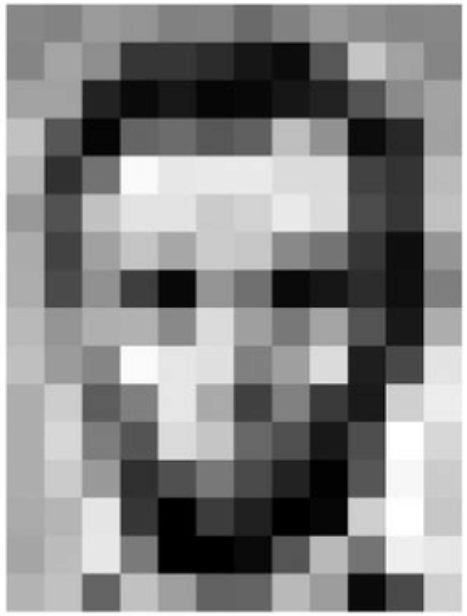
Convolutional Neural Networks (Forward)

11785 Introduction To Deep Learning (Spring 22) – Recitation 5

Aparajith Srinivasan

Image Basics

- Images are nothing but a function of Intensity values *f(x, y)* = Intensity at (x, y) pixel



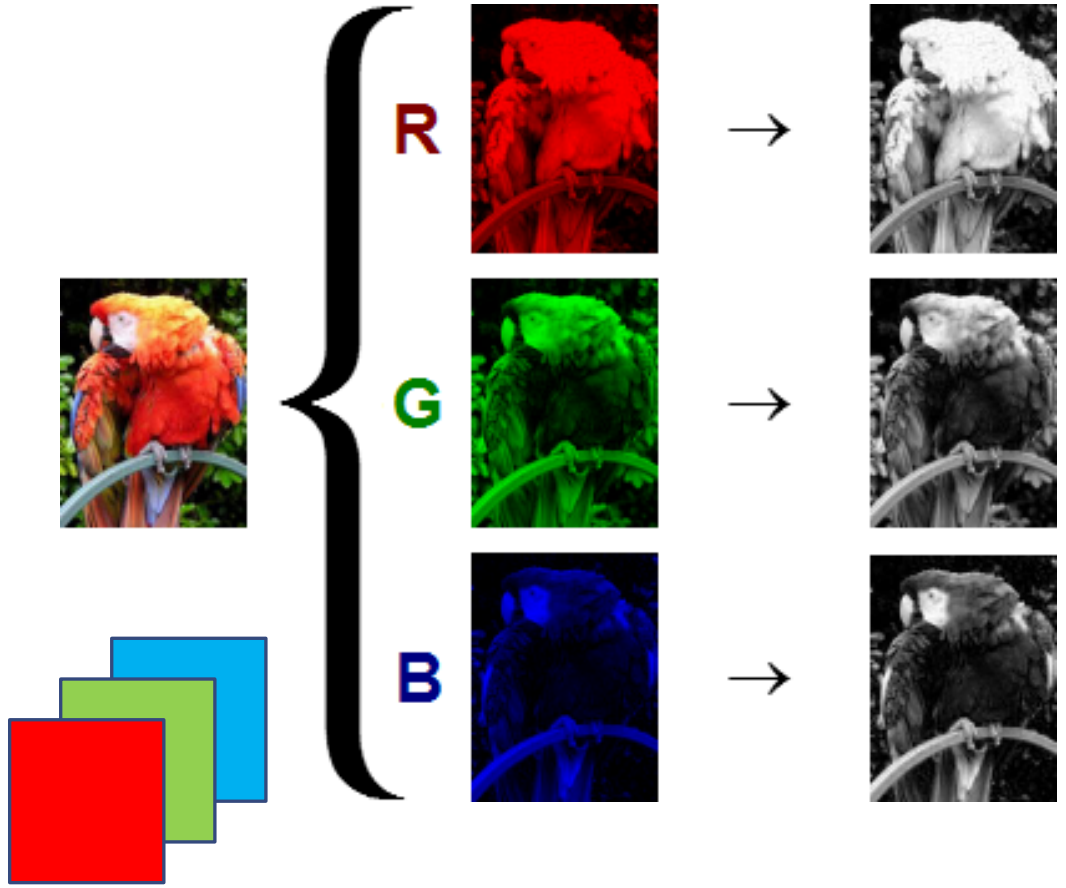
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	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	86	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	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	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	86	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

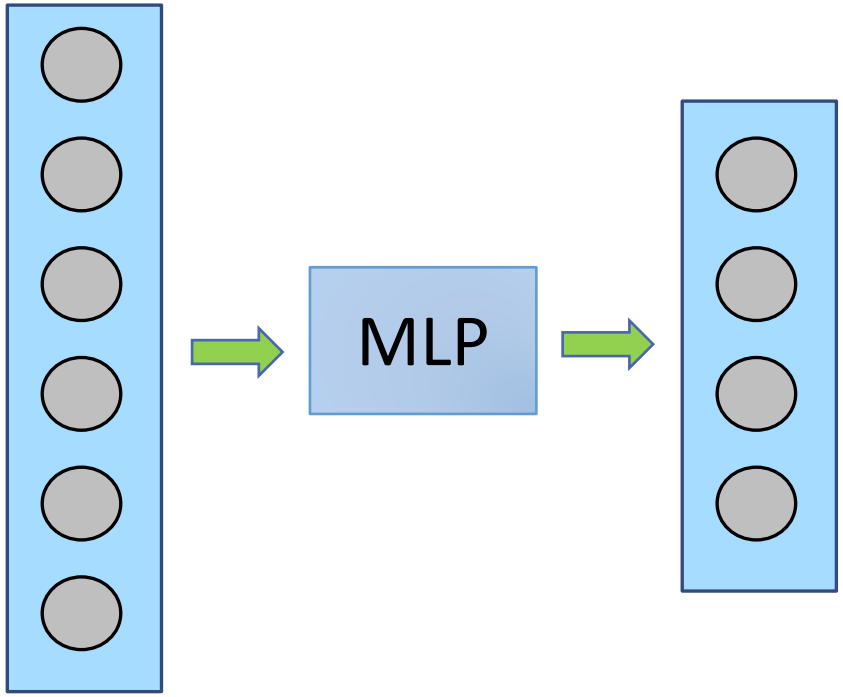
- Single channel – single 2D Image (M x N) – grayscale

Image Basics

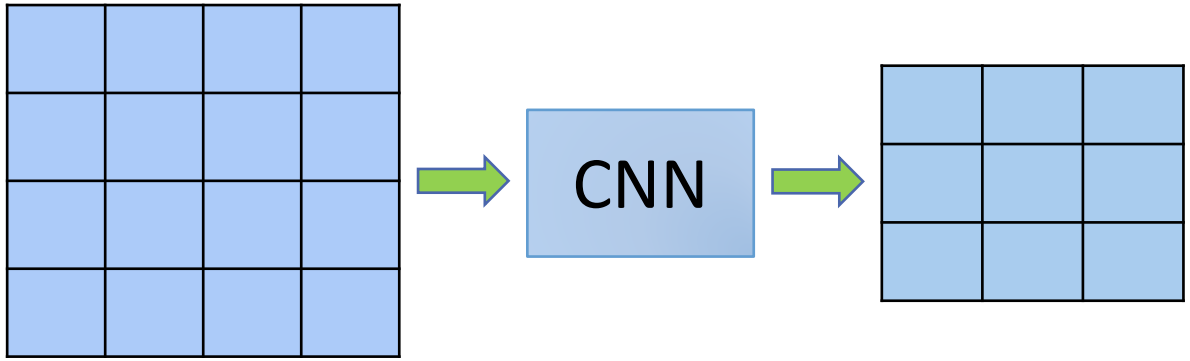
- Multichannel images are formed by stacking many grayscale images together
- Color image:
  - 3 Channels RGB
  - Shape: (3 x M x N) or (M x N x 3)



MLP vs CNN



Feature vector to feature vector



Feature map to feature map

CNN Components

- Any NN will have an input and weights (also bias)
- Consider the components for a single channel input

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

Input
A

w11	w12
w21	w22

Kernel
W

b1

Bias
b

CNN Components

- Any NN will have an input and weights (also bias)
- Consider the components for a single channel input

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

Input
A

w11	w12
w21	w22

Kernel
W

b1

Bias
b

$$Z = A \otimes W + b$$

CNN Components

- Any NN will have an input and weights (also bias)
- Consider the components for a single channel input

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

Input
A

w11	w12
w21	w22

Kernel
W

b1

Bias
b

$$Z = A \otimes W + b$$

What is this $\otimes$? 🤔

CNN Steps

- Nothing but an element wise product and summation

CNN Steps

- Nothing but an element wise product and summation

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

w11	w12
w21	w22

z’11		

z’11 = x11*w11 + x12*w12 + x13*w21 + x22*w22

CNN Steps

- Nothing but an element wise product and summation

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

w11	w12
w21	w22

z’11	z’12	

z’12 = x12*w11 + x13*w12 + x22*w21 + x23*w22

CNN Steps

- Nothing but an element wise product and summation

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

w11	w12
w21	w22

z'11	z'12	z'13

$$z'13 = x13*w11 + x14*w12 + x23*w21 + x24*w22$$

CNN Steps

- Nothing but an element wise product and summation

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

w11	w12
w21	w22

z'11	z’12	z’13
z’21		

z’21 = x21*w11 + x22*w12 + x31*w21 + x32*w22

CNN Steps

- Nothing but an element wise product and summation

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

w11	w12
w21	w22

z’11	z’12	z’13
z’21	z’22	

z’22 = x22*w11 + x23*w12 + x32*w21 + x33*w22

CNN Steps

- Nothing but an element wise product and summation

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

w11	w12
w21	w22

z'11	z'12	z'13
z'21	z'22	z'23
z'31	z'32	z'33

$$z'33 = x33*w11 + x34*w12 + x43*w21 + x44*w22$$

CNN Steps

- Nothing but an element wise product and summation

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

w11	w12
w21	w22

z'11	z'12	z'13
z'21	z'22	z'23
z'31	z'32	z'33

$z'33 = x33*w11 + x34*w12 + x43*w21 + x44*w22$

Kernel crosses 1 pixel at each step, stride = 1

CNN Steps

- Nothing but an element wise product and summation

z'11	z'12	z'13
z'21	z'22	z'23
z'31	z'32	z'33

b1

$z_{ij} = z'_{ij} + b1$

z11	z12	z13
z21	z22	z23
z31	z32	z33

Final step is to add bias to all elements

CNN Steps

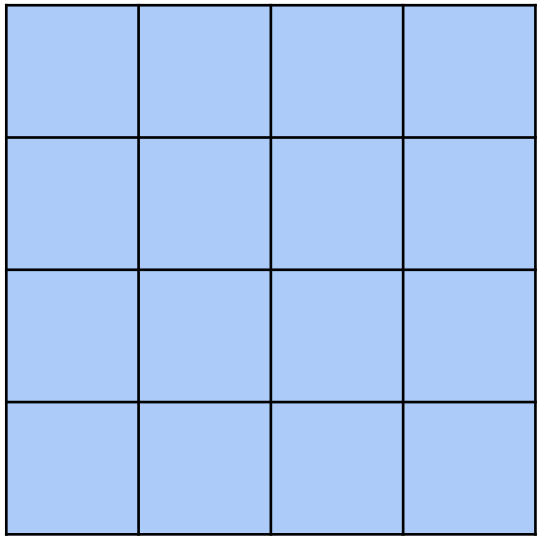
- Note is that the output map size is decreased

CNN Steps

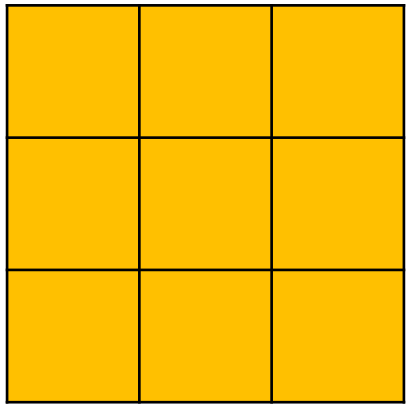
- Note is that the output map size is decreased
- Output size = (Input_size – Kernel_size)//stride + 1

CNN Steps

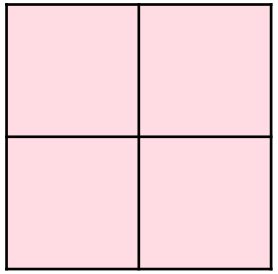
- Note is that the output map size is decreased
- Output size = (Input_size – Kernel_size)//stride + 1



Input (4x4)



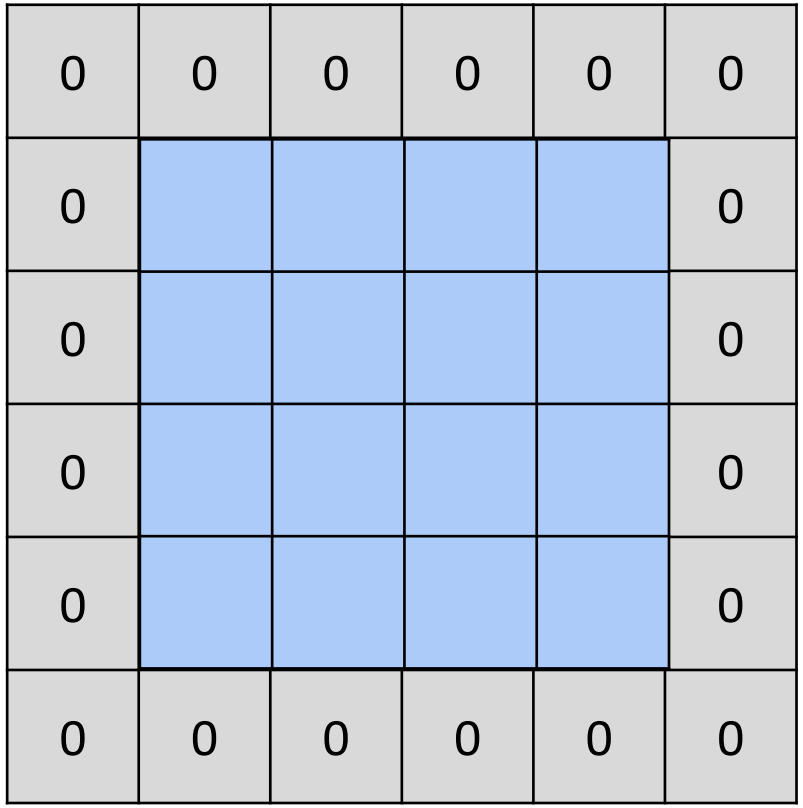
Kernel (3x3)



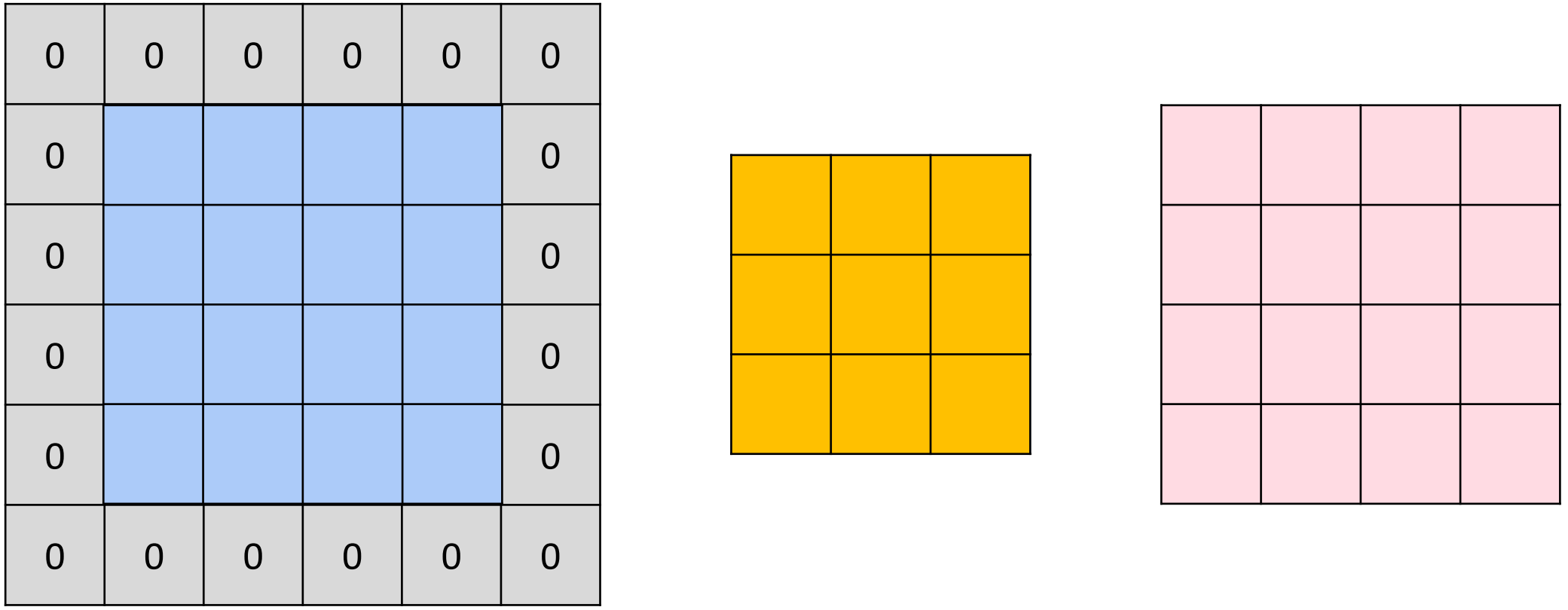
Output (2x2)
(stride = 1)

CNN Steps

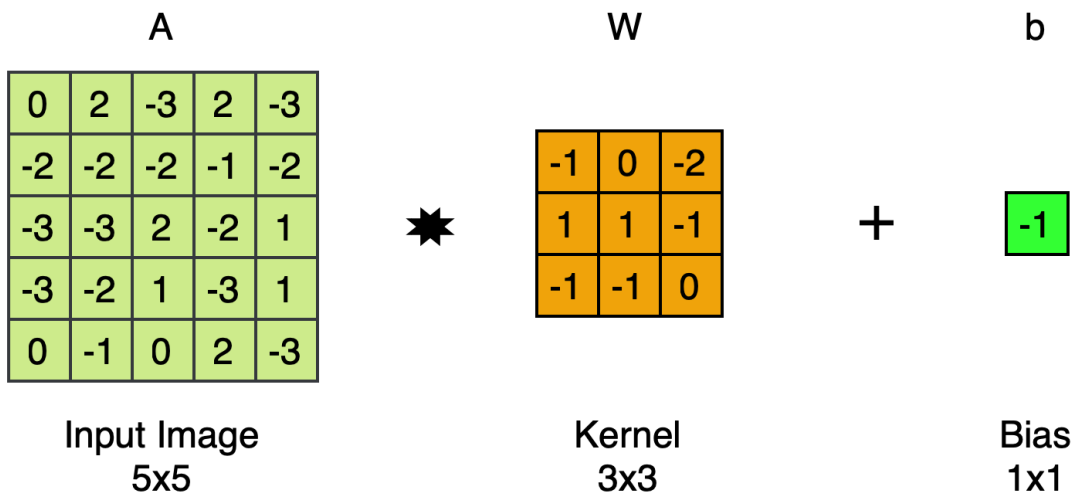
- Note is that the output map size is decreased
- Output size = (Input_size – Kernel_size)//stride + 1
- Pad and convolve to conserve size



CNN Steps



CNN stride 2 Example

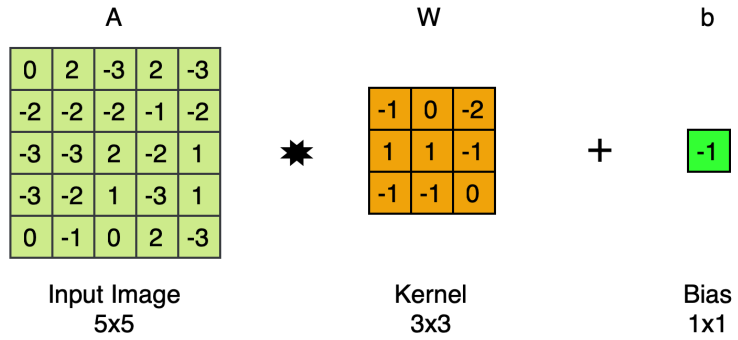


CNN stride 2 Example

0	2	-3	2	-3
-2	-2	-2	-1	-2
-3	-3	2	-2	1
0	-1	0	2	-3
-3	-2	1	-3	1

10	

Step 1

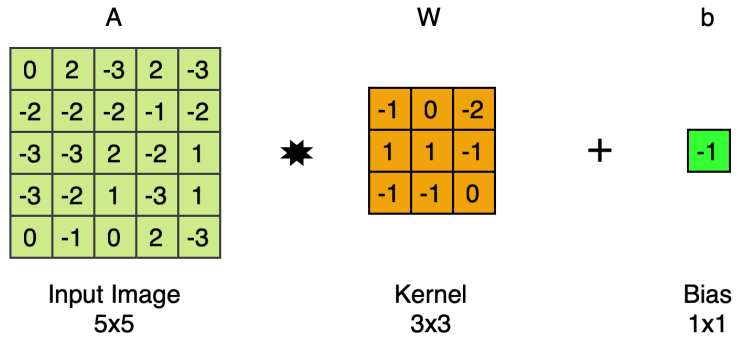


CNN stride 2 Example

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

10	8

Step 2

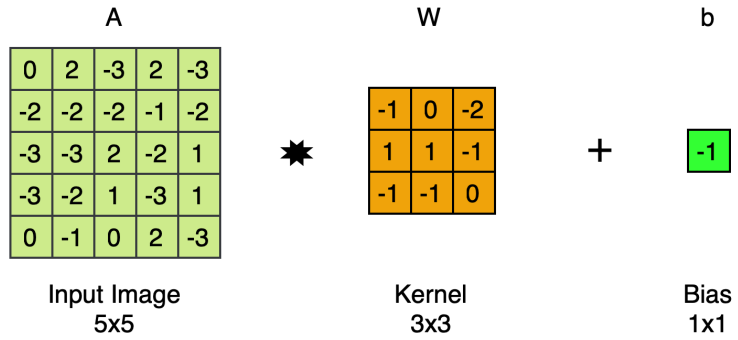


CNN stride 2 Example

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

10	8
-6	

Step 3

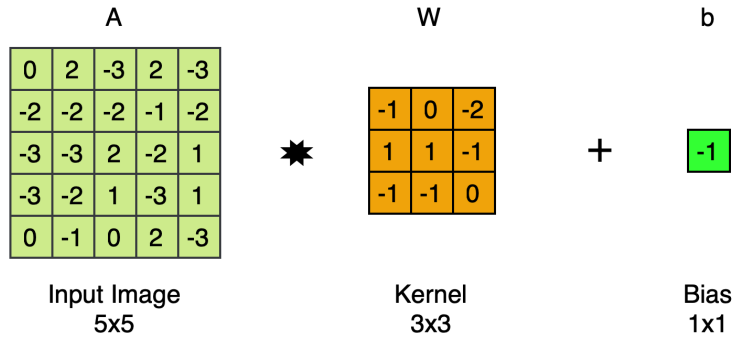


CNN stride 2 Example

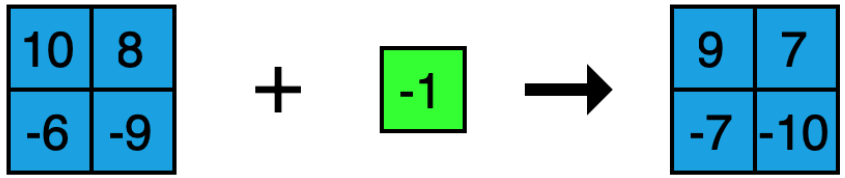
0	2	-3	2	-3
-2	-2	-2	-1	-2
-3	-3	2	-2	1
-3	-2	1	-3	1
0	-1	0	2	-3

10	8
-6	-9

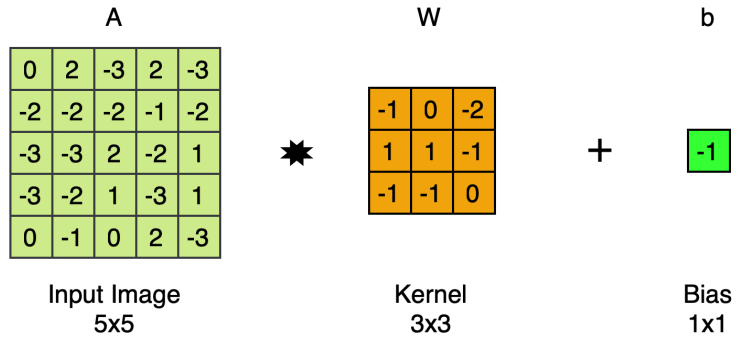
Step 4



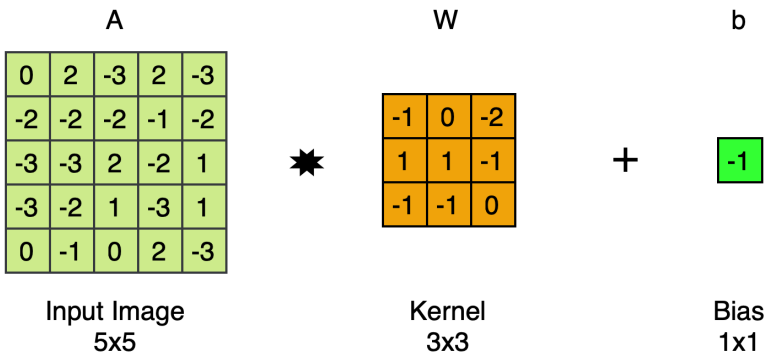
CNN stride 2 Example



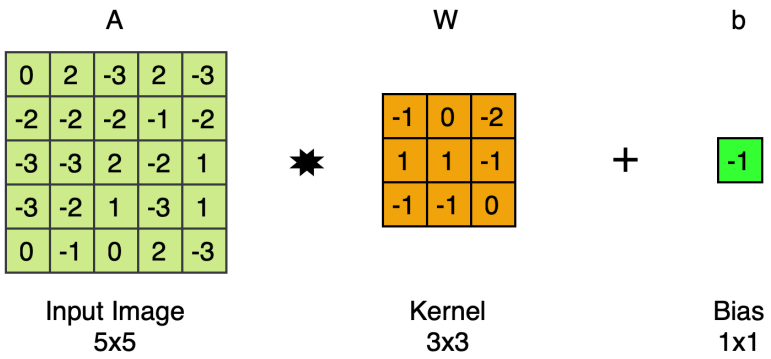
Step 5



Interpreting stride > 1



Interpreting stride > 1



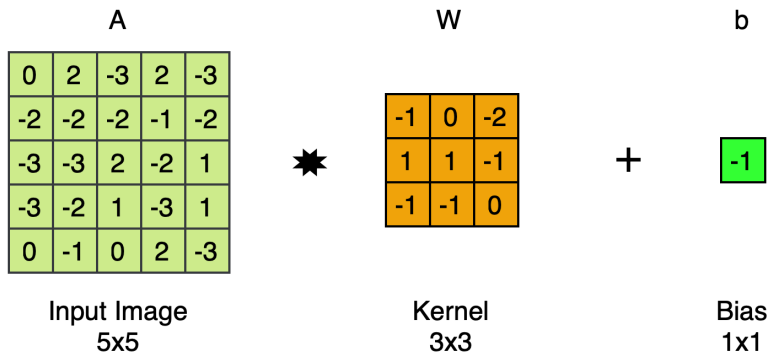
9	-9	7
2	5	6
-7	9	-10

Stride 1 output

9	7
-7	-10

Stride 2 output

Interpreting stride > 1



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

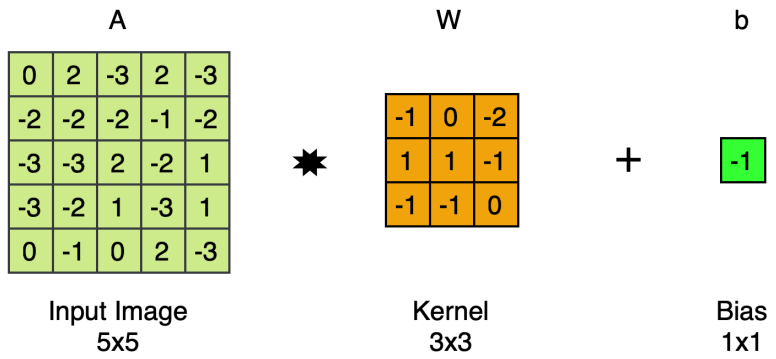
Stride 1 output

????

9	7
-7	-10

Stride 2 output

Interpreting stride > 1



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

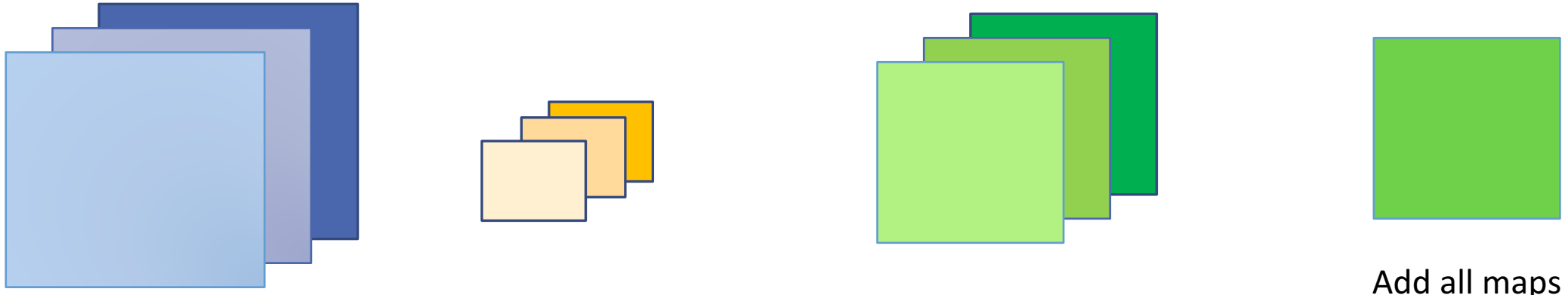
Stride 2 output

Multichannel CNN

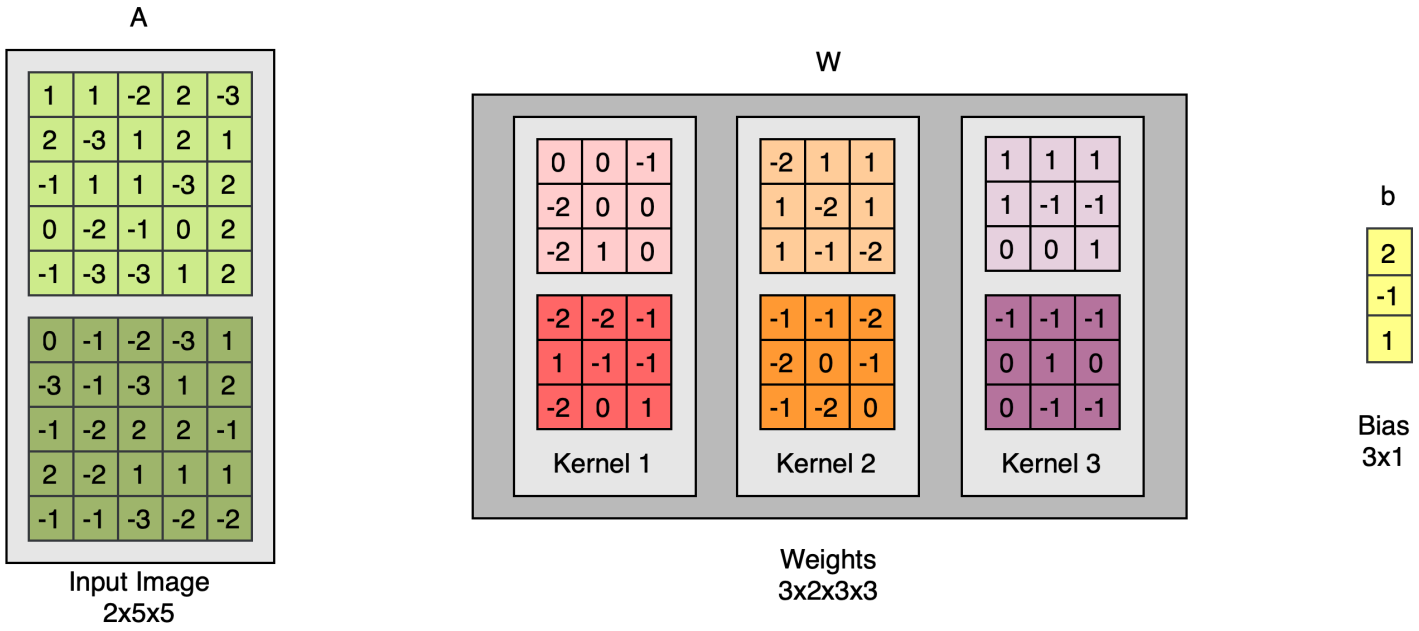
- When input has multiple channels, kernel has the same number of channels
- Each channel of the kernel convolves with the corresponding input channel to produce an output channel – all output maps obtained from this convolution are added together
- 1 filter produces 1 output channel. N filters produce N output channel

Multichannel CNN

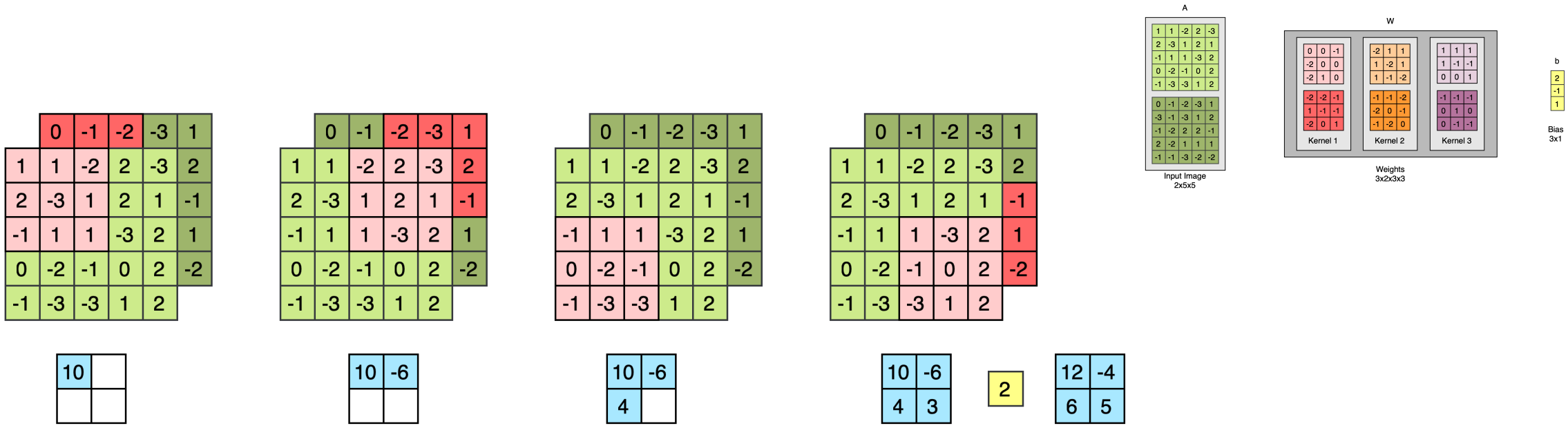
- When input has multiple channels, kernel has the same number of channels
- Each channel of the kernel convolves with the corresponding input channel to produce an output channel – all output maps obtained from this convolution are added together
- 1 filter produces 1 output channel. N filters produce N output channel



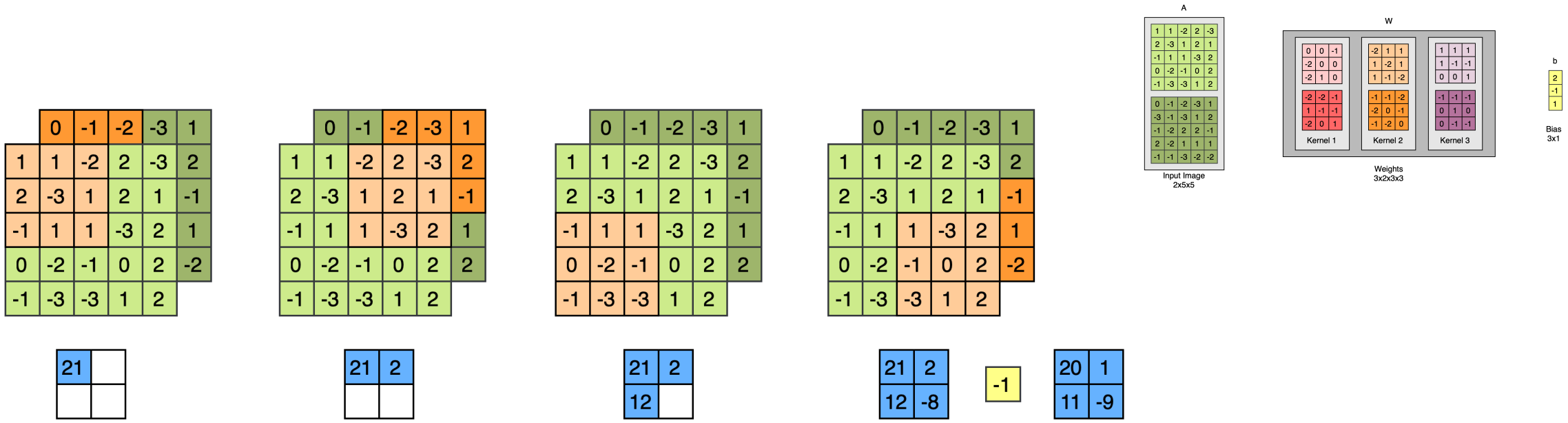
Multichannel CNN



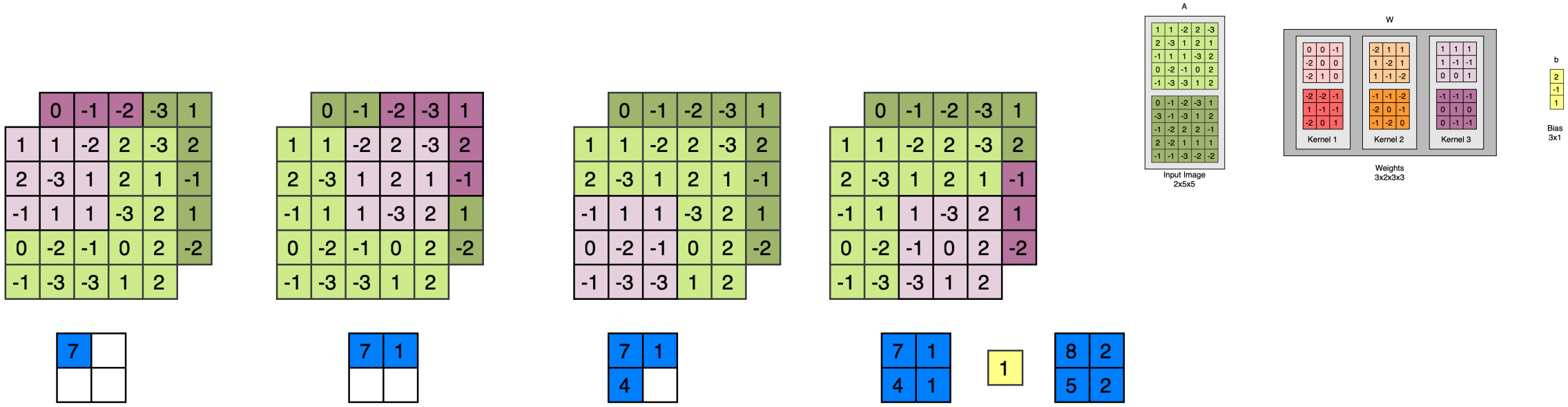
Multichannel CNN



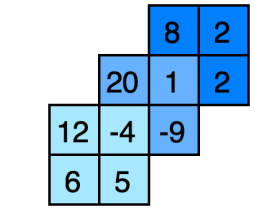
Multichannel CNN



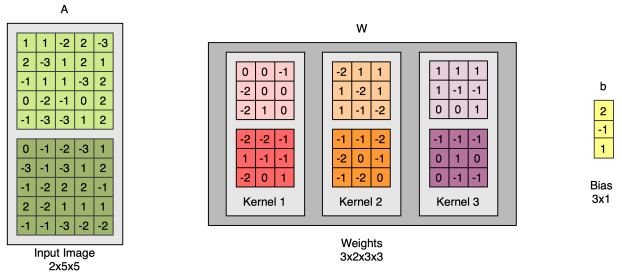
Multichannel CNN



Multichannel CNN



Output Response
3x2x2

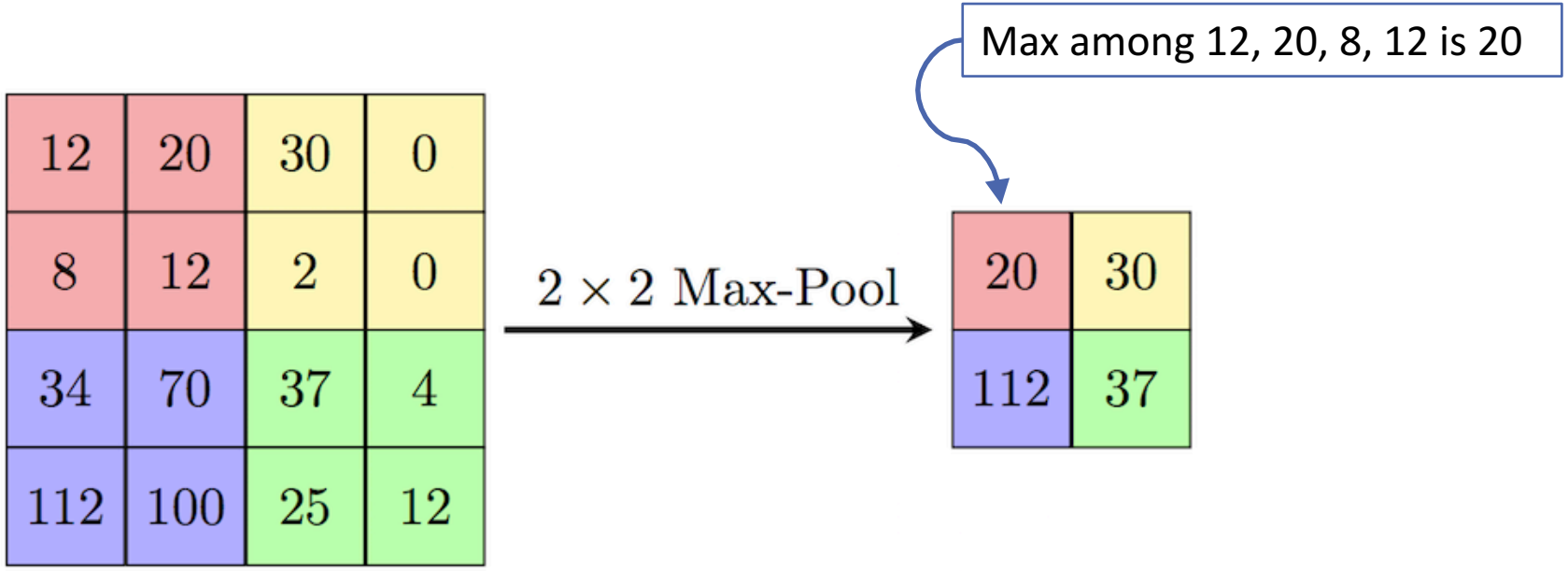


Pooling

- Step followed by convolution
- For jitter invariance (also downsamples if stride > 1)

Pooling

- Max Pooling
  - Kernel_Size = 2x2; stride = 2



Pooling

- Mean Pooling
  - Kernel_Size = 2x2; stride = 2