Fast 2D Convolution Algorithms summary

Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 3.1

Artificial Intelligence &
Information Analysis Lab

Outline

- 2D linear systems
- 2D convolutions
    - Discrete-time 2D Systems
    - Linear & Cyclic 2D convolutions
    - 2D Discrete Fourier Transform, 2D Fast Fourier Transform
- Other convolution algorithms
    - Winograd algorithm
    - Block methods
- Applications in Machine Learning
    - Convolutional neural networks

Convolution and correlation

2D convolution applications:

- Machine Learning (Convolutional neural networks)
- Image processing

2D correlation applications:

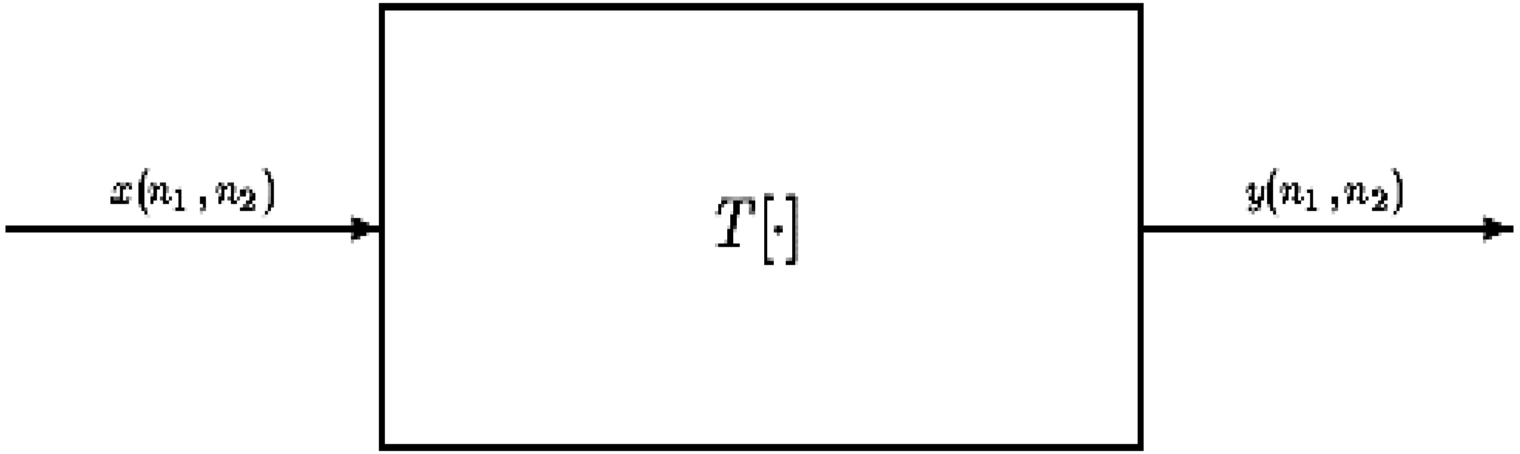
- Feature matching
- Template matching
- Object detection and tracking

2D Discrete Systems

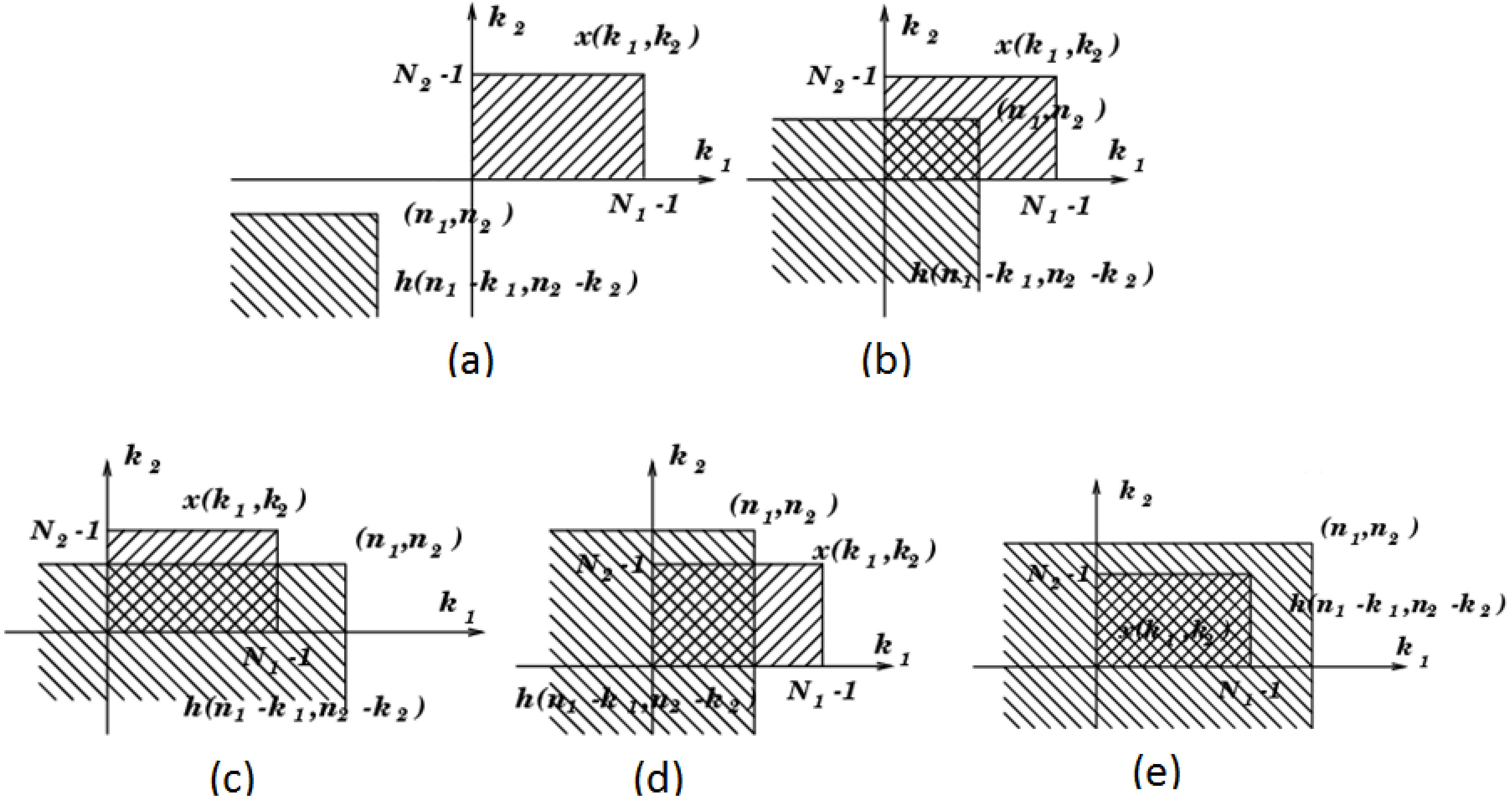
2D system:

- It ransforms a 2D discrete input signal $x(n_1, n_2)$ into a 2D discrete-time output signal $y(n_1, n_2)$:

$$y(n_1, n_2) = T[x(n_1, n_2)].$$



2D Discrete Systems



Visualization of 2D convolution calculation.

Artificial Intelligence & Information Analysis Lab

2D Discrete Systems

- Finite impulse response (FIR):

  $h(n_1, n_2)$ is zero outside some filter mask (region) $M_1 \times M_2$,

  $0 \leq n_1 < M_1, 0 \leq n_2 < M_2$.

- FIR filters are described by a 2D linear convolution with convolutional kernel h of size $M_1 \times M_2$ is given by:

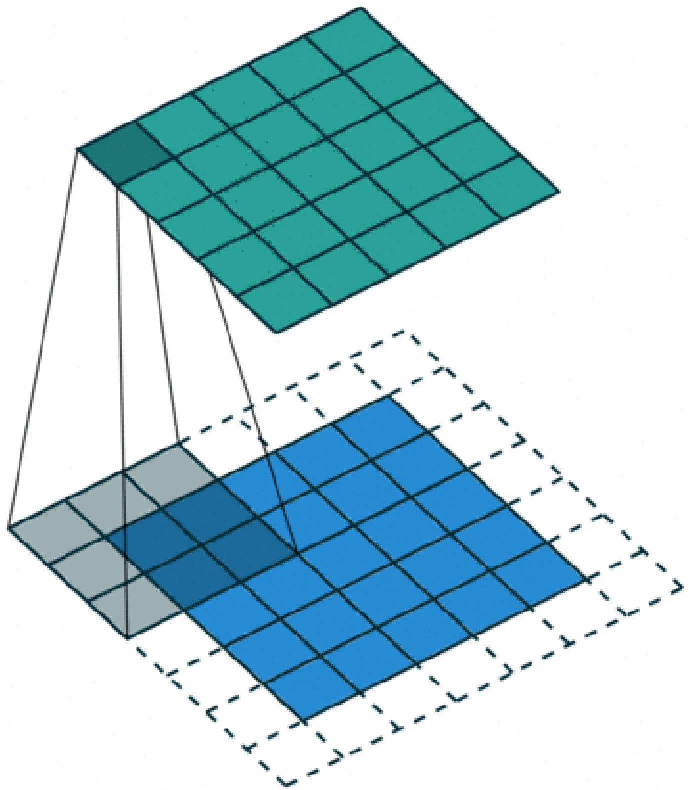
$$y(k_1, k_2) = h(k_1, k_2) ** x(k_1, k_2) = \sum_{i_1=0}^{M_1-1} \sum_{i_2=0}^{M_2-1} h(i_1, i_2) x(k_1 - i_1, k_2 - i_2).$$

- Usually discrete systems without feedback are FIR ones.

2D Discrete Systems

a) Image Lena; b) 5 × 5 moving average filter output.

2D Discrete Systems



Animation of 2D Convolution with input padding.

2D Discrete Systems

IIR Edge Detector output.

2D linear correlation

2D ***correlation*** of template image h and input image x (inner product):

$$r_{hx}(n_1,n_2) = \sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} h(k_1,k_2)\,x(n_1+k_1,n_2+k_2) = \mathbf{h}^T\mathbf{x}(n_1,n_2).$$

- $\mathbf{h} = [h(0,0),\dots,h(N_1-1,N_2-1)]^T$: template image vector.
- $\mathbf{x}(n_1,n_2) = [x(n_1,n_2),\dots,x(n_1+N_1-1,n_2+N_2-1)]^T$: local neighborhood (window) image vector.

2D Discrete Fourier Transform

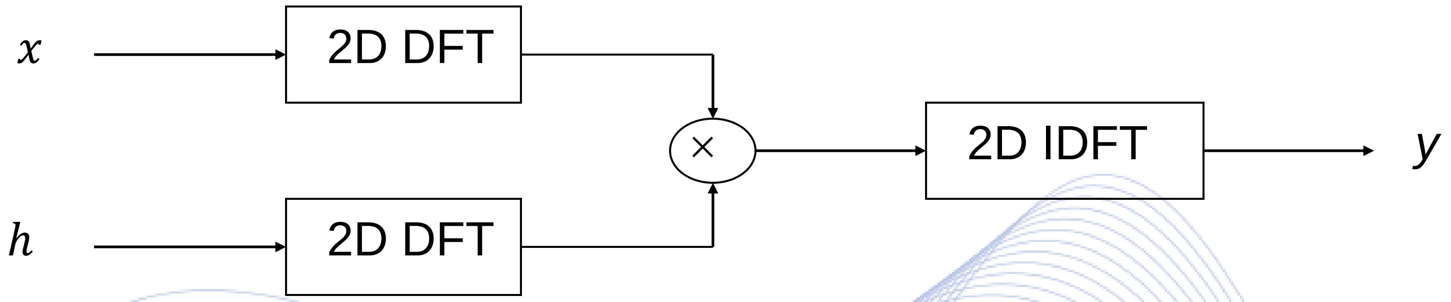
- Cyclic Convolution Theorem:

$$y(n_1,n_2) = x(n_1,n_2) \circledast\circledast\, h(n_1,n_2),$$
$$Y(k_1,k_2) = X(k_1,k_2)H(k_1,k_2).$$

- Cyclic Correlation:

$$r_{hx}(n_1,n_2) = h(n_1,n_2) \circledast\circledast\, x(-n_1,-n_2),$$
$$R_{hx}(k_1,k_2) = H^*(k_1,k_2)X(k_1,k_2).$$

2D Cyclic Convolution Calculation with DFT



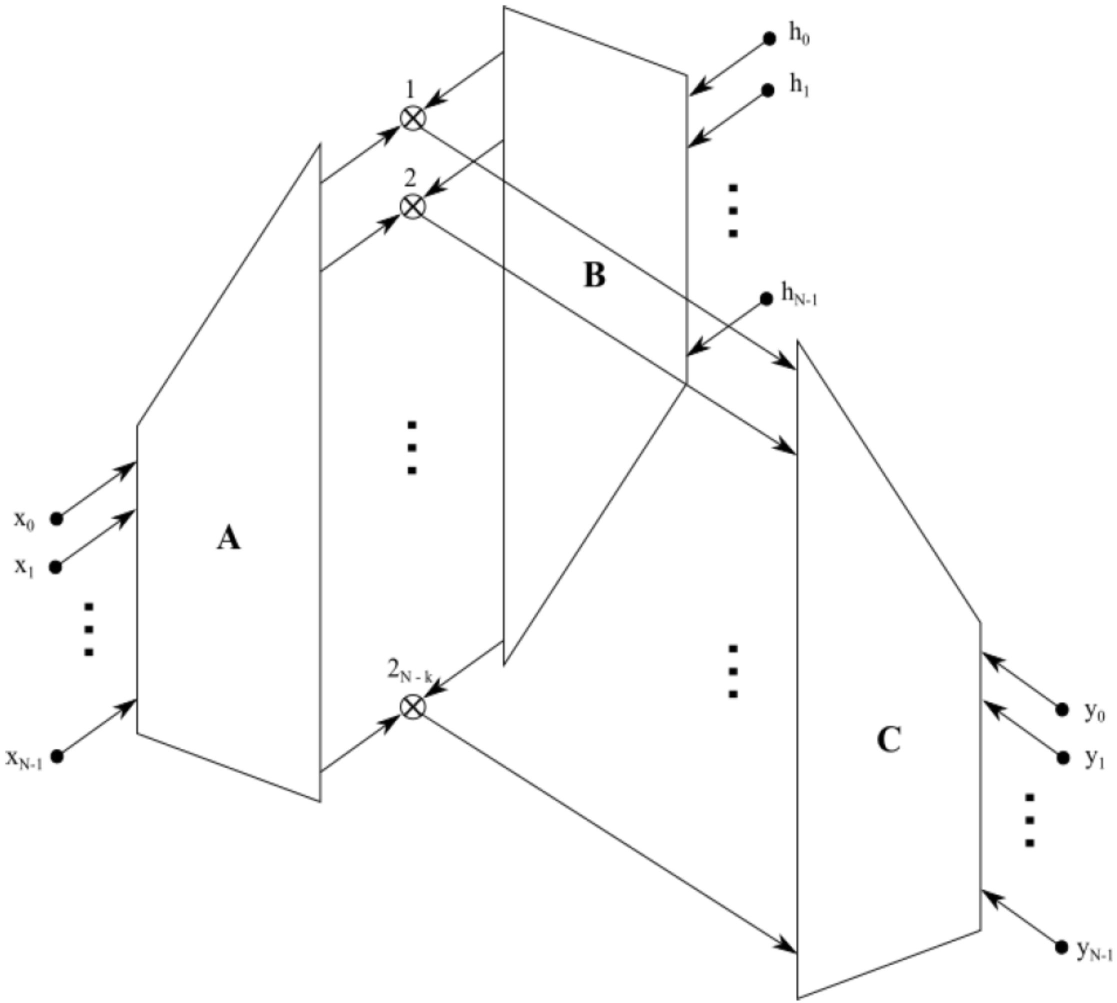
2D convolution calculation using the DFTs.

Winograd 2D cyclic convolution algorithm

- Winograd 2D convolution algorithms or fast 2D filtering:

$$\mathbf{y} = \mathbf{C}(\mathbf{A}\mathbf{x} \otimes \mathbf{B}\mathbf{h}).$$

- GEneral Matrix Multiplication (GEMM) BLAS or cuBLAS routines can be used.



Nested convolutions

- Winograd algorithms exist for relatively short convolution lengths.
- Use of efficient short-length convolution algorithms iteratively to build long convolutions
- Does not achieve minimal multiplication complexity
- Good balance between multiplications and additions

Decomposition:

- 2D convolution : $N \times N = N_1 N_2 \times N_1 N_2$, for N_1, N_2 coprime integers $(N_1, N_2) = 1$, can be implemented using nested $N_1 \times N_1$, $N_2 \times N_2$ convolutions.

Artificial Intelligence & Information Analysis Lab

Block-based convolution calculation

2D overlap-add algorithm is based on the distributive property of convolution:

- An image $x(i_1, i_2)$ can be divided into $K_1 \times K_2$ non-overlapping subsequences, having dimensions $N_{B1} \times N_{B2}$ each:

$$x_{k_1 k_2}(i_1, i_2) = \begin{cases} x(i_1, i_2) & k_1 N_{B1} \leq i_1 < (k_1+1)N_{B1},\ k_2 N_{B2} \leq i_2 < (k_2+1)N_{B2} \\ 0 & \text{otherwise.} \end{cases}$$

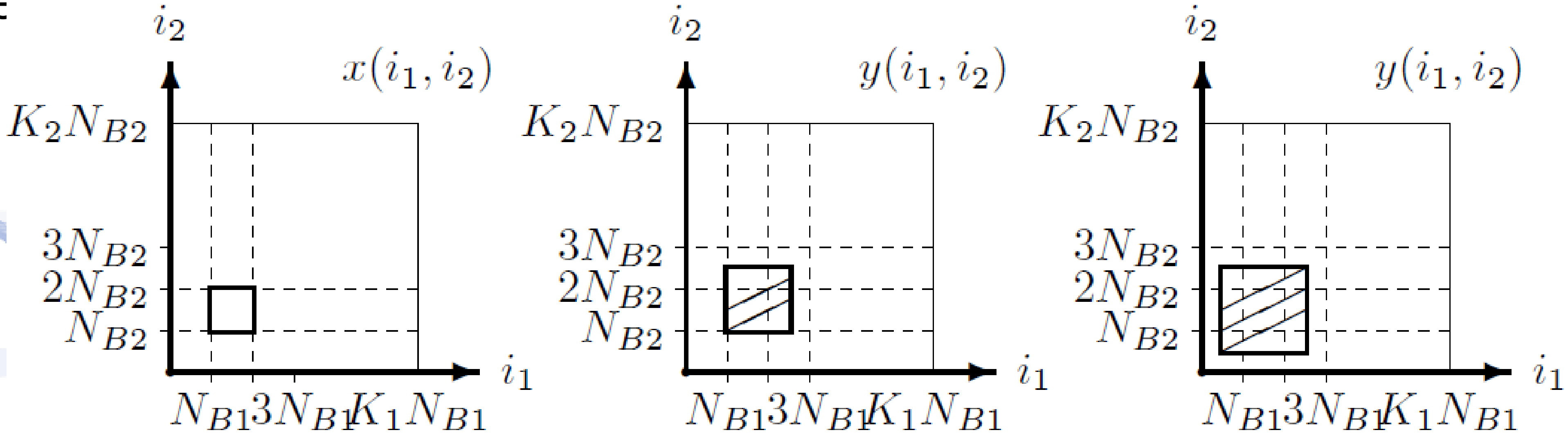
- The linear convolution output $y(n_1, n_2)$ is the sum of the convolution outputs produced by the input sequence blocks:

$$y(i_1, i_2) = x(i_1, i_2) ** h(i_1, i_2) = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \left(x_{k_1 k_2}(i_1, i_2)\right) ** h(i_1, i_2) = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} y_{k_1 k_2}(i_1, i_2).$$

Overlap-add algorithm

The 'partial' convolutions are performed using FFT and then adding the results:

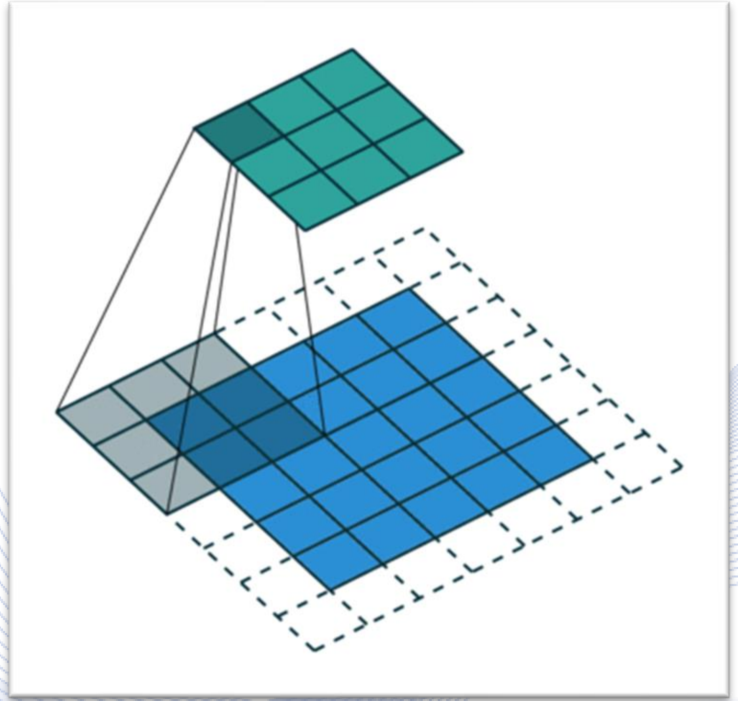
- The blocks and the filter are transformed to the frequency domain.
- Partial output blocks are calculated using the IFFT of the product as usual.
- Then all the overlapping blocks are added to construct the final output image



Convolutional Neural Networks

- The convolution kernel is described by the 4D tensor $\mathbf{W} \in \mathbb{R}^{h_1 \times h_2 \times d_{in} \times d_{out}}$:

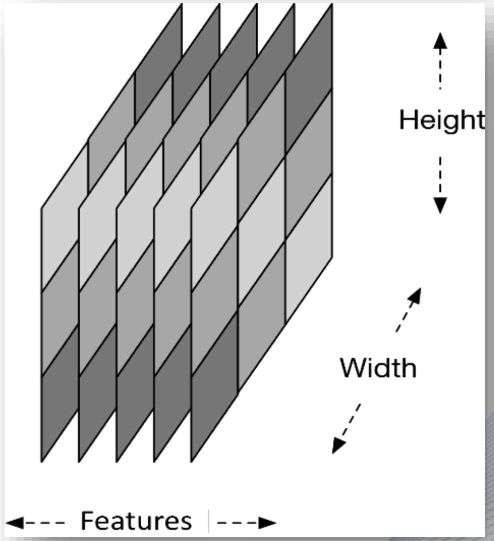
$$\mathbf{W} = [w_{k_1,k_2,r,o}: \quad k_1 = 1,\ldots,h_1, k_2 = 1,\ldots,h_2, r = 1,\ldots,d_{in}, o = 1,\ldots,d_{out}].$$



- r, o: they define the input and output channels.
- $h_1 \times h_2$: convolution mask sizes.

Convolutional CNN Layers

- For a convolutional layer l with an activation function $f_l(\cdot)$, multiple incoming features d_{in} and one single output feature o.



Multiple input features to single feature o transformation

$$y^{(l)}(i,j,o) = f_l\left(b^{(l)} + \sum_{r=1}^{d_{in}} \sum_{k_1=-q_1}^{q_1^{(l)}} \sum_{k_2=-q_2}^{q_2^{(l)}} w^{(l)}(k_1,k_2,r,o)\, x^{(l)}(i-k_1, j-k_2, r) \right)$$

Convolutional Layer Activation Volume (3D tensor)

$$a_{ij}^{(l)}(o) = f_l\left(b^{(l)}(o) + \sum_{r=1}^{d_{in}} \boldsymbol{W}^{(l)}(r,o) * \boldsymbol{X}_{ij}^{(l)}(r) \right) \quad \boldsymbol{A}^{(l)} = [a_{ij}^{(l)}(o){:}\, i = 1,..,n^{(l)}, j = 1,..,m^{(l)}, o = 1,\ldots,d_{out}]$$

where $\boldsymbol{A}^{(l)}$ is the activation volume for the convolutional layer l, $\boldsymbol{W}^{(l)}(r,o)$ is a 2D slice of the convolutional kernel $\boldsymbol{W}^{(l)} \in \mathbb{R}^{h_1 \times h_2 \times d_{in} \times d_{out}}$ for input feature r and output feature o , $b^{(l)}(o)$ a scalar bias and $\boldsymbol{X}_{ij}^{(l)}(r)$ a region of input feature r centered at $[i,j]^T$, e.g. $\boldsymbol{X}^{(1)}(1)$ the R channel of an image $d_{in} = C = 3$.

Q & A

Thank you very much for your attention!

Contact: Prof. I. Pitas
pitas@csd.auth.gr