UNIVERSITY OF HULL

Filters & Convolution

Computer Vision

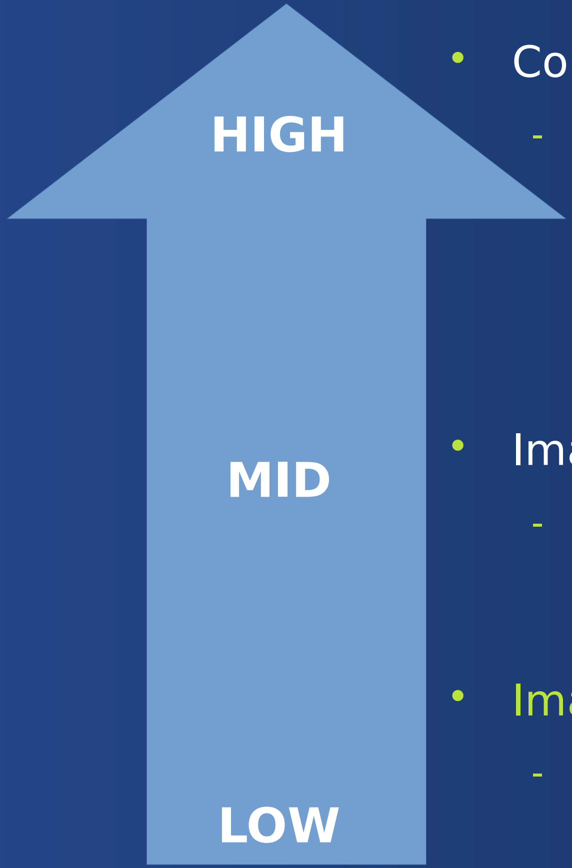
Module Code: 600100

Department of Computer Science and Technology

Today

- Limitations of Point Operations
- From Point to Spatial
- Convolution Operation
  - Have you heard of Convolutional Neural Networks?
  - Padding and Stride

Overview of Computer Vision



- Computer Vision
  - Detection/ Recognition
- Image Analysis
  - Segmentation, etc
- Image Processing
  - Enhancements, Feature Extraction

- High Level – Collection of information in; decision processes
- Mid Level – Enhanced images in; Extracted information out.
- Low Level – Processing of input image into an ‘enhanced’ state.

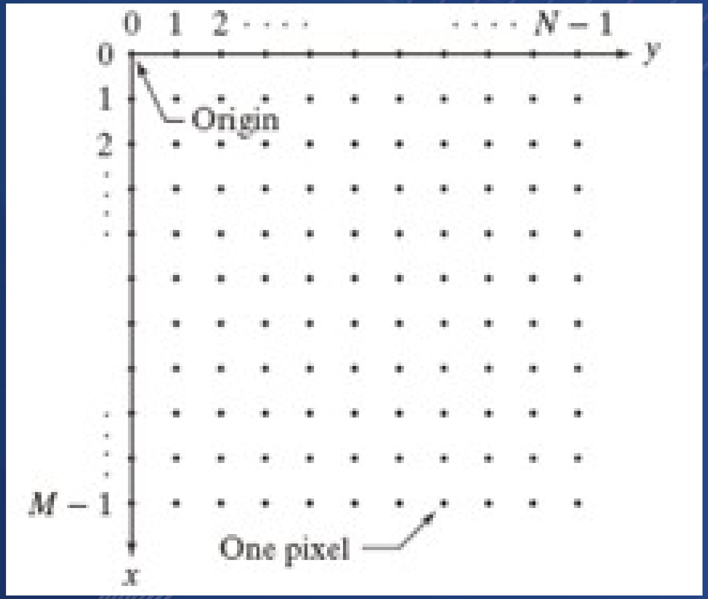
Recap: Image Representation

- Image as a function
  - Remember! we sample a real-world object, based on our sensor arrangement.

$$f(x,y) \quad \begin{array}{l} x \in \{0,1,2,\ldots,M-1\} \\ y \in \{0,1,2,\ldots,N-1\} \end{array}$$

{} SET

MEMBER OF



  - Intensity value at coords (x,y)
  - A Matrix of pixels / pixel elements

Recap: Point Operations

- We can apply arithmetic operations on our image function

$$s(x,y) = f(x,y) + g(x,y)$$

$$d(x,y) = f(x,y) - g(x,y)$$

$$p(x,y) = f(x,y) \times g(x,y)$$

$$v(x,y) = f(x,y) \div g(x,y)$$

<u>For all (x,y) pairs!</u>

Operations are performed 'Array-like' in a pixel-wise manner

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} = \begin{bmatrix} 6 & 7 \\ 8 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} = \begin{bmatrix} 5 & 10 \\ 15 & 20 \end{bmatrix}$$

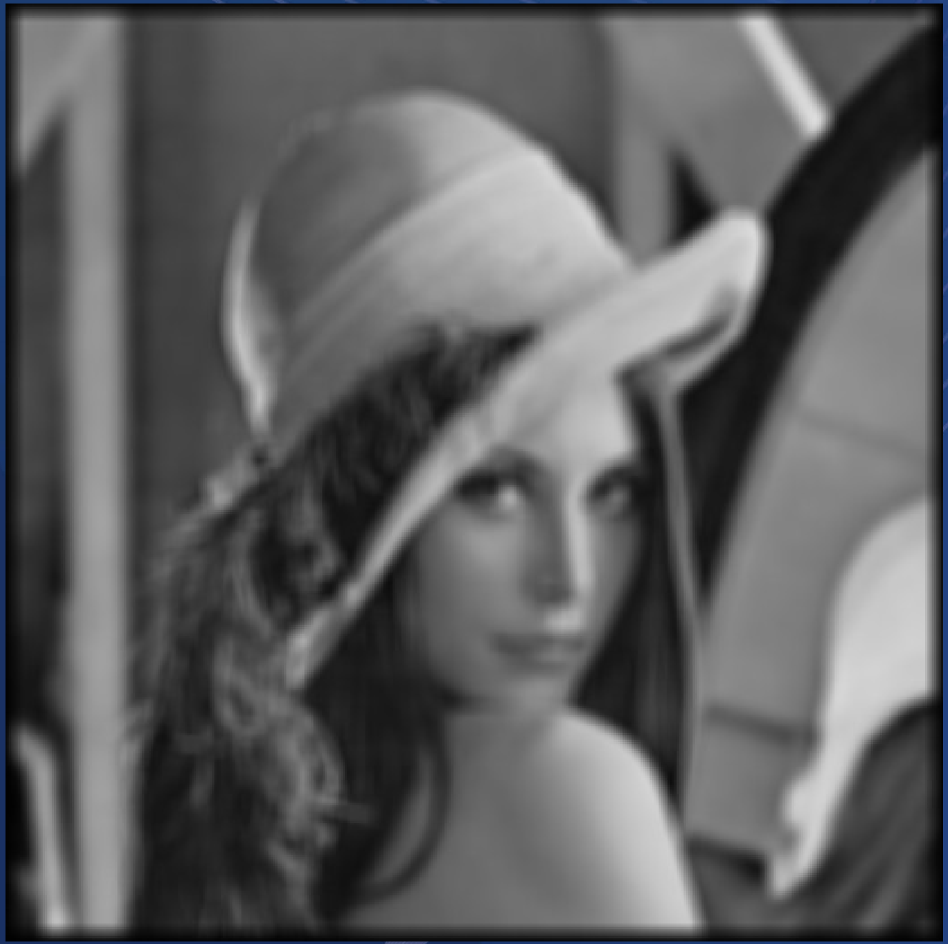
Filtering in the Spatial Domain

Point Operations → Spatial Operations

- Q: What are the limitations of Point Operations?
- A:
  - We can only change each value individually
    - No real context used.
  - We have no information about surrounding points
- Most useful information from images requires more than just a singular value.
  - E.g Edges, Texture, Shapes.
- Therefore, we need to move to the spatial domain to get that information!

Example Limitation

- What can't Point Operations do? Blurring



Example Limitation

- What can’t Point Operations do? Sharpening

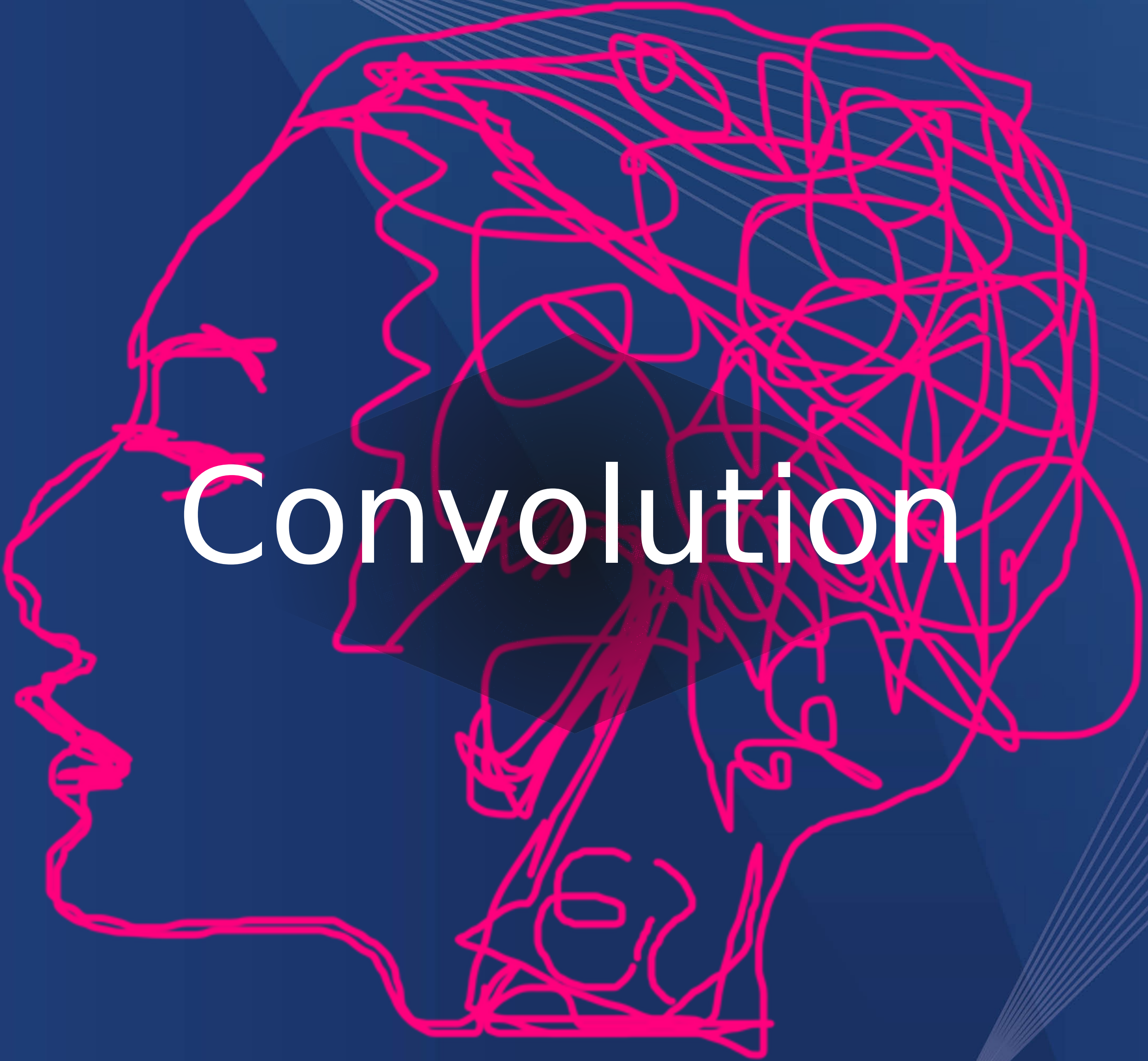
Point Operations → Spatial Operations

- In a similar way to how we can use histograms to provide context, the pixel neighbourhood of a point can tell us a lot of information.
- Q: How could context around a pixel be useful to us, as Computer Vision Detectives?
- Example:
  - Are we a low value in this neighbourhood?
  - How does this intensity value change?
    - Not a lot (low variance), a lot (high variance), not at all?!?
  - Is this region of the image ‘smooth’?

Spatial Filters

- To understand the neighbourhood, we need filters to tell us whether something is ‘important’.
  - A filter is something which passes/modifies/rejects certain inputs (based on their value)
  - Borrowed from the signal processing domain
  - Consists of a separately defined matrix/array from the input image.
- A Spatial Filter makes a decision on a single pixel, based on the neighbourhood.

Convolution



UNIVERSITY OF HULL

Convolution

- Related to Signal Processing
- Comprised of two components:
  - 1) A Filter (Often called a Kernel or Convolution Mask)
  - 2) Neighbourhood around a Point – x, y.
- Defined as the “Integral of the product of two functions”
  - I.e The input image, and your filter.

$$g(x,y) = \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} w(i,j) f(x+i, y+j)$$

Example: 3x3 neighbourhood

UNIVERSITY OF HULL

1) The Filter

$$w(i,j) = \begin{bmatrix} w_{\frac{-m}{2}\frac{-n}{2}} & \cdots & w_{\frac{-m}{2}\frac{+n}{2}} \\ \vdots & 0 & \vdots \\ w_{\frac{+m}{2}\frac{-n}{2}} & \cdots & w_{\frac{+m}{2}\frac{+n}{2}} \end{bmatrix}$$

All relative to the center

$$= \begin{bmatrix} w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-1} & w_{1,0} & w_{1,1} \end{bmatrix}$$

What is that maths?

w(i,j) = 3x3 kernel in this example.

=> -1 → 0 → 1 for both axes

Our filter coefficient at i,j

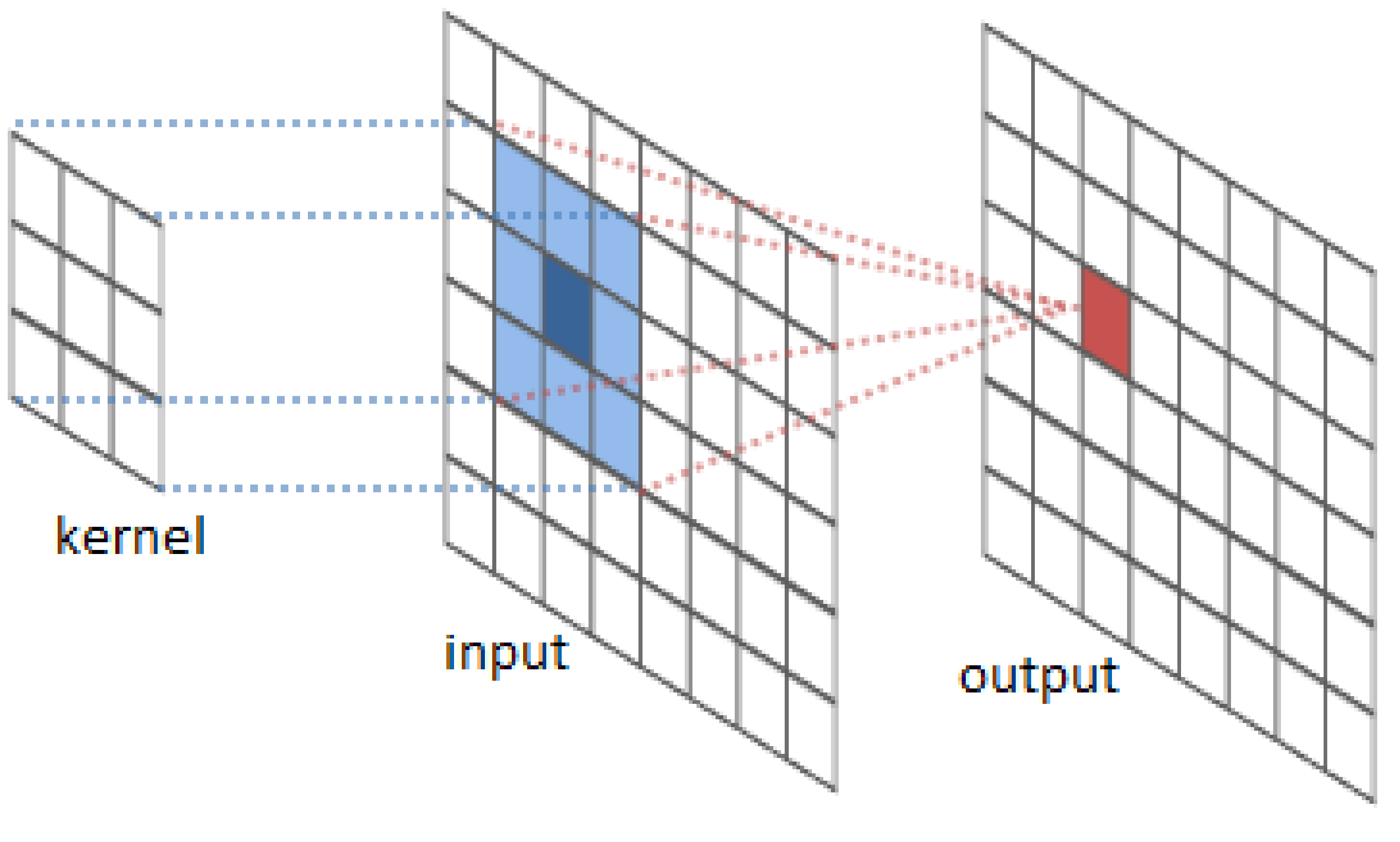
$$g(x,y) = \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} \underline{w(i,j)}\, \underline{f(x+i, y+j)}$$

Our 'Convolved' Image

Double Sum
(Double for loop)

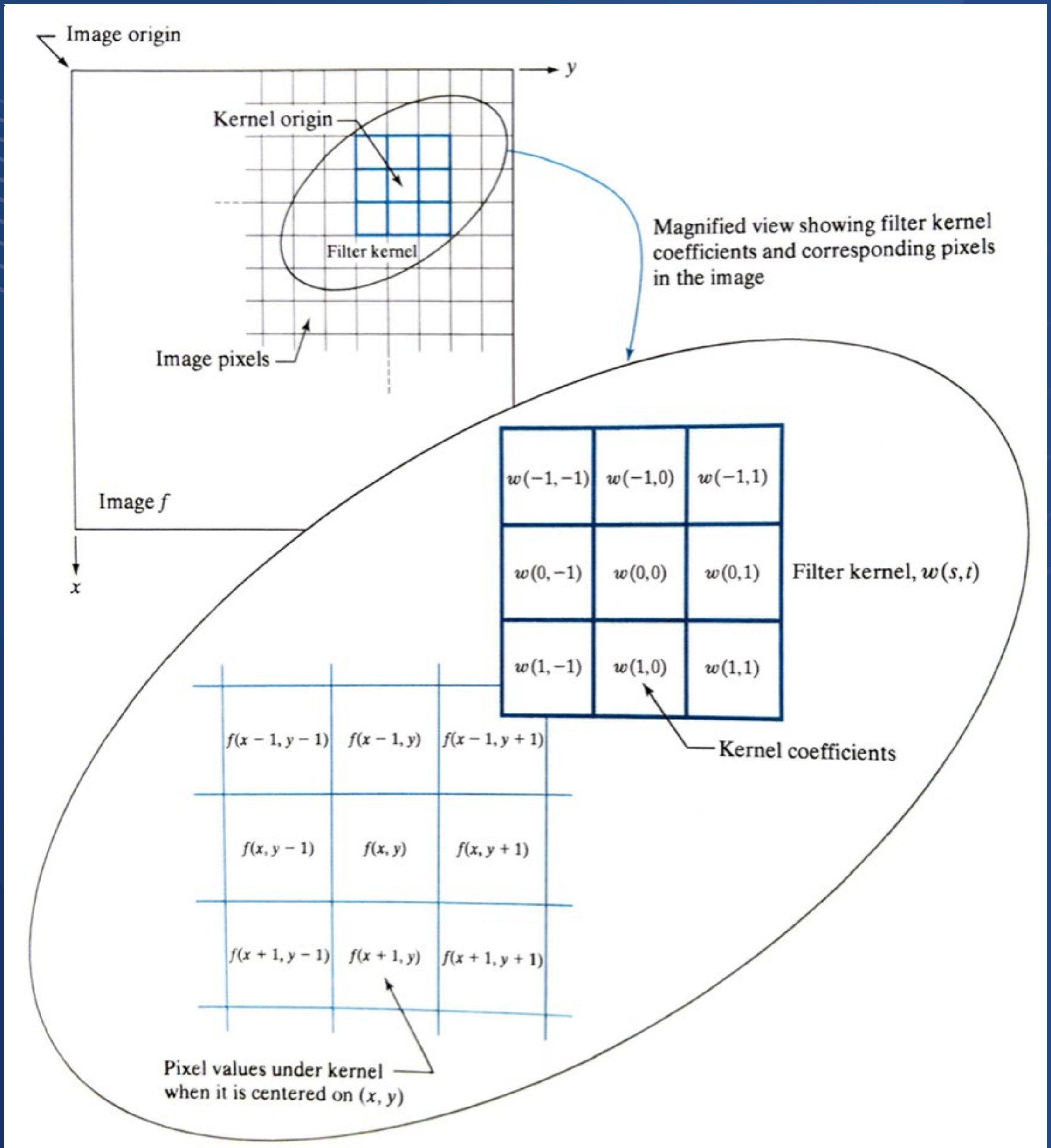
Grey Level f(x,y)
Getting neighbourhood via offsets, i, j.

Visual Aid



Today

- Slide the filter (Kernel) over all x, y pairs in the input image
  - For each x,y in f(x,y).
- Convolve the input image neighbourhood, with the filter
  - Filter is centered around a given x,y
- Generates a new output image, g(x,y)
  - Considered the weighted sum of the neighbourhood.



Convolution Example 1

$$f = \begin{bmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{bmatrix} \quad w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$f \star w = ?$$

f convolved with w

UNIVERSITY OF HULL

Convolution Example 1

$$f = \begin{bmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{bmatrix} \quad w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} &(0 \times 2) + (0 \times 7) + (0 \times 6) \qquad 0+0+0 \\ = &+(0 \times 9) + (1 \times 5) + (0 \times 1) = +0+5+0 = 5 \\ &+(0 \times 4) + (0 \times 3) + (0 \times 8) \qquad +0+0+0 \end{aligned}$$

Convolution Example 2

$$f = \begin{bmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{bmatrix} \quad w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$f \star w \;=?$$

Convolution Example 2

$$f = \begin{bmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{bmatrix} \quad w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} & (0\times 2)+(0\times 7)+(0\times 6) & & 0+0+0 & \\ = & +(0\times 9)+(0\times 5)+(1\times 1) & = & +0+0+1 & = 1 \\ & +(0\times 4)+(0\times 3)+(0\times 8) & & +0+0+0 & \end{aligned}$$

Convolution Example 3

$$f = \begin{bmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{bmatrix} \quad w = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix}$$

$$f \star w \quad = ?$$

Convolution Example 3

$$f = \begin{bmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{bmatrix} \; w = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix}$$

$$= \begin{matrix} (\frac{1}{4} \times 2) + (0 \times 7) + (\frac{1}{4} \times 6) \\ + (0 \times 9) + (0 \times 5) + (0 \times 1) \\ + (\frac{1}{4} \times 4) + (0 \times 3) + (\frac{1}{4} \times 8) \end{matrix} = \begin{matrix} \frac{1}{2} + 0 + \frac{3}{2} \\ +0+0+0 \\ +1+0+2 \end{matrix} = \begin{matrix} +2 \\ +0 \\ +3 \\ \hline 5 \end{matrix}$$

UNIVERSITY OF HULL

Boundary Conditions

- Convolving 3x3 neighbourhood with a 3x3 filter
  - Produces a 1x1 output.

- How can we apply this to a whole image?
  - Sliding the filter through the original input image.
  - Filter does ‘fit’ perfectly into the very corner pixels
    - Filter would ‘hang off’ the page…

- Solution: Padding!

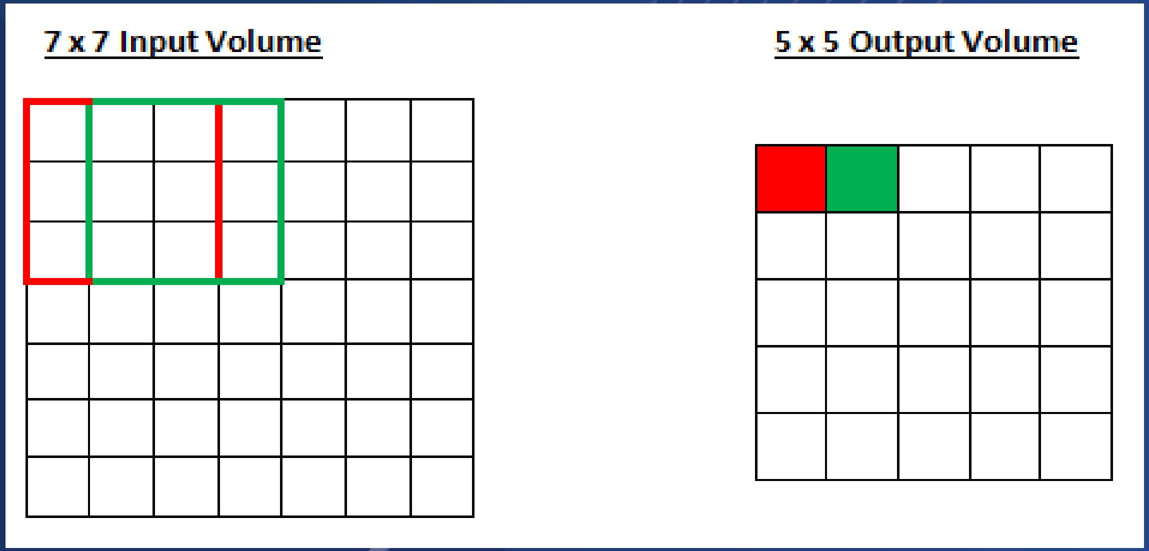
UNIVERSITY
OF HULL

Padding

No Padding

- Simplest form of padding is No Padding
- If we can't center the filter on all our input x,y pairs, that's okay.
  - → We will put the filter as close as possible for boundaries.
- Results in an overall decrease in image size
  - E.g 7x7 → 5x5 after Convolution.

We lose ceil(Filter Width / 2)
E.g 3x3 filter → ceil(3/2) = ceil(1.5) = 2
Total loss of 2.
Per-axis, as it is a square matrix.

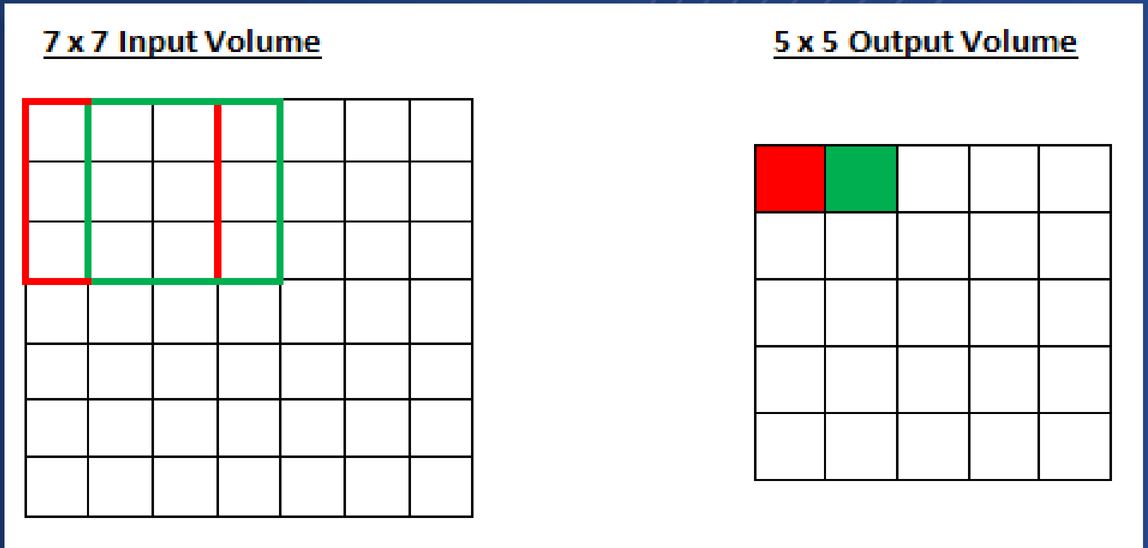


‘Same’ Padding (sometimes called full / zero)

- Provides the ‘same’ output resolution, as input. (full padding in MATLAB)
- Pads the original image with 0 around all sides.
  - Amount of zeros depends on how big our kernel is.
- Typically denoted by:

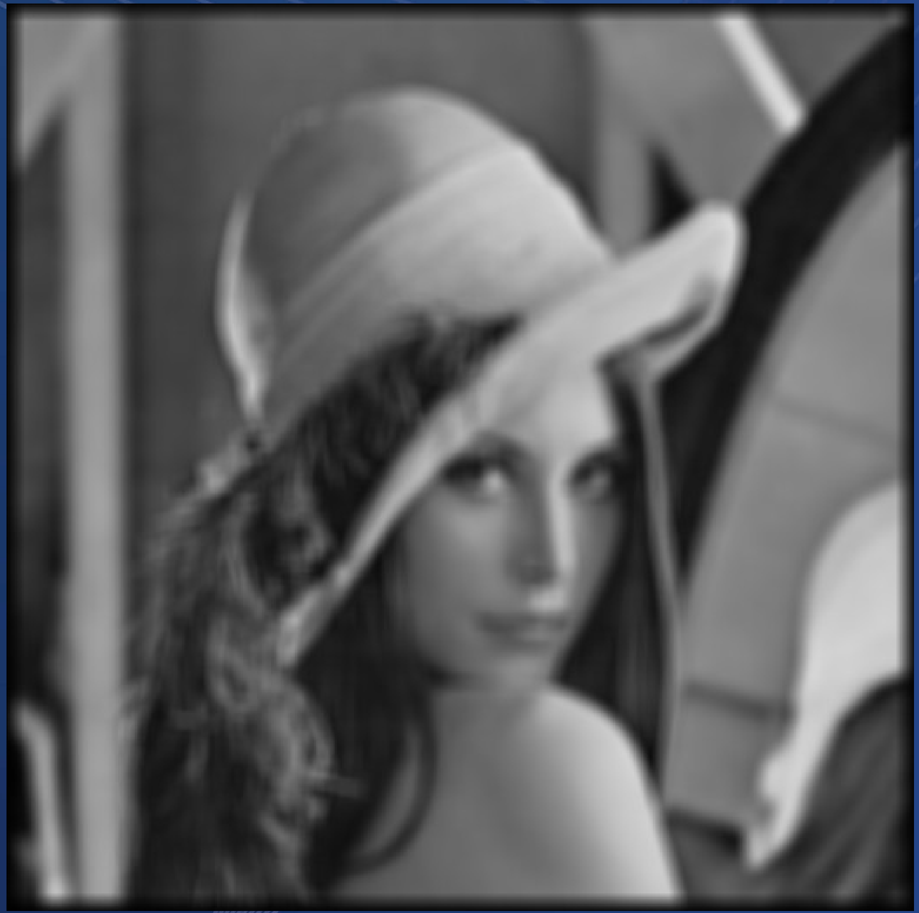
$$Padding = \frac{FilterSize - 1}{2}$$

- Padding = (3 – 1) / 2 = 1 pixel padding
- This is intuitive.



Padding Artefacts

- Padding edges with zeros adds a dark ‘border’
- Depending on kernel size, this may have an impact on the output result.
- If kernel weights are non-central:
  - Lots of those zeros will be used in the full convolution calculation
  - → Massively skewed values.
- E.g Zero Padding causes Black Borders to appear in our ‘blur’ example.



Zero Padding

$$f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 7 & 6 & 0 \\ 0 & 9 & 5 & 1 & 0 \\ 0 & 4 & 3 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$f \star w = \boxed{\begin{matrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{matrix}}$$

UNIVERSITY OF HULL

Stride

- Filters so far have moved one pixel at a time
  - E.g Stride = 1.
- Stride represents how far to move the filter each time.
- Higher Stride → Smaller Output (Think sampling)
- Q: Why do this?
- A: Quicker, Cheaper, Less Memory Usage
- A: We might not need to do it on every pixel, depends on our pixel resolution, and how detailed our subject is.

Zero Padding w/ Stride = 2

$$f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 7 & 6 & 0 \\ 0 & 9 & 5 & 1 & 0 \\ 0 & 4 & 3 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$f \star w = \begin{bmatrix} 2 & 6 \\ 4 & 8 \end{bmatrix}$$

UNIVERSITY
OF HULL

UNIVERSITY OF HULL

Questions?

(now, after, or e-mail)