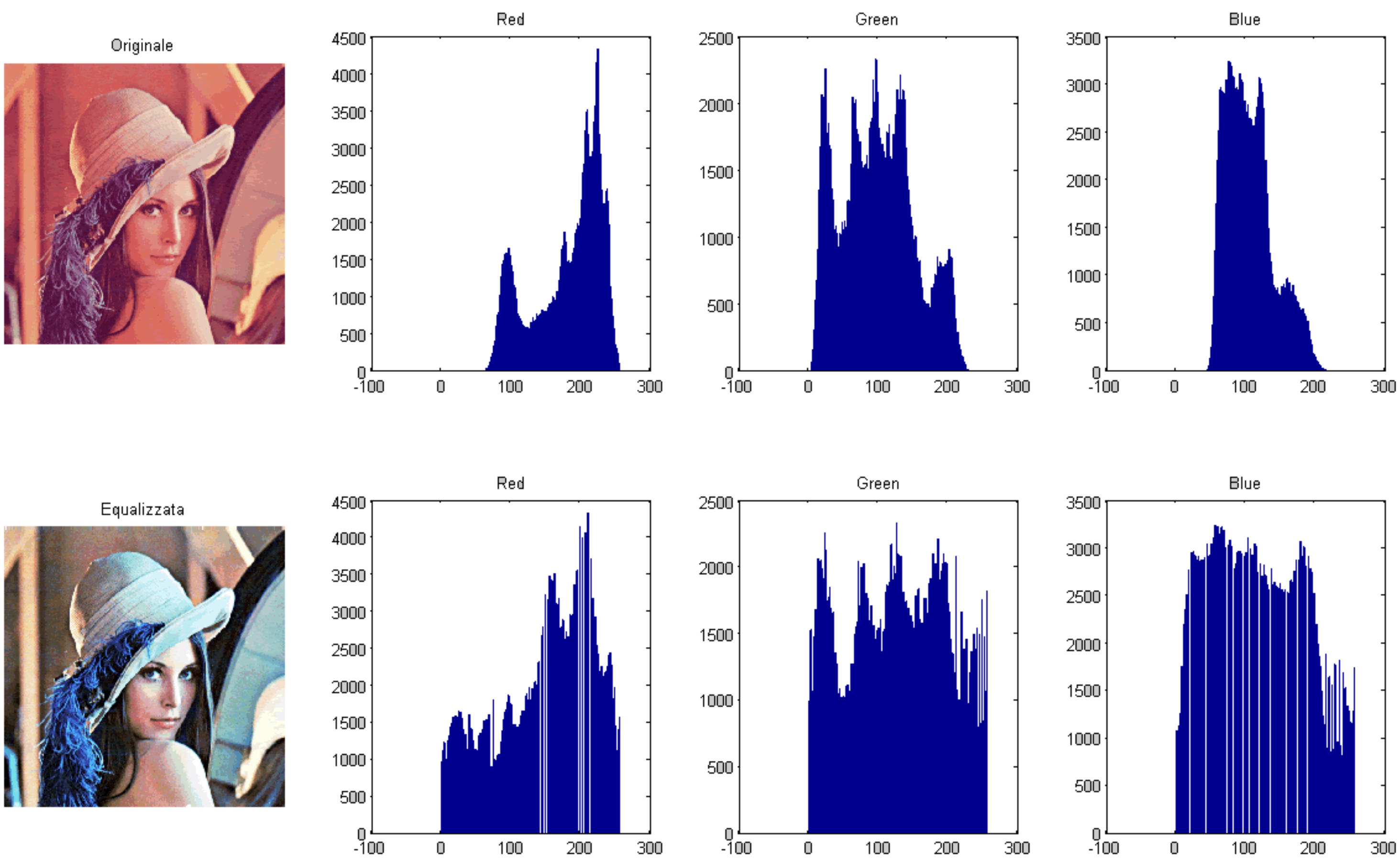
Course on Python and Data Science:
Lecture-4

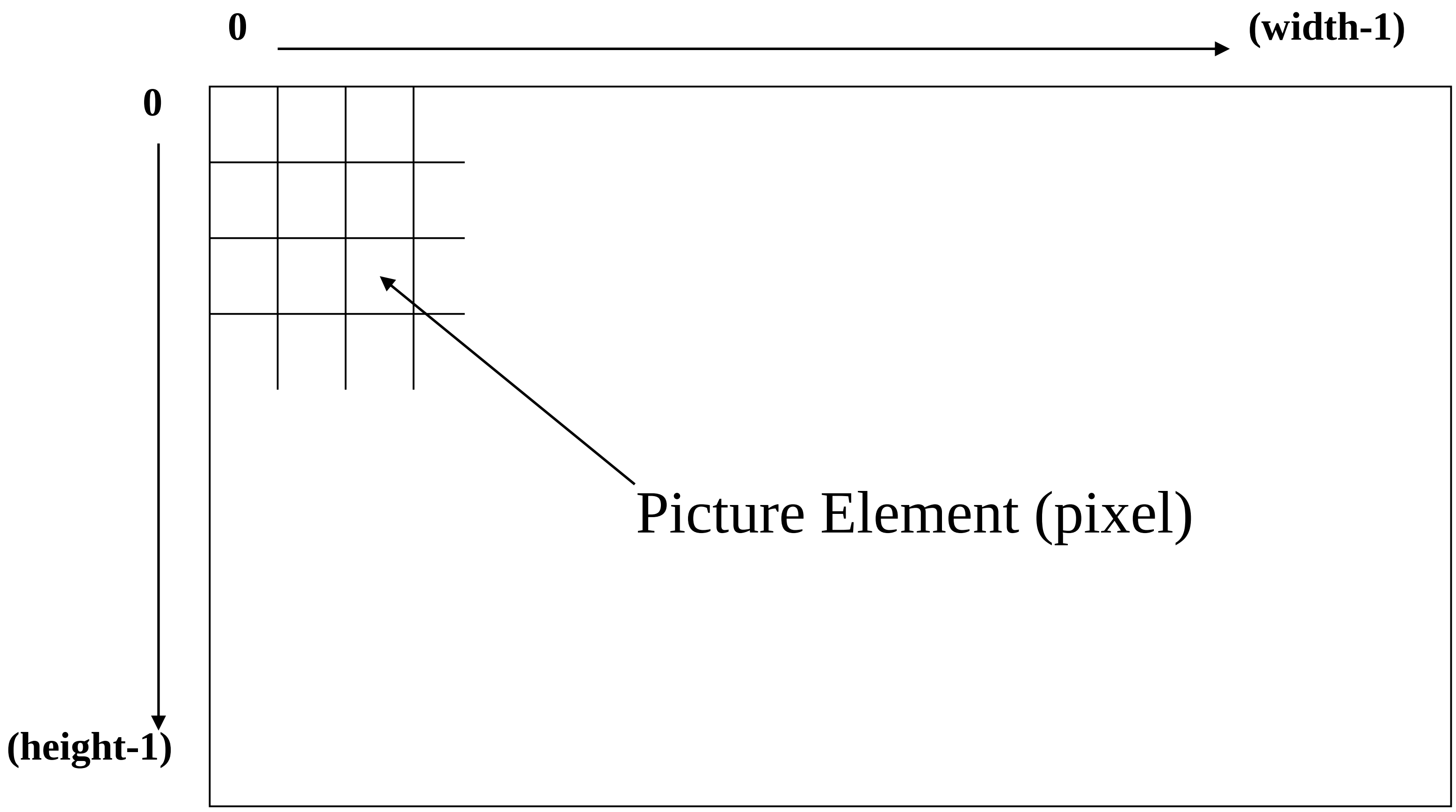
Image Processing & Computer Vision

Suhail Najeeb
Undergraduate student,
Department of Electrical and Electronic Engineering,
Bangladesh University of Engineering & Technology
Mail: suhailnajeeb19@gmail.com
25 November, 2017

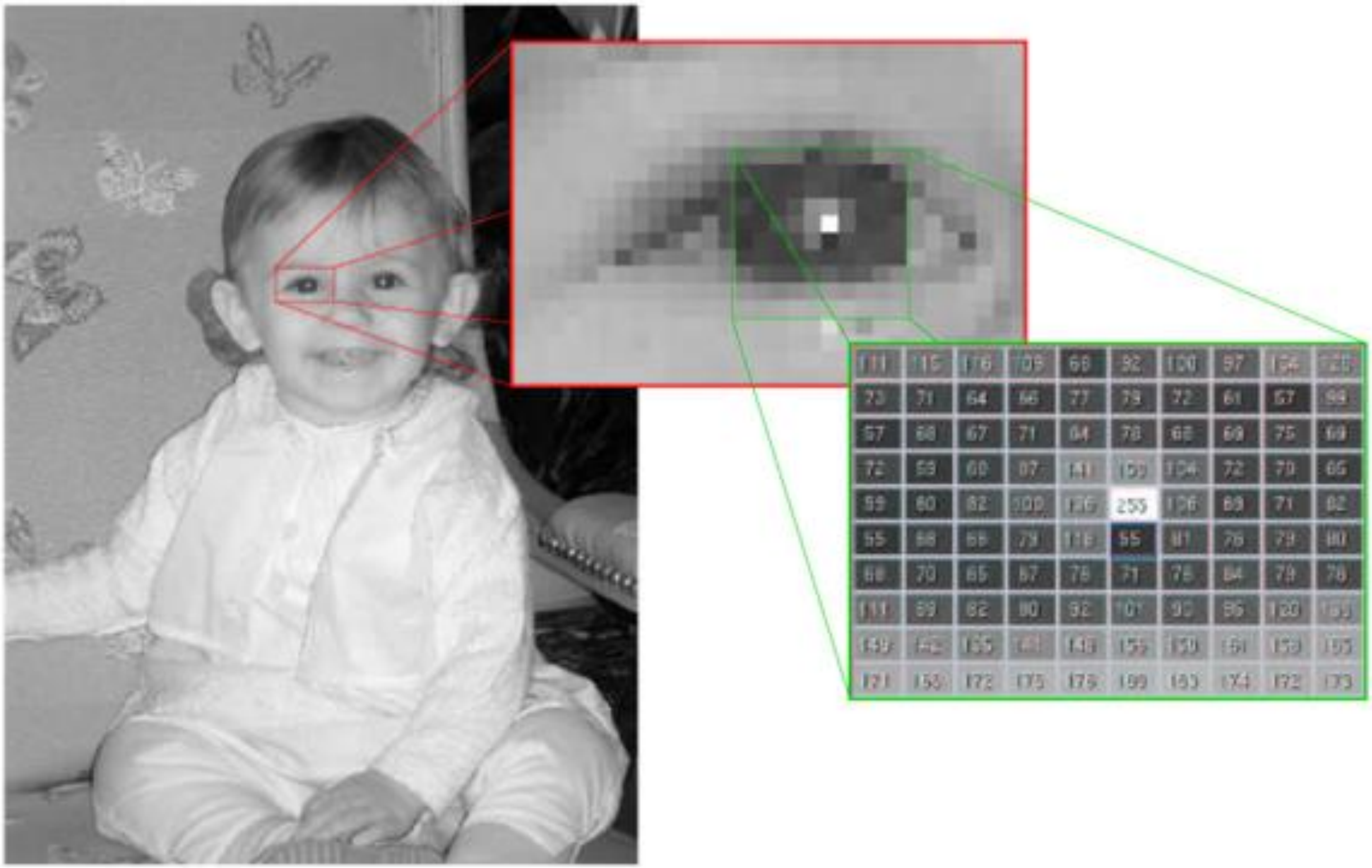
Image Processing



Representing an Image



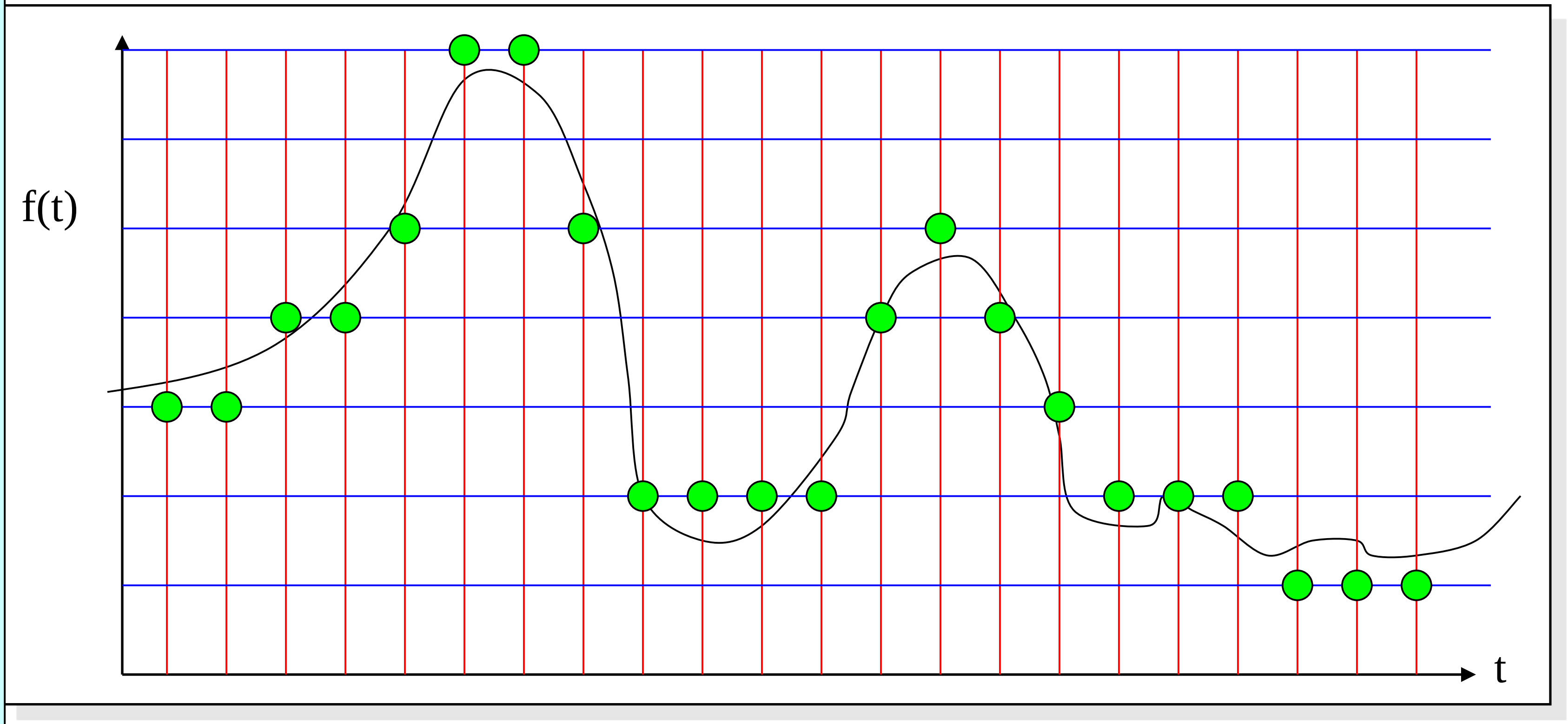
Representing an Image



Sampling & Quantization

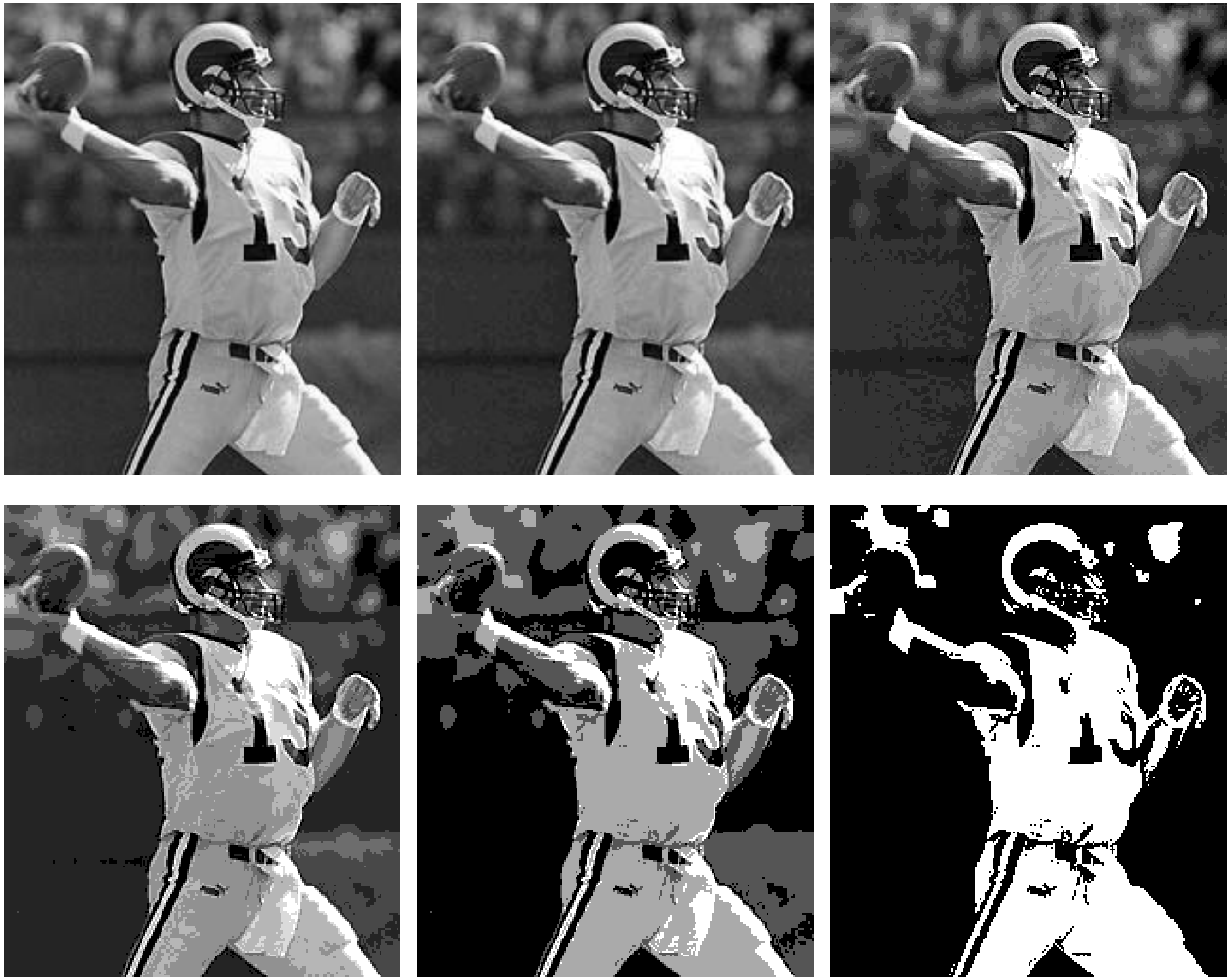
Problem: Sampling a 1D continuous function (usually of time)

Quantization (how accurately can a signal be measured)



Sampling rate (how often is a signal measured)

Effect of Quantization



A	B	C
D	E	F

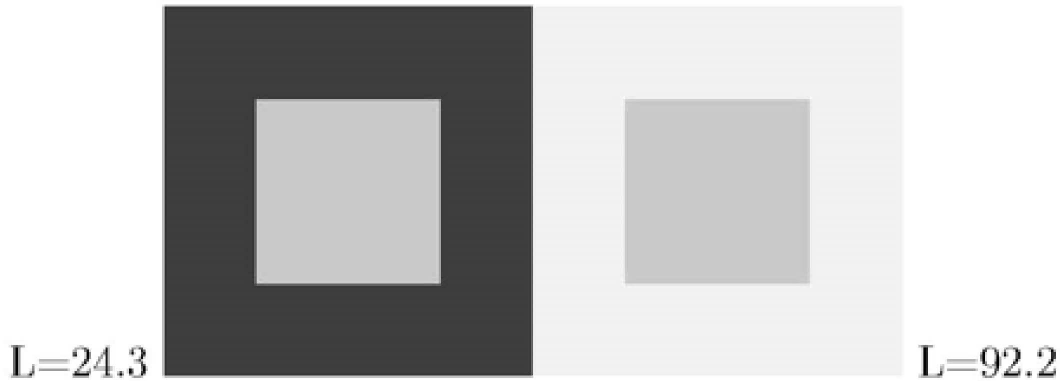
Number of Gray Levels

A: 256
B: 32
C: 16
D: 8
E: 4
F: 2

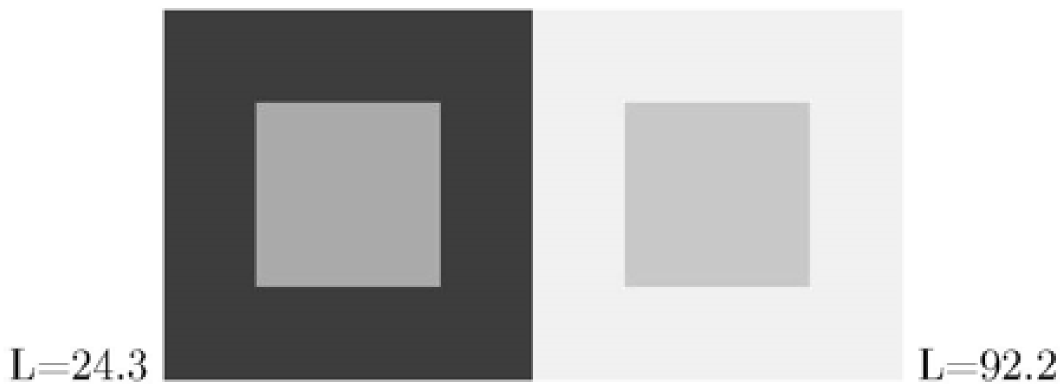
Each image is 210 x 250

Luminance and Brightness

Inside squares have same luminance ($L_L = L_R = 78.6$).



Inside squares have same brightness ($L_L = 67.0$ and $L_R = 78.6$)



Contrast and Dynamic Range

Linear Mapping

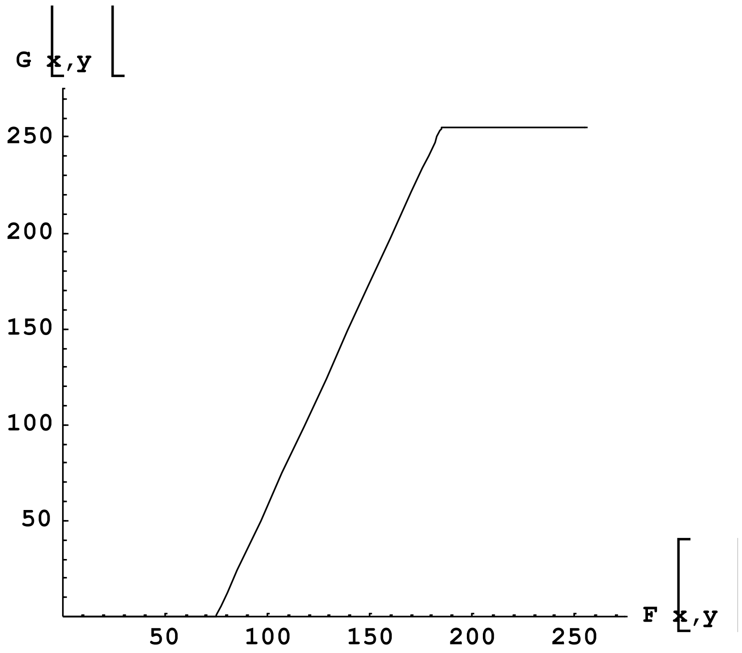
A grayscale image $f(x,y)$ can be transformed into image $g(x,y)$ using a linear gain function given as

$$g(x,y) = a * f(x,y)$$

- If $a > 1$ the image is made brighter else the image is darkened.
- a is known as "gain."

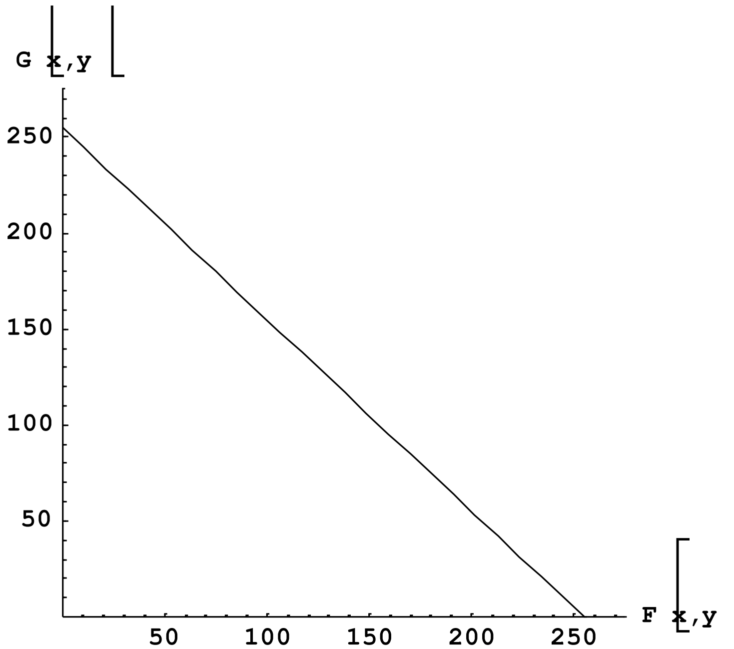
Original gain = +1.5 gain = -2.0

Linear Mapping Example



Linear (Inverse) Mapping Example

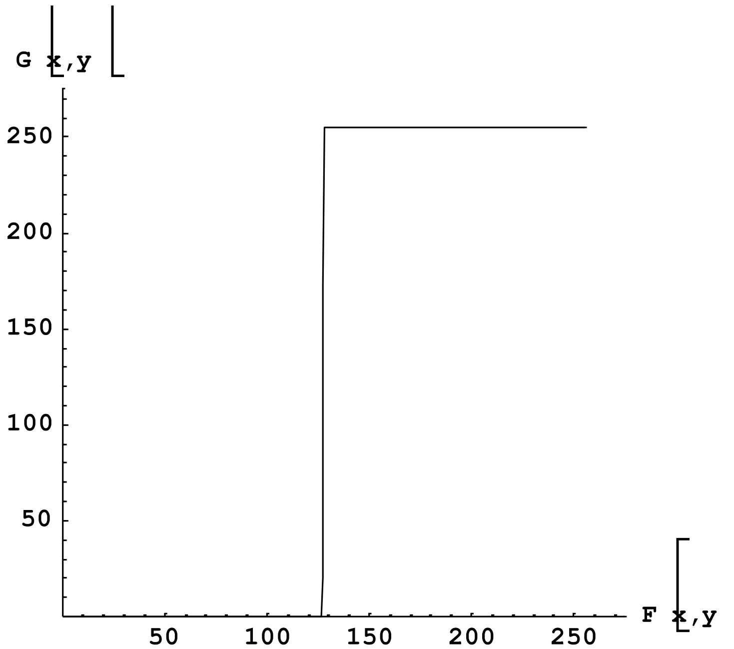
Digital Negative



Digital Negative

Linear Mapping (Thresholding)

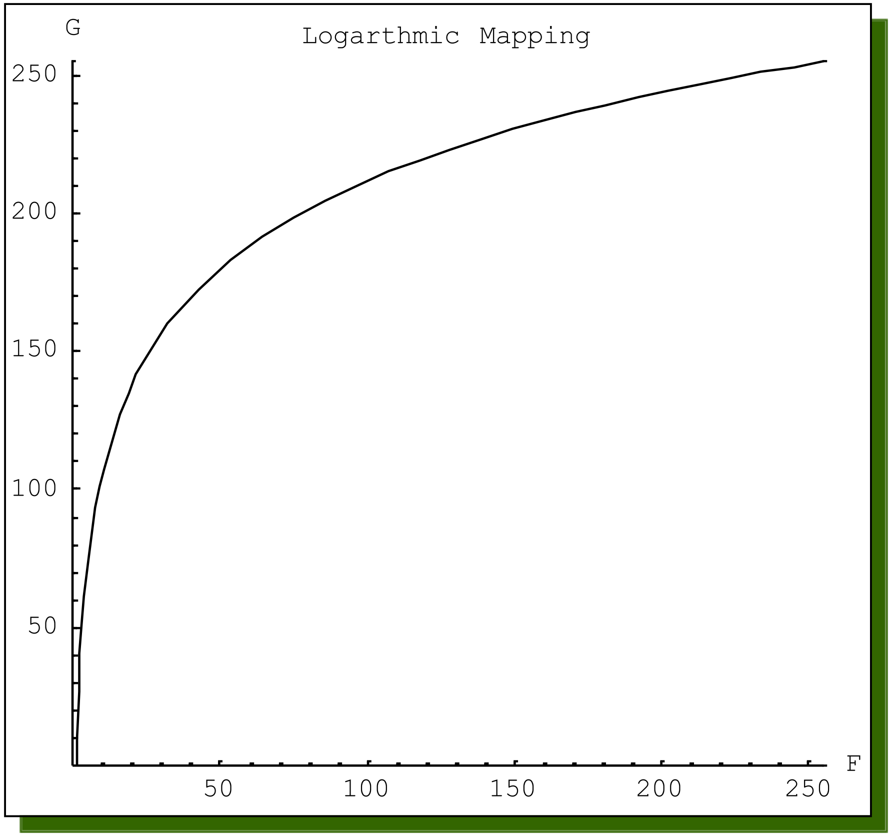
Binarization



Bilevel (binary) threshold mapping

Log (dynamic range) Compression

$$v = c\log_{10}(1+|u|)$$



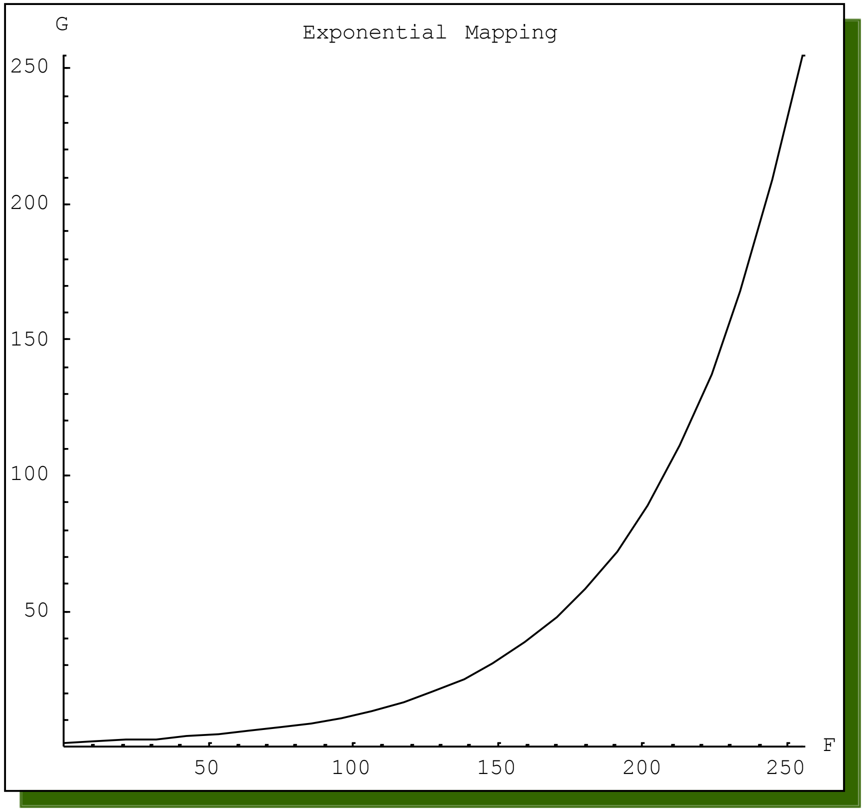
Log compression takes a range of values and "boosts" the lower end.

Notice that the higher end is "compressed" into a small range of output values.

Log Compression

Notice how the darker areas appear brighter and contain more easily viewed details. The brighter areas loose some detail but remain largely unchanged.

Exponential Mapping



Exponential Mapping takes a range of values and “boosts” the upper end.

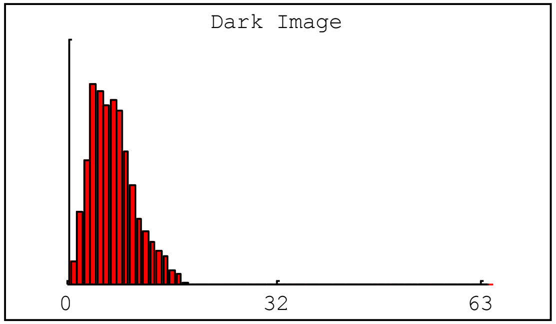
Notice that the lower range is “compressed” into a small range of output values.

Linear Contrast Stretching

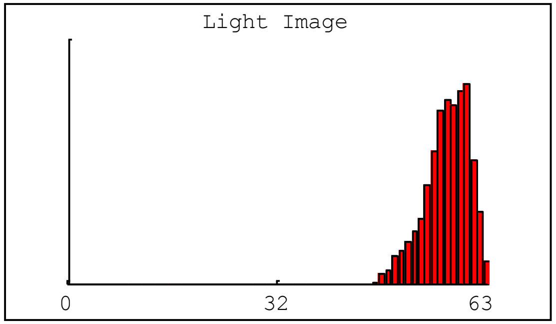
- ✓ A linear mapping that enhances the contrast of an image without removing any detail.
- ✓ Spreads the visual information available across a greater range of gray scale intensities.

Notice that the left image appears washed-out (most of the intensities are in a narrow band due to poor contrast). The right image maps those values to the full available dynamic range.

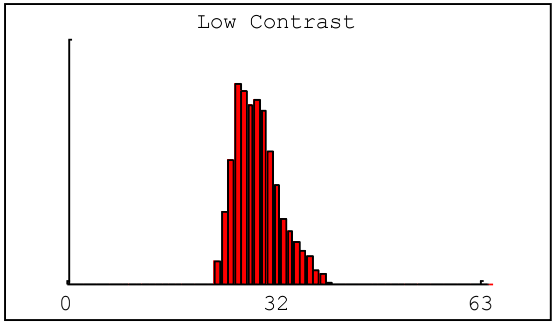
Image Histogram



The histogram on the left is representative of an "**under-exposed**" image. It has very few "bright" pixels and doesn't make good use of the full **dynamic range** available.

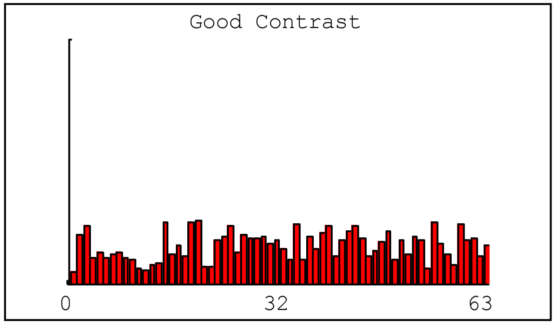


The histogram on the left is representative of an "**over-exposed**" image. It has very few "dark" pixels and doesn't make good use of the full **dynamic range** available.



The histogram on the left is representative of an "**poor contrast**" image. It has very few "dark" and very few "light" pixels. It doesn't make good use of the full **dynamic range** available.

Image Histogram



The histogram on the left is representative of an image with good contrast. It makes good use of the full **dynamic range** available.

Cumulative Distribution Function

- The CDF of an image is a table containing (for every gray level K) the probability of a pixel of level K OR LESS actually occurring in the image

- The CDF can be computed from the **histogram** as:

$$C_j = \sum_{i=0}^{j} H_i$$

Cumulative Distribution Function

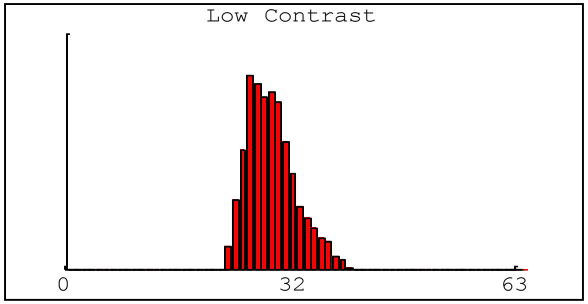


Image histogram

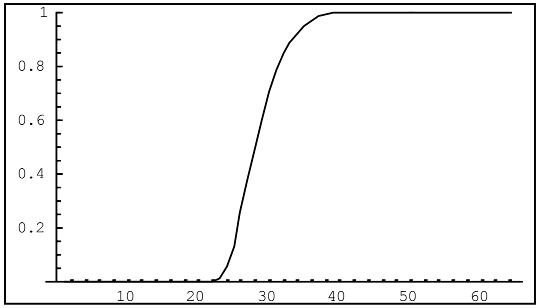


Image CDF

- ✓ The CDF of an image having uniformly distributed pixel levels is a straight-line with slope 1 (using normalized gray levels). The derivative of the CDF is constant.

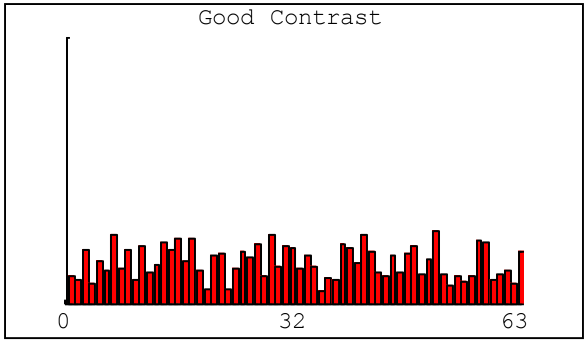


Image histogram

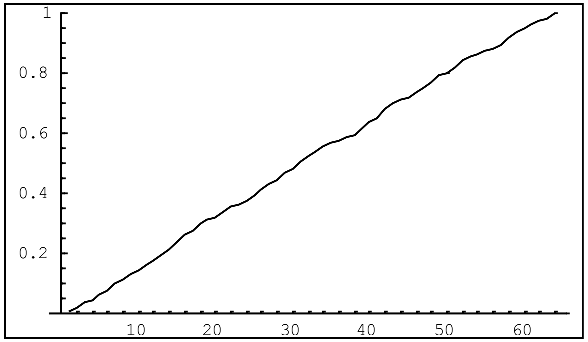


Image CDF

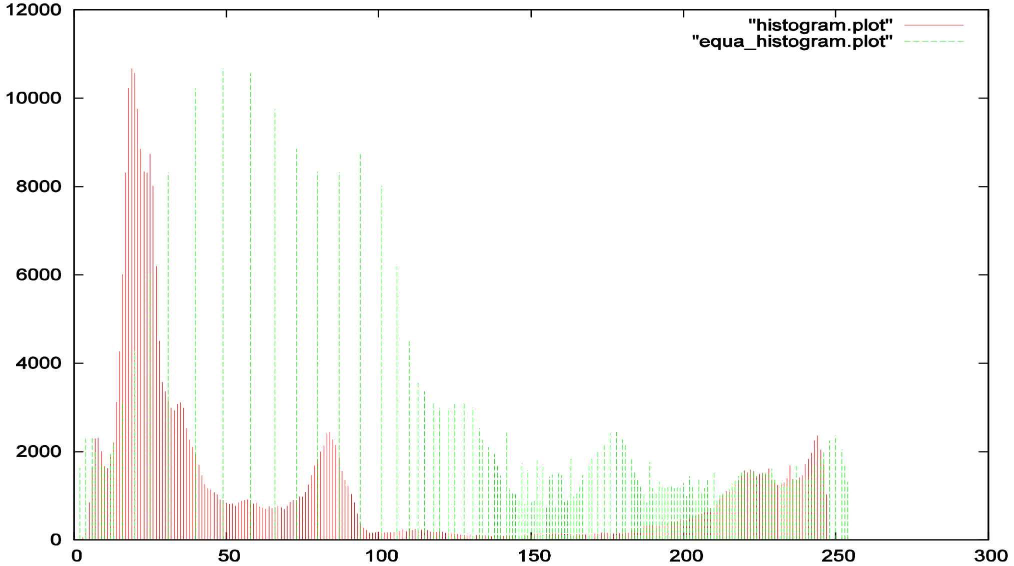
Histogram Equalization

- *Is a process of automatically distribute pixel values evenly throughout the image*
  - *Each gray level should appear with **identical** probability in the image.*
  - *Often enhances an image, but not always.*

Histogram Equalization

Original image

Equalized image



Neighborhood (Spatial) Operations

- Neighborhood operations modify pixel values based on the values of nearby pixels. Convolution and Correlation are fundamental neighborhood operations.

Convolution is used to filter images for specific reasons – to remove noise, to remove motion blur, to enhance image features, etc…

Correlation is used to determine the similarity of regions of an image to other regions of interest. Used in pattern recognition, motion analysis and image registration.

Convolution

- The value of a pixel is determined by computing a weighted sum of nearby pixels.

$$g(x,y)=\sum_{k=-1}^{1}\sum_{j=-1}^{1}h(j,k)\,f(x-j,y-k)$$

$$g=h\otimes f$$

	-1	0	+1
-1	-1	0	1
0	-2	0	2
+1	-1	0	1

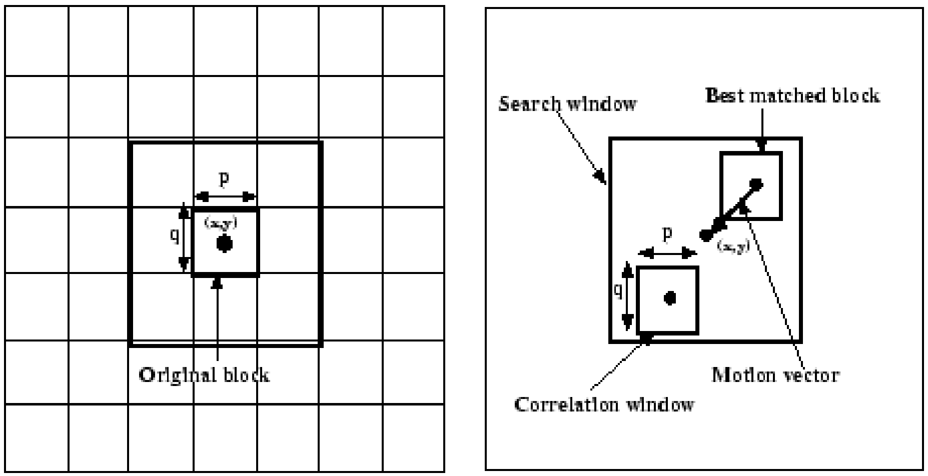
Given a "kernel" of weights to be centered on the pixel of interest

	72	53	60	
	76	56	65	
	88	78	82	

Compute the new value of the center pixel by "overlaying" the kernel and computing the weighted sum

Correlation

2D normalized cross-correlation equation:



nth frame (n-1)th frame

$$R(s,k) = \frac{\sum_{i=1}^{p}\sum_{j=1}^{q} f_n(i,j)\, f_{n-1}(i+s, j+k)}{\sqrt{\sum_{i=1}^{p}\sum_{j=1}^{q} f_n^2(i,j)}\sqrt{\sum_{i=1}^{p}\sum_{j=1}^{q} f_{n-1}^2(i+s, j+k)}}$$

Filtering

- Convolution will have different effects depending upon the values of the Kernel.

- Filtering is a way of tuning image frequencies – much like a graphic equalizer
  - Low Pass Filtering: allows only the “low-frequency signals through”
  - High Pass Filtering: allows only the “high-frequency signals through”

Low-Pass Filtering

$$g(x,y)=\sum_{k=-1}^{1}\sum_{j=-1}^{1}a(j,k)\,f(x-j,y-k)$$

- Any kernel (window) having all positive coefficients will act as a low-pass filter

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1\\\hline 1 & 1 & 1\\\hline 1 & 1 & 1\\\hline\end{array}$$

A 3 × 3 window

The center pixel becomes the average of all neighboring pixels. Also known as a **mean filter.**

Mean Filter Example

Original Image

3x3 Mean Kernel

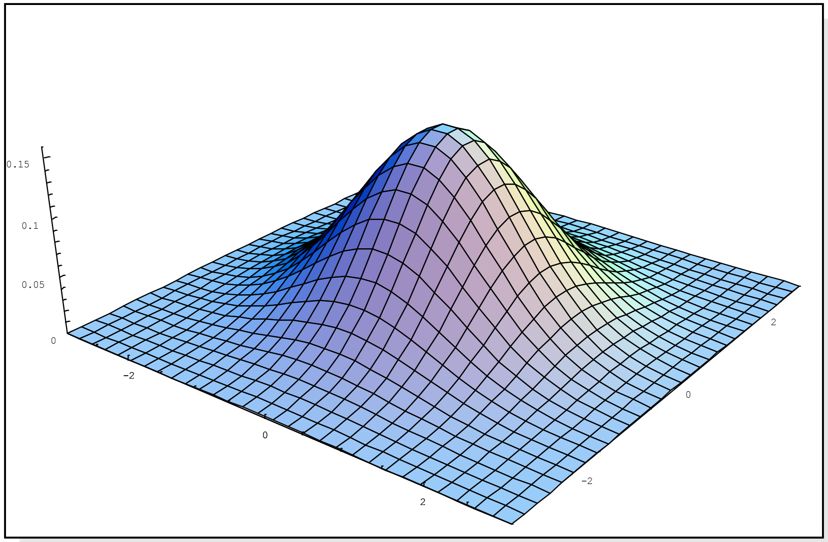
5x5 Mean Kernel

7x7 Mean Kernel

Gaussian Filter

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The gaussian kernel is separable and symmetric.
- To construct the kernel we must sample and quantize!
- The kernel below is an example where sigma = 1



1	4	7	4	1
4	16	28	16	4
7	28	49	28	7
4	16	28	16	4
1	4	7	4	1

Gaussian Filtering Example

Original Image

3x3 Gaussian Kernel

5x5 Gaussian Kernel

7x7 Gaussian Kernel

Comparison Between Gaussian and Mean Filtering

5x5 Gaussian Kernel

5x5 Mean Kernel

Noise Reduction

- Gaussian and mean filters are usually used to reduce “noise” in images

The above image has been corrupted by impulse (binary) noise.

Noise Reduction

5x5 Gaussian Kernel

5x5 Mean Kernel

Median Filtering

- **Median filter**: The value of the output pixel is the value of the "median" pixel.

$$g(x,y) = median\left(\sum_{k=-1}^{1} \sum_{j=-1}^{1} f(x-j, y-k) \right)$$

Median Filtering

Original Image (impulse noise)

Median Filtered Image (3x3)

Median Filters are great at preserving edges and eliminating impulse noise!

Sharpening Filters

- These filters highlight fine image detail or de-blur an image.

- Highpass filter: allows only high-frequency information to retain.
  - Main feature is a positive center coefficient and negative perimeter values.
  - The sum of the coefficients is zero which means that areas of constant intensity are completely eliminated.

-1	-1	-1
-1	8	-1
-1	-1	-1

A Laplacian Kernel

$$h(m,n) = \left[1 - \frac{m^2 + n^2}{\sigma^2}\right] e^{\left(-\frac{m^2+n^2}{\sigma^2}\right)}$$

HighPass (Sharpening) Filtering Example

A HighPass filtered image can be computed as the difference between the original and the low-frequency components

HighPass = Original – LowPass

- Geometric Transformation
- Image Gradients
- Canny Edge Detection
- Template Matching
- Object Detection