



# Convolutional Neural Networks

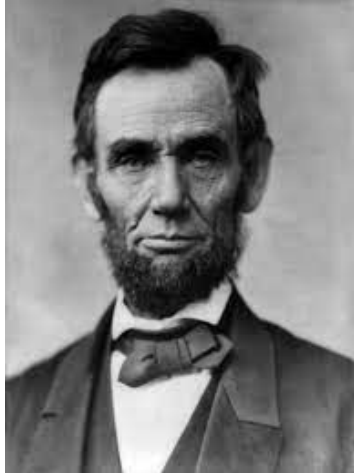
Forward Pass

Introduction to Deep Learning - 11-485/685/785 - Fall 2022

Abuzar Khan

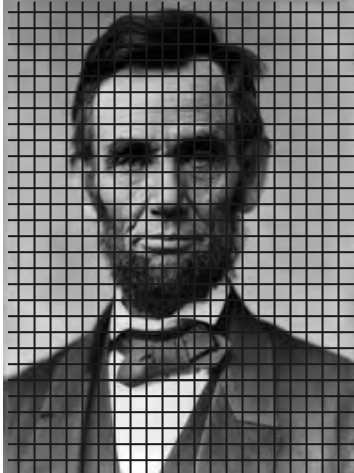
# What is an image?

A thousand words.



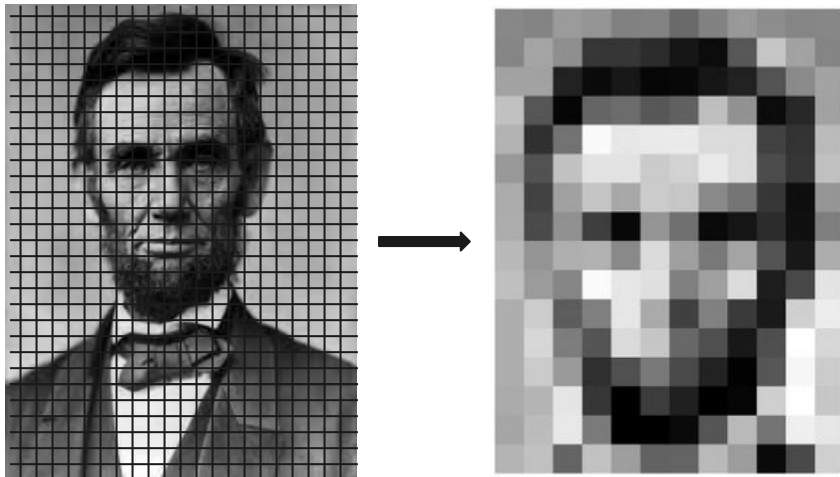
# What is an image?

A thousand words.



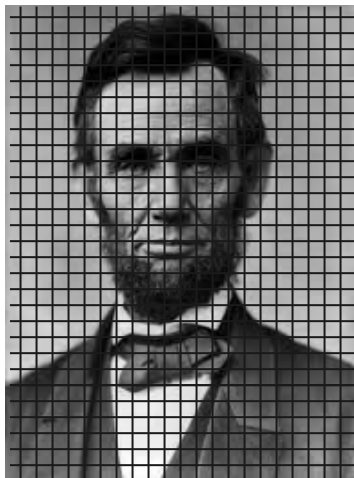
# What is an image?

A thousand words.



# What is an image?

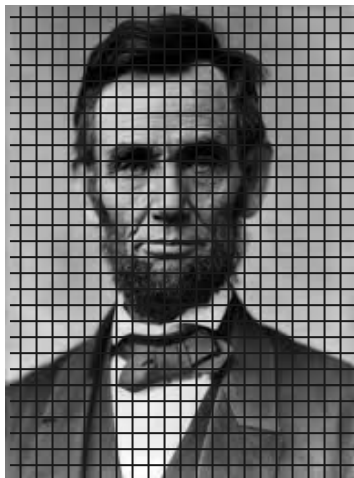
A thousand words.



157	153	174	168	150	162	129	161	172	163	165	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	54	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	96	74	206
188	88	179	200	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	73	1	81	47	0	6	217	256	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

# What is an image?

A thousand words.

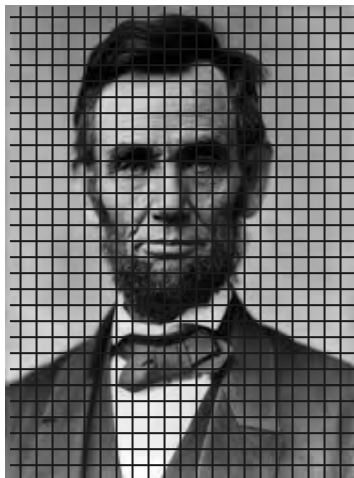


157	153	174	168	150	162	129	161	172	161	155	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	54	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	162	129	161	172	161	155	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	54	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

# What is an image?

~~A thousand words.~~ A Matrix  $I$  of dimensions  $(M,N)$  with  $I[i][j] = \text{intensity}(\text{pixel}(i,j))$



157	153	174	168	150	162	129	161	172	161	155	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	54	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	162	129	161	172	161	155	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	54	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

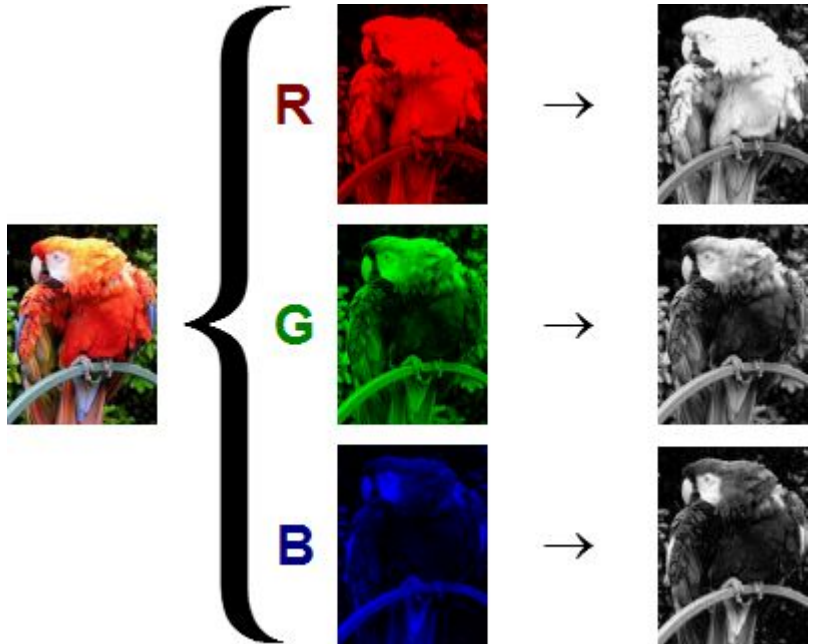
# Beyond B/W

Colored Images:

- Multi-channel
- R,G,B (an example)
- $I \rightarrow (3, M, N)$

$I[c][i][j] =$

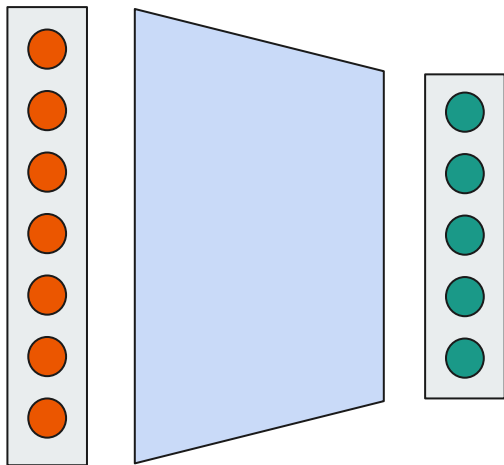
**Intensity** at **pixel**( $i, j$ ) for  
channel  $c$







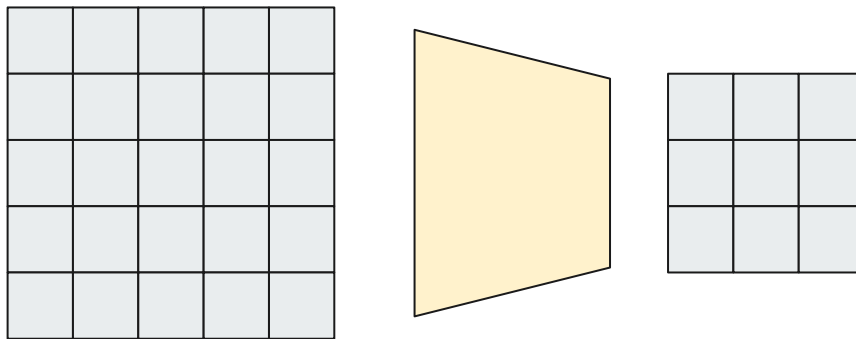
# MLP



Vector to Vector



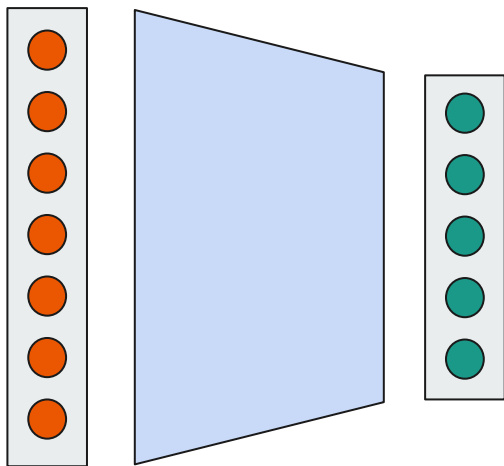
# CNN



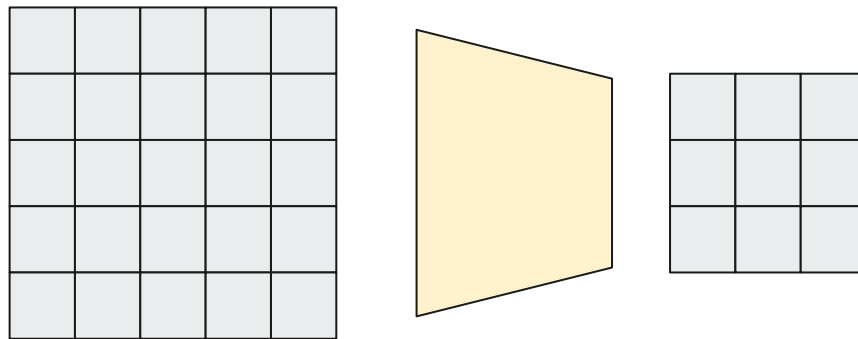
Feature map to Feature map



## MLP Vs. CNN



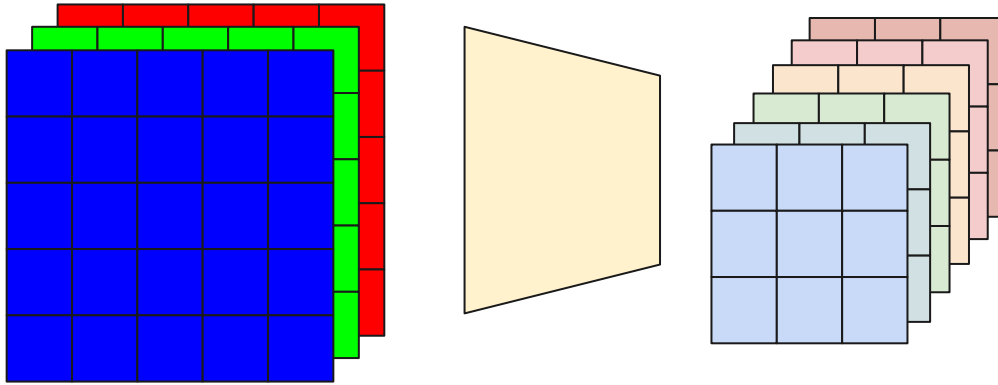
Vector to Vector



Feature map to Feature map



## Multi-channel CNN





# Components of a CNN

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$

Input - **A**

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

Kernel - **W**

$B_{1,1}$
-----------

Bias - **B**

# Components of a CNN

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$

Input -  $\mathbf{A}$

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

Kernel -  $\mathbf{W}$

$B_{1,1}$
-----------

Bias -  $\mathbf{B}$

$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$

Output -  $\mathbf{Z}$

$$\mathbf{Z} = (\mathbf{A} \otimes \mathbf{W}) + \mathbf{B}$$



# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations

# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations

The diagram illustrates the calculation of  $z_{1,1}$  in a CNN. It shows a 4x4 input matrix  $A$ , a 2x2 weight matrix  $W$ , and a bias  $B_{1,1}$  being combined via element-wise multiplication and summation to produce  $z_{1,1}$ .

Input Matrix  $A$  (4x4):

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$

Weight Matrix  $W$  (2x2):

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

Bias  $B_{1,1}$  (1x1):

$B_{1,1}$
-----------

Calculation:

$$A * W + B_{1,1} = z_{1,1}$$

Result  $z_{1,1}$  (1x1):

$z_{1,1}$
-----------

Formula for  $z_{1,1}$ :

$$z_{1,1} = (A_{1,1} * W_{1,1}) + (A_{1,2} * W_{1,2}) + (A_{2,1} * W_{2,1}) + (A_{2,2} * W_{2,2}) + B$$



# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations

The diagram illustrates a CNN step involving element-wise multiplication and summation. It shows a 4x4 input matrix  $A$  (with elements  $A_{1,1}$  to  $A_{4,4}$ ), a 2x2 weight matrix  $W$  (with elements  $W_{1,1}$  to  $W_{2,2}$ ), a bias  $B_{1,1}$ , and a resulting 2x2 output matrix  $Z$  (with elements  $Z_{1,1}$  to  $Z_{1,2}$ ). The operation is represented as  $A * W + B = Z$ .

$$Z_{1,2} = (A_{1,2} * W_{1,1}) + (A_{1,3} * W_{1,2}) + (A_{2,2} * W_{2,1}) + (A_{2,3} * W_{2,2}) + B$$

# CNN Steps

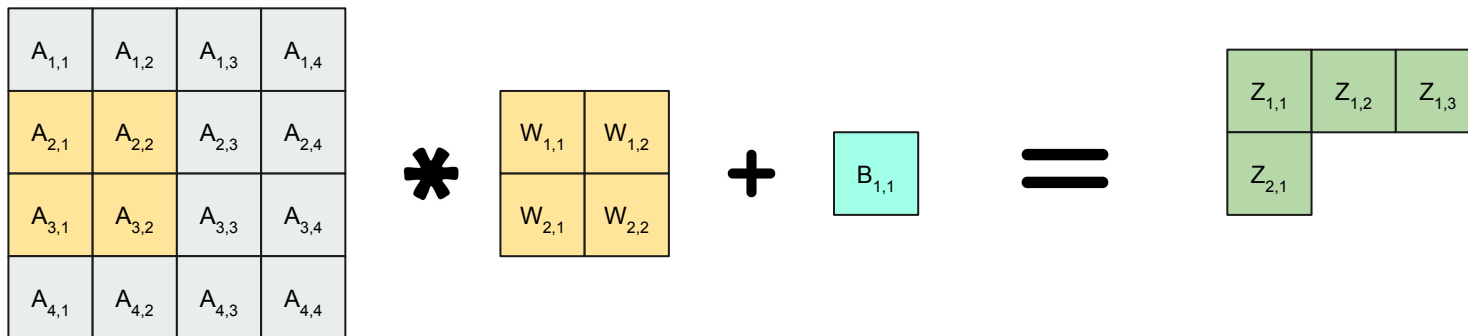
Essentially element-wise (Hadamard) multiplications and summations

The diagram illustrates a CNN step involving element-wise multiplication and summation. It shows a 4x4 input matrix  $A$  (with elements  $A_{1,1}$  to  $A_{4,4}$ ), a 2x2 weight matrix  $W$  (with elements  $W_{1,1}$  to  $W_{2,2}$ ), a bias  $B_{1,1}$ , and a resulting 1x3 output vector  $Z$  (with elements  $Z_{1,1}$  to  $Z_{1,3}$ ). The operation is represented as  $A * W + B = Z$ .

$$Z_{1,3} = (A_{1,3} * W_{1,1}) + (A_{1,4} * W_{1,2}) + (A_{2,3} * W_{2,1}) + (A_{2,4} * W_{2,2}) + B$$

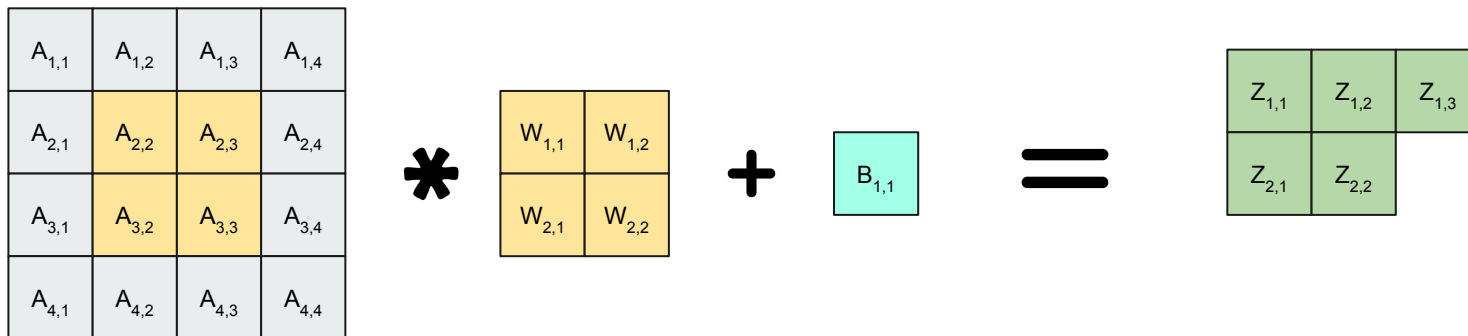
# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations



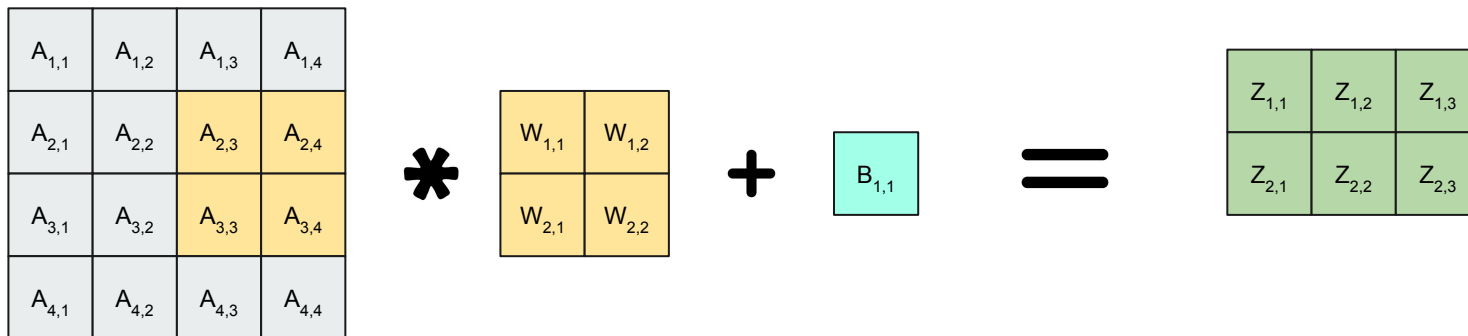
# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations



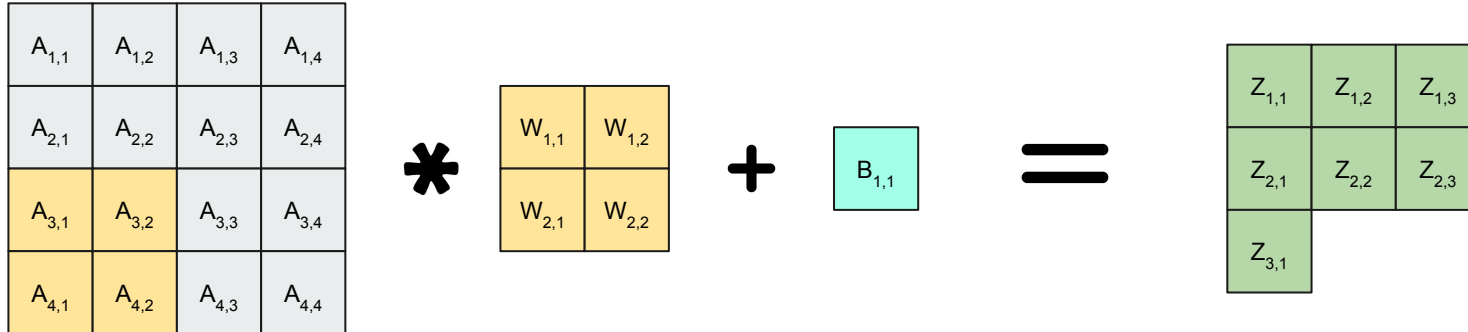
# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations



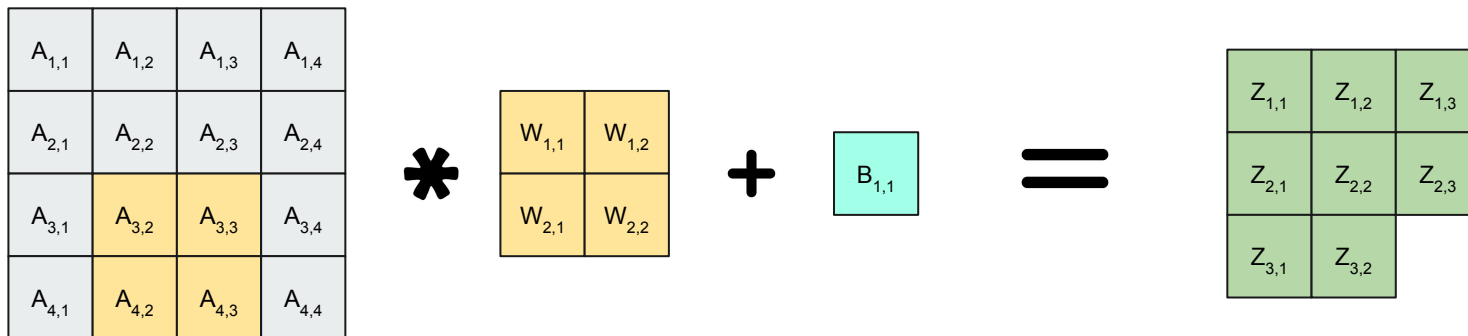
# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations



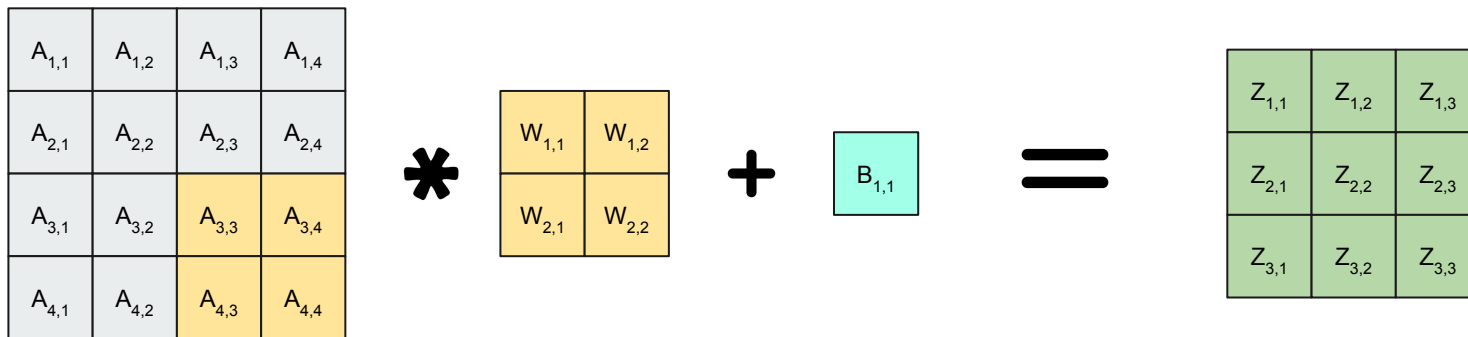
# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations



# CNN Steps

Essentially element-wise (Hadamard) multiplications and summations







# Output Size

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$



$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$



## Output Size

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$



$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$

$$\text{Output Width} = \left[ \frac{(W_{\text{in}} - W_k + 2P)}{(S)} \right] + 1$$

Same goes for Height.



## Output Size

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$



$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$

$$\text{Output Width} = \left[ \frac{(W_{\text{in}} - W_k + 2\mathbf{P})}{(\mathbf{S})} \right] + 1$$

# Output Size

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$



$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$

$$\text{Output Width} = \left[ \frac{(W_{\text{in}} - W_k + 2\mathbf{P})}{(\mathbf{S})} \right] + 1$$

**P**: Padding (here - 0)

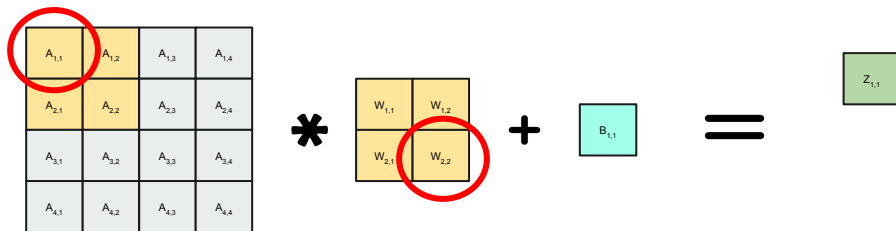
**S**: Stride (here - 1)



# Padding

- Attaching zeros (usually) around inputs.
- Seen it before in HW1.
- Images can be padded to the left, right, top, and bottom.

# Padding



$$z_{1,1} = (A_{1,1} * W_{1,1}) + (A_{1,2} * W_{1,2}) + (A_{2,1} * W_{2,1}) + (A_{2,2} * W_{2,2}) + B_{1,1}$$

# Padding

Diagram illustrating a 1D convolution operation. A 4x4 input matrix  $A$  is multiplied (indicated by  $*$ ) by a 2x2 kernel matrix  $W$ . The result is then added (indicated by  $+$ ) to a 1x1 bias matrix  $B_{1,1}$  to produce a 1x1 output matrix  $Z_{1,1}$ . Red circles highlight the top-left element  $A_{1,1}$  in the input matrix and the bottom-right element  $W_{2,2}$  in the kernel matrix.

Diagram illustrating a 2D convolution operation. A 4x4 input matrix  $A$  is multiplied (indicated by  $*$ ) by a 2x2 kernel matrix  $W$ . The result is then added (indicated by  $+$ ) to a 1x1 bias matrix  $B_{1,1}$  to produce a 3x3 output matrix  $Z$ . Red circles highlight the top-left element  $W_{1,1}$  in the kernel matrix and the bottom-right element  $A_{4,4}$  in the input matrix.

# Padding

Diagram illustrating a matrix operation:

$$\begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \\ A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} \end{bmatrix} * \begin{bmatrix} W_{1,1} & W_{1,2} \\ W_{2,1} & W_{2,2} \end{bmatrix} + \begin{bmatrix} B_{1,1} \end{bmatrix} = \begin{bmatrix} Z_{1,1} \end{bmatrix}$$

The element  $A_{1,1}$  in the first matrix and  $W_{2,2}$  in the second matrix are circled in red.

Diagram illustrating a matrix operation:

$$\begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \\ A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} \end{bmatrix} * \begin{bmatrix} W_{1,1} & W_{1,2} \\ W_{2,1} & W_{2,2} \end{bmatrix} + \begin{bmatrix} B_{1,1} \end{bmatrix} = \begin{bmatrix} Z_{1,1} & Z_{1,2} & Z_{1,3} \\ Z_{2,1} & Z_{2,2} & Z_{2,3} \\ Z_{3,1} & Z_{3,2} & Z_{3,3} \end{bmatrix}$$

The elements  $W_{1,1}$  and  $A_{4,4}$  are circled in red.

Never Meet...





# Padding

Increase output size

Preserve input size

**More Kernel Interactions!**

# Padding

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$

\*

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

+

$B_{1,1}$
-----------

=

$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$

# Padding

0	0	0	0	0	0
0	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	0
0	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	0
0	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$	0
0	$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$	0
0	0	0	0	0	0

\*

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

+

$B_{1,1}$
-----------

=

$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$	$Z_{1,4}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$	$Z_{2,4}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$	$Z_{3,4}$
$Z_{4,1}$	$Z_{4,2}$	$Z_{4,3}$	$Z_{4,4}$

# Padding

0	0	0	0	0	0
0	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	0
0	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	0
0	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$	0
0	$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$	0
0	0	0	0	0	0

\*

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

+

$B_{1,1}$
-----------

=

$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$	$Z_{1,4}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$	$Z_{2,4}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$	$Z_{3,4}$
$Z_{4,1}$	$Z_{4,2}$	$Z_{4,3}$	$Z_{4,4}$

# Padding

0	0	0	0	0	0
0	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	0
0	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	0
0	$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$	0
0	$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$	0
0	0	0	0	0	0

\*

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

+

$B_{1,1}$
-----------

=

$Z_{1,1}$	$Z_{1,2}$	$Z_{1,3}$	$Z_{1,4}$
$Z_{2,1}$	$Z_{2,2}$	$Z_{2,3}$	$Z_{2,4}$
$Z_{3,1}$	$Z_{3,2}$	$Z_{3,3}$	$Z_{3,4}$
$Z_{4,1}$	$Z_{4,2}$	$Z_{4,3}$	$Z_{4,4}$



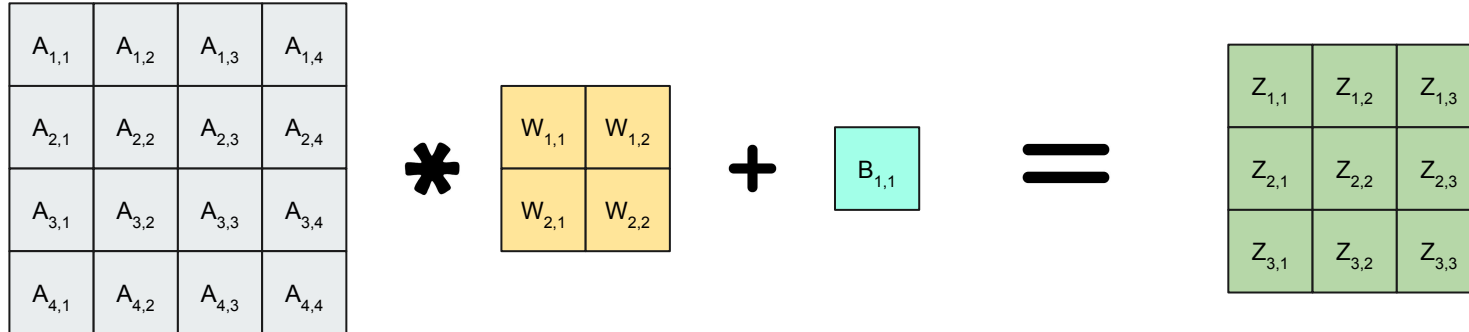


# Stride

Taking bigger steps!

# Stride = 1

What we did before - The kernel “moves” one pixel (or element) at a time.



## Stride = 2

Start at the same place

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$

\*

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

+

$B_{1,1}$
-----------

=

$Z_{1,1}$
-----------

$$Z_{1,1} = (A_{1,1} * W_{1,1}) + (A_{1,2} * W_{1,2}) + (A_{2,1} * W_{2,1}) + (A_{2,2} * W_{2,2}) + B$$



## Stride = 2

Move two elements to the right

$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{2,4}$
$A_{3,1}$	$A_{3,2}$	$A_{3,3}$	$A_{3,4}$
$A_{4,1}$	$A_{4,2}$	$A_{4,3}$	$A_{4,4}$

\*

$W_{1,1}$	$W_{1,2}$
$W_{2,1}$	$W_{2,2}$

+

$B_{1,1}$
-----------

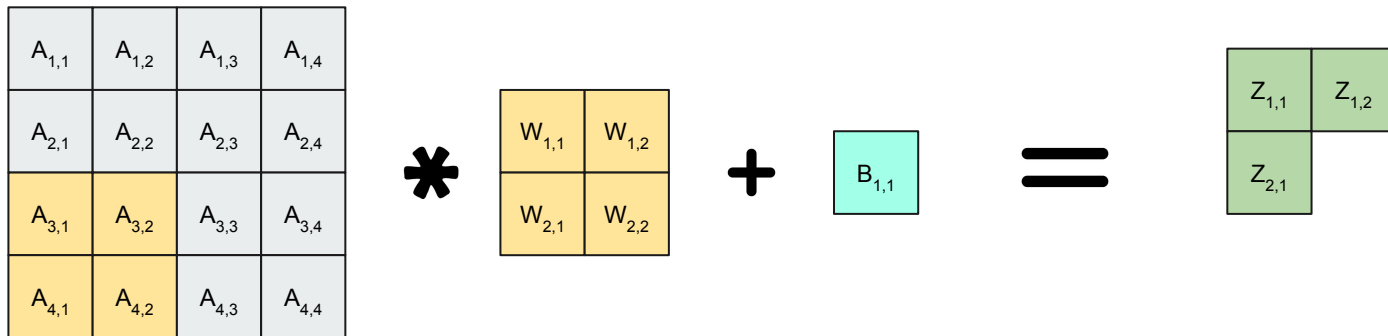
=

$Z_{1,1}$	$Z_{1,2}$
-----------	-----------

$$Z_{1,2} = (A_{1,3} * W_{1,1}) + (A_{1,4} * W_{1,2}) + (A_{2,3} * W_{2,1}) + (A_{2,4} * W_{2,2}) + B$$

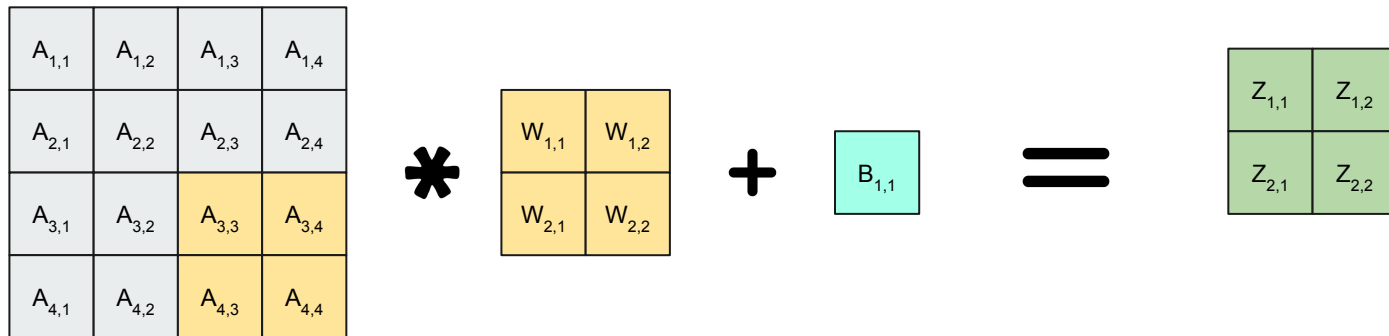
## Stride = 2

Move two elements down.

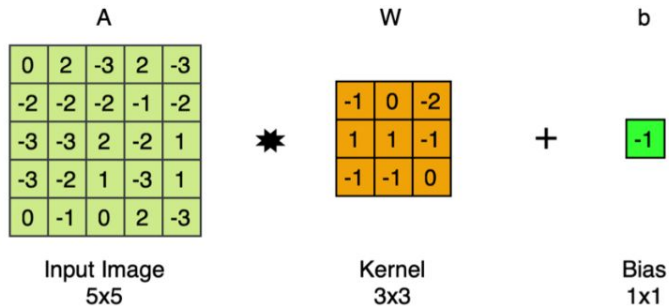


## Stride = 2

Move two elements to the right.



# Interpreting Stride > 1



# Interpreting Stride > 1

$$\begin{array}{c}
 \text{A} \\
 \begin{array}{|c|c|c|c|c|}
 \hline
 0 & 2 & -3 & 2 & -3 \\
 \hline
 -2 & -2 & -2 & -1 & -2 \\
 \hline
 -3 & -3 & 2 & -2 & 1 \\
 \hline
 -3 & -2 & 1 & -3 & 1 \\
 \hline
 0 & -1 & 0 & 2 & -3 \\
 \hline
 \end{array} \\
 \text{Input Image} \\
 5 \times 5
 \end{array}
 *
 \begin{array}{c}
 \text{W} \\
 \begin{array}{|c|c|c|}
 \hline
 -1 & 0 & -2 \\
 \hline
 1 & 1 & -1 \\
 \hline
 -1 & -1 & 0 \\
 \hline
 \end{array} \\
 \text{Kernel} \\
 3 \times 3
 \end{array}
 +
 \begin{array}{c}
 \text{b} \\
 \begin{array}{|c|}
 \hline
 -1 \\
 \hline
 \end{array} \\
 \text{Bias} \\
 1 \times 1
 \end{array}$$

9	-9	7
2	5	6
-7	9	-10

Stride 1 output

9	7
-7	-10

Stride 2 output

# Interpreting Stride > 1

$$\begin{array}{c}
 \text{A} \\
 \begin{bmatrix} 0 & 2 & -3 & 2 & -3 \\ -2 & -2 & -2 & -1 & -2 \\ -3 & -3 & 2 & -2 & 1 \\ -3 & -2 & 1 & -3 & 1 \\ 0 & -1 & 0 & 2 & -3 \end{bmatrix} \\
 \text{Input Image} \\
 5 \times 5
 \end{array}
 \star
 \begin{array}{c}
 \text{W} \\
 \begin{bmatrix} -1 & 0 & -2 \\ 1 & 1 & -1 \\ -1 & -1 & 0 \end{bmatrix} \\
 \text{Kernel} \\
 3 \times 3
 \end{array}
 +
 \begin{array}{c}
 \text{b} \\
 \begin{bmatrix} -1 \end{bmatrix} \\
 \text{Bias} \\
 1 \times 1
 \end{array}$$

9	-9	7
2	5	6
-7	9	-10

Stride 1 output



9	-9	7
2	5	6
-7	9	-10

Drop intermediates

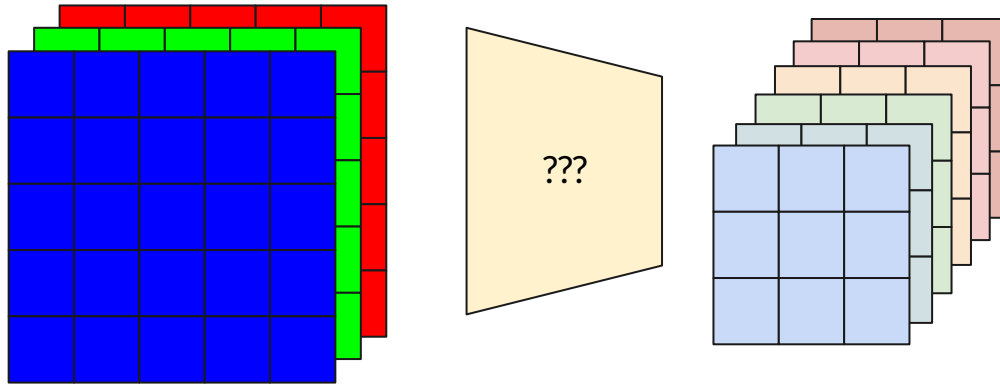


9	7
-7	-10

Stride 2 output



## Multi-channel CNN





## Multi-channel CNN

- Each kernel (or **filter**) has as many channels as the input does
- Channel **c** of the **kernel** convolves with channel **c** (corresponding) of the **input**.
- The number of output channels from the convolution = number of **filters**

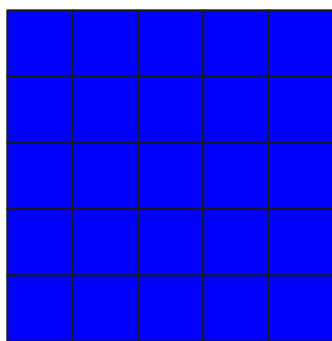
$C_{in}$  = Input channels

$C_{kernel} = \text{Kernel channels} = C_{in}$

$K = \text{Number of Kernels} = C_{out} = \text{Number of output channels}$

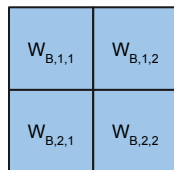


# 1 Filter with 3-channel input



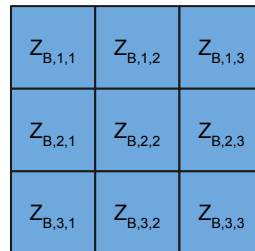
1 channel input

$\otimes$



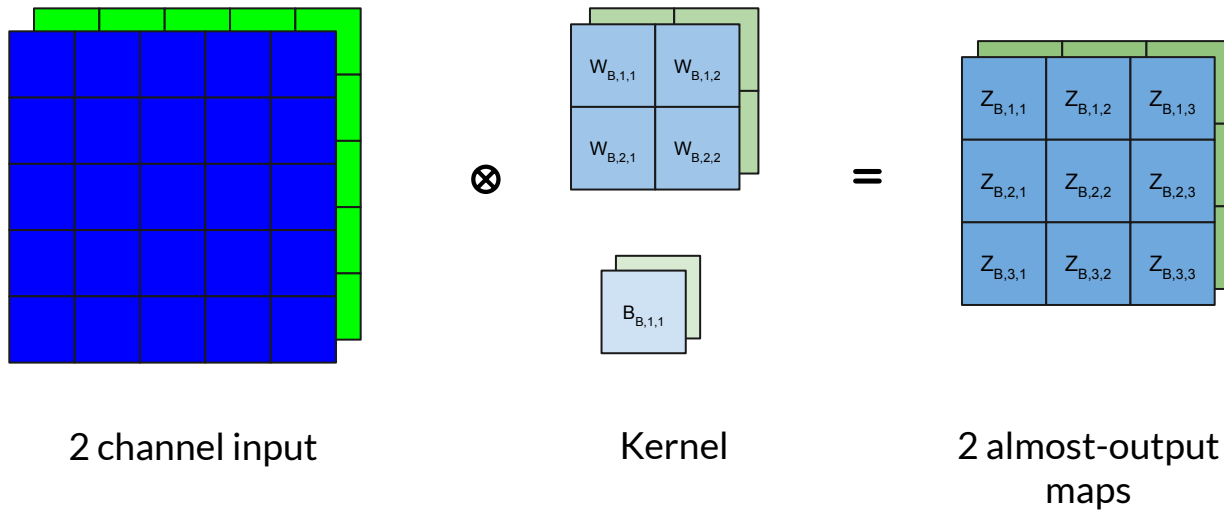
Kernel

=

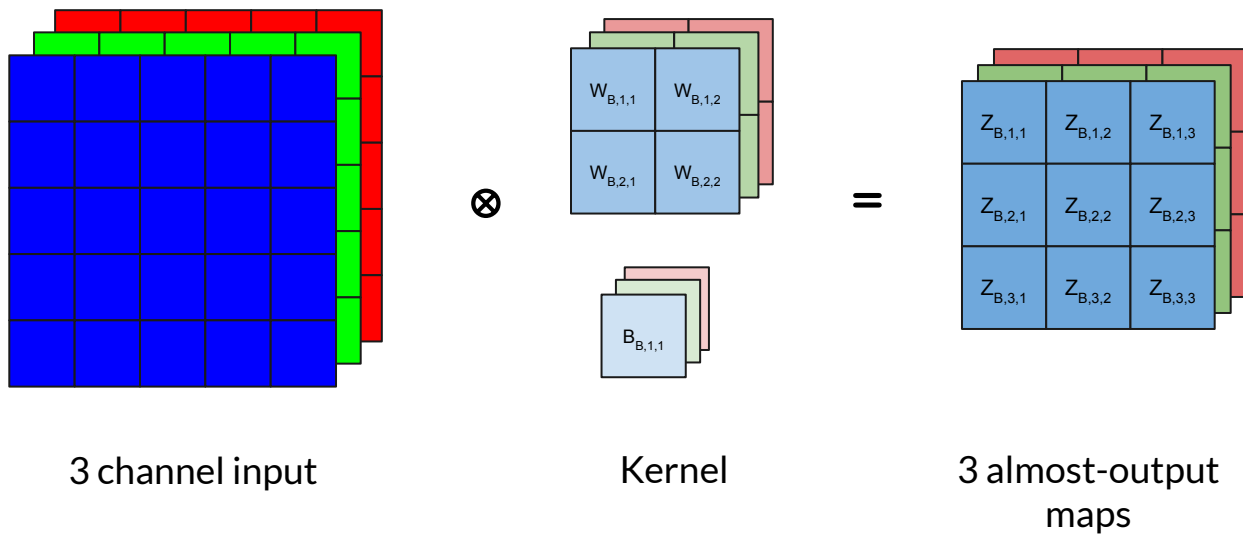


1 almost-output  
map

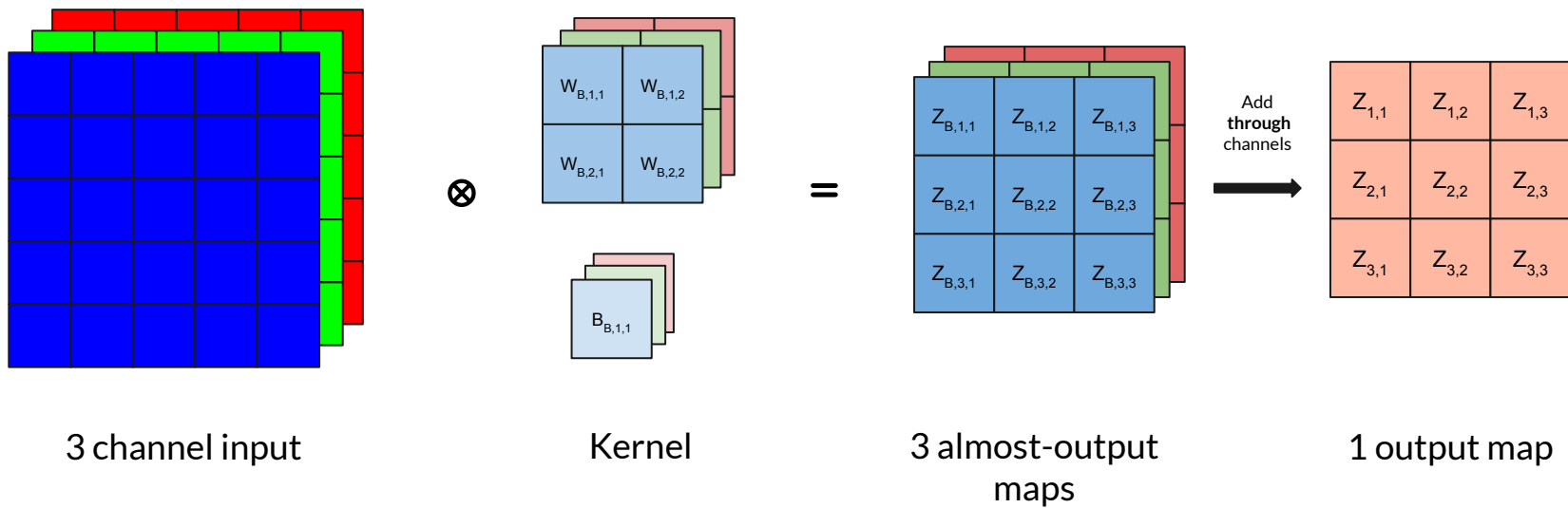
# 1 Filter with 3-channel input



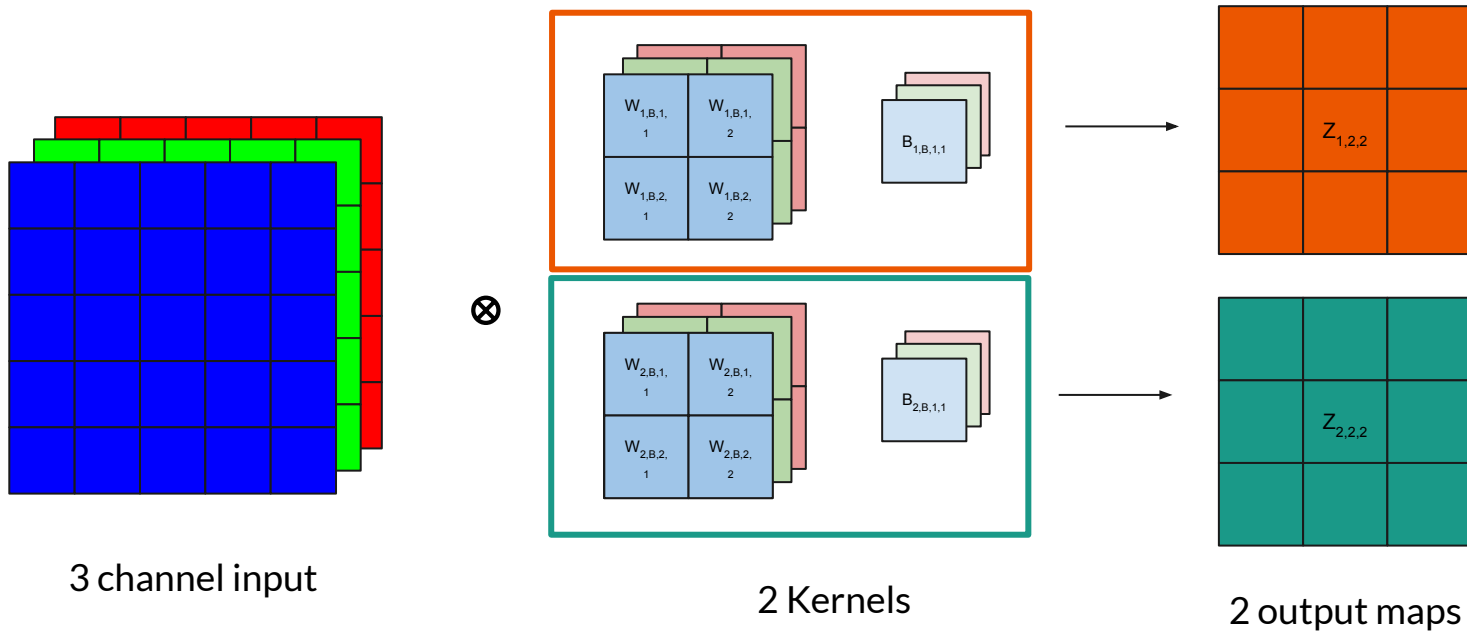
# 1 Filter with 3-channel input



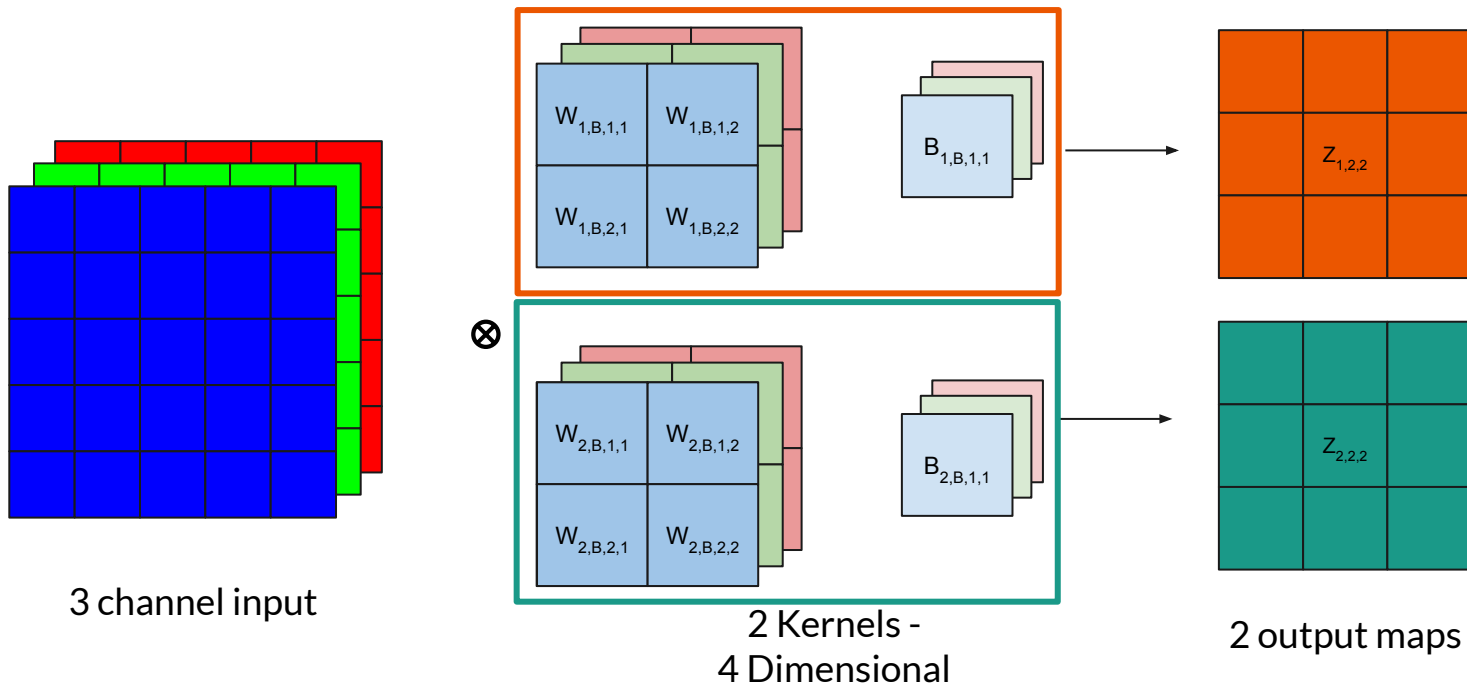
# 1 Filter with 3-channel input



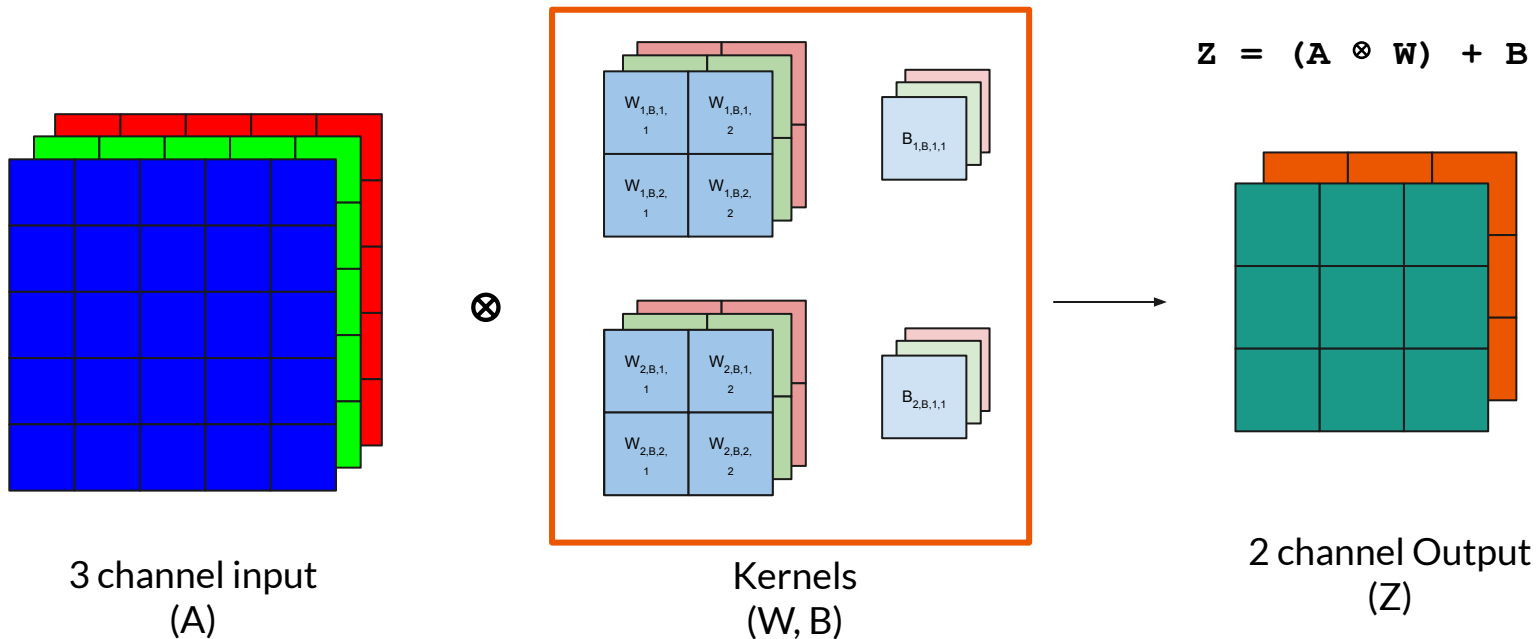
## 2 Filters with 3-channel input



## 2 Filters with 3-channel input



## 2 Filters with 3-channel input



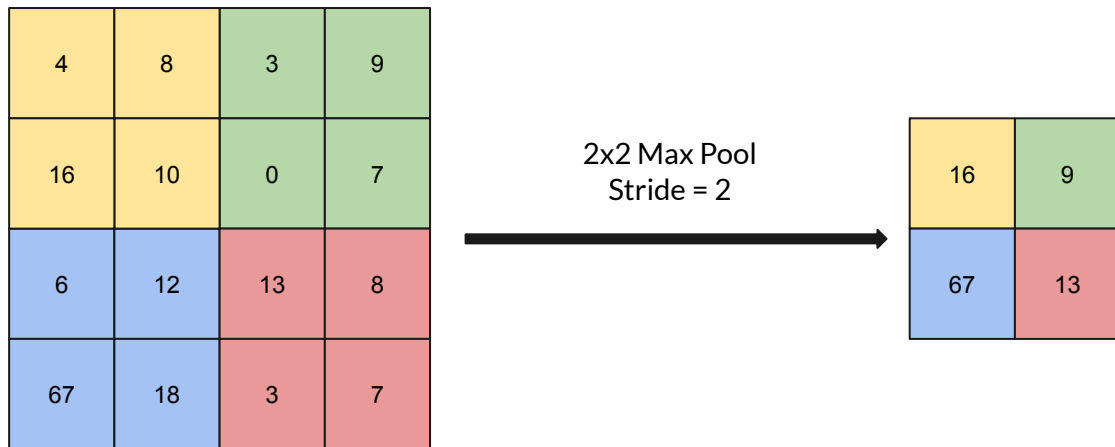


# Pooling

- Usually follows convolutions
- Introduces Jitter Invariance
- Reduces feature-map size
- **Max, Mean, Min**

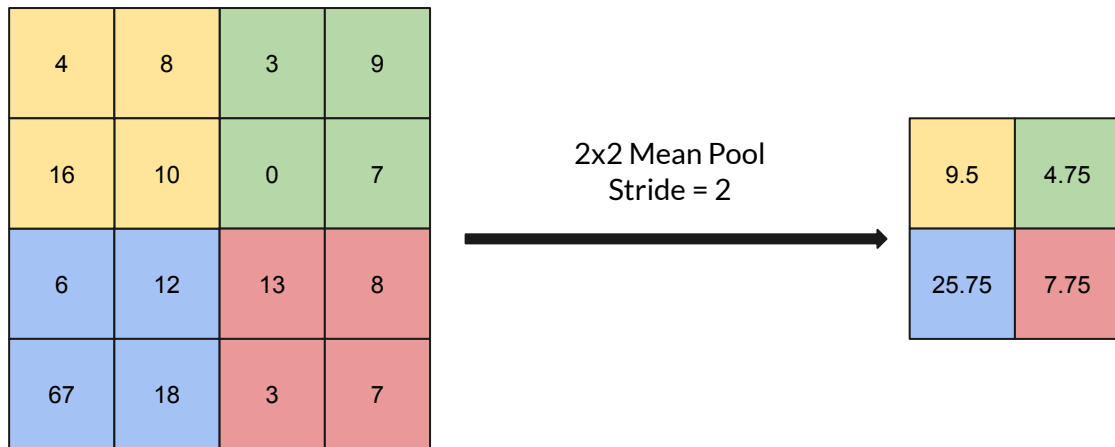


# Pooling





# Pooling





# Pooling

- Usually follows convolutions
- Introduces Jitter Invariance
- Reduces feature-map size
- **Max, Mean, Min**
- What happens to the channels in pooling?



# Pooling

- Usually follows convolutions
- Introduces Jitter Invariance
- Reduces feature-map size
- **Max, Mean, Min**
- Pooling preserves number of channels



## Onto Backward...