

Deep Learning for Computer Vision

Linear Filtering, Correlation and Convolution

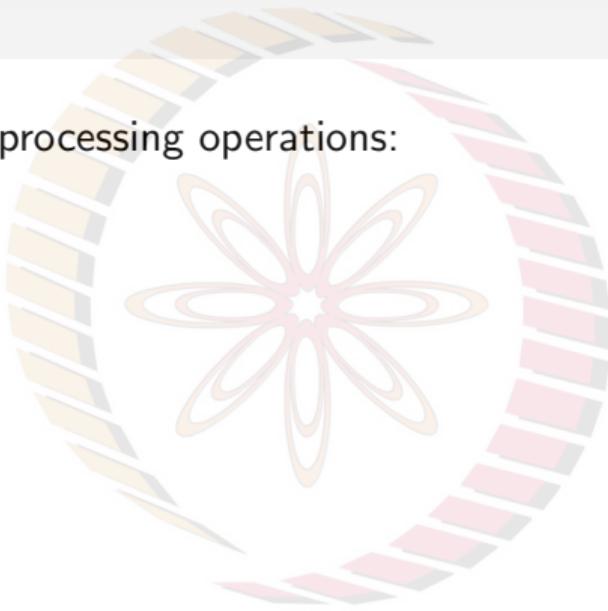
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad



Review

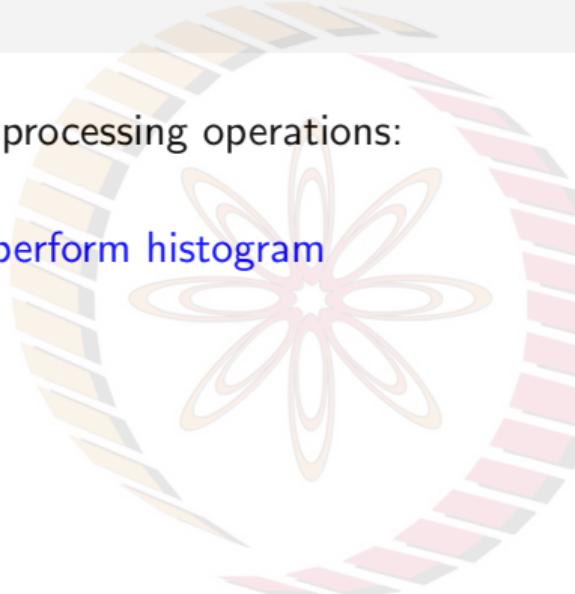
- Different types of image processing operations:
point, local and global



NPTEL

Review

- Different types of image processing operations:
point, local and global
- **Question:** How do you perform histogram equalization?

The logo consists of a circular emblem with a stylized flower in the center, surrounded by a ring of alternating light blue and white segments.

NPTEL

Review

- Different types of image processing operations:
point, local and global
- **Question:** How do you perform histogram equalization?
- Let I be the image with $M \times N$ pixels in total;
 I_{MAX} be the maximum image intensity value
(255); $h(I)$ be the image histogram

The NPTEL logo features the word "NPTEL" in a large, bold, sans-serif font. The letters are colored in a gradient from light orange at the top to light yellow at the bottom. Behind the text is a circular emblem consisting of two concentric rings. The inner ring is light orange and contains a stylized red flower with five petals. The outer ring is light yellow and has horizontal segments of varying shades of gray.

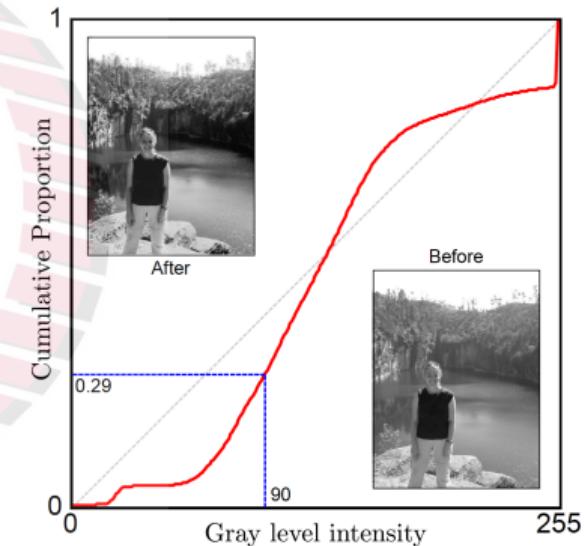
Review

- Different types of image processing operations:
point, local and global
- **Question:** How do you perform histogram equalization?
- Let I be the image with $M \times N$ pixels in total;
 I_{MAX} be the maximum image intensity value
(255); $h(I)$ be the image histogram
- Integrate $h(I)$ to obtain the cumulative distribution $c(I)$, whose each value
 $c_k = \frac{1}{M \times N} \sum_{l=1}^k h(l)$

The NPTEL logo consists of the word "NPTEL" in a large, bold, sans-serif font. The letters are colored in a gradient from light orange at the top to light yellow at the bottom. Behind the text is a circular emblem featuring a stylized flower or gear design with radiating lines in shades of orange, yellow, and red.

Review

- Different types of image processing operations: *point, local and global*
- **Question:** How do you perform histogram equalization?
- Let I be the image with $M \times N$ pixels in total; I_{MAX} be the maximum image intensity value (255); $h(I)$ be the image histogram
- Integrate $h(I)$ to obtain the cumulative distribution $c(I)$, whose each value $c_k = \frac{1}{M \times N} \sum_{l=1}^k h(l)$
- The transformed image $\hat{I}(i, j) = I_{MAX} \times c_{p_{ij}}$
- E.g., in figure, value 90 will be mapped to $I_{MAX} \times 0.29$ (rounded off)



Credit: Simon Prince, Computer Vision:
Models, Learning, and Inference,
Cambridge University Press

Image Filters: Linear Filter

- **Image Filter:** Modify image pixels based on some function of a local neighbourhood of each pixel

10	5	3
4	5	1
1	1	6

Some
function



What's the function?

NPTEL

Image Filters: Linear Filter

- **Image Filter:** Modify image pixels based on some function of a local neighbourhood of each pixel
- **Linear Filter:** Replace each pixel by linear combination (a weighted sum) of neighbours
- Linear combination called **kernel**, **mask** or **filter**

10	5	3
4	5	1
1	1	6

Some
function

What's the function?

NPTEL

10	5	3
4	5	1
1	1	6

Local
image data

0	0	0
0	0.5	0
0	1	0.5

Kernel

Modified
image data

Linear Filter: Cross-Correlation

Given a kernel of size $(2k + 1) \times (2k + 1)$:

- **Correlation** defined as:

$$G(i, j) = \underbrace{\frac{1}{(2k+1)^2}}_{\text{Uniform weight to each pixel}} \underbrace{\sum_{u=-k}^k \sum_{v=-k}^k I(i+u, j+v)}_{\text{Loop over pixels in considered neighbourhood around } I(i, j)}$$

NPTEL

Linear Filter: Cross-Correlation

Given a kernel of size $(2k + 1) \times (2k + 1)$:

- **Correlation** defined as:

$$G(i, j) = \underbrace{\frac{1}{(2k+1)^2}}_{\text{Uniform weight to each pixel}} \underbrace{\sum_{u=-k}^k \sum_{v=-k}^k I(i+u, j+v)}_{\text{Loop over pixels in considered neighbourhood around } I(i, j)}$$

- **Cross-correlation** defined as:

$$G(i, j) = \sum_{u=-k}^k \sum_{v=-k}^k \underbrace{H(u, v)}_{\text{Non-uniform weights}} I(i+u, j+v)$$

Linear Filter: Cross-Correlation

Given a kernel of size $(2k + 1) \times (2k + 1)$:

- **Correlation** defined as:

$$G(i, j) = \underbrace{\frac{1}{(2k+1)^2}}_{\text{Uniform weight to each pixel}} \sum_{u=-k}^k \sum_{v=-k}^k I(i+u, j+v)$$

Loop over pixels in considered neighbourhood around $I(i, j)$

- **Cross-correlation** defined as:

$$G(i, j) = \sum_{u=-k}^k \sum_{v=-k}^k \underbrace{H(u, v)}_{\text{Non-uniform weights}} I(i+u, j+v)$$

- Cross-correlation denoted by $G = H \otimes I$
- Can be viewed as “dot product” between local neighbourhood and kernel for each pixel
- Entries of kernel or mask $H(u, v)$ called **filter co-efficients**

Moving Average: Linear Filter

What values belong in the kernel H for the moving average example we saw earlier?

$$I(i, j)$$

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

$$\otimes \quad H(u, v)$$



$$= \quad G(i, j)$$

0	10	20	30	30
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Credit: K Grauman, Univ of Texas Austin

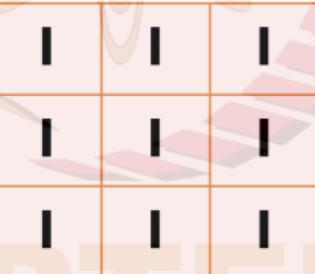
Moving Average: Linear Filter

What values belong in the kernel H for the moving average example we saw earlier?

$$I(i, j)$$

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

$$1/9$$



$$\otimes H(u, v)$$

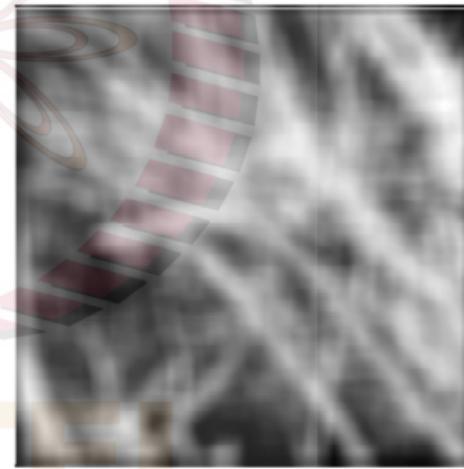
$$= G(i, j)$$

0	10	20	30	30
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Credit: K Grauman, Univ of Texas Austin

Moving Average Filter: Example

Effect of moving average filter (also known as **box filter**):



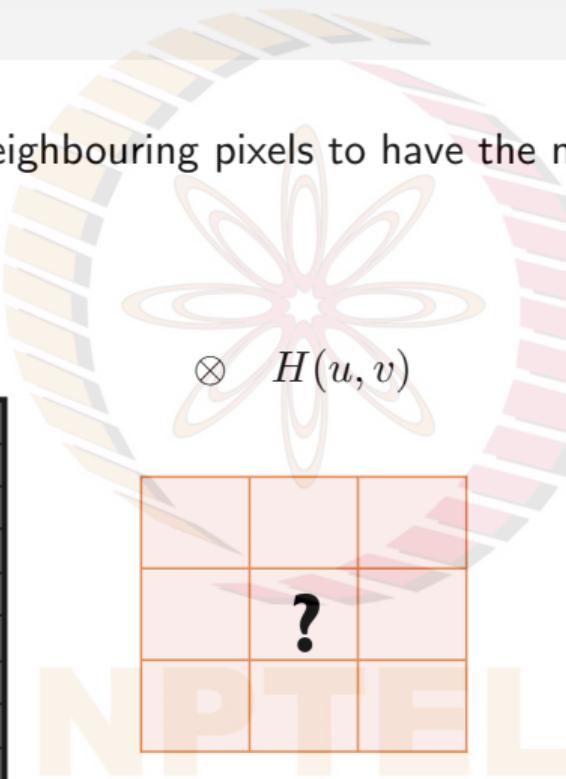
Credit: K Grauman, Univ of Texas Austin

Gaussian Average Filter

What if we want nearest neighbouring pixels to have the most influence on the output?

$$I(i, j)$$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0



Credit: K Grauman, Univ of Texas Austin

Gaussian Average Filter

What if we want nearest neighbouring pixels to have the most influence on the output?

$I(i, j)$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
0	0	0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
0	0	0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
0	0	0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
0	0	0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
0	0	0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
0	0	0	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

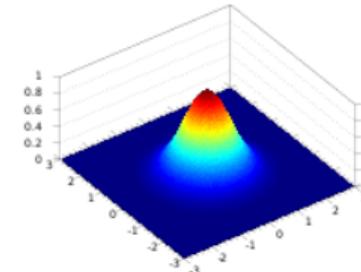
$\otimes H(u, v)$

$$\frac{1}{16}$$

1	2	1
2	4	2
1	2	1

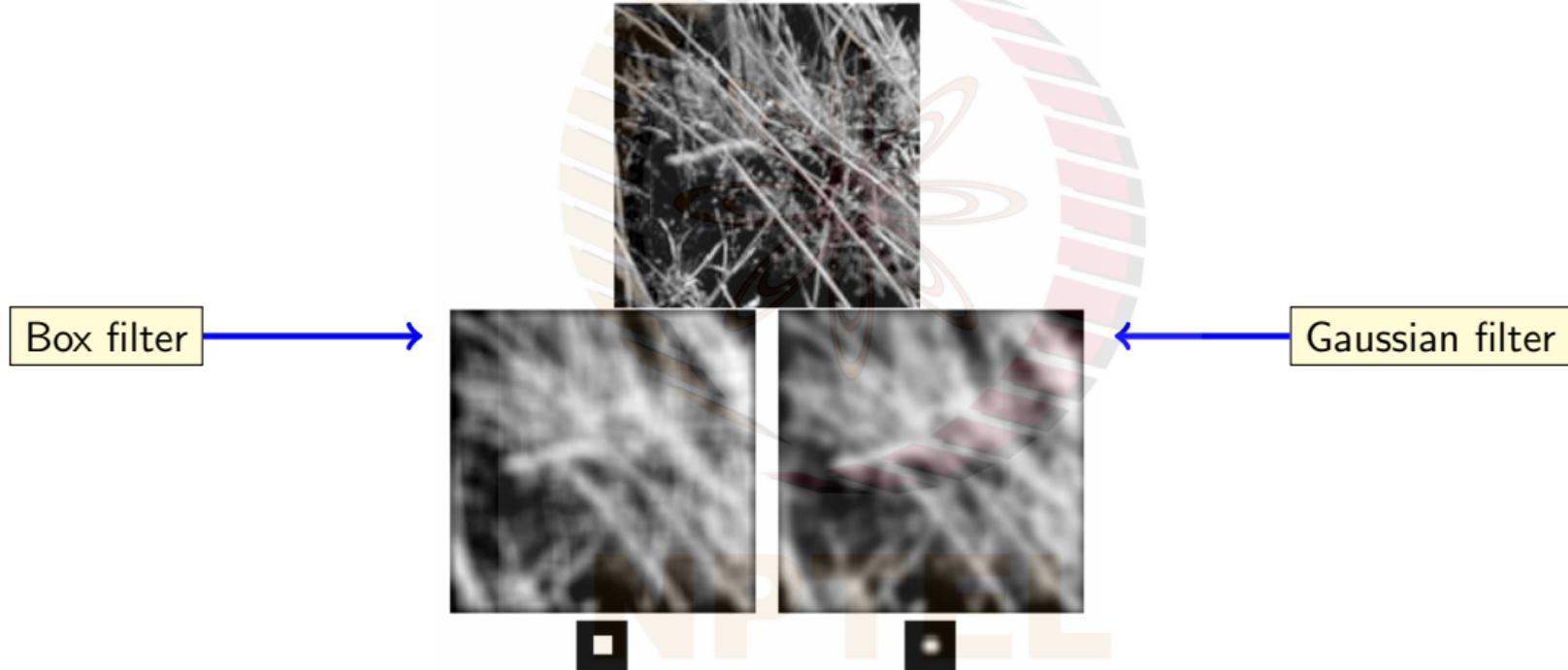
This kernel is an approximation of a 2D Gaussian function:

$$h(u, v) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{u^2+v^2}{\sigma^2}}$$



Credit: K Grauman, Univ of Texas Austin

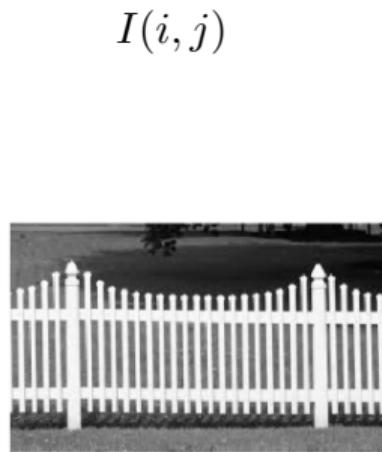
Averaging Filters: A Comparison



Credit: K Grauman, Univ of Texas Austin

Other Filters: The Edge Filter

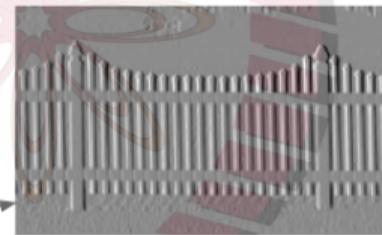
What should H look like to find the edges in a given image?



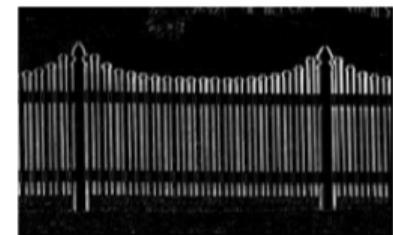
$$H(u, v)$$

$H(u, v)$ for
Vertical Edges?

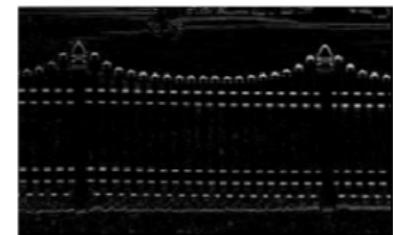
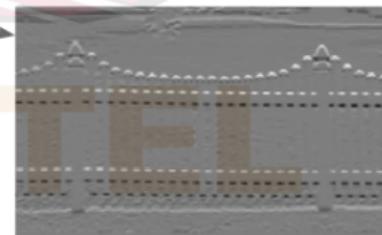
$$G(i, j)$$



$$|G(i, j)|$$

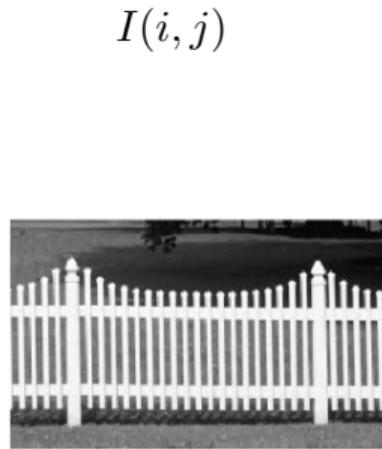


$H(u, v)$ for
Horizontal Edges?



Other Filters: The Edge Filter

What should H look like to find the edges in a given image?



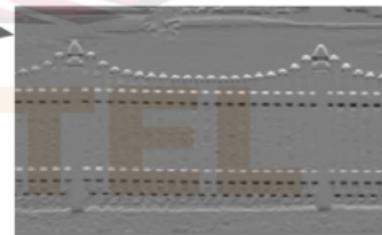
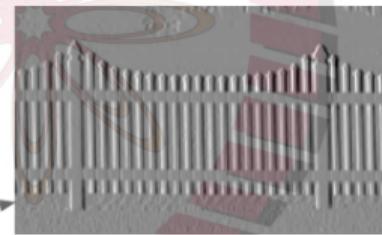
$$H(u, v)$$
$$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$$

1/9

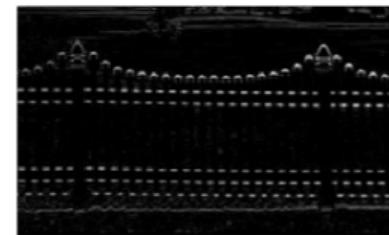
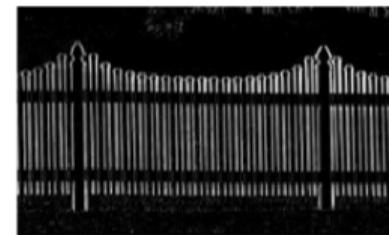
$$\begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$$

1/9

$$G(i, j)$$



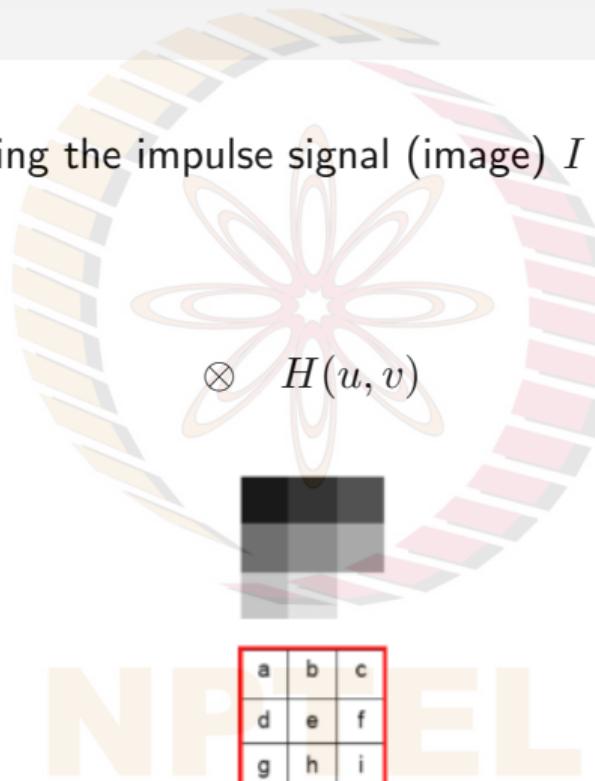
$$|G(i, j)|$$



Credit: KiwiWorker

Beyond Correlation

What is the result of filtering the impulse signal (image) I with the arbitrary kernel H ?



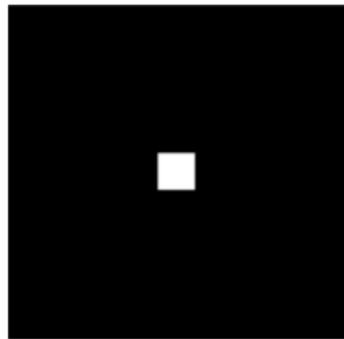
$$G(i, j)$$

?

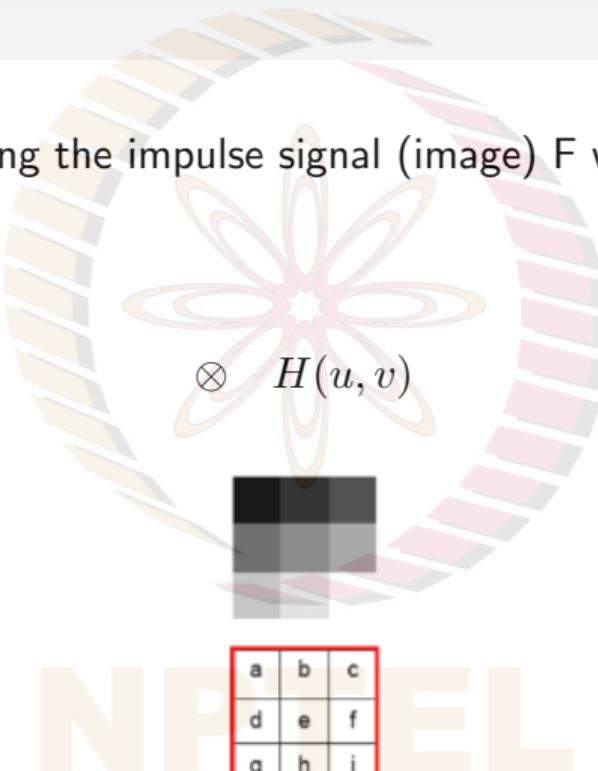
Beyond Correlation

What is the result of filtering the impulse signal (image) I with the arbitrary kernel H ?

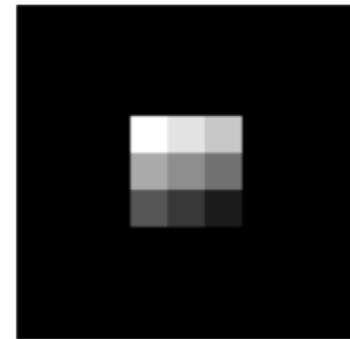
$$I(i, j)$$



$$\otimes H(u, v)$$



$$G(i, j)$$



!	q	b
j	e	p
c	g	a

Introducing Convolution

Given a kernel of size $(2k + 1) \times (2k + 1)$:

- **Convolution** defined as:

$$G(i, j) = \sum_{u=-k}^k \sum_{v=-k}^k H(u, v)I(i - u, j - v)$$

The NPTEL logo is a circular emblem. It features a stylized flower or mandala design in the center, composed of several overlapping, rounded, light-colored shapes. This central design is surrounded by two concentric rings of alternating colored rectangles. The top ring consists of light orange and grey rectangles, while the bottom ring consists of pink and grey rectangles. The entire logo is semi-transparent and centered on the slide.

NPTEL

Introducing Convolution

Given a kernel of size $(2k + 1) \times (2k + 1)$:

- **Convolution** defined as:

$$G(i, j) = \sum_{u=-k}^k \sum_{v=-k}^k H(u, v)I(i - u, j - v)$$

- Equivalent to flip the filter in both directions (bottom to top, right to left) and apply cross-correlation
- Denoted by $G = H * I$

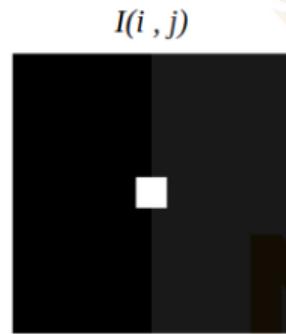
The NPTEL logo is a stylized, blocky text "NPTEL" in a light beige color. It is surrounded by a circular emblem composed of several concentric, overlapping curved bands in shades of pink, red, and orange.

Introducing Convolution

Given a kernel of size $(2k + 1) \times (2k + 1)$:

- **Convolution** defined as:

$$G(i, j) = \sum_{u=-k}^k \sum_{v=-k}^k H(u, v) I(i - u, j - v)$$



$* H(u, v)$

a	b	c
d	e	f
g	h	i

$\otimes H(u, v)$

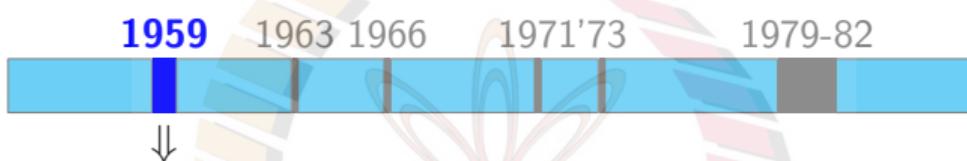
!	4	6
j	e	p
o	q	s

- Equivalent to flip the filter in both directions (bottom to top, right to left) and apply cross-correlation
- Denoted by $G = H * I$

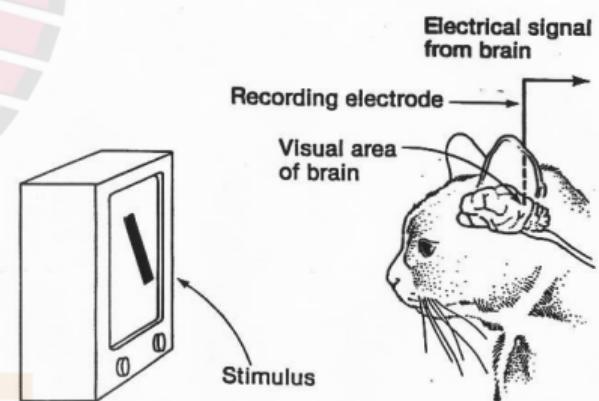


a	b	c
d	e	f
g	h	i

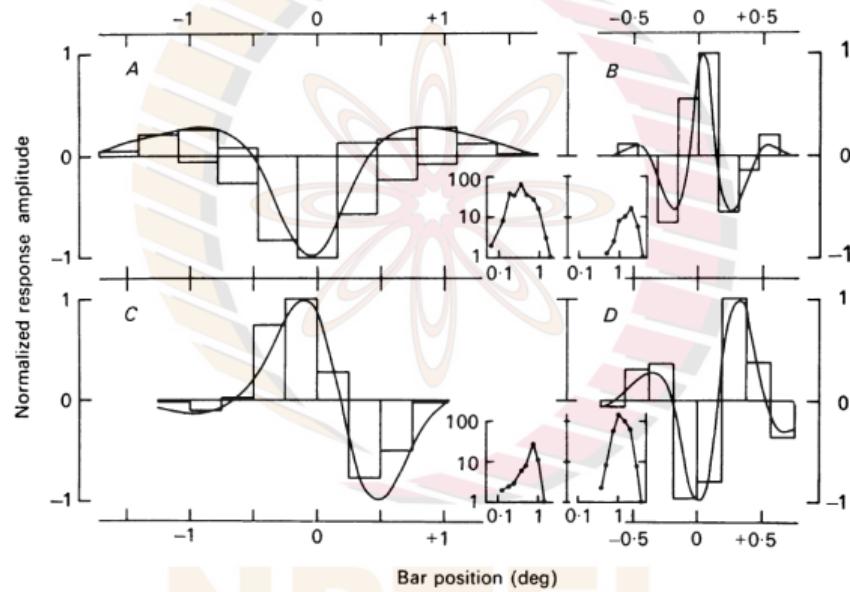
Recall: Early History



- David Hubel and Torsten Wiesel publish their work “*Receptive fields of single neurons in the cat’s striate cortex*”
- Placed electrodes into primary visual cortex area of an anesthetized cat’s brain
- Showed that simple and complex neurons exist, and that visual processing starts with simple structures such as oriented edges



Linear Summation in the Visual Cortex



Simple cells perform linear spatial summation over their receptive fields¹

¹ Movshon, Thompson and Tolhurst, *Spatial Summation in the Receptive Fields of Simple Cells in the Cat's Striate Cortex*, JP 1978

Linear Shift-Invariant Operators

- Both correlation and convolution are **Linear Shift-Invariant operators**, which obey:
 - Linearity (or Superposition principle):**
$$I \circ (h_0 + h_1) = I \circ h_0 + I \circ h_1$$
 - Shift-Invariance:** shifting (or translating) a signal commutes with applying the operator
$$g(i, j) = h(i + k, j + l) \iff (f \circ g)(i, j) = (f \circ h)(i + k, j + l)$$
- Equivalent to saying that the effect of the operator is the same everywhere. Why do we need this in computer vision?

NPTEL

Source: Raquel Urtasun, Univ of Toronto

Linear Shift-Invariant Operators

- Both correlation and convolution are **Linear Shift-Invariant operators**, which obey:
 - Linearity (or Superposition principle):**
$$I \circ (h_0 + h_1) = I \circ h_0 + I \circ h_1$$
 - Shift-Invariance:** shifting (or translating) a signal commutes with applying the operator
$$g(i, j) = h(i + k, j + l) \iff (f \circ g)(i, j) = (f \circ h)(i + k, j + l)$$
- Equivalent to saying that the effect of the operator is the same everywhere. Why do we need this in computer vision?

NPTEL

Source: Raquel Urtasun, Univ of Toronto

Properties of Convolution

- **Commutative:** $a * b = b * a$
 - Conceptually no difference between filter and signal
- **Associative:** $a * (b * c) = (a * b) * c$
 - We often apply filters one after the other: $((a * b1) * b2) * b3$
 - This is equivalent to applying one cumulative filter: $a * (b1 * b2 * b3)$
- **Distributive over addition:** $a * (b + c) = (a * b) + (a * c)$
 - We can combine the responses of a signal over two or more filters by combining the filters
- **Scalars factor out:** $ka * b = a * kb = k(a * b)$
- **Identity:** Unit impulse $e = [..., 0, 0, 1, 0, 0, ...]$, $a * e = a$

Properties of Convolution

- **Commutative:** $a * b = b * a$
 - Conceptually no difference between filter and signal
- **Associative:** $a * (b * c) = (a * b) * c$
 - We often apply filters one after the other: $((a * b1) * b2) * b3$
 - This is equivalent to applying one cumulative filter: $a * (b1 * b2 * b3)$
- **Distributive over addition:** $a * (b + c) = (a * b) + (a * c)$
 - We can combine the responses of a signal over two or more filters by combining the filters
- **Scalars factor out:** $ka * b = a * kb = k(a * b)$
- **Identity:** Unit impulse $e = [..., 0, 0, 1, 0, 0, ...]$, $a * e = a$

Properties of Convolution

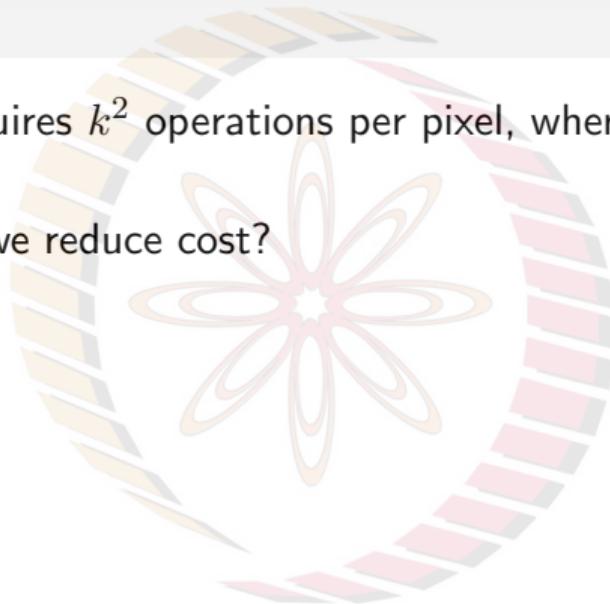
- **Commutative:** $a * b = b * a$
 - Conceptually no difference between filter and signal
- **Associative:** $a * (b * c) = (a * b) * c$
 - We often apply filters one after the other: $((a * b1) * b2) * b3$
 - This is equivalent to applying one cumulative filter: $a * (b1 * b2 * b3)$
- **Distributive over addition:** $a * (b + c) = (a * b) + (a * c)$
 - We can combine the responses of a signal over two or more filters by combining the filters
- **Scalars factor out:** $ka * b = a * kb = k(a * b)$
- **Identity:** Unit impulse $e = [..., 0, 0, 1, 0, 0, ...]$, $a * e = a$

Properties of Convolution

- **Commutative:** $a * b = b * a$
 - Conceptually no difference between filter and signal
- **Associative:** $a * (b * c) = (a * b) * c$
 - We often apply filters one after the other: $((a * b1) * b2) * b3$
 - This is equivalent to applying one cumulative filter: $a * (b1 * b2 * b3)$
- **Distributive over addition:** $a * (b + c) = (a * b) + (a * c)$
 - We can combine the responses of a signal over two or more filters by combining the filters
- **Scalars factor out:** $ka * b = a * kb = k(a * b)$
- **Identity:** Unit impulse $e = [..., 0, 0, 1, 0, 0, ...]$, $a * e = a$

Separability

- Convolution operator requires k^2 operations per pixel, where k is the width (and height) of a convolution kernel.
- Can be costly. How can we reduce cost?



NPTEL

Separability

- Convolution operator requires k^2 operations per pixel, where k is the width (and height) of a convolution kernel.
- Can be costly. How can we reduce cost?
- For certain filters, can be sped up by performing a 1D horizontal convolution followed by a 1D vertical convolution, requiring $2k$ operations \implies convolution kernel is **separable**.
-

$$K = vh^T$$

where v, h are 1D kernels, and K is the 2D kernel

NPTEL

Separability

- Convolution operator requires k^2 operations per pixel, where k is the width (and height) of a convolution kernel.
- Can be costly. How can we reduce cost?
- For certain filters, can be sped up by performing a 1D horizontal convolution followed by a 1D vertical convolution, requiring $2k$ operations \implies convolution kernel is **separable**.
-

$$K = vh^T$$

where v, h are 1D kernels, and K is the 2D kernel

Example 1:

1	2	1
2	4	2
1	2	1

NPTEL

Separability

- Convolution operator requires k^2 operations per pixel, where k is the width (and height) of a convolution kernel.
- Can be costly. How can we reduce cost?
- For certain filters, can be sped up by performing a 1D horizontal convolution followed by a 1D vertical convolution, requiring $2k$ operations \Rightarrow convolution kernel is **separable**.
-

$$K = vh^T$$

where v, h are 1D kernels, and K is the 2D kernel

Example 1:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \implies v = h = \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Separability

- Convolution operator requires k^2 operations per pixel, where k is the width (and height) of a convolution kernel.
- Can be costly. How can we reduce cost?
- For certain filters, can be sped up by performing a 1D horizontal convolution followed by a 1D vertical convolution, requiring $2k$ operations \Rightarrow convolution kernel is **separable**.
-

$$K = vh^T$$

where v, h are 1D kernels, and K is the 2D kernel

Example 1:

1	2	1
2	4	2
1	2	1

$$\Rightarrow v = h = \frac{1}{4}$$

1
2
1

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

Example 2:

Separability

- Convolution operator requires k^2 operations per pixel, where k is the width (and height) of a convolution kernel.
- Can be costly. How can we reduce cost?
- For certain filters, can be sped up by performing a 1D horizontal convolution followed by a 1D vertical convolution, requiring $2k$ operations \Rightarrow convolution kernel is **separable**.
-

$$K = vh^T$$

where v, h are 1D kernels, and K is the 2D kernel

Example 1:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \implies v = h = \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Example 2:

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \implies v = \frac{1}{4} \begin{bmatrix} -1 \\ -2 \\ -1 \end{bmatrix} \text{ & } h = \frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Separable Convolution

How can we tell if a given kernel K is separable?



NPTEL

Separable Convolution

How can we tell if a given kernel K is separable?

- Visual inspection



NPTEL

Separable Convolution

How can we tell if a given kernel K is separable?

- Visual inspection
- Analytically, look at the Singular Value Decomposition (SVD), and if only one singular value is non-zero, then it is separable.

$$K = U\Sigma V^T = \sum_i \sigma_i u_i v_i^T$$

where $\Sigma = \text{diag}(\sigma_i)$

$\sqrt{\sigma_1}u_1$ and $\sqrt{\sigma_1}v_1$ are the vertical and horizontal kernels

Source: Raquel Urtasun, Univ of Toronto

Practical Issues

Ideal size for the filter?



NPTEL

Practical Issues

Ideal size for the filter?

The bigger the mask:

- more neighbours contribute
- smaller noise variance of output
- bigger noise spread
- more blurring
- more expensive to compute

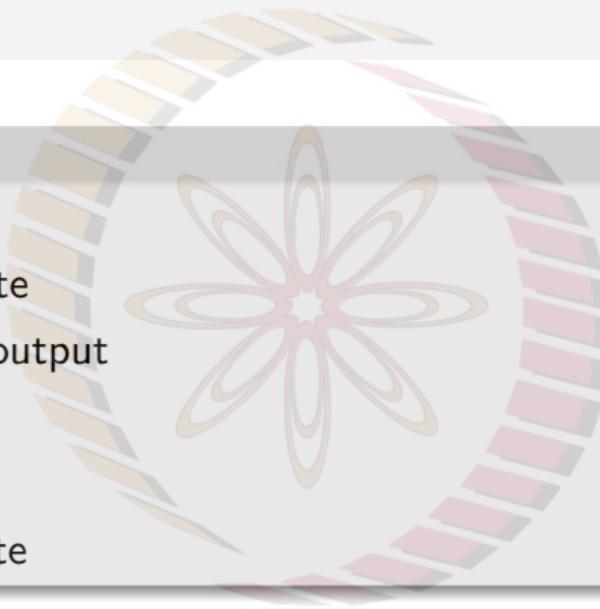
The NPTEL logo consists of the word "NPTEL" in a large, bold, sans-serif font. The letters are a light beige color. Behind the letters is a circular emblem. The emblem features a stylized flower or mandala design in the center, composed of several overlapping ovals in shades of pink, red, and orange. This central design is surrounded by a ring of alternating colored rectangles in shades of pink, red, and orange, which in turn is surrounded by a ring of diagonal stripes in the same color palette.

Practical Issues

Ideal size for the filter?

The bigger the mask:

- more neighbours contribute
- smaller noise variance of output
- bigger noise spread
- more blurring
- more expensive to compute



What about the boundaries? Do we lose information?

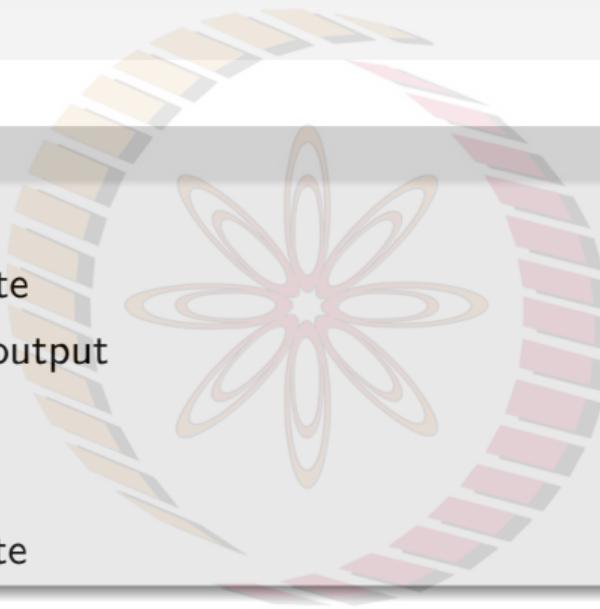
NPTEL

Practical Issues

Ideal size for the filter?

The bigger the mask:

- more neighbours contribute
- smaller noise variance of output
- bigger noise spread
- more blurring
- more expensive to compute

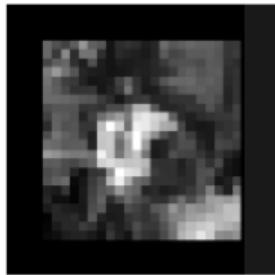


What about the boundaries? Do we lose information?

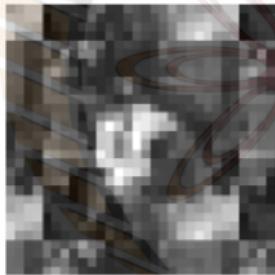
- Without padding, we lose out on information at the boundaries.
- We can use a variety of strategies such as zero padding, wrapping around, copy the edge

Practical Issues

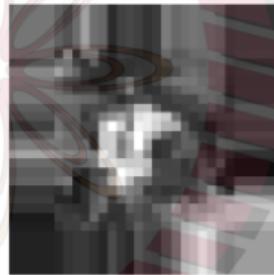
Different padding strategies:



zero



wrap



clamp



mirror



blurred zero



normalized zero



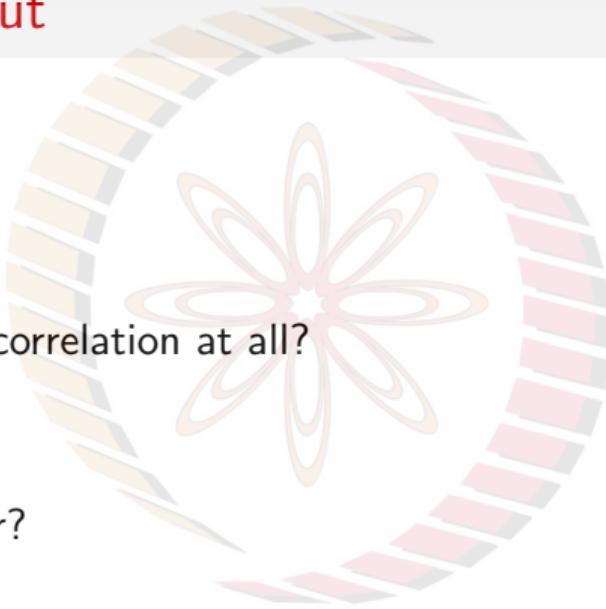
blurred clamp



blurred mirror

Questions to Think About

- Do we then need (cross)-correlation at all?
- Are all filters always linear?

A circular watermark logo for NPTEL. It features a stylized flower or star shape in the center, composed of several curved, overlapping lines. This central design is surrounded by two concentric rings. The inner ring consists of alternating light blue and white segments. The outer ring consists of alternating light blue and grey segments.

NPTEL

Homework

Readings

- Chapter 3 (§3.1-3.3), Szeliski, *Computer Vision: Algorithms and Applications*, 2010 draft
- Chapter 7 (§7.1-7.2), Forsyth and Ponce, *Computer Vision: A Modern Approach*, 2003 edition

The NPTEL logo consists of the letters "NPTEL" in a large, bold, sans-serif font. The letters are colored in a light beige or cream shade. Behind the letters is a circular emblem featuring a stylized flower design with multiple petals in shades of pink and orange.