

Document Smoothing

I:Blur

1: Blur (Low-Pass Filtering)

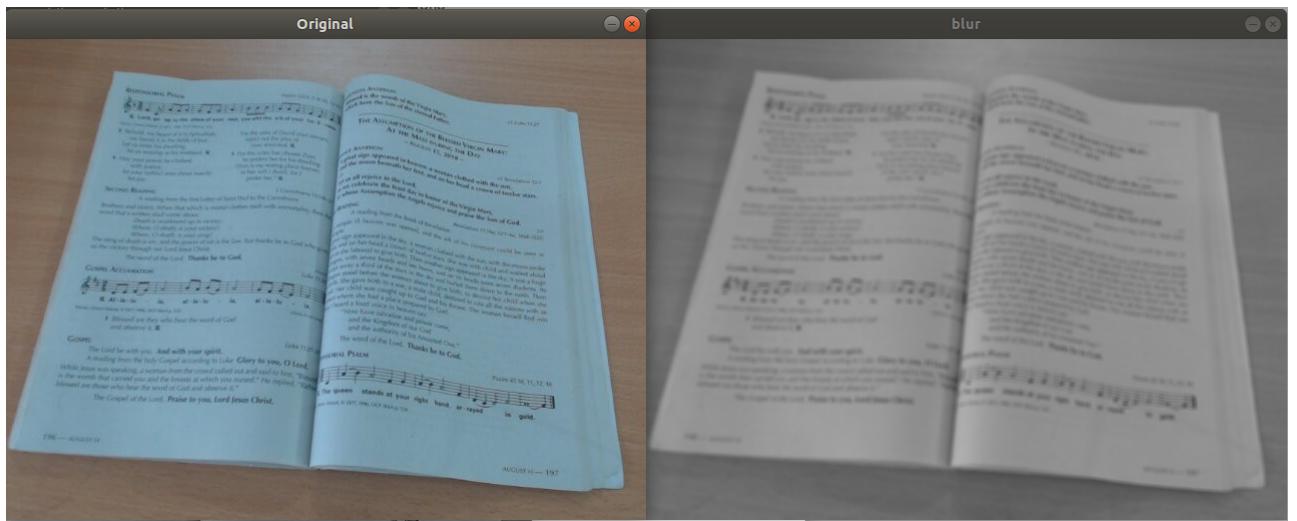
LPF

Blur

- Kernel với mỗi phần tử có giá trị bằng 1, kích thước kernel do mình chỉ định sau đó tính trung bình các pixel trong kernel.
- Hình ảnh minh họa

$$K = \frac{1}{\text{ksize.width} * \text{ksize.height}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & 1 & \cdots & 1 & 1 \\ \cdots & & & & & \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

- Ex

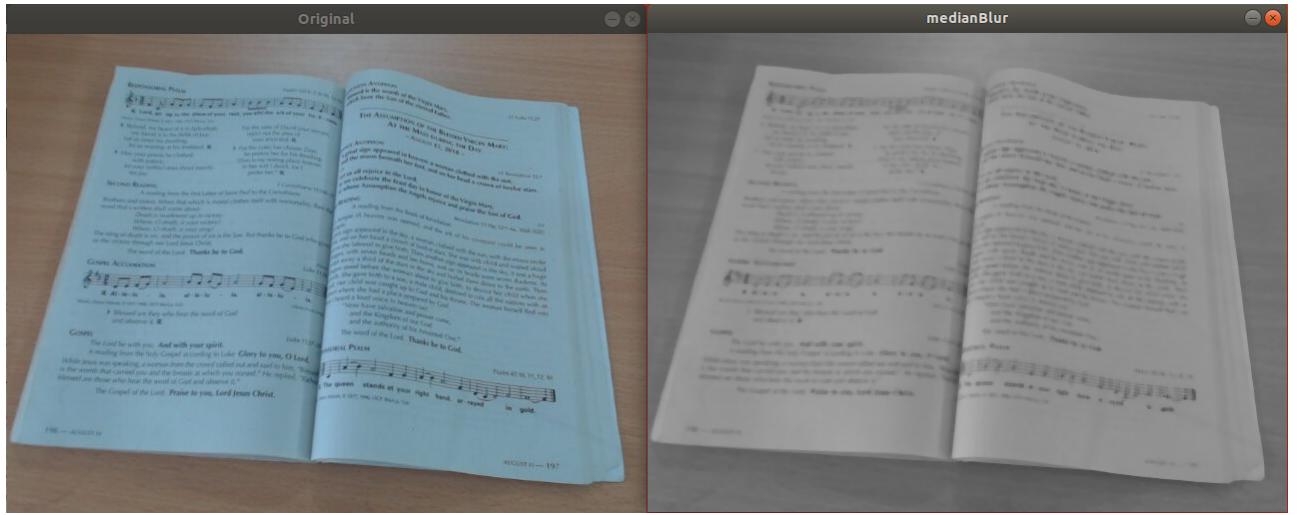


2: MedianBlur

MedianBlur

- Kernel với sẽ chuyển về mảng 1 chiều đã được sắp xếp pixel trung tâm sẽ được thay bằng trung vị của mảng.
- Ví Dụ: mảng đã được sắp xếp {3,4,5,6,7,7,8,9,10} thì median sẽ là 7.

- Ex



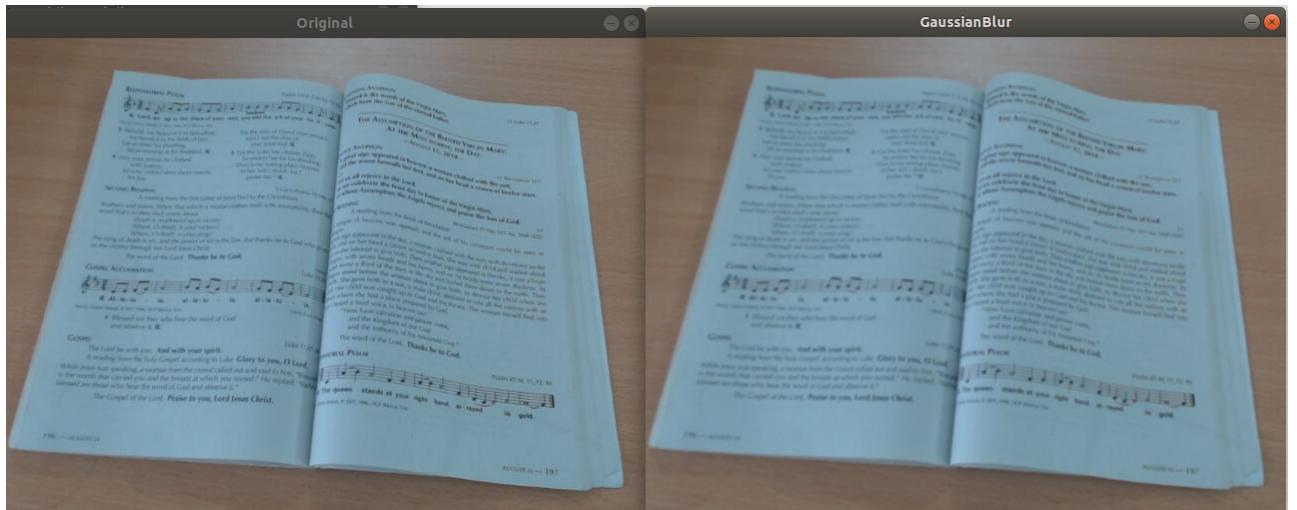
3: GaussianBlur

Gaussian Smoothing

- Kernel sau khi tính toán theo hàm Gauss sẽ có dạng ma trận mà tâm của ma trận sẽ lớn nhất và nhỏ dần khi mở rộng ra ngoài biên.
- Hình ảnh minh họa

$$\frac{1}{273} \begin{matrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{matrix}$$

- Ex

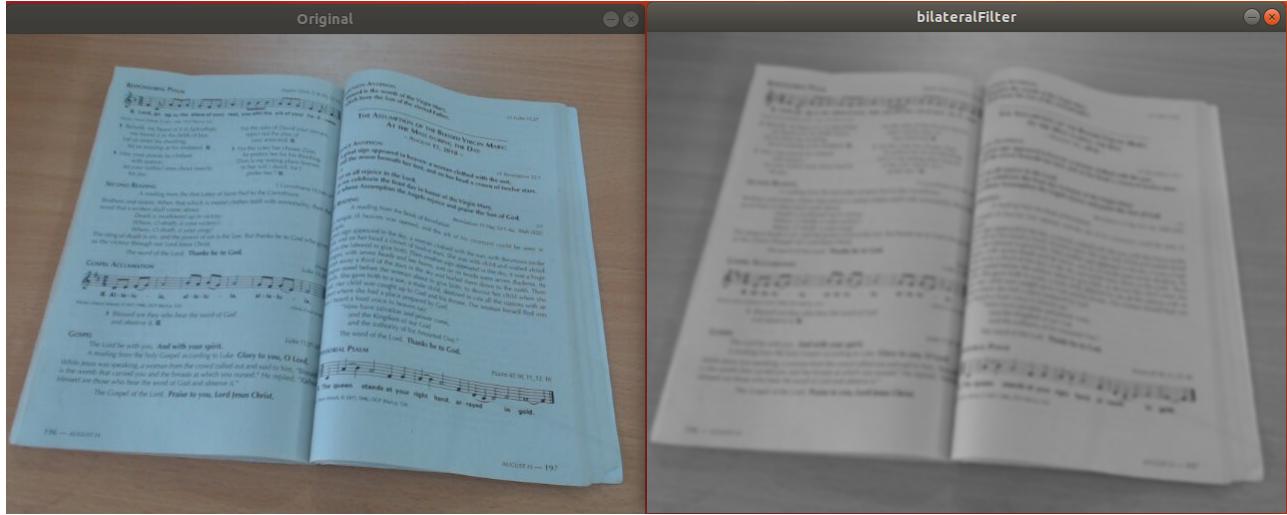


4: BilateralFilter

BilateralFilter

- Loại bỏ nhiễu nhưng vẫn giữ được biên. BilateralFilter cũng sử dụng bộ lọc Gauss nhưng có thêm một bước là nó chỉ xét những điểm có mức xám tương đồng với pixel trung tâm để làm mịn do những điểm ảnh ở biên có sự thay đổi mức xám rất rõ ràng.

Ex



5: Filter2D

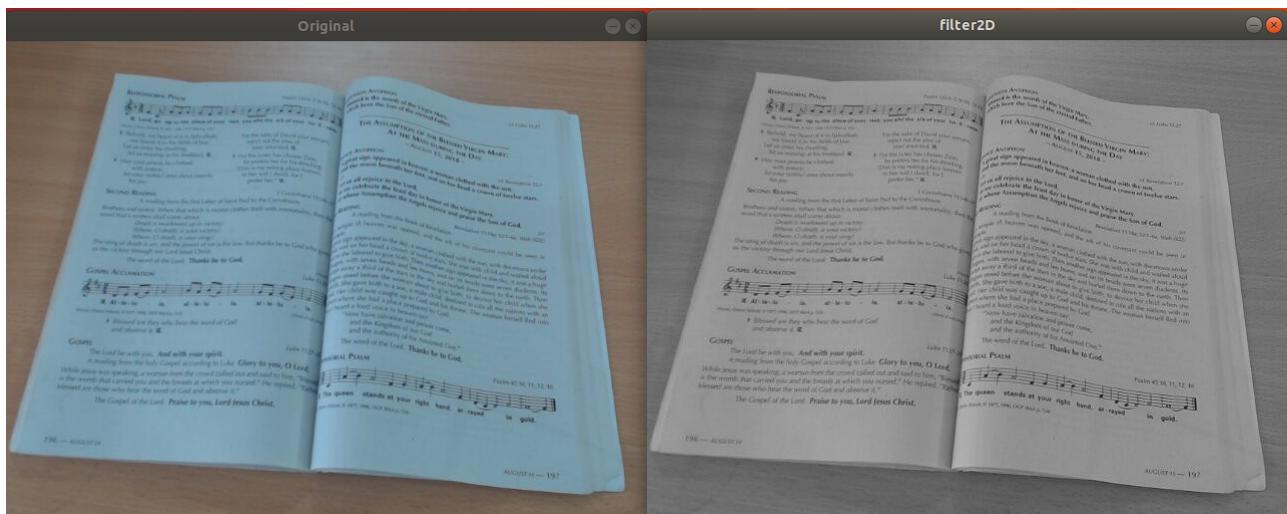
- Convolution với kernel mình tạo ra

Tạo kernel

```
float array[3][3]={{0, -0.25, 0}, {-0.25, 2, -0.25}, {0, -0.25, 0}};  
  
Mat kernel(Kernel_size, Kernel_size, CV_32F, array);
```

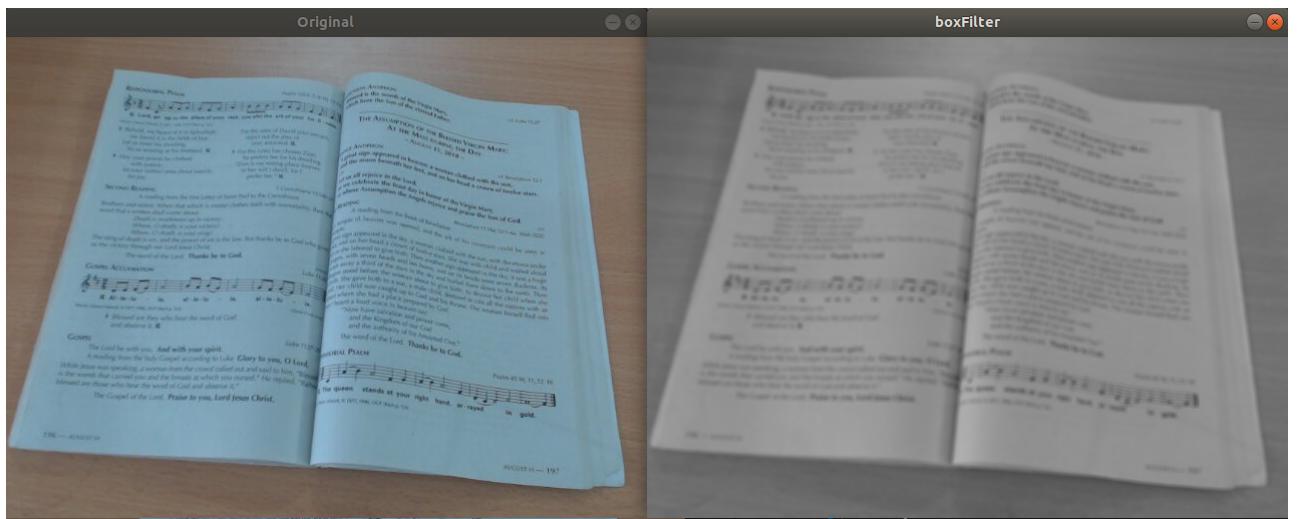
- link [Filter2D](#)

- Ex



6: BoxFilter

- Giống với Blur boxFilter cũng dùng Mat::ones/(ksizeWidth*KsizeHeight) nhưng boxfilter có thêm tham số độ sâu của ảnh và nếu normalize = False thì kernel sẽ là chỉ là Mat::ones.
- Link [BoxFilter](#)
- Ex

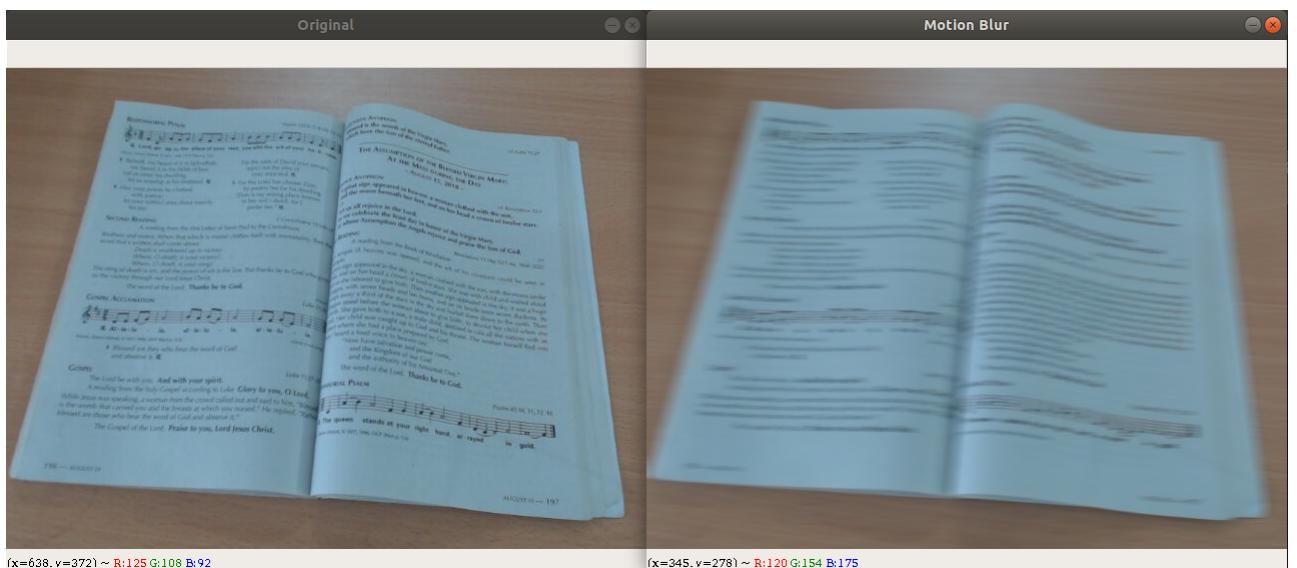


7: Motion blur

- link [MotionBlur](#)
- Motion Blur sẽ làm nhòe ảnh trông vật thể như đang di chuyển khi được chụp
- Kernel sẽ là ma trận full 0 và giá trị hàng thứ kernel_size/2 = 1/kernel_size (thường kernel_size = 15)
- VD: Kernel_size = 5

$$\begin{bmatrix} [0. & 0. & 0. & 0. & 0.] \\ [0. & 0. & 0. & 0. & 0.] \\ [0.2 & 0.2 & 0.2 & 0.2 & 0.2] \\ [0. & 0. & 0. & 0. & 0.] \\ [0. & 0. & 0. & 0. & 0.] \end{bmatrix}$$

- Ex



II: Sharpen

1: High-Pass Filtering

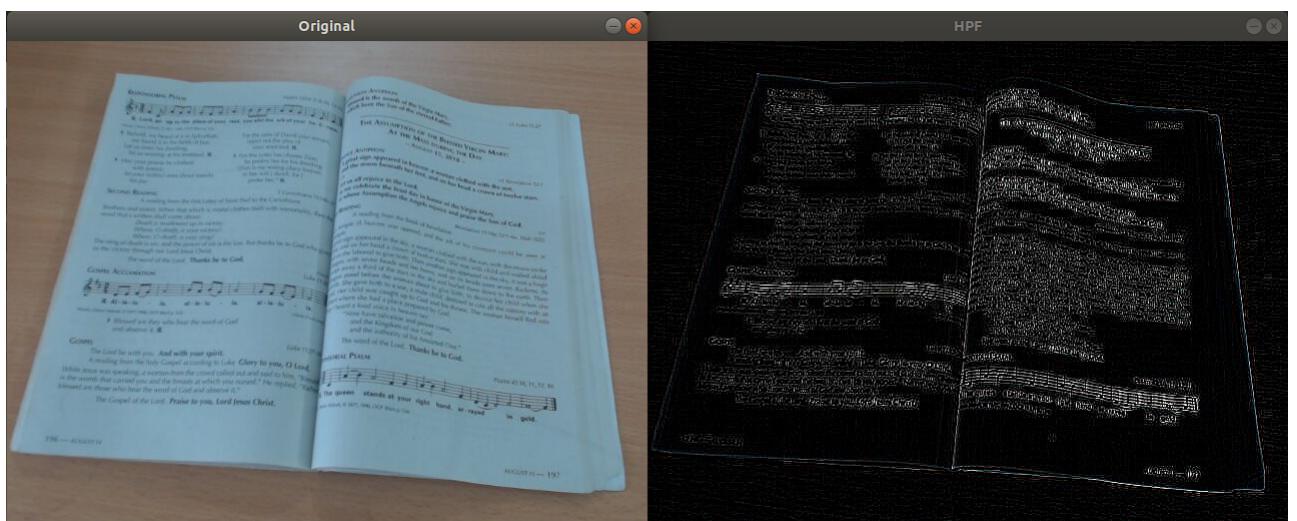
- link [HPF](#)

- Sử dụng bộ lọc với nhân là dương và các phần tử xung quanh là âm.

0	-1	0
-1	4	-1
0	-1	0

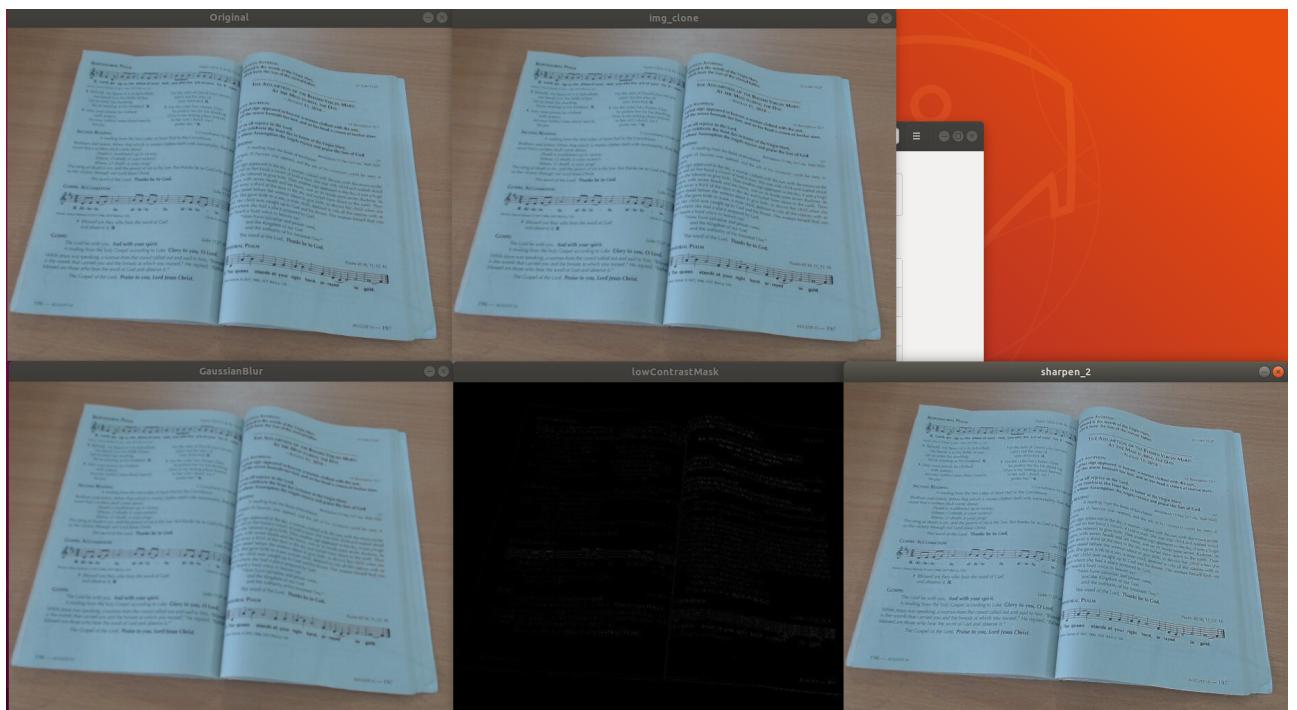
-1	-2	-1
-2	12	-2
-1	-2	-1

- Ex



2: Unsharp Filter

- link [Unsharp](#)
- Sử dụng mask bị làm mờ rồi lấy ảnh gốc trừ đi.
- Ex



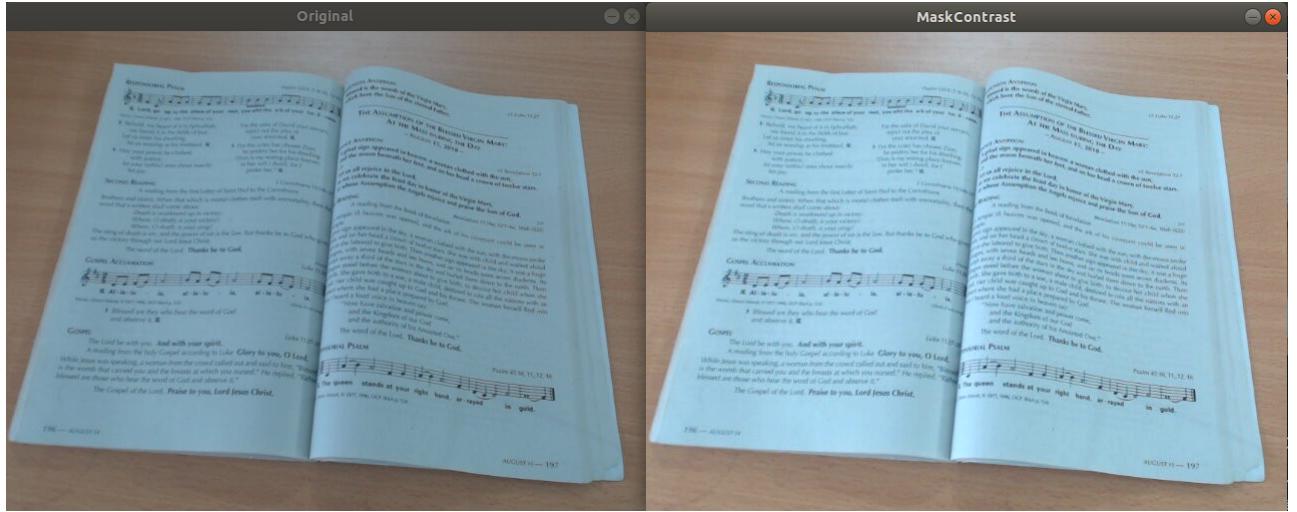
III: Contrast

1: The Contrast and Brightness of an image

- link [Contrast & Brightness](#)
- Để tăng ngưỡng sáng của ảnh lên ta sử dụng công thức:

$$g(i,j) = \alpha \cdot f(i,j) + \beta$$

- The parameters $\alpha > 0$ and β are often called the gain and bias parameters
- Ex



2: Histogram Equalization (HE)

- link [HE](#)
- Ý tưởng của HE sẽ là xếp các giá trị pixel thành các cột và san đều chúng ra bằng nhau.

- EX

2	3	2	1	2
1	2	0	2	3
0	1	1	3	1
2	3	0	1	2
0	1	2	2	0

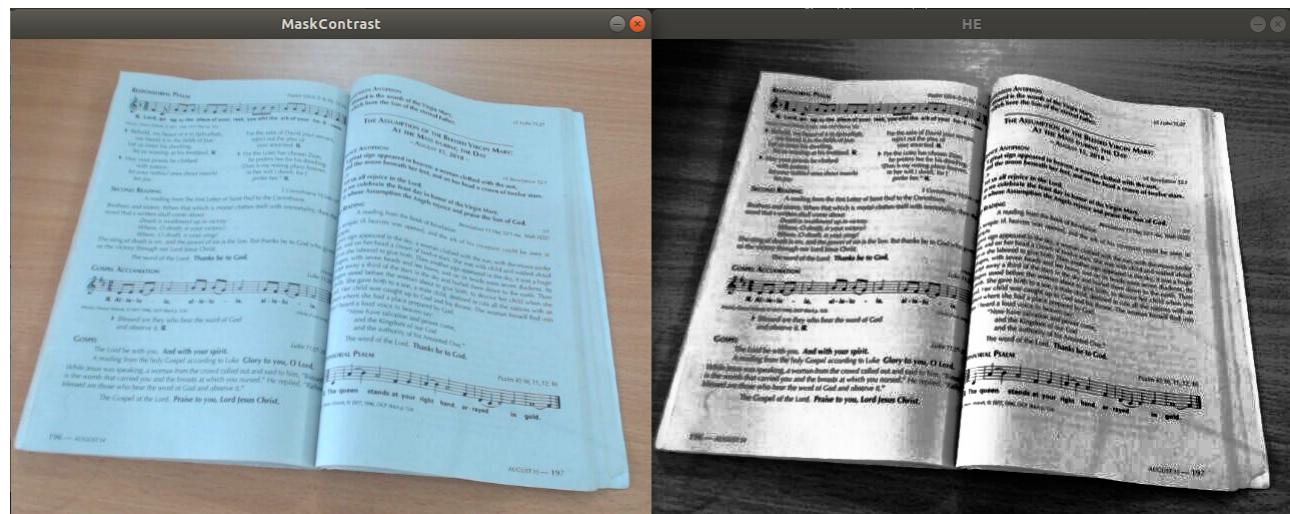
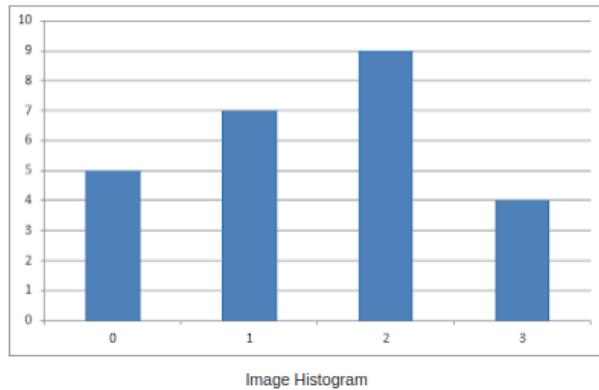
Sample Image

As you can see in the above image, there are 5 pixels with value 0, 7 pixels with value 1, 9 pixels with value 2 and 4 pixels with value 3. These information is tabulated as follows.

Pixel Value	Number of Pixels
0	5
1	7
2	9
3	4

Intensity distribution of the above image

The histogram of an image is usually presented as a graph. The following graph represents the histogram of the above image.

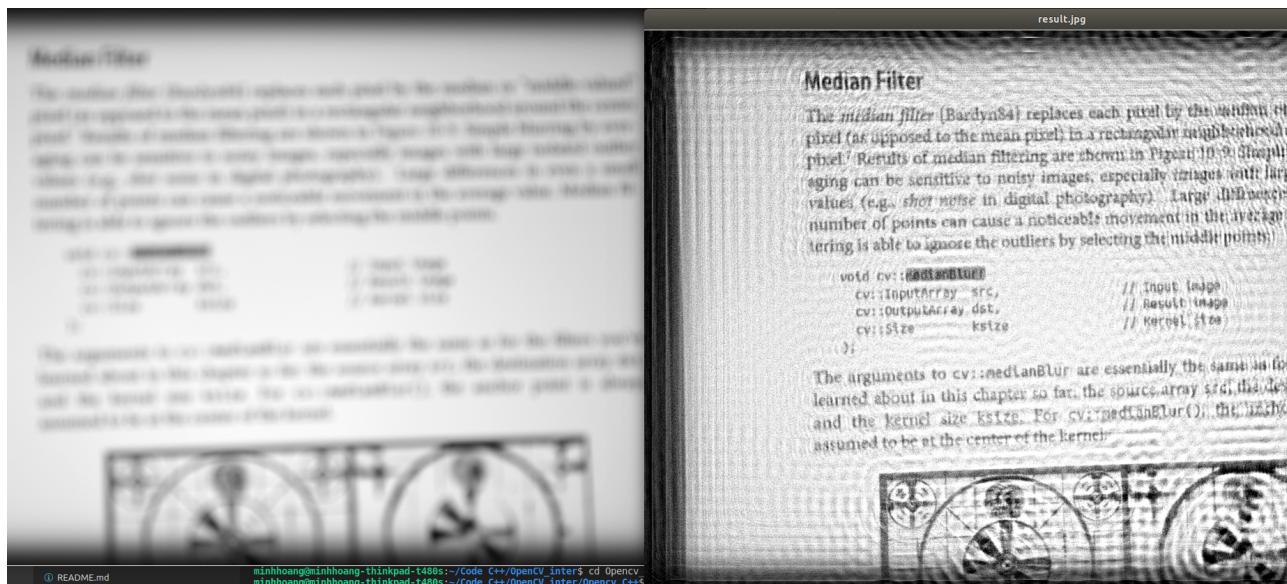


IV Image Deblurring

1: Out-of-focus Deblur Filter

- link [OoF Deblur](#)

- EX



- Mathematical model of the image degradation in frequency domain representation:

$$S = H \cdot U + N$$

- S is a spectrum of blurred image, U is a spectrum of original true image
- H is a frequency response of point spread function (PSF)
- N is a spectrum of additive noise
- How to restore a blurred image?
- The restoration formula in frequency domain is:

$$U' = H_w \cdot S$$

- U' is the spectrum of estimation of original image U
- Hw is the restoration filter, for example, the Wiener filter.

$$H_w = \frac{H}{|H|^2 + \frac{1}{SNR}}$$

-
- SNR signal-to-noise ratio