IMAGE ENHANCEMENT

Dr. Nursuriati Jamil

TOPICS

1. Histogram processing
2. Gray level transformations
3. Logical and arithmetic operations
4. Image smoothing
5. Image sharpening

HISTOGRAM PROCESSING

Dr. Nursuriati Jamil

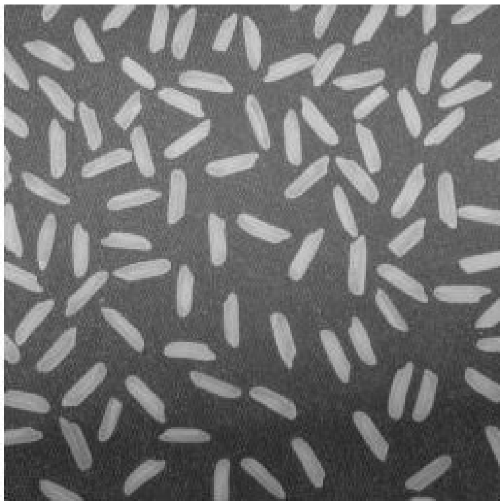
LEARNING OUTCOME

1. Able to comprehend the theory and importance of histogram in image processing
2. Able to perform histogram stretching and sliding.
3. Able to describe and calculate histogram equalization
4. Able to solve image enhancement problem by using histogram

WHY HISTOGRAM IS IMPORTANT?

In most image analysis application, useful data often populates only a small portion of the available range of gray level values. Contrast enhancement involves changing the original values so that more of the available range is used, thereby increasing the contrast between objects and their backgrounds This can be done by stretching the gray levels distribution, and equalizing the distribution of gray levels to utilize the full range of colours.

IMAGE HISTOGRAM



Grayscale image

122	92	95	99	102	107	89	90	95	122
99	99	102	82	100	89	91	87	86	99
97	107	103	86	98	92	93	96	96	97
102	100	99	87	97	89	110	95	93	102
84	107	98	99	92	94	104	91	104	84
86	107	93	107	91	109	92	105	91	86
97	104	90	93	93	96	89	121	100	97
105	102	110	97	100	93	89	106	102	105
111	97	100	95	110	98	103	105	93	111
97	88	114	93	96	87	101	94	102	97

Intensity matrix

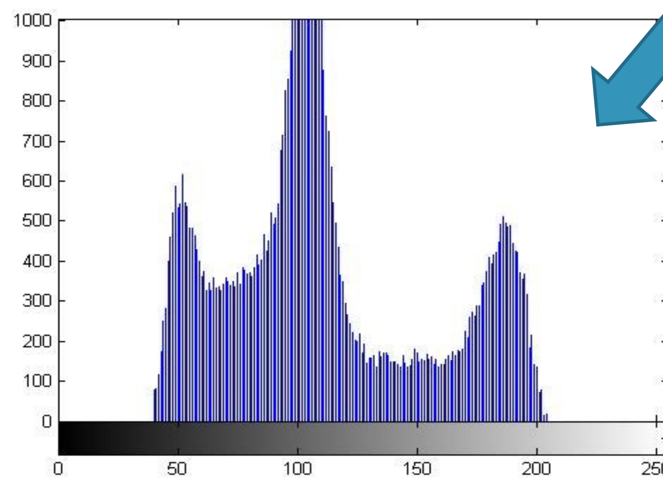
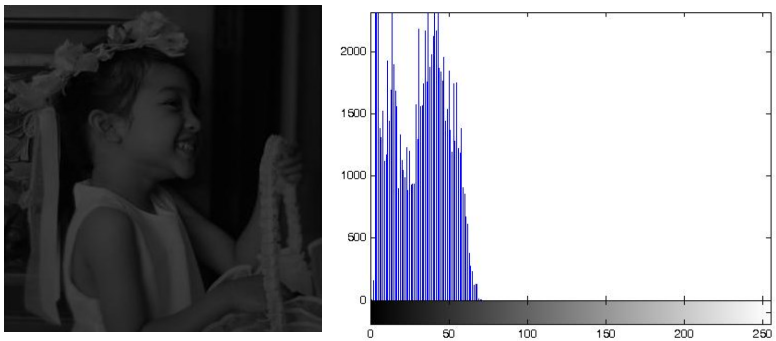
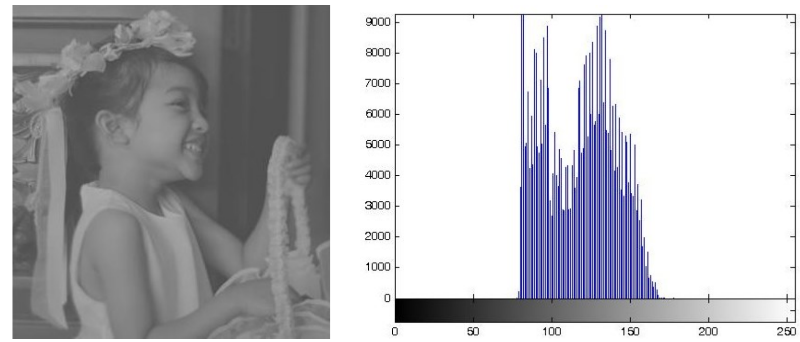


Image histogram

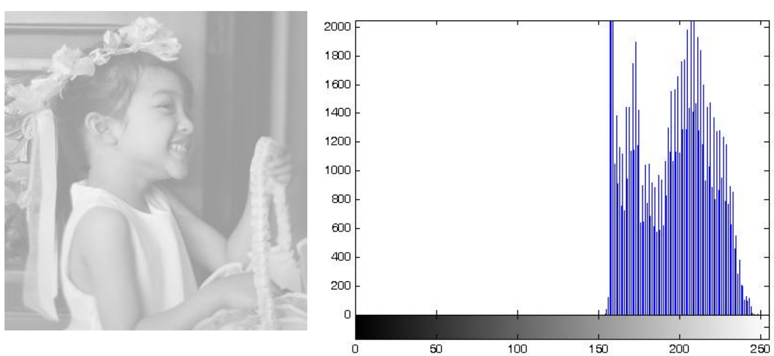
IMAGE TYPE AND HISTOGRAM



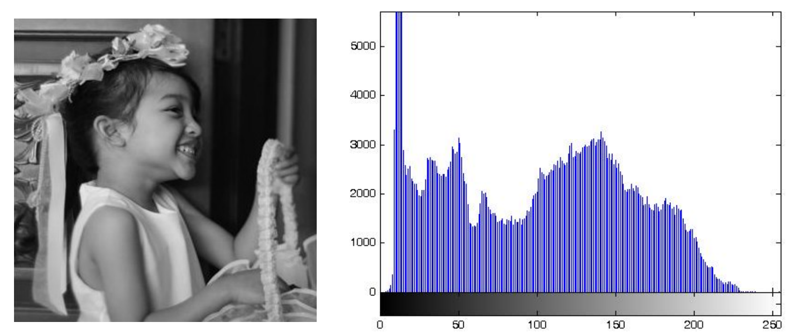
Dark image



Low-contrast image



Bright image



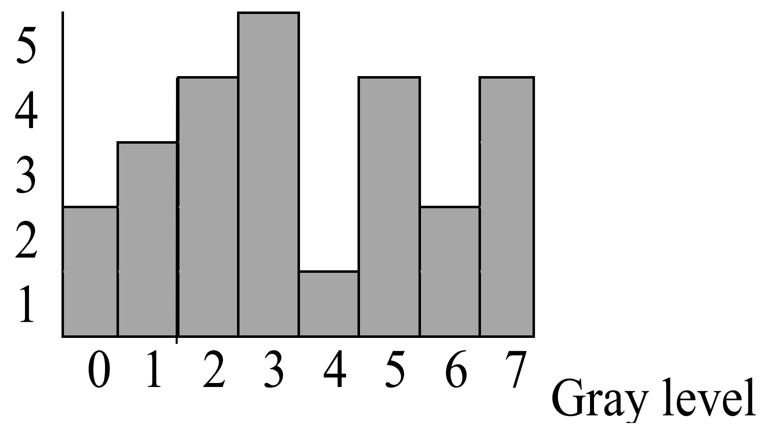
High-contrast image

HOW IS IT CONSTRUCTED?

The histogram of a digital image with gray levels from 0 to L-1 is a discrete function $h(r_k)=n_k$, where

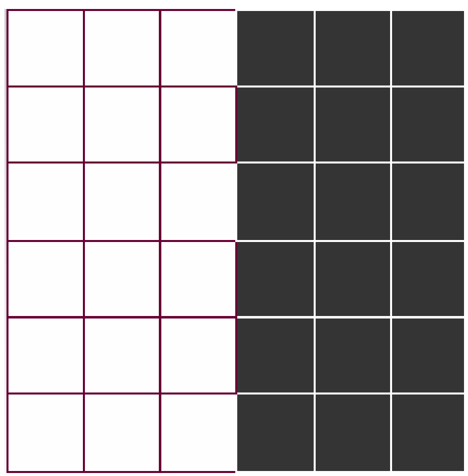
- r_k is the kth gray level
- n_k is the # pixels in the image with that gray level
- n is the total number of pixels in the image
- k = 0, 1, 2, …, L-1

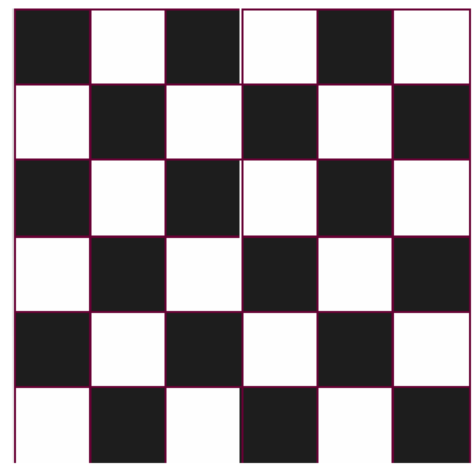
0	0	3	3	7
1	3	3	5	5
7	7	6	5	4
3	1	2	2	2
2	5	6	7	1



ANSWER THIS!

Do these images have the same histogram?





Yes, histogram contains only information about gray level distribution. It doesn't contain information about the spatial distribution (i.e. location) of the gray level

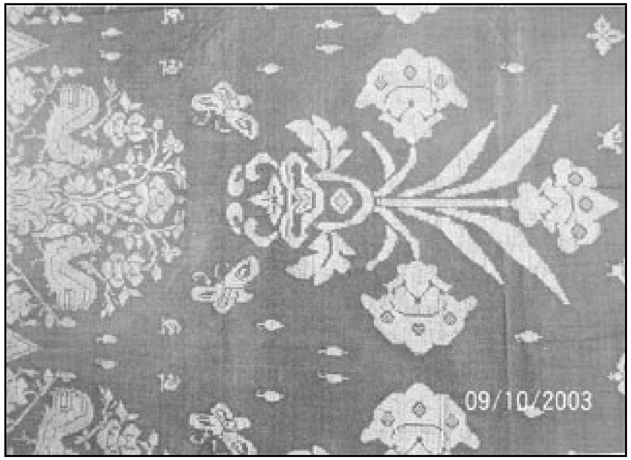
HOW CAN A HISTOGRAM ENHANCE AN IMAGE?

- **Histogram stretching** - identifying minimum and maximum brightness values from the histogram and applying a transformation to uniformly stretch this range to fill the full range.
- **Histogram sliding** - If an image needs to be brightened or darkened without changing the relationship between the gray level values, histogram sliding method can be used.
- **Histogram equalization** - creates an image with equally distributed brightness levels over the whole brightness scale in the histogram

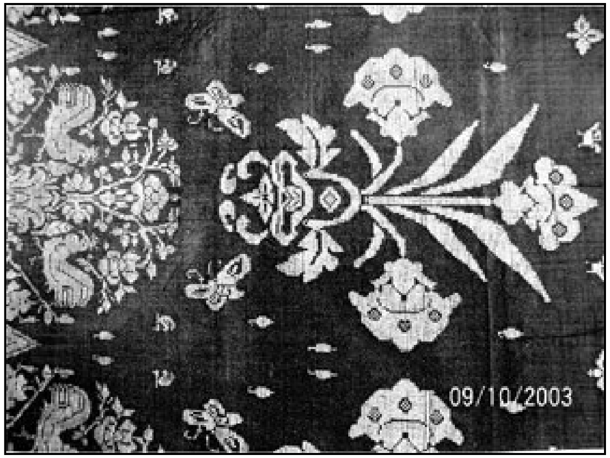
HISTOGRAM STRETCHING

One way to increase the contrast of an image the pixel values could be stretched using this equation:

$$J = 255 \cdot \frac{I - I_{\min}}{I_{\max} - I_{\min}}$$

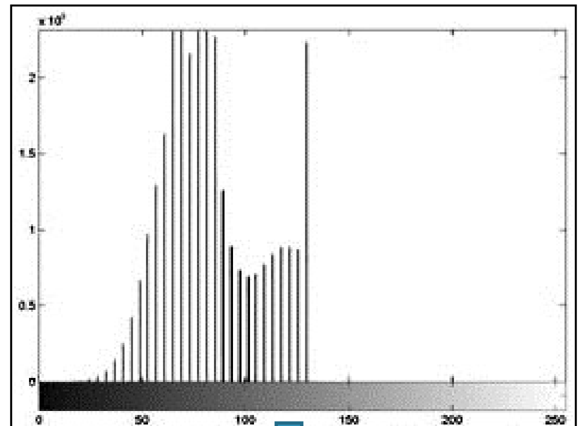


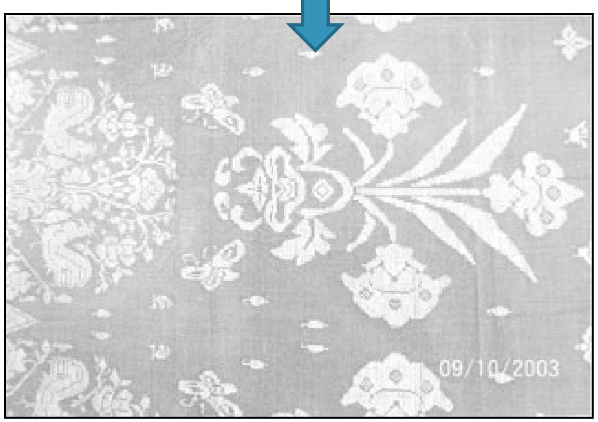
Original image

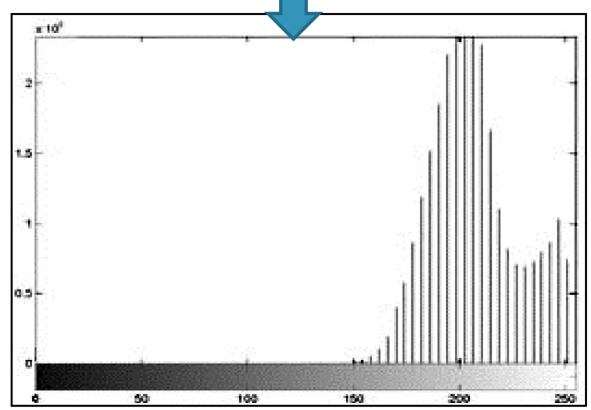


Stretched image

HISTOGRAM SLIDING







HISTOGRAM EQUALIZATION

Four steps:

1. Build the image histogram.
2. Normalize the histogram to its size.
   - Normalized histogram: , where n_k is the number of pixels in the image of size M x N with intensity r_k.
3. Accumulate the histogram using cumulative function T(r_k), where:

$$T(r_k) = \sum_{r=0}^{K} p(r_k)$$

4. Assign the new equalized pixels to the original image.

HISTOGRAM EQUALIZATION –STEP 1

Build the image histogram.

0	6	2	7	2	3	7	1
2	2	3	1	1	5	5	1
7	7	6	3	3	6	6	3
0	0	4	4	4	1	3	1
4	4	5	5	5	2	2	2
0	0	7	7	7	5	5	5
1	1	1	3	3	3	2	2
7	7	6	3	3	6	6	3

Image

Graylevel	0	1	2	3	4	5	6	7
Histogram	5	9	9	12	5	8	7	9

Histogram

HISTOGRAM EQUALIZATION –STEP 2

Normalize the histogram.

Graylevel	0	1	2	3	4	5	6	7
Histogram	5	9	9	12	5	8	7	9

Histogram

Graylevel	0	1	2	3	4	5	6	7
Histogram	5	9	9	12	5	8	7	9
Normalized	0.08	0.14	0.14	0.19	0.08	0.13	0.11	0.14

Normalized histogram

HISTOGRAM EQUALIZATION –STEP 3

Built cumulative histogram.

Graylevel	0	1	2	3	4	5	6	7
Histogram	5	9	9	12	5	8	7	9
Normalized	0.08	0.14	0.14	0.19	0.08	0.13	0.11	0.14
Cumulative	**0.08**	**0.22**	**0.36**	**0.55**	**0.63**	**0.76**	**0.87**	**1.0**

Cumulative histogram

HISTOGRAM EQUALIZATION –STEP 4

Calculate new gray levels

Graylevel	0	1	2	3	4	5	6	7
Histogram	5	9	9	12	5	8	7	9
Normalized	0.08	0.14	0.14	0.19	0.08	0.13	0.11	0.14
Cumulative	0.08	0.22	0.36	0.55	0.63	0.76	0.87	1.0
New Gray level T(r_k) * 255	20	56	92	140	161	194	222	255

Equalized gray levels

HISTOGRAM EQUALIZATION –RESULT

0	6	2	7	2	3	7	1
2	2	3	1	1	5	5	1
7	7	6	3	3	6	6	3
0	0	4	4	4	1	3	1
4	4	5	5	5	2	2	2
0	0	7	7	7	5	5	5
1	1	1	3	3	3	2	2
7	7	6	3	3	6	6	3

Image

20	222	92	255	92	140	255	56
92	92	140	56	56	194	194	56
255	255	194	140	140	222	222	140
20	20	161	161	161	56	140	56
161	161	194	194	194	92	92	92
20	20	255	255	255	194	194	194
56	56	56	140	140	140	92	92
255	255	222	140	140	222	222	140

Equalized image

GRAY LEVEL TRANSFORMATIONS

LEARNING OUTCOME

1. Able to describe the process of each gray-level transformation function: negative, log, thresholding, contrast stretching, gray-level slicing and bit-plane slicing.
2. Able to solve image enhancement problem by using the correct gray-level transformation function.

WHY APPLY A TRANSFORMATION FUNCTION ON AN IMAGE?

It is the simplest method of enhancing an image. Other than that, you can also manipulate an image and select certain region of interest in the image. This method applies a transformation function, *T*, on each pixels in the image.

POINTWISE TRANSFORMATION

Gray level transformation functions are pixel-based or pointwise transformation functions.

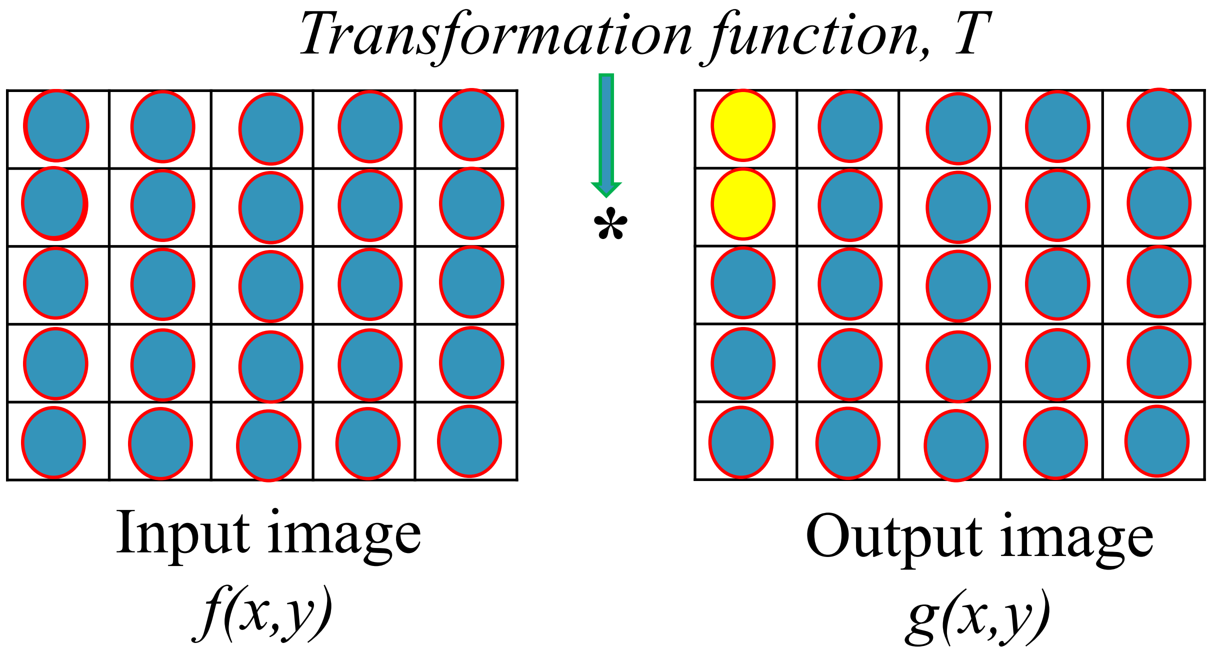


IMAGE NEGATIVES

Obtained by using : $g(x, y) = T(\ f(x,y)\ )$,

where $T = L - 1 - f(x,y)$ and L = maximum gray level value.

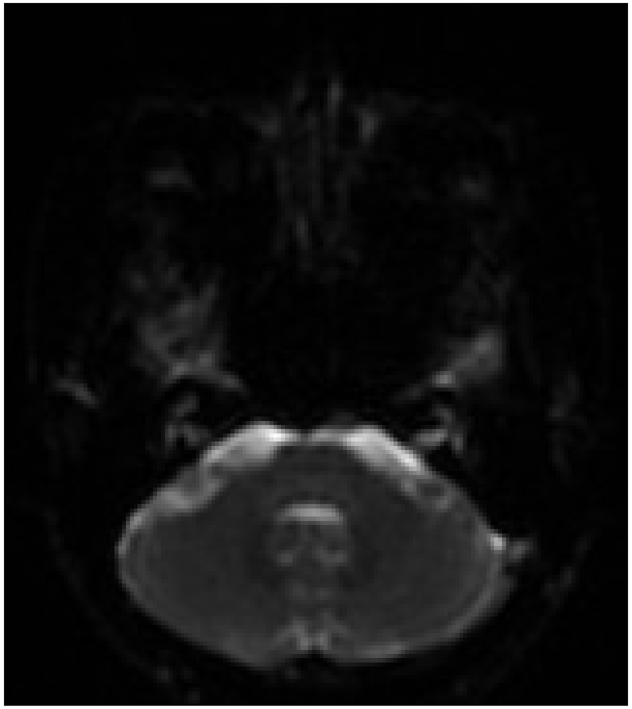
If L = 256, what is the negative value of pixel intensity 141?

$$256 - 1 - 141 = 114$$

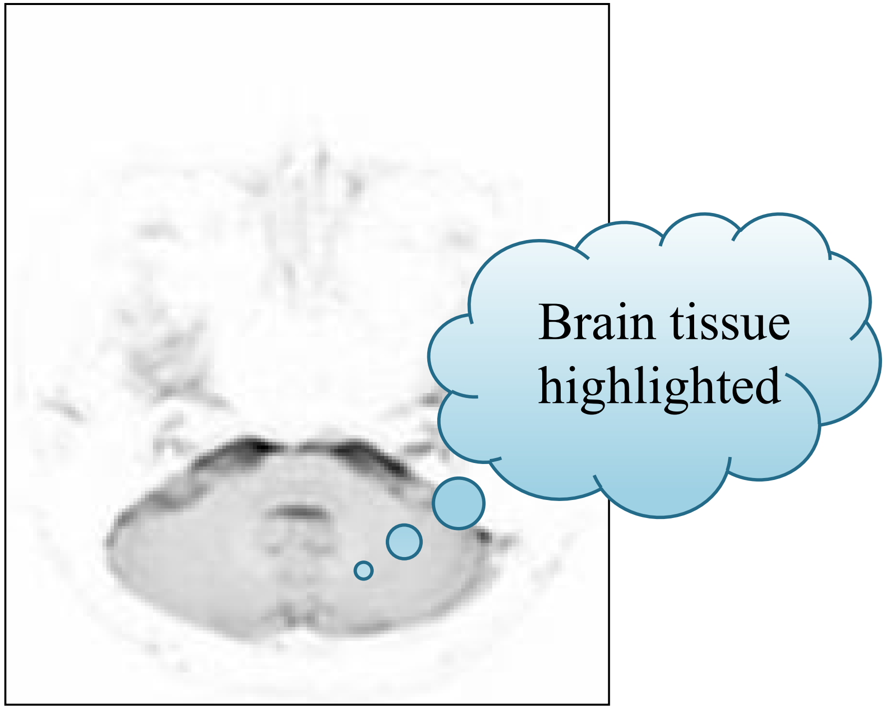
141	149
123	127

114	149
123	127

IMAGE NEGATIVE DEMO



Before



After

LOG TRANSFORMATIONS

$s = c \log(1 + f(x,y))$, c: constant

Compresses the dynamic range of images with large variations in pixel values so that the image can be display.

-1555.3	273.6
158.11	-392.7
1631.3	-1160.5
1348.3	-706.7

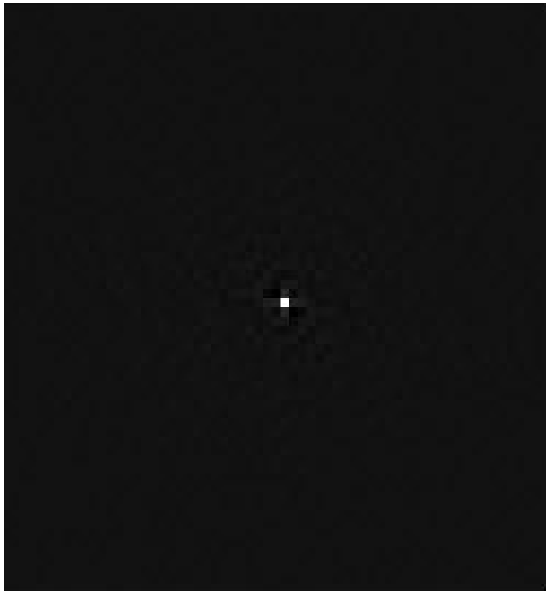
→

8.34	6.61
6.06	6.97
8.39	8.05
8.20	7.56

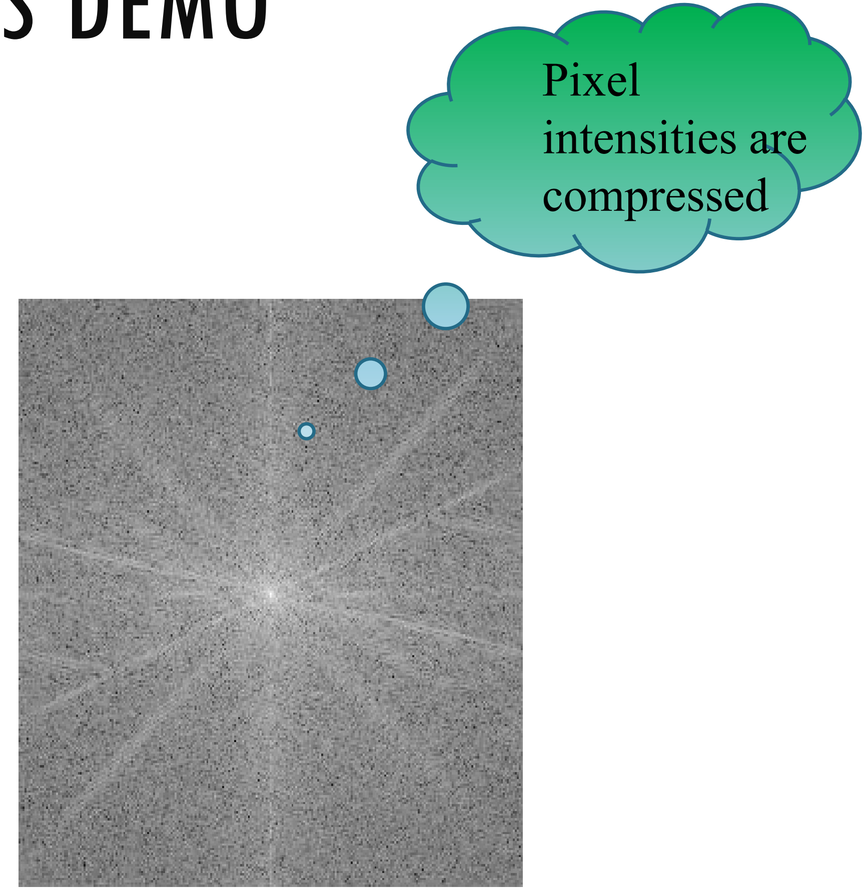
$$1 + \log(1348.3) = 8.20$$

Pixel intensities are compressed

LOG TRANSFORMATIONS DEMO



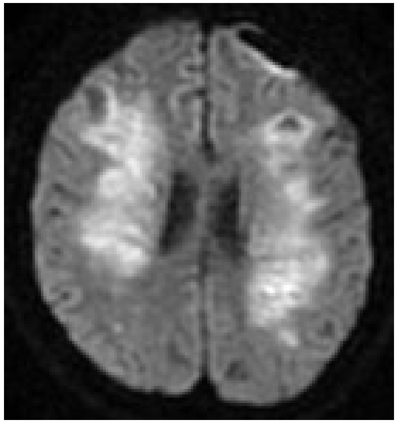
Before



After

GRAY LEVEL SLICING

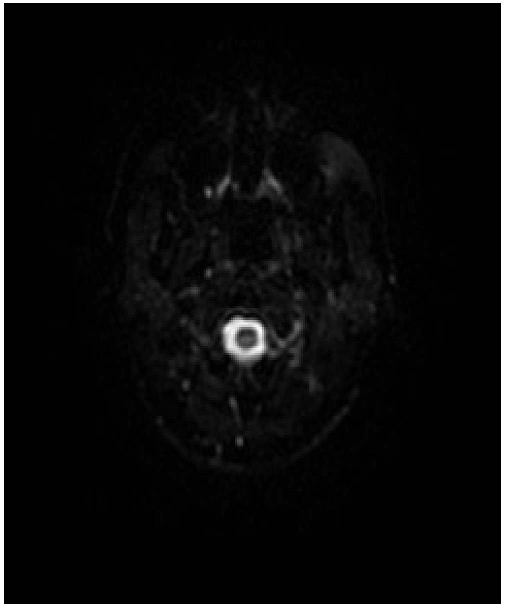
To highlight a specific range of gray levels in an image (e.g. to enhance certain features).



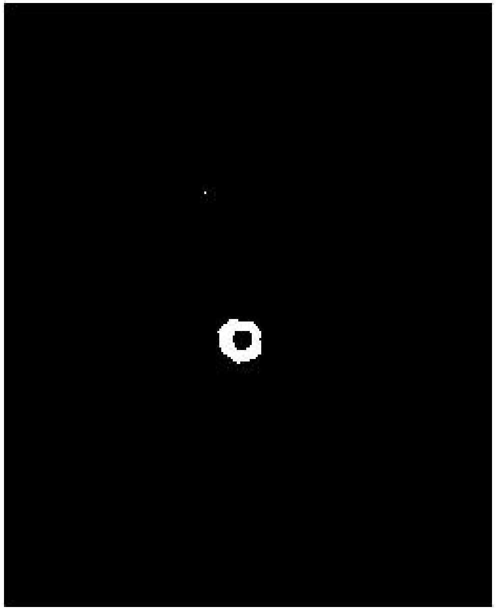
THRESHOLDING

More about this in image segmentation chapter

Thresholding is a process of converting an image to black and white.



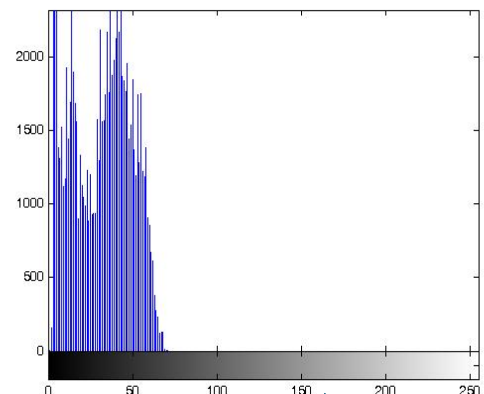
Grayscale image

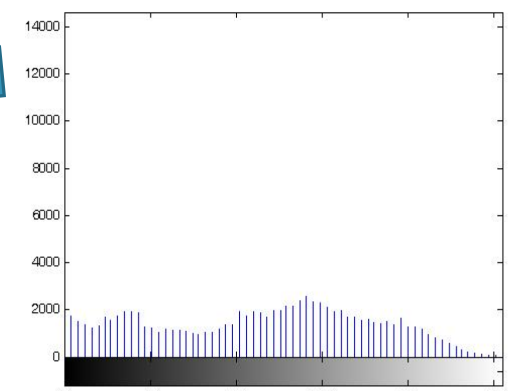


Binary image

CONTRAST STRETCHING

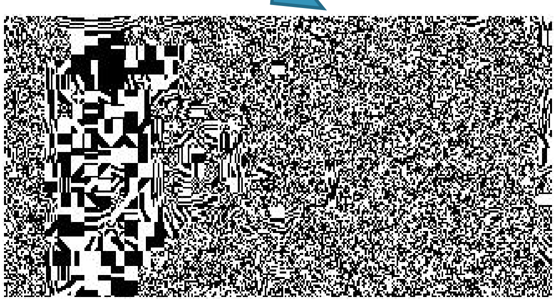
Works by expanding range of intensities, *(r1, r2)* in the input image to the specified range, *(s1, s2)*.



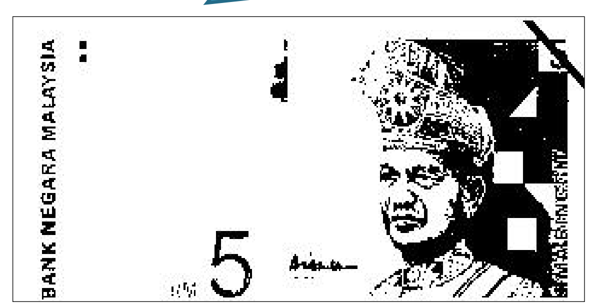


BIT-PLANE SLICING

Selects certain bits plane of an image. Higher order bit planes contain visually significant data that describes the general shape of the image. Meanwhile, the lower order bit planes contributes the details of the image.



Bit plane 0



Bit plane 7

LOGICAL AND ARITHMETIC FUNCTIONS

LEARNING OUTCOME

1. Able to describe the mathematical concept of each image logical and arithmetic operations: NOT, AND , OR, addition, multiplication, subtraction, division.
2. Able to solve image enhancement problem and manipulate images by using logical and arithmetic operations.

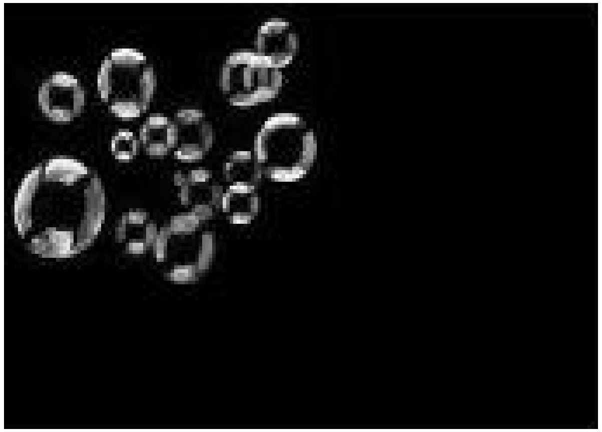
HOW CAN ARITHMETIC ENHANCE AN IMAGE?

An image may also be enhanced using the logical operations NOT, AND and OR which operate on strings of binary numbers. Arithmetic operations such as addition (+), subtraction (-), multiplication (*) and division (/) are also commonly used for enhancement and region selection purposes.

LOGICAL OPERATIONS

NOT is used to inverse and image, AND and OR operations are commonly used for region of interest (ROI) processing in an image.

IMAGE ADDITION



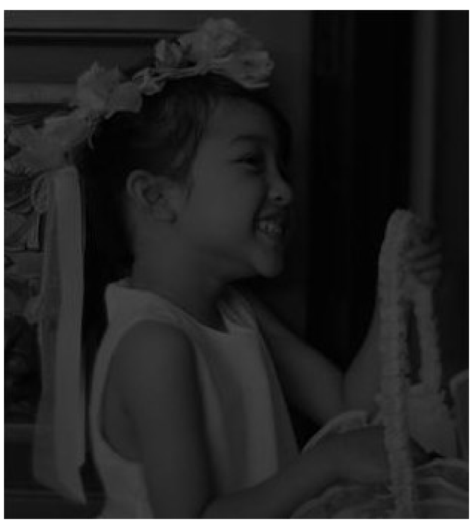
+

=

Both input images must be the same size and type.

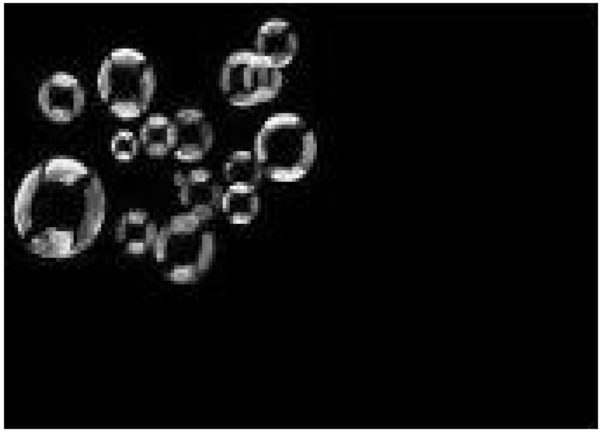
IMAGE MULTIPLICATION

By multiplying an image, you can change the brightness of an image.



X 4 =

IMAGE SUBTRACTION

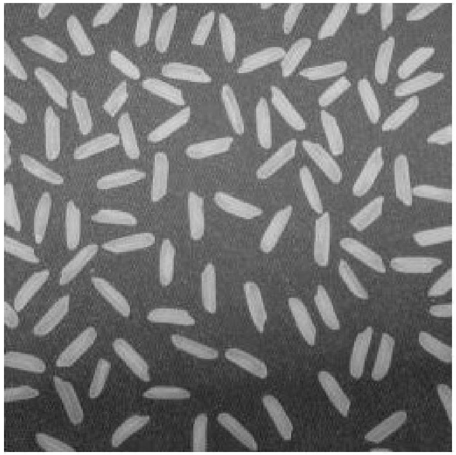


+

=

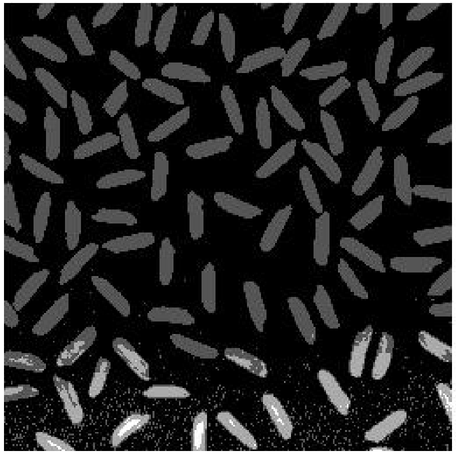
Both input images must be the same size and type.

IMAGE DIVISION



÷

=



Both input images must be the same size and type.

IMAGE FILTERING

LEARNING OUTCOME

1. Able to calculate linear and order-statistic filtering.
2. Able to solve noise removal problem by choosing the correct filter types and sizes.
3. Able to calculate sharpening filtering
4. Able to enhance image by choosing the correct filter types and sizes.

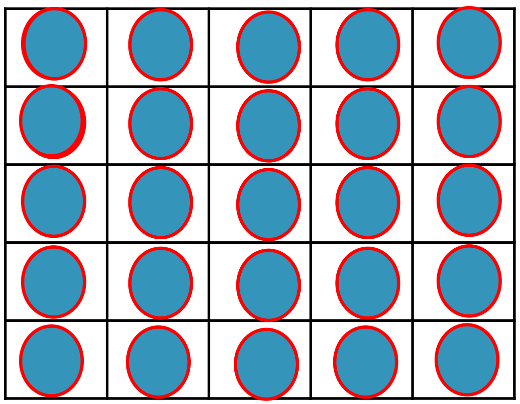
WHY DO WE FILTER IMAGE?

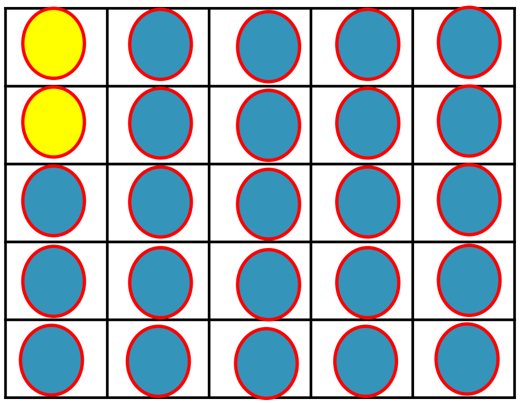
Image filtering allows you to apply various effects to an image such as removing noise, and sharpening images. Common ways of filtering are using linear and order-statistic filtering.

Linear filters : Average, Laplacian, Gaussian

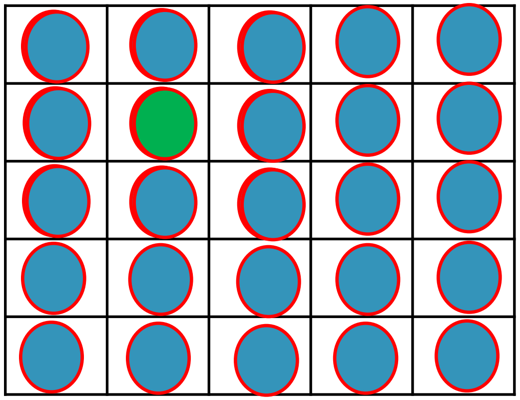
Order-statistic filters: Min, max, median

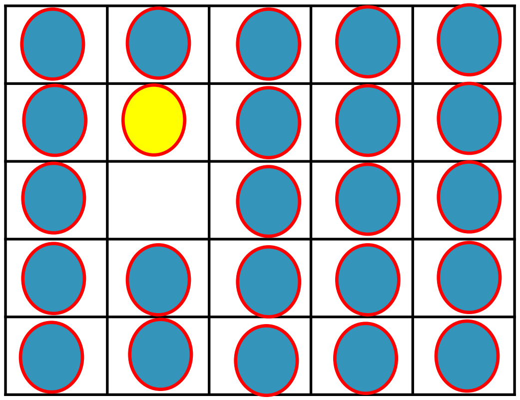
HOW DO LINEAR FILTER WORKS?





Pointwise transformation





Linear filter works by using neighbourhood transformation

HOW DO LINEAR FILTER WORKS?

Given a mask, *m*

m_1	m_2	m_3
m_4	m_5	m_6
m_7	m_8	m_9

and an image, *f(x, y)*

f(1,1)	…	…	…	…	…	***f(1,n)***
:		***f(x-1, y-1)***	***f(x-1, y)***	***f(x-1, y+1)***		**:**
:		***f(x, y-1)***	***f(x, y)***	***f(x, y+1)***		**:**
:		***f(x+1, y-1)***	***f(x+1, y)***	***f(x+1, y+1)***		**:**
f(m,0)	…	…	…	…	…	***f(m,n)***

HOW DO LINEAR FILTER WORKS?

Step 1. Position the mask, *m* centred at the current pixel to be filtered, *f(x,y)*.

m_1	m_2	m_3
m_4	m_5	m_6
m_7	m_8	m_9

f(1,1)	…	…	…	…	…	*f(1,n)*
:		*f(x-1, y-1)*	*f(x-1, y)*	*f(x-1, y+1)*		:
:		*f(x, y-1)*	*f(x, y)*	*f(x, y+1)*		:
:		*f(x+1, y-1)*	*f(x+1, y)*	*f(x+1, y+1)*		:
f(m,0)	…	…	…	…	…	*f(m,n)*

HOW DO LINEAR FILTER WORKS?

Step 2. Form all products of the mask coefficients with the corresponding neighbourhood pixels.

$m_1 * f(x-1, y-1)$	$m_2 * f(x-1, y)$	$m_3 * f(x-1, y+1)$
$m_4 * f(x, y-1)$	$m_5 * f(x, y)$	$m_6 * f(x, y+1)$
$m_7 * f(x+1, y-1)$	$m_8 * f(x+1, y)$	$m_9 * f(x+1, y+1)$

HOW DO LINEAR FILTER WORKS?

Step 3. Add all the products

$$f(x, y) = m_1.f(x-1,y-1) + m_2.f(x-1,y) + m_3.f(x-1,y+1) + m_4.f(x,y-1) + m_5.f(x,y) + m_6.f(x,y+1) + m_7.f(x+1,y-1) + m_8.f(x+1,y+1)$$

$m_1 * f(x-1, y-1)$	$m_2 * f(x-1, y)$	$m_3 * f(x-1, y+1)$
$m_4 * f(x, y-1)$	$m_5 * f(x, y)$	$m_6 * f(x, y+1)$
$m_7 * f(x+1, y-1)$	$m_8 * f(x+1, y)$	$m_9 * f(x+1, y+1)$

AVERAGE FILTERS

Used for blurring (removal of small details prior to large object extraction, bridging small gaps in lines) and noise reduction.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Box filter

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Weighted filter

AVERAGE FILTERING

How to filter this pixel?

Step 1. Move the average filter to the centre pixel

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3 x 3 box filter

Image

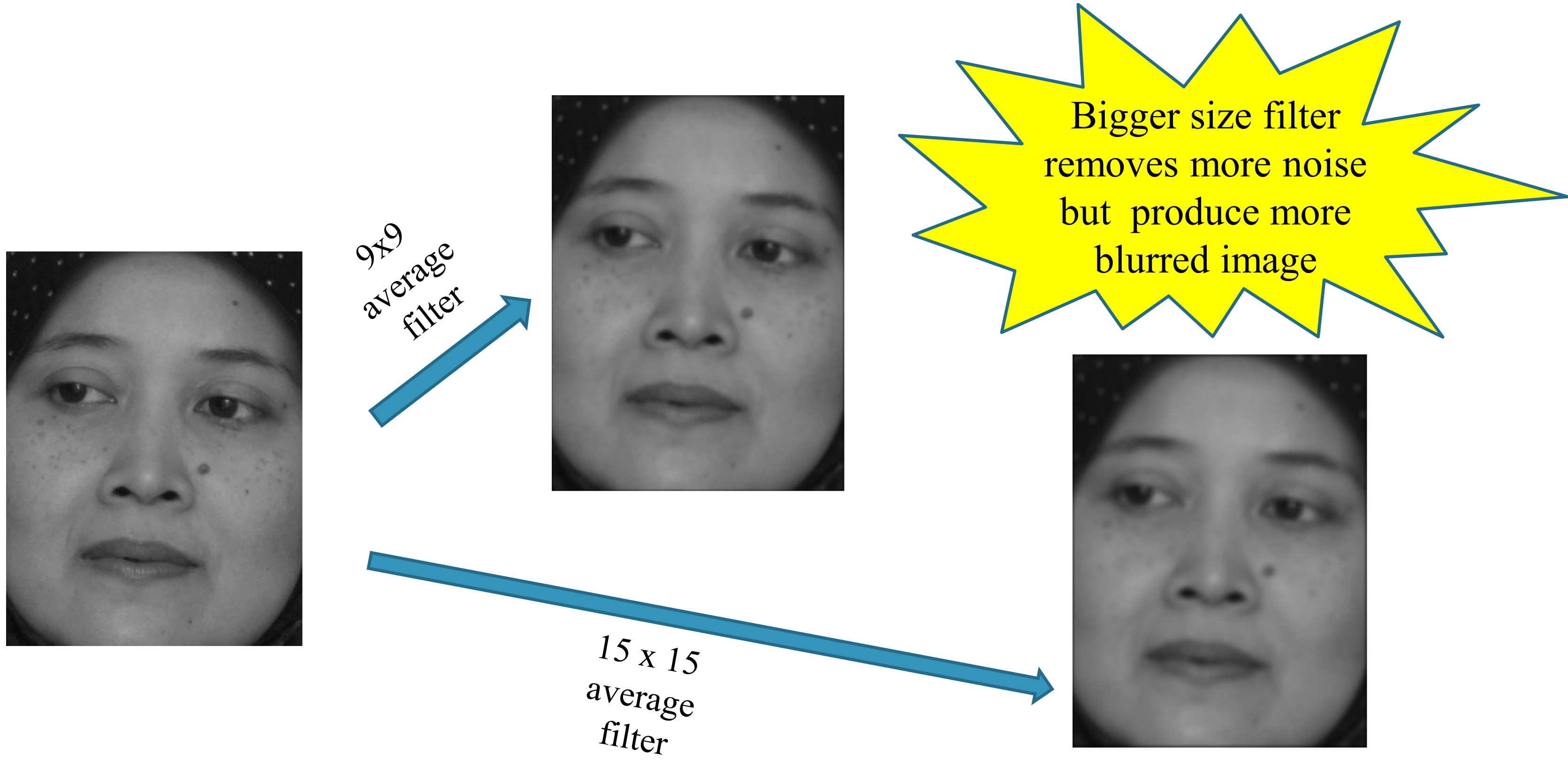
100	110	120	30
15	16	130	140
110	120	22	23
111	111	140	150

Step 2. Calculate average filtering using neighbourhood transformation

New pixel = (1/9*16) + (1/9*130) + (1/9*140) + (1/9*120) + (1/9*22) + (1/9*23) + (1/9*111) + (1/9*140) + (1/9*150) = **94.6**

AVERAGE FILTERING DEMO

Average filter can be used to smooth out fine lines and details, thus the name smoothing filter. Can be used to remove flaws in an image.



MEDIAN FILTER

An order-statistic filter whose result is based on ranking the pixels in the filter and replacing the center pixel with the ranking result. Min and max are also order statistic filters which are also known as nonlinear filters.

HOW DOES MEDIAN FILTER WORKS?

Given a 4x4 image:

100	110	120	130
15	16	130	140
110	120	**120**	23
111	111	140	150

Lets filter the centre pixel with a 3x3 median filter

Step 1. Rank the 9 neighbourhood pixels
16, 22, 23, 111, 120, 130, 140, 140, 150

Step 2. Replace the centre pixel with the median value
16, 22, 23, 111, 130, 140, 140, 150

AVERAGE OR MEDIAN FILTERING?

Median filter

- Best for salt & pepper noise.
- Does not create new unrealistic pixel values at edges. Thus, much better at preserving sharp edges.
- However, relatively expensive and complex to compute.

SHARPENING FILTERS

When do use sharpening filers?

- To highlight fine detail or to enhance blurred detail.

How does it work?

- It boosts areas where there is a repeated change and deemphasize areas of slow change in intensity.

What are their characteristics?

- High positive value for the centre pixel and negative values or 0 for the rest of the template cell

LAPLACIAN FILTERS

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

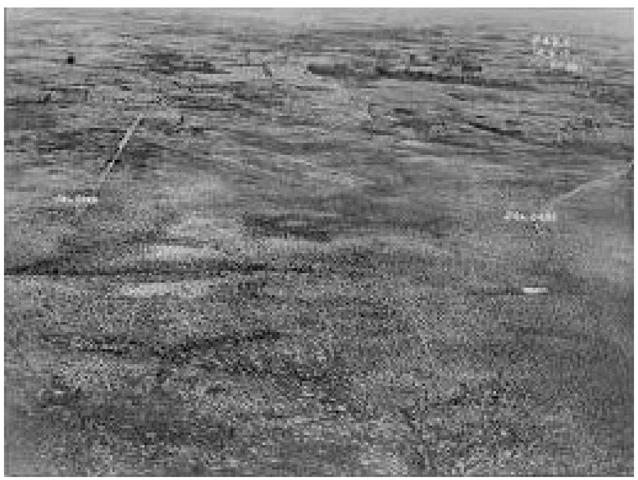
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

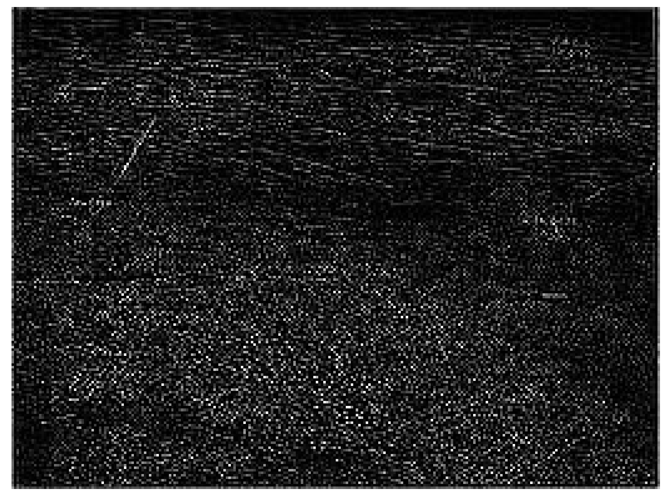
Particularly good at finding the fine details in an image.
Any feature with a sharp discontinuity be enhanced by a Laplacian operator.

LAPLACIAN FILTERING DEMO

Convolve the 3 x 3 positive Laplacian filter with the grayscale image.



Grayscale image



Convolved image

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

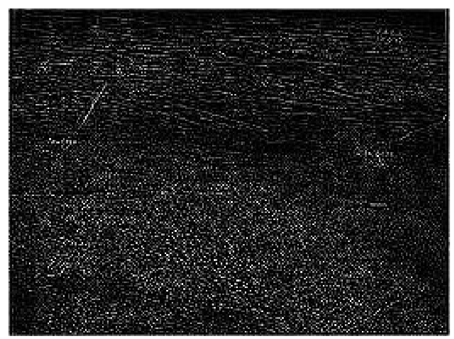
Laplacian filter

LAPLACIAN FILTERING

To obtain the sharpened image, the original grayscale image is added to the Laplacian filtered image

Grayscale image

+



Convolved image

=

Sharpened image

UNSHARP MASKING

Un-sharpens an image and uses the difference with the original image as a mask to increase the contrast of the image.

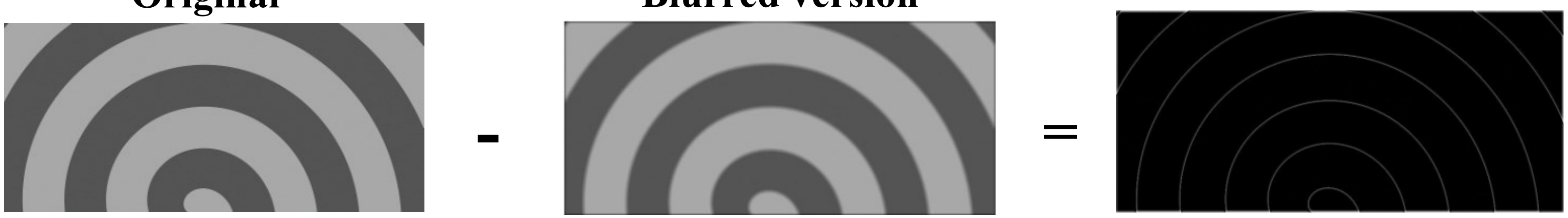
Steps:

1. Blur the original image
2. Subtract the blurred image from the original
3. Add the mask to the original

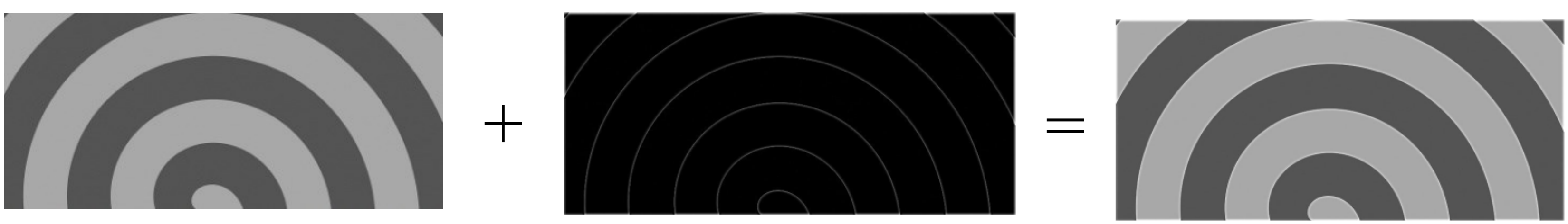
UNSHARP MASKING DEMO

Step 1. Subtract the original image with its blurred version (can be achieved by using average filtering) to create the mask.

Original **Blurred version** **Mask**



Step 2. Add the mask with the original image for the sharpened result



Original **Mask** **Sharpened**