력

image(original\_image,'CDataMapping','scaled')

title([num2str(height),' \* ',num2str(width),' Original Image']);

figure %NN이미지 출력

image(uint8(scaled\_image),'CDataMapping','scaled')

title([num2str(new\_height),' \* ',num2str(new\_width),' (Image Scale = ',num2str(scale),')']);

2 – Bilinear 코드

function out=BilinearInterpolation2(im,s)

%원본 영상의 크기 읽어들이기

[r c d]=size(im); %r:original rows, c:original cols, d:image data

%s배만큼 scale된 영상의 너비,높이

rn=floor(s\*r); %resized row

cn=floor(s\*c); %resized col

scaled\_im=zeros(rn,cn,d); %출력시킬 영상값 초기화

for i=1:rn

x1=cast(floor(i/s),'uint32'); %uint32 : 32bits 정수형 반환

x2=cast(ceil(i/s),'uint32');

if x1==0 %매트랩은 (1,1)부터이므로 x1,y1값이 0일 때 1로 지정해준다.

x1=1;

end

x=rem(i/s,1);

for j=1:cn

y1=cast(floor(j/s),'uint32');

y2=cast(ceil(j/s),'uint32');

if y1==0

y1=1;

end

y=rem(i/s,1);

%밝기값을 구하고자 하는 픽셀에서 가장 가까운 네 픽셀의 좌표

A=im(x1,y1,:);

B=im(x2,y1,:);

C=im(x1,y2,:);

D=im(x2,y2,:);

%수학식을 이용한 Bilinear Interpolation

scaled\_im(i,j,:)=A\*(1-x)\*(1-y)+B\*x\*(1-y)+C\*(1-x)\*y+D\*x\*y;

end

end

%scale된 영상 출력하기

out=cast(scaled\_im,'uint8'); %8bits 정수형 반환

end

3 – Bicubic 코드

function P=padding(image)

[r,c,d] = size(image);

P = zeros(r+4,c+4,d);

%0으로된 (r+4)x(c+4)행렬 가운데 image pixel값 넣기

P(3:r+2,3:c+2,:) = image;

%image 경계값을 바로위에 한번씩 더 넣기

P(3:r+2,2,:)=image(:,1,:);

P(3:r+2,c+3,:)=image(:,c,:);

P(2,3:c+2,:)=image(1,:,:);

P(r+3,3:c+2,:)=image(r,:,:);

P(3:r+2,1,:)=image(:,1,:);

P(3:r+2,c+4,:)=image(:,c,:);

P(1,3:c+2,:)=image(1,:,:);

P(r+4,3:c+2,:)=image(r,:,:);

%모서리 부분 값 넣기

P(2,2)=image(1,1);P(2,c+3)=image(1,c);

P(r+3,2)=image(r,1);P(r+3,c+3)=image(r,c);

function [yint]=Newtint(x,y,xx)

%xx:내삽하는 값

%yint:내삽했을 때 값

n=length(x);

b=zeros(n,n);

b(:,1)=y(:);

for j=2:n

for i=1:n-j+1

b(i,j)=(b(i+1,j-1)-b(i,j-1))/(x(i+j-1)-x(i));

end %newton raphson method 수식

end

g=1;

f=b(1,1);% 값 초기화

for j=2:n

g=g\*(xx-x(j-1)); % 3차 함수에 내삽값을 넣음.

f=f+b(1,j)\*g; %내삽값을 계산함.

end

yint=f;

function [out]=mybicubic2(image,zoom)

%image=imread('woman2.png');

%zoom=4;

[r,c,d] = size(image); %image 높이, 너비

rn = round(zoom\*r); %스케일된 image높이

cn = round(zoom\*c); %스케일된 image 너비

s = zoom;

out = zeros(rn,cn,d); %출력을 0으로 초기화

P=padding(image); %image padding

P = cast(P,'double'); %계산을 위해 double로 형변환

for m = 1:rn

x1=floor(m/s); %원본 image에서 주변 4개중 가장 작은 row

for n = 1:cn

y1=floor(n/s); %원본 image에서 주변 4개중 가장 작은 col

for d=1:3

m1 = floor(s\*(x1-1)); m2 = floor(s\*(x1));

m3 = floor(s\*(x1+1)); m4 = floor(s\*(x1+2));

%위치를 알려주기 위한 가상의 row값

mm=[m1,m2,m3,m4];

n1 = floor(s\*(y1-1)); n2 = floor(s\*(y1));

n3 = floor(s\*(y1+1)); n4 = floor(s\*(y1+2));

%위치를 알려주기 위한 가상의 row값

nn=[n1,n2,n3,n4];

%같은 row위치의 밝기 값 4개씩

sample1 = P(x1+1,y1+1:y1+4,d);

sample2 = P(x1+2,y1+1:y1+4,d);

sample3 = P(x1+3,y1+1:y1+4,d);

sample4 = P(x1+4,y1+1:y1+4,d);

%위의 밝기 값으로 3차 함수를 피팅한뒤 n값을 넣어 밝기 구하기

yint1=Newtint(nn,sample1,n);

yint2=Newtint(nn,sample2,n);

yint3=Newtint(nn,sample3,n);

yint4=Newtint(nn,sample4,n);

yint=[yint1,yint2,yint3,yint4];

%위의 밝기 값으로 3차 함수 피팅한뒤 m값을 넣어 밝기 구하기

out(m,n,d)=Newtint(mm,yint,m);

end

end

end

out = cast(out,'uint8');

Ibic=imresize(image,1.4,'bicubic'); %내장함수와 비교

subplot(1,3,1); imshow(image); title('Original');

subplot(1,3,2); imshow(Ibic); title('Built-in function');

subplot(1,3,3); imshow(out); title('MyBicubic');

4 – PSNR 코드

function PSNR\_Value = PsnrCal(I ,K)

tic; % 처리속도 측정

%c=imread('changebaby.png'); % 변환할 이미지 읽기

%K=imresize(c,4,'bicubic'); %내장함수를 이용한 imageresize

%nearest

%bilinear

%bicubic

%K = imread('changeimage.png');

%I=imread('baby.png'); % 변환된 이미지와 크기가 같은 원본 이미지

%Builnt\_in\_psnr = psnr(I,K); %내장함수를 이용하여 psnr을 구한뒤 psnr이란 변수에 저장

[rows columns ~] = size(I); %color image의 경우 3차원 배열

%각 차원의 길이를 개별 변수에 할당

% RED, BLUE, GREEN 각각의 mse값을 구한다.

% 이미지와 같이 int8형과 같은 배열은 double 함수를 이용하여 데이터타입을 변환하여 연산

% int8형은 연산 불가능

mseRImage = (double(I(:,:,1)) - double(K(:,:,1))) .^ 2;

mseGImage = (double(I(:,:,2)) - double(K(:,:,2))) .^ 2;

mseBImage = (double(I(:,:,3)) - double(K(:,:,3))) .^ 2;

% sum : 배열 요소의 합

% 행렬 배열의 경우 sum을 하면 각열의 합이 포함된 행 벡터를 반환 -> 행 벡터로 저장 (1 \* 512)

% 행 벡터로 저장되어 있는 배열을 한번더 sum을 하여 하나의 값으로 반환한다.

mseR = sum(sum(mseRImage)) / (rows \* columns);

mseG = sum(sum(mseGImage)) / (rows \* columns);

mseB = sum(sum(mseBImage)) / (rows \* columns);

% RED, GREEN,BLUE 각각 구한 mse의 평균을 구한다.

mse = (mseR + mseG + mseB)/3;

% PSNR 계산

PSNR\_Value = 10 \* log10( 255^2 / mse);

%하나의 창에 여러 사진 displayPsnrCal

subplot(1,2,1)

imshow(I); title('Original image');

subplot(1,2,2)

imshow(K); title(['The PSNR\_Value is ',num2str(PSNR\_Value)]);

toc;

5 – SOTA

