IDC410
A course on Image Processing and Machine Learning
(Lecture 05)

Shashikant Dugad,

IISER Mohali

Filters (contd..)

Properties of Convolution Filter

- Notation: $b = c \star a$
- Convolution is a multiplication-like operation
  - Commutative: $a \star b = b \star a$
  - Associative: $a \star (b \star c) = (a \star b) \star c$
  - Distributes over addition: $a \star (b + c) = (a \star b) + (a \star c)$
  - Scalars factor out: $\alpha a \star b = a \star \alpha b = \alpha(a \star b)$
  - Identity: unit impulse $e = [\ldots, 0, 0, 1, 0, 0, \ldots]$: $a \star e = a$
- Usefulness of associativity
  - Often apply several filters one after another: $(((a \star b1) \star b2) \star b3)$
  - This is equivalent to applying one filter: $a * (b1 \star b2 \star b3)$

Padding and Strides

- **In Convolutional Neural Networks (CNNs), stride refers to the step size by which the filter/kernel moves across the input image during the convolution operation in horizontal and vertical direction**
    - **Stride defines how big of steps filters should take (i.e., how many pixels our filters should skip) while sliding over the image**
    - **Minimum value of stride = 1**
    - **Smaller strides (1 or 2) offer detailed feature extraction, while larger strides (3+) helps in down-sampling.**

- **CNN is like a collection of small, overlapping magnifying glasses called filters. These filters scan over different parts of a image to find interesting features, like edges, shapes, or colours. These filters slide or convolve over the entire image as defined by the stride.**

- **The kernel size, stride value can substantially reduce the size of output im the output size and computational efficiency of the network, influencing feature extraction and spatial dimensions of image.**

Movement of Filter

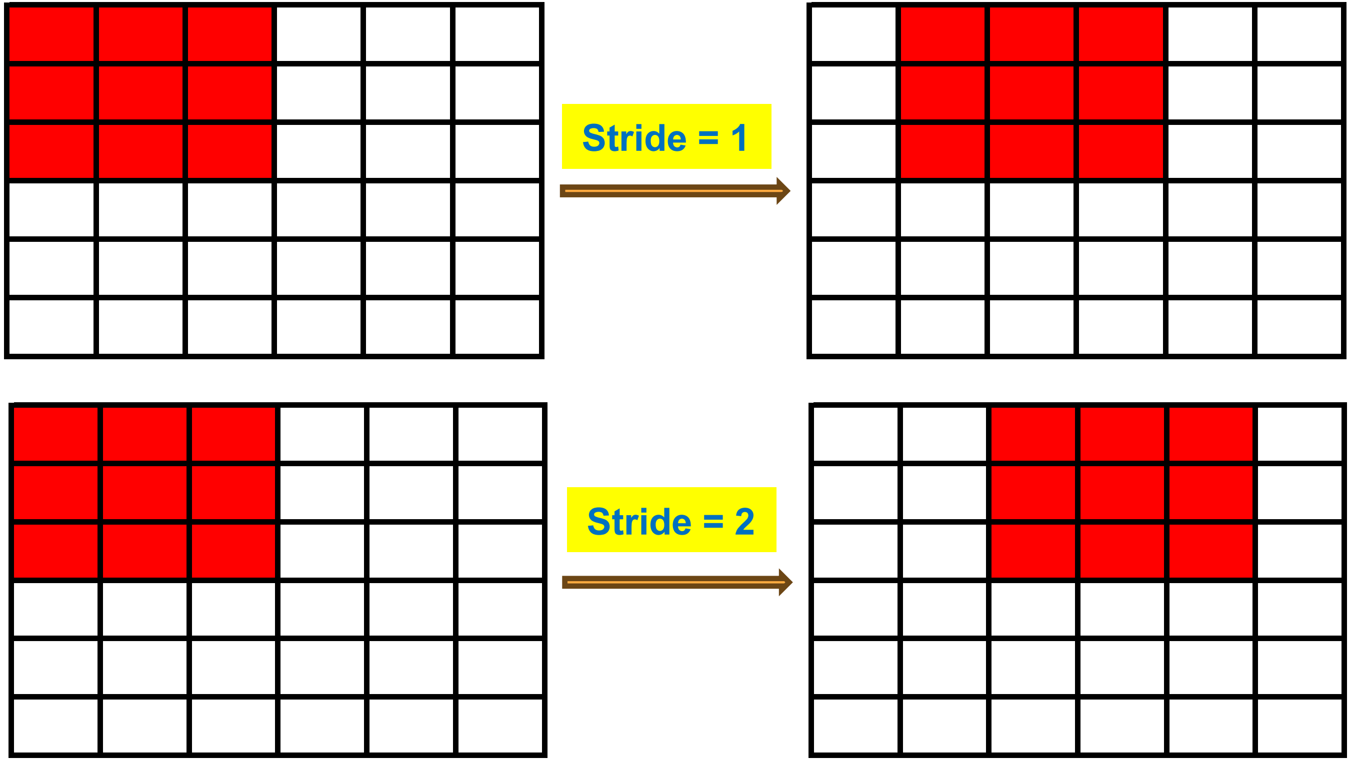


Image Padding

Input Image (5 x 5)

22	145	23	167	67
45	110	45	119	29
78	99	78	88	112
145	100	99	38	164
223	110	23	45	29

Pad = 1

Output Image (7 x 7) [Pad = 1]

0	0	0	0	0	0	0
0	22	145	23	167	67	0
0	45	110	45	119	29	0
0	78	99	78	88	112	0
0	145	100	99	38	164	0
0	223	110	23	45	29	0
0	0	0	0	0	0	0

Input Image (5 x 5)

22	145	23	167	67
45	110	45	119	29
78	99	78	88	112
145	100	99	38	164
223	110	23	45	29

Pad = 2

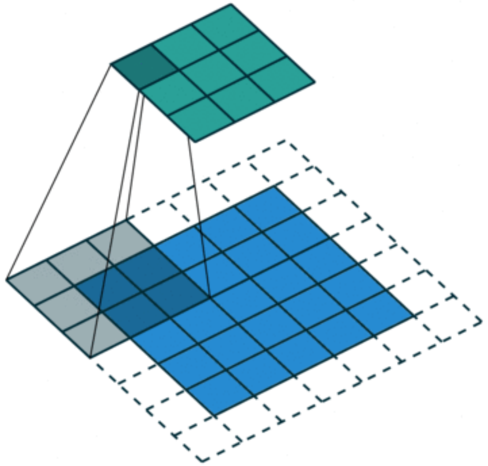
Output Image (9 x 9) [Pad = 2]

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	22	145	23	167	67	0	0
0	0	45	110	45	119	29	0	0
0	0	78	99	78	88	112	0	0
0	0	145	100	99	38	164	0	0
0	0	223	110	23	45	29	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

- **Padding means adding extra columns and rows with ZERO pixel intensity before doing any operations.**
- **Helps in keeping the spatial info intact, particularly prevents data loss at the edges hence stabilises training**
- **It also helps to keep the output size consistent with the input and makes training more stable.**

Padding and Strides

- Image Dimension: N_{row} X M_{col}
- Padding: P   Strides: S_{row}, $S_{col.}$ Dilation: D_{row}, D_{col}
- Kernal Size: K_{row}, K_{col}



$$N_{row}^{out} = \frac{N_{row}^{in} + 2\ X\ P\ \ - D_{row}\ X\ (K_{row} - 1) - 1}{S_{row}} + 1$$

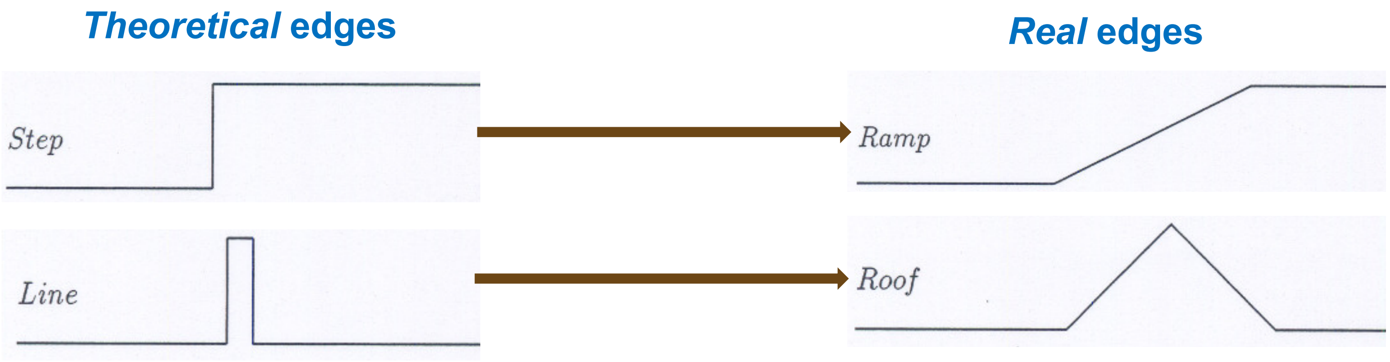
$$M_{col}^{out} = \frac{M_{col}^{in} + 2\ X\ P\ \ - D_{col}\ X\ (K_{col} - 1) - 1}{S_{col}} + 1$$

Edges

Edges

- Feature extraction is one of the crucial steps in image processing and machine learning
- Edges in the image is one such key feature.
  - Edges typically occur on the boundary between two different regions in an image
  - An edge in an image is usually associated with a discontinuity in the image intensity resulting large difference or a large amplitude of the first derivative of the image intensity
- Discontinuities in the image intensity can be either
  - Step discontinuities: the intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side
  - Line discontinuities: the intensity abruptly changes value but then returns to the starting value within some short distance.
- However, step and line edges are rare in real images due to the low-frequency components or the smoothing of the image.
- Step edges become ramp edges and line edges become roof edges

Edge Profile and Gradient



- An edge is associated with the maxima in the first derivative (gradient) of intensity in local region of an image
- The gradient is the two-dimensional equivalent of the first derivative and is defined as the vector

$$\vec{G}(f(x,y)) = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

Gradient

- The vector G[f(x, y)] points in the direction of the maximum rate of increase of the function f(x, y)

- The magnitude of the gradient, given by

$$|G(f(x, y)| = \sqrt{G_x^2 + G_y^2}$$

- The *direction* of the gradient is defined as,

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Gradient for digital image

- Gradient for a digital image can be defined as,

$$G_x(i,j) \cong f(i,j+1) - f(i,j). \qquad G_y(i,j) \cong f(i+1,j) - f(i,j)$$

- These can be implemented with simple convolution masks as shown below:

G_x =

-1	+1

G_y =

+1
-1

- Gradient has to be computed at exactly the same position in space.
  - However, gradients G_x and G_y are calculated at different points, [i, j+ 1/2] and [i+1/2, j] which are **NOT** the same
  - 3x3 convolution mask is preferred to maintain this criteria

Edge Detection Operators

- **Roberts Operator: Provides a convolution mask for following simple gradient operation: $G[f(i, j)] = G_x + G_y = |f(i, j) - f(i+1, j+1)| + |f(i+1, j) - f(i, j+1)|$**
- $G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ $\qquad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
- **The Roberts operator is NOT located at the desired point [i,j].**
- **In order to have an edge operation at fixed point [i,j] in both the directions, we should have a mask of 3x3 size**
- **Sobel operator is the magnitude of the gradient computed by,**

$$M = \sqrt{S_x^2 + S_y^2}$$

Edge Detection Operators

- Partial derivative are defined as per following matrix:

a0	a1	a2
a7	[i, j]	a3
a6	a5	a4

$S_x = (a2-a0) + c(a3-a7) + (a4-a6)$

$S_y = (a0-a6) + c(a1-a5) + (a2-a6)$

Higher weightage can be given to closer pixels by appropriately choosing c

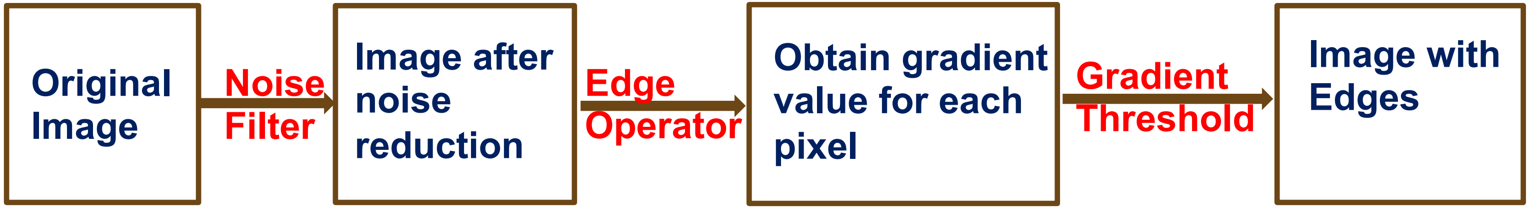
- *For Sobel operator c = 2;* S_x and S_y can be implemented using the following convolution masks:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- For Prewitt Operator c = 1: No emphasis is given to the closer pixels! Edge operators obtained with c = 1

Application of Edge Detection Technique

- Direct application of edge operator on a image may result in fake edges due to the noise in image
- First remove noise in the image
- Choose one of the edge detection operator appropriately and apply it on the image
- Obtain gradient for each pixel and apply a threshold on the gradient value to identify the pixel as a pixel representing edge



Edge Detection with and without Noise Filter

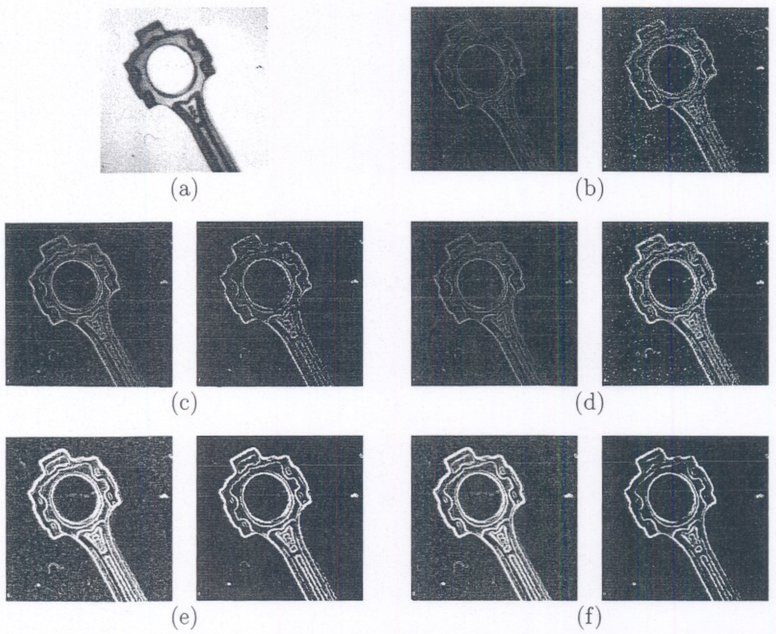


Figure 5.7: A comparison of various edge detectors on a noisy image without filtering. (a) Noisy image. (b) Simple gradient using 1×2 and 2×1 masks, $T = 64$. (c) Gradient using 2×2 masks, $T = 128$. (d) Roberts cross operator, $T = 64$. (e) Sobel operator, $T = 225$. (f) Prewitt operator, $T = 225$.

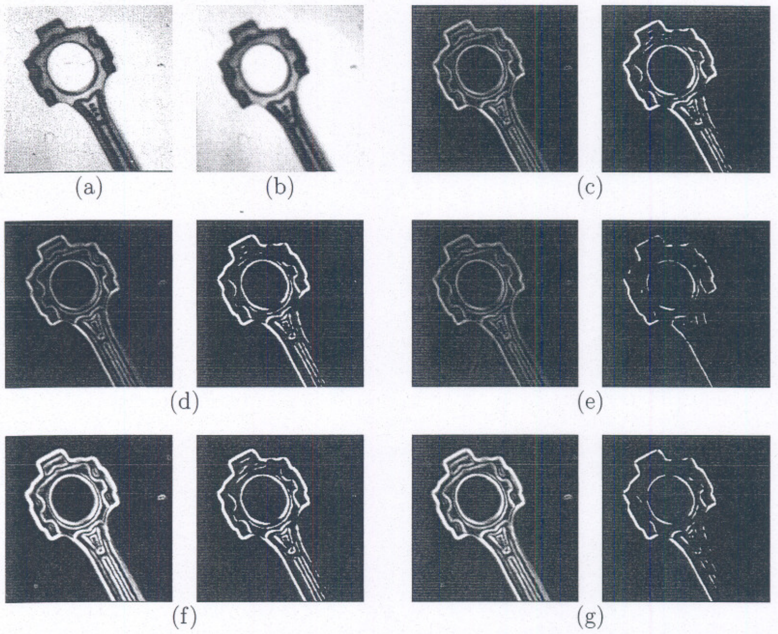
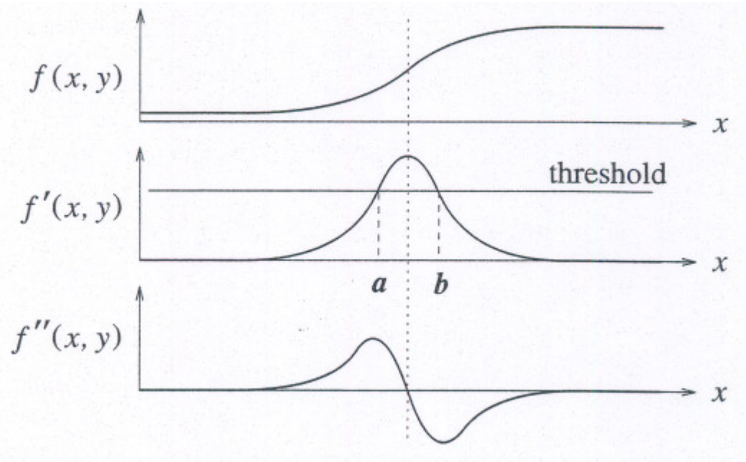


Figure 5.6: A comparison of various edge detectors on a noisy image. (a) Noisy image. (b) Filtered image. (c) Simple gradient using 1×2 and 2×1 masks, $T = 32$. (d) Gradient using 2×2 masks, $T = 64$. (e) Roberts cross operator, $T = 64$. (f) Sobel operator, $T = 225$. (g) Prewitt operator, $T = 225$.

https://cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/

Edge Detection with Second Derivative Operators

- **A single derivate edge operators with a threshold provides too many edge points depending on noise in the image**
- **A better approach would be to find only the points that have local maxima in gradient values and consider them a edge points.**
  - **This means that at edge points, there will be a peak in the first derivative and, equivalently, there will be a zero crossing in the second derivative.**



If a threshold is used for detection of edges, all points between *a* and *b* will be marked as edge pixels. However, by removing points that are *not* a local maximum in the first derivative, edges can be detected more accurately. Local maximum in the first derivative corresponds to a zero crossing in the second derivative.

Edge Detection with Second Derivative Operators

- **There are two operators in two dimensions that correspond to the second derivative: the Laplacian and second directional derivative.**

$$Gx = \frac{\partial f(x,y)}{\partial x} = f(i,j+1) - f(i,j)$$

$$\therefore \quad \frac{\partial^2 f(x,y)}{\partial x^2} = \frac{\partial G_x}{\partial x} = \frac{\partial f(i,j+1)}{\partial x} - \frac{\partial f(i,j)}{dx} = f(i,j+2) - f(i,j+1) - [f(i,j+1) + f(i,j)]$$

$$\frac{\partial^2 f(x,y)}{\partial x^2} = f(i,j+2) - 2f(i,j+1) + f(i,j)$$

- **However, this approximation is centered about the pixel *[i,j* + 1]. Therefore, by replacing j with j - 1,**

$$\frac{\partial^2 f(x,y)}{\partial x^2} = f(i,j+1) - 2f(i,j) + f(i,j-1). \qquad \frac{\partial^2 f(x,y)}{\partial y^2} = f(i+1,j) - 2f(i,j) + f(i-1,j)$$

Second Derivative Laplacian Operators

- **By combining these two equations into a single operator, the following mask can be used to approximate the Laplacian:**

$$\nabla^2 = \nabla^2_x + \nabla^2_y = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 0 & -2 & 0 \\ \hline 0 & 1 & 0 \\ \hline\end{array} + \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & -2 & 1 \\ \hline 0 & 0 & 0 \\ \hline\end{array} = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array}$$

- **It is desired to give weight to the corner pixels as below:**

$$\nabla^2 = \begin{array}{|c|c|c|}\hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline\end{array}$$

Laplacian of Gaussian (LoG) Operators

- Very small local peaks in the first derivative will also result in zero crossings the double derivative operators are quite sensitive to the noise

- The Laplacian operators has to be used in conjunction with powerful filtering methods

- Laplacian operator combined Gaussian filter referred as LoG operator

- The detection criterion: presence of a zero crossing in the second derivative with a corresponding large peak in the first derivative.

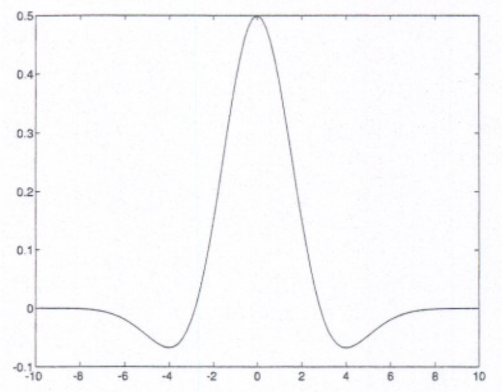
Laplacian of Gaussian (LoG) Operators

- **The output of the LoG operator, h(x, y), is obtained by the convolution operation**

- **h(x,y) = ∇^2[g(x,y)★f(x,y)] = [∇^2g(x,y)]★f(x,y)**

$$\nabla^2 g(x,y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right) e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

- **∇^2g(x,y) is offerent referred as *Mexican Hat operator* as shown in Figure**



5x5 LoG Convolution Mask

0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Laplacian of Gaussian (LoG) Operators

- Zero crossings may happen due to the noisy region of image
- The slope of the zero crossing depends on the contrast or sharpness of the change in image intensity across the edge.
- To obtain real edges in an image, it may be necessary to combine information from operators with several filter sizes or look at the amplitude of variation (1^{st} derivative)
- A larger σ results in better noise filtering but may lose important edge information, which may affect the performance of an edge detector. If a small filter is used, there is likely to be more noise due to insufficient averaging.

Laplacian of Gaussian (LoG) Operators

- The LoG operator which is symmetric; can reduce noise by smoothing the image, but it also dilutes the real edges resulting in uncertainty to the accurate location of the edge

- The gradient may have greater sensitivity to the presence of edges, but it has higher sensitivity to the noise.

- There is a trade-off between noise suppression, edge determination and localization.

- The linear operator that provides the best compromise between noise immunity and localization, is the first derivative of a Gaussian.