Al-Mustaqbal University
College of Sciences
Intelligent Medical System Department

جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الانظمة الطبية الذكية

Lecture (6 & 7): Image Convolution

Subject: Image Processing
Level: Third
Lecturer: Asst. Lecturer Qusai AL-Durrah

Image Convolution:

Convolution: a mathematical operation that processes an image using a small matrix called a **kernel (or filter)**. It is used to extract features such as edges, textures, and patterns in medical images like X-rays, MRIs, and CT scans.

Why is Convolution Important in Medical Imaging?

- **Enhances image quality** (e.g., noise reduction, sharpening)
- **Feature extraction** for segmentation and classification
- **Foundation for deep learning** in medical AI (e.g., CNNs)

Applications of Convolution in Medical Imaging

a) Noise Reduction (Smoothing)

- Reduces noise in X-rays, MRI scans, and ultrasound images.

b) Edge Detection

- Identifies **tumors**, **blood vessels**, and **organ boundaries**.
- Useful in **cancer detection** and **retinal disease analysis**.

c) Image Sharpening

- Enhances details in CT and MRI scans for better diagnosis.

d) Deep Learning in Medical Image Analysis

- **CNNs (Convolutional Neural Networks)** use multiple convolution layers to detect patterns in medical images.
- Used in **tumor classification**, **disease detection**, and **organ segmentation**.

First-Order Hold Convolution Mask: this process requires a mathematical process to enlarge an image. This method required two steps:

1. Extend the image by adding rows and columns of zeros between the existing rows and columns.
2. Perform the convolution.

The image is extended as follows:

Original Image Array

$$\begin{pmatrix} 3 & 5 & 7 \\ 2 & 7 & 6 \\ 3 & 4 & 9 \end{pmatrix}$$

Image extended with zeros

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 7 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

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

Convolution mask for first –order hold

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

The convolution process requires us to overlay the mask on the image, multiply the coincident values and sum all these results. This is equivalent to finding the vector inner product of the mask with underlying sub image. The vector inner product is

found by overlaying mask on sub image. Multiplying coincident terms, and summing the resulting products.

For example, if we put the mask over the upper-left corner of the image, we obtain (from right to left, and top to bottom):

1/4(0) +1/2(0) +1/4(0) +1/2(0) +1(3) +1/2(0) + 1/4(0) +1/2(0) +1/4(0) =3

Note that the existing image values do not change. The next step is to slide the mask over by on pixel and repeat the process, as follows:

1/4(0) +1/2(0) +1/4(0) +1/2(3) +1(0) +1/2(5) + 1/4(0) +1/2(0) +1/4(0) =4

Note this is the average of the two existing neighbors. This process continues until we get to the end of the row, each time placing the result of the operation in the location corresponding to center of the mask.

When the end of the row is reached, the mask is moved down one row, and the process is repeated row by row. This procedure has been performed on the entire image, the process of sliding, multiplying and summing is called convolution.

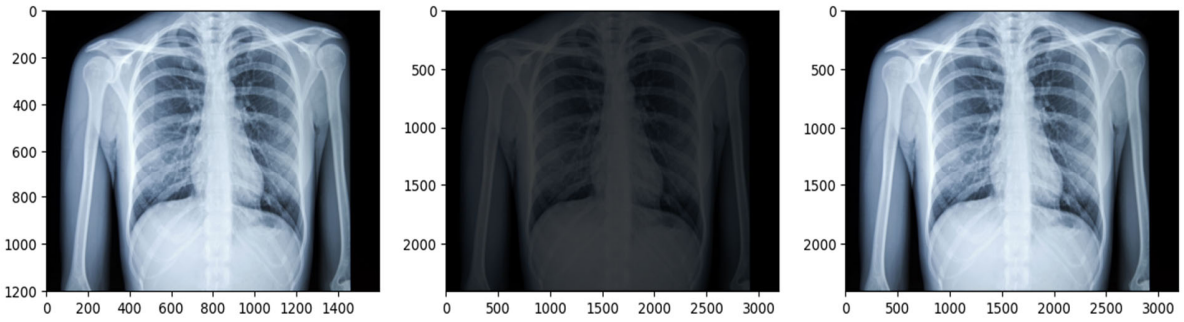


Figure (1): First _Order Hold Method

First-Order Hold Method Requirements

Minimum Enlargement: 2x

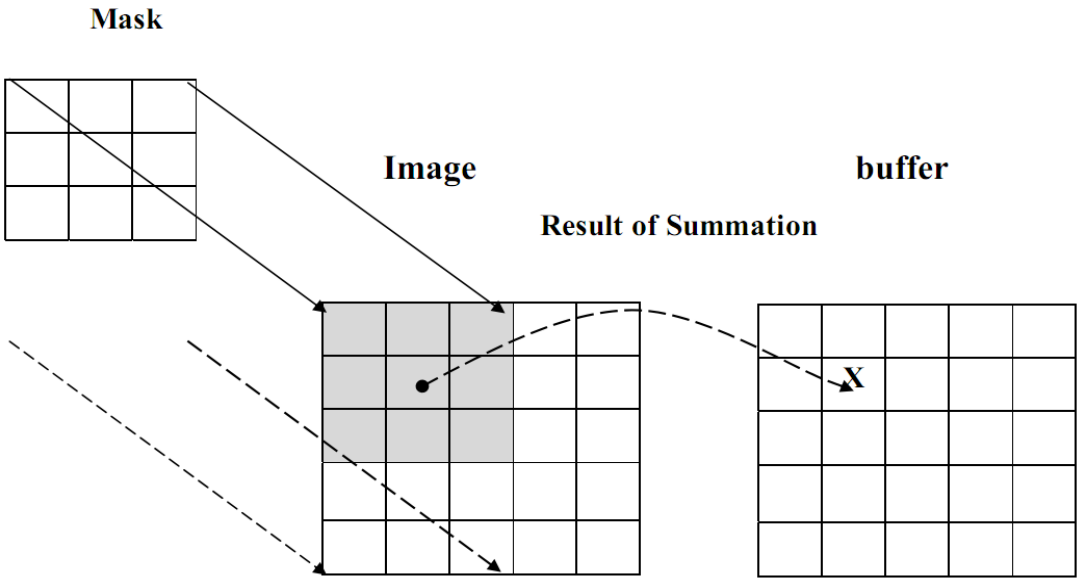
Mask Size: 3x3

Buffer Required: 1

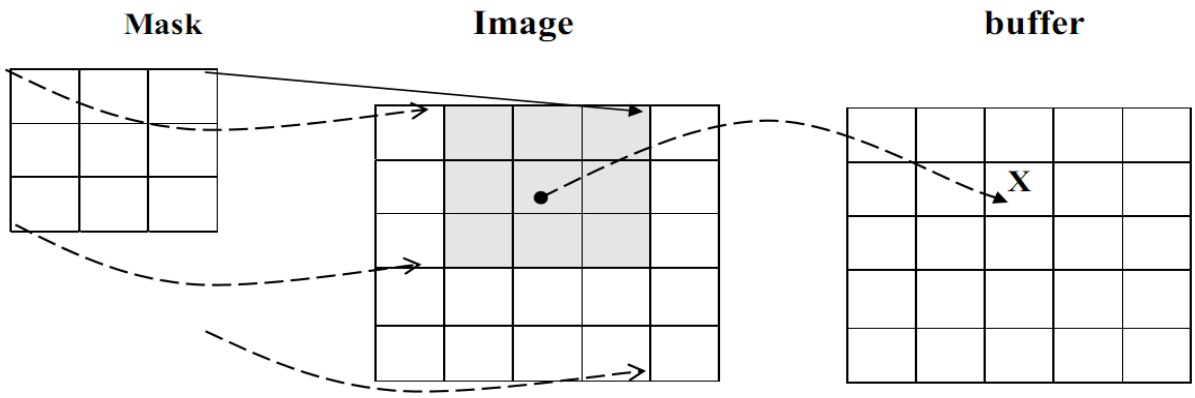
Separate image array needed to prevent overwriting values

Note that the output image must be put in a separate image array called a buffer, so that the existing values are not overwritten during the convolution process.

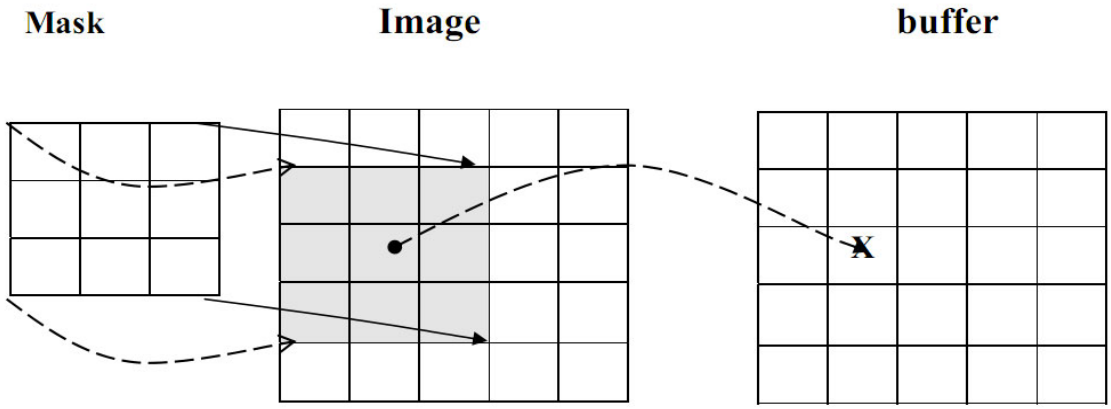
The convolution process



a. Overlay the convolution mask in the upper-left corner of the image. Multiply coincident terms, sum, and put the result into the image buffer at the location that corresponds to the masks current center, which is (r,c)=(1,1).



b. Move the mask one pixel to the right, multiply coincident terms sum, and place the new results into the buffer at the location that corresponds to the new center location of the convolution mask which is now at (r,c)=(1,2), continue to the end of the row.



c. Move the mask down on row and repeat the process until the mask is convolved with the entire image. Note that we lose the outer row(s) and columns(s).

Application of FOH Masks in Medical Image Processing

- **Image Upsampling**: Helps reconstruct higher-resolution images from low-resolution scans (e.g., MRI, CT).
- **Denoising**: Reduces minor noise while preserving details.

- **Feature Enhancement**: Useful in preprocessing for segmentation tasks.

Note, not only first-order hold be performed via convolution, but zero-order hold can also be achieved by extending the image with zeros and using the following convolution mask.

Zero-order hold convolution mask

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

Note that 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 allow 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)?

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 three times its original size, and we have two adjacent pixel values 125 and 140.

1. Find the difference between the two values, 140-125 =15.

2. The desired enlargement is k=3, so we get 15/3=5.

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

K-1=3-1=2. The two-pixel values between the 125 and 140 are:

125+5=130 and 125+2*5 = 135.

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

• To process opposite to enlarging an image is shrinking. This process is done by reducing the amount of data that need to be processed.

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