# Arithmetic operations in Image Enhancement

# Enhancement Using Arithmetic/Logic Operations

- AND
- OR
- NOT
- Subtraction
- Addition
- Multiplication
- Division

# Image Enhancement in Spatial domain

- Arithmetic and Logic Operations are used for image enhancement
  - Arithmetic / logic operations are performed on a **pixel by pixel bases** between two or more images
  - Basic Arithmetic operations are:
  - Addition : $s(x,y) = f(x,y) + g(x,y)$
  - Subtraction : $d(x,y) = f(x,y) – g(x,y)$
  - Multiplication : $p(x,y) = f(x,y) . g(x,y)$
  - Division: $v(x,y) = f(x,y) / g(x,y)$

# Image Enhancement in Spatial Domain

- Logical operations: In gray scale images each pixel is of 8 bits. Logical operations are performed **bit by bit**

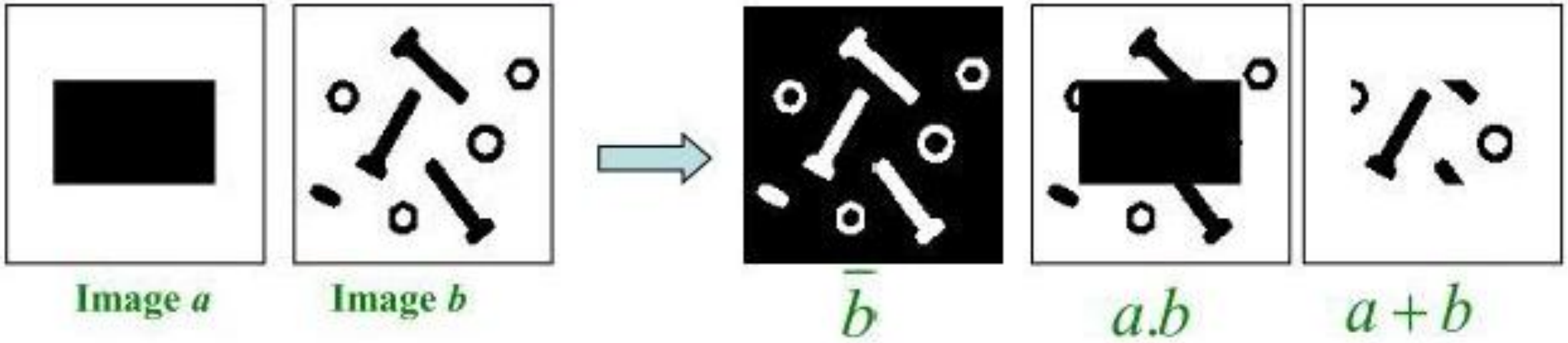
$$NOT \qquad c = \bar{a}$$

$$OR \qquad c = a + b$$

$$AND \qquad c = a \bullet b$$

$$XOR \qquad c = a \oplus b = a \bullet \bar{b} + \bar{a} \bullet b$$

# Example (logical operation)




Note: The images can be *binary (bi-level)* images. Each pixel is 1 *(True)* or 0 *(False)*.

AND and OR operations are used for Image Masking operation (i.e. to extract the **Region of Interest (ROI)** of a image

# Image Masking

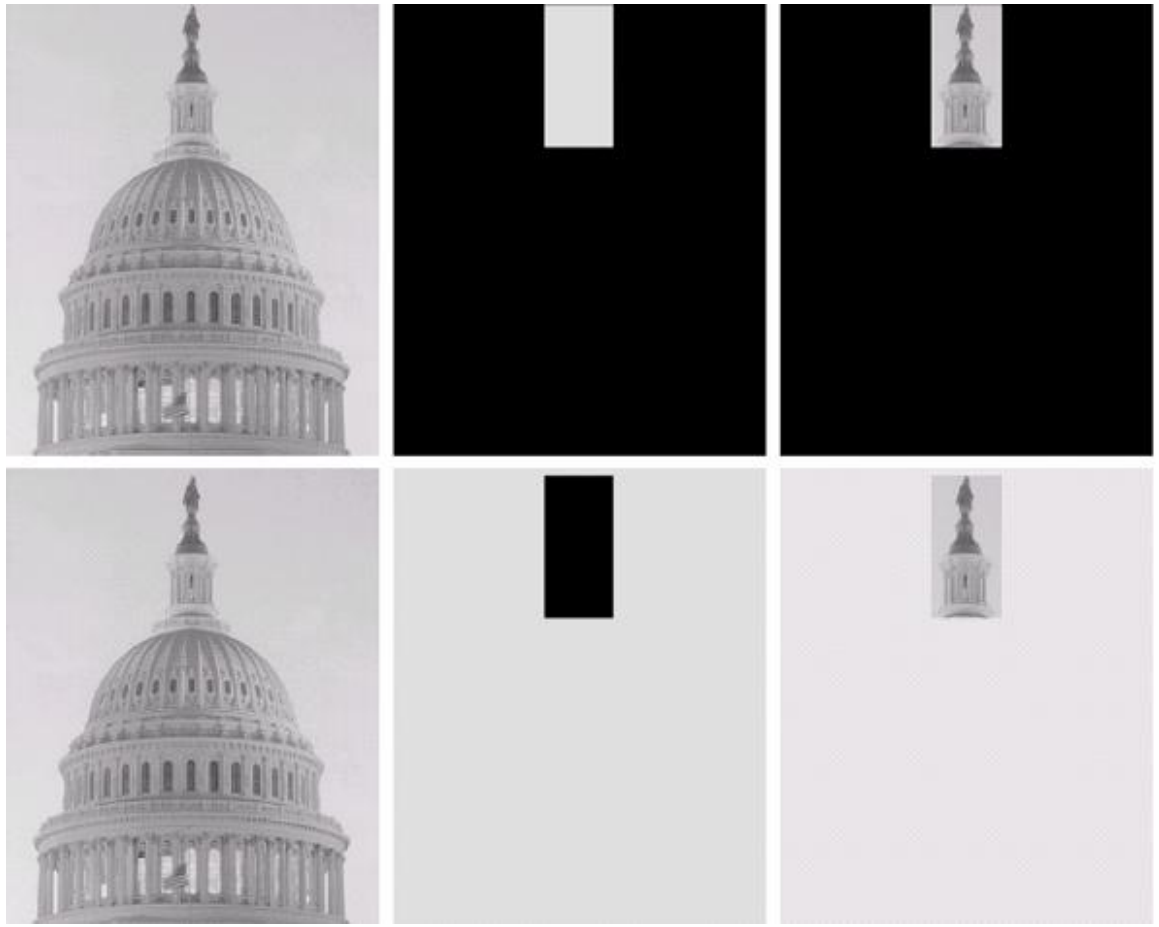


Image 1

Image 2
Mask

Result of And Operation

Portion of image which coincides with mask appears at the output with background turned black

Result of OR Operation

Portion of image which coincides with mask appears at the output with background turned white

# Example of Addition

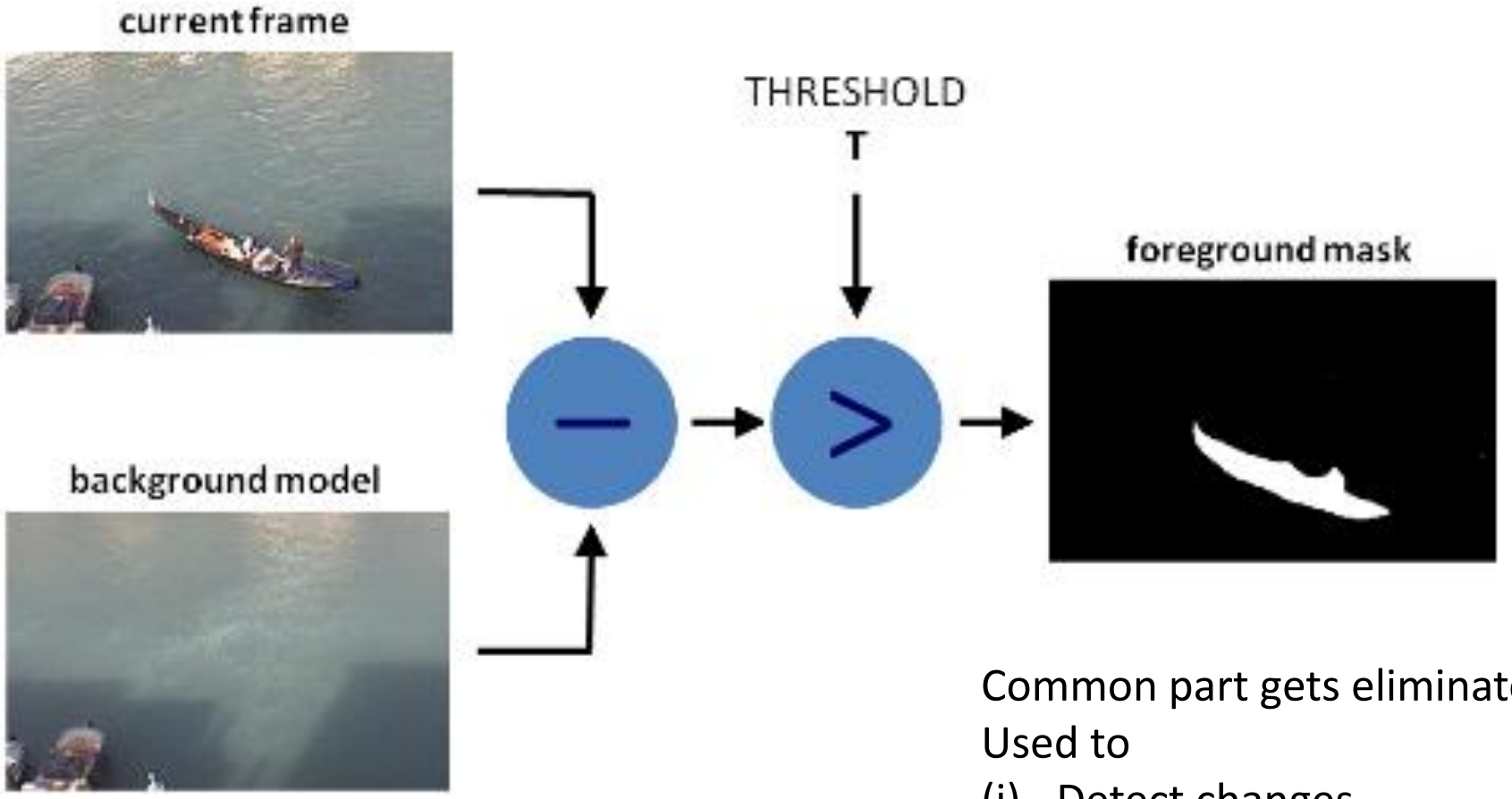
a) 

b) 

c) 

- Superimposes one image on other
- Used to Change the image background
- Watermarking Images

# Example of subtraction




Common part gets eliminated
Used to
(i) Detect changes
(ii) Survillance
(iii) Medical applications
(iv) etc

# Example of Division

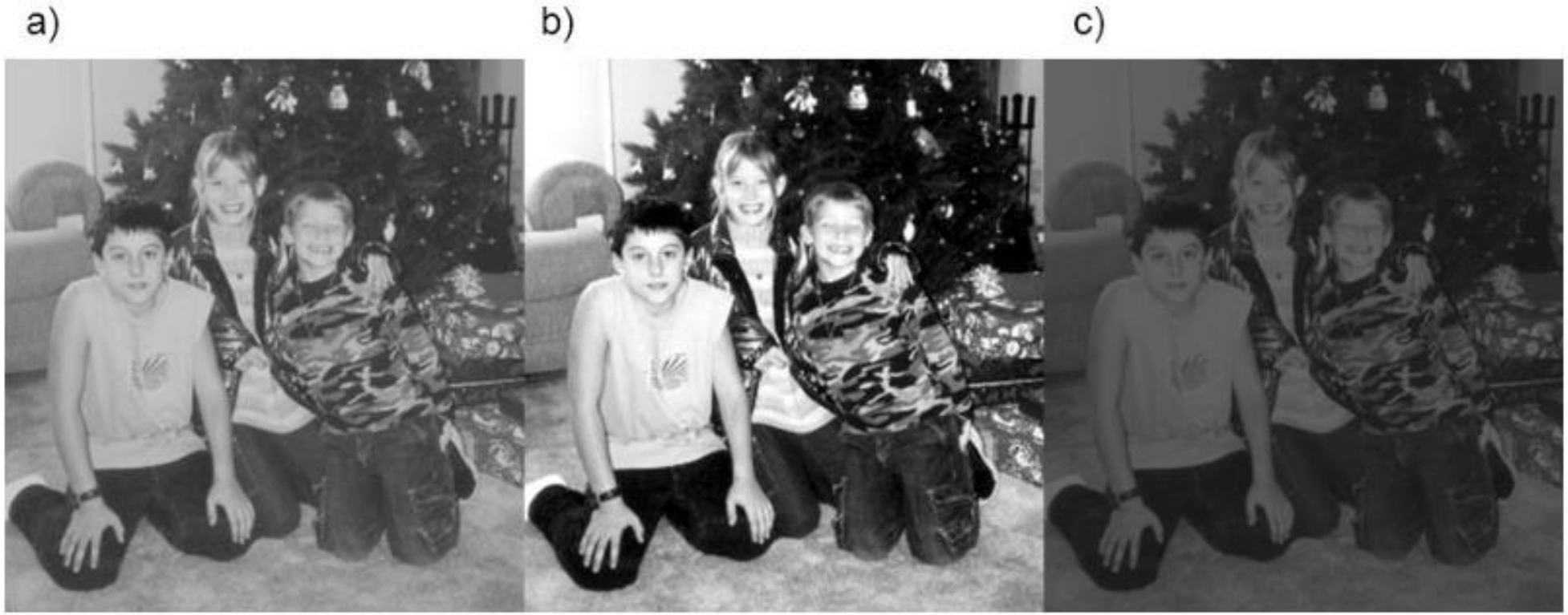



**Image Division**. a) original image, b) image divided by a value less than 1 to brighten, c) image divided a value greater than 1 to darken

# Example of Arithmetic Operation for Image Enhancement

$Addition$

$I = I_1 + I_2$

$I_1$

| | | |
|---|---|---|
| 10 | 10 | 10 |
| 11 | 11 | 10 |

+

$I_2$

| | | |
|---|---|---|
| 0 | 0 | 11 |
| 0 | 0 | 10 |

=

$I$

| | | |
|---|---|---|
| 10 | 10 | 21 |
| 11 | 11 | 20 |

$Subtraction$

$I = I_1 - I_2$

$I_1$

| | | |
|---|---|---|
| 10 | 10 | 10 |
| 11 | 11 | 10 |

−

$I_2$

| | | |
|---|---|---|
| 0 | 0 | 11 |
| 0 | 10 | 11 |

=

$I$

| | | |
|---|---|---|
| 10 | 10 | 1 |
| 11 | 1 | 1 |

# Few important rules

- If the result is a floating point number, round off its value
- If the result is above pixel range, select the max range
- If the result is below the pixel range select the min range value
- If the result is infinity write as 0

# Example of Arithmetic Operation for Image Enhancement

$$Addition$$

$$I = I_1 + I_2$$

$I_1$

| | | |
|---|---|---|
| 0 | 100 | 10 |
| 4 | 0 | 10 |
| 8 | 0 | 5 |

+

$I_2$

| | | |
|---|---|---|
| 10 | 100 | 5 |
| 2 | 0 | 0 |
| 0 | 10 | 10 |

=

$I$

| | | |
|---|---|---|
| 10 | 200 | 15 |
| 6 | 0 | 10 |
| 8 | 10 | 15 |

$$Subtraction$$

$$I = I_1 - I_2$$

$I_1$

| | | |
|---|---|---|
| 0 | 100 | 10 |
| 4 | 0 | 10 |
| 8 | 0 | 5 |

−

$I_2$

| | | |
|---|---|---|
| 10 | 100 | 5 |
| 2 | 0 | 0 |
| 0 | 10 | 10 |

=

$I$

| | | |
|---|---|---|
| 0 | 0 | 5 |
| 2 | 0 | 10 |
| 8 | 0 | 0 |

# Example of Arithmetic Operation for Image Enhancement

$$Multiplicaiton$$

$$I = I_1 * I_2$$

$I_1$

| | | |
|---|---|---|
| 0 | 100 | 10 |
| 4 | 0 | 10 |
| 8 | 0 | 5 |

*

$I_2$

| | | |
|---|---|---|
| 10 | 100 | 5 |
| 2 | 0 | 0 |
| 0 | 10 | 10 |

=

$I$

| | | |
|---|---|---|
| 0 | 255 | 50 |
| 8 | 0 | 0 |
| 0 | 0 | 50 |

Uses :

Shading correction

Marking or region of interest (ROI) operations

# Example of Arithmetic Operation for Image Enhancement

$$Division$$

$$I = I_1 / I_2$$

$I_1$

| | | |
|---|---|---|
| 0 | 100 | 10 |
| 4 | 0 | 10 |
| 8 | 0 | 5 |

/

$I_2$

| | | |
|---|---|---|
| 10 | 100 | 5 |
| 2 | 0 | 0 |
| 0 | 10 | 10 |

=

$I$

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 2 | 0 | 0 |
| 0 | 0 | 0 |

Uses :

Shading correction

Marking or region of interest (ROI) operations

# Logical Operations on Images

- And Operation

*AND*

| *x* | *y* | *xy* |
|---|---|---|
| *0* | *0* | *0* |
| *0* | *1* | *0* |
| *1* | *0* | *0* |
| *1* | *1* | *1* |

| $I_1$ | | |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 0 | 1 |

AND

| $I_2$ | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 0 |
| 0 | 0 | 1 |

=

| $I$ | | |
|---|---|---|
| 0 | 1 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 1 |

# Logical Operations on Images

- OR Operation

*OR*

| *x* | *y* | *x+y* |
|---|---|---|
| *0* | *0* | *0* |
| *0* | *1* | *1* |
| *1* | *0* | *1* |
| *1* | *1* | *1* |

$I_1$

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 0 | 1 |

OR

$I_2$

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 0 |
| 0 | 0 | 1 |

=

$I$

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |

# Not Operation

NOT

| $x$ | $x'$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

I

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 0 | 1 |

Not of I

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 0 |

# AND and OR used for image masking

Original Image

Mask for AND operation

Output of AND operation

a) b) c)

Mask for OR operation is by performing NOT of mask for AND operation

Output of OR operation for original image and OR mask