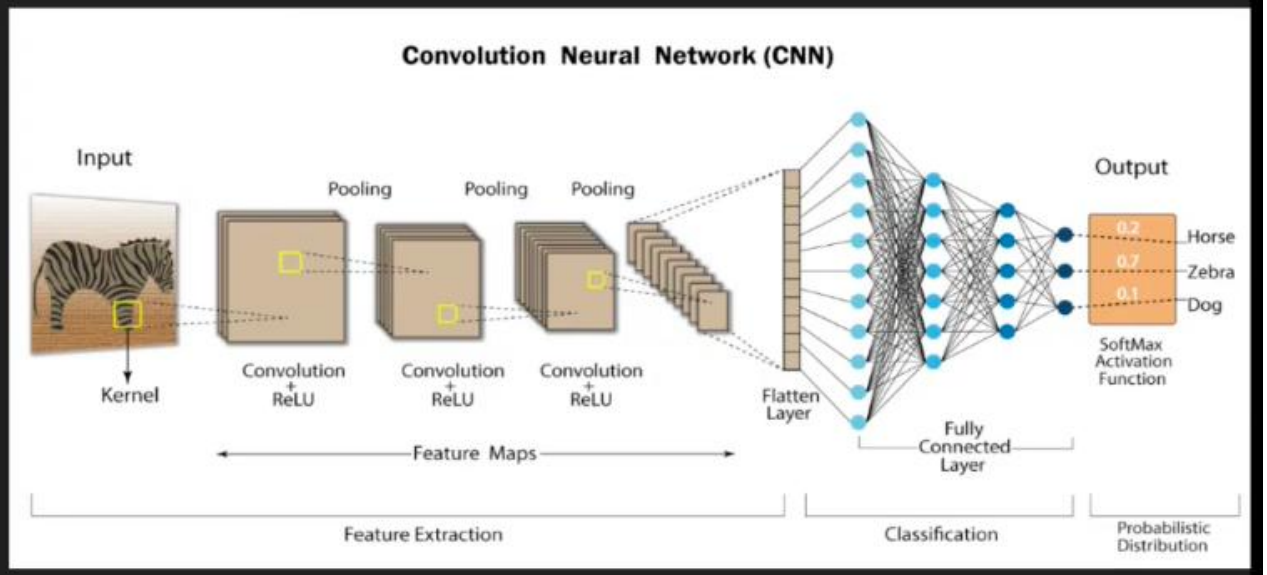
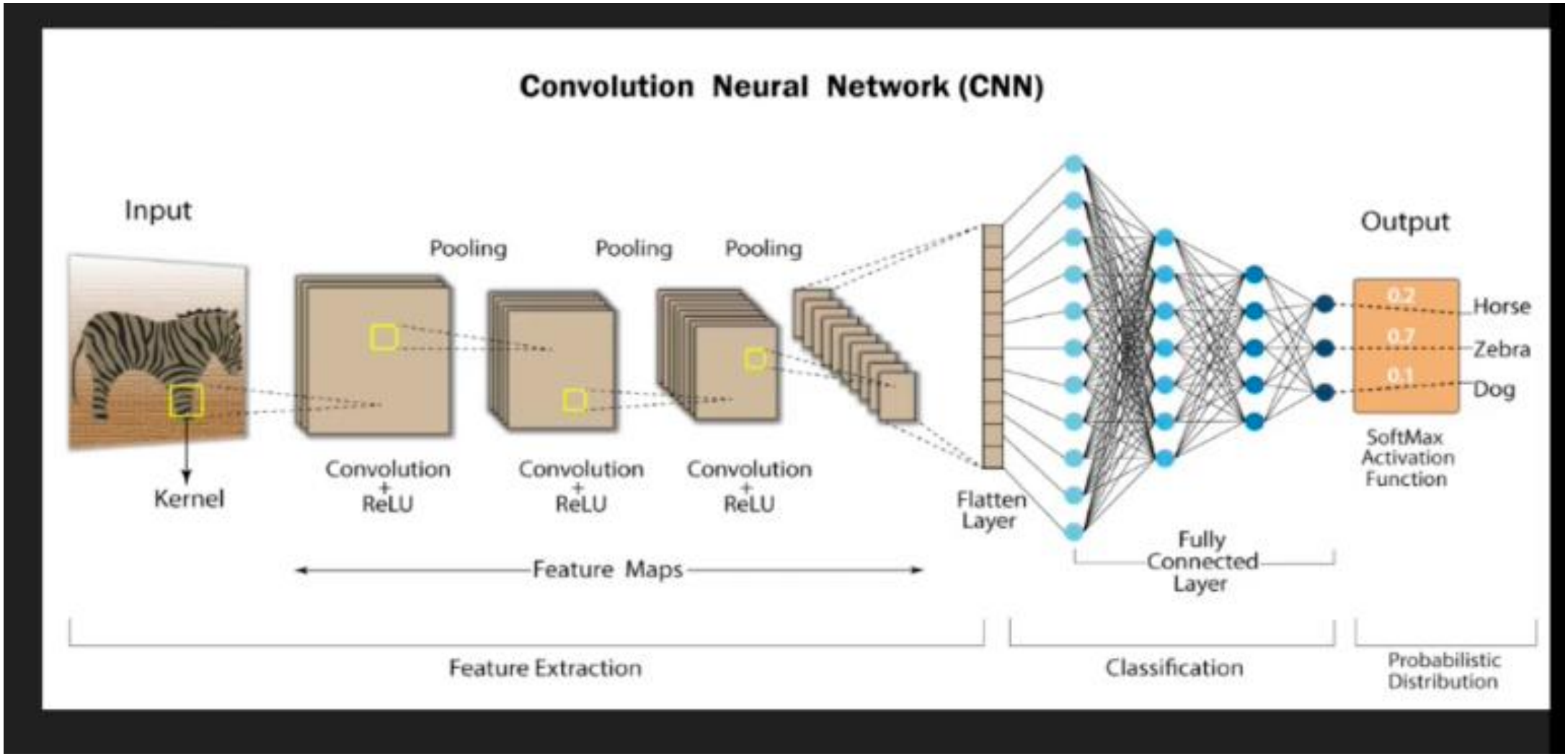
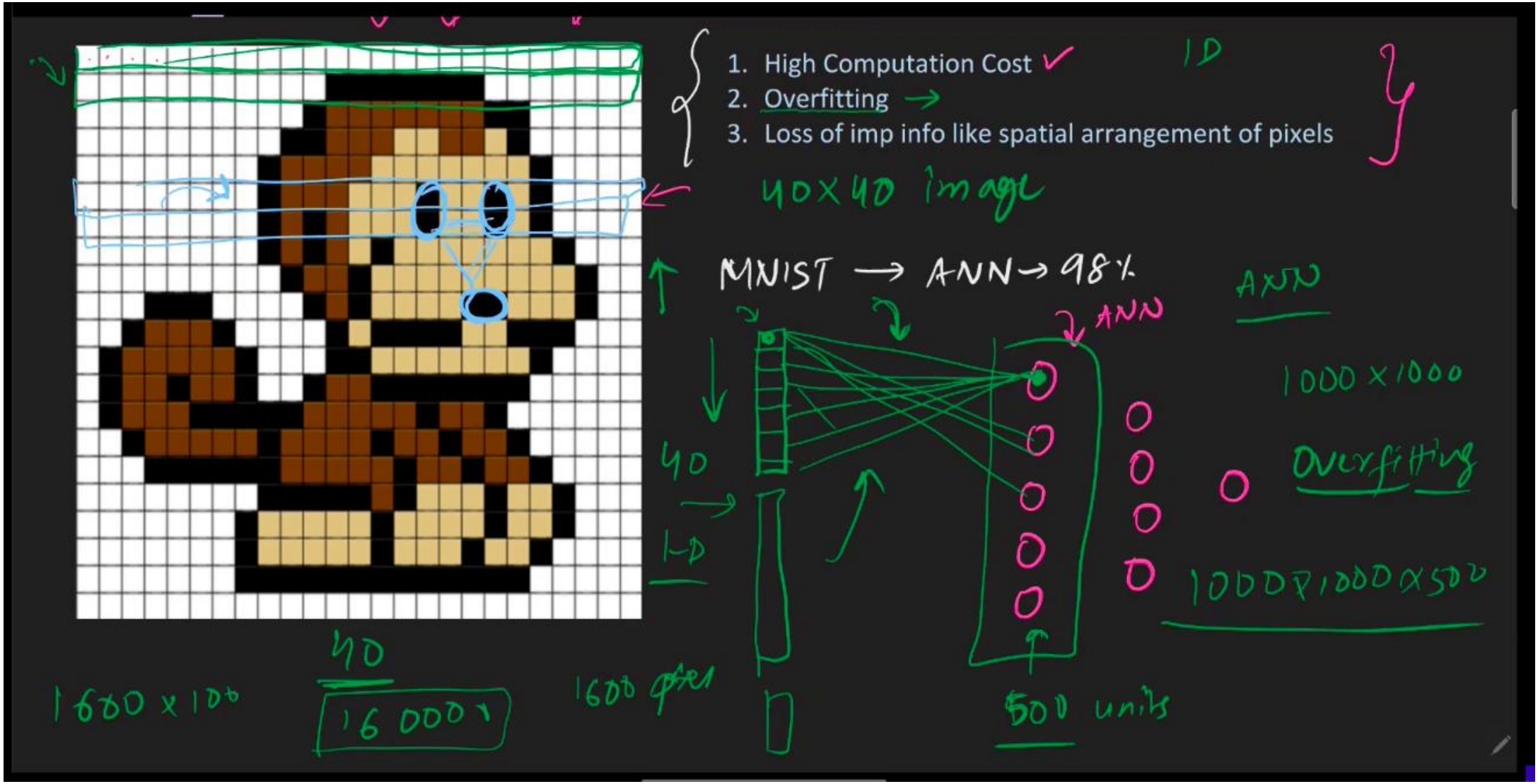
What is a CNN?

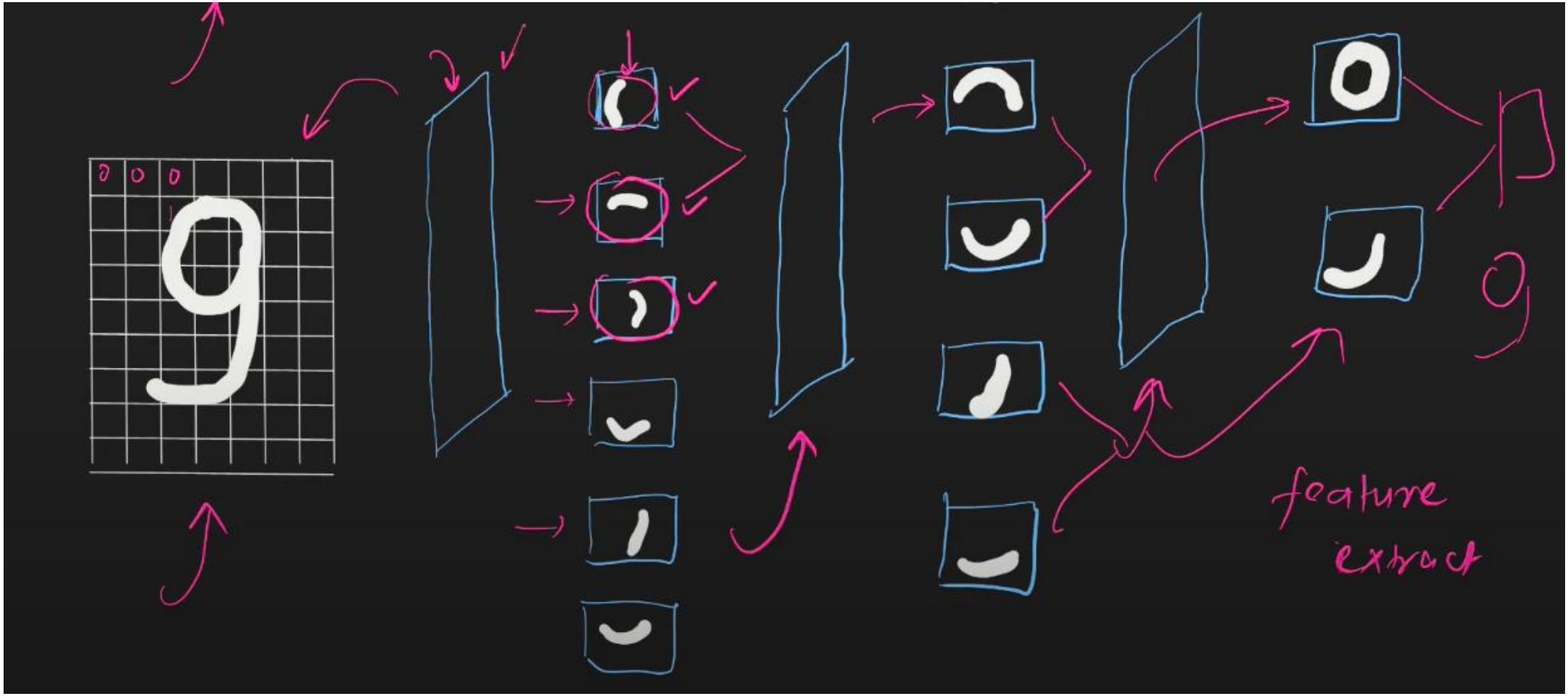
17 August 2022 06:47

Convolutional neural networks, also known as convnet, or CNNs, are a special kind of neural network for processing data that has a known grid-like topology like time series data(1D) or images(2D).

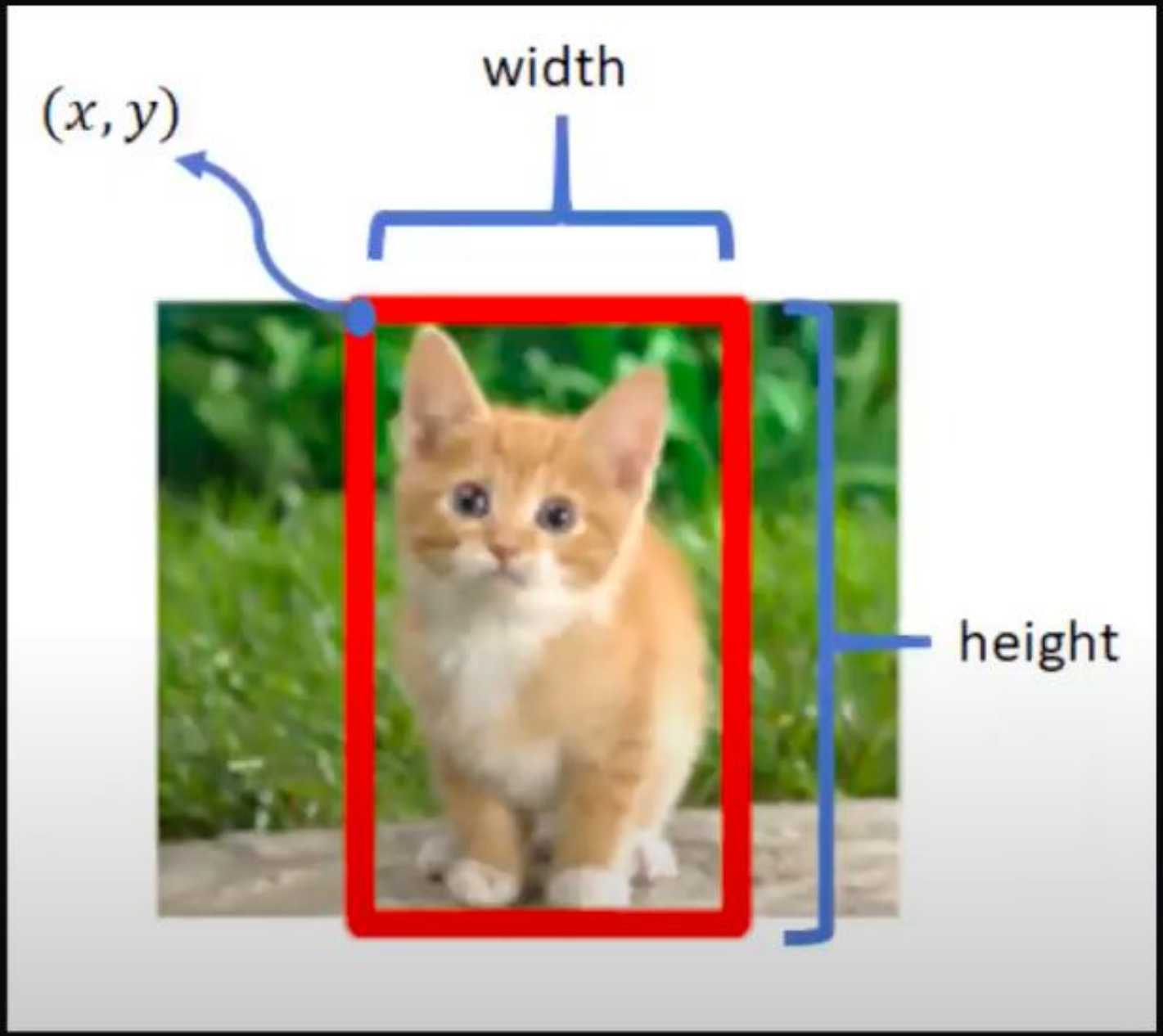



Convolution Neural Network (CNN)
Input
Kernel
Pooling
Pooling
Pooling
Convolution + ReLU
Convolution + ReLU
Convolution + ReLU
Feature Maps
Flatten Layer
Fully Connected Layer
Output
0.2
0.7
0.1
Horse
Zebra
Dog
SoftMax Activation Function
Feature Extraction
Classification
Probabilistic Distribution

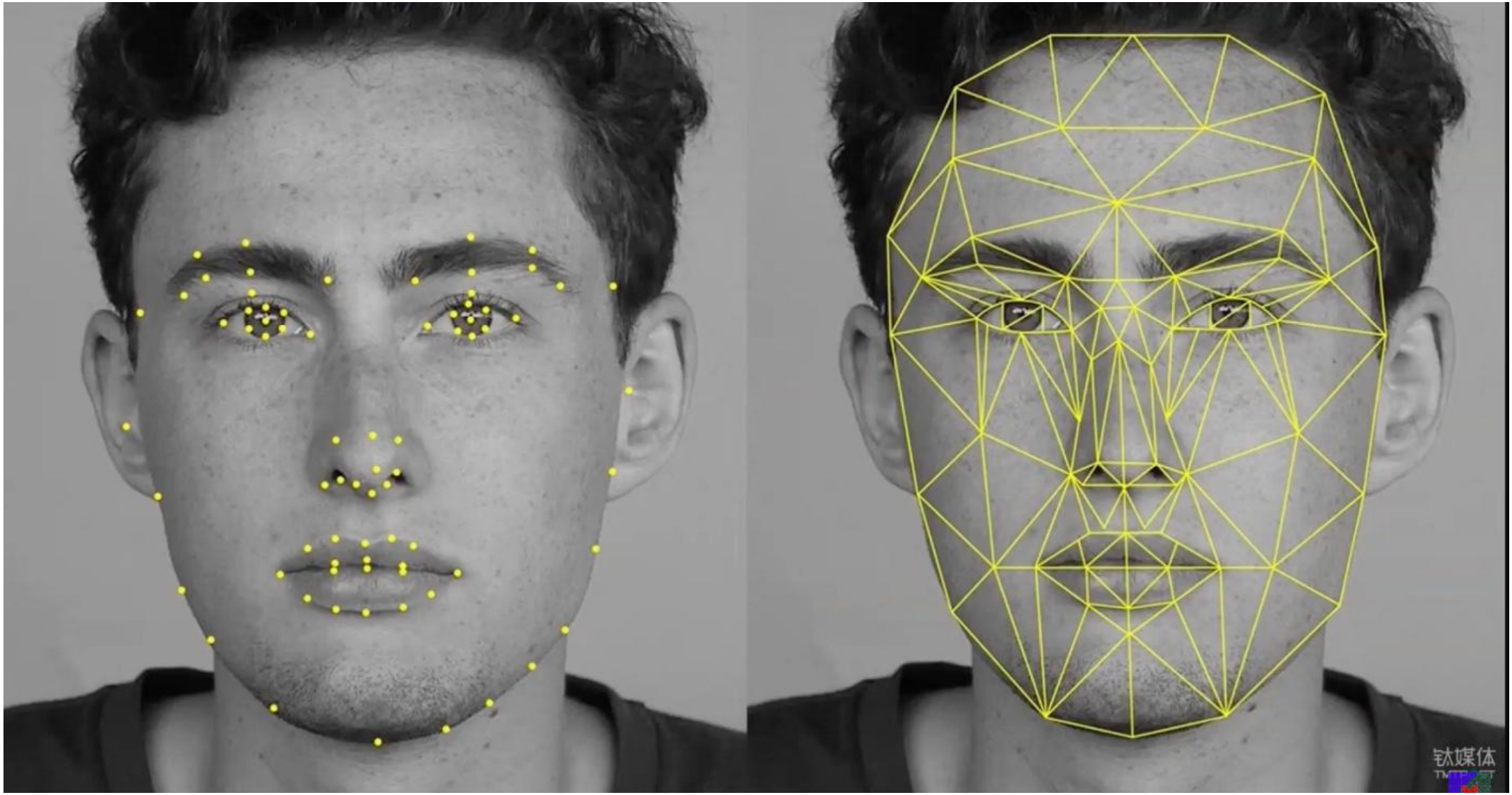

1. High Computation Cost
2. Overfitting
3. Loss of imp info like spatial arrangement of pixels
1D
40x40 image
MNIST → ANN → 98%
ANN
1000 x 1000
Overfitting
1000x1000x500
40
1-D
500 units
1600 x 100
40
160000
1600 pixel

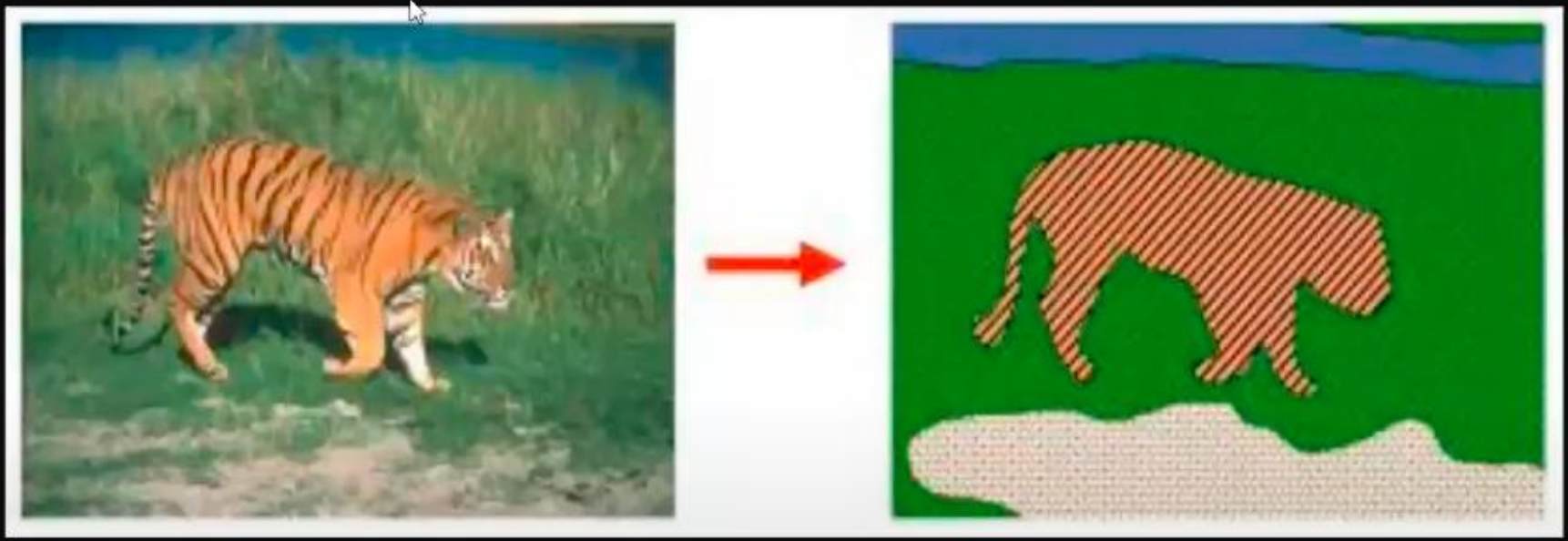

feature
extract


mite
container ship
motor scooter
leopard
mite
black widow
cockroach
tick
starfish
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
go-kart
moped
bumper car
golfcart
leopard
jaguar
cheetah
snow leopard
Egyptian cat


width
(x, y)
height

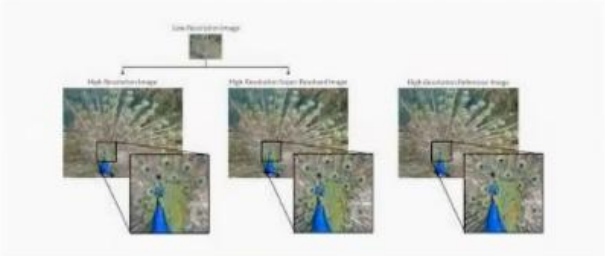

bus 98.009
Victoria 73
truck 63.110
person 77.116
bicycle 80.304


钛媒体

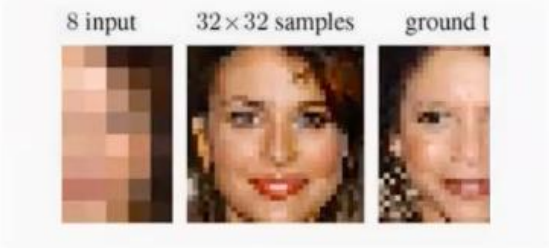


Google

super resolution deep learning



Increase Image Resolution Using Deep ...
mathworks.com



Deep Learning based image Super ...
cv-tricks.com

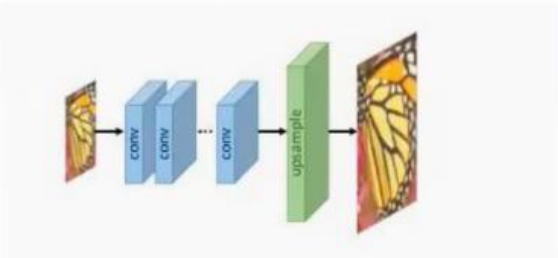
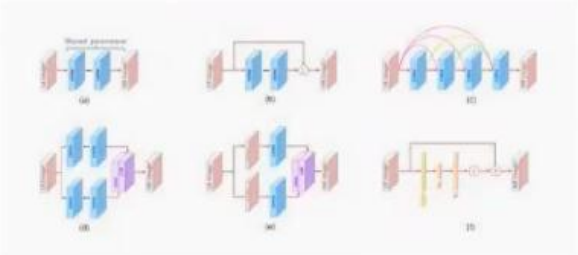
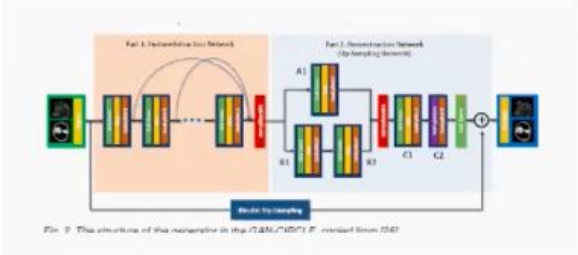


Image Super Resolution | Deep Learning ...
analyticsvidhya.com

Nonlinear Multi-scale Super-resolution ...
semanticscholar.org

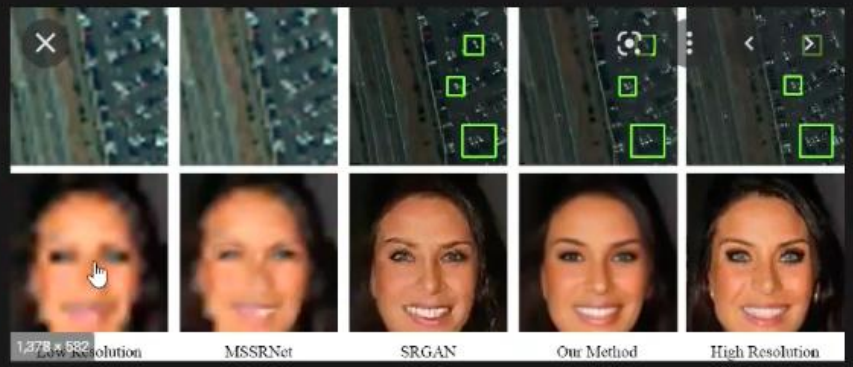


Deep-learning network structures for ...
researchgate.net



PDF] Super-resolution MRI through Deep ...
semanticscholar.org

https://www.semanticscholar.org/paper/Nonlinear-Multi-scale-Super-resolution-Using-Deep-Tran-Panahi/8385720cf6c8562b09946a61f48c032e94e994f0



Semantic Scholar

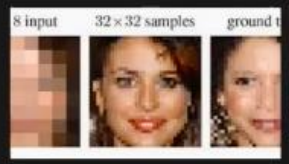
Nonlinear Multi-scale Super-resolution Using Deep Learning | Semantic Scholar

Visit

Images may be subject to copyright. **Learn More**

Related images

See more



Deep Learning based image Su...
cv-tricks.com

PDF] Real-World Super-Resoluti...
semanticscholar.org

GANs for Super-Resolution - DE...
dev.to

Nvidia's Super Resolution is an ...
trustedreviews.com

Image Super Resolution | Deep ...
analyticsvidhya.com

Unsupervised Single Image Su...
openaccess.thecvf.com


Original
Colorized
(x=477, y=13) ~ R:124 G:77 B:49

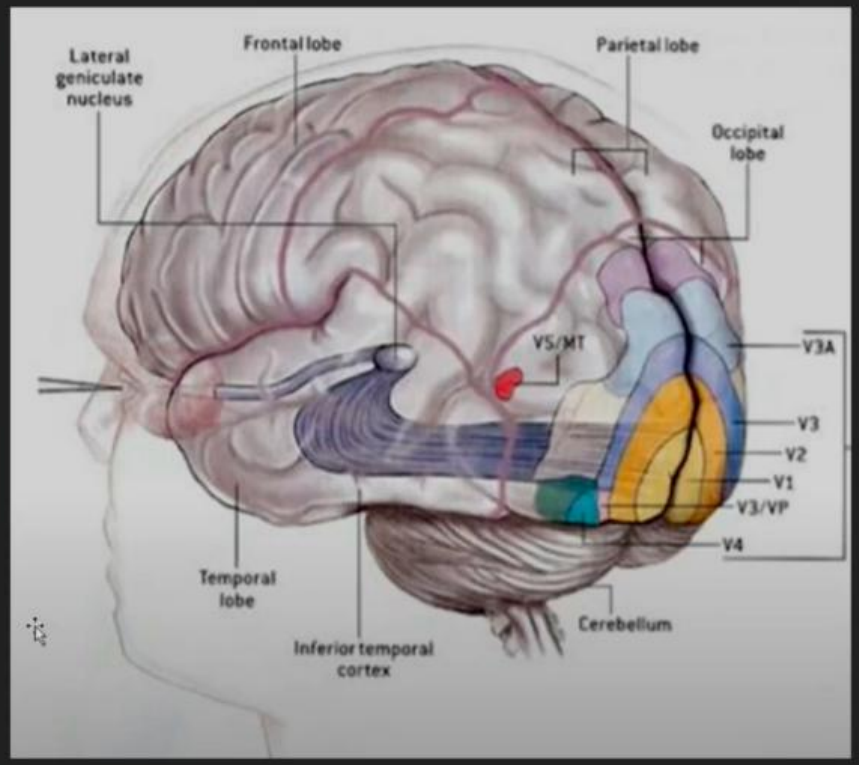
5:37 53%

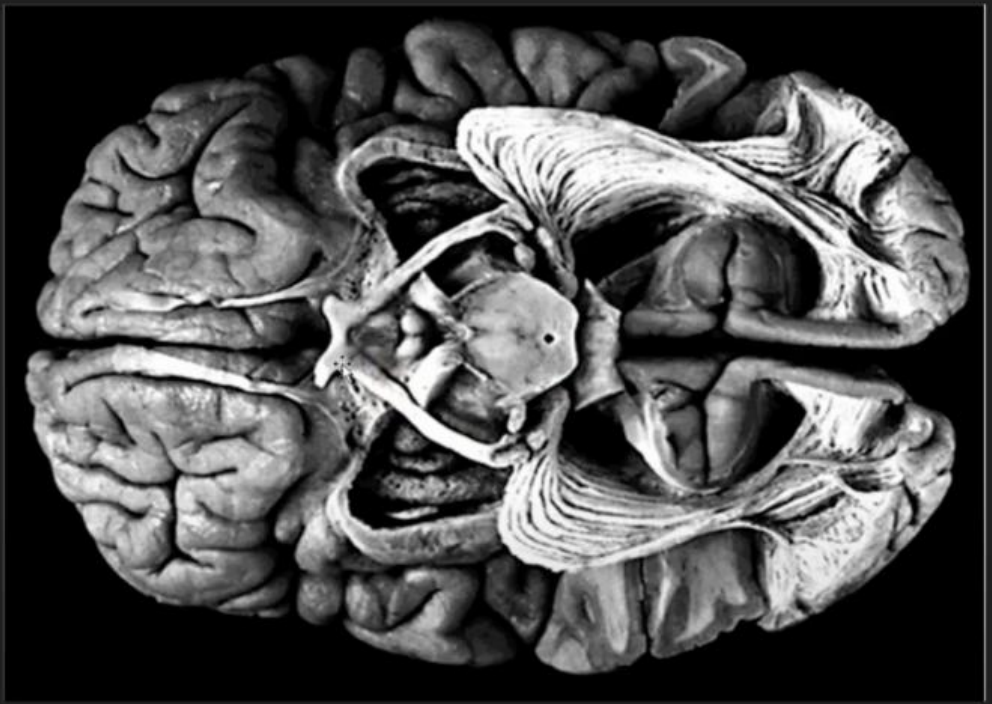
D36 CNN vs Visual Cortex HOME INSERT DRAW VIEW

Stop

Human Visual Cortex

18 August 2022 16:05





5:40 53%

D36 CNN vs Visual Cortex HOME INSERT DRAW VIEW

Stop

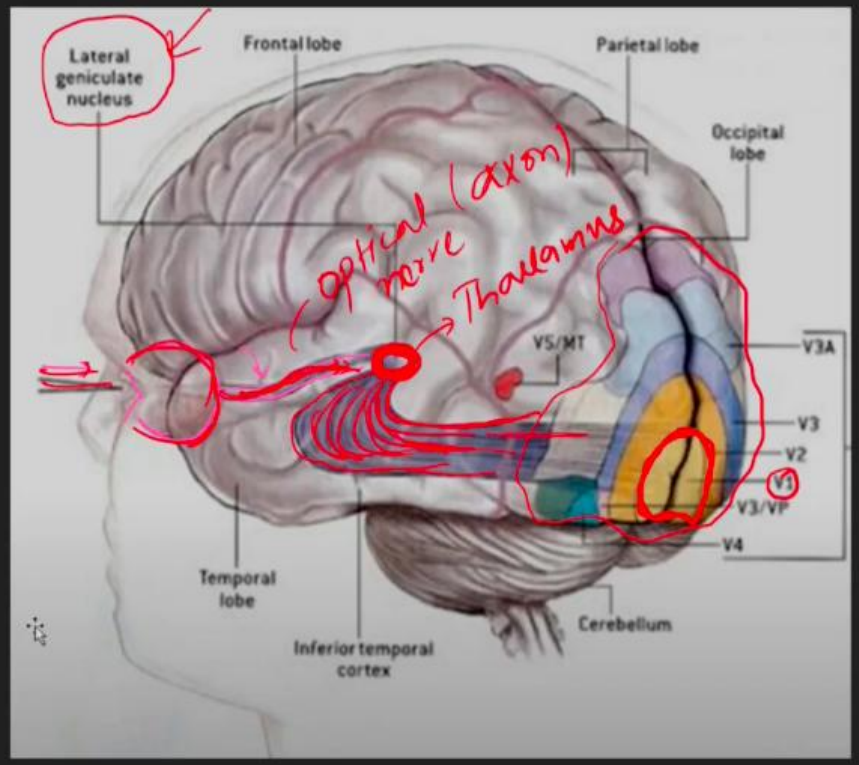
Human Visual Cortex

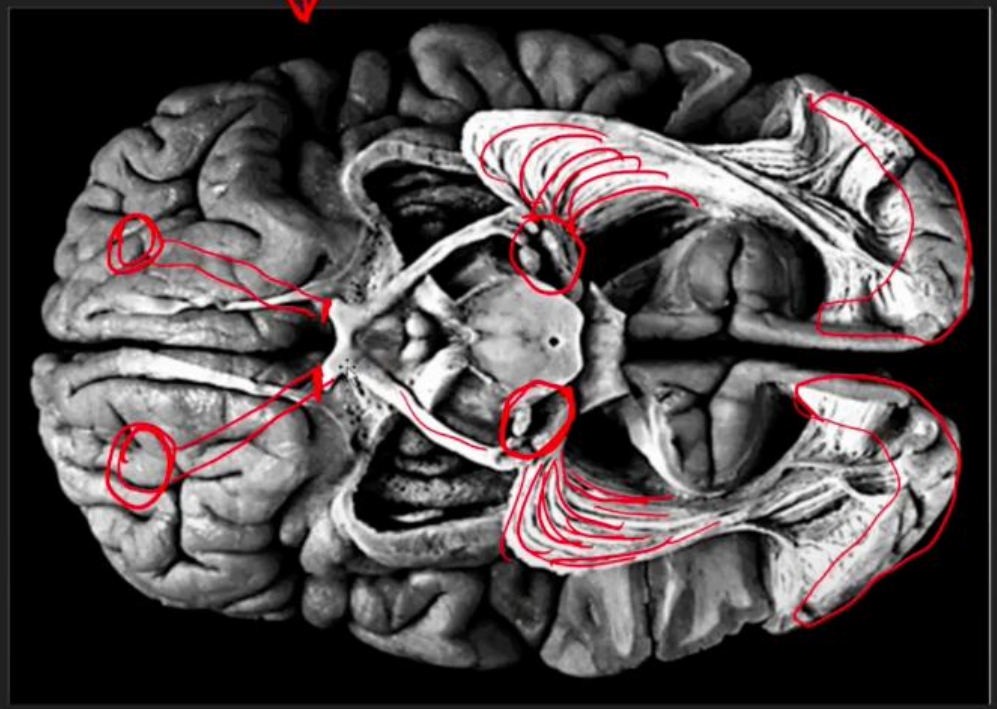
18 August 2022 16:05

2D sheet

primary V cortex

actual brain

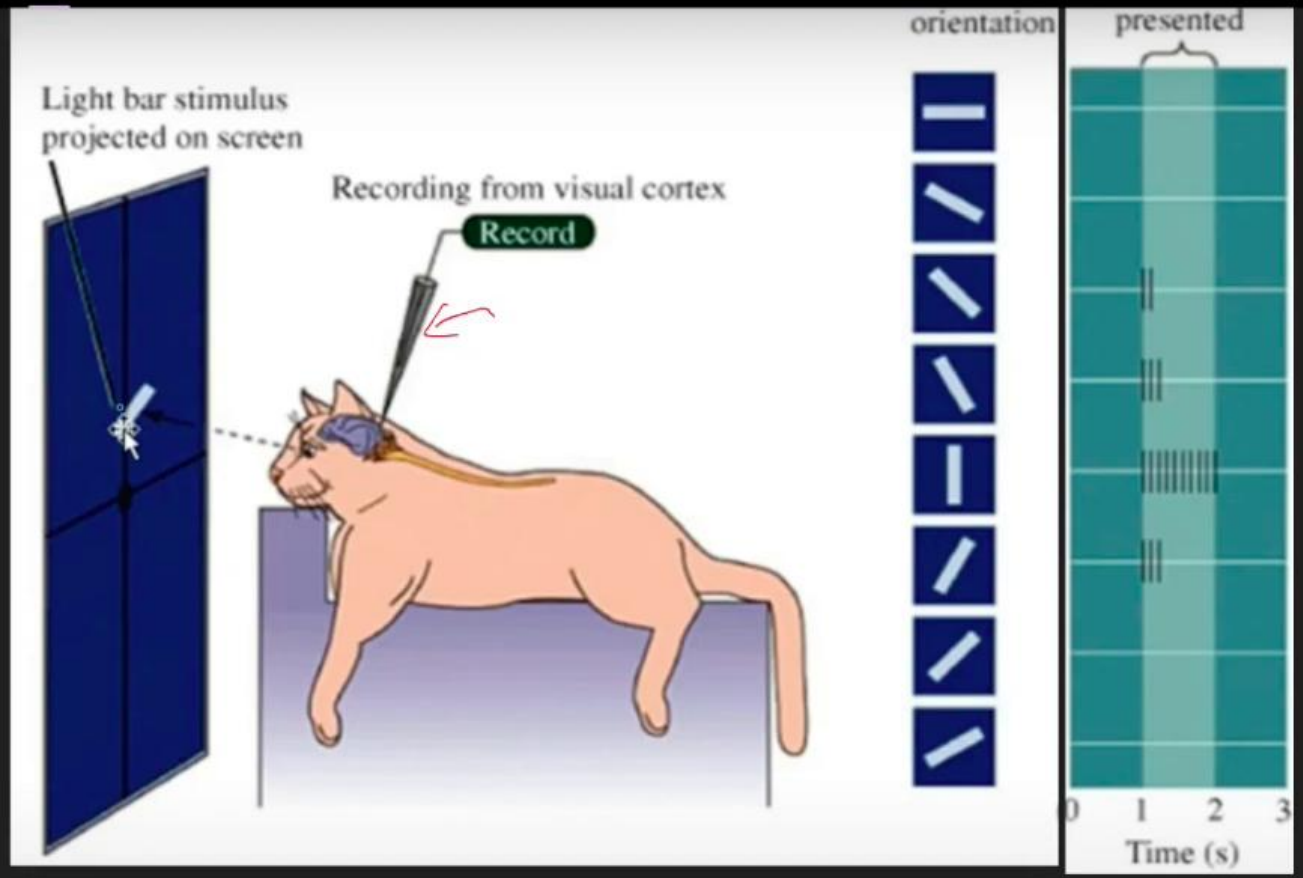




6:05 49%

D36 CNN vs Visual Cortex HOME INSERT DRAW VIEW

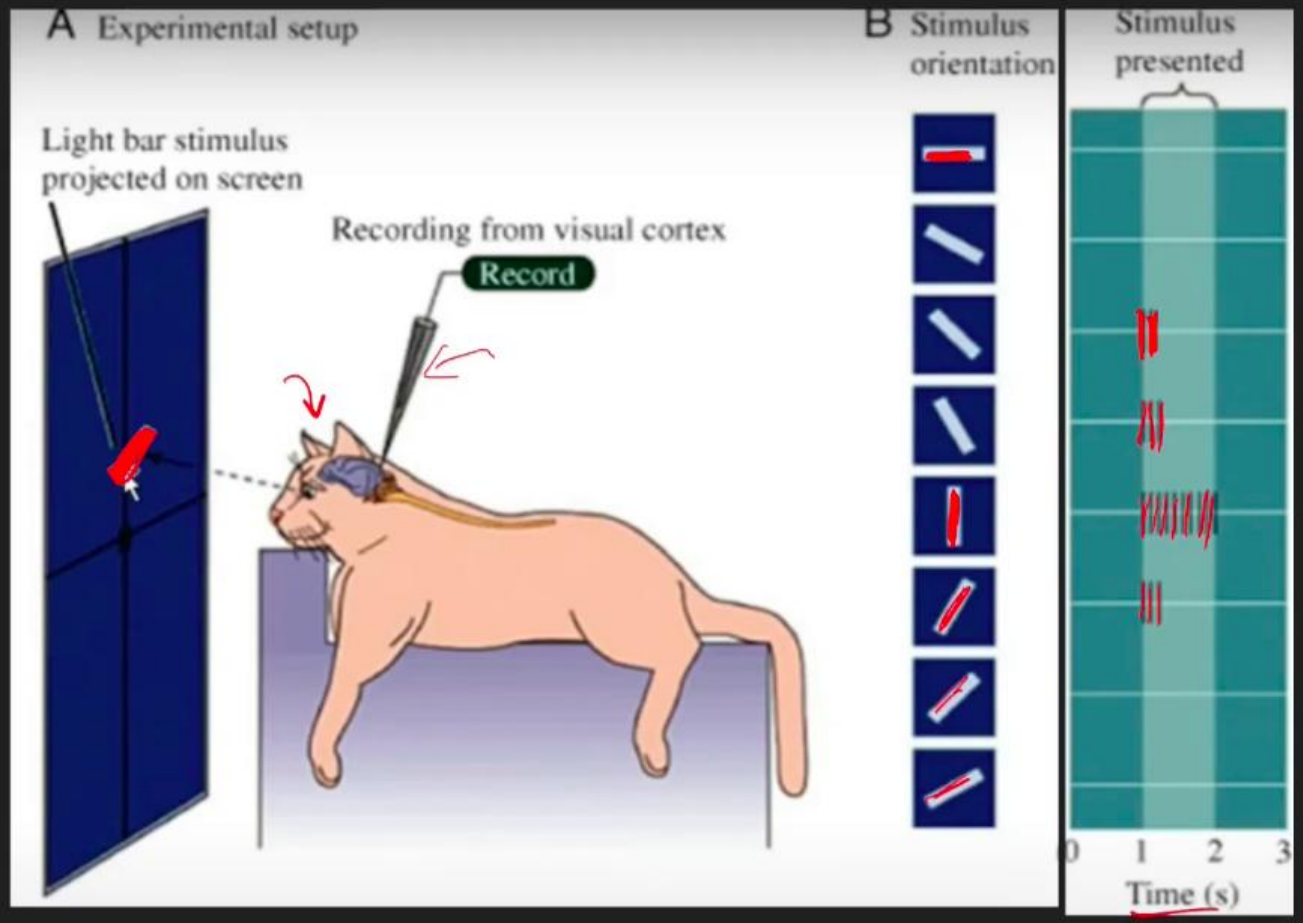
Stop



6:06 49%

D36 CNN vs Visual Cortex HOME INSERT DRAW VIEW

Stop



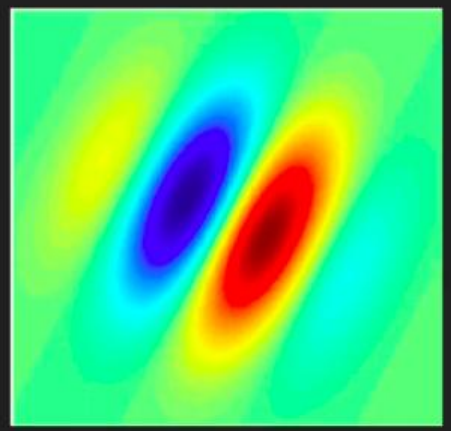
6:24 47%

D36 CNN vs Visual Cortex HOME INSERT DRAW VIEW

Stop

Conclusion

18 August 2022 17:57



Simple cell → Orientation cell → feature detect

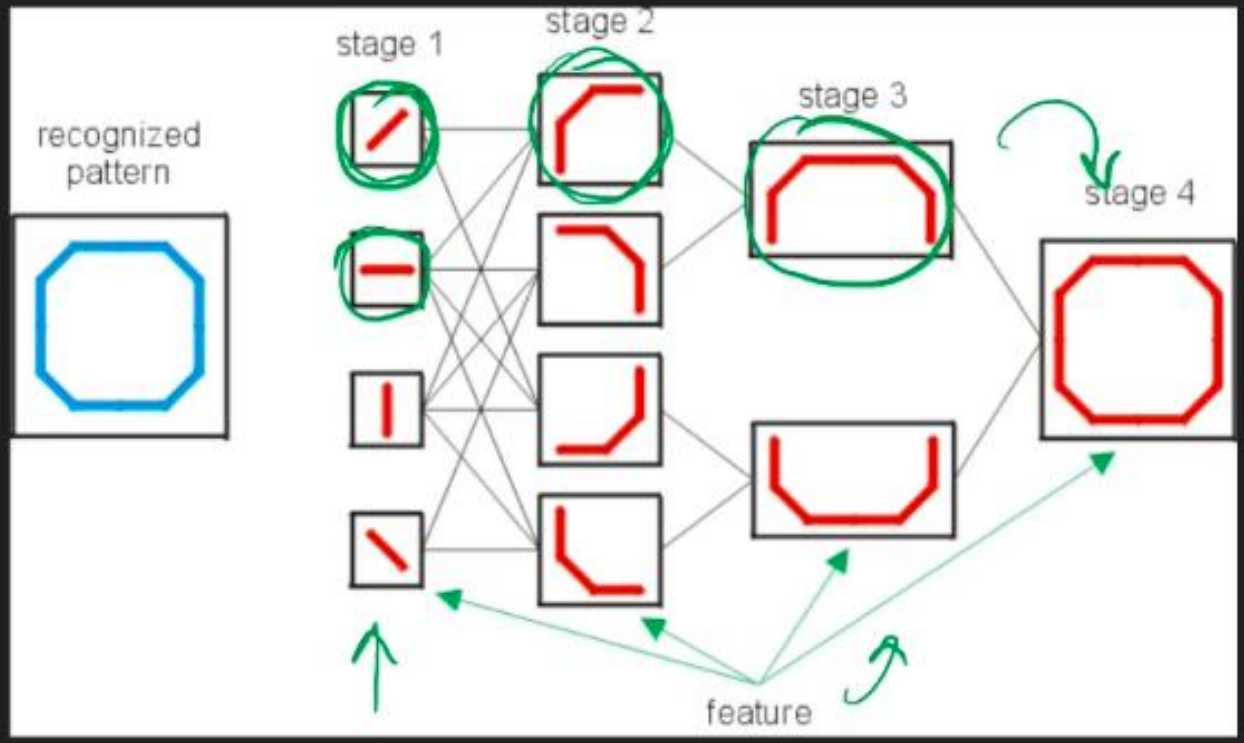
→ smaller receptive field

Complex cell

→ bigger receptive field

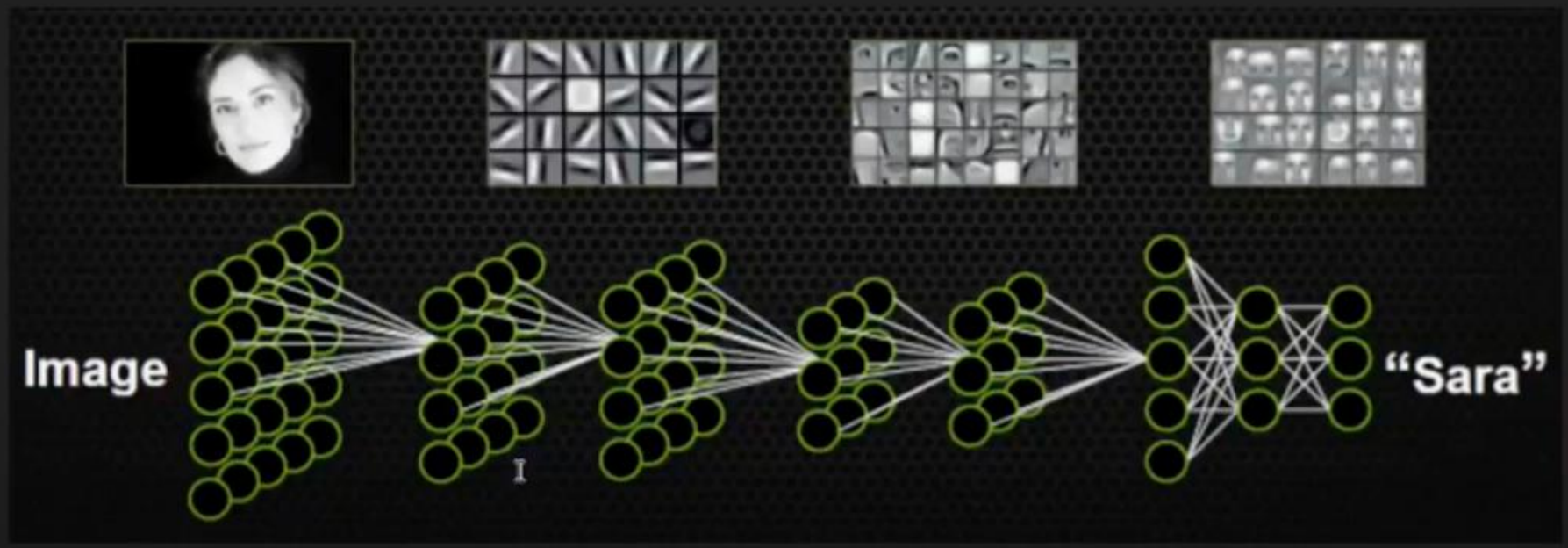
Development

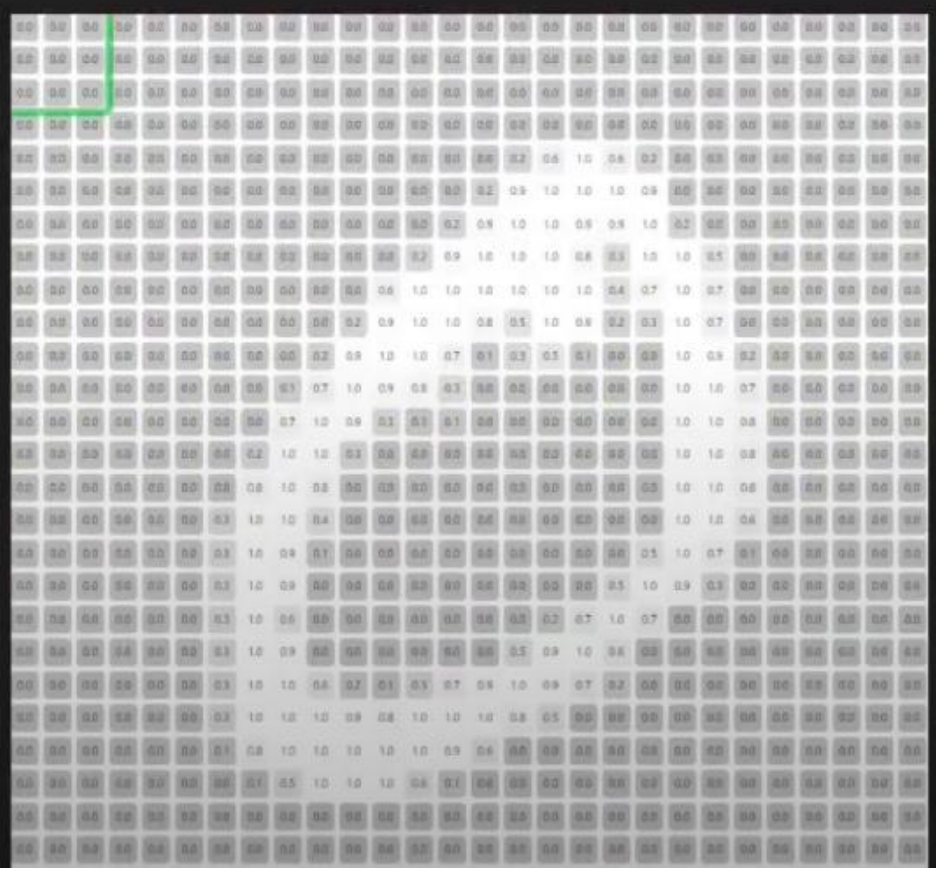
18 August 2022 16:15

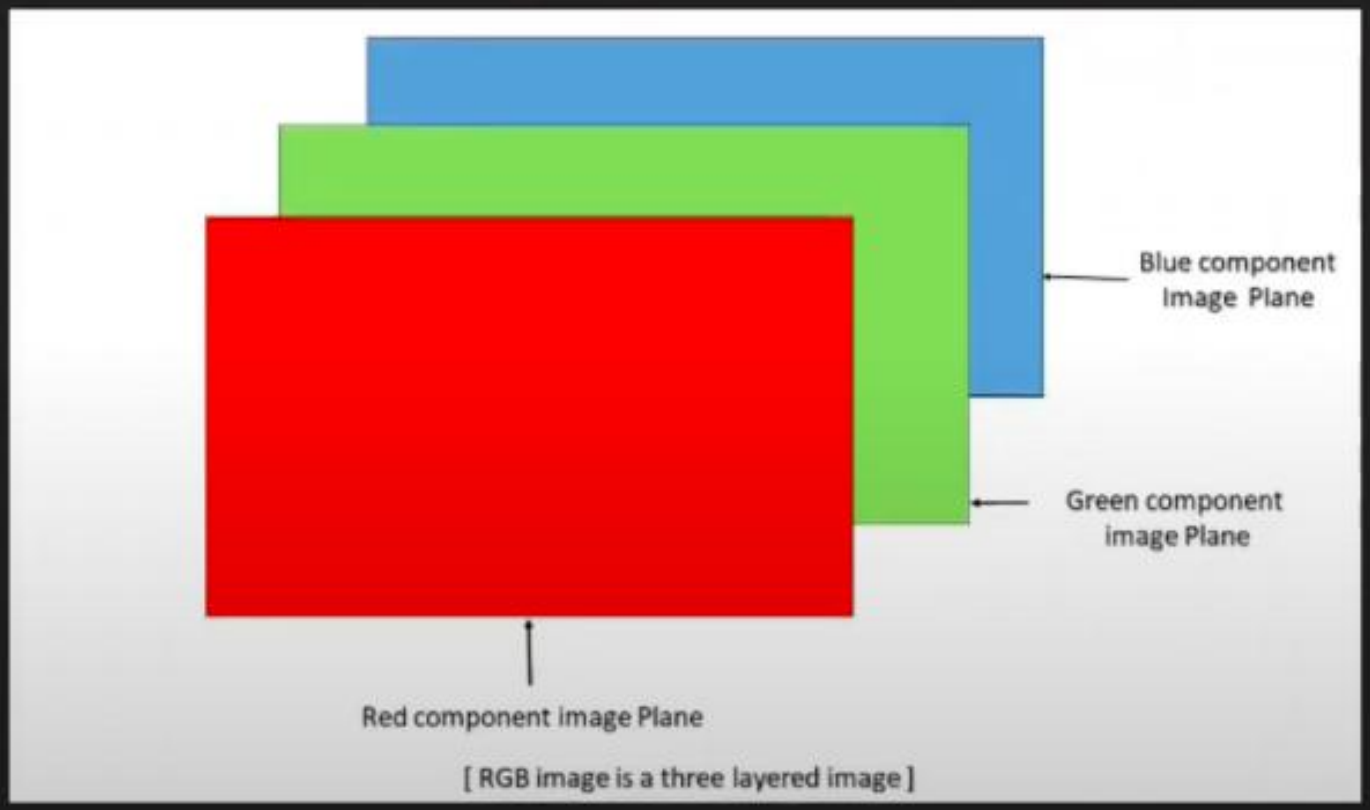


Introduction

19 August 2022 16:45

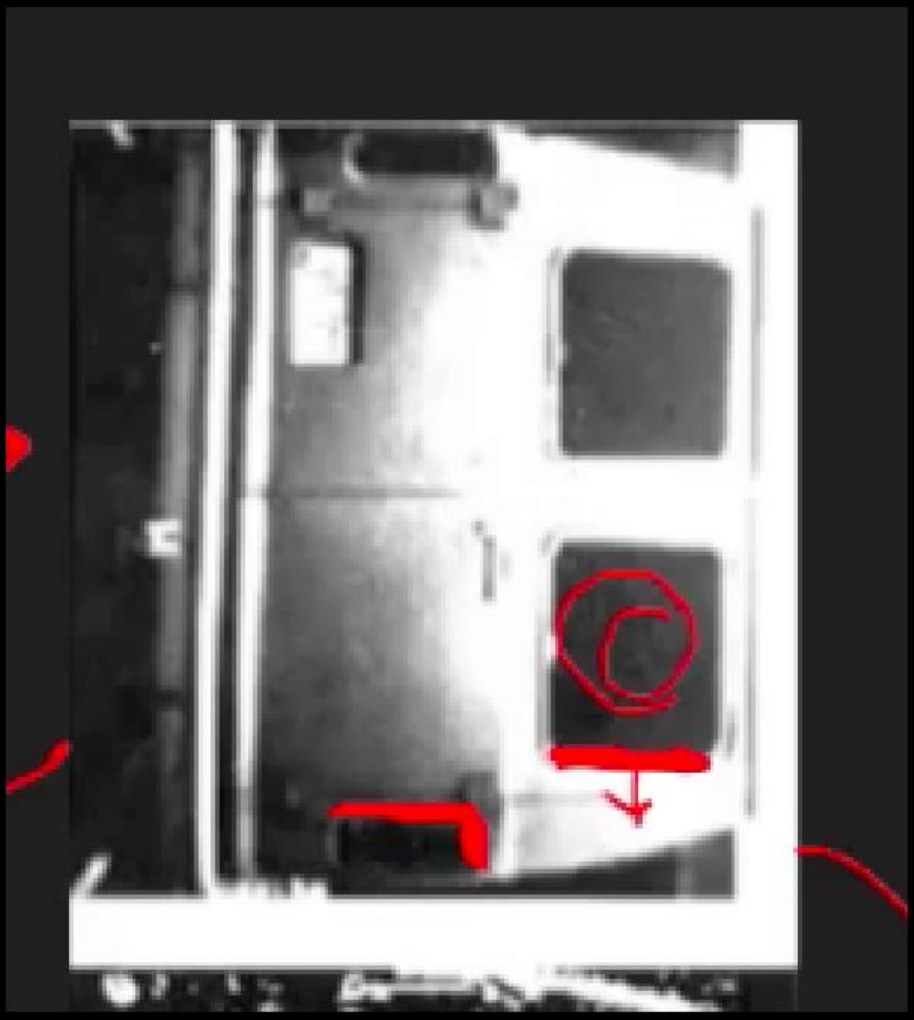





Blue component
Image Plane
Green component
image Plane
Red component image Plane
[RGB image is a three layered image]

Edge Detection (Convolution Operation)

19 August 2022 16:53



0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

*

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

=

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

*

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

=

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

*

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

=

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

*

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

=

0	0	0	0

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

*

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

=

0	0	0	0
255	255	255	255

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

*

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

=

0	0	0	0
255	255	255	255
255	255	255	255

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

*

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

=

0	0	0	0
255	255	255	255
255	255	255	255
0	0	0	0

End-to-End-DL - Google Sheets | (6) Session 1 on CNN | Batch 6 | deeplizard - Convolution Demo | (8) C4W1L07 One Layer of a Con

deeplizard.com/resource/pavq7noze2

CONVOLUTION OPERATION DEMO

Application

Full Screen | MNIST | 0 - Zero

Left Edge Filter | Step | Play

Window

0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

*

Filter

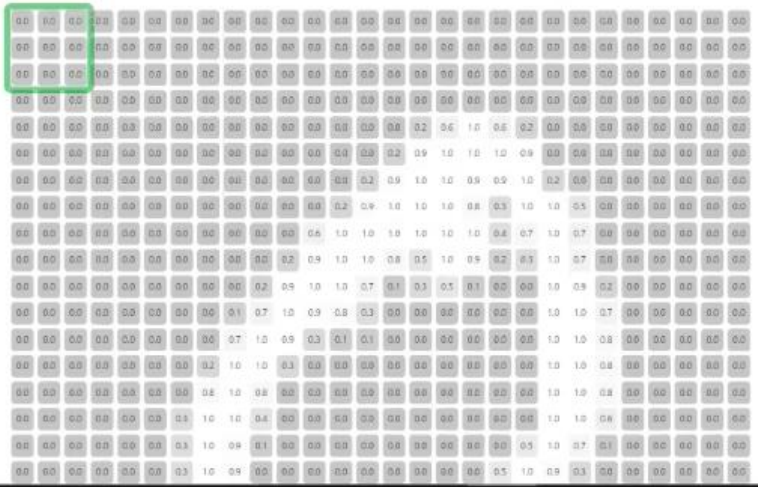
-1.0	1.0	0.0
-1.0	1.0	0.0
-1.0	1.0	0.0

=

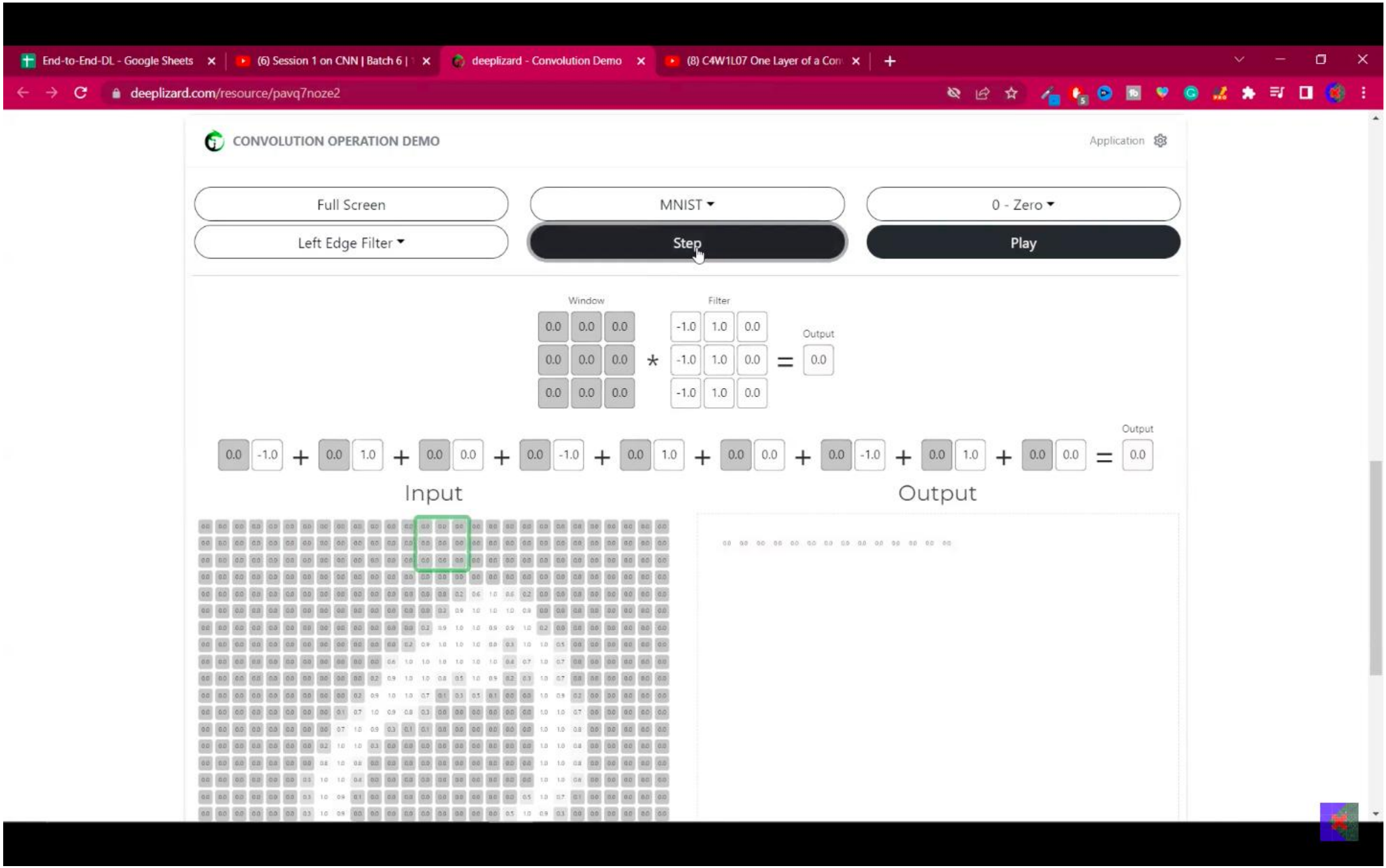
Output: 0.0

0.0 -1.0 + 0.0 1.0 + 0.0 0.0 + 0.0 -1.0 + 0.0 1.0 + 0.0 0.0 + 0.0 -1.0 + 0.0 1.0 + 0.0 0.0 = Output 0.0

Input



Output


End-to-End-DL - Google Sheets
(6) Session 1 on CNN | Batch 6 |
deeplizard - Convolution Demo
(8) C4W1L07 One Layer of a Con
deeplizard.com/resource/pavq7noze2
CONVOLUTION OPERATION DEMO
Application
Full Screen
MNIST
0 - Zero
Left Edge Filter
Step
Play
Window
Filter
Output
Input
Output

End-to-End-DL - Google Sheets | (6) Session 1 on CNN | Batch 6 | deeplizard - Convolution Demo | (8) C4W1L07 One Layer of a Con

deeplizard.com/resource/pavq7noze2

CONVOLUTION OPERATION DEMO

Application

Full Screen | MNIST | 0 - Zero

Left Edge Filter | Step | Play

Window

0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

*

Filter

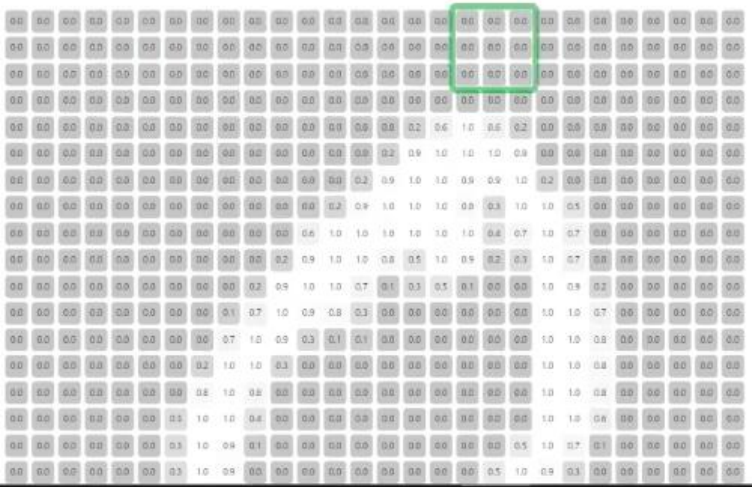
-1.0	1.0	0.0
-1.0	1.0	0.0
-1.0	1.0	0.0

=

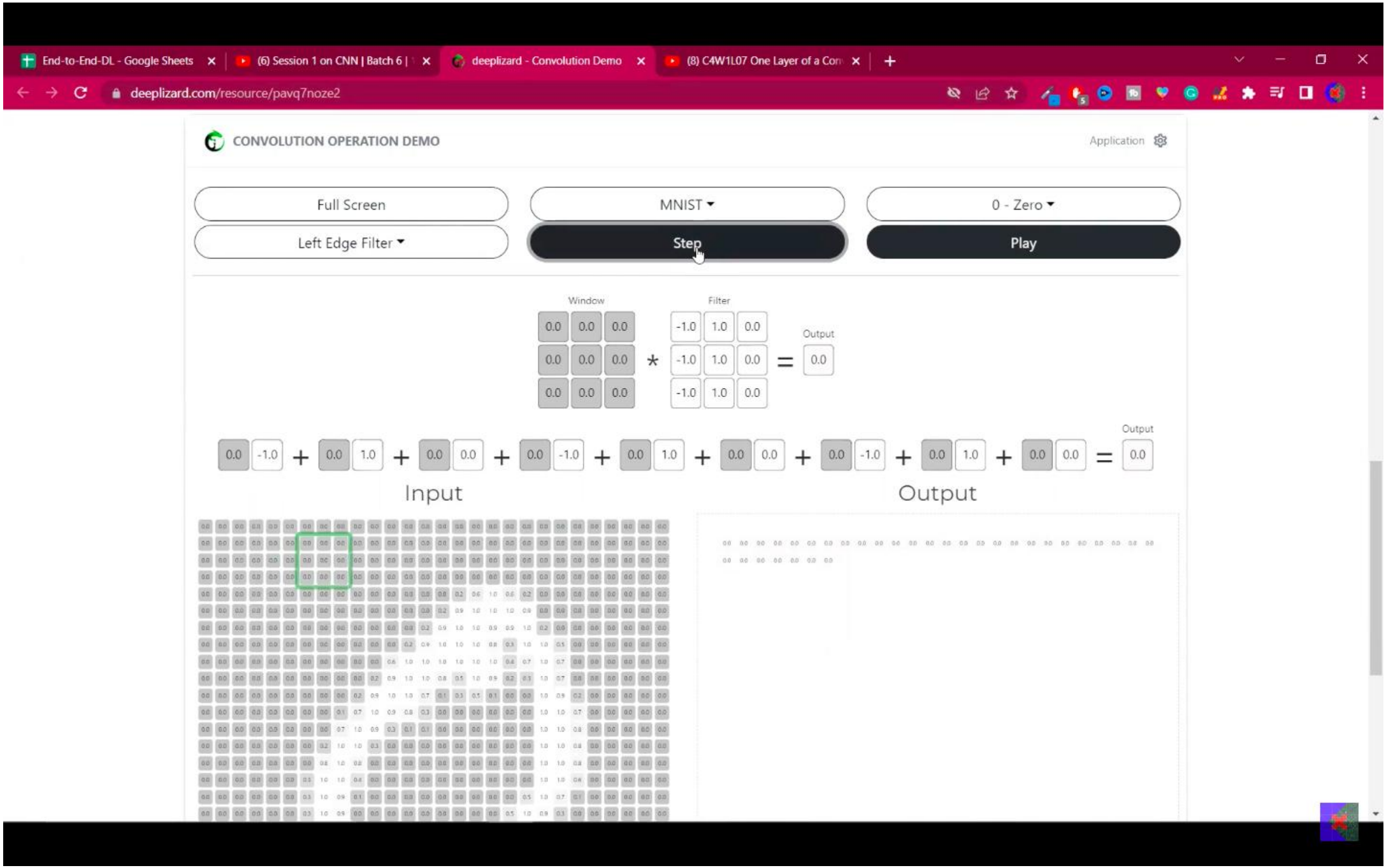
Output: 0.0

0.0 -1.0 + 0.0 1.0 + 0.0 0.0 + 0.0 -1.0 + 0.0 1.0 + 0.0 0.0 + 0.0 -1.0 + 0.0 1.0 + 0.0 0.0 = Output 0.0

Input



Output


End-to-End-DL - Google Sheets
(6) Session 1 on CNN | Batch 6 |
deeplizard - Convolution Demo
(8) C4W1L07 One Layer of a Con
deeplizard.com/resource/pavq7noze2
CONVOLUTION OPERATION DEMO
Application
Full Screen
MNIST
0 - Zero
Left Edge Filter
Step
Play
Window
Filter
Output
Input
Output

End-to-End-DL - Google Sheets | (6) Session 1 on CNN | Batch 6 | deeplizard - Convolution Demo | (8) C4W1L07 One Layer of a Con

deeplizard.com/resource/pavq7noze2

CONVOLUTION OPERATION DEMO

Application

Full Screen | MNIST | 0 - Zero

Left Edge Filter | Step | Play

Window

0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0

*

Filter

-1.0	1.0	0.0
-1.0	1.0	0.0
-1.0	1.0	0.0

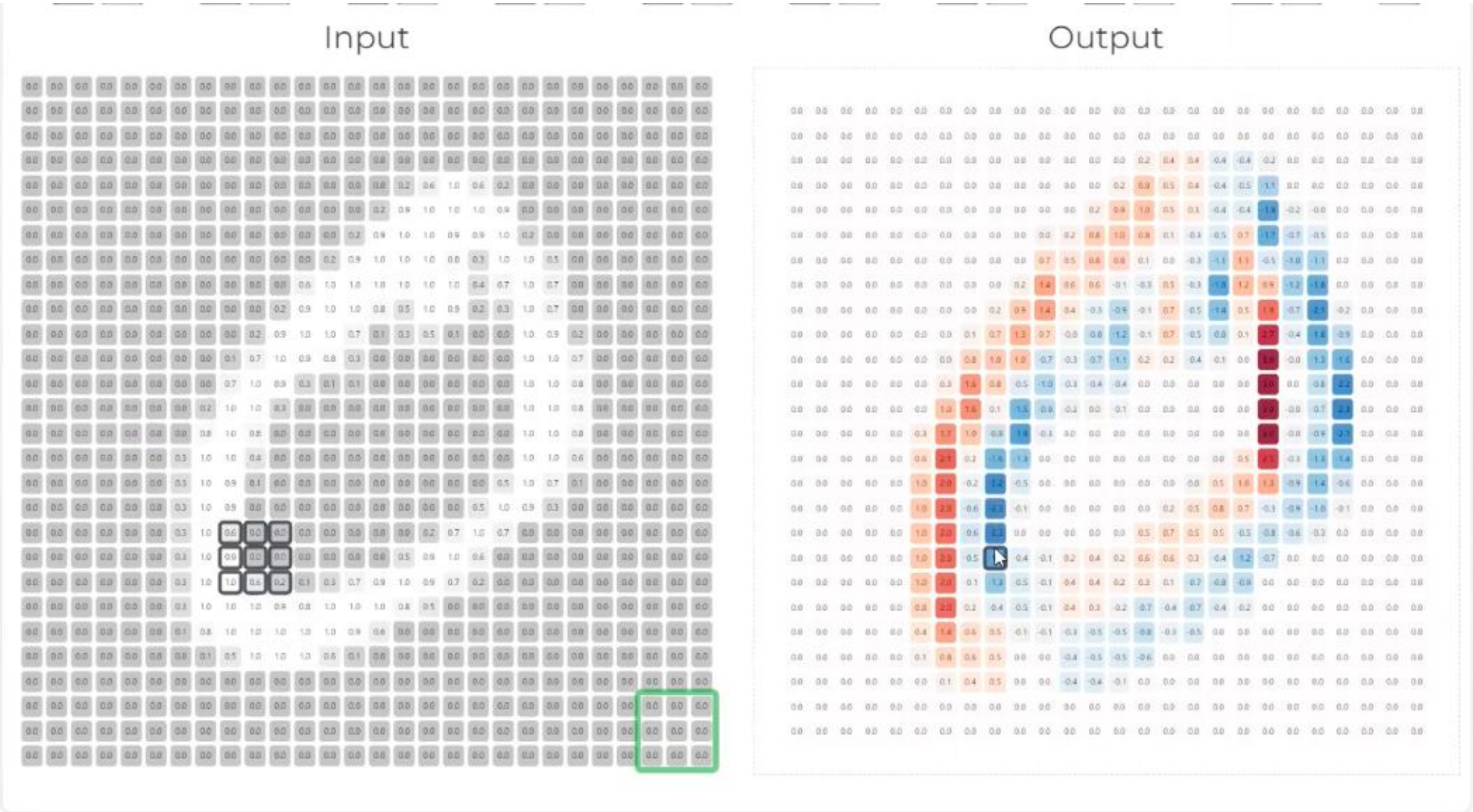
=

Output: 0.0

0.0 -1.0 + 0.0 1.0 + 0.0 0.0 + 0.0 -1.0 + 0.0 1.0 + 0.0 0.0 + 0.0 -1.0 + 0.0 1.0 + 0.0 0.0 = Output 0.0

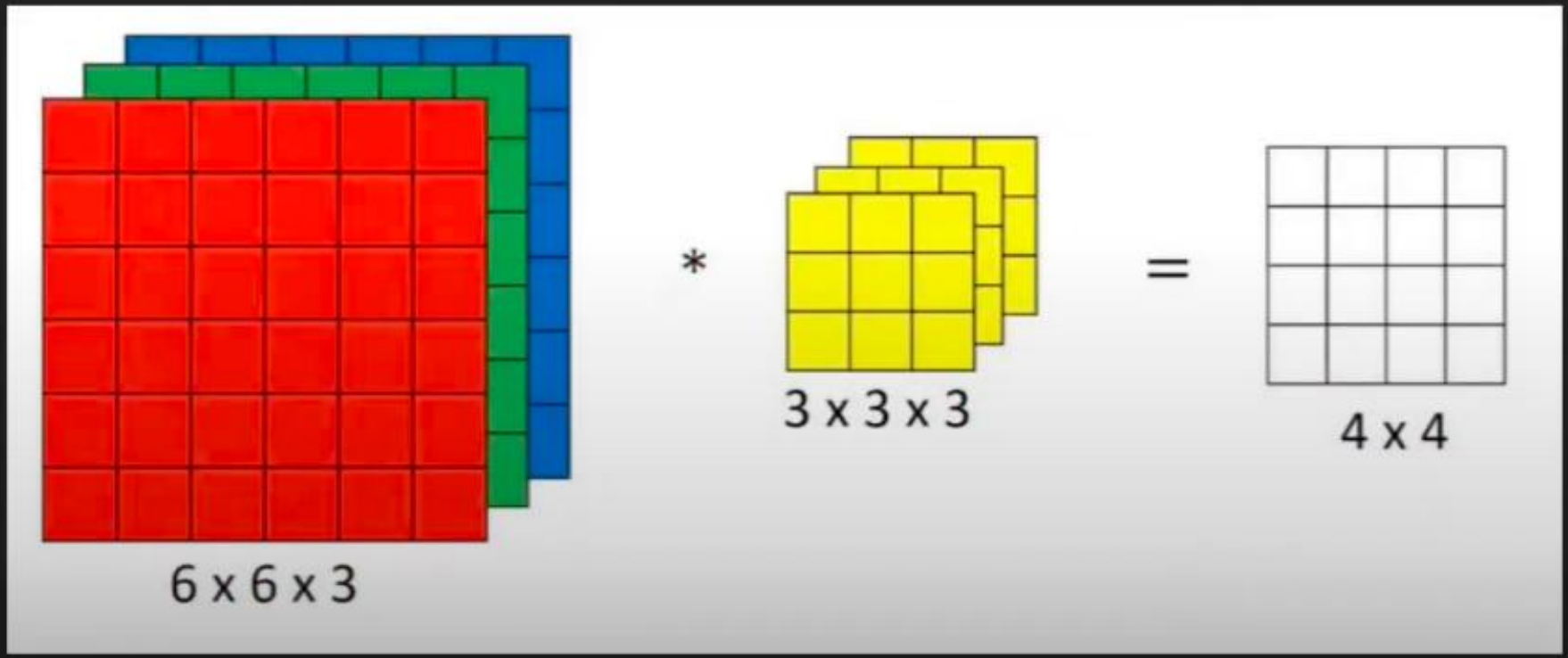
Input

Output

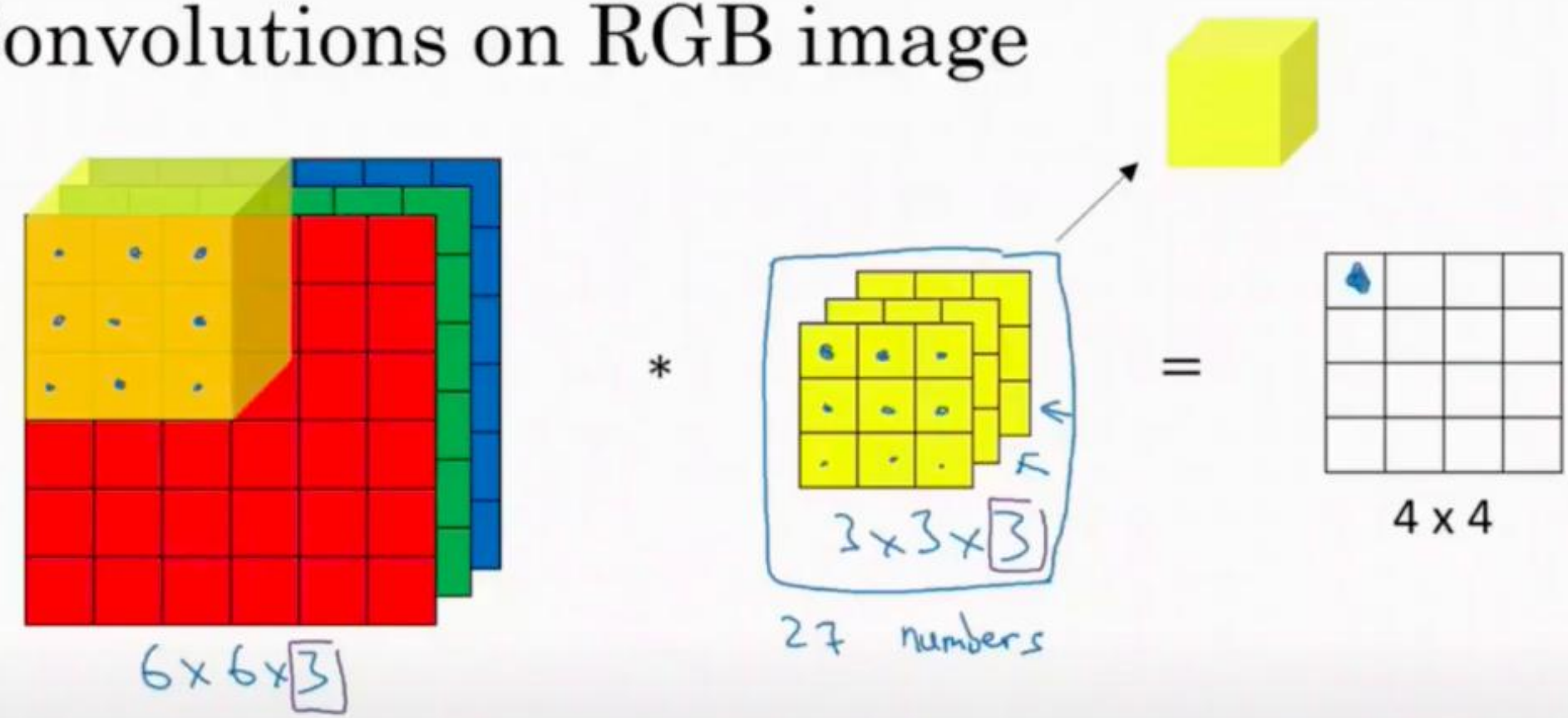

Input
Output

Working with RGB Images

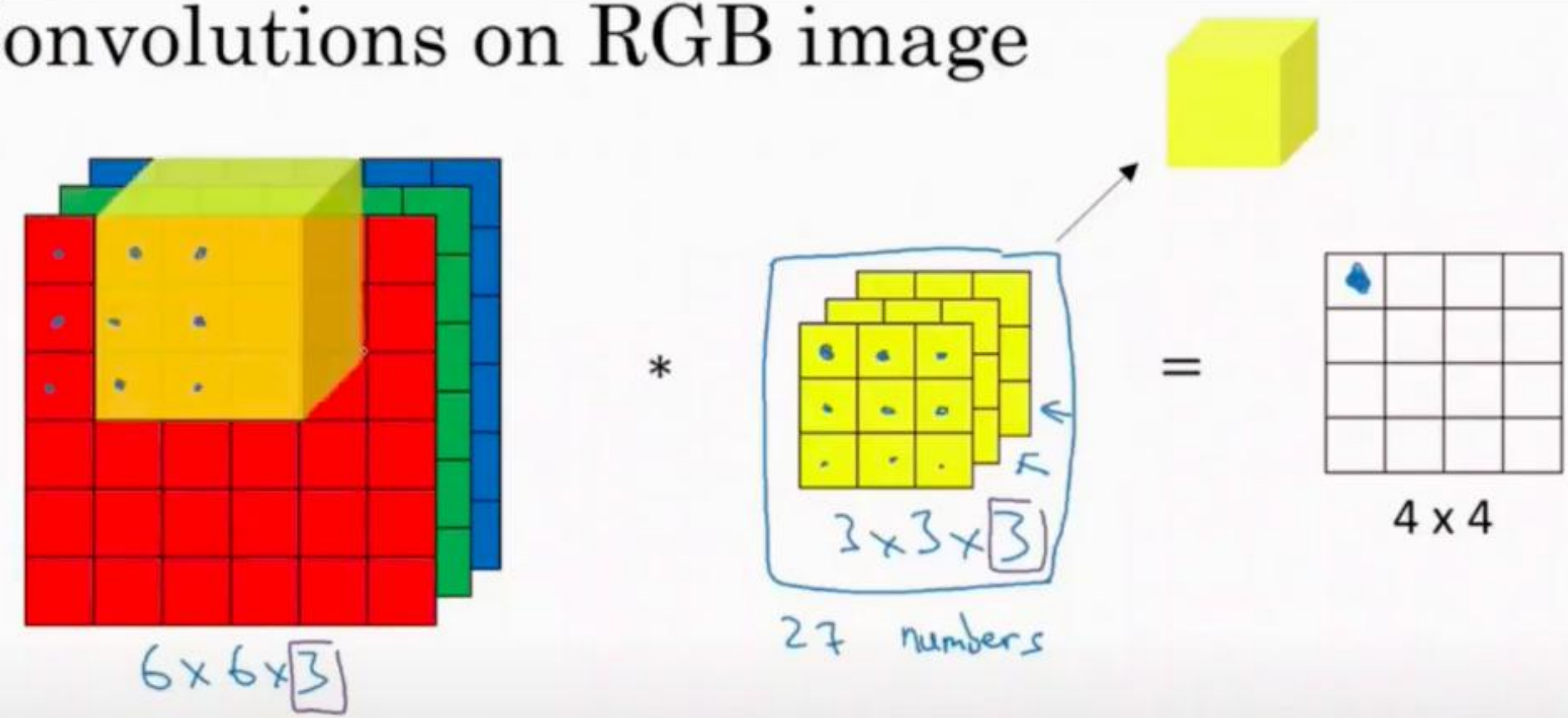
19 August 2022 16:54



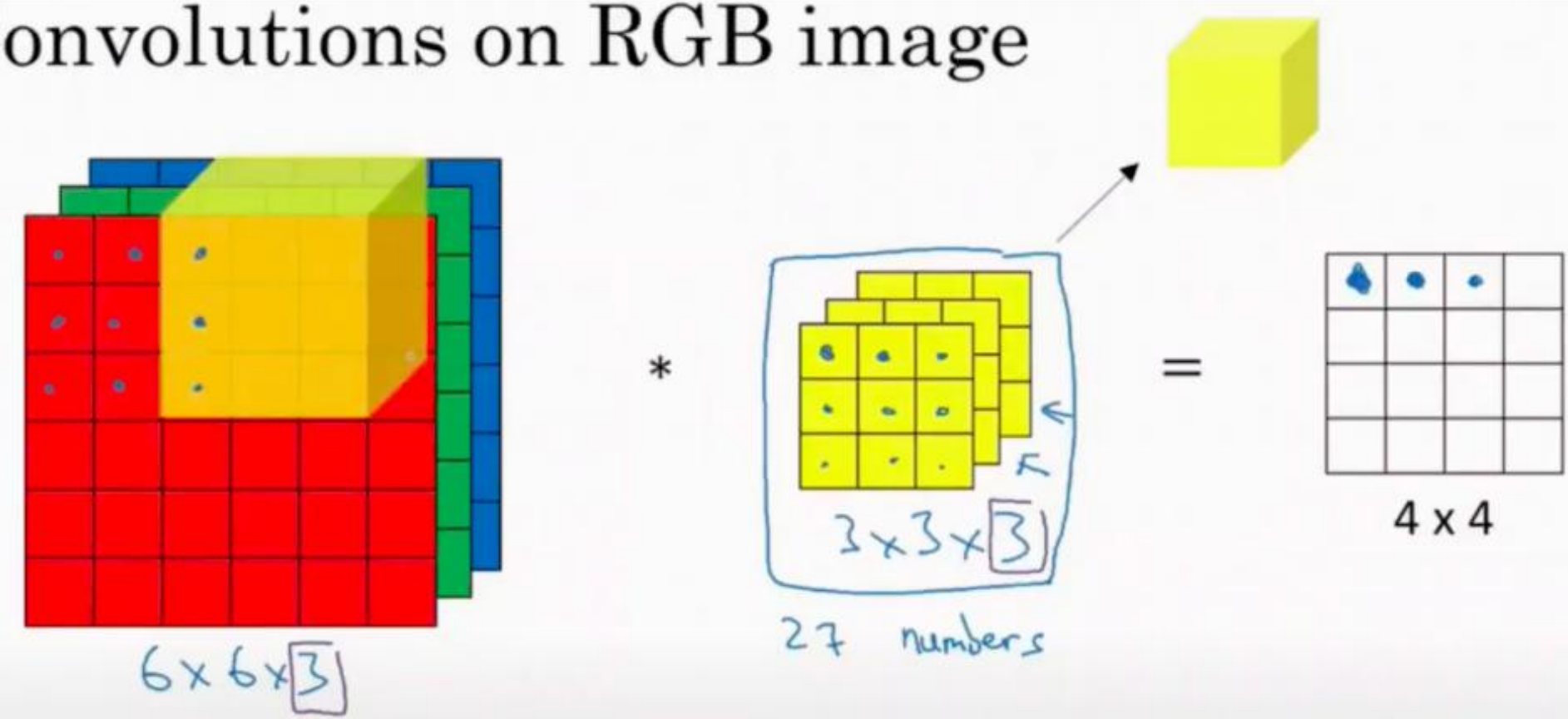
Convolutions on RGB image



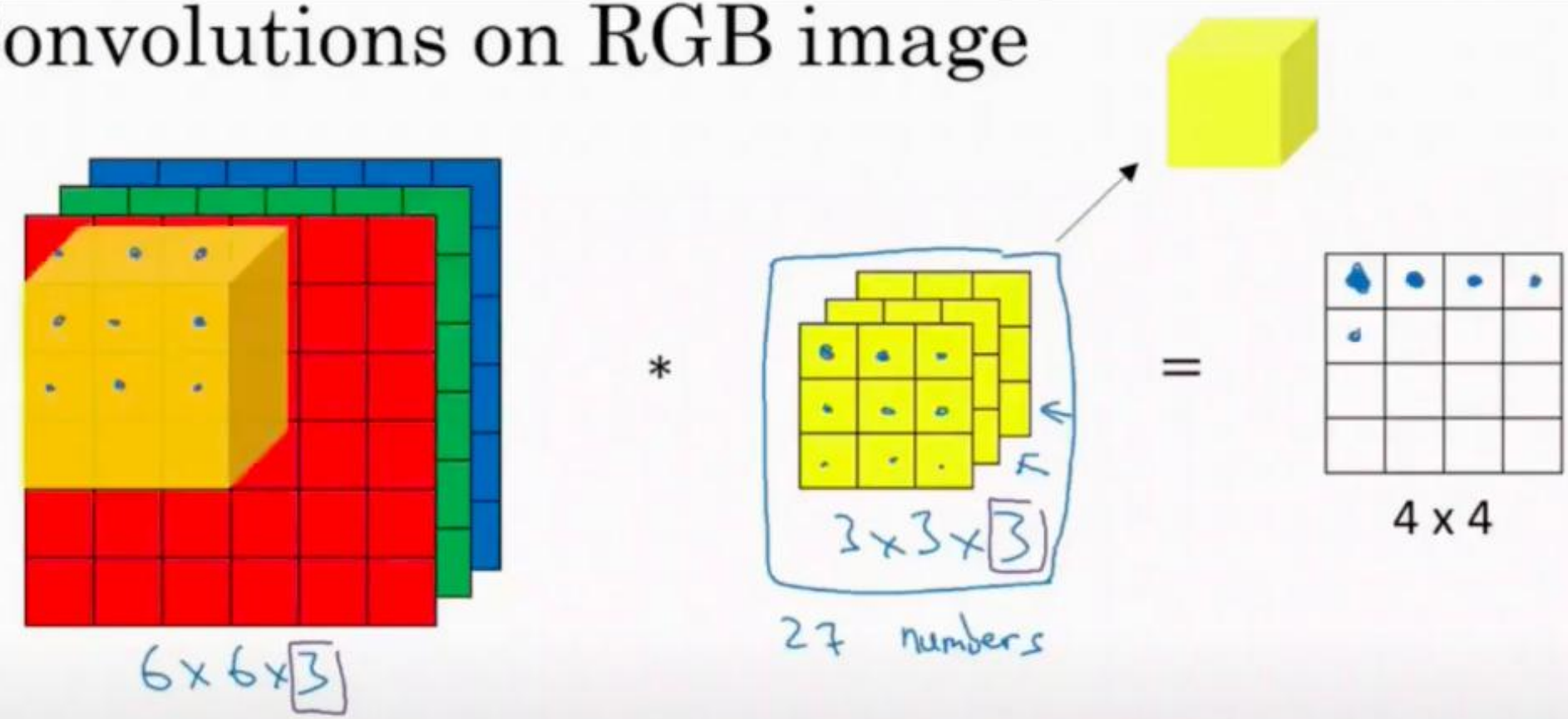
Convolutions on RGB image

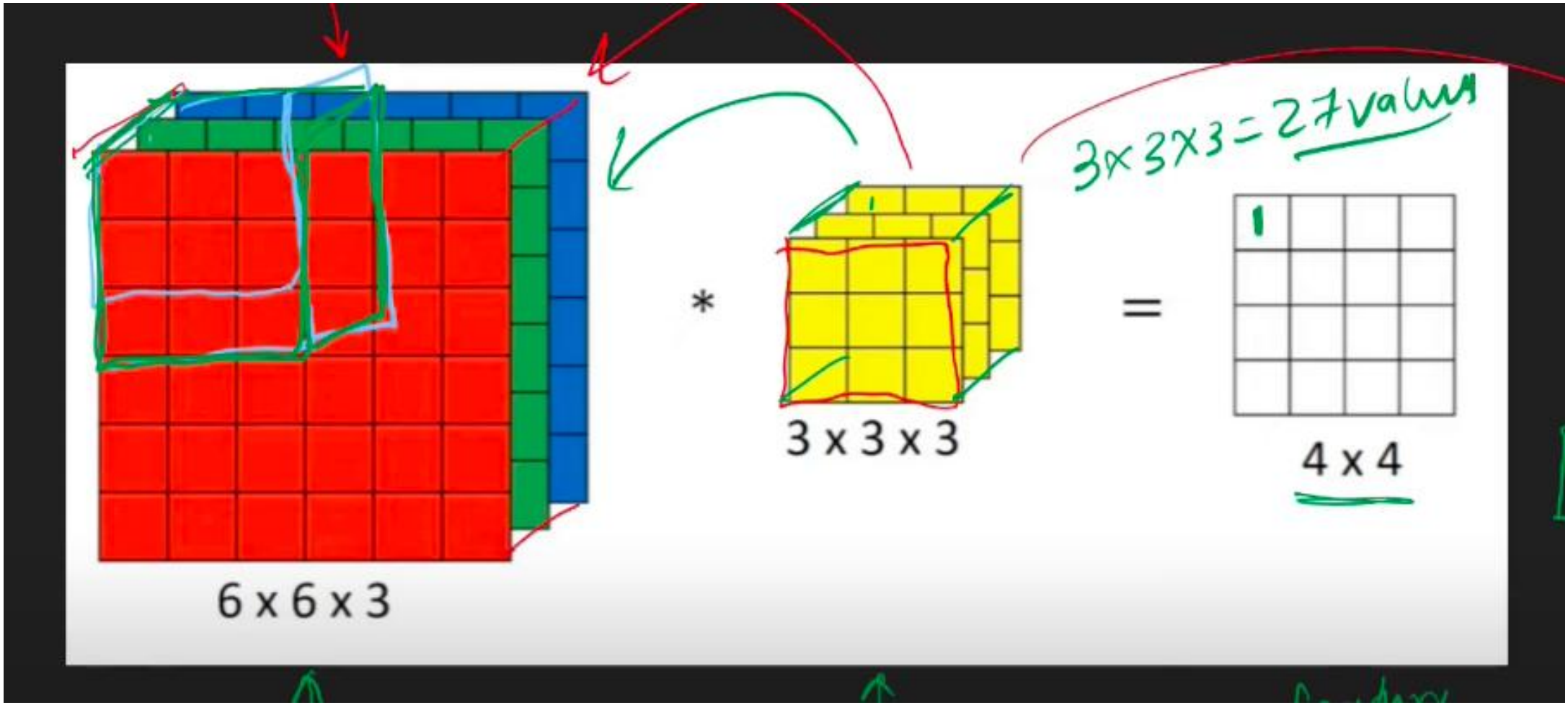


Convolutions on RGB image



Convolutions on RGB image




3x3x3=27values
6 x 6 x 3
*
3 x 3 x 3
=
1
4 x 4

Multiple Filters

23 August 2022 08:24

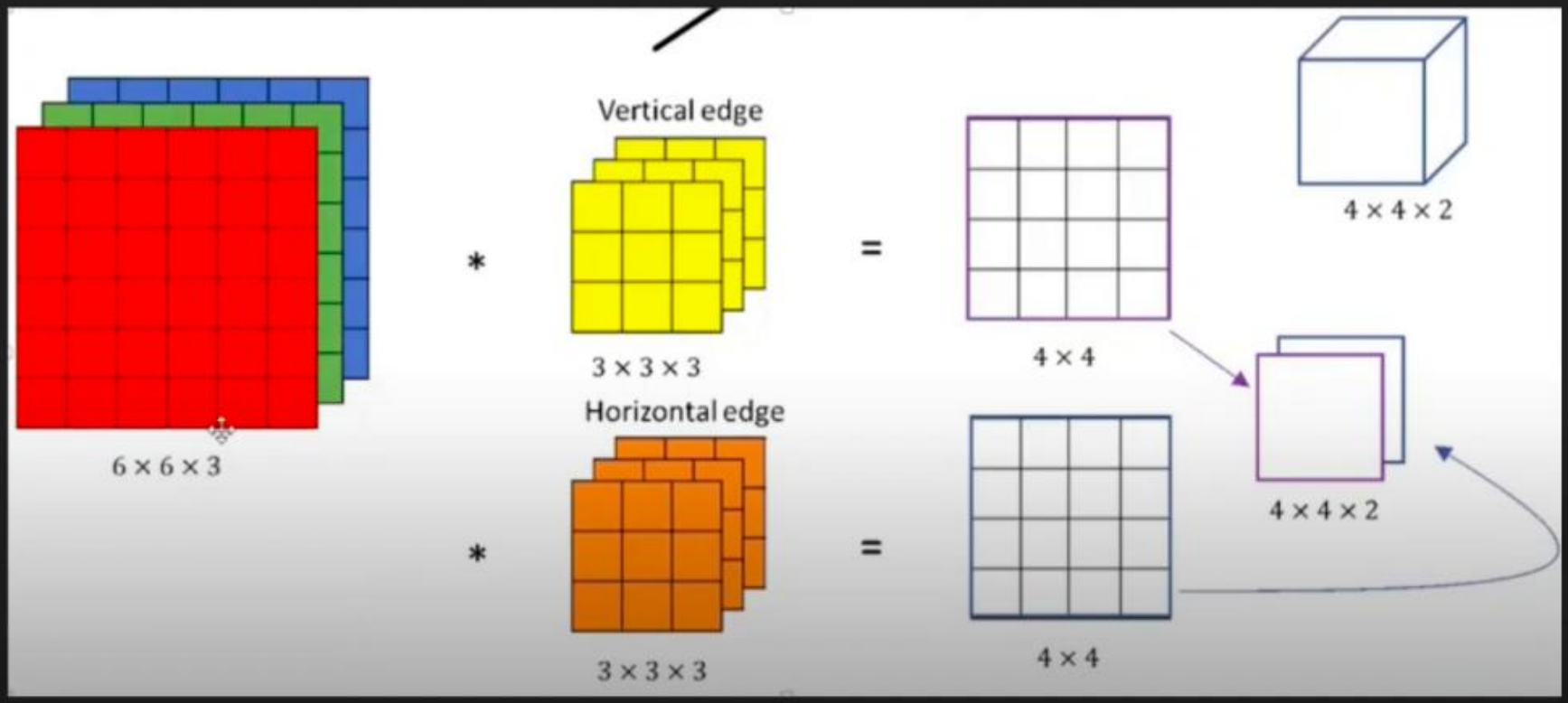


Image taken from Andrew NG's lecture

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

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

=

6		

7x1+4x1+3x1+
2x0+5x0+3x0+
3x-1+3x-1+2x-1
= 6

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

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

=

6	-9	-8
-3	-2	

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

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

=

6	-9	-8
-3	-2	-3
-3		

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

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

=

6	-9	-8
-3	-2	-3
-3	0	-2

What is Padding?

26 August 2022 14:26

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

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

=

6		

7x1+4x1+3x1+
2x0+5x0+3x0+
3x-1+3x-1+2x-1
= 6

$n \times n$

5×5

$f \times f$ $\rightarrow$ $(n-f+1)\ (n-f+1)$

3×3

$(5-3+1) = 3 \times 3$

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

0	-1	0
-1	5	-1
0	-1	0

114	328	-26		

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

0	-1	0
-1	5	-1
0	-1	0

114	328	-26	470	158
53	266	-61	-30	

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

0	-1	0
-1	5	-1
0	-1	0

114	328	-26	470	158
53	266	-61	-30	344
403	116	-47	295	244

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

0	-1	0
-1	5	-1
0	-1	0

114	328	-26	470	158
53	266	-61	-30	344
403	116	-47	295	244
108	-135	256	-128	344
314	346	279	153	421

Strides

27 August 2022 08:23

Stride = 1

Ipixel

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

7x7

0	0	1
1	0	0
0	1	1

3x3

(1, 1)

right

bottom

Strides

27 August 2022 08:23

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

7x7

Stride = 1

0	0	1
1	0	0
0	1	1

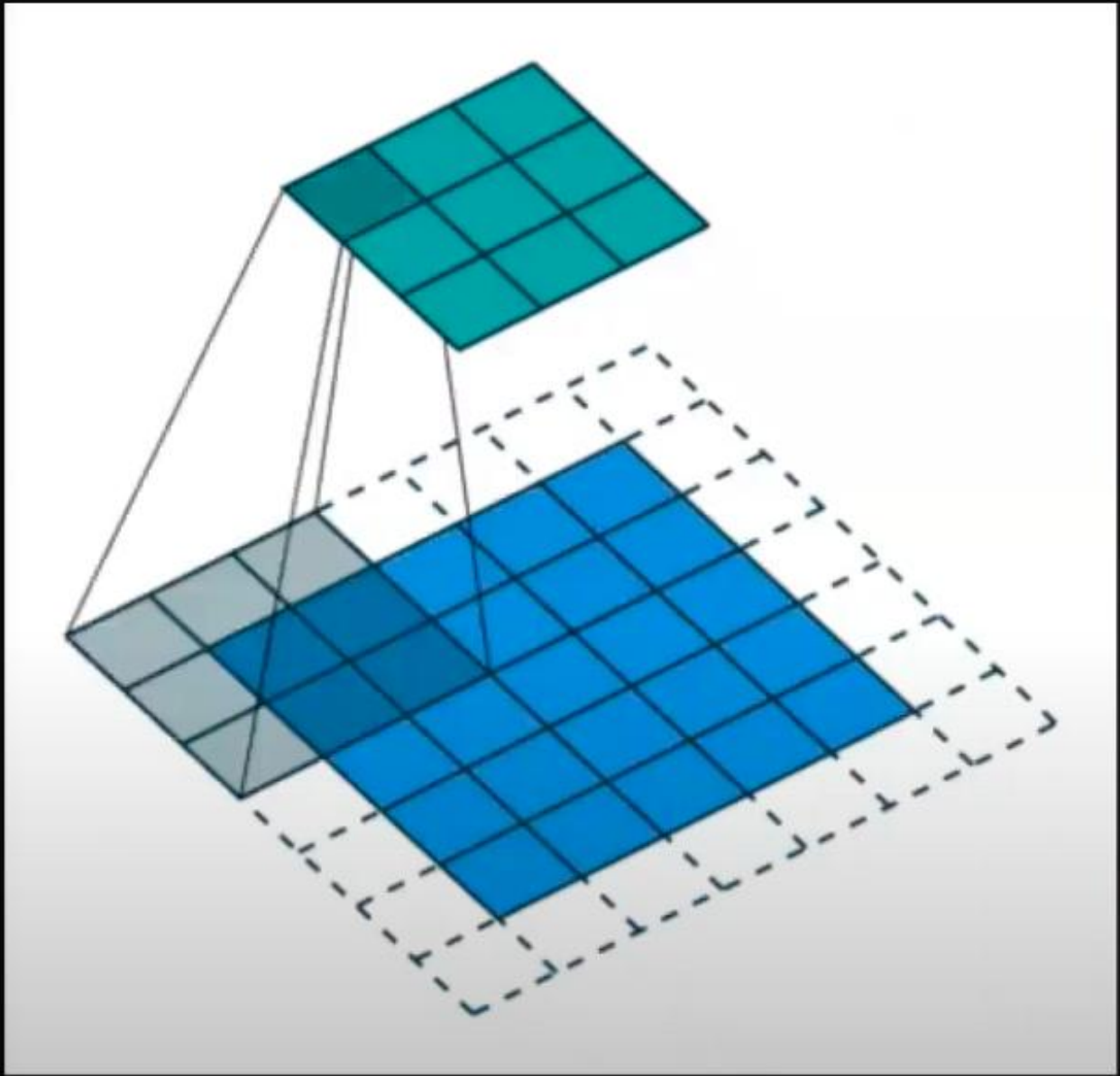
3x3

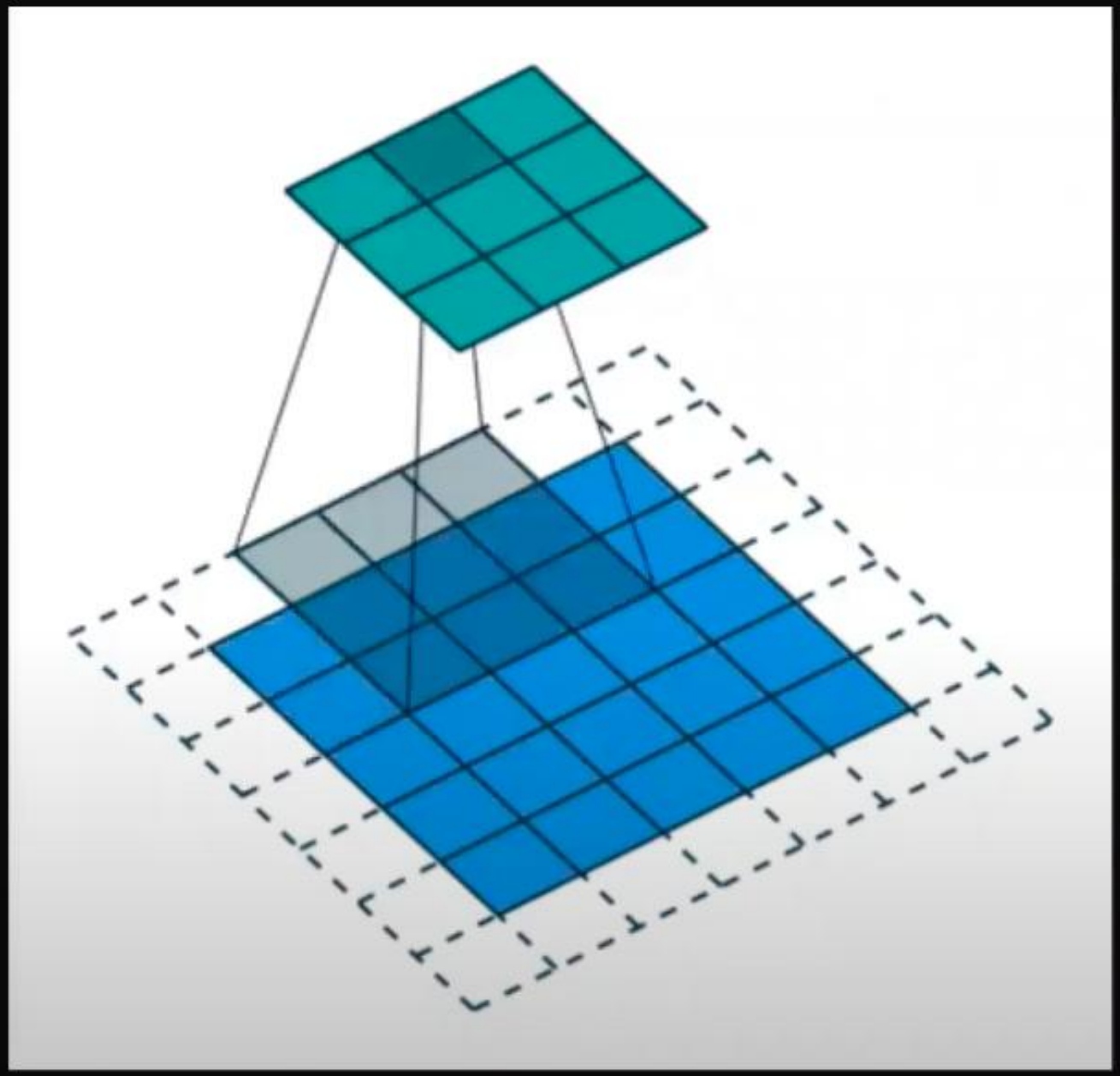
(1, 1)

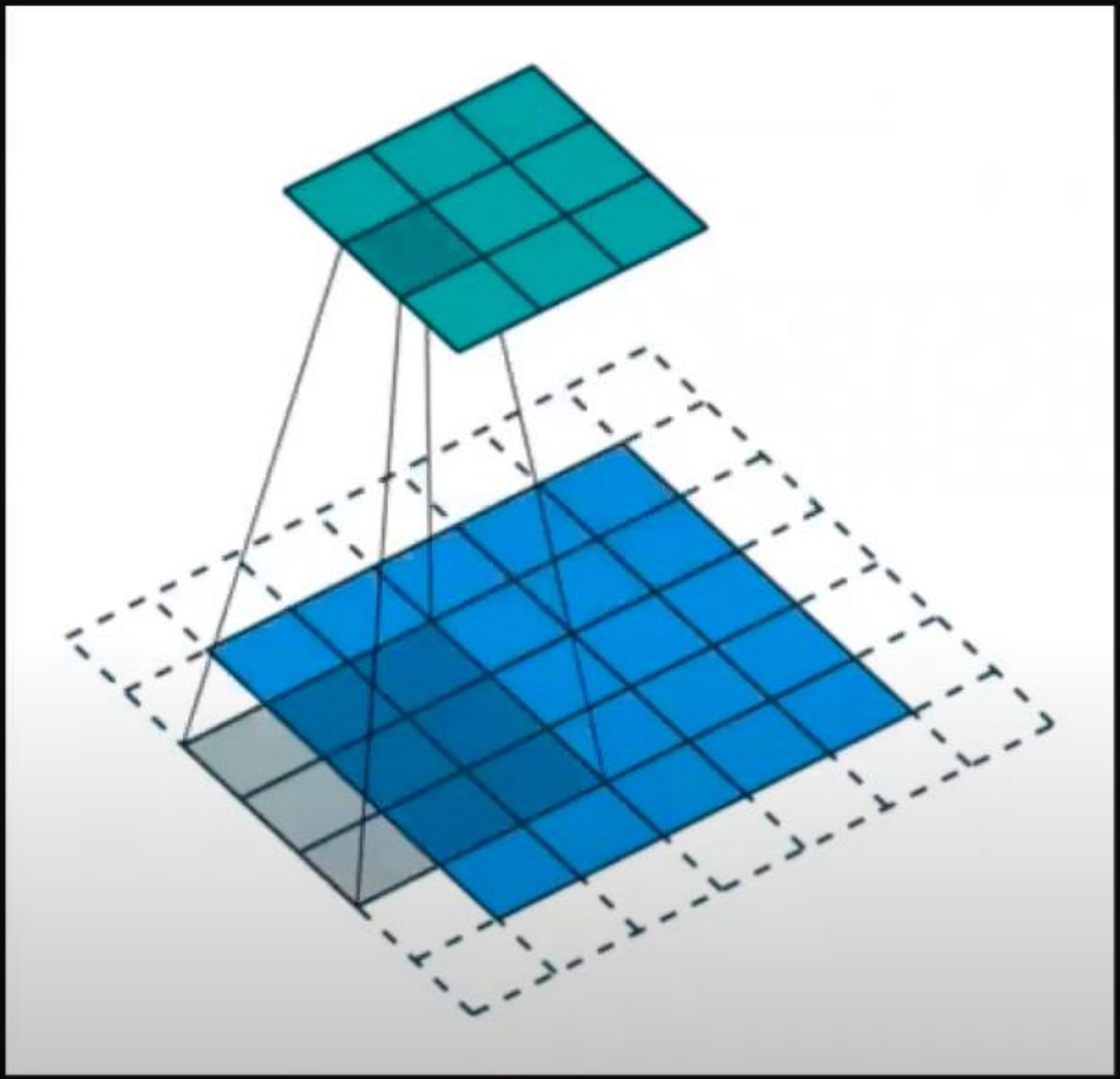
right

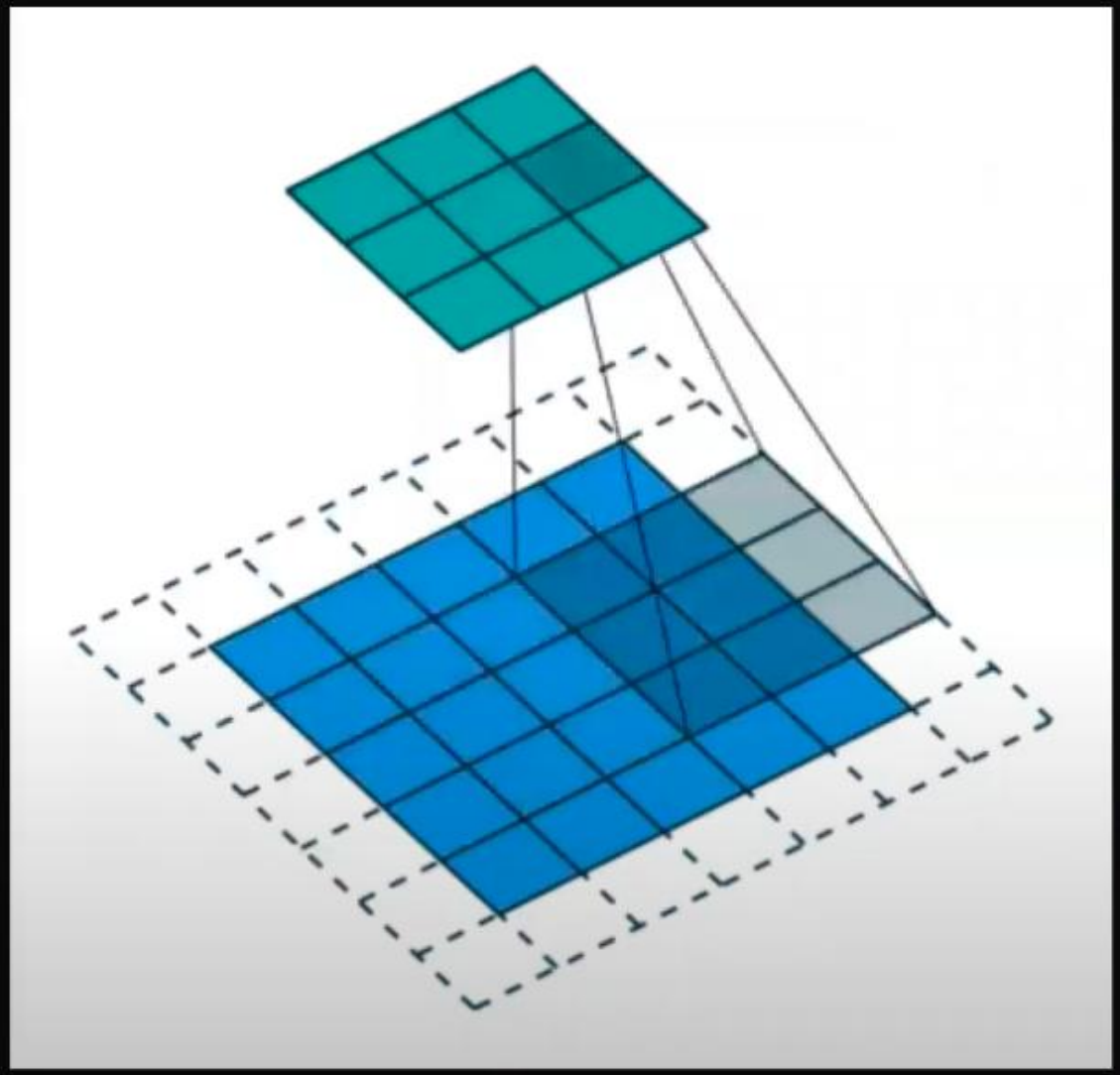
bottom

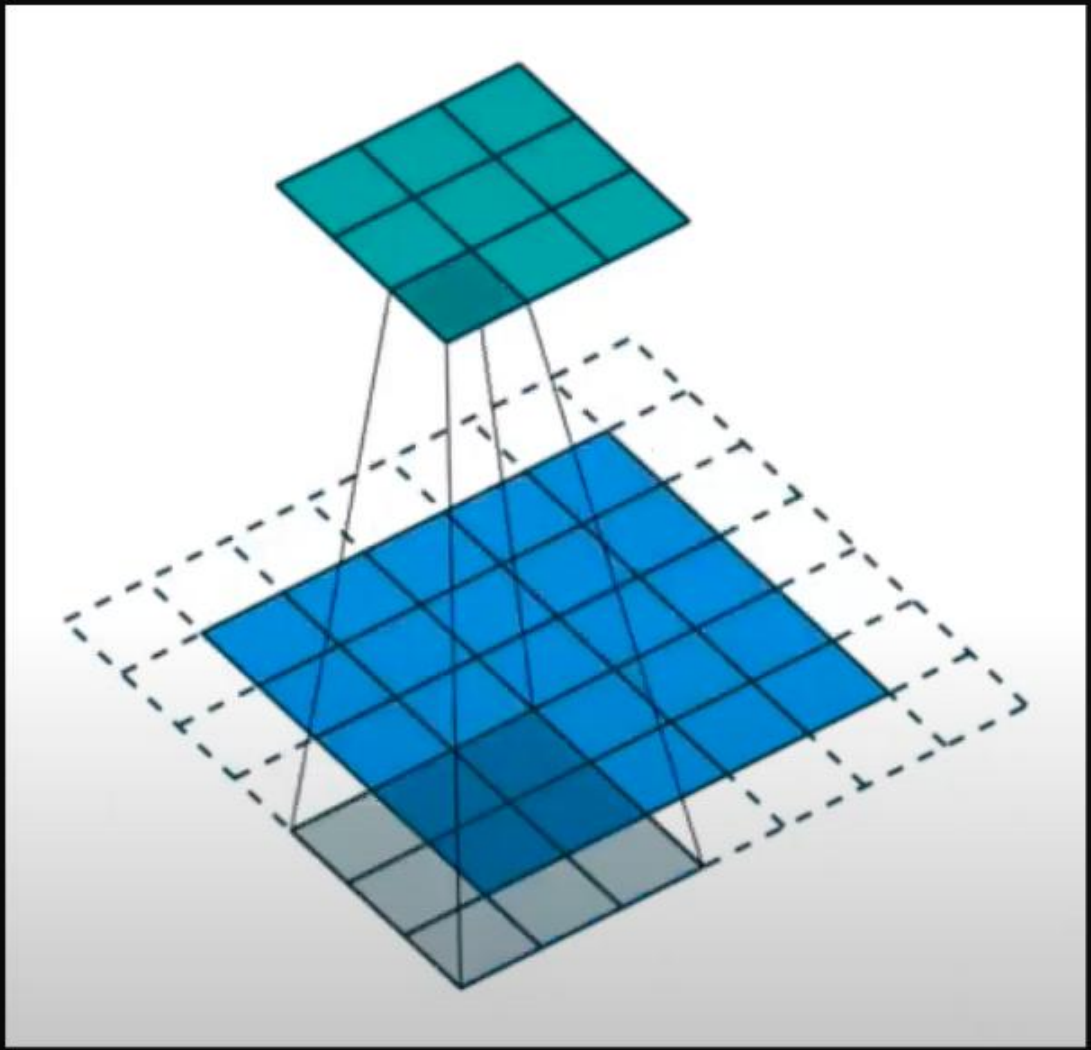
Stride = (2, 2)

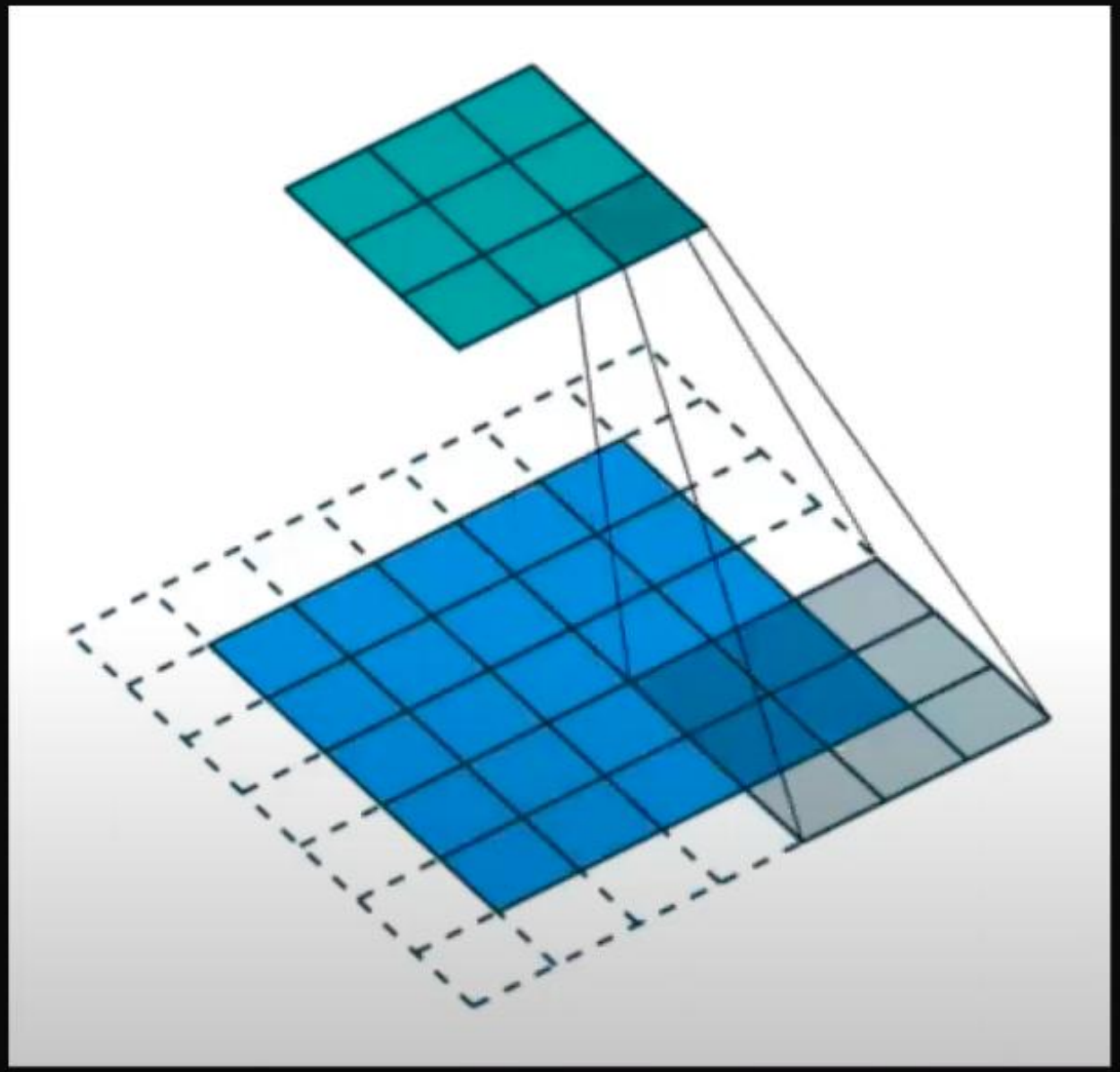












Cat

Cat

Pooling

01 September 2022 09:55

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

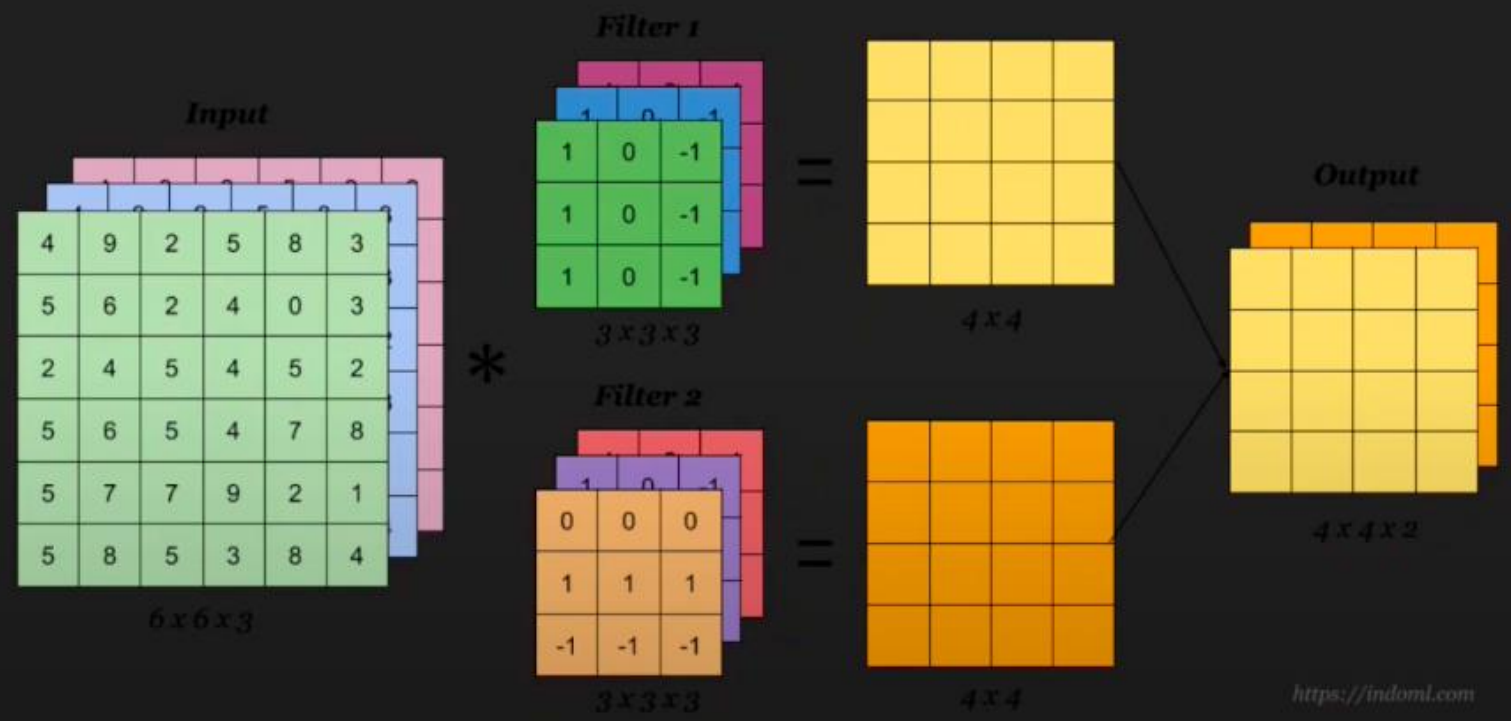
*

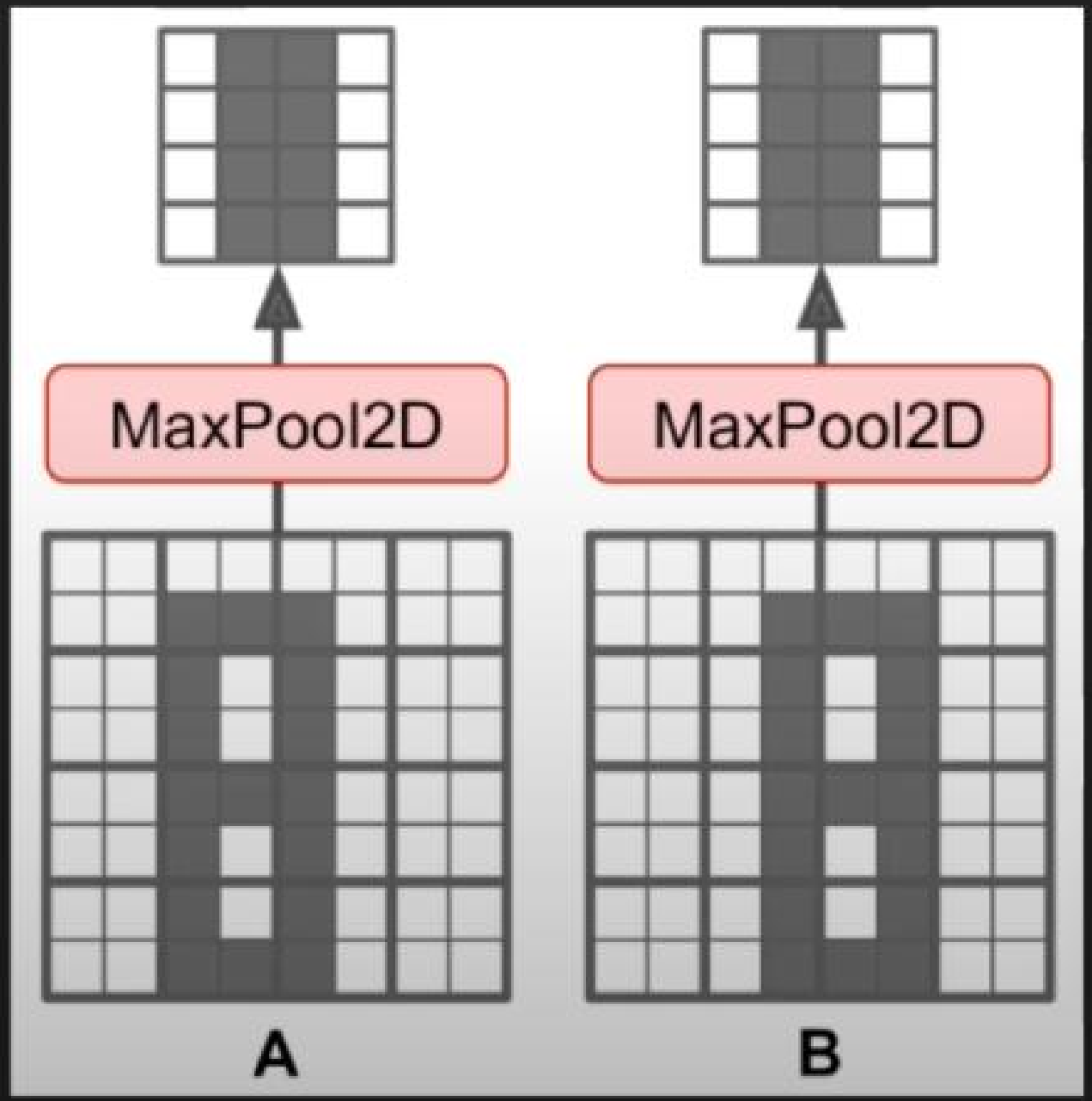
-1	-1	-1
0	0	0
1	1	1

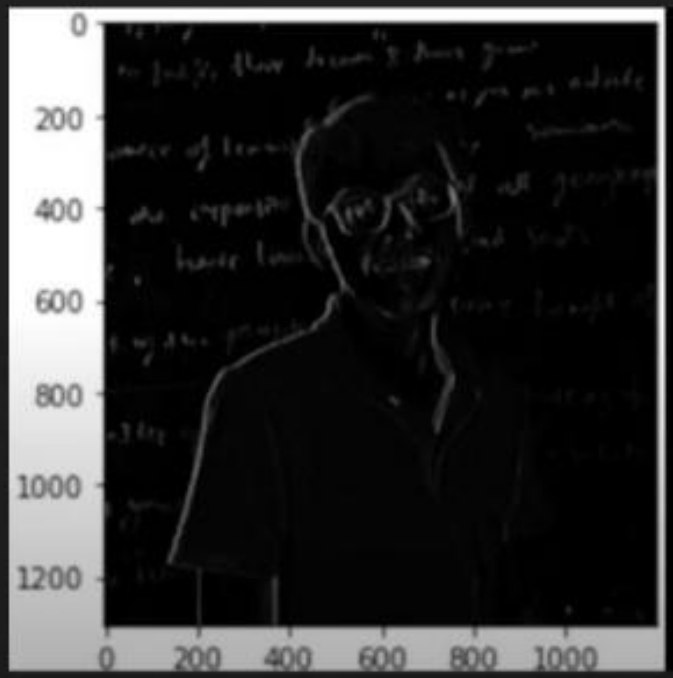
=

Pooling on Volumes

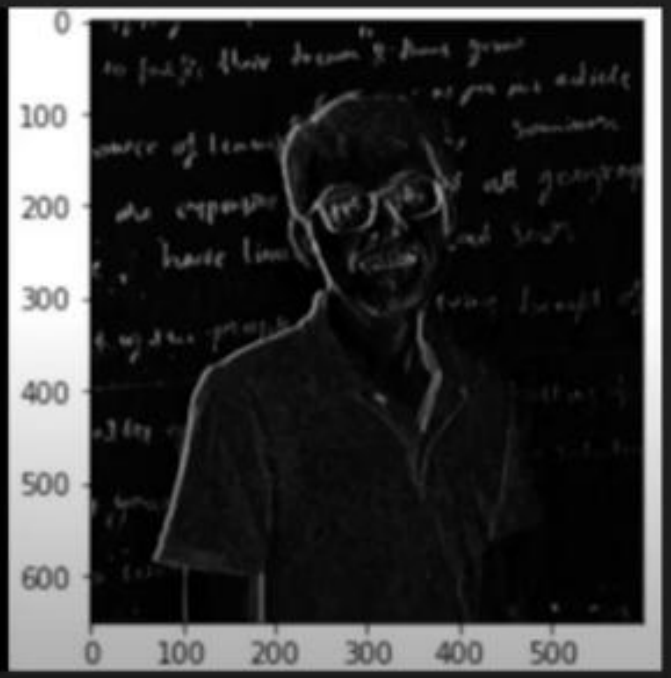
01 September 2022 09:56

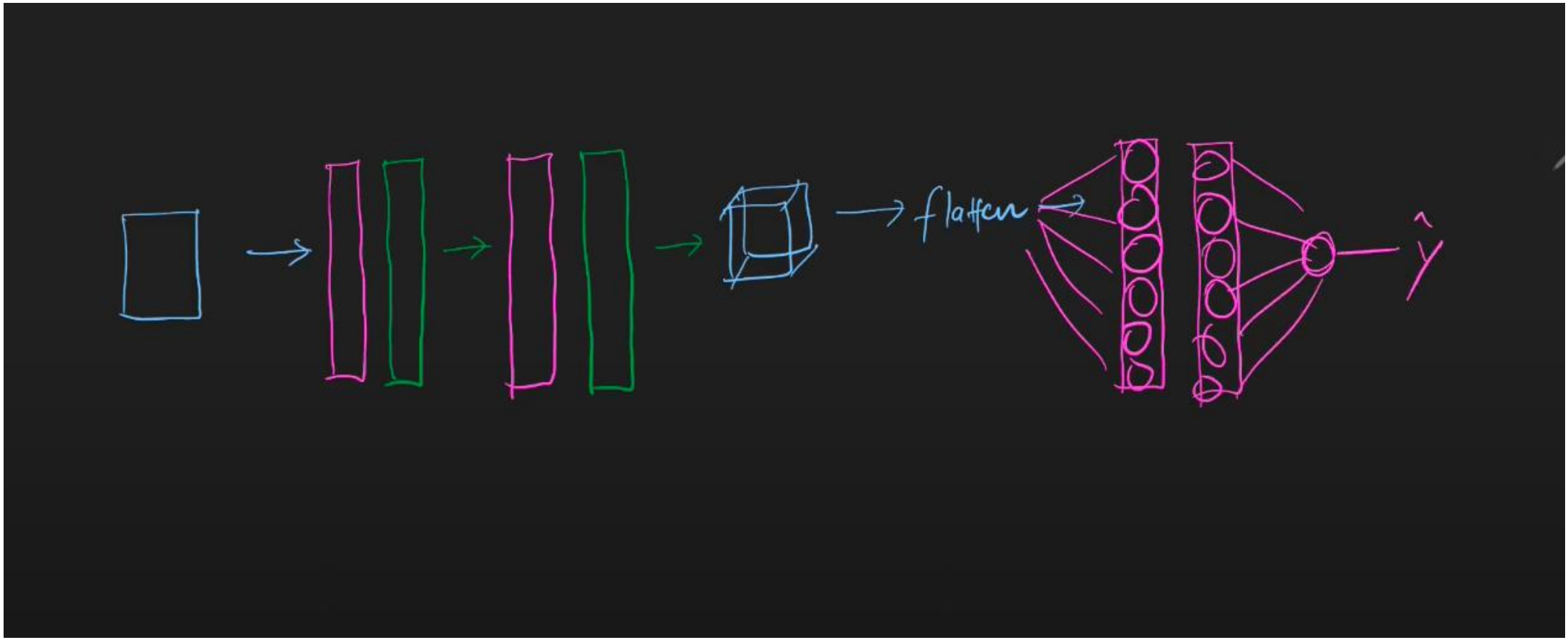


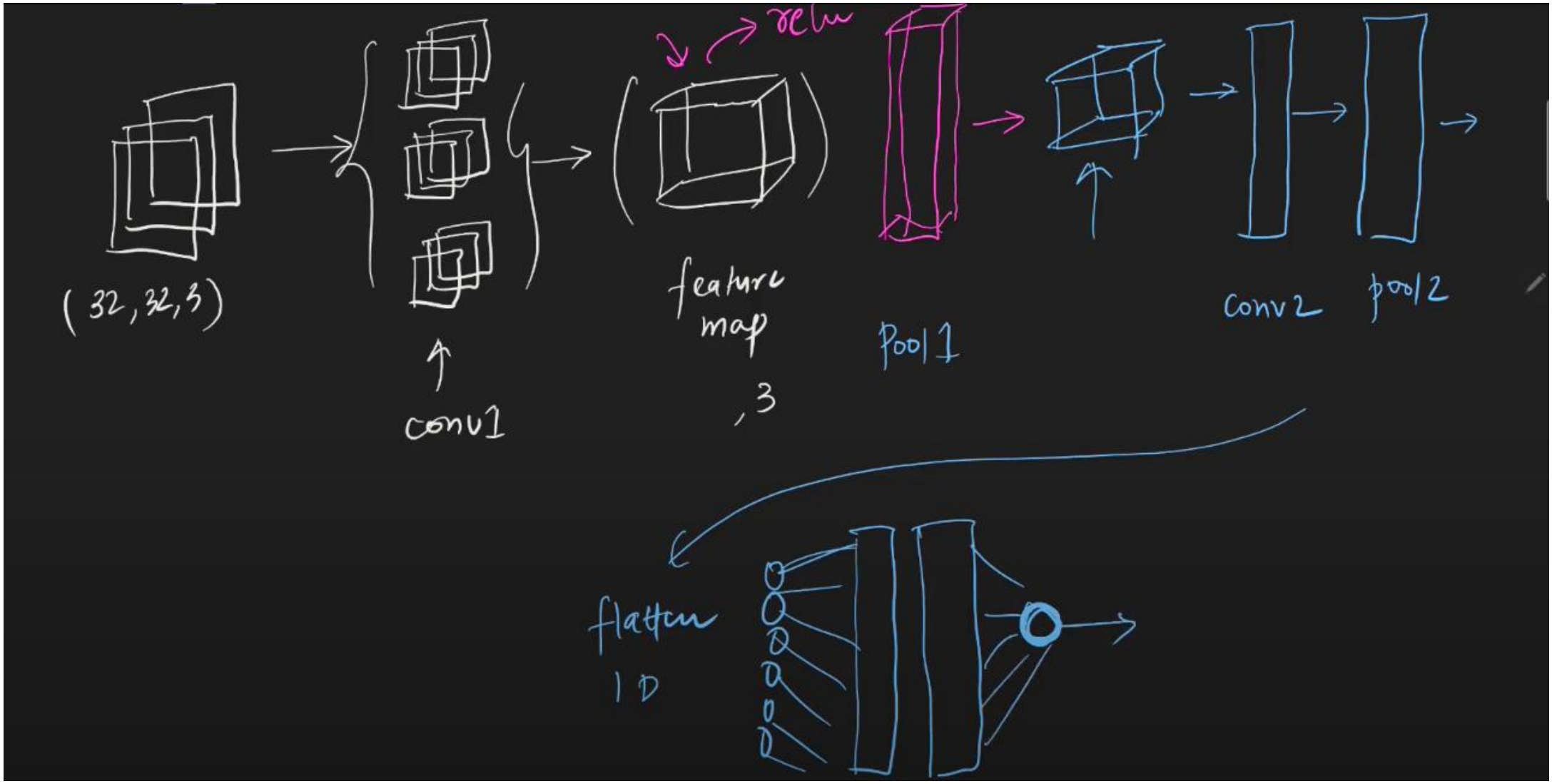

MaxPool2D
MaxPool2D
A
B

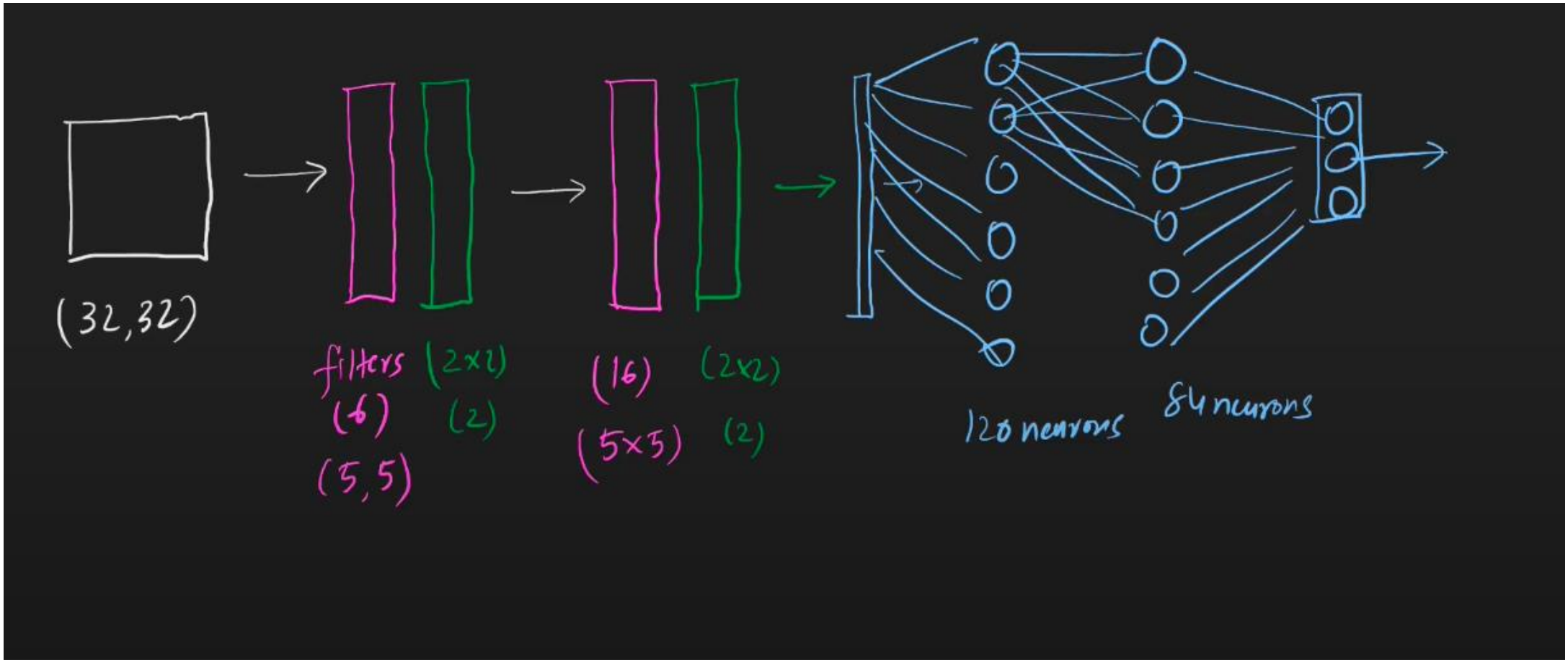

0
200
400
600
800
1000
1200
0
200
400
600
800
1000

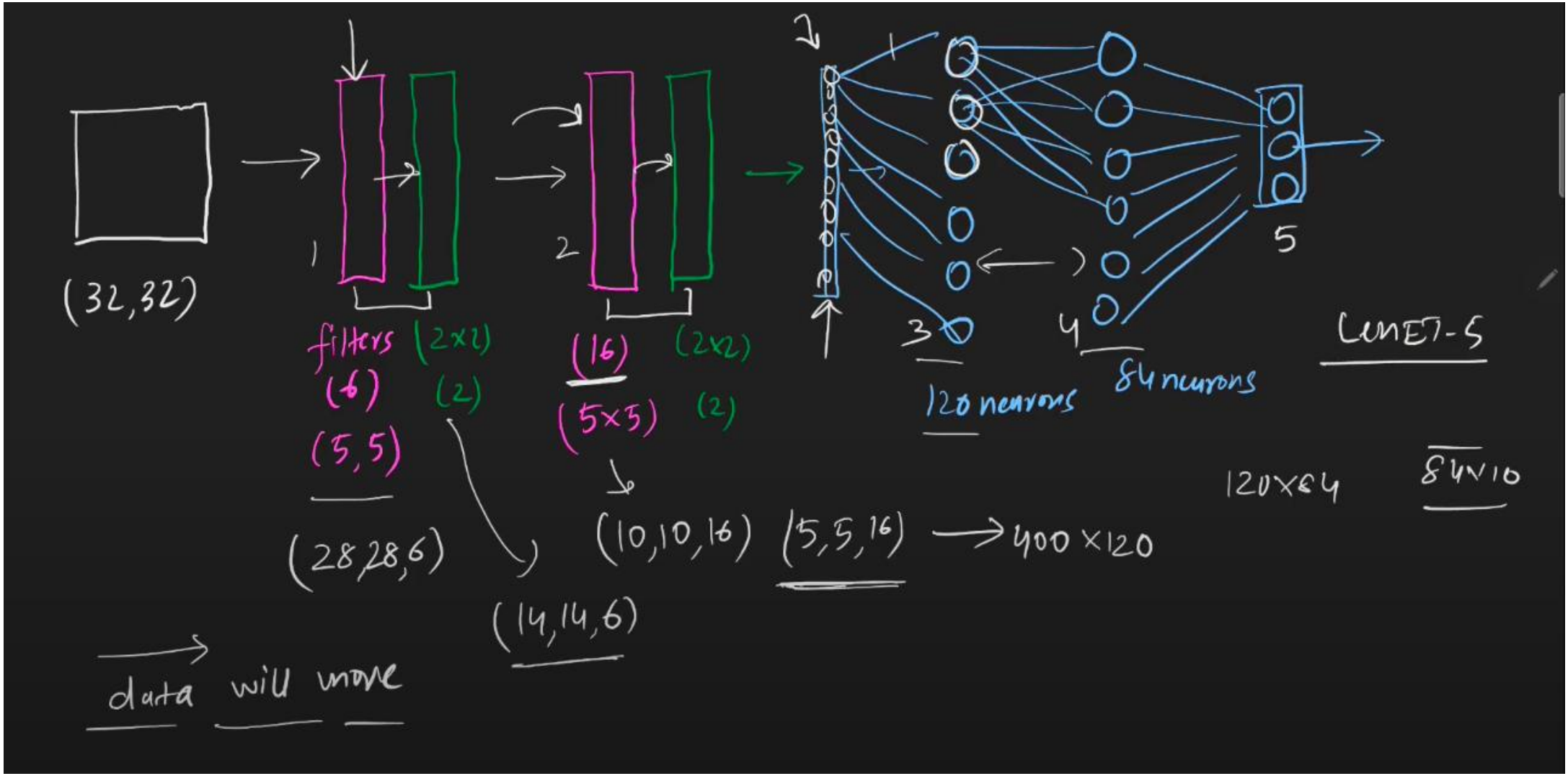
Max Pooling


0
100
200
300
400
500
600
0
100
200
300
400
500


flatten
$\hat{y}$


(32, 32, 3)
conv1
relu
feature map
, 3
Pool 1
conv2
pool2
flatten
1D


(32,32)
filters
(6)
(5,5)
(2x2)
(2)
(16)
(5x5)
(2x2)
(2)
120 neurons
84 neurons


(32,32)
filters (2x2)
(6) (2)
(5,5)
(28,28,6)
(14,14,6)
(16) (2x2)
(5x5) (2)
(10,10,16) (5,5,16) → 400 × 120
120 neurons
84 neurons
LENET-5
120×84
84×10
data will move

```
import tensorflow
from tensorflow import keras
from keras.layers import Dense,Conv2D,Flatten,AveragePooling2D
from keras import Sequential
from keras.datasets import mnist
```

```
[ ] (X_train, y_train), (X_test, y_test) = mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
```

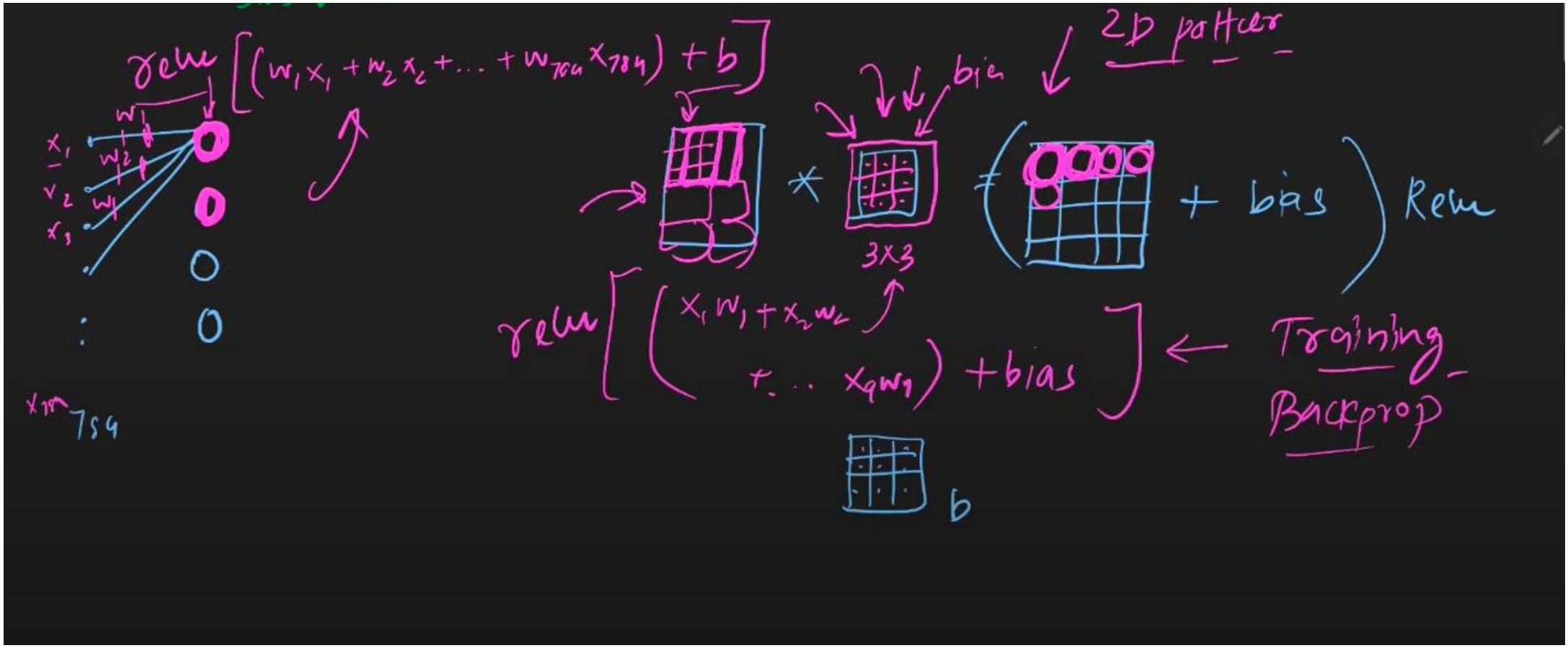
```
[ ]
```

LeNet Architecture

LeNet Architecture

```
model = Sequential()

model.add(Conv2D(6,kernel_size=(5,5),padding='valid', activation='tanh', input_shape=(32,32,1)))
model.add(AveragePooling2D(pool_size=(2, 2), strides=2, padding='valid'))

model.add(Conv2D(16,kernel_size=(5,5),padding='valid', activation='tanh'))
model.add(AveragePooling2D(pool_size=(2, 2), strides=2, padding='valid'))

model.add(Flatten())

model.add(Dense(120, activation='tanh'))
model.add(Dense(84, activation='tanh'))
model.add(Dense(10, activation='softmax'))
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 28, 28, 6)         156

 average_pooling2d_2 (Averag  (None, 14, 14, 6)        0
 ePooling2D)

 conv2d_3 (Conv2D)           (None, 10, 10, 16)        2416

 average_pooling2d_3 (Averag  (None, 5, 5, 16)         0
 ePooling2D)

 flatten_1 (Flatten)         (None, 400)               0

 dense_3 (Dense)             (None, 120)               48120

 dense_4 (Dense)             (None, 84)                10164

 dense_5 (Dense)             (None, 10)                850
```


relu [(w1x1 + w2x2 + ... + w784x784) + b]
w1
w2
w1
x1
x2
x3
x784
2D patter
bia
3x3
+ bias
Relu
relu [(x1w1 + x2w2 + ... x9w9) + bias]
Training
Backprop
b