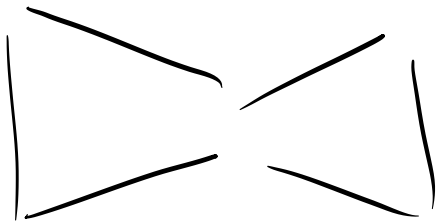
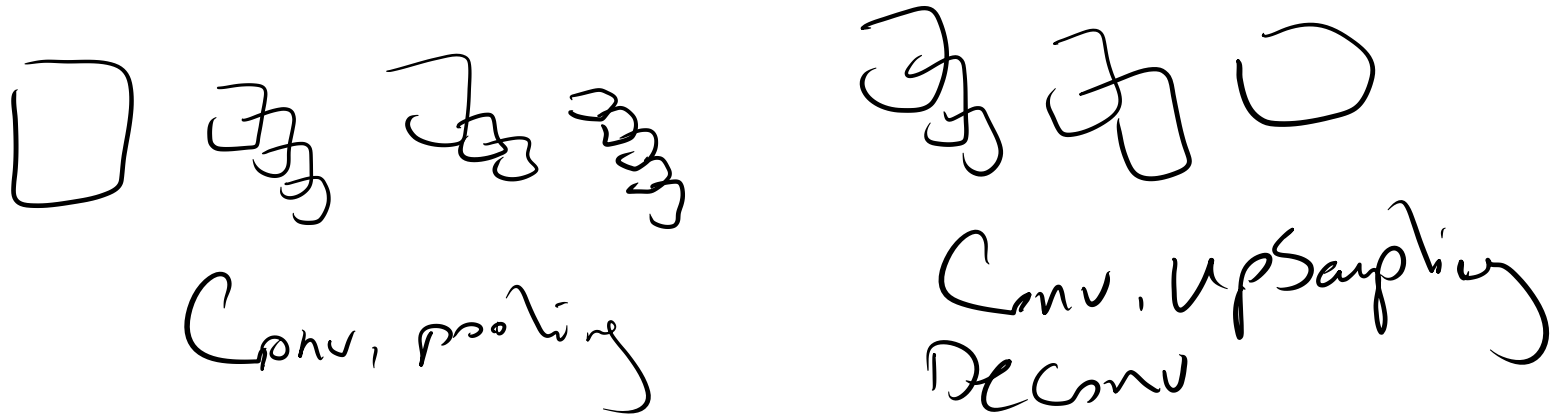
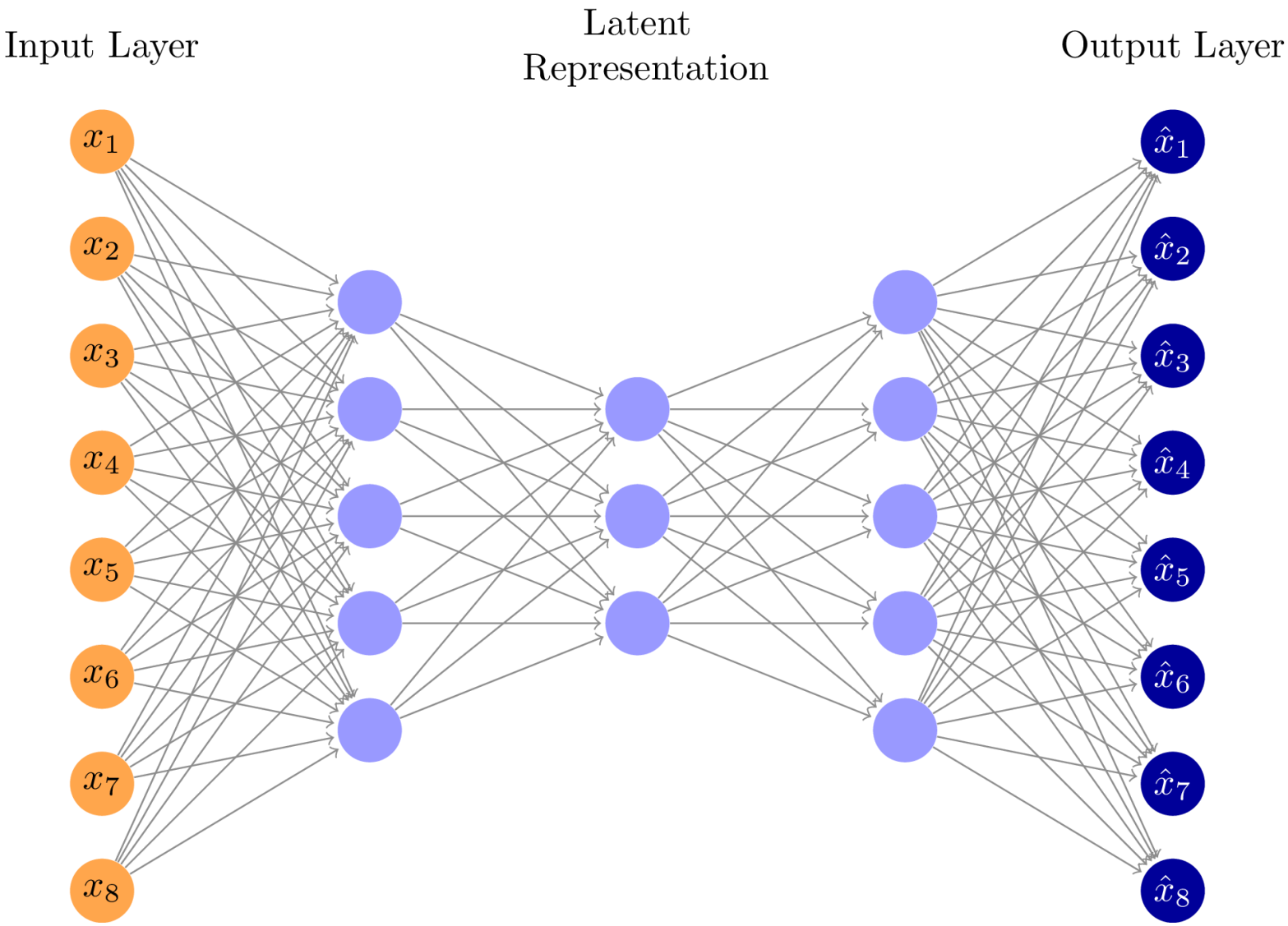
Encoder – Decoder

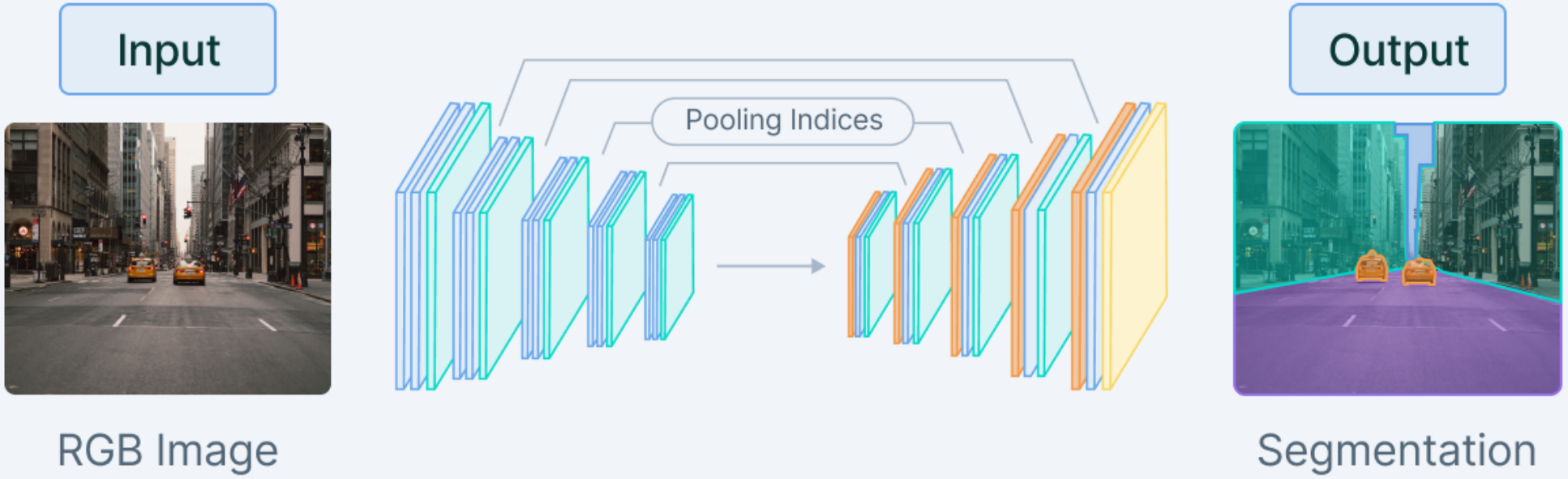
AutoEncoder





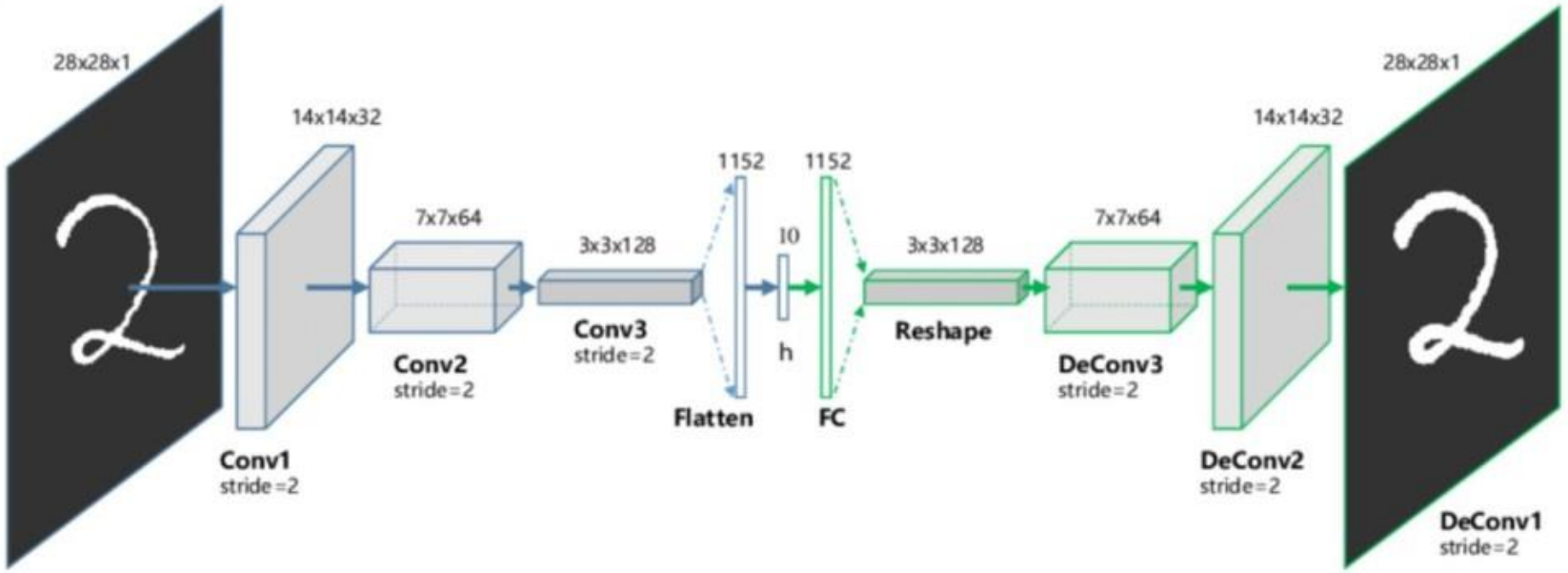

Input Layer
Latent Representation
Output Layer
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$
$\hat{x}_7$
$\hat{x}_8$

Convolutional encoder-decoder



V7 Labs

What is Convolutional Autoencoder ?



$$I * F = G$$

filter

$$I_f \cdot F_f = G_f$$

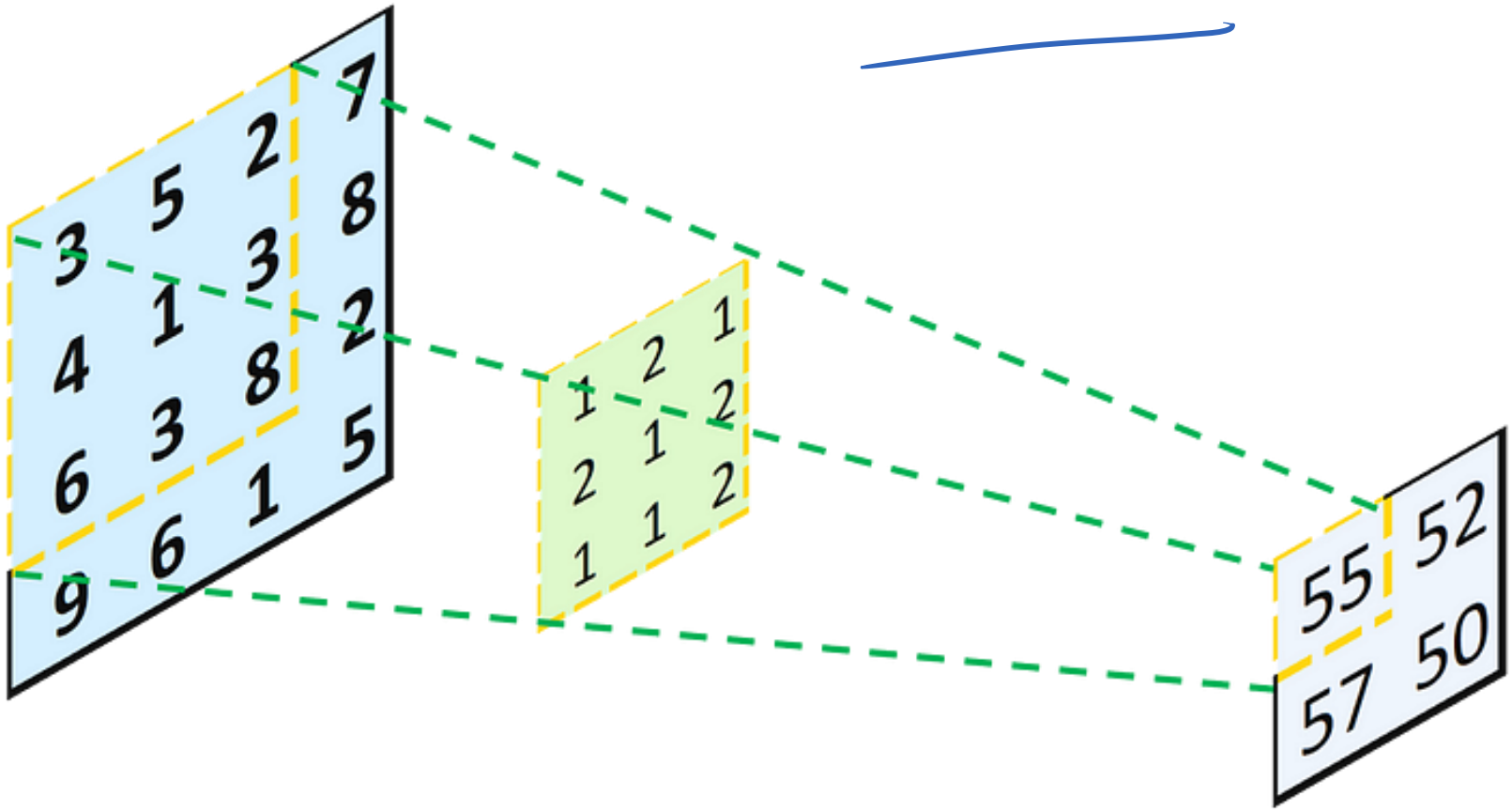
$$I_f = \frac{G_f}{F_f}$$

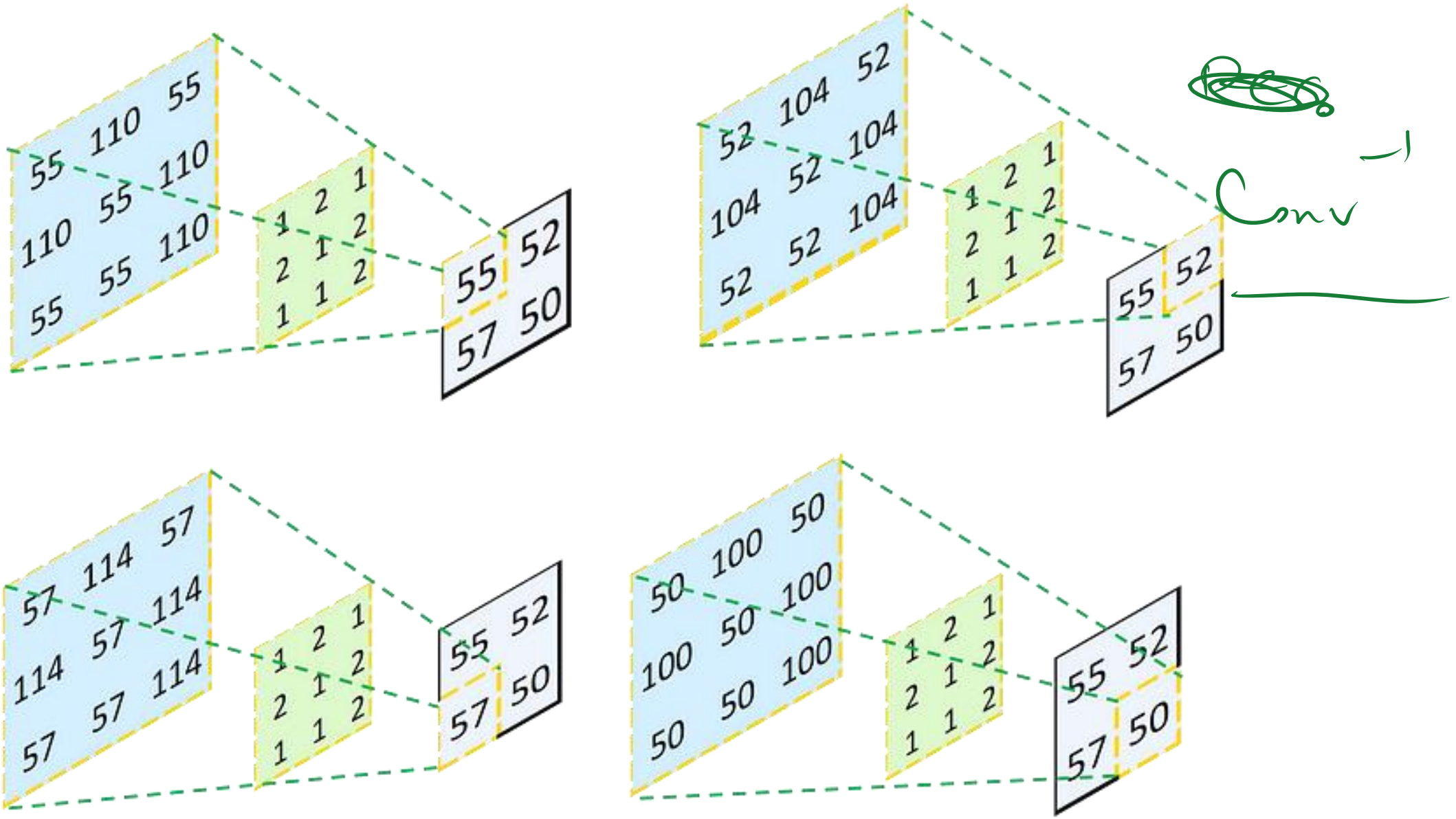
I

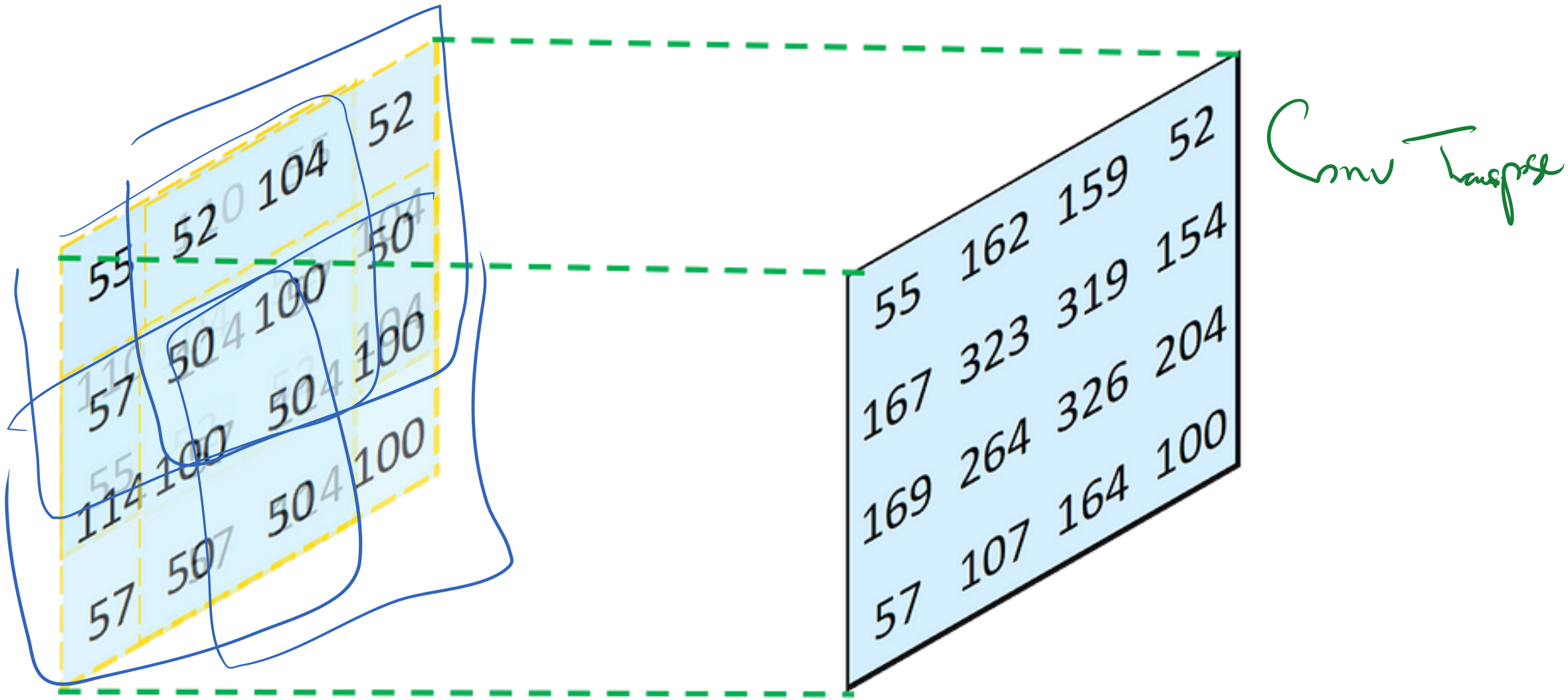
Conv Transpose

DeConv

Conv.



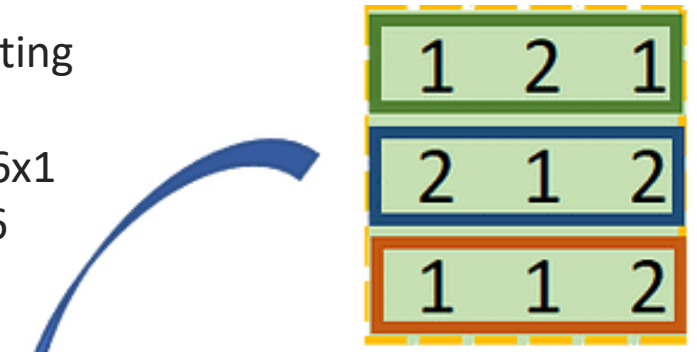

55 110 55
55 110 110
55 55 110
110 55 110
1 2 1
2 1 2
1 1 2
55 52
57 50
52 104 52
104 52 104
52 52 104
104 52 104
1 2 1
2 1 2
1 1 2
55 52
57 50
57 114 57
114 57 114
57 57 114
114 57 114
1 2 1
2 1 2
1 1 2
55 52
57 50
50 100 50
100 50 100
50 50 100
100 50 100
1 2 1
2 1 2
1 1 2
55 52
57 50

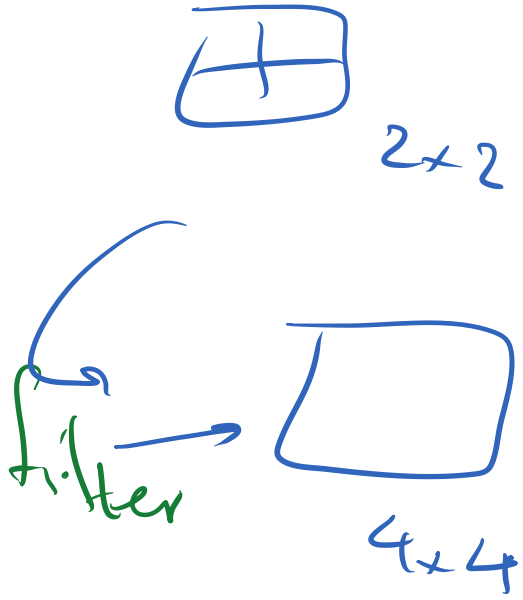

55 52 104 52
57 50 100 50
114 100 50 100
57 50 50 100
55 162 159 52
167 323 319 154
169 264 326 204
57 107 164 100
Conv Transpose

The reason it is 4x16 matrix is that:

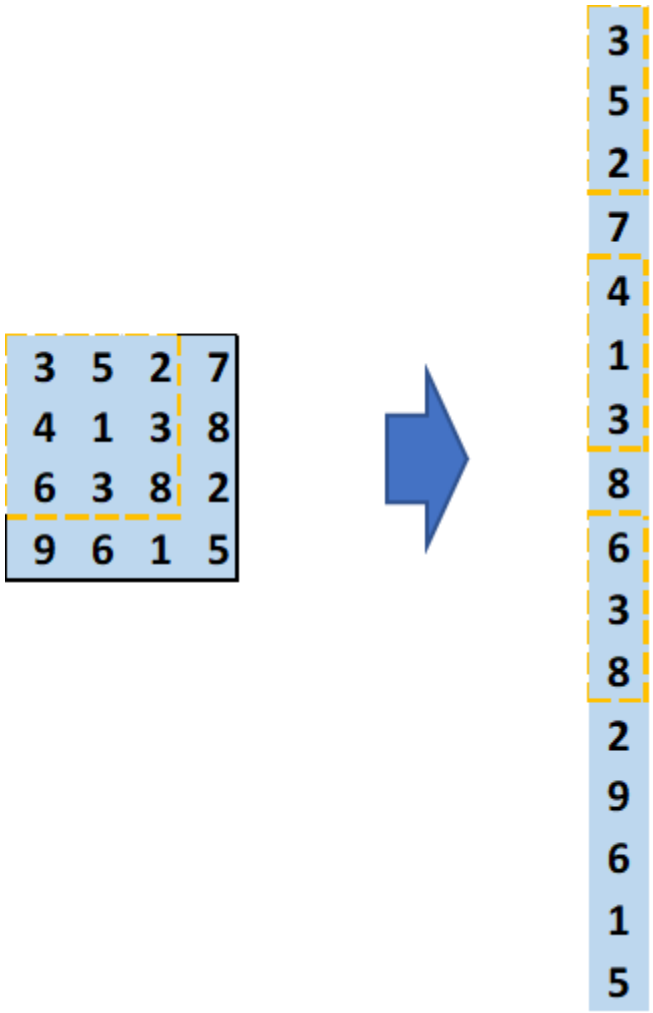
•4 rows: in total, we can perform four convolutions by splitting a 4x4 input matrix into four 3x3 matrices;

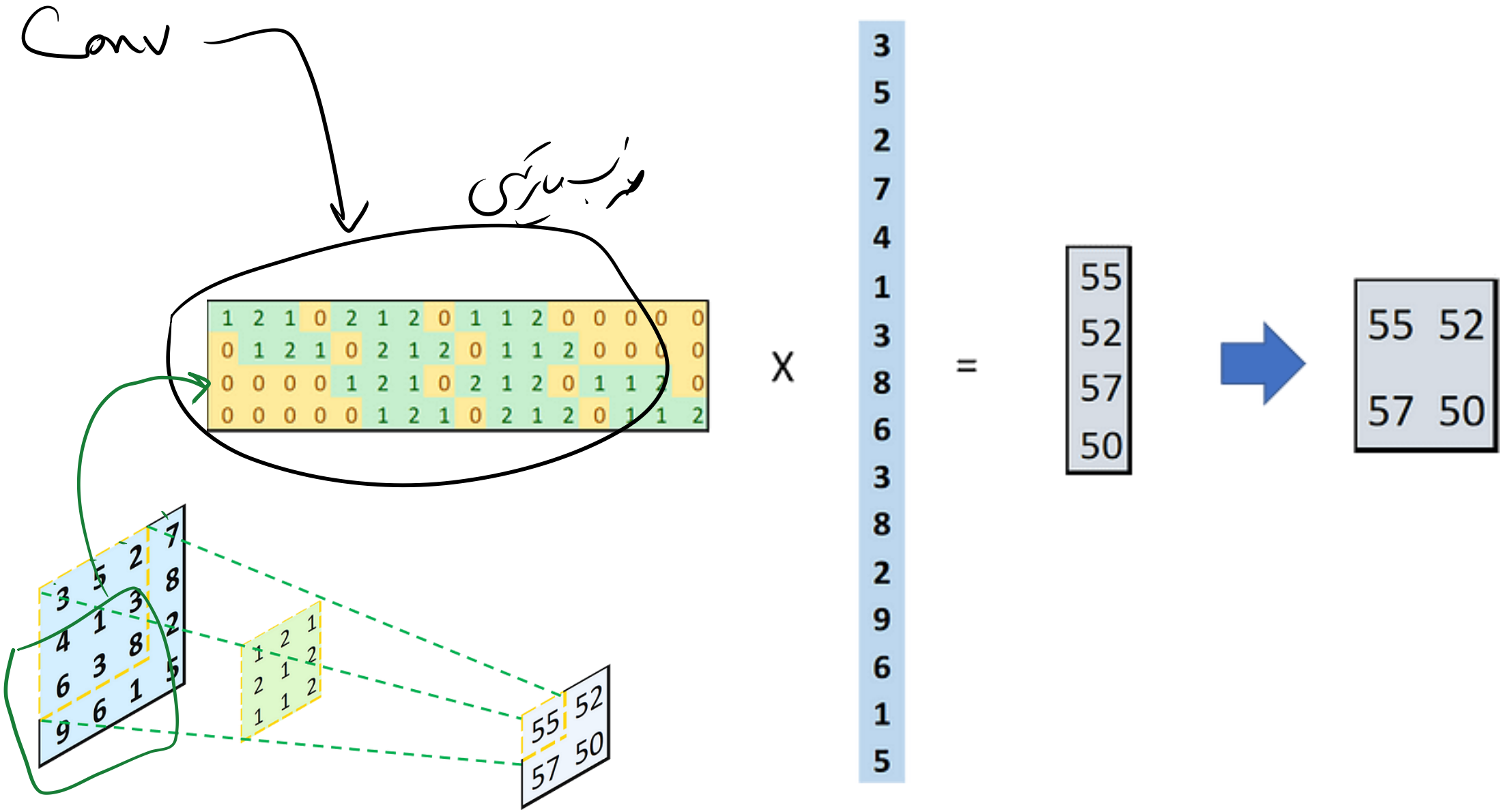
•16 columns: the input matrix will be transformed into a 16x1 vector. To perform the matrix multiplication, it has to be 16 columns.

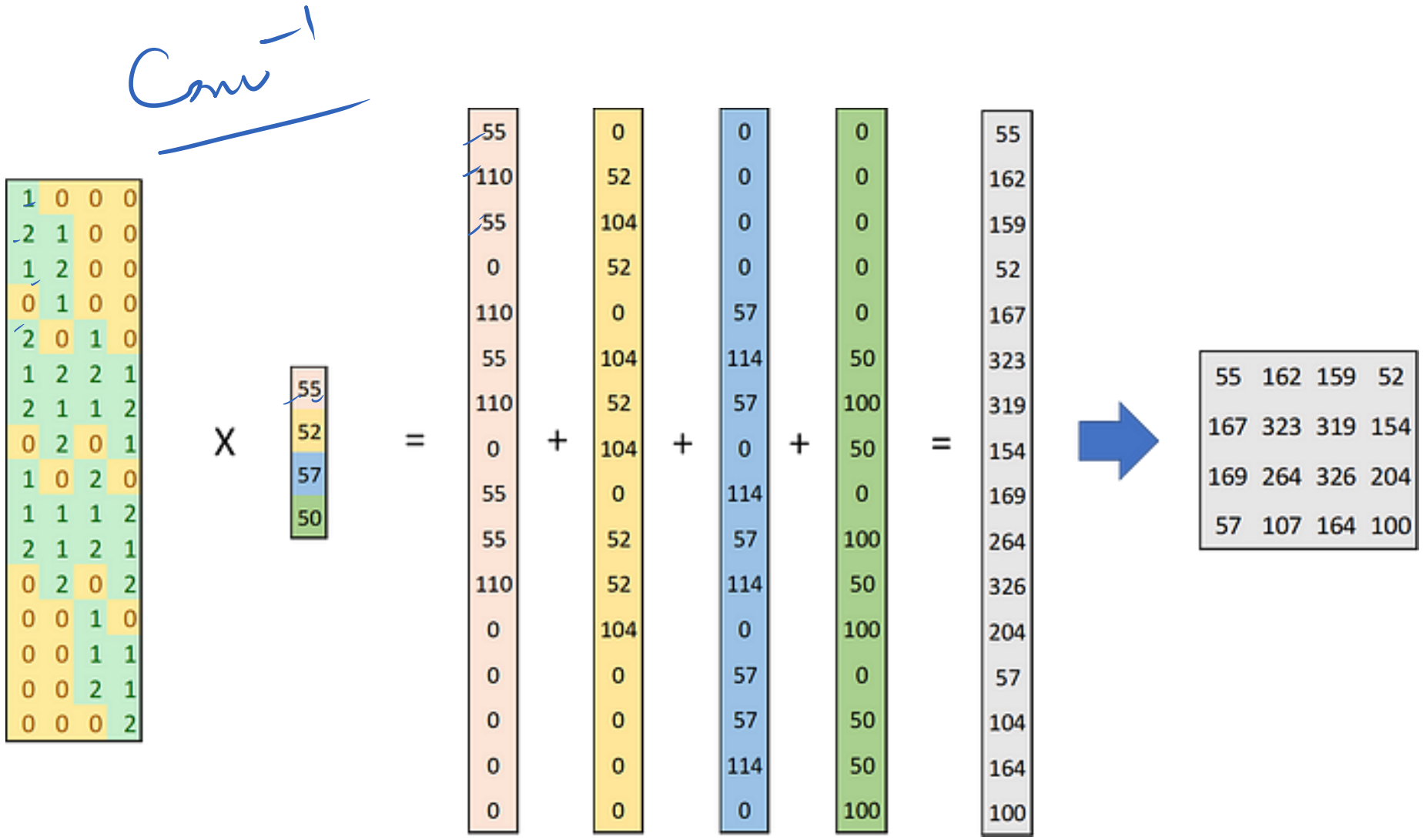


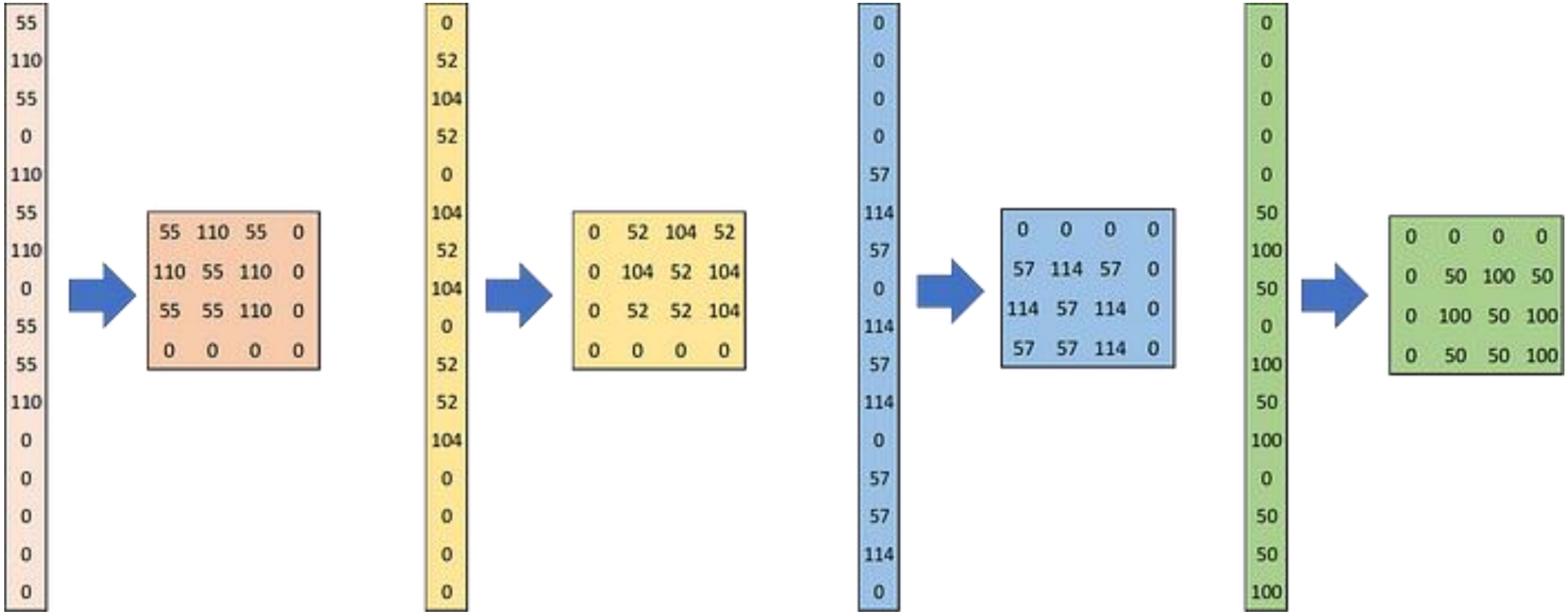


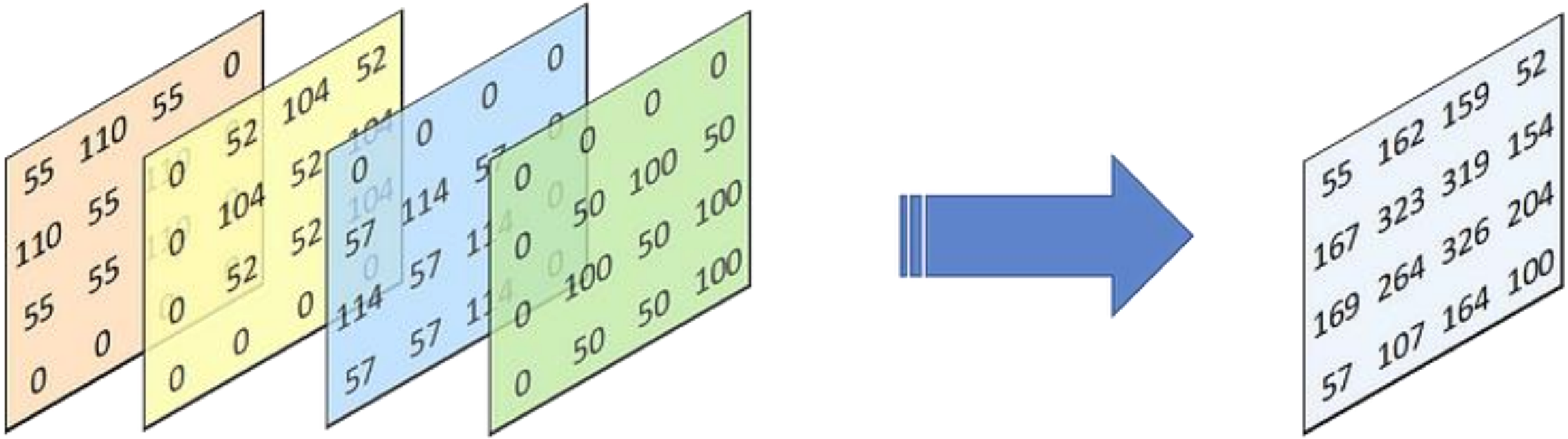
1	2	1	0	2	1	2	0	1	1	2	0	0	0	0	0
0	1	2	1	0	2	1	2	0	1	1	2	0	0	0	0
0	0	0	0	1	2	1	0	2	1	2	0	1	1	2	0
0	0	0	0	0	1	2	1	0	2	1	2	0	1	1	2

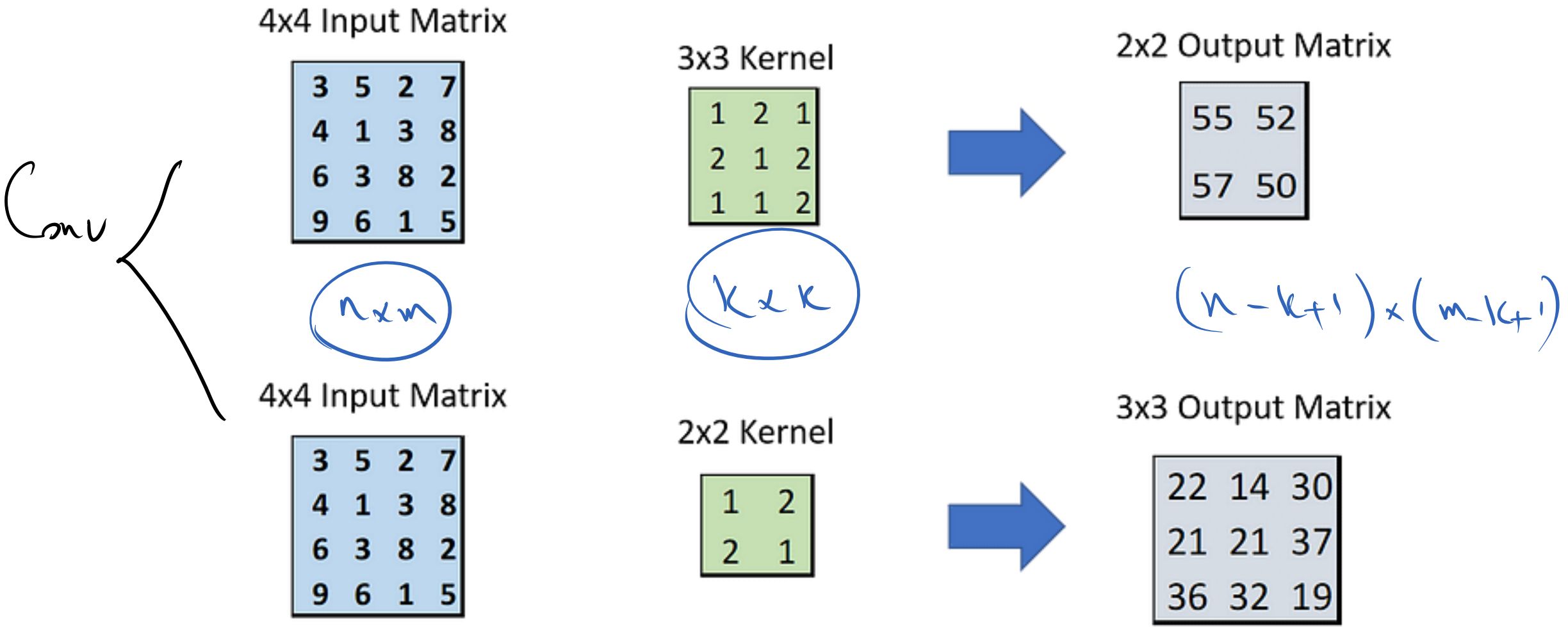

3 5 2 7
4 1 3 8
6 3 8 2
9 6 1 5
3
5
2
7
4
1
3
8
6
3
8
2
9
6
1
5


Conv
ضرب ماتریسی
1 2 1 0 2 1 2 0 1 1 2 0 0 0 0 0
0 1 2 1 0 2 1 2 0 1 1 2 0 0 0 0
0 0 0 0 1 2 1 0 2 1 2 0 1 1 2 0
0 0 0 0 0 1 2 1 0 2 1 2 0 1 1 2
X
3 5 2 7
4 1 3 8
6 3 8 2
9 6 1 5
1 2 1
2 1 2
1 1 2
55 52
57 50
=
55
52
57
50
55 52
57 50


Conv-1
1 0 0 0
2 1 0 0
1 2 0 0
0 1 0 0
2 0 1 0
1 2 2 1
2 1 1 2
0 2 0 1
1 0 2 0
1 1 1 2
2 1 2 1
0 2 0 2
0 0 1 0
0 0 1 1
0 0 2 1
0 0 0 2
X
55
52
57
50
=
55 110 55 0 110 55 110 0 55 55 110 0 0 0 0 0
+
0 52 104 52 0 104 52 104 0 52 52 104 0 0 0 0
+
0 0 0 0 57 114 57 0 114 57 114 0 57 57 114 0
+
0 0 0 0 0 50 100 50 0 100 50 100 0 50 50 100
=
55 162 159 52 167 323 319 154 169 264 326 204 57 104 164 100
55 162 159 52
167 323 319 154
169 264 326 204
57 107 164 100


55
110
55
0
110
55
110
0
55
55
110
0
0
0
0
0
55 110 55 0
110 55 110 0
55 55 110 0
0 0 0 0
0
52
104
52
0
104
52
104
0
52
52
104
0
0
0
0
0 52 104 52
0 104 52 104
0 52 52 104
0 0 0 0
0
0
0
0
57
114
57
0
114
57
114
0
57
57
114
0
0 0 0 0
57 114 57 0
114 57 114 0
57 57 114 0
0
0
0
0
0
50
100
50
0
100
50
100
0
50
50
100
0 0 0 0
0 50 100 50
0 100 50 100
0 50 50 100


55 110 55 0
110 55 0 0
55 55 0 0
0 0 0 0
0 52 104 52
0 104 52 52
0 52 0 0
0 0 0 0
0 0 0 0
57 114 57 114
114 57 57 114
57 57 114 0
0 0 0 0
0 50 100 50
0 100 50 100
0 50 50 100
55 162 159 52
167 323 319 154
169 264 326 204
57 107 164 100


4x4 Input Matrix
3 5 2 7
4 1 3 8
6 3 8 2
9 6 1 5
n x m
3x3 Kernel
1 2 1
2 1 2
1 1 2
k x k
2x2 Output Matrix
55 52
57 50
(n-k+1) x (m-k+1)
Conv
4x4 Input Matrix
3 5 2 7
4 1 3 8
6 3 8 2
9 6 1 5
2x2 Kernel
1 2
2 1
3x3 Output Matrix
22 14 30
21 21 37
36 32 19

Conv-1

2x2 Input Matrix

55	52
57	50

$n \times m$

3x3 Kernel

1	2	1
2	1	2
1	1	2

$K \times K$

4x4 Output Matrix

55	162	159	52
167	323	319	154
169	264	326	204
57	107	164	100

$(n+K-1) \times (m+k-1)$

2x2 Input Matrix

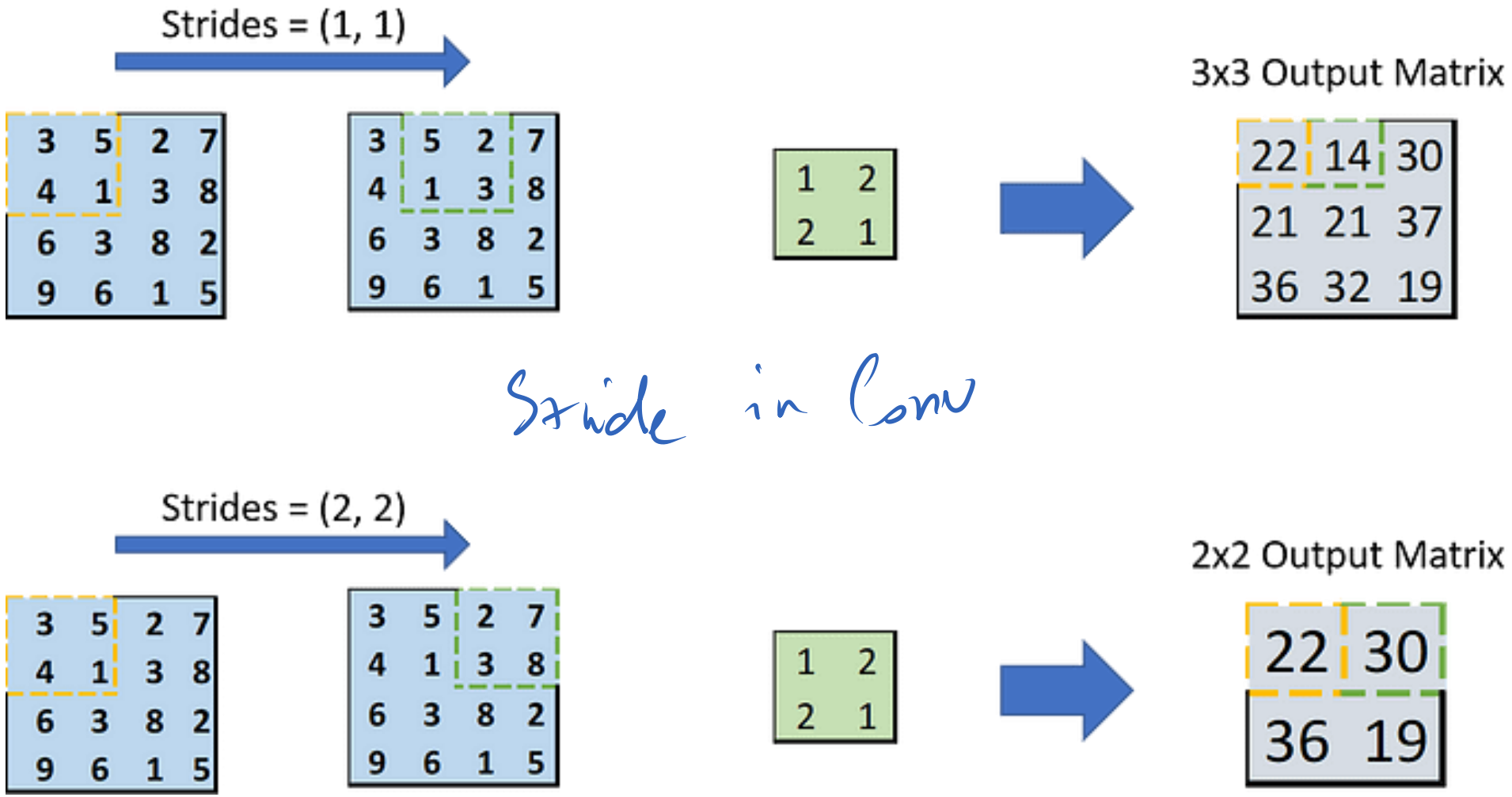
55	52
57	50

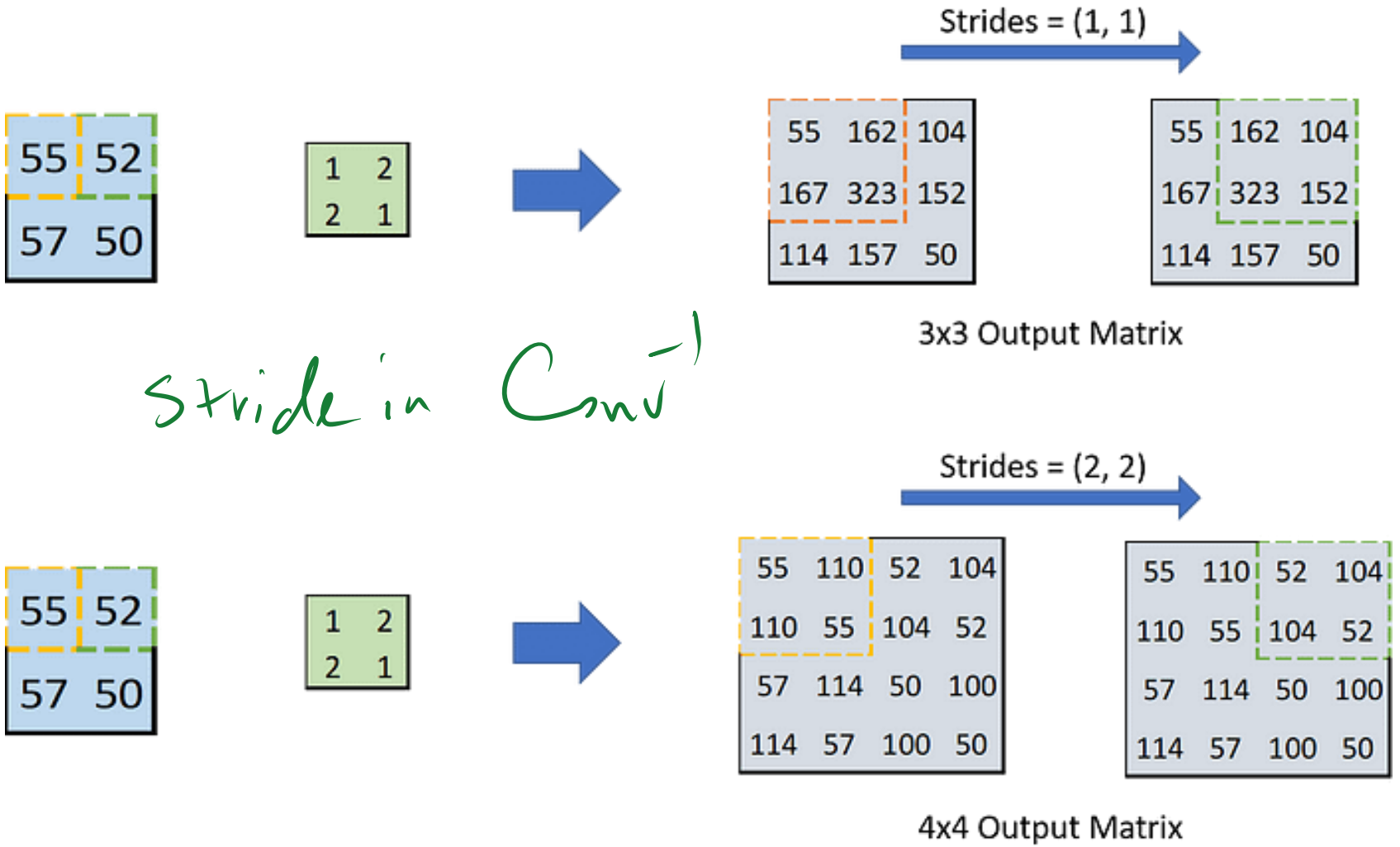
2x2 Kernel

1	2
2	1

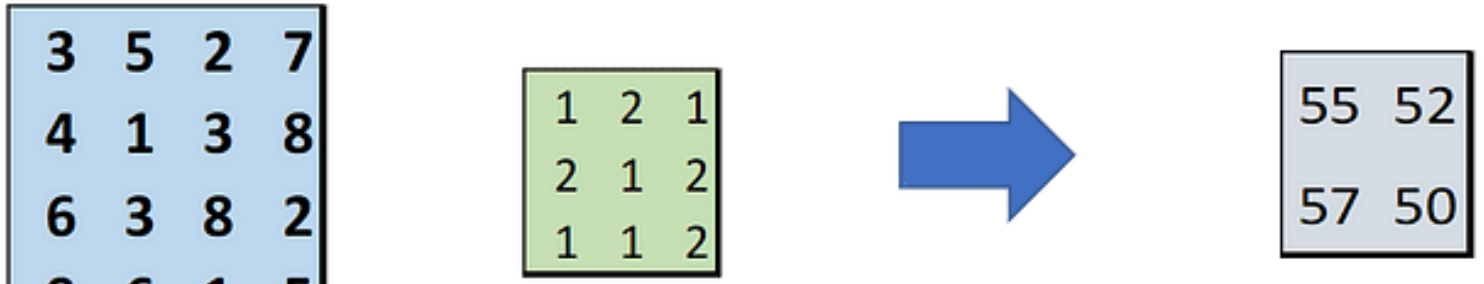
3x3 Output Matrix

55	162	104
167	323	152
114	157	50


Strides = (1, 1)
3 5 2 7
4 1 3 8
6 3 8 2
9 6 1 5
1 2
2 1
3x3 Output Matrix
22 14 30
21 21 37
36 32 19
Stride in Conv
Strides = (2, 2)
2x2 Output Matrix
22 30
36 19

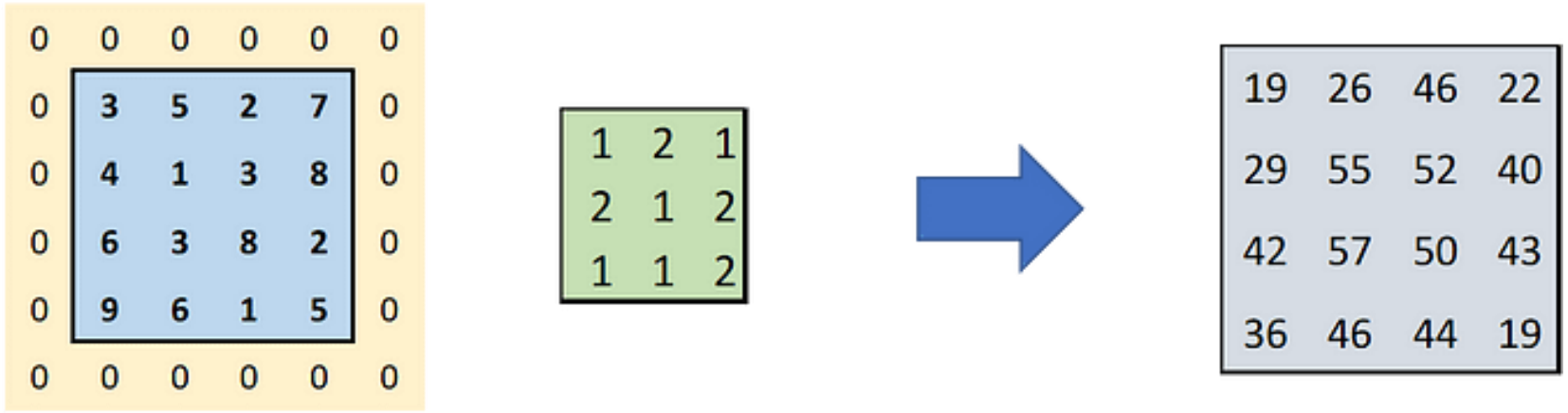

Strides = (1, 1)
55 52
57 50
1 2
2 1
55 162 104
167 323 152
114 157 50
55 162 104
167 323 152
114 157 50
3x3 Output Matrix
Stride in Conv
Strides = (2, 2)
55 52
57 50
1 2
2 1
55 110 52 104
110 55 104 52
57 114 50 100
114 57 100 50
55 110 52 104
110 55 104 52
57 114 50 100
114 57 100 50
4x4 Output Matrix

Padding: “valid”

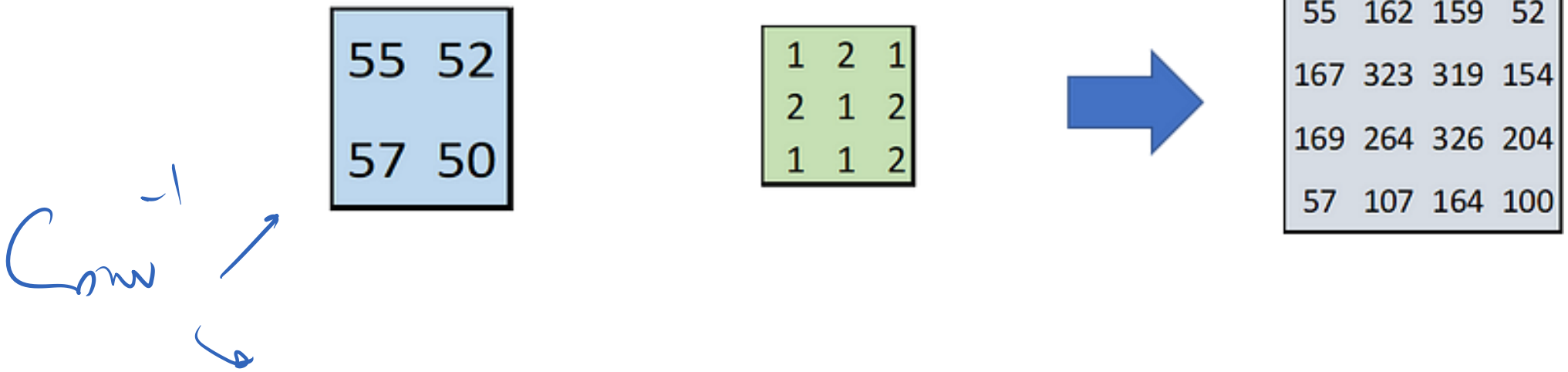


Conv

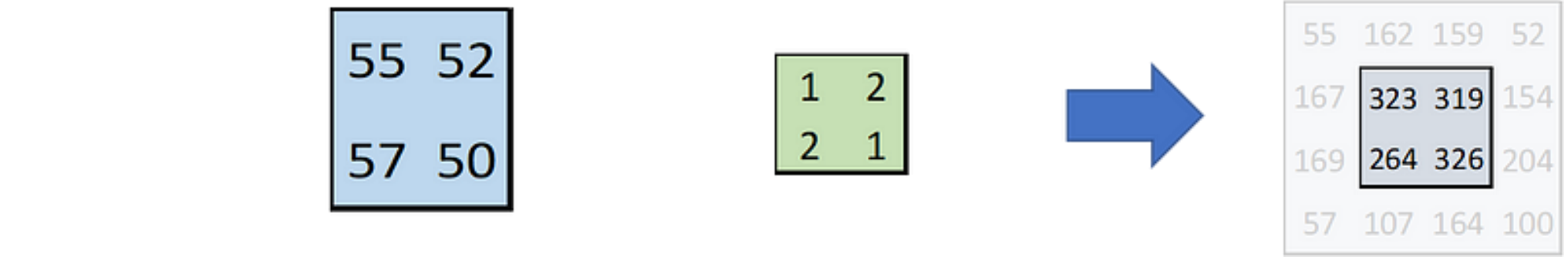
Padding: “same”

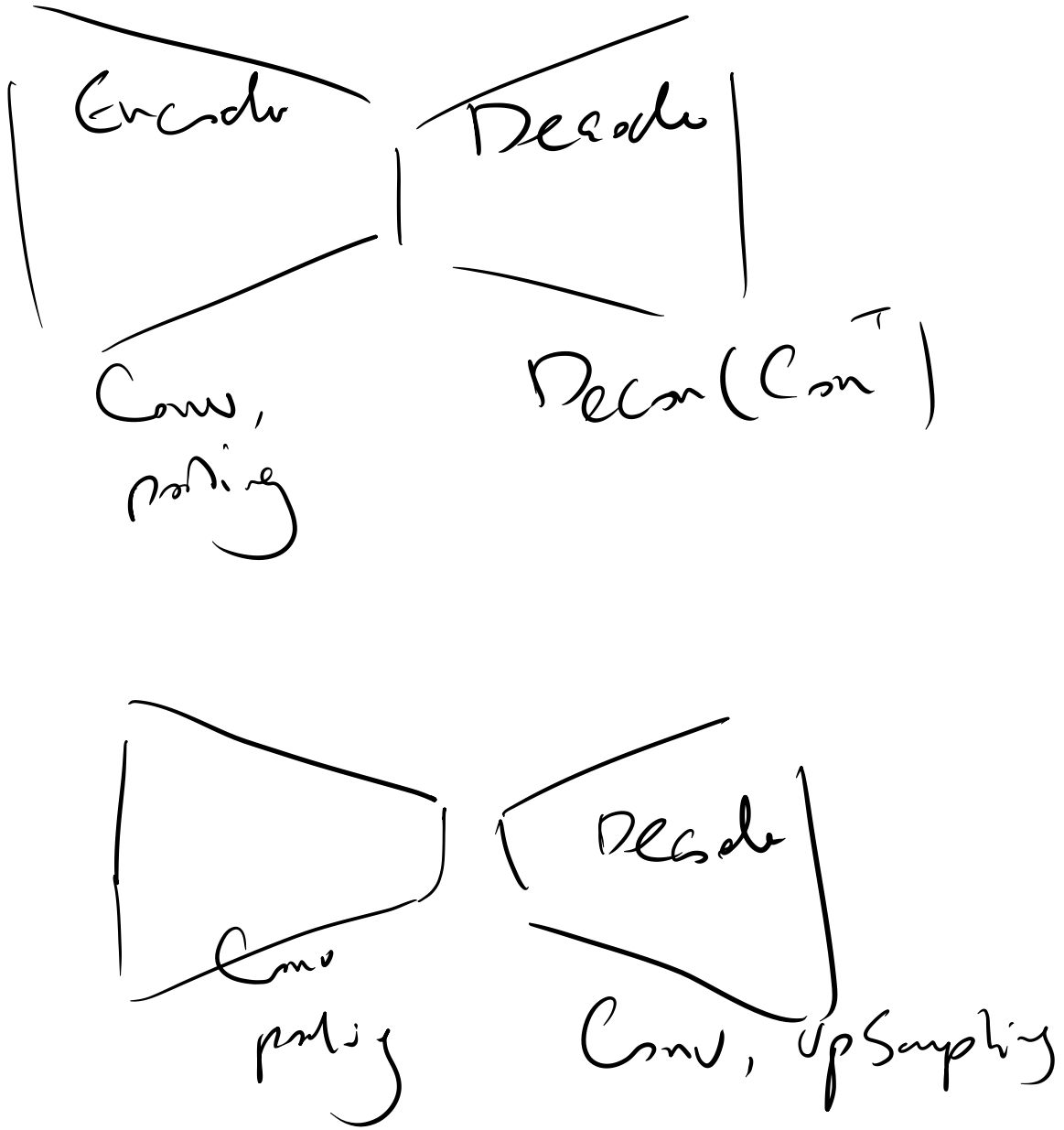


Padding: “valid”

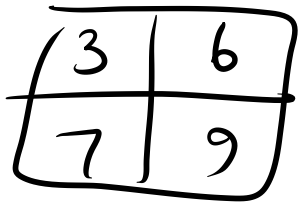


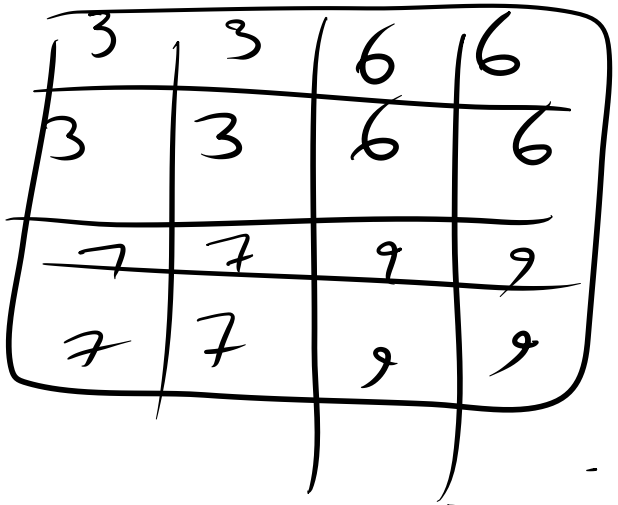
Padding: “same”



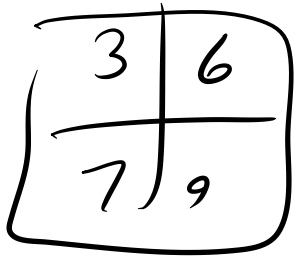

Encoder
Decoder
Conv, pooling
Decon(Conv^T)
Encoder
Decoder
Conv, pooling
Conv, UpSampling

Upsampling

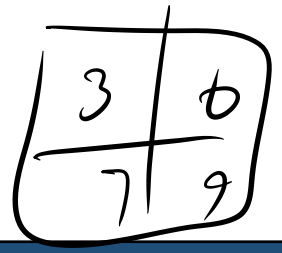




X repeat



Bilinear



Bicubic