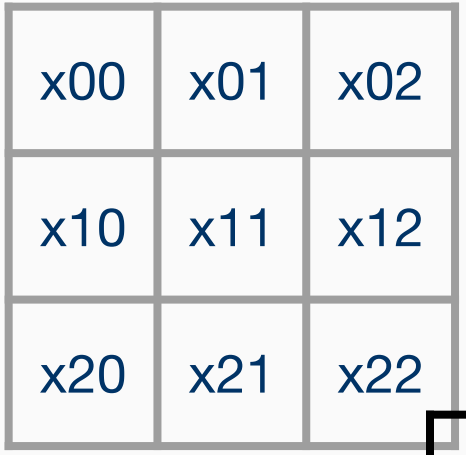
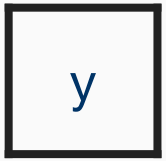
Convolution Layer and Pooling Operation

Motivation and Architecture

AUA, DS330 Deep Learning 2021, Hovhannes Tamoyan





$y=x00*w00+x01*w01+...+x22*w22$

$y=sum(xij*wij)$

Base way

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output

image: n x n
kernel: f x f
output: n-f+1 x n-f+1

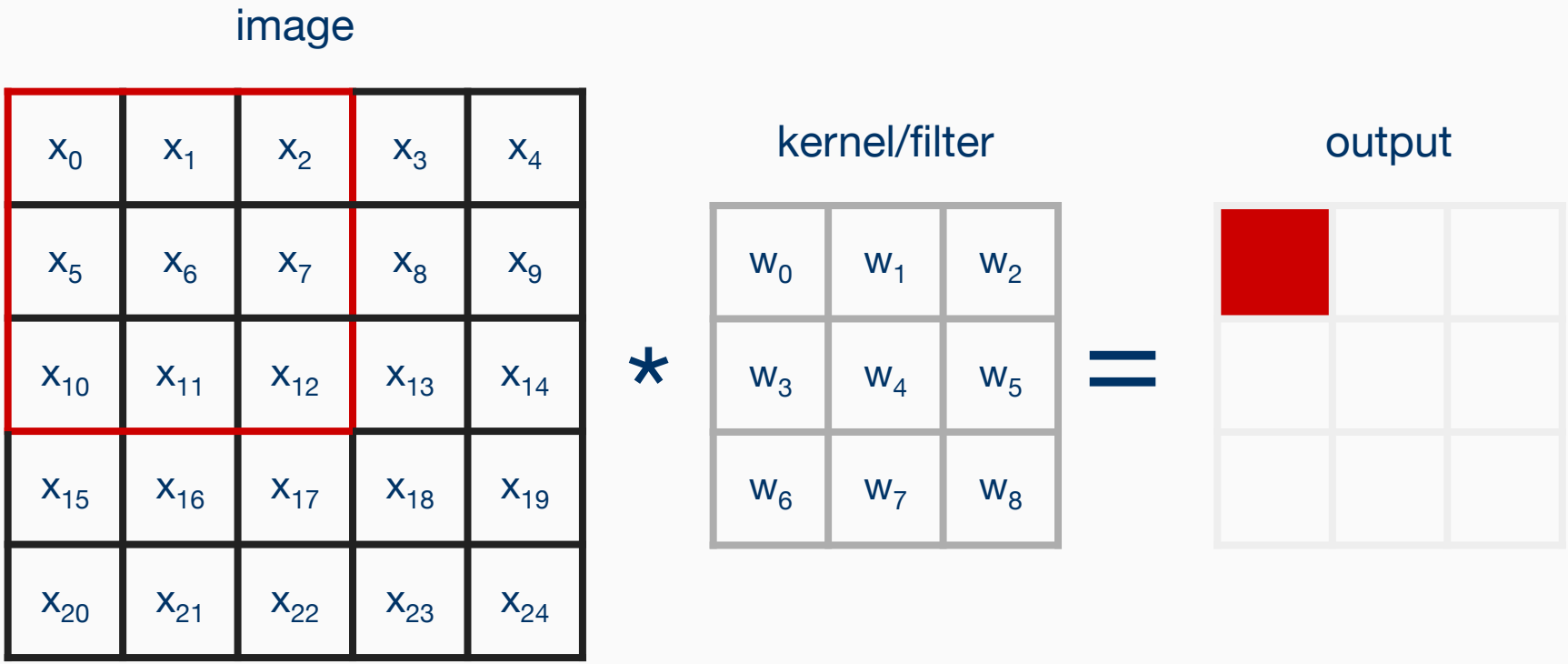


image
x_0 x_1 x_2 x_3 x_4
x_5 x_6 x_7 x_8 x_9
x_{10} x_{11} x_{12} x_{13} x_{14}
x_{15} x_{16} x_{17} x_{18} x_{19}
x_{20} x_{21} x_{22} x_{23} x_{24}
*
kernel/filter
w_0 w_1 w_2
w_3 w_4 w_5
w_6 w_7 w_8
=
output

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

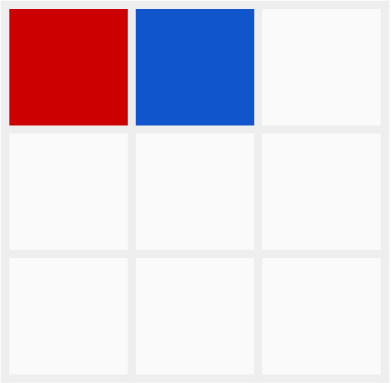
$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output



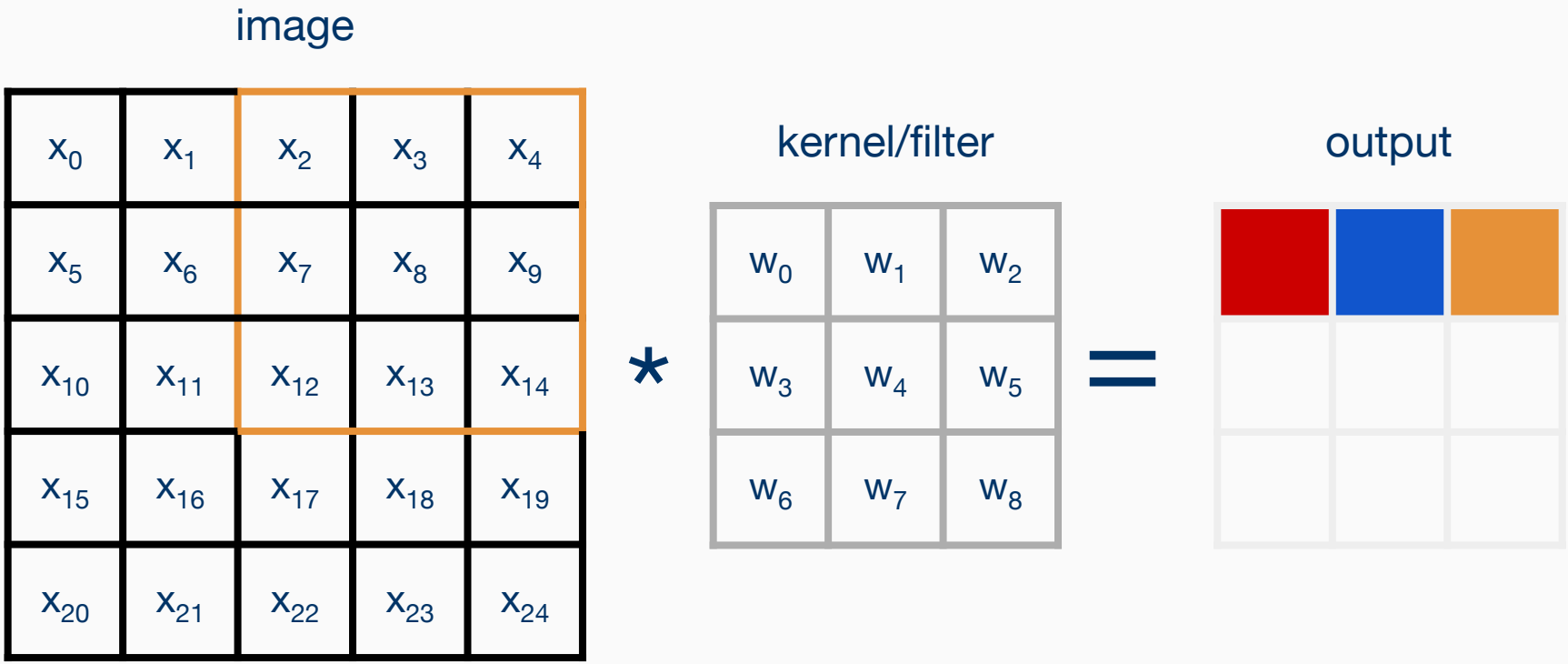


image
x_0 x_1 x_2 x_3 x_4
x_5 x_6 x_7 x_8 x_9
x_{10} x_{11} x_{12} x_{13} x_{14}
x_{15} x_{16} x_{17} x_{18} x_{19}
x_{20} x_{21} x_{22} x_{23} x_{24}
*
kernel/filter
w_0 w_1 w_2
w_3 w_4 w_5
w_6 w_7 w_8
=
output

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

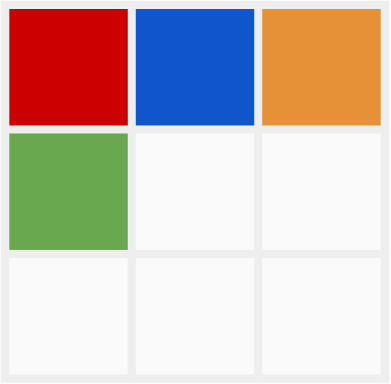
$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output



image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

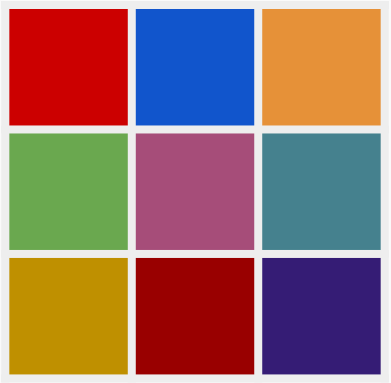
$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output



Stride

stride=2

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output

image: n x n
kernel: f x f
output: (n-f)/s+1 x (n-f)/s+1

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

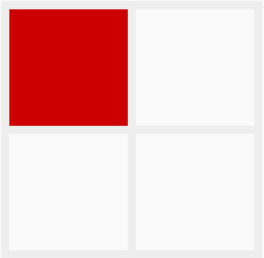
$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output



stride=2

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

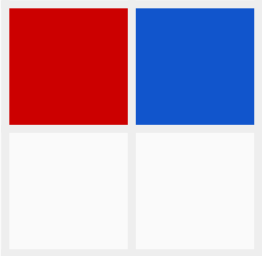
$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output



stride=2

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

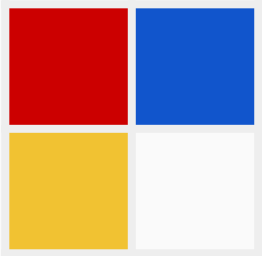
$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output



stride=2

image

x_0	x_1	x_2	x_3	x_4
x_5	x_6	x_7	x_8	x_9
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}

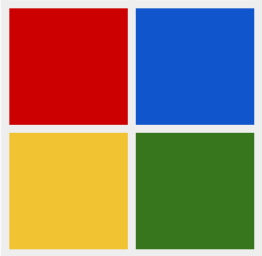
$*$

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$

output



Padding

image

0	0	0	0	0	0	0
0	x_0	x_1	x_2	x_3	x_4	0
0	x_5	x_6	x_7	x_8	x_9	0
0	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	0
0	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	0
0	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	0
0	0	0	0	0	0	0

*

stride=2
padding=1

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

=

output

image: n x n
kernel: f x f
output: (n+2p-f)/s+1 x (n+2p-f)/s+1

image

stride=2
padding=1

kernel/filter

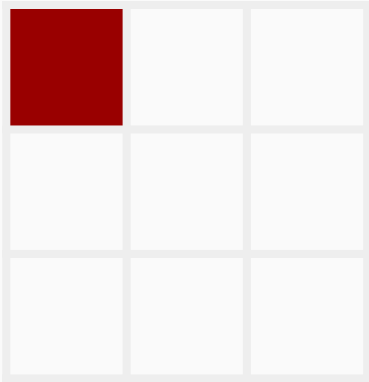
output

0	0	0	0	0	0	0
0	x_0	x_1	x_2	x_3	x_4	0
0	x_5	x_6	x_7	x_8	x_9	0
0	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	0
0	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	0
0	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	0
0	0	0	0	0	0	0

$*$

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$



image

stride=2
padding=1

kernel/filter

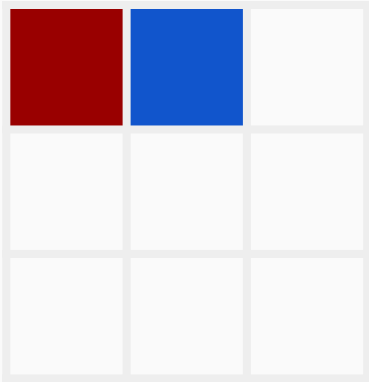
output

0	0	0	0	0	0	0
0	x_0	x_1	x_2	x_3	x_4	0
0	x_5	x_6	x_7	x_8	x_9	0
0	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	0
0	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	0
0	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	0
0	0	0	0	0	0	0

$*$

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$



image

0	0	0	0	0	0	0
0	x_0	x_1	x_2	x_3	x_4	0
0	x_5	x_6	x_7	x_8	x_9	0
0	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	0
0	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	0
0	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	0
0	0	0	0	0	0	0

*

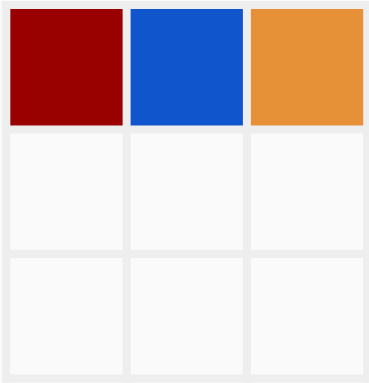
stride=2
padding=1

kernel/filter

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

=

output



image

stride=2
padding=1

kernel/filter

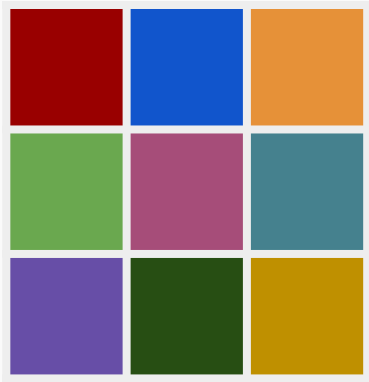
output

0	0	0	0	0	0	0
0	x_0	x_1	x_2	x_3	x_4	0
0	x_5	x_6	x_7	x_8	x_9	0
0	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	0
0	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	0
0	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	0
0	0	0	0	0	0	0

$*$

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

$=$



Final “equation” of Convolution shapes

image: n x n
kernel: f x f
output: (n+2p-f)/s+1 x (n+2p-f)/s+1

Filter is a collection of kernels. The term kernel and filter are interchangeable in one channel case, but when the input channels are more than one we need to use the term filter.
There being one kernel for every single input channel to the layer, and each kernel being unique.

Pooling

Popular types/operations of Pooling

Max pooling
Min pooling
Mean(Average) pooling

image

stride=2

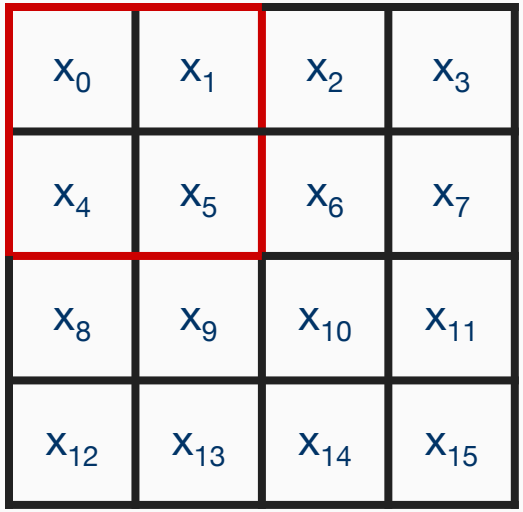
x_0	x_1	x_2	x_3
x_4	x_5	x_6	x_7
x_8	x_9	x_{10}	x_{11}
x_{12}	x_{13}	x_{14}	x_{15}

output

image: n x n
pooling: m x m
output: (n-m)/s + 1 x (n-m)/s + 1

stride=2

image



op(matrix)

output

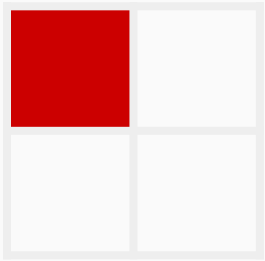


image: n x n
pooling: m x m
output: (n-m)/s + 1 x (n-m)/s + 1

stride=2

image

x_0	x_1	x_2	x_3
x_4	x_5	x_6	x_7
x_8	x_9	x_{10}	x_{11}
x_{12}	x_{13}	x_{14}	x_{15}

op(matrix)

output

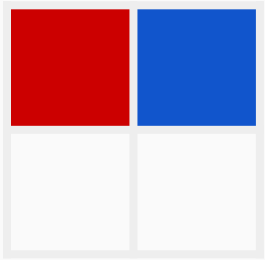


image: n x n
pooling: m x m
output: (n-m)/s + 1 x (n-m)/s + 1

image

stride=2

x_0	x_1	x_2	x_3
x_4	x_5	x_6	x_7
x_8	x_9	x_{10}	x_{11}
x_{12}	x_{13}	x_{14}	x_{15}

op(matrix)

output

image: n x n
pooling: m x m
output: (n-m)/s + 1 x (n-m)/s + 1

image

stride=2

x_0	x_1	x_2	x_3
x_4	x_5	x_6	x_7
x_8	x_9	x_{10}	x_{11}
x_{12}	x_{13}	x_{14}	x_{15}

op(matrix)

output

image: n x n
pooling: m x m
output: (n-m)/s + 1 x (n-m)/s + 1

Final “equation” of Convolution shapes

image: n x n
pooling: m x m
output: (n-m)/s + 1 x (n-m)/s + 1

Thank You