18.11.2020

Robot Vision

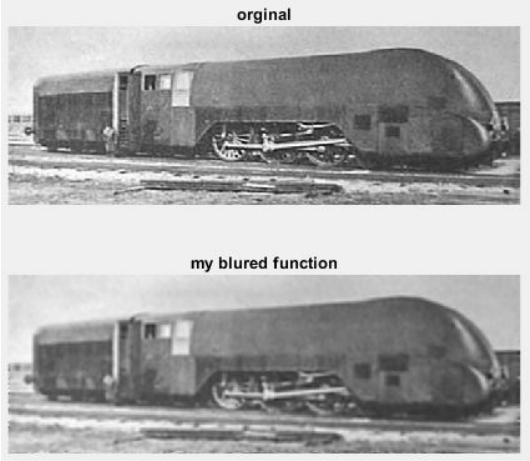
RV ex3 Linear image filtering

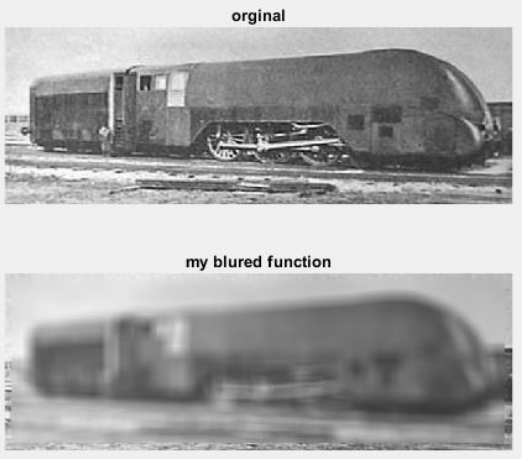
Mateusz Warmuz

18.11.2020

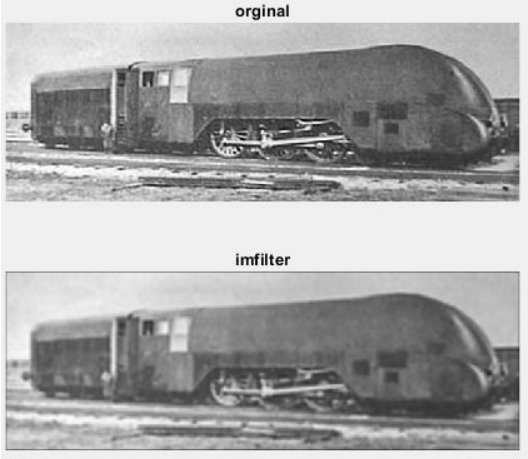
Task 2 Implement algorithm of universal image linear filter.

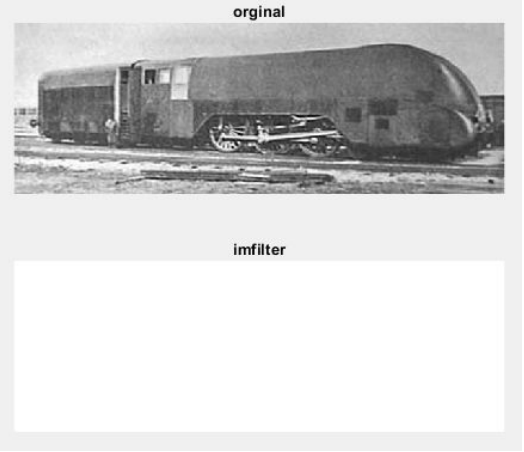
Comparison of original image and my blurred function





Comparison of my blurred function to matlab function imfilter





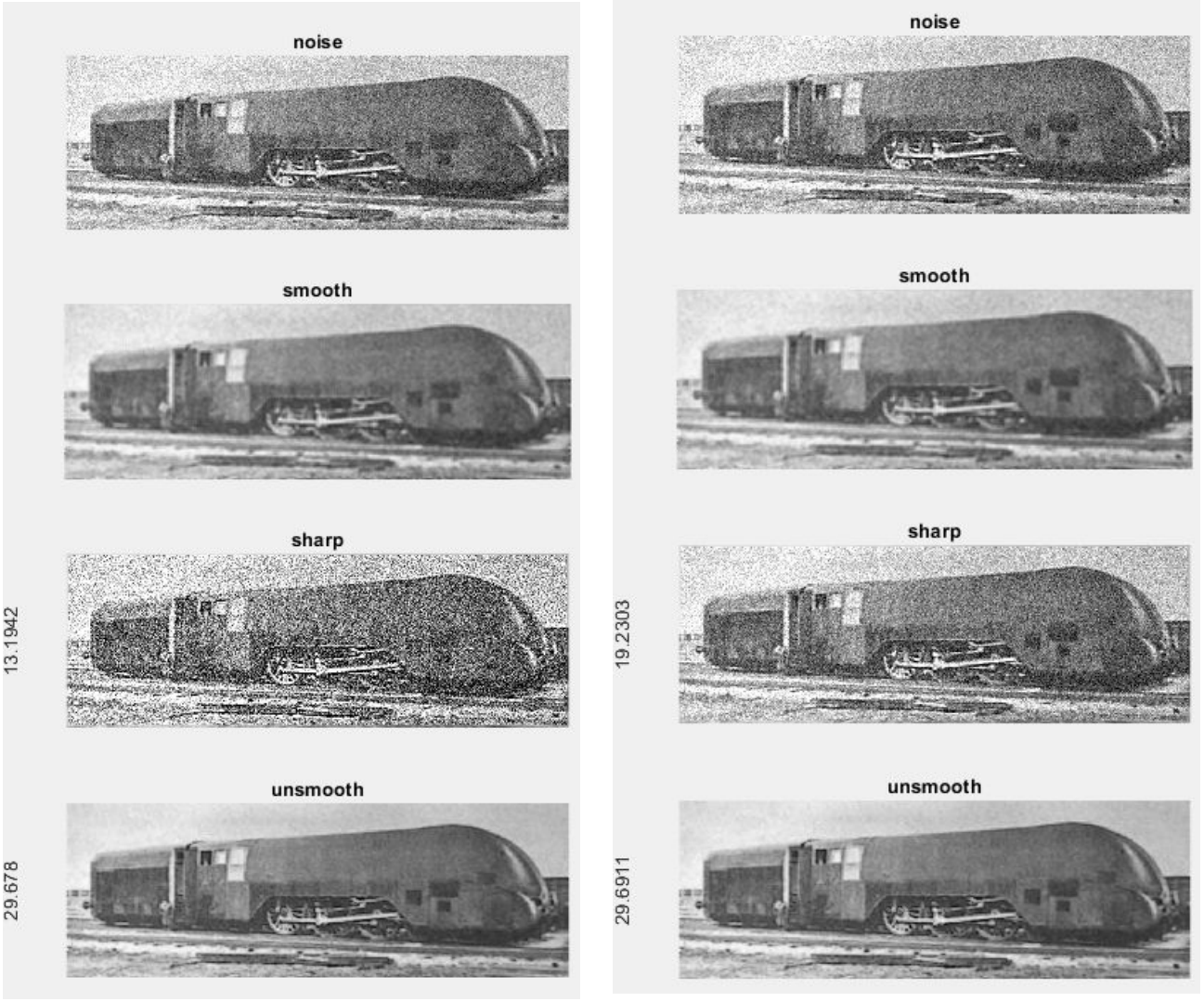
The masks are 5x5 and 17x17 respectively. We can see that increasing the size of the mask, the image becomes more blurred. Also we can notice some strange frame appear on the image due to an algorithm which does not take into account edge values of the original picture. Simply the frame is not blurred. Also we can notice that the imfilter function with a high mask makes the image become invisible.

Code:

```
photo=imread('test4.jpg');
grey=rgb2gray(photo);
grey2=rgb2gray(photo);
grey3=rgb2gray(photo);

% task 2

%mask 5x5

h = [1, 1, 1, 1, 1;
    1, 1, 1, 1, 1;
    1, 1, 1, 1, 1;
    1, 1, 1, 1, 1;
    1, 1, 1, 1, 1]/25;

% h1 = ones(17,17);
% sizeofmask1 = size(h,1)/2-0.5;
% sizeofmask2 = size(h1,1)/2-0.5;

point = 2;

subplot(2,1,1)
imshow(grey);
title('orginal');
for row = (point+1):size(grey,1)-(point+1)
   for col = (point+1):size(grey,2)-(point+1)
      grey2(row,col)=conv2(grey((row-point):(row+point),(col-point):(col+point)),h, 'valid')/sum(h,'all');
   end
end
% subplot(2,1,1)
% imshow(grey3);
% title('orginal');
subplot(2,1,2)
imshow(grey2);
title('my blured function');

new = imfilter(grey3, h);
subplot(2,1,2)
imshow(new);
title('imfilter')
```

18.11.2020

Task 4 Implement Gaussian noise reduction method based on a sequence of linear filters.



The only difference on the above images is that the first sharp picture has a small middle point (equal 13). The mask is 5x5. The second picture is has a significantly huge middle point of mask (equal =100), Comparing two images of sharp filters we can observe that the image for the second picture looks better than the first, also my observation confirms peak signal to ratio calculation which is higher than in first image. Additionally I compare a sharp filter with smooth. We can observe that the unsmooth method works significantly better than sharp.

Code:

```
photo=imread('test4.jpg');
grey=rgb2gray(photo);
grey2=rgb2gray(photo);
grey3=rgb2gray(photo);
grey_org=rgb2gray(photo);
%task 4
h = [1, 1, 1, 1, 1;
     1, 1, 1, 1, 1;
     1, 1, 1, 1, 1;
     1, 1, 1, 1, 1;
     1, 1, 1, 1, 1]/25;

 point = 2;

noise = imnoise(grey,'gaussian',0.02);
subplot(4,1,1);
imshow(noise);
title('noise');
for row = (point+1):size(grey,1)-(point+1)
    for col = (point+1):size(grey,2)-(point+1)
        grey2(row,col)=conv2(noise((row-point):(row+point),(col-point):(col+point)),h, 'valid')/sum(h,'all');
    end
end
subplot(4,1,2);
imshow(grey2);
title('smooth');

h_sharp = [0, 0, -1, 0, 0;
           0, -1, -1, -1, 0;
           -1, -1, 100, -1, -1;
           0, -1, -1, -1, 0;
           0, 0, -1, 0, 0];

for row = (point+1):size(grey,1)-(point+1)
    for col = (point+1):size(grey,2)-(point+1)
        grey3(row,col)=conv2(noise((row-point):(row+point),(col-point):(col+point)),h_sharp,
'valid')/sum(h_sharp,'all');
    end
end
Peak_signal_to_noise_ratio=psnr(grey3, grey);
subplot(4,1,3);
imshow(grey3);
title('sharp');
ylabel(Peak_signal_to_noise_ratio);

%task 6
alfa = 1;

U = grey_org - alfa*(grey_org-grey2);
subplot(4,1,4);
imshow(U);
title('unsmooth')
ylabel(psnr(U,grey_org));
```

18.11.2020

Task 6 Implement method of unsharp masking. Compare results with results of contrast enhancement obtained using methods from Ex. 3.

orginal

unsharped

Stretching image

It’s hard to tell If something significantly differs from each other. Or the difference is imperceptible comparing stretching an image with original. On unsharpened method we can some very small blurred picture than in original.

code:

```
photo=imread('test4.jpg');
grey=rgb2gray(photo);
grey1=rgb2gray(photo);
grey2=rgb2gray(photo);
grey3=rgb2gray(photo);

% task 6

%mask 5x5

h = [1, 1, 1, 1, 1;
     1, 1, 1, 1, 1;
     1, 1, 1, 1, 1;
     1, 1, 1, 1, 1;
     1, 1, 1, 1, 1]/25;

 point = 2;

subplot(3,1,1)
imshow(grey);
title('orginal');
for row = (point+1):size(grey1,1)-(point+1)
    for col = (point+1):size(grey1,2)-(point+1)
        grey2(row,col)=conv2(grey1((row-point):(row+point),(col-point):(col+point)),h, 'valid')/sum(h,'all');
    end
end
alfa = 1;

U = grey1 - alfa*(grey-grey2);
subplot(3,1,2)
imshow(U);
title('unsharped');

%Stretching

minimum = min(grey3,[],'all');
maximum = max(grey3,[],'all');

for i = 1:size(grey3,1)
    for j = 1:size(grey3,2)
        grey3(i,j) = 255*double(grey3(i,j)-minimum)/(maximum-minimum);
    end
end

subplot(3,1,3);
imshow(grey3);
title('Stretching image');
```