[Image Processing project]

By

Mohamed Badawy Sayed
Mostafa Ahmed Hasan El-Gelany
Mahmoud Ahmed Khalel
Ahmed Nageh Abbas
Nada Maaman Abdul Karim
Basant Benyamen

Edits in this Document

- **Apply Filter on color images**
- **Using Data Store for retrieving images**
- **Compare between color and grayscale pages results (9 - 10)**

GitHub Repo

Dr.

Muhammad A. O. Khfagy

Table of Contents

[Histogram Equalization]

Abstract:

histogram equalization is a method in image proccessing baesd on histogram of an image. it is used especially when the image is represented by a narrow range of intensity values.
its main functionality is to make the image's pixels have better distributed on the histogram.c

[difference of gaussian (DOG)]

Abstract:

In order to explain the DOG filter, we should first explain gaussian filter.

Gassian filter is based on the mathematical concept of the Gaussian function, which is a bell-shaped curve.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

●π is the mathematical constant pi (approximately 3.14159).

●σ is the standard deviation of the Gaussian distribution, controlling the spread of the

curve.

●e is the mathematical constant Euler's number (approximately 2.71828).

●G(x,y) is the value of the 2D Gaussian function at position (x,y)

The primary purpose of applying a Gaussian filter to an image is to reduce noise by smoothing the image

Now let us explain DOG filter:
The Difference of Gaussians (DoG) filter is used in image processing to enhance edges and identify features at different scales.
It is obtained by subtracting one blurred version of an image from another, where each version is created using a Gaussian filter with a different standard deviation.

[MSR Filter]

Abstract:

The Multiscale Retinex algorithm is an image enhancement technique that aims to improve the contrast of an image by normalizing pixel values across different scales. The algorithm operates by applying a series of operations at multiple scales.

1- define some parameters for applying the algorithm

sigma: the standard deviation for the Gaussian filter.

scales: the scales at which the MSR algorithm operates.

filt_size: the size of the Gaussian filter.

enhanced: a matrix filled with zeros which will accumulates the Retinex components of the image.

2- iterate through each scale specified in scales and do the following:

- apply the Gaussian Filter to the image at the current scale (Which was explained in the DOG filter).
- calculate the Retinex component for the current scale.
- accumulate the Retinex component to %enhanced%

3- Normalize the enhanced image to the range [0, 255] and convert it back to uint8

[PSNR]

Abstract:

The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image.

PSNR = 10log10((L-1)^2/MSE

Here, L is the number of maximum possible intensity levels (minimum intensity level suppose to be 0) in an image.

MSE = 1/m*n ΣΣ(O(i,j)-D(i,j))^2

Where, O represents the matrix data of original image. D represents the matrix data of degraded image. m represents the numbers of rows of pixels and i represents the index of that row of the image. n represents the number of columns of pixels and j represents the index of that column of the image.

[SSIM]

Abstract:

Structural Similarity Index is a metric used to measure the similarity between two images. SSIM takes into account luminance, contrast, and structure, which are important aspects of human perception. The index produces a value between -1 and 1, where 1 indicates perfect similarity, -1 indicates perfect dissimilarity, and 0 means no similarity.

SSIM is widely used in image processing and computer vision to evaluate the quality of compressed images, assess the impact of image processing algorithms, or compare the similarity between an original image and a processed one. It provides a more comprehensive assessment than traditional metrics like Mean Squared Error (MSE), as it considers both global and local variations in image structure.

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

contrast

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

structure

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

luminance

Code :

```
input=datastore("dataset");
gaussian1 = fspecial('Gaussian', 10, 1.07);
gaussian2 = fspecial('Gaussian', 10, 1);
dog = gaussian1 - gaussian2;
sigma = 5;
scales = [15, 80, 250];
filt_size = 2 * ceil(3 * sigma) + 1;
Psnr_histo=0;
Psnr_MSR=0;
Psnr_DoG=0;
SSIM_histo=0;
SSIM_MSR=0;
SSIM_DoG=0;

for i=1:length(input.Files)
    img=input.read();
    img=rgb2ycbcr(img);
    img2=img;
    GrayLevel=img(:,:,1);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%');


    %%%%%%%%%%%%% apply histogram%%%%%%%%%%%%%%%%%
    HistImage=histeq(GrayLevel);
    img(:,:,1)=HistImage;
    imwrite(ycbcr2rgb(img),['Histogram/img' int2str(i) '.jpg']);
    Psnr_histo=Psnr_histo+psnr(img,img2);
    SSIM_histo=SSIM_histo+ssim(img,img2);
    disp (['psnr for image ' int2str(i) ' after apply Histogram equalization equals
to '  num2str(psnr(img,img2))]);
    disp (['ssim for image ' int2str(i) ' after apply Histogram equalization equals
to '  num2str(ssim(img,img2))]);



    %%%%%%%%%%%%% apply DoG %%%%%%%%%%%%%%%%%%%%%%
    DoGImage=conv2(double(GrayLevel), dog, 'same');
    DoGImage=uint8(255*DoGImage);
    img(:,:,1)=DoGImage;
    imwrite(ycbcr2rgb(img),['DOG/img' int2str(i) '.jpg']);
    Psnr_DoG=Psnr_DoG+psnr(img,img2);
    SSIM_DoG=SSIM_DoG+ssim(img,img2);
    disp (['psnr for image ' int2str(i) ' after apply DoG equals to '
num2str(psnr(img,img2))]);
    disp (['ssim for image ' int2str(i) ' after apply DoG equals to '
num2str(ssim(img,img2))]);
```

```
    %%%%%%%%%%%%% apply MSR %%%%%%%%%%%%%%%%%%%%%
    Img = double(GrayLevel);
    enhanced = zeros(size(Img));
    for j = 1:length(scales)
        gaussian_filter = fspecial('gaussian', [filt_size, filt_size], sigma *
scales(j));
        blurred_image = imfilter(Img, gaussian_filter, 'conv');

        retinex_component = log(Img + 1) - log(blurred_image + 1);

        enhanced = enhanced + retinex_component;
    end
    enhanced = uint8( (enhanced - min(enhanced(:))) / (max(enhanced(:)) -
min(enhanced(:))) * 255 );
    img(:,:,1)=enhanced;
    imwrite(ycbcr2rgb(img),['MSR/img' int2str(i) '.jpg']);
    Psnr_MSR=Psnr_MSR+psnr(img,img2);
    SSIM_MSR=SSIM_MSR+ssim(img,img2);
    disp (['psnr for image ' int2str(i) ' after apply MSR equals to '
num2str(psnr(img,img2))]);
    disp (['ssim for image ' int2str(i) ' after apply MSR equals to '
num2str(ssim(img,img2))]);


    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%');


end
Psnr_MSR=Psnr_MSR/length(input.Files);
Psnr_DoG=Psnr_DoG/length(input.Files);
Psnr_histo=Psnr_histo/length(input.Files);
SSIM_MSR=SSIM_MSR/length(input.Files);
SSIM_DoG=SSIM_DoG/length(input.Files);
SSIM_histo=SSIM_histo/length(input.Files);
disp(['average Psnr for histogram equalization is ' num2str(Psnr_histo)]);
disp(['average Psnrfor MSR is ' num2str(Psnr_MSR)]);
disp(['average Psnr for DoG is ' num2str(Psnr_DoG)]);
disp(['average ssim for histogram equalization is ' num2str(SSIM_histo)]);
disp(['average ssim for MSR is ' num2str(SSIM_MSR)]);
disp(['average ssim for DoG is ' num2str(SSIM_DoG)]);
```

[Image Number 1 - grayscale]

Original image			
Filters	 histogram equalization	 difference of gaussian	 MSR Filter
PSNR	PSNR: 18.37 dB	PSNR: 6.44 dB	PSNR: 15.28 dB
SSIM	hist: 0.8719	-0.0335	0.7834

[Image Number 1 – colored]

Original image

Filters	 histogram equalization	 difference of gaussian	 MSR Filter
PSNR	PSNR: 21.5347 dB	PSNR: 11.9755 dB	PSNR: 20.7709 dB
SSIM	0.84837	0.22967	0.57473

[Image Number 2]

Original image

Filters

		
histogram equalization	difference of gaussian	MSR Filter

	histogram equalization	difference of gaussian	MSR Filter
PSNR	PSNR: 17.33 dB	PSNR: 4.37 dB	PSNR: 11.53 dB
SSIM	0.7424	-0.0421	0.8173

[Image Number 3]

Original image			
Filters	 histogram equalization	 difference of gaussian	 MSR Filter
PSNR	PSNR: 11.41 dB	PSNR: 8.11 dB	PSNR: 7.93 dB
SSIM	0.7316	-0.0752	0.5823

[Image Number 4]

Original image

Filters

histogram equalization

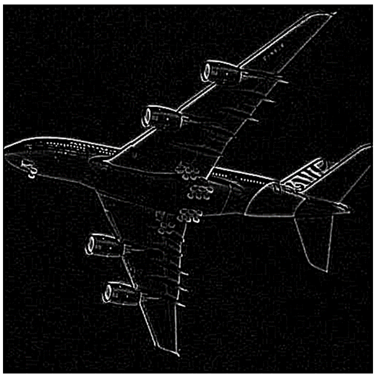
difference of gaussian

MSR Filter

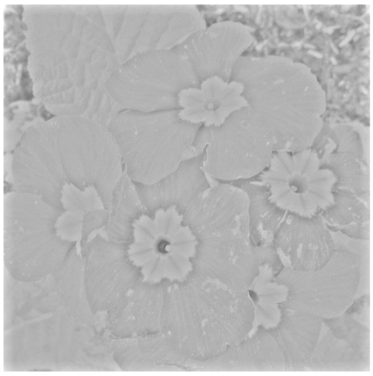
	histogram equalization	difference of gaussian	MSR Filter
PSNR	PSNR: 19.16 dB	PSNR: 5.44 dB	PSNR: 11.27 dB
SSIM	0.8787	-0.0754	0.7698

[Image Number 5]

Original image

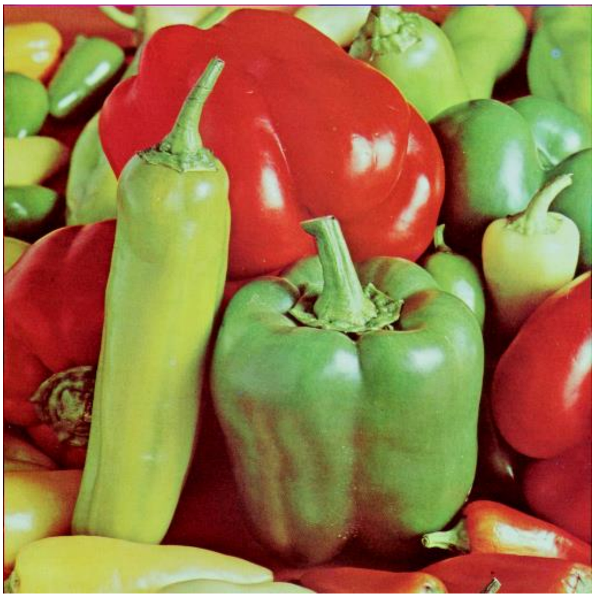
Filters	 histogram equalization	 difference of gaussian	 MSR Filter
PSNR	PSNR: 11.81 dB	PSNR: 8.13 dB	PSNR: 8.36 dB
SSIM	0.4913	0.0092	0.7423

[Image Number 6]

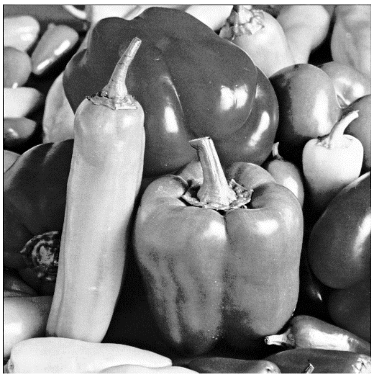
Original image			
Filters	 histogram equalization	 difference of gaussian	 MSR Filter
PSNR	PSNR: 18.18 dB	PSNR: 5.68 dB	PSNR: 11.53 dB
SSIM	0.7822	-0.0814	0.6728

[Image Number 7]

Original image

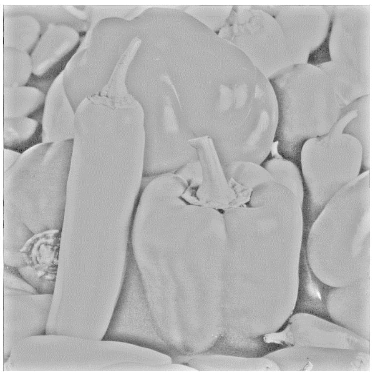


Filters



histogram equalization

difference of gaussian



MSR Filter

	histogram equalization	difference of gaussian	MSR Filter
PSNR	PSNR: 20.99 dB	PSNR: 5.73 dB	PSNR: 10.86 dB
SSIM	0.8863	-0.0452	0.6794

[Image Number 8]

Original image

Filters

histogram equalization

difference of gaussian

MSR Filter

	histogram equalization	difference of gaussian	MSR Filter
PSNR	PSNR: 13.73 dB	PSNR: 5.46 dB	PSNR: 10.71 dB
SSIM	0.8749	-0.0861	0.7262

[Image Number 9]

Original image

Filters

histogram equalization

difference of gaussian

MSR Filter

	histogram equalization	difference of gaussian	MSR Filter
PSNR	PSNR: 21.98 dB	PSNR: 4.83 dB	PSNR: 10.63 dB
SSIM	0.8343	-0.0295	0.7393

[Image Number 10]

Original image

Filters

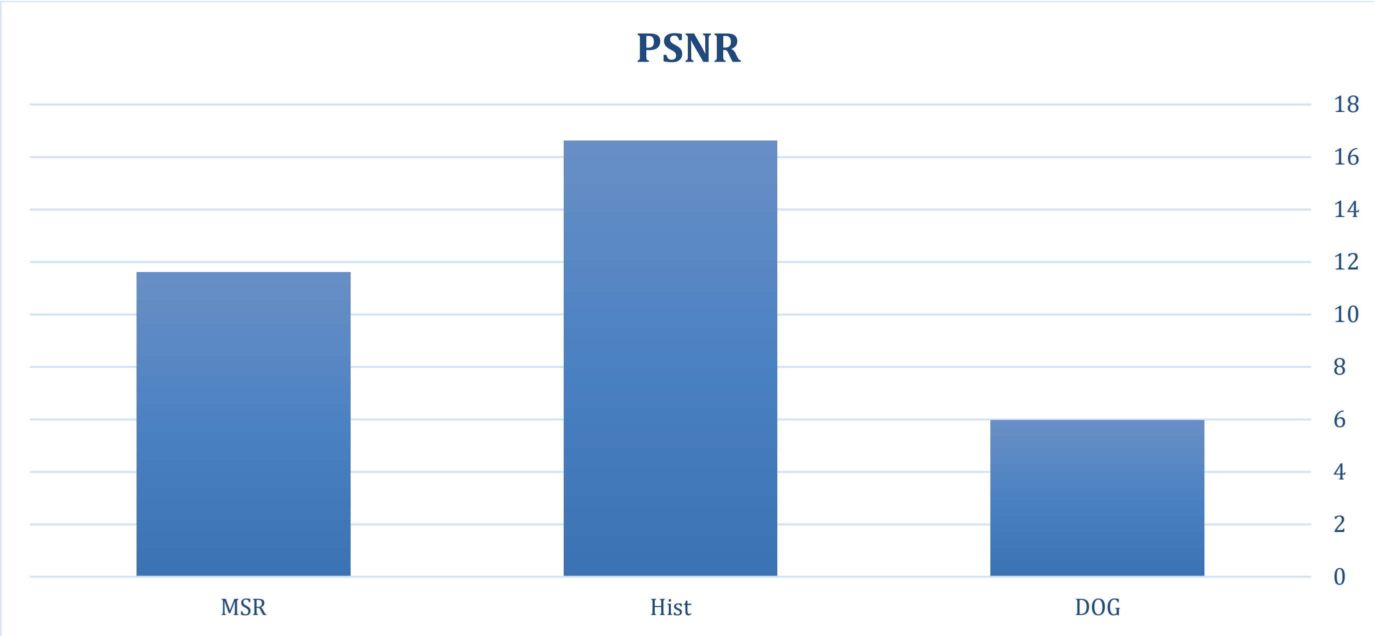
histogram equalization

difference of gaussian

MSR Filter

	histogram equalization	difference of gaussian	MSR Filter
PSNR	PSNR: 20.40 dB	PSNR: 4.82 dB	PSNR: 11.27 dB
SSIM	0.8218	-0.1541	0.5980

	DOG	Hist	MSR
PSNR	5.97	16.6246	11.6054



	DOG	Hist	MSR
SSIM	-0.0811	0.7583	0.713