

# Decision Science for Sustainable Livestock Systems

Yalong Pi

Associate Research Scientist

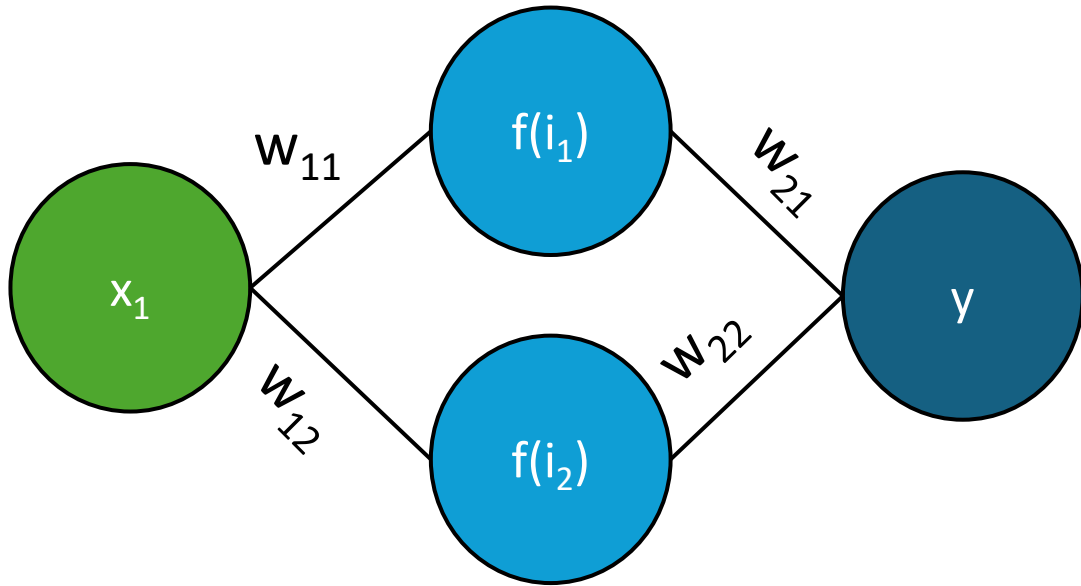
Texas A&M Institute of Data Science

[piyalong@tamu.edu](mailto:piyalong@tamu.edu)

20250327

# Agenda

- Fundamentals of machine learning
- Feed grading with computer vision
- Transformers for Language Models (GPT)



$$y_{pred} = w_{21} * x_1 * w_{11} + w_{22} * x_1 * w_{12}$$

$$Loss = (y_{pred} - y_{true})^2$$

$$W_n' = W_n - LR (\partial Loss / \partial W_n)$$

x	y
1	10
2	20
5	30

$$y_{pred} = w_{21} * x_1 * w_{11} + w_{22} * x_1 * w_{12}$$

$$Loss = (y_{pred} - y_{true})^2$$

$$W_n' = W_n - LR (\partial Loss / \partial W_n)$$

x	y
1	10
2	20
5	30

$$y_{pred} = w_{21} * x_1 * w_{11} + w_{22} * x_1 * w_{12}$$

$$Loss = (y_{pred} - y_{true})^2$$

$$W_n' = W_n - LR (\partial Loss / \partial W_n)$$

x	y
1	10
2	20
5	30

For each connection:

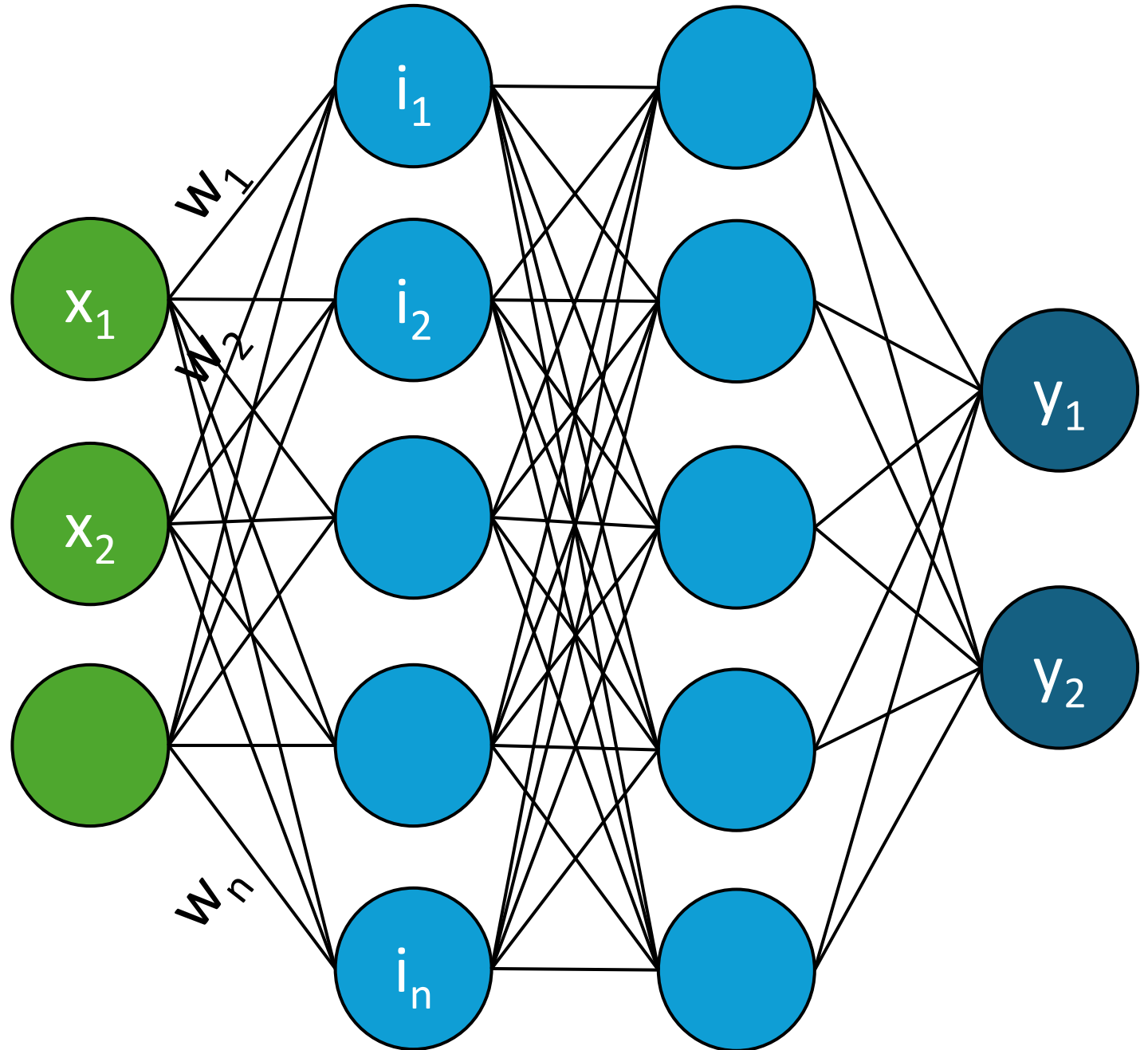
$$I_n = f\left(\sum_n x_n w_n + b\right)$$

□  $f$  is the activation function

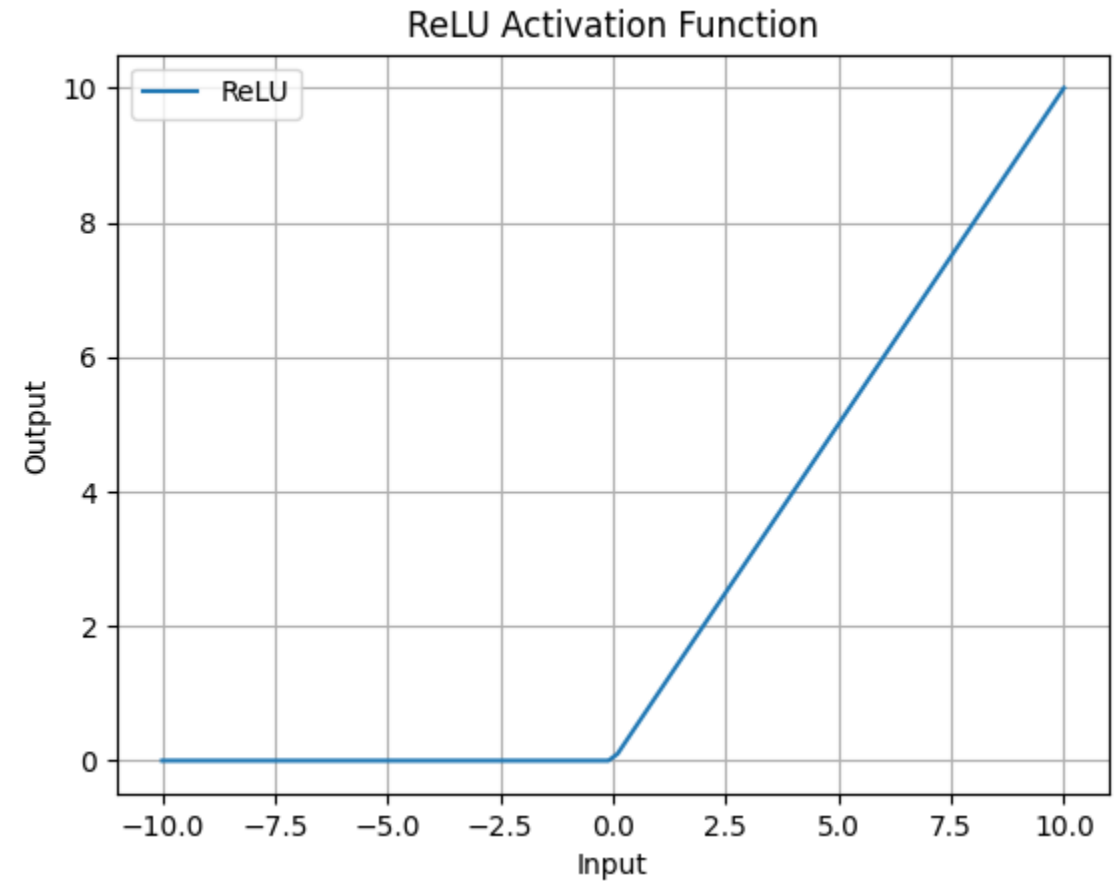
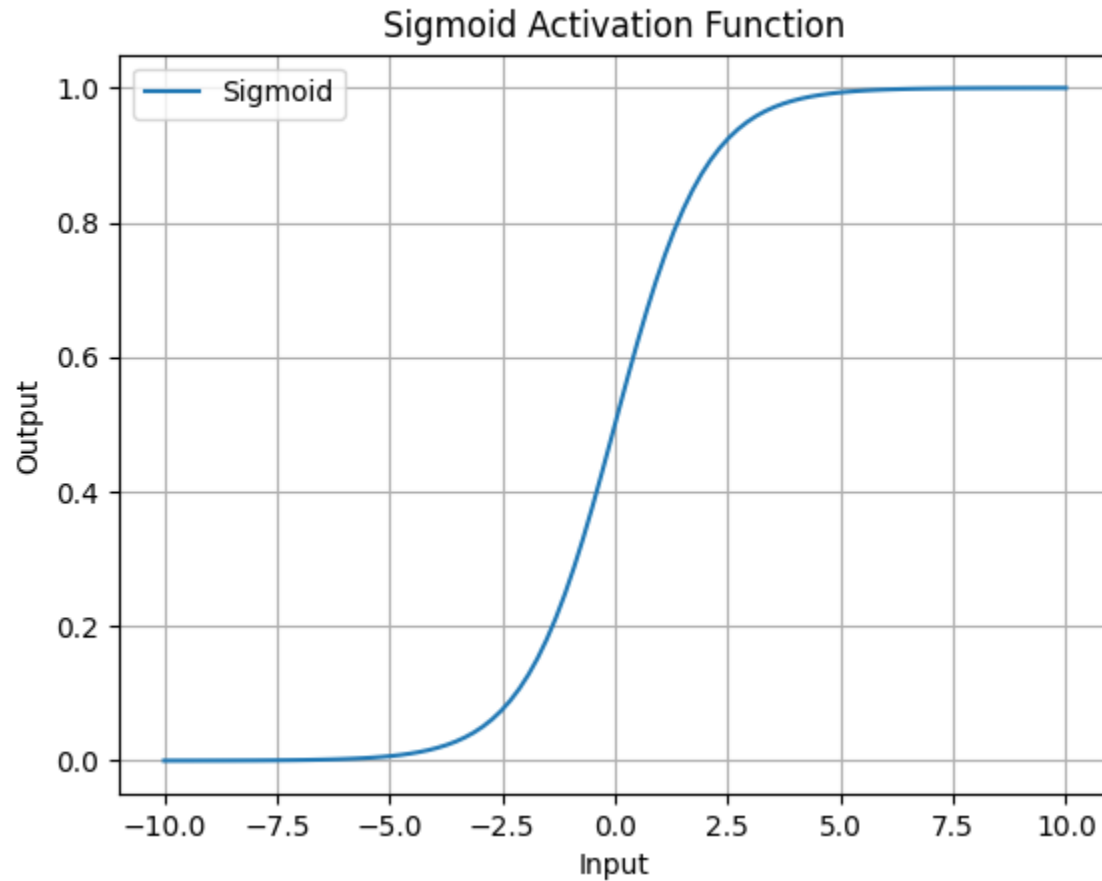
□  $w_n$  is the weight

□  $b$  is the bias.

□ A DNN has millions of weights and biases



# Activation Functions



# Softmax

$$\sigma(Z)_i = \exp(z_i) / \Sigma(\exp(z_j))$$

Where:

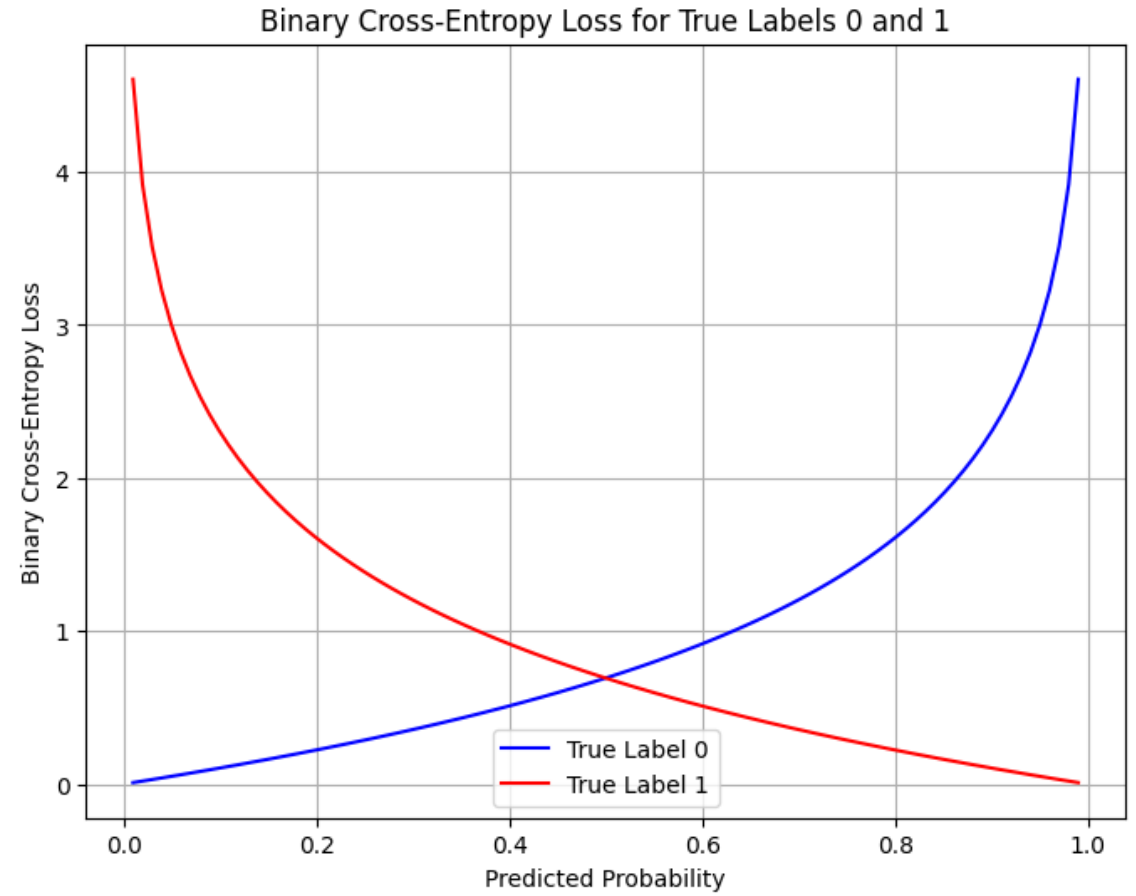
- $z_i$  is the  $i$  input score (logits)
- $\Sigma$  is the sum over all input scores (logits)
- $\sigma(z)_i$  is the probability assigned to class  $i$

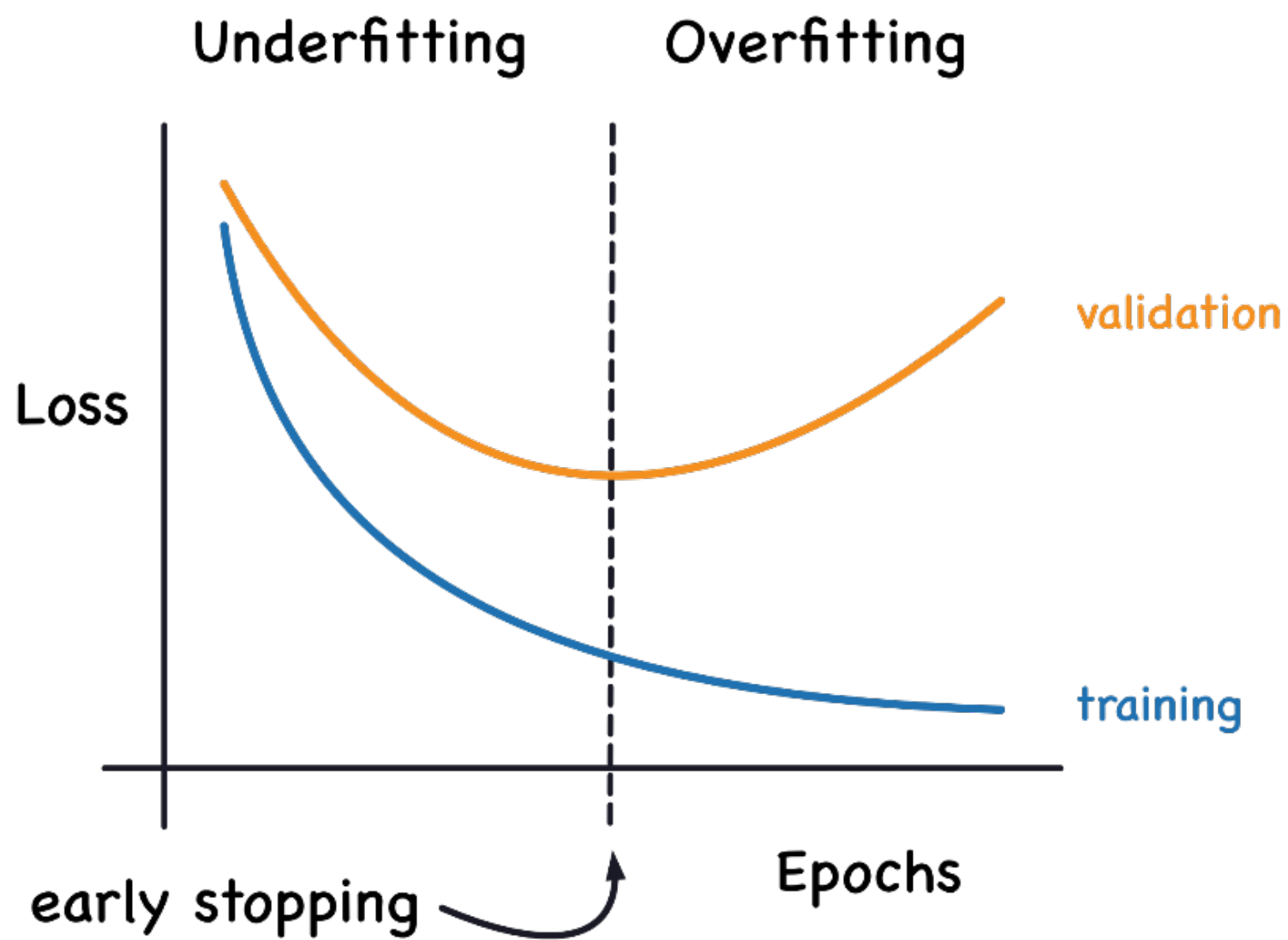
Example:

$[-0.37, -1.06, -0.07, -1.47, -0.90] \rightarrow [0.265, 0.133, 0.358, 0.088, 0.155]$



# Binary cross entropy





Prediction\Ground Truth	Positive	Negative
Positive	TP	FP
Negative	FN	TN

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

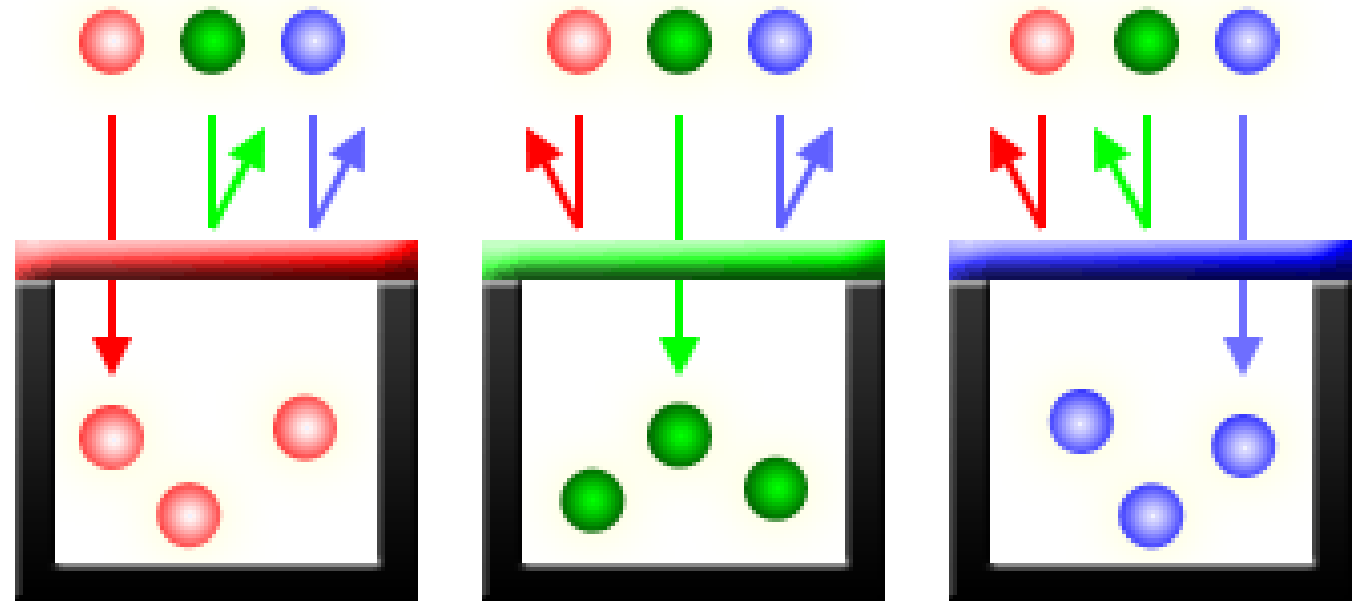
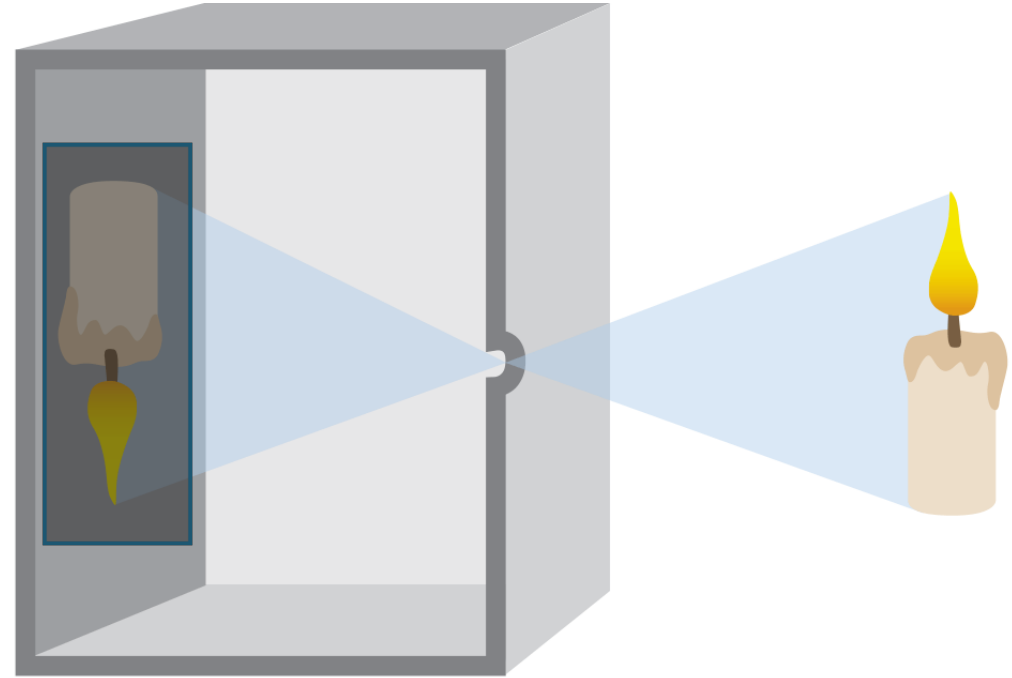
$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

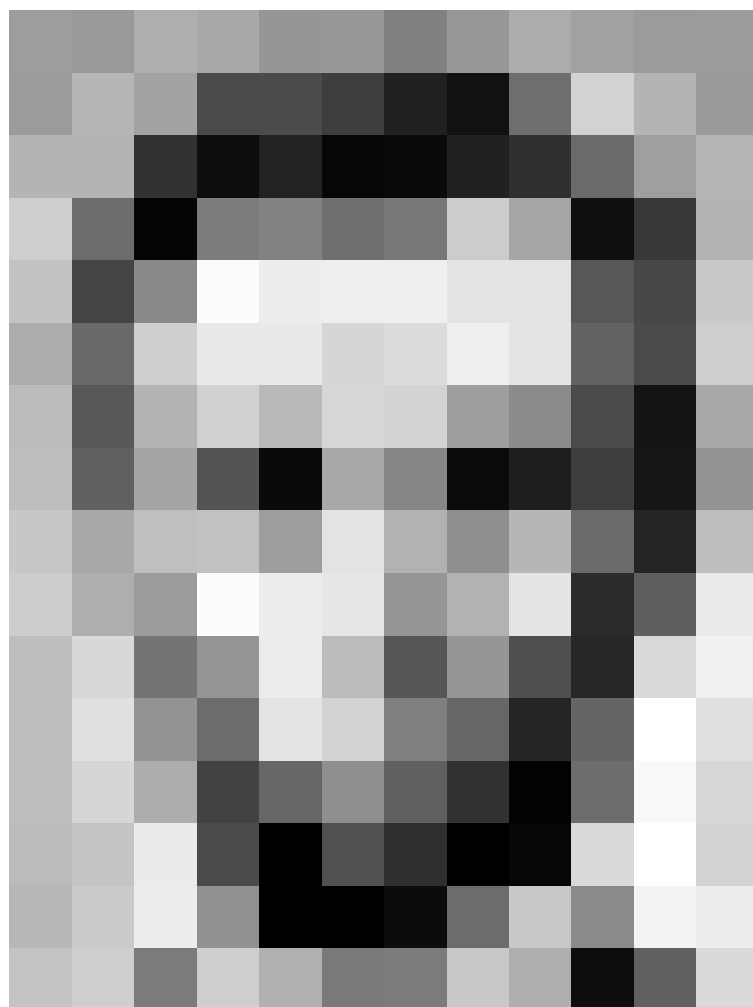
# Computer vision



- ❑ Pinhole principle
- ❑ Traditional film
- ❑ Digital sensors (CCD and CMOS)
- ❑ Red Green Blue (RGB) channels



# Grey Scale



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	106	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	105	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	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	106	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	105	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



# KERNELS AND CONVOLUTION



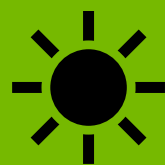
Original Image



Blur



Sharpen



Brighten



Darken





# KERNELS AND CONVOLUTION

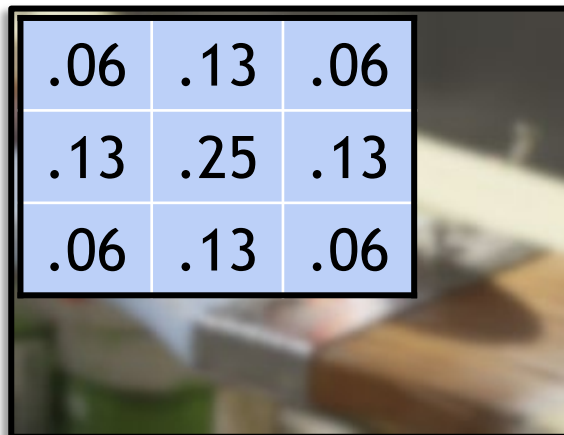


Original Image



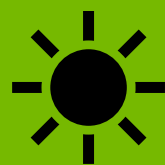
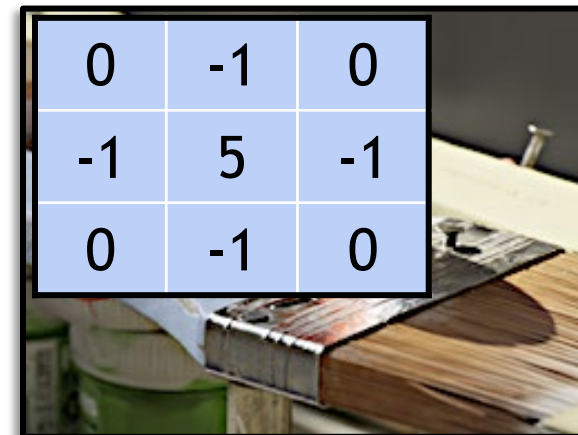
Blur

.06	.13	.06
.13	.25	.13
.06	.13	.06



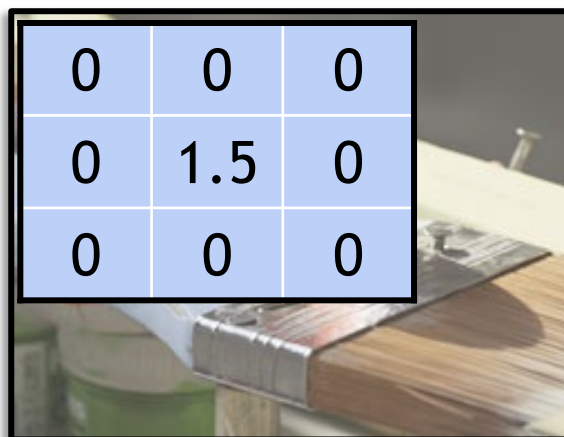
Sharpen

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



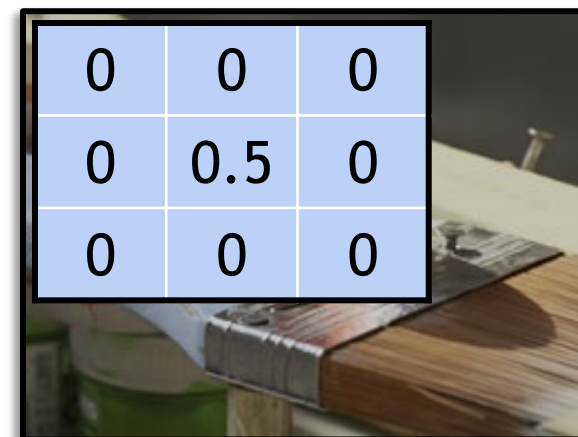
Brighten

0	0	0
0	1.5	0
0	0	0



Darken

0	0	0
0	0.5	0
0	0	0



# KERNELS AND CONVOLUTION

Blur Kernel

.06	.13	.06
.13	.25	.13
.06	.13	.06

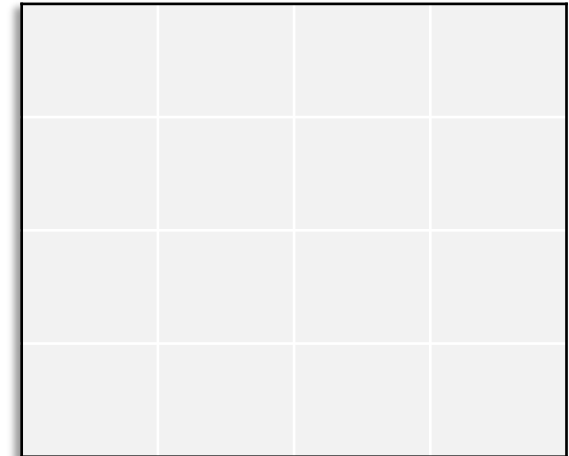
\*

Original Image

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

=

Convolved Image



# KERNELS AND CONVOLUTION

Blur Kernel

.06	.13	.06
.13	.25	.13
.06	.13	.06

\*

Original Image

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

=

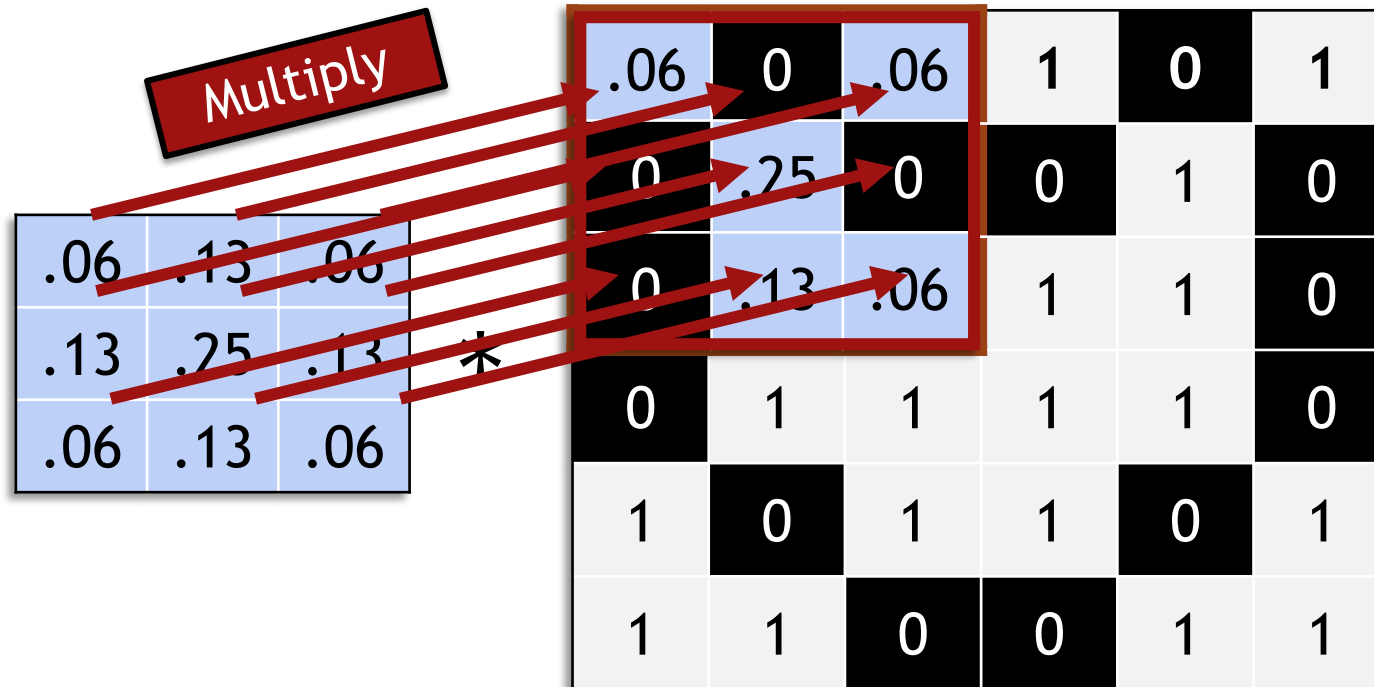
Convolved Image


# KERNELS AND CONVOLUTION

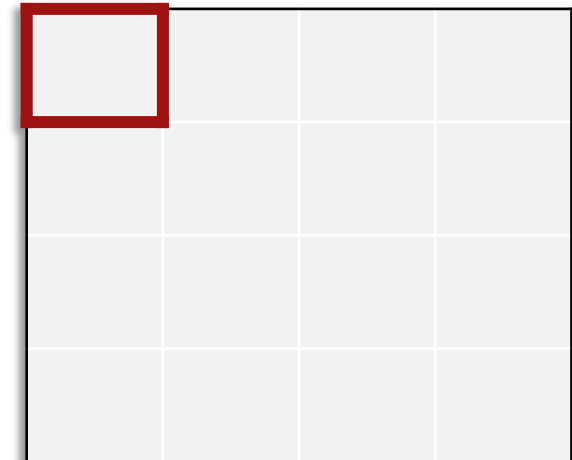
Blur Kernel

Original Image

Convolved Image



=



# KERNELS AND CONVOLUTION

Blur Kernel

.06	.13	.06
.13	.25	.13
.06	.13	.06

\*

Original Image

.06	0	.06	1	0	1
0	.25	0	0	1	0
0	.13	.06	1	1	0
0	1	1	1	1	0
1	0	1	1	0	1
1	1	0	0	1	1

Total

=

Convolved Image

.56			

# KERNELS AND CONVOLUTION

Blur Kernel

.06	.13	.06
.13	.25	.13
.06	.13	.06

\*

Original Image

1	0	.13	.06	0	1
0	.13	0	0	1	0
0	.06	.13	.06	1	0
0	1	1	1	1	0
1	0	1	1	0	1
1	1	0	0	1	1

=

Convolved Image

.56	.57		

# KERNELS AND CONVOLUTION

Blur Kernel

.06	.13	.06
.13	.25	.13
.06	.13	.06

\*

Original Image

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

=

Convolved Image

.56	.57	.57	.56
.7	.82	.82	.7
.69	.95	.95	.69
.64	.69	.69	.64

# Convolution Computation

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



Kernel at position 1



Kernel at position 2



Kernel at position n

7	7	7	7	5
7	7	7	5	5
7	7	5	5	5
7	5	5	5	5
5	5	5	5	5

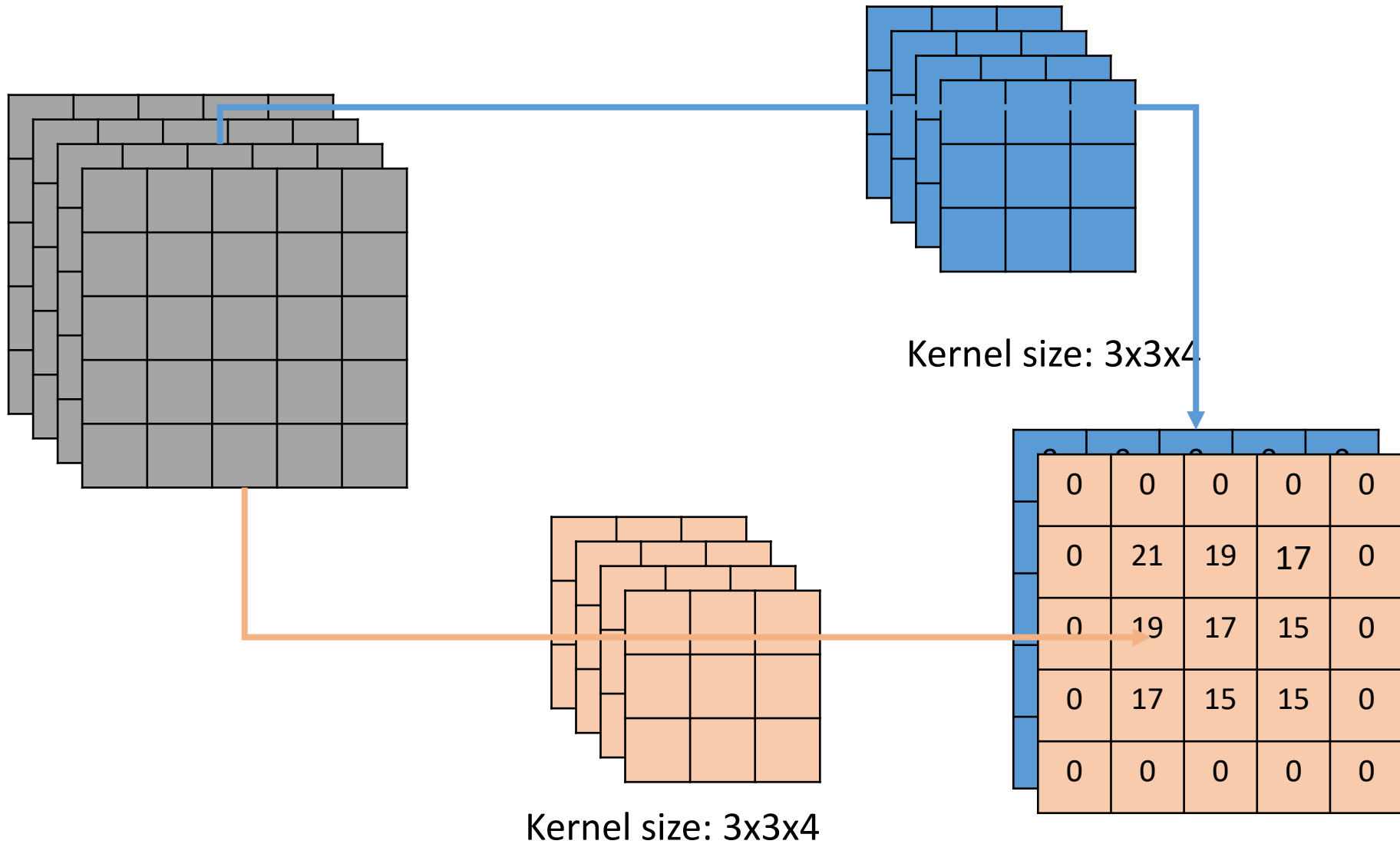
7	7	7	7	5
7	7	7	5	5
7	7	5	5	5
7	5	5	5	5
5	5	5	5	5

7	7	7	7	5
7	7	7	5	5
7	7	5	5	5
7	5	5	5	5
5	5	5	5	5

0	0	0	0	0
0	21	19	17	0
0	19	17	15	0
0	17	15	15	0
0	0	0	0	0



# Convolution Computation for more than one dimension

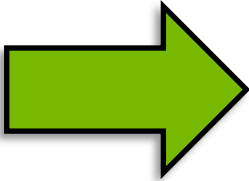




# STRIDE

Stride 1

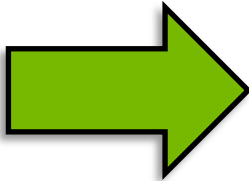
1	0	1	1	0	1
0	1	0	0	1	0
0	1	1	1	1	0



.56	.57	.57	.56
-----	-----	-----	-----

Stride 2

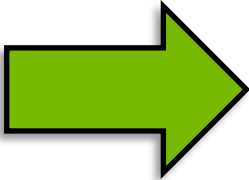
1	0	1	1	0	1
0	1	0	0	1	0
0	1	1	1	1	0



.56	.57
-----	-----

Stride 3

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



.56	.56
-----	-----



# PADDING

Original Image

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

Zero Padding

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



# PADDING

Original Image

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

Mirror Padding

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

# KERNELS AND NEURAL NETWORKS

Kernel

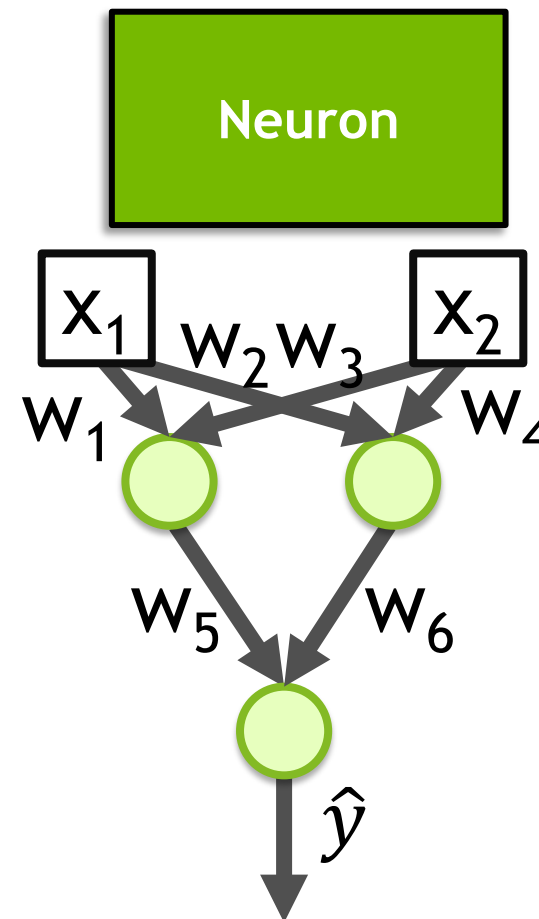
$W_1$	$W_2$	$W_3$
$W_4$	$W_5$	$W_6$
$W_7$	$W_8$	$W_9$

# KERNELS AND NEURAL NETWORKS

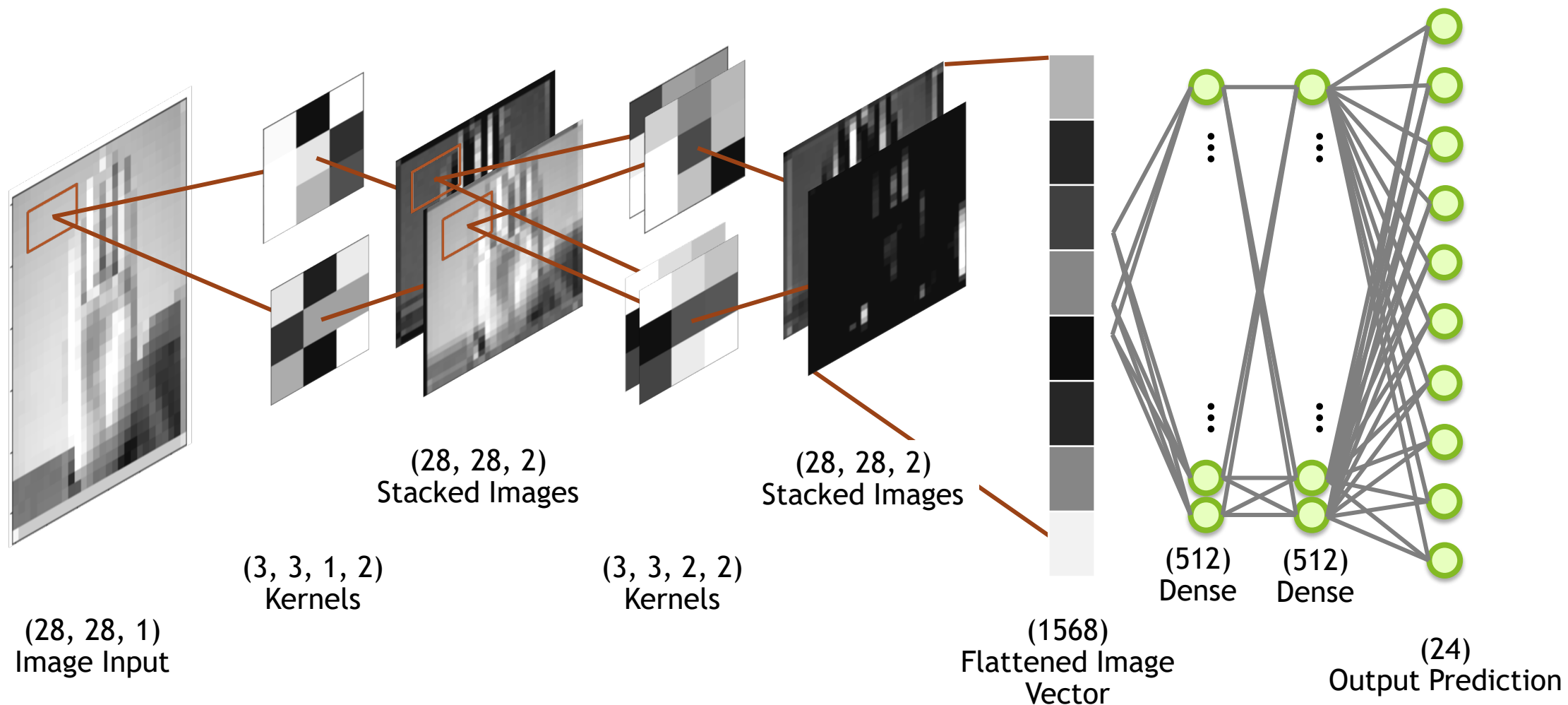
Kernel

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

Neuron



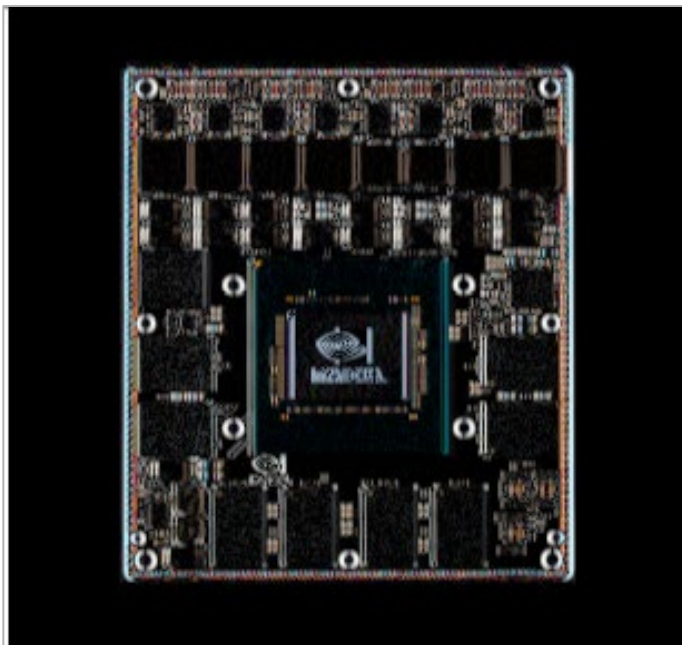
# KERNELS AND NEURAL NETWORKS





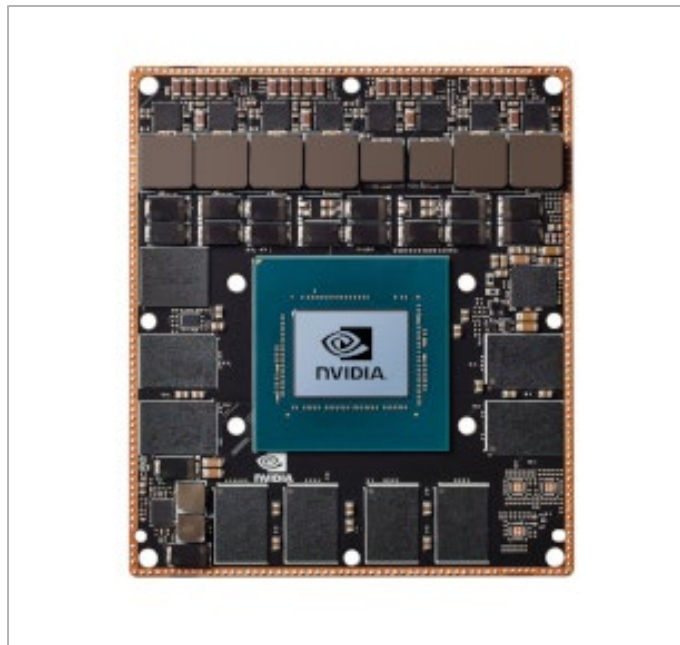
# FINDING EDGES

Vertical Edges



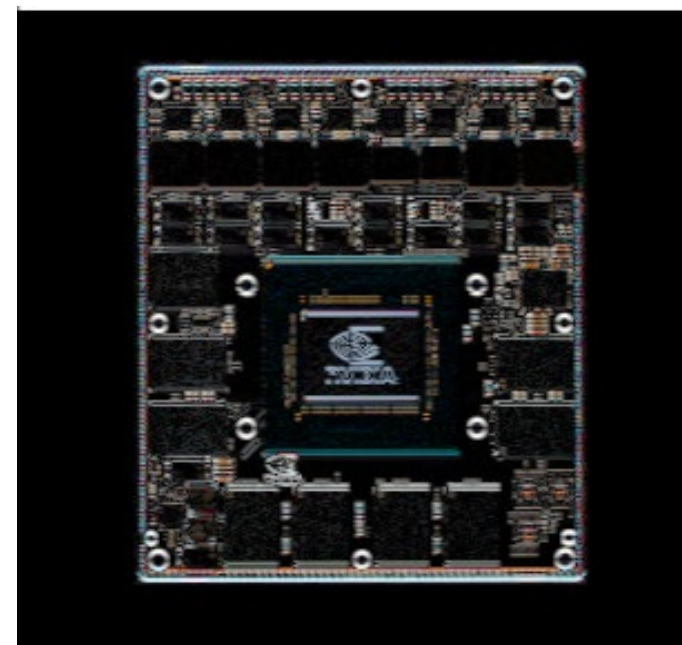
1	0	-1
2	0	-2
1	0	-1

Original Image



0	0	0
0	1	0
0	0	0

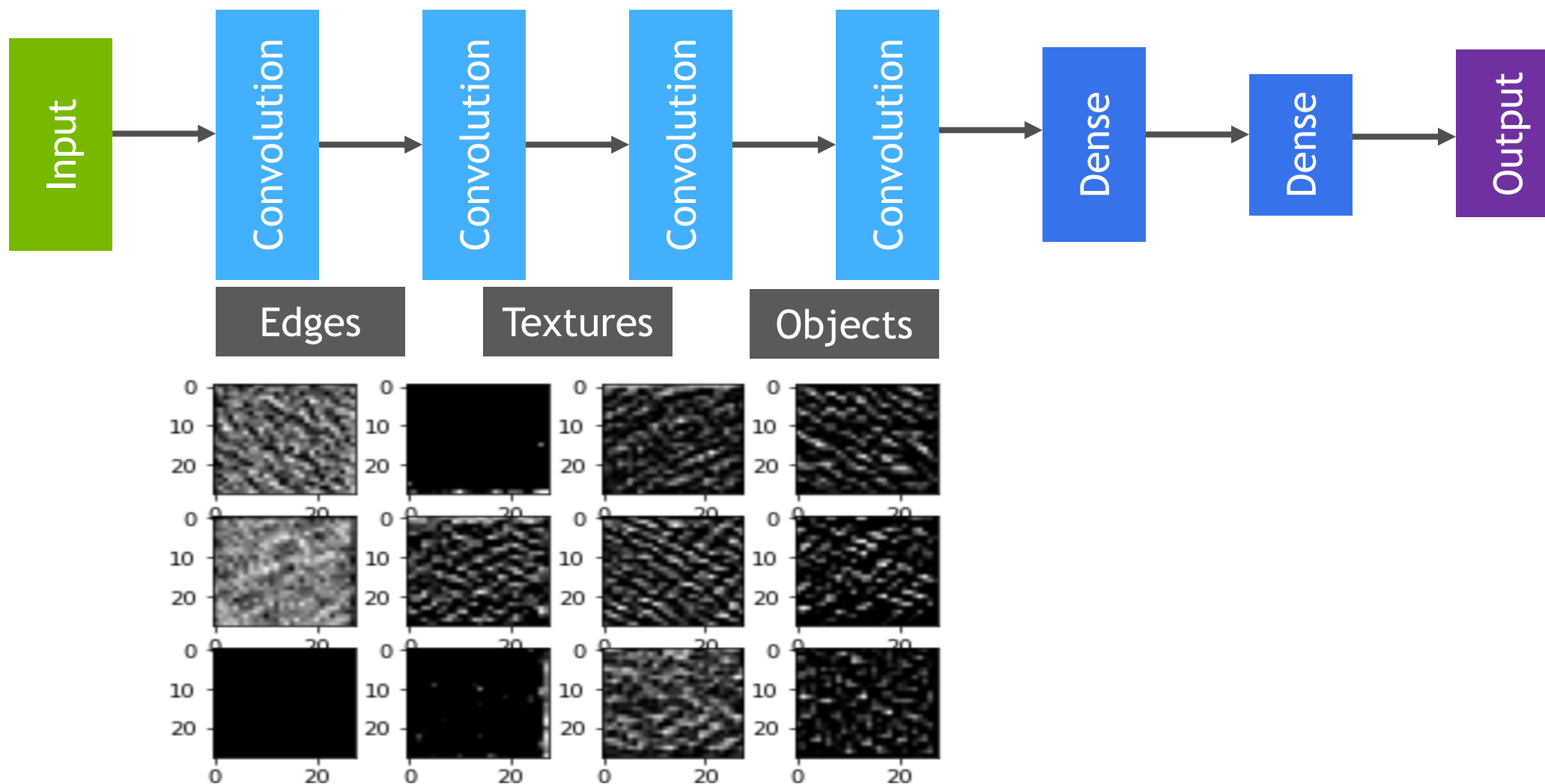
Horizontal Edges



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



# NEURAL NETWORK PERCEPTION





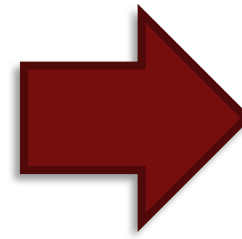
# NEURAL NETWORK PERCEPTION





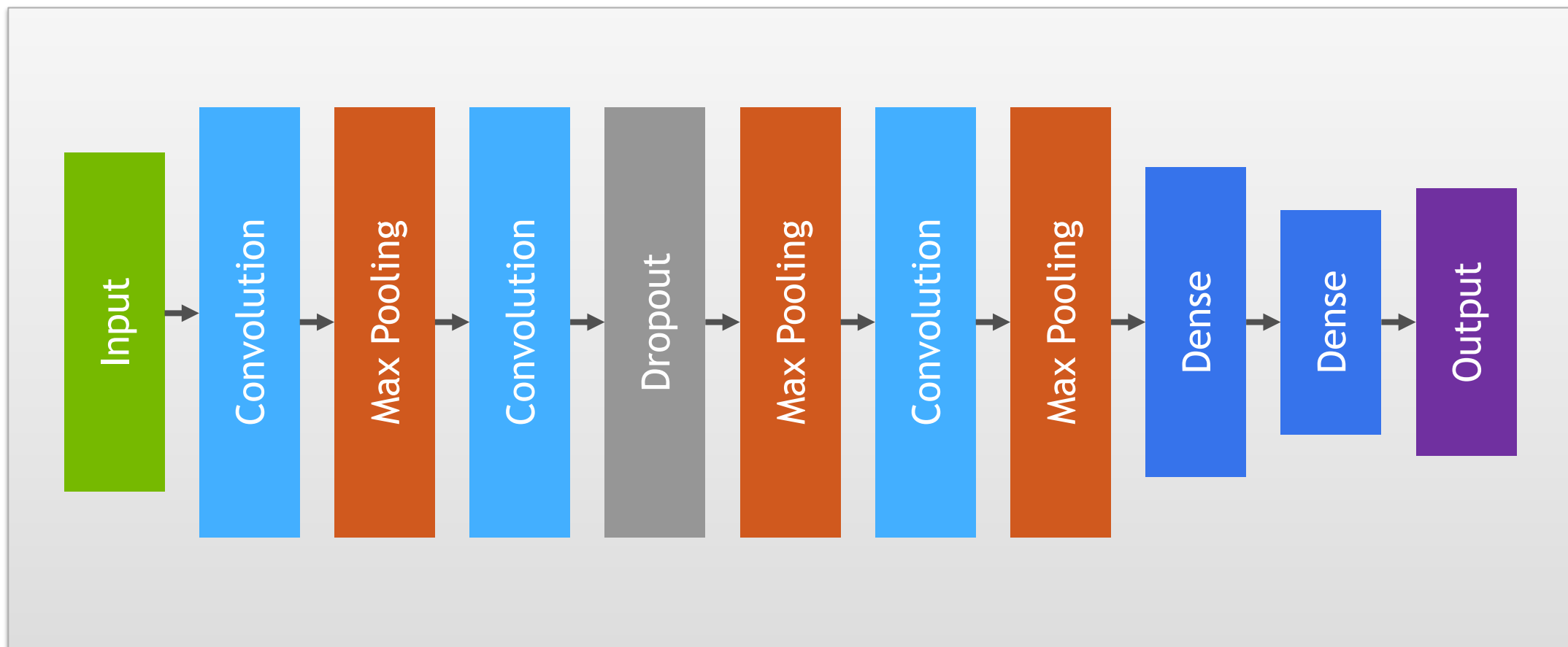
# MAX POOLING

110	256	153	67
12	89	88	43
10	15	50	55
23	9	49	23



256	153
23	55

# WHOLE ARCHITECTURE





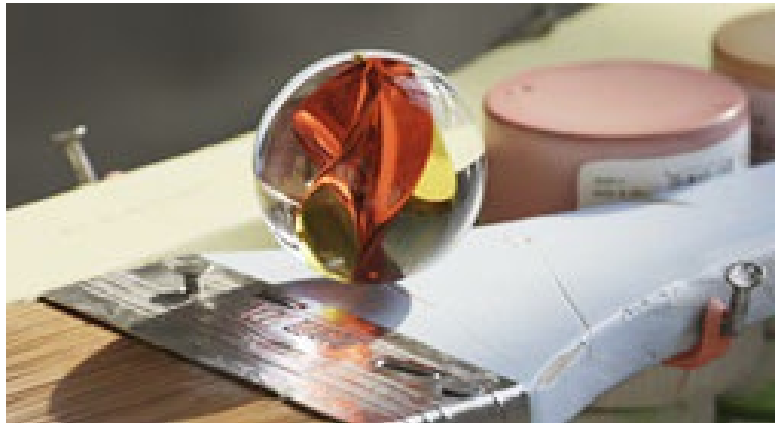
# DATA AUGMENTATION



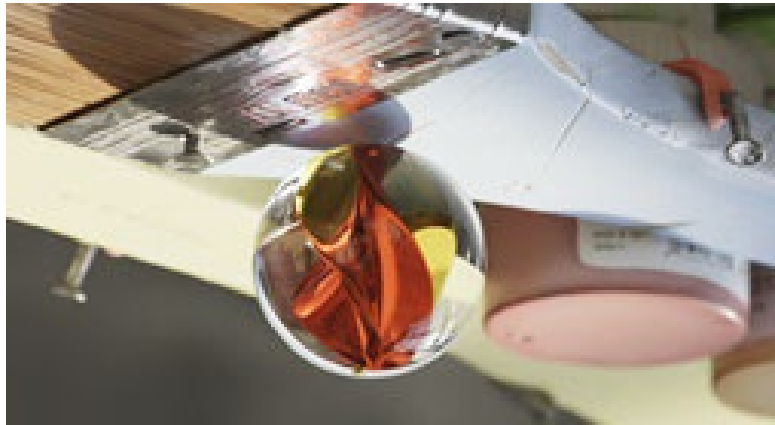


# IMAGE FLIPPING

Horizontal Flip

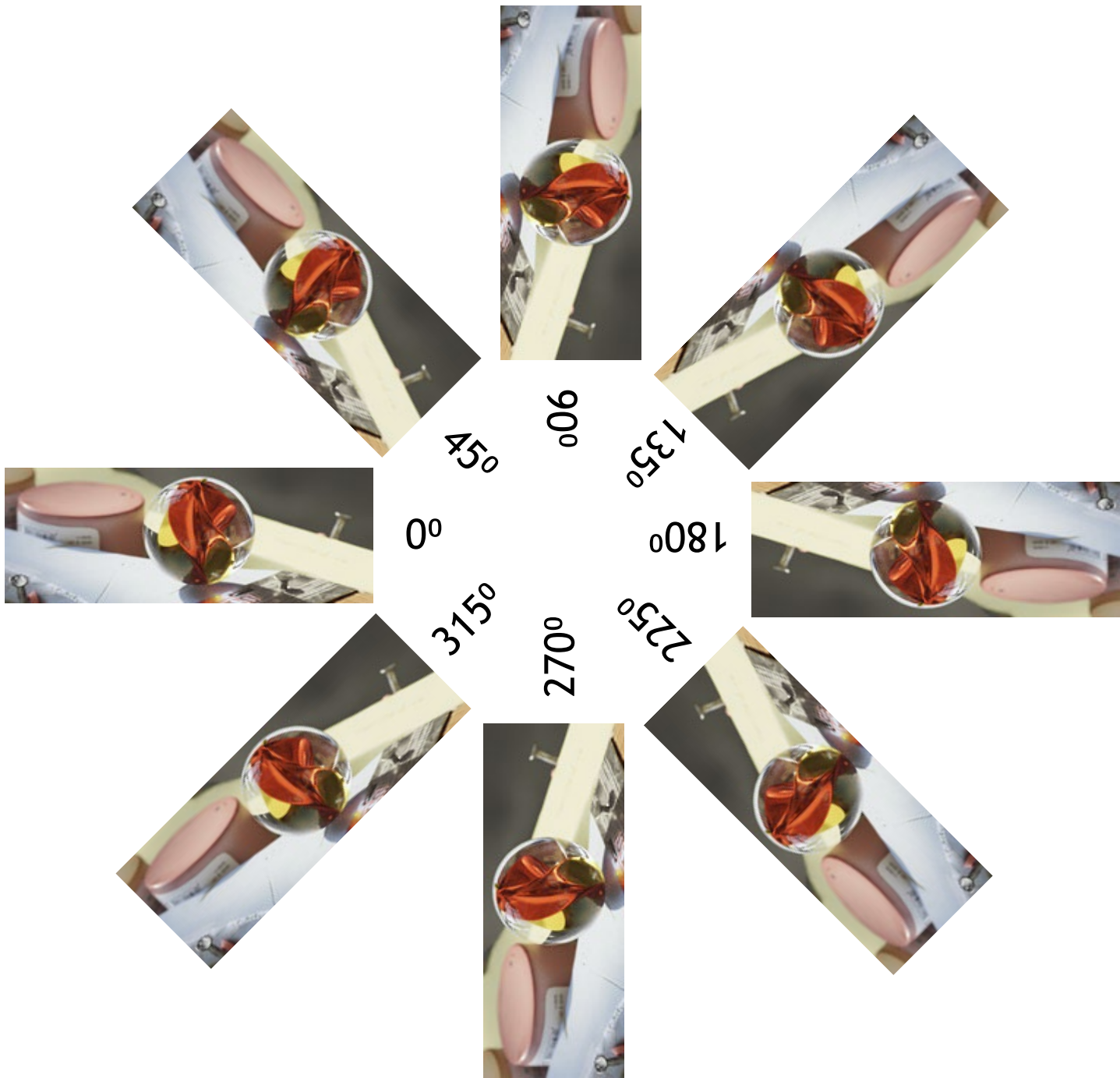


Vertical Flip



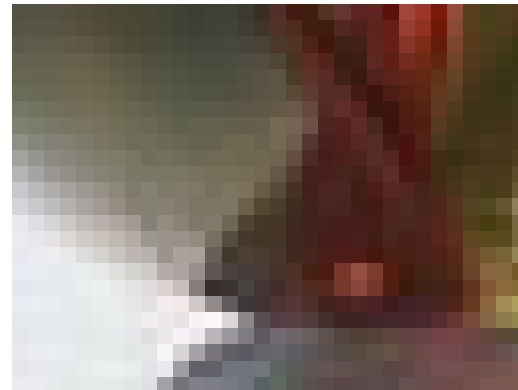


# ROTATION





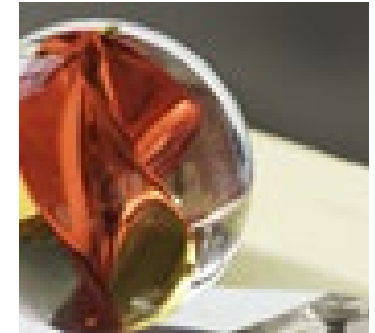
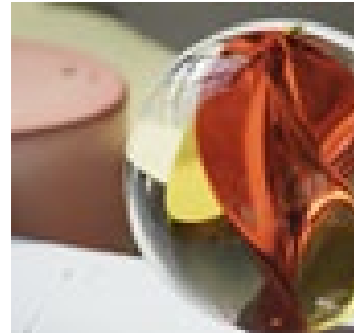
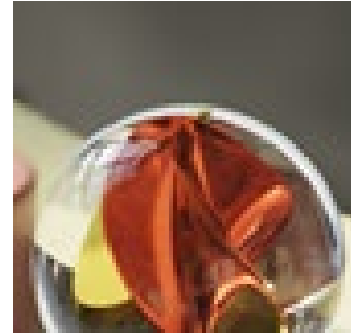
# ZOOMING





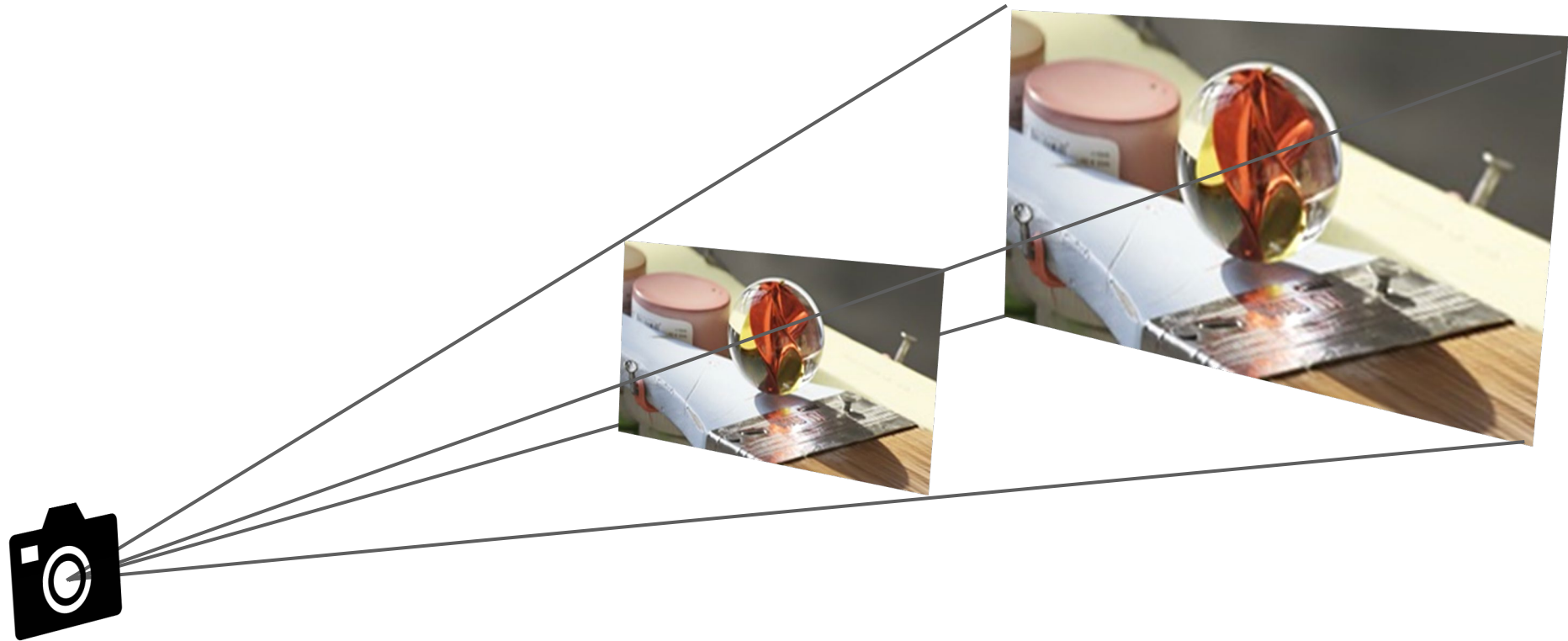


# WIDTH AND HEIGHT SHIFTING





# HOMOGRAPHY





# BRIGHTNESS

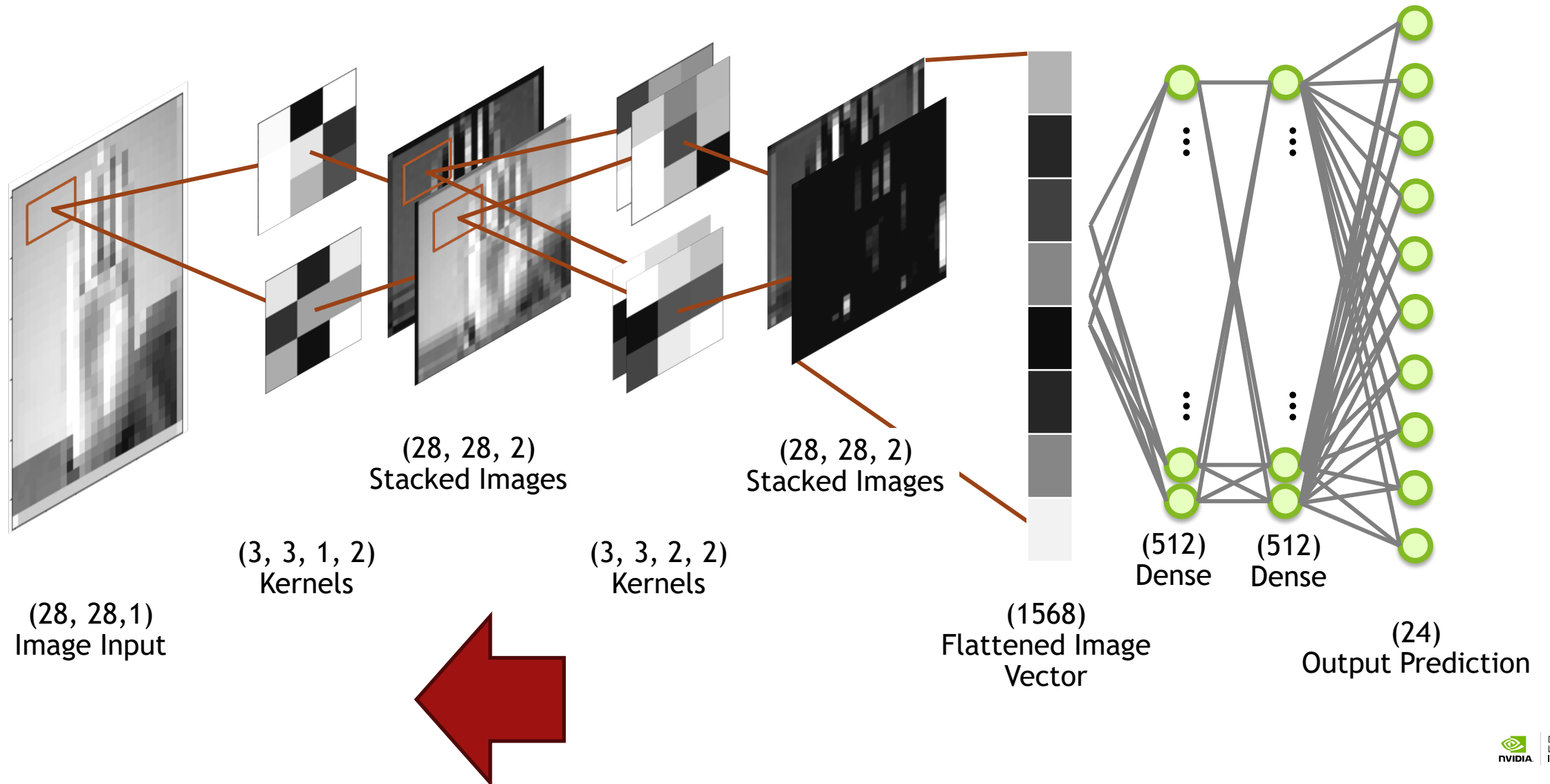




# CHANNEL SHIFTING



# MODEL DEPLOYMENT





# PRE-TRAINED MODELS

## VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

**Karen Simonyan\* & Andrew Zisserman<sup>+</sup>**

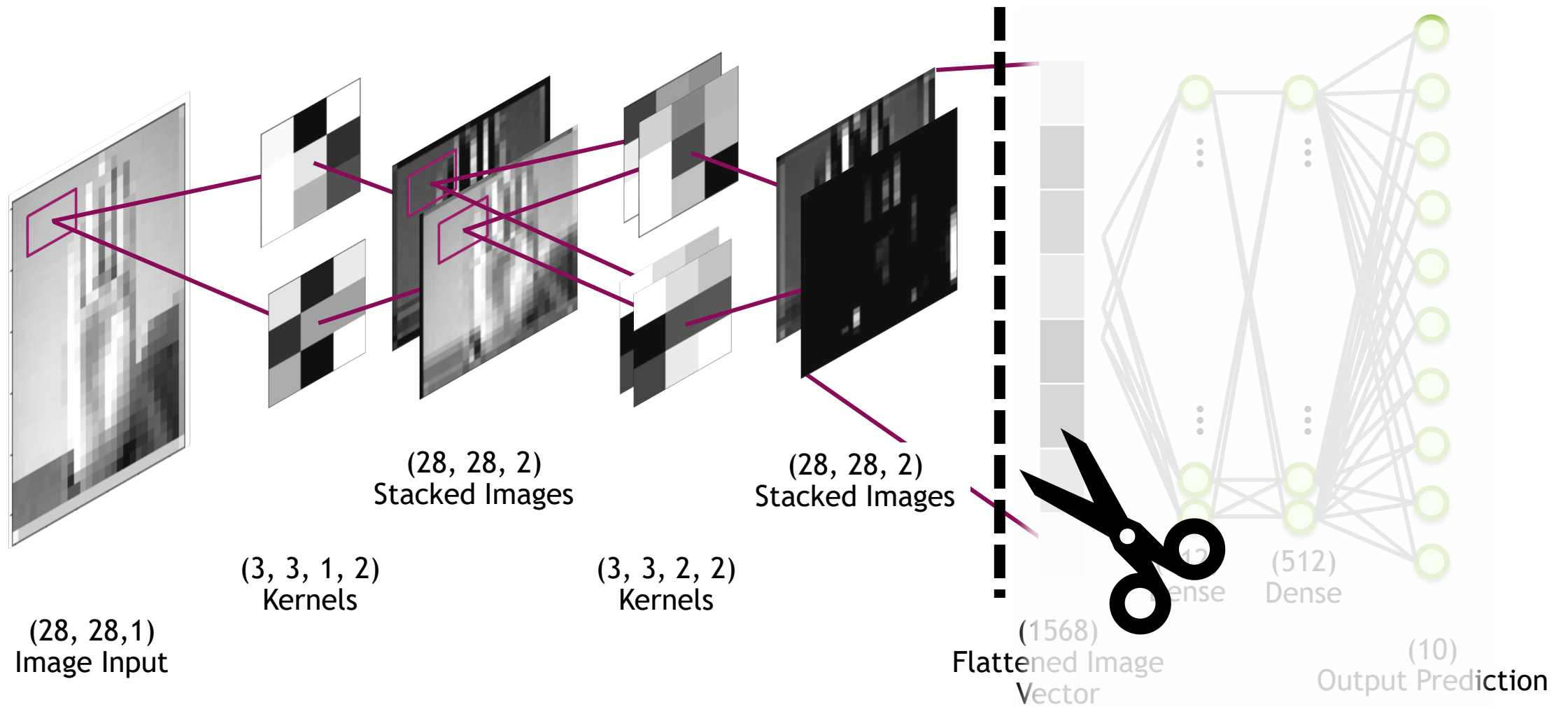
Visual Geometry Group, Department of Engineering Science, University of Oxford

{karen, az}@robots.ox.ac.uk

IM  GENET



# TRANSFER LEARNING

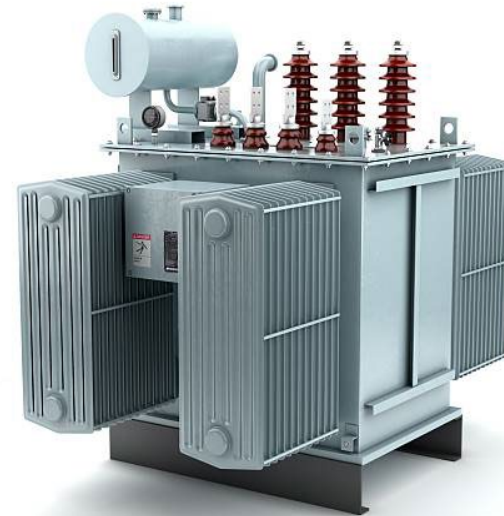


# Transformer

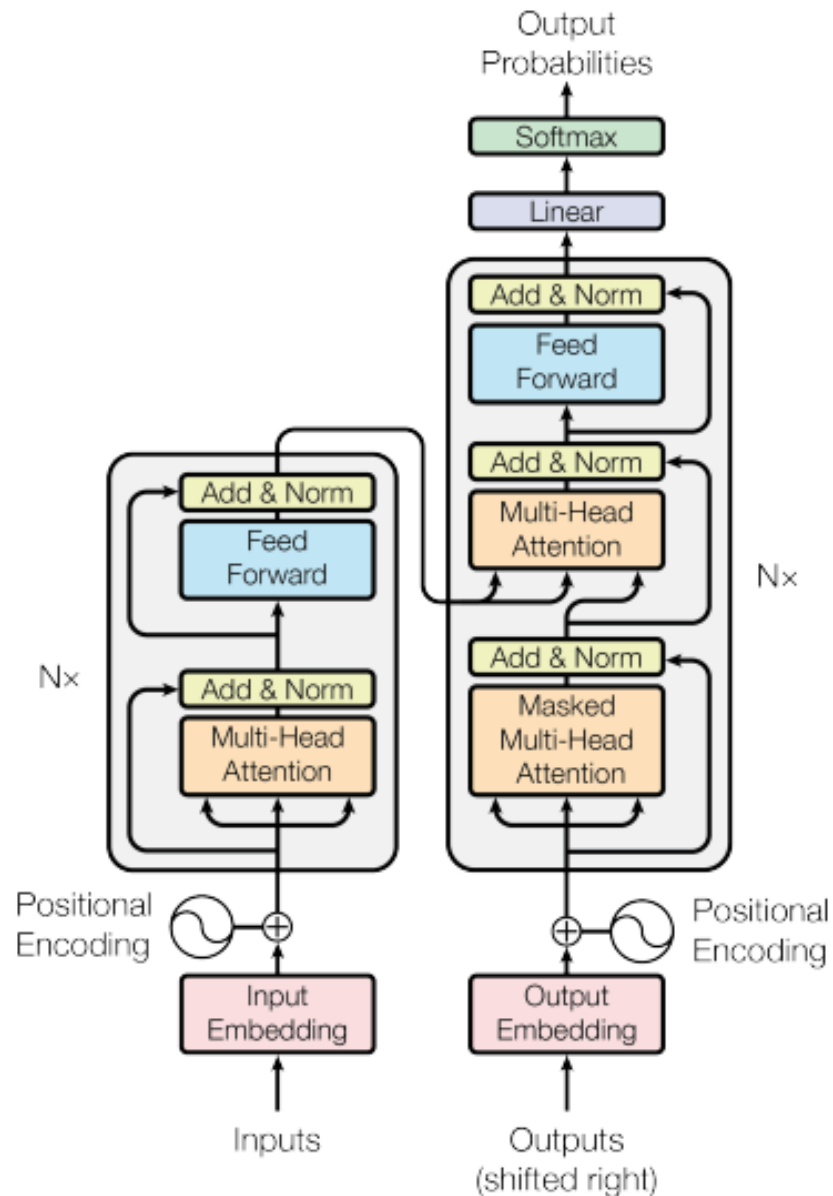
- GPT: Generative Pre-trained Transformer
- Vision Transformer: DETR
- Attention mechanism



© shutterstock.com - 2311573547







Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

# GPT 3

- Parameters 175 B
- Dataset 45T
- 96 attention heads
- 2048 token size
- Learn from their chief scientist:  
<https://www.youtube.com/watch?v=kCc8FmEb1nY>