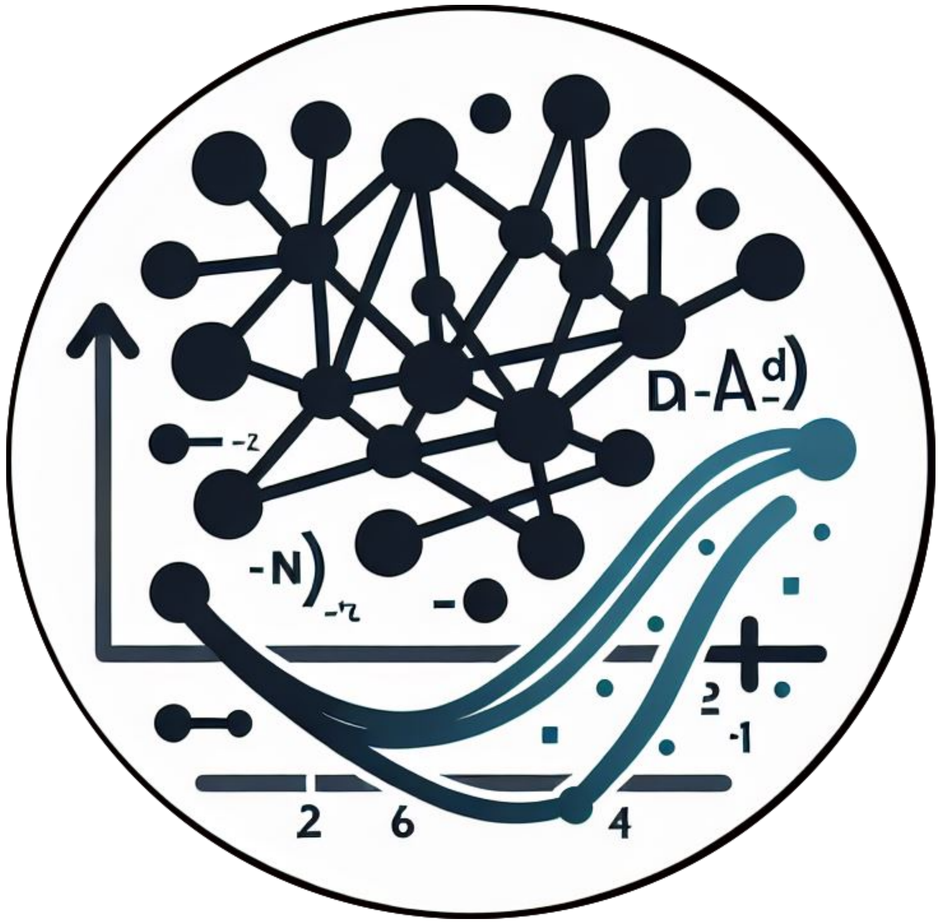
Convolutional Neural Nets



Deep Learning for Engineers
Andrew Ning
aning@byu.edu

What do we use Neural ODEs for?

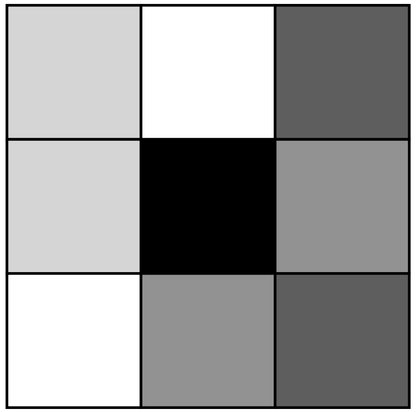
Predict forward in time (e.g., weather problem)

Predict different initial conditions/parameters

Increase or reduce dimensionality to make problem better behaved numerically, or linear, etc.

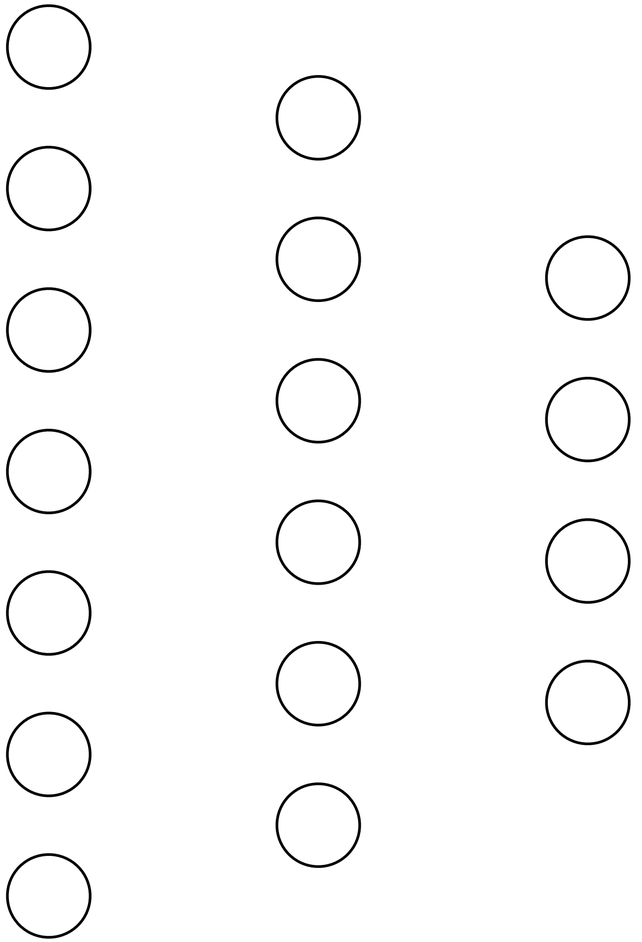
Symbolic regression

Speed

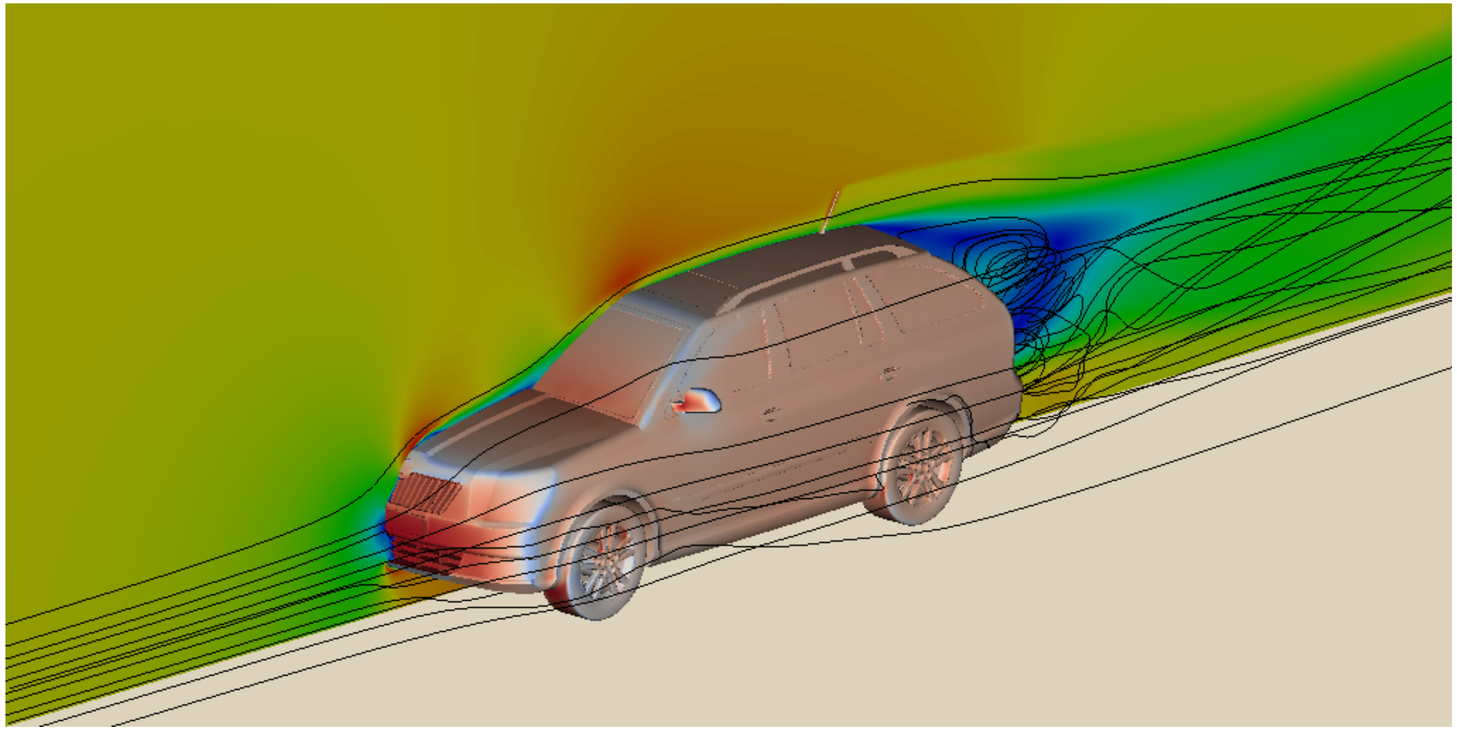


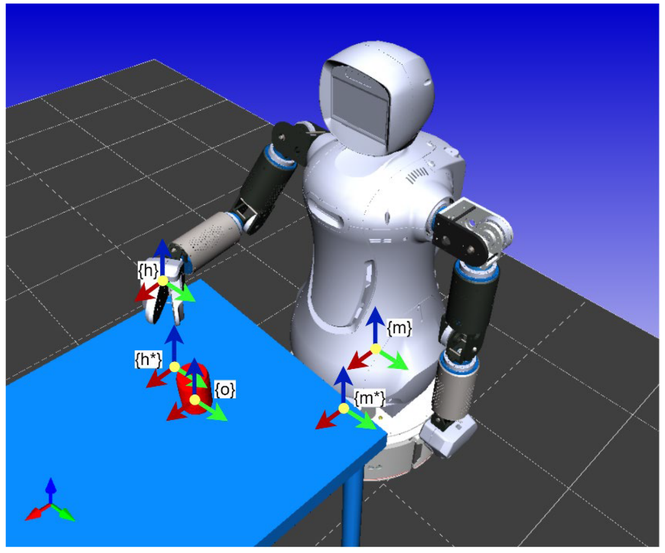
.75	1	.25
.75	0	.5
1	.5	.25

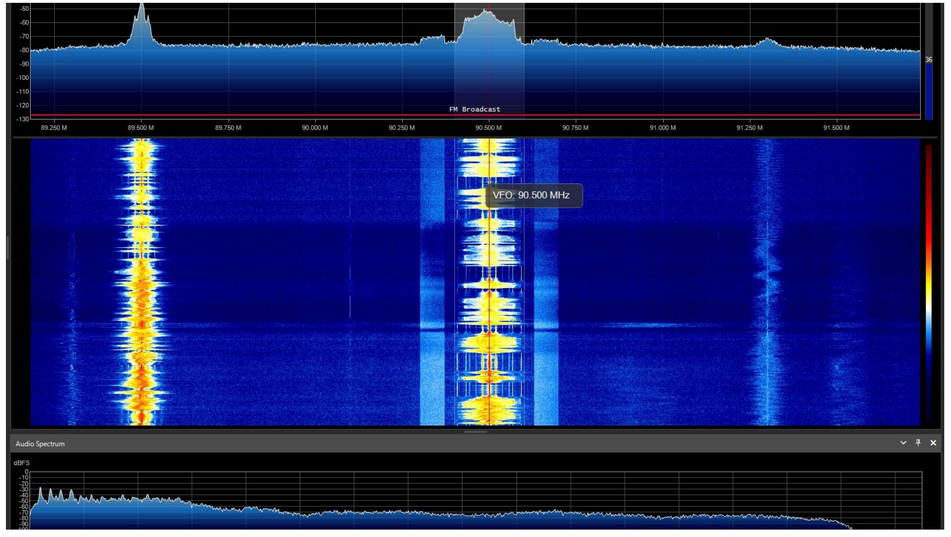
.75
1
.25
.75
0
.5
1
.5
.25

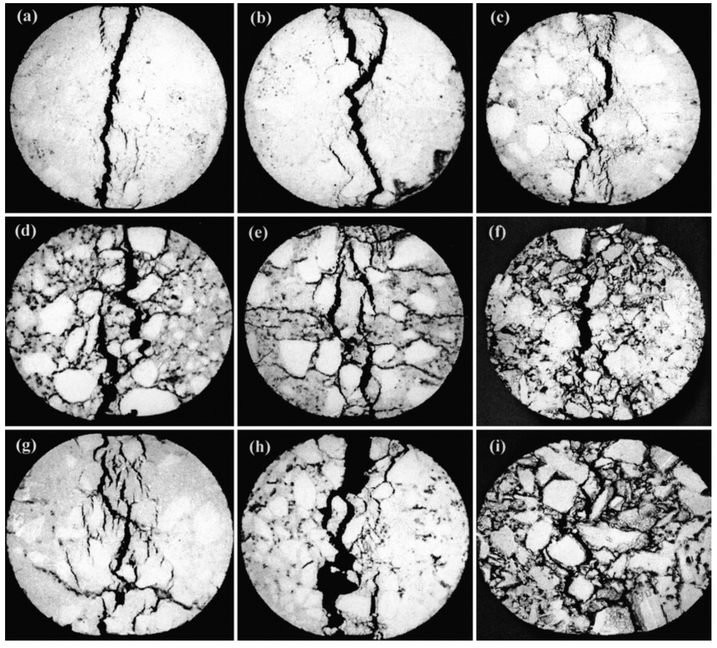


https://www.cs.montana.edu/courses/fall2006/430/lectures/02/lect02.html

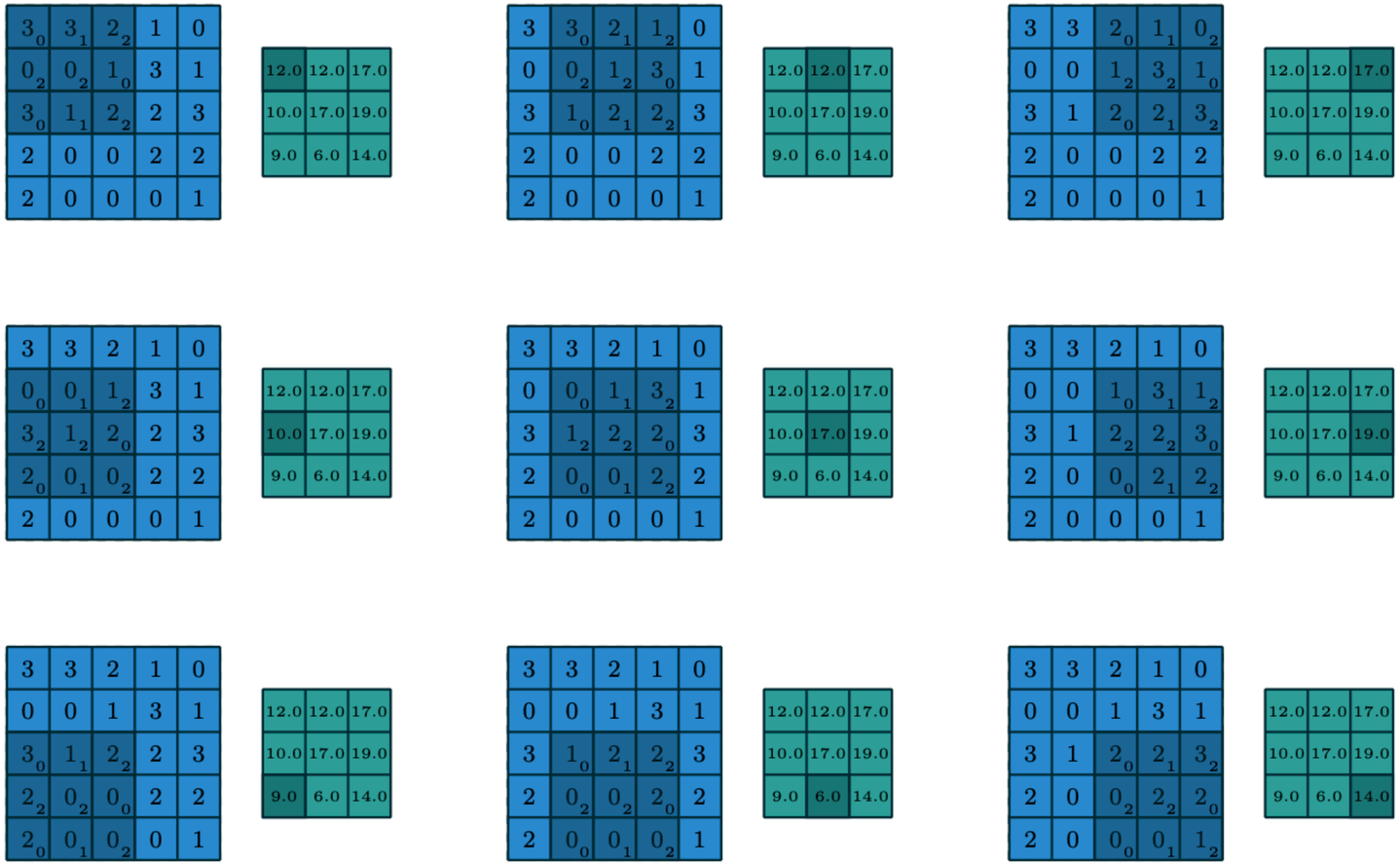



{h}
{h*}
{o}
{m}
{m*}


FM Broadcast
89.250 M
89.500 M
89.750 M
90.000 M
90.250 M
90.500 M
90.750 M
91.000 M
91.250 M
91.500 M
VFO: 90.500 MHz
Audio Spectrum
dBFS


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

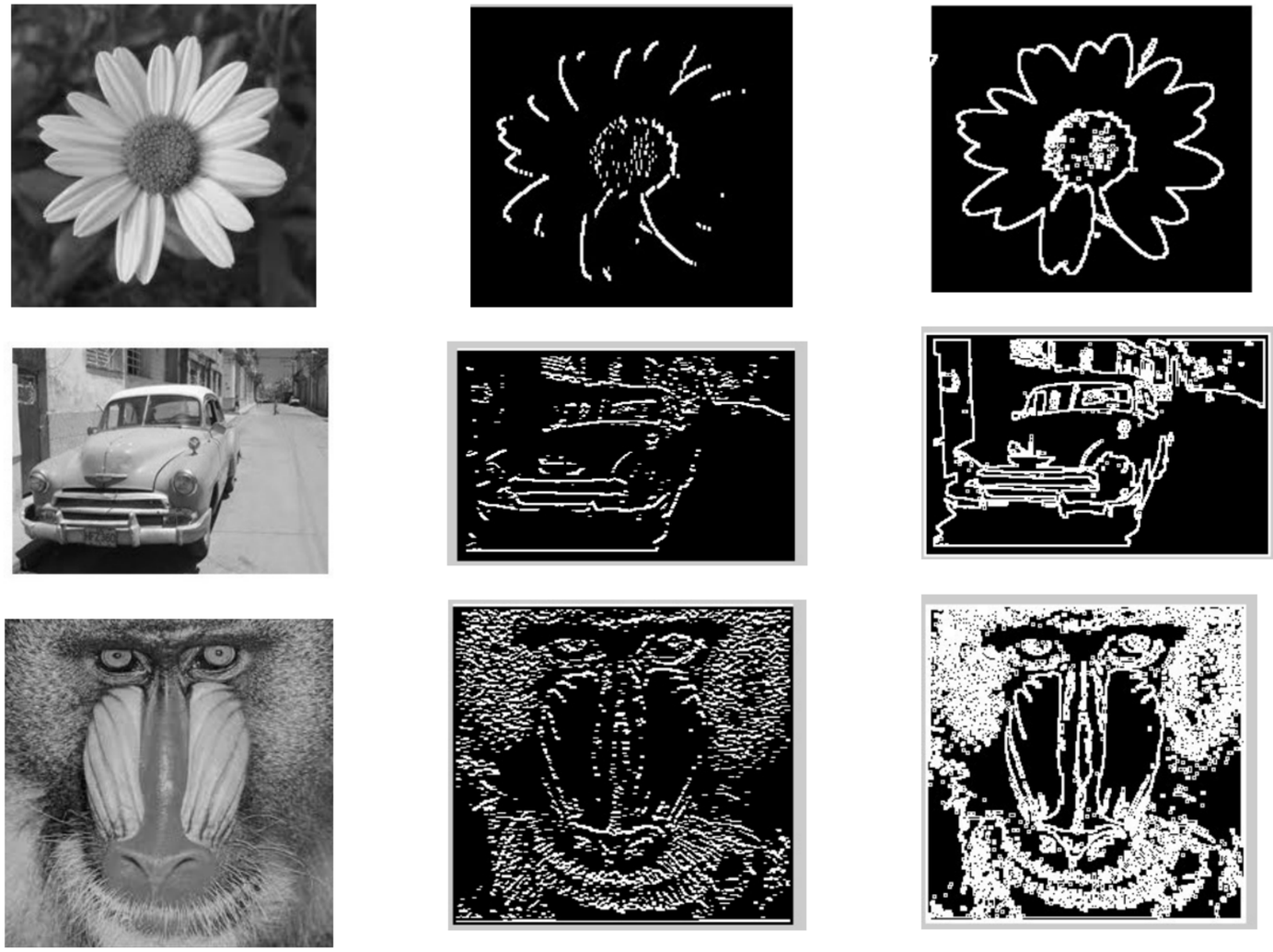
Kernel



A guide to convolution arithmetic for deep learning, Dumoulin and Visinn

Edge Detection

1	0	-1
1	0	-1
1	0	-1



How Image Edge Detection Works, Aryaman Sharda

kernel

0	1
2	2

input

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

output

kernel

0	1
2	2

input

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

output

kernel

0	1
2	2

input

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

output

3			

kernel

0	1
2	2

input

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

output

3	4		

kernel

0	1
2	2

input

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

output

3	4	9	

kernel

0	1
2	2

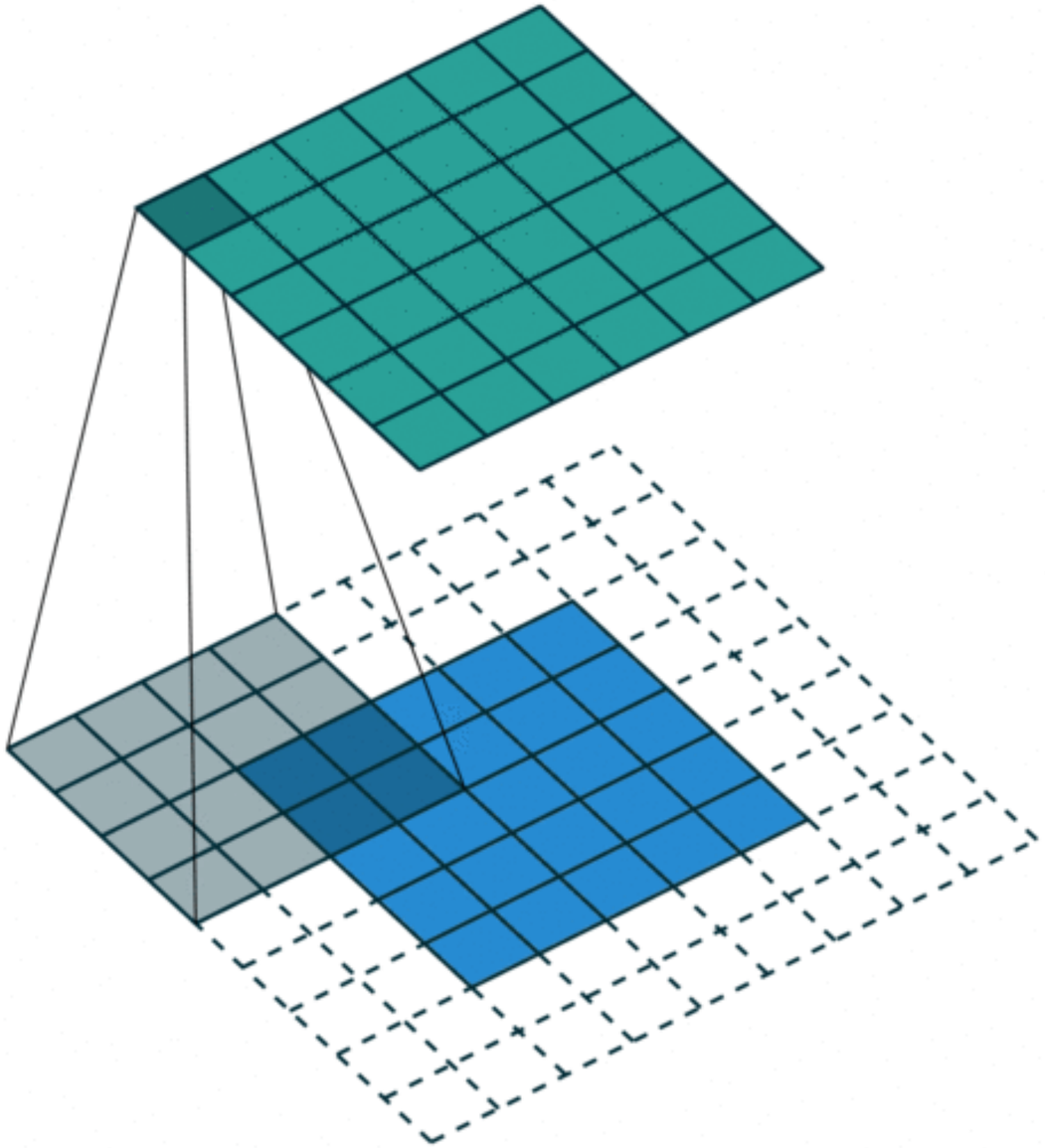
input

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

output

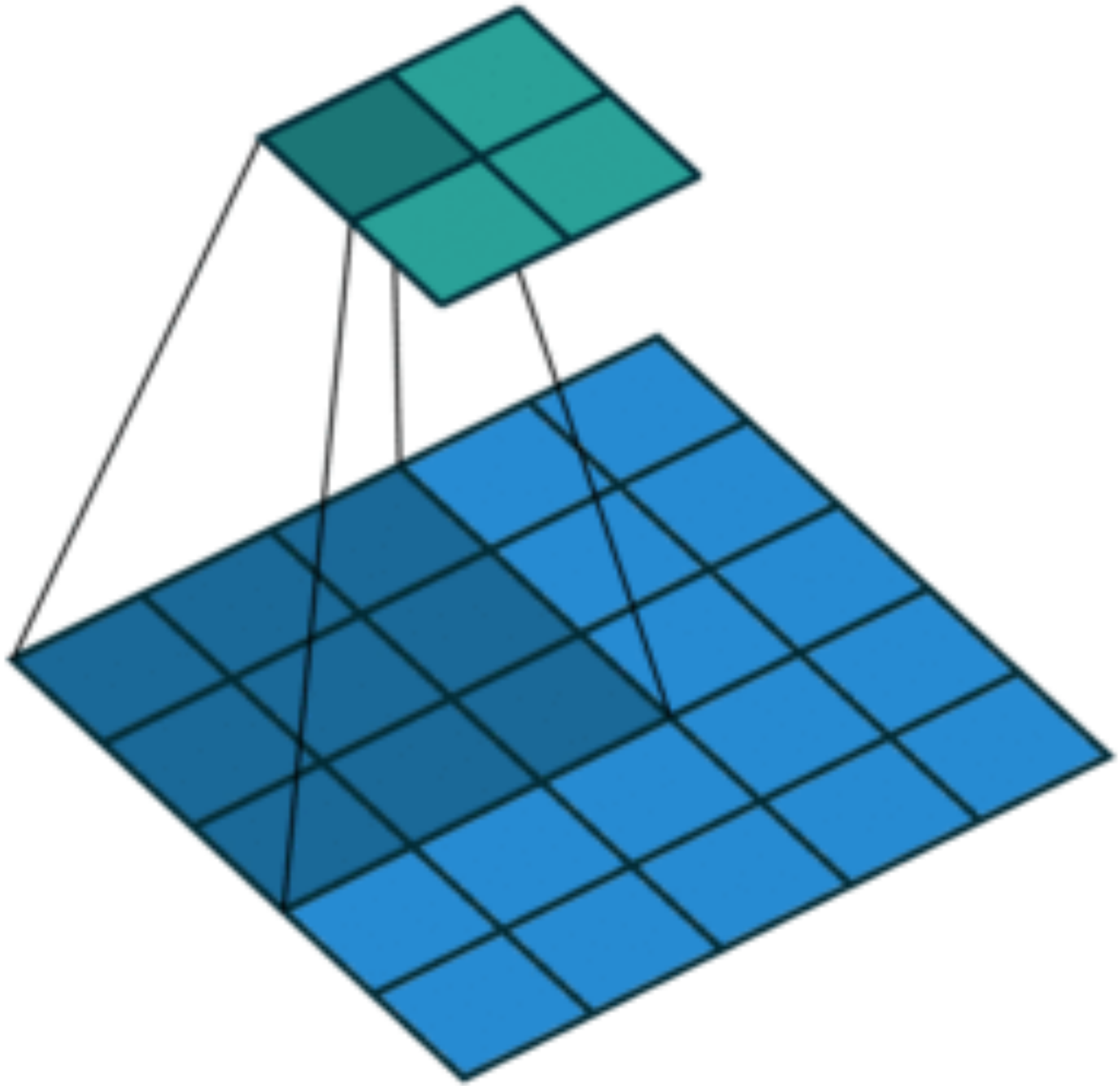
3	4	9	8

Padding

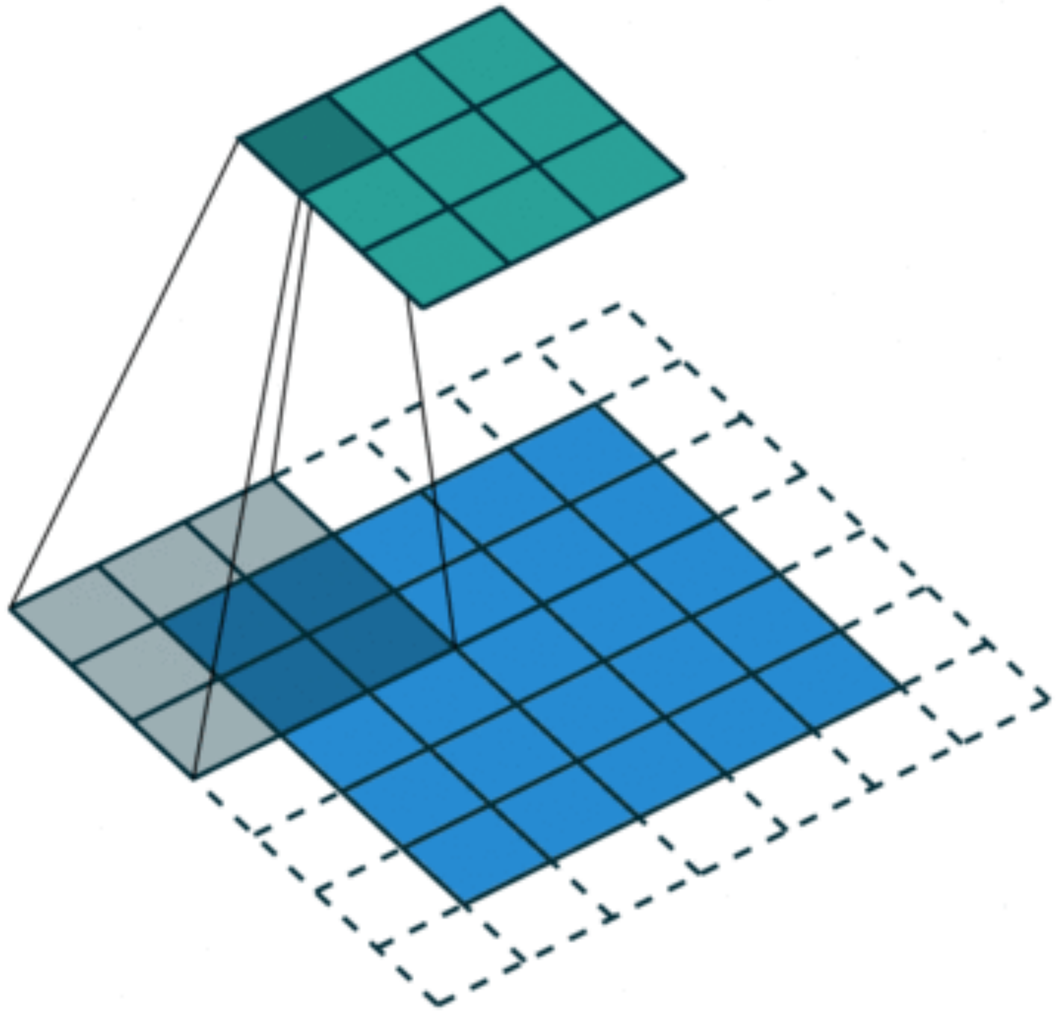


A guide to convolution arithmetic for deep learning, Dumoulin and Visinn

Strides



Padding and Strides

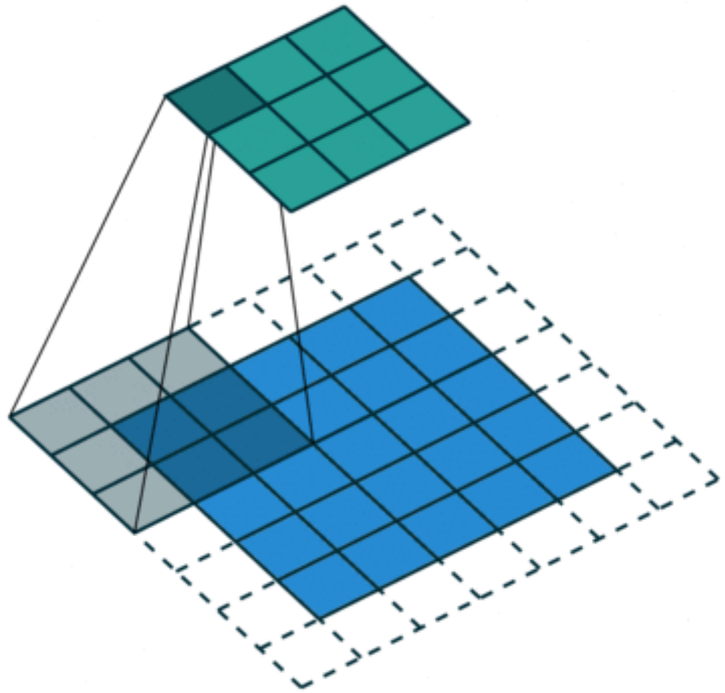


new size

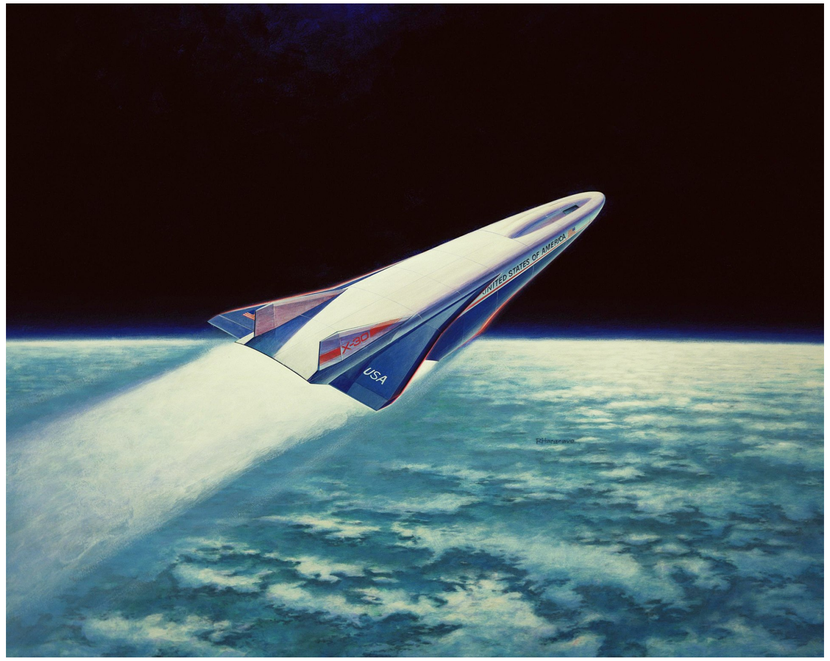
$$H_{out} = \frac{H_{in} - \text{kernel} + (2 \times \text{padding})}{\text{stride}} + 1$$

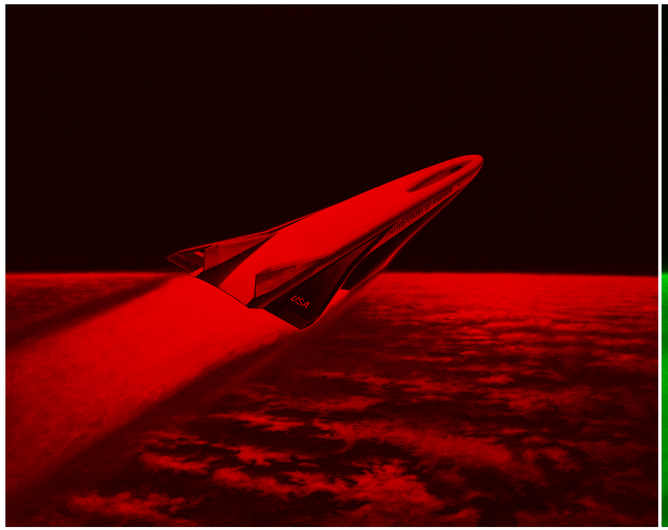
new size

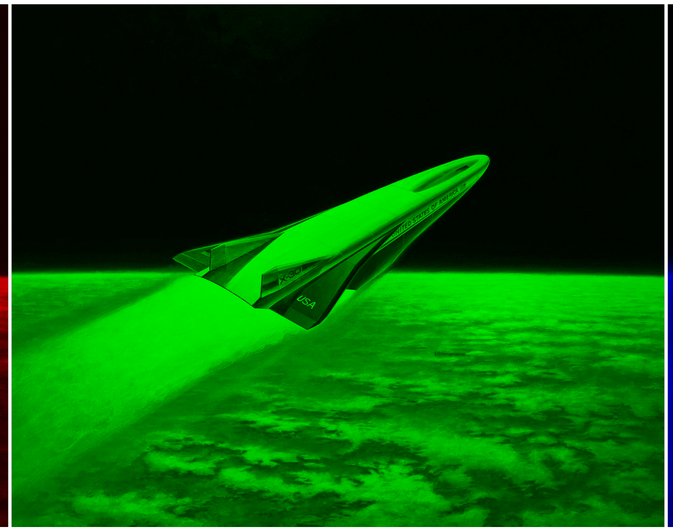
$$H_{out} = \frac{H_{in} - \text{kernel} + (2 \times \text{padding})}{\text{stride}} + 1$$

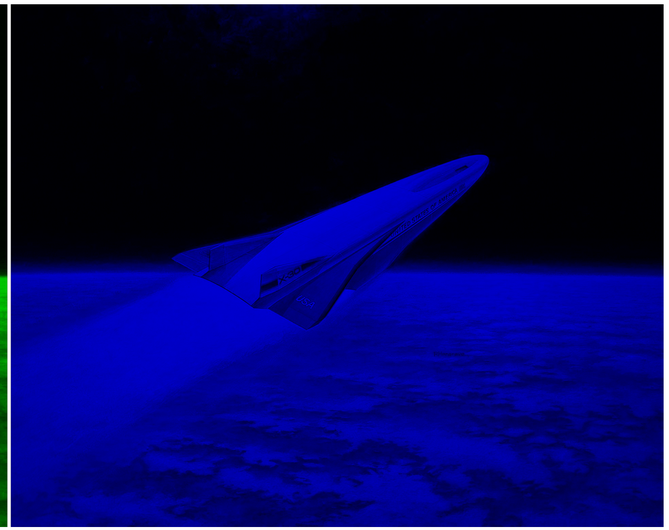


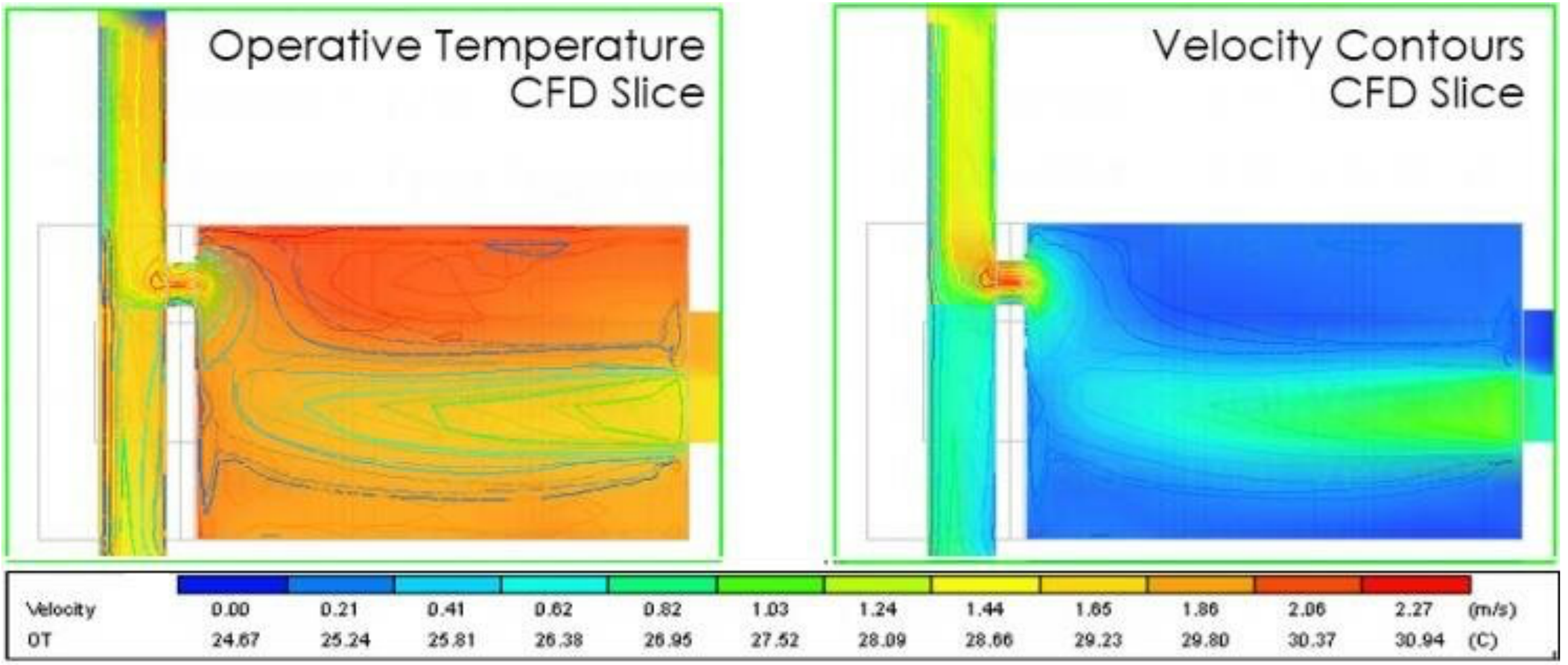
Channels





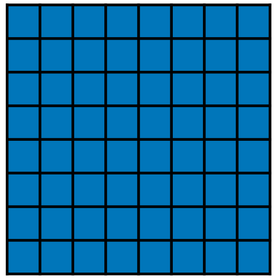




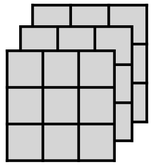


Solar Chimney for Enhanced Natural Ventilation Based on CFD-Simulation for a Housing Prototype in Alexandria, Egypt. Gehad Mekkawi and Rana Elgendy

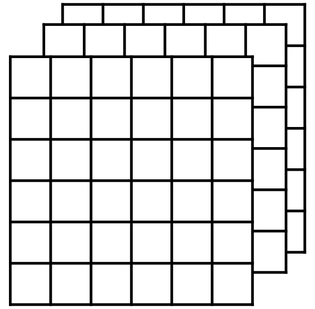
Multiple Outputs



8 x 8 x 1

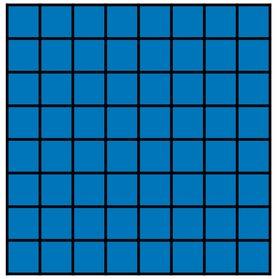


3 x 3 x 1 x 3



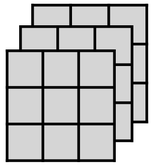
6 x 6 x 3

Multiple Outputs



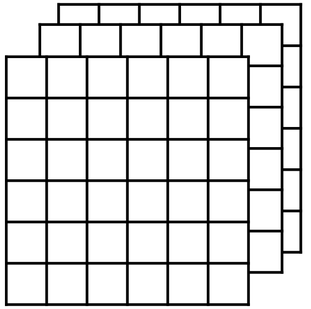
8 x 8 x 1

in channels



3 x 3 x 1 x 3

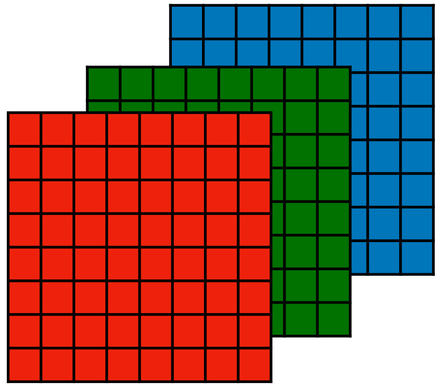
kernel size



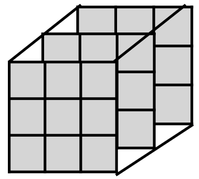
6 x 6 x 3

out channels

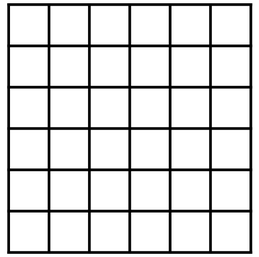
Multiple Inputs



8 x 8 x 3

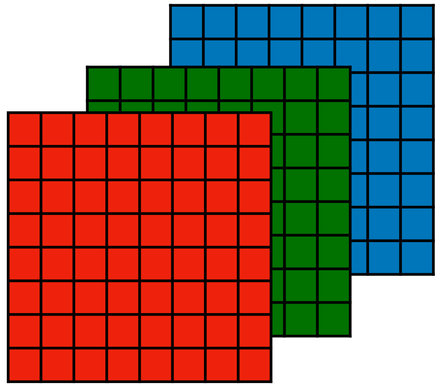


3 x 3 x 3 x 1



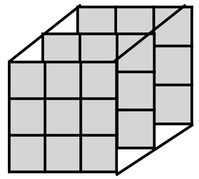
6 x 6 x 1

Multiple Inputs



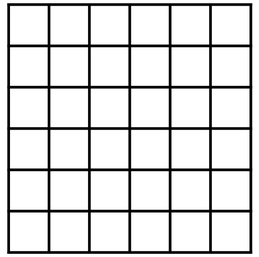
8 x 8 x 3

in channels



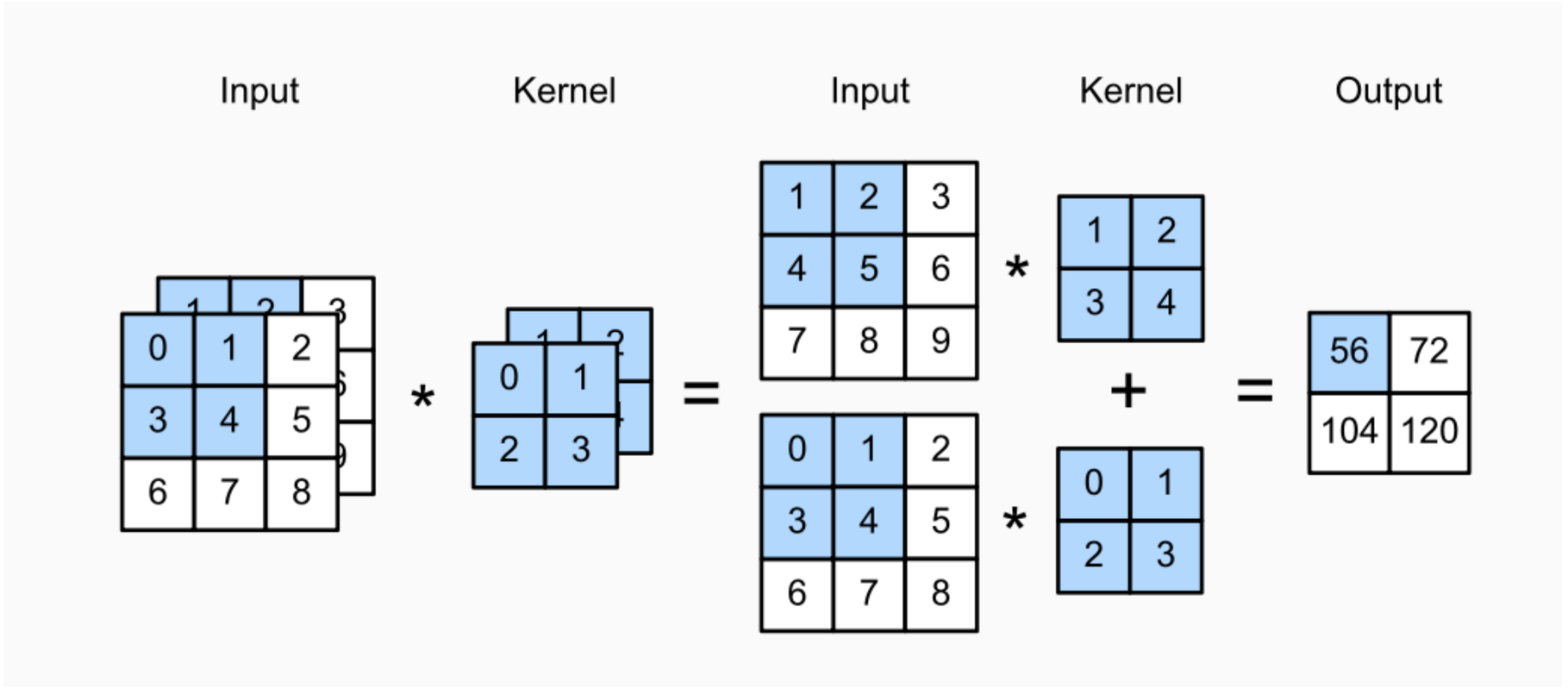
3 x 3 x 3 x 1

kernel size



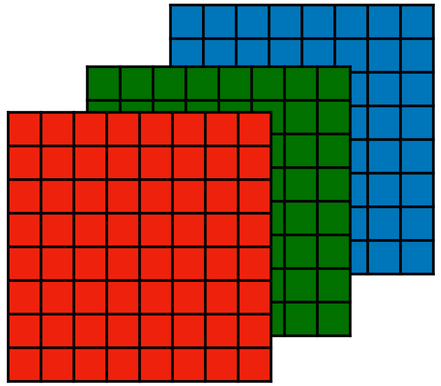
6 x 6 x 1

out channels

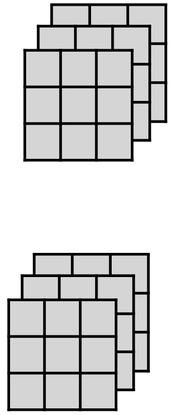

Input
Kernel
Input
Kernel
Output
0 1 2
3 4 5
6 7 8
0 1
2 3
1 2 3
4 5 6
7 8 9
1 2
3 4
0 1 2
3 4 5
6 7 8
0 1
2 3
56 72
104 120

http://d2l.ai

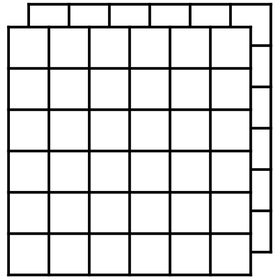
Multiple Inputs and Multiple Outputs



8 x 8 x 3

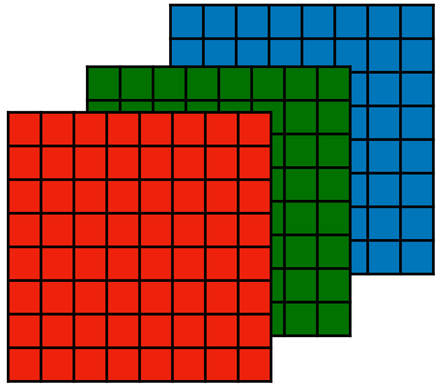


3 x 3 x 3 x 2



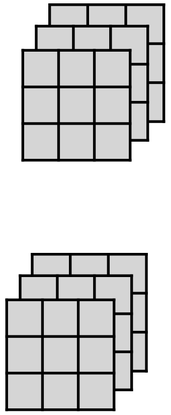
6 x 6 x 2

Multiple Inputs and Multiple Outputs



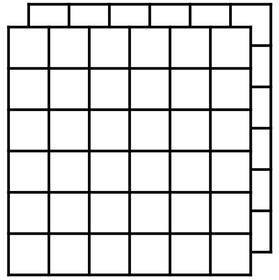
8 x 8 x 3

in channels



3 x 3 x 3 x 2

kernel size



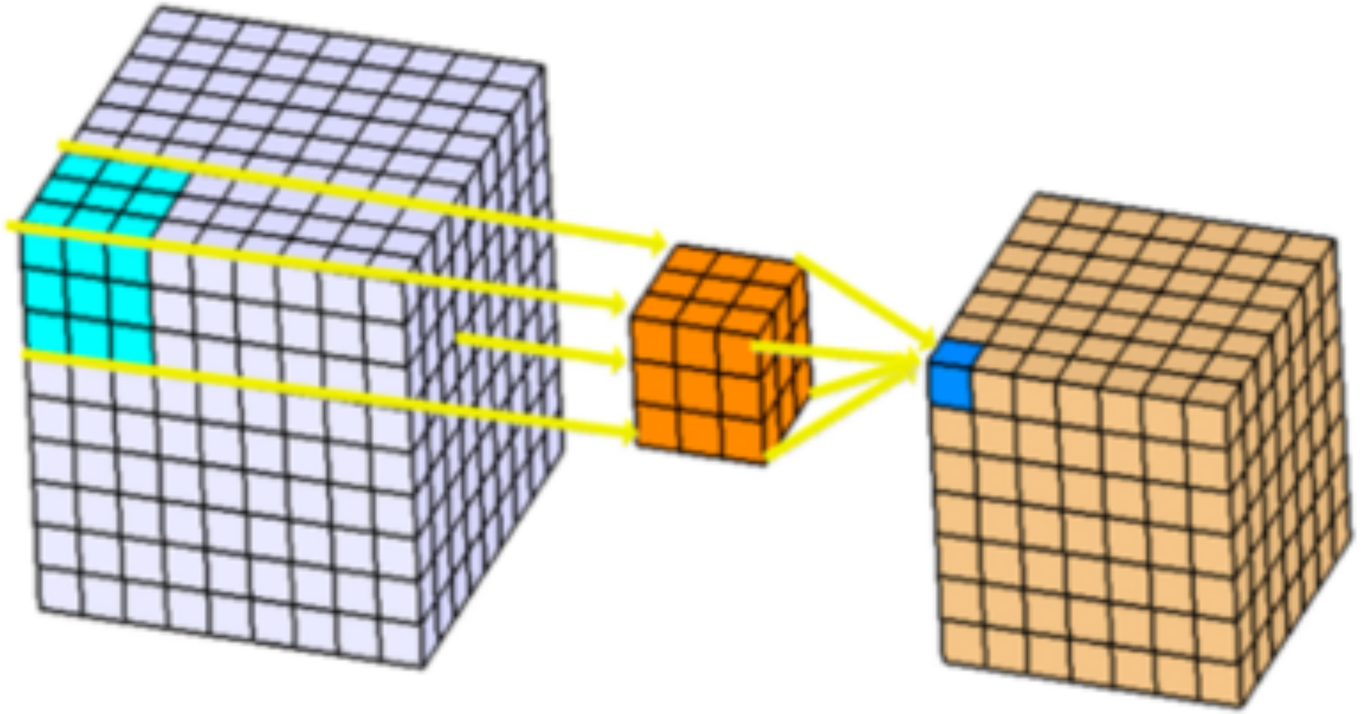
6 x 6 x 2

out channels

Conv2d

torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1, padding=0)

3D convolutions



https://www.kaggle.com/code/shivamb/3d-convolutions-understanding-use-case

Pooling

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

max pool

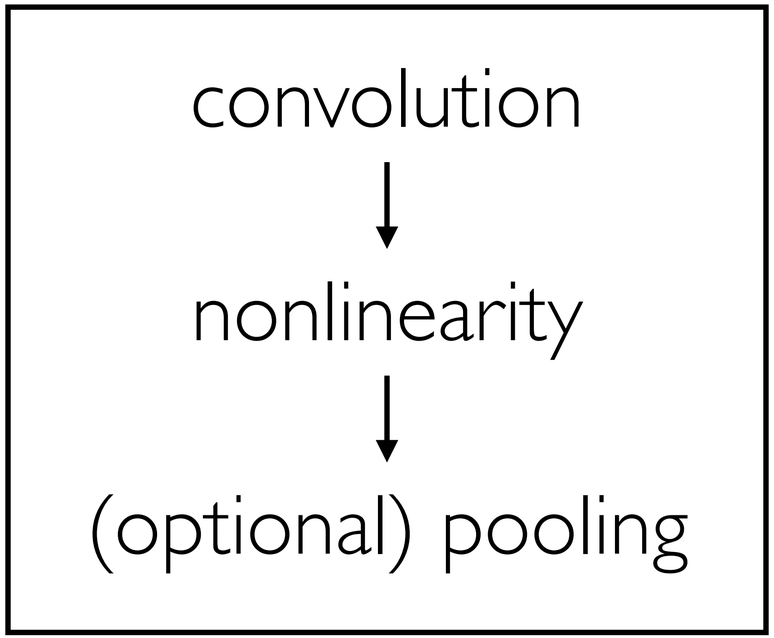
2x2 filter

stride = 2

3	2
1	2

torch.nn.MaxPool2d

torch.nn.AvgPool2d



(potentially a linear layer(s) after all the convolution layers)