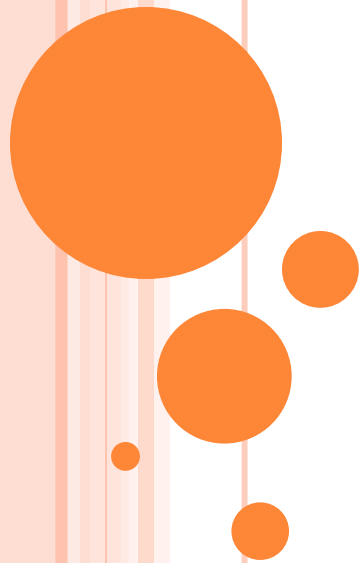


# IMAGE PROCESSING

## Image Analysis

### Ch2 – part 1



## Image Analysis

---

Image analysis methods extract information from an image by using automatic or semiautomatic techniques.

Image analysis differs from other types of image processing methods, such as enhancement or restoration in that the final result of image analysis procedures is a numerical output rather than a picture.

Image analysis is primarily data reduction process. As we have seen, images contain enormous amount of data, typically on the order hundreds of kilobytes or even megabytes. Often much of this information is not necessary to solve a specific computer imaging problem, so primary part of the image analysis task is to determine exactly what information is necessary.



## Image Analysis

---

In image processing application, image analysis methods may be used to help determine the type of processing required and the specific parameters needed for that processing. For example, determine the degradation function for an image restoration procedure, developing an enhancement algorithm and determining exactly what information is visually important for image compression methods



# System Model

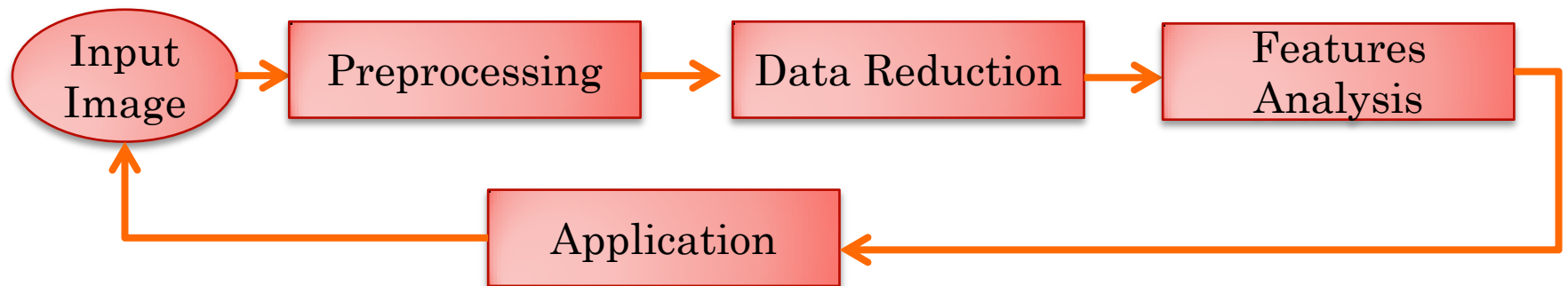
---

The image analysis process can be broken down into three primary stages:

**1-Preprocessing**

**2-Data Reduction**

**3-Features Analysis**



# Preprocessing operations

---

The preprocessing algorithm, techniques and operators are used to perform initial processing that makes the primary data reduction and analysis task easier. They include operations related to:

- Extracting regions of interest.
- Performing basic algebraic operation on image.
- Enhancing specific image features.
- Reducing data in resolution and brightness.



## Region of Interest Image Geometry

---

For image analysis we want to investigate more closely a specific area within the image, called region of interest (ROI). To do this we need operation that modifies the spatial coordinates of the image, and these are categorized as image geometry operations. The image geometry operations include:

### Crop, Zoom, Translate, Rotate

The image crop process is the process of selecting a small portion of the image, a sub image and cutting it away from the rest of the image. After cropped a sub image from the original image we can enlarge sub image by numerous ways:




# Region of Interest Image Geometry

---

For image analysis we want to investigate more closely a specific area within the image, called region of interest (ROI). To do this we need operation that modifies the spatial coordinates of the image, and these are categorized as image geometry operations. The image geometry operations include:

## Crop, Zoom, Translate, Rotate

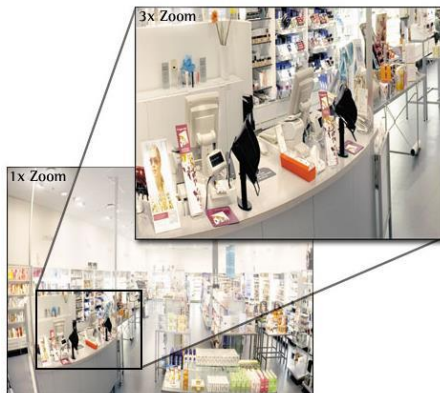
The image crop process is the process of selecting a small portion of the image, a sub image and cutting it away from the rest of the image. After cropped a sub image from the original image we can **enlarge** sub image by numerous ways:



# 1-Zero Order Hold

is performed by repeating previous pixel values, In zero order hold the output image size is double original image size (  $2n \times 2n$  ).

Original Image Array	Image with Rows Expanded	Image with rows and columns expanded
$\begin{bmatrix} 8 & 4 & 8 \\ 4 & 8 & 4 \\ 8 & 2 & 8 \end{bmatrix}$	$\begin{bmatrix} 8 & 8 & 4 & 4 & 8 & 8 \\ 4 & 4 & 8 & 8 & 4 & 4 \\ 8 & 8 & 2 & 2 & 8 & 8 \end{bmatrix}$	$\begin{bmatrix} 8 & 8 & 4 & 4 & 8 & 8 \\ 8 & 8 & 4 & 4 & 8 & 8 \\ 4 & 4 & 8 & 8 & 4 & 4 \\ 4 & 4 & 8 & 8 & 4 & 4 \\ 8 & 8 & 2 & 2 & 8 & 8 \\ 8 & 8 & 2 & 2 & 8 & 8 \end{bmatrix}$





## 2-First Order Hold

is performed by finding the average value between two adjacent pixels and use that as the pixel value between those two, This method allows us to enlarge an  $N \times N$  sized image to a size of  $(2N-1) \times (2N-1)$ .

Original Image Array	Image with Rows Expanded	Image with rows and columns expanded
$\begin{bmatrix} 8 & 4 & 8 \\ 4 & 8 & 4 \\ 8 & 2 & 8 \end{bmatrix}$	$\begin{bmatrix} 8 & 6 & 4 & 6 & 8 \\ 4 & 6 & 8 & 6 & 4 \\ 8 & 5 & 2 & 5 & 8 \end{bmatrix}$	$\begin{bmatrix} 8 & 6 & 4 & 6 & 8 \\ 6 & 6 & 6 & 6 & 6 \\ 4 & 6 & 8 & 6 & 4 \\ 6 & 5.5 = 6 & 5 & 5.5 = 6 & 6 \\ 8 & 5 & 2 & 5 & 8 \end{bmatrix}$



## 3-Convolution

---

This method required two steps:

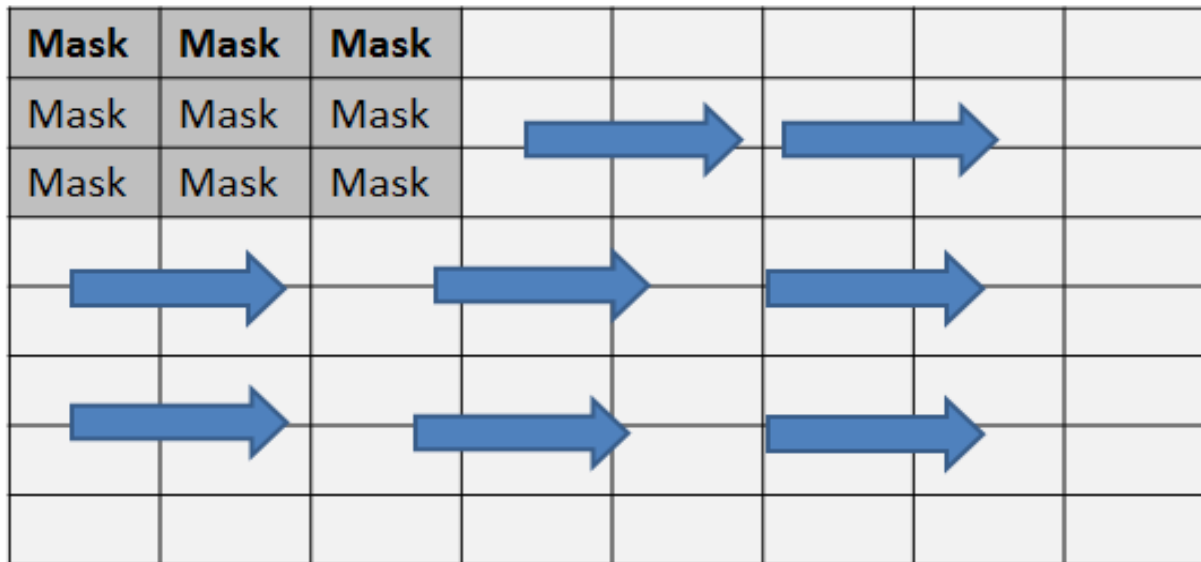
- A. Extend the image by adding rows and columns of zeros between the existing rows and columns.
- B. Perform the convolution.

Original Image Array	Image extended with zeros
$\begin{bmatrix} 3 & 5 & 7 \\ 2 & 7 & 6 \\ 3 & 4 & 9 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 7 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

Next, we use convolution mask, which is slide a cross the extended image, and perform simple arithmetic operation at each pixel location

## Convolution mask for first-order hold

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

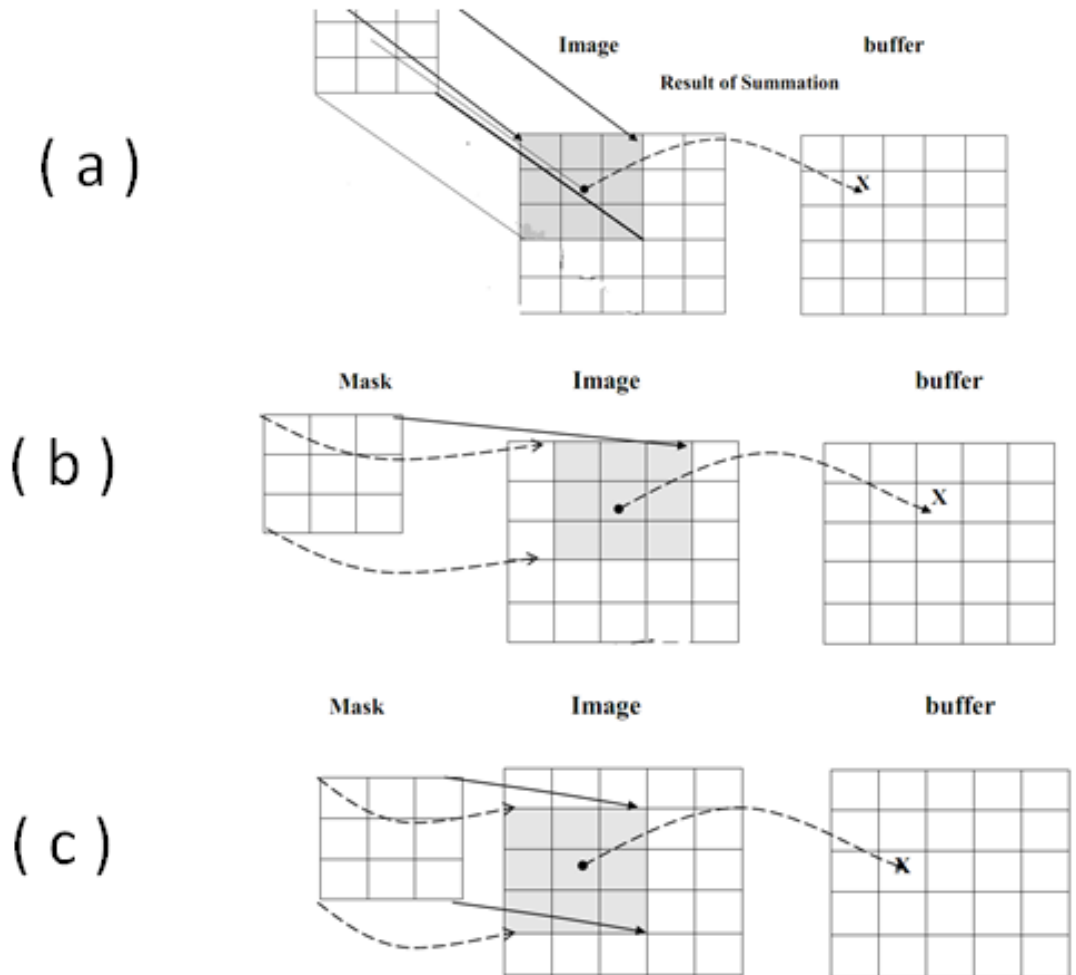


convolution process (from left to right, and top to bottom).  
the process of sliding, multiplying and summing is called convolution.

# The convolution process

The output image must be put in a separate image array called a buffer

**Note** : many computer boards can perform convolution in hardware, which is generally very fast, typically much faster than applying a faster algorithm in software.



## *Zero-order hold convolution mask*

---

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

**Note** for this mask we will need to put the result in the pixel location corresponding to the lower-right corner because there is no center pixel.

These methods will **only** allows us to enlarge an image by a factor of  $(2N-1)$ , but what if we want to enlarge an image by something other than a factor of  $(2N-1)$ ?



## *If we need to enlarge image more of 2 times*

---

To do this we need to apply a more general method. We take two adjacent values and linearly interpolate more than one value between them. This is done by define an enlargement number  $k$  and then following this process:

1. Subtract the result by  $k$ .
2. Divide the result by  $k$ .
3. Add the result to the smaller value, and keep adding the result from the second step in a running total until all  $(k-1)$  intermediate pixel locations are filled.



**Example: we want to enlarge an image to five times its original size, and we have two adjacent pixel values 210 and 240 .**

1. Find the difference between the two values,  $240 - 210 = 30$  .

2. The desired enlargement is  $k= 5$ , so we get  $30 /5 = 6$  .

3. Next determine how many intermediate pixel values .we need :

$K-1= ( 5 -1 )= 4$ . The four pixel values between the 210 and 240 are :

Where :

$K = 1 :$

$$210 + ( 6*1 ) = 216 .$$

$K= 2:$

$$210 + ( 6*2 ) = 222 .$$

$K= 3 :$

$$210 + ( 6*3 ) = 228 .$$

$K= 4 :$

$$210 + ( 6*4 ) = 234 .$$

We do this for every pair of adjacent pixels .first along the rows and then along the columns. This will allows us to enlarge the image by any factor of  $K (N-1) +1$  where  $K$  is an integer and  $N \times N$  is the image size.

# Image Shrinking

Image shrinking is accomplished by taking groups of pixels that are spatially adjacent and mapping them to one pixel. This can be done in one of three ways:

- **Averaging**
- **Median**
- **Decimation.**

**EXAMPLE:** For the below  $2 \times 2$  block of pixels, shrink this block using the three methods of shrinking.

$$\begin{bmatrix} 2 & 6 \\ 8 & 4 \end{bmatrix}$$

Averaging	Median	Decimation
$2+6+4+8=20/4=5$	$2,4,6,8=(4+6)/2=5$	Eliminate the first row and first column and the result is 4





# Translation and Rotation

---

Two other operations of interest in image geometry are: Translation and Rotation. These processes may be performed for many application specific reasons, for example to align an image with a known template in a pattern matching process or to make certain image details easier to see.

