



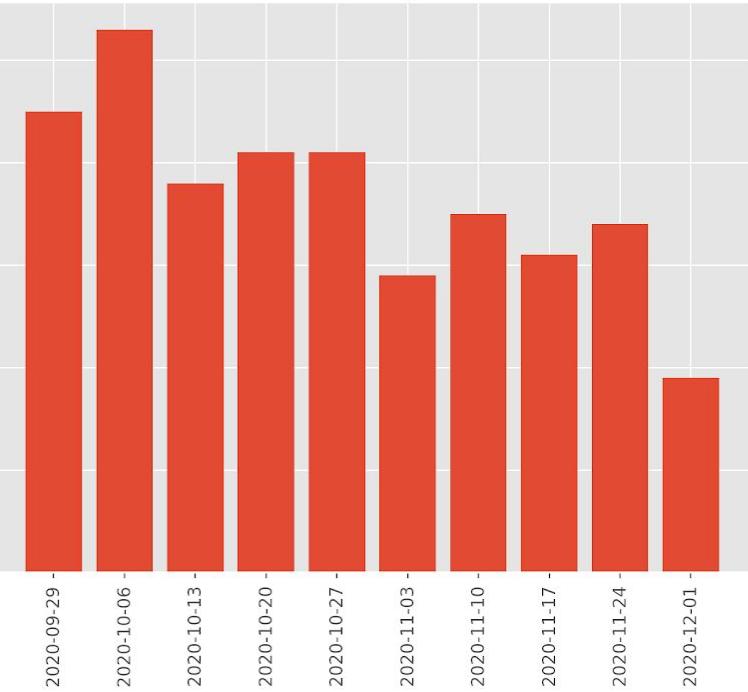
Лекция 11. Сверточные нейронные сети

Господинов Георгий

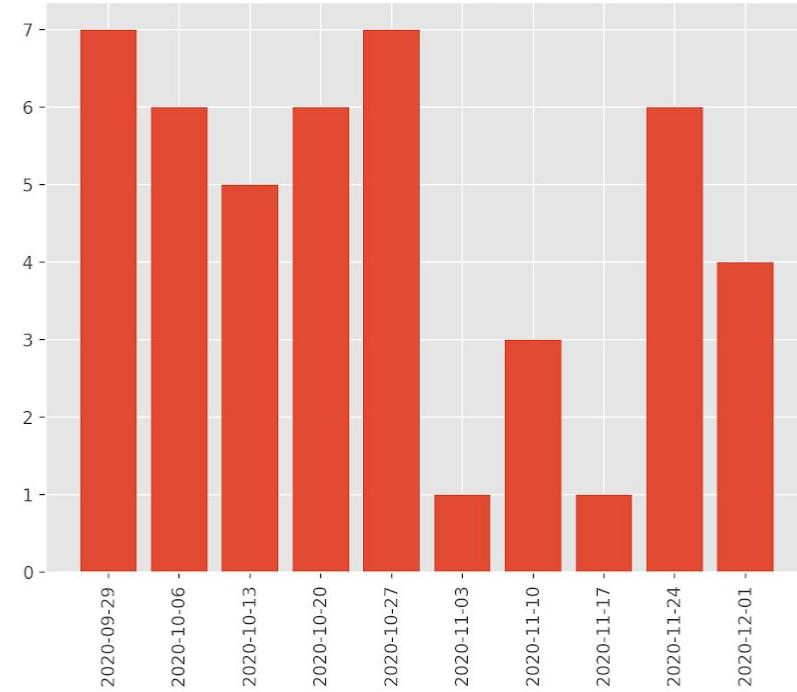


Статистика обратной связи по курсу

Количество слушателей



Количество оценок



Количество отзывов



Не забудьте отметиться на портале!

План занятия

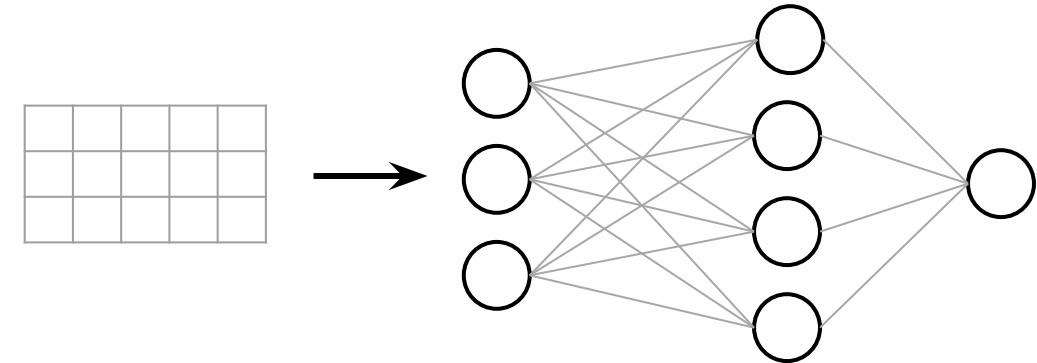
- 1.** Рассказать план
- 2.** Полносвязные сети (Fully Connected)
- 3.** Задачи компьютерного зрения
- 4.** ImageNet
- 5.** Представление изображений
- 6.** Convolution
 - Feature Transformation
 - Pattern Finding
 - Weight Sharing
 - Receptive Field
- 7.** Pooling
- 8.** Борьба с переобучением
- 9.** Transfer Learning
 - Feature Extraction
 - Fine Tuning
- 10.** CNN Architectures
- 11.** Домашнее задание

A grayscale photograph of a computer setup on a desk. It includes a monitor displaying a grid pattern, a keyboard, a mouse, and a closed laptop. The background is plain white.

Fully Connected Neural Networks

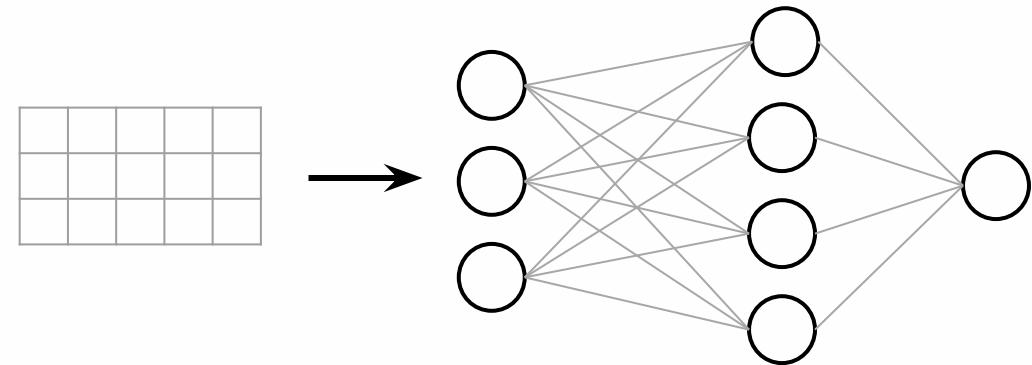
Recap: Fully Connected NN

- зачем нужны?
- из чего состоят?



Recap: Fully Connected NN

- зачем нужны?
- из чего состоят?
- зачем нужна нелинейность?



Recap: Fully Connected NN

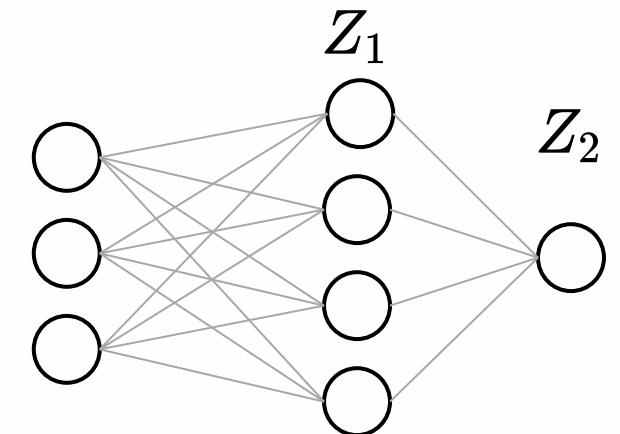
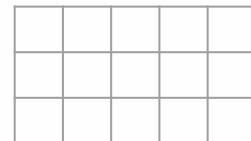
- зачем нужны?
- из чего состоят?
- зачем нужна нелинейность?

$$Z_1 = f(W_1 X + b_1)$$

$$Z_2 = f(W_2 Z_1 + b_2)$$

$$Z_2 = f(W_2 f(W_1 X + b_1) + b_2)$$

$$X : (3 \times \text{batch_size})$$



$$W_1 : (4 \times 3)$$

$$b_1 : (4 \times 1)$$

$$W_2 : (1 \times 4)$$

$$b_2 : (1 \times 1)$$

Recap: Fully Connected NN

- зачем нужны?
- из чего состоят?
- зачем нужна нелинейность?

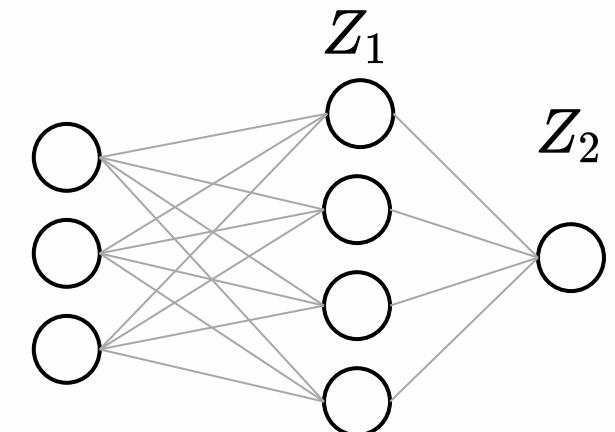
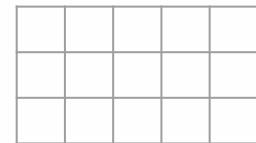
$$Z_1 = f(W_1 X + b_1)$$

$$Z_2 = f(W_2 Z_1 + b_2)$$

$$Z_2 = f(W_2 f(W_1 X + b_1) + b_2)$$

Если бы не было нелинейности:

$$X : (3 \times \text{batch_size})$$



$$W_1 : (4 \times 3)$$

$$b_1 : (4 \times 1)$$

$$W_2 : (1 \times 4)$$

$$b_2 : (1 \times 1)$$

$$Z_2 = W_2 W_1 X + W_2 b_1 + b_2$$

Recap: Fully Connected NN

- зачем нужны?
- из чего состоят?
- зачем нужна нелинейность?

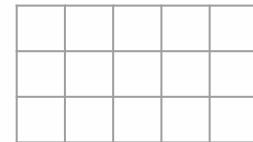
$$Z_1 = f(W_1 X + b_1)$$

$$Z_2 = f(W_2 Z_1 + b_2)$$

$$Z_2 = f(W_2 f(W_1 X + b_1) + b_2)$$

Если бы не было нелинейности:

$$X : (3 \times \text{batch_size})$$



$$Z_1$$

$$Z_2$$

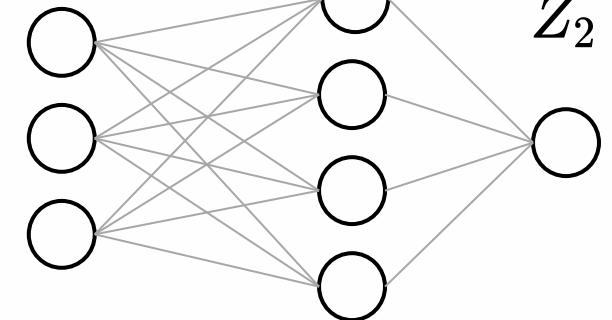
$$W_1 : (4 \times 3)$$

$$W_2 : (1 \times 4)$$

$$b_1 : (4 \times 1)$$

$$b_2 : (1 \times 1)$$

$$Z_2 = W_2 W_1 X + W_2 b_1 + b_2$$
$$(1 \times 4)(4 \times 3) \quad (1 \times 4)(4 \times 1) \quad (1 \times 1)$$



Recap: Fully Connected NN

- зачем нужны?
- из чего состоят?
- зачем нужна нелинейность?

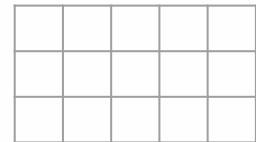
$$Z_1 = f(W_1 X + b_1)$$

$$Z_2 = f(W_2 Z_1 + b_2)$$

$$Z_2 = f(W_2 f(W_1 X + b_1) + b_2)$$

Если бы не было нелинейности:

$$X : (3 \times \text{batch_size})$$



$$Z_1$$

$$Z_2$$

$$W_1 : (4 \times 3)$$

$$W_2 : (1 \times 4)$$

$$b_1 : (4 \times 1)$$

$$b_2 : (1 \times 1)$$

$$Z_2 = W_2 W_1 X + W_2 b_1 + b_2$$

$$(1 \times 4)(4 \times 3) \quad (1 \times 4)(4 \times 1) \quad (1 \times 1)$$

$$Z_2 = \underset{(1 \times 3)}{W' X} + \underset{(1 \times 1)}{b'}$$

Recap: Fully Connected NN



<https://vk.com/weirdreparametrizationtrick>



<https://www.facebook.com/convolutionalmemes/>

#011



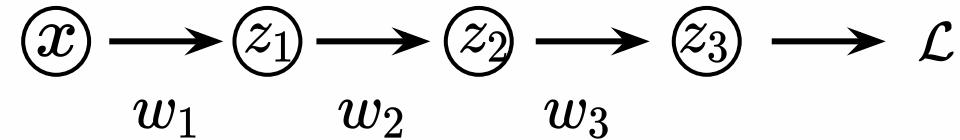
Vanishing Gradients Problem

$$f(x) = \sigma(x)$$

$$z_1 = f(w_1 x + b_1)$$

$$z_2 = f(w_2 z_1 + b_2)$$

$$z_3 = f(w_3 z_2 + b_3)$$





Vanishing Gradients Problem

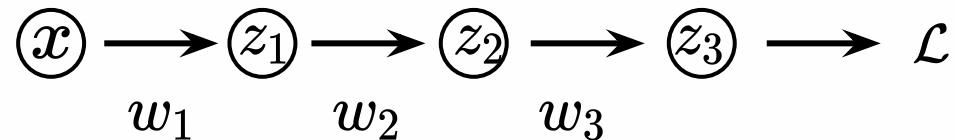
$$f(x) = \sigma(x)$$

$$z_1 = f(w_1 x + b_1)$$

$$z_2 = f(w_2 z_1 + b_2)$$

$$z_3 = f(w_3 z_2 + b_3)$$

$$\mathcal{L} = \frac{1}{2}(z_3 - t)^2$$

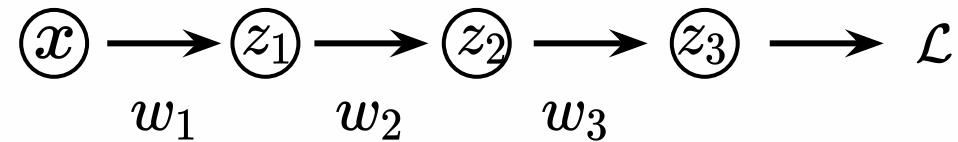




Vanishing Gradients Problem

$$f(x) = \sigma(x)$$

$$z_1 = f(w_1 x + b_1)$$



$$z_2 = f(w_2 z_1 + b_2)$$

$$z_3 = f(w_3 z_2 + b_3)$$

$$\mathcal{L} = \frac{1}{2}(z_3 - t)^2$$

$$w_k^{t+1} = w_k^t - \eta \frac{\partial \mathcal{L}}{\partial w_k}$$

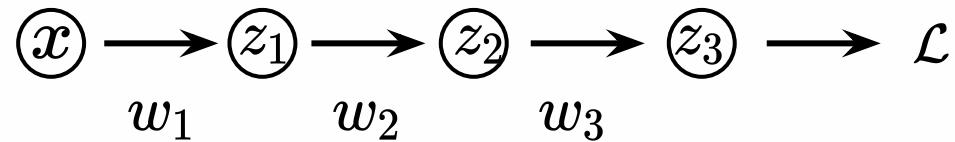
$$\frac{\partial \mathcal{L}}{\partial w_3} =$$



Vanishing Gradients Problem

$$f(x) = \sigma(x)$$

$$z_1 = f(w_1 x + b_1)$$



$$z_2 = f(w_2 z_1 + b_2)$$

$$z_3 = f(w_3 z_2 + b_3)$$

$$\mathcal{L} = \frac{1}{2}(z_3 - t)^2$$

$$w_k^{t+1} = w_k^t - \eta \frac{\partial \mathcal{L}}{\partial w_k}$$

$$\frac{\partial \mathcal{L}}{\partial w_3} =$$

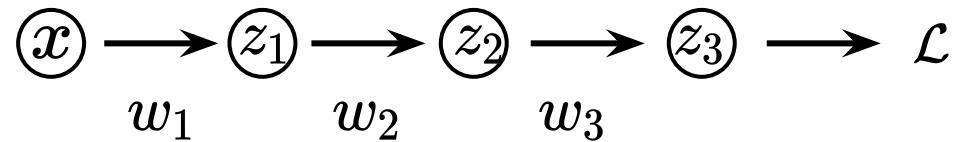
$$\frac{\partial \mathcal{L}}{\partial w_3} = \frac{\partial \mathcal{L}}{\partial z_3} \frac{\partial z_3}{\partial w_3} =$$



Vanishing Gradients Problem

$$f(x) = \sigma(x)$$

$$z_1 = f(w_1 x + b_1)$$



$$z_2 = f(w_2 z_1 + b_2)$$

$$z_3 = f(w_3 z_2 + b_3)$$

$$\mathcal{L} = \frac{1}{2}(z_3 - t)^2$$

$$w_k^{t+1} = w_k^t - \eta \frac{\partial \mathcal{L}}{\partial w_k}$$

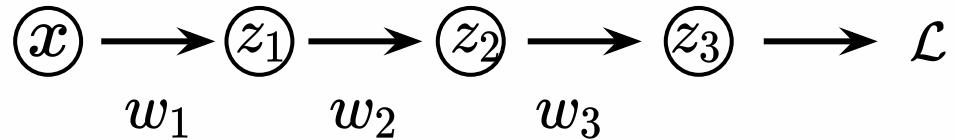
$$\frac{\partial \mathcal{L}}{\partial w_3} =$$

$$\frac{\partial \mathcal{L}}{\partial w_3} = \frac{\partial \mathcal{L}}{\partial z_3} \frac{\partial z_3}{\partial w_3} = (z_3 - t) f'(w_3 z_2 + b_3) z_2$$

Vanishing Gradients Problem

$$f(x) = \sigma(x)$$

$$z_1 = f(w_1 x + b_1)$$



$$z_2 = f(w_2 z_1 + b_2)$$

$$z_3 = f(w_3 z_2 + b_3)$$

$$\mathcal{L} = \frac{1}{2}(z_3 - t)^2$$

$$w_k^{t+1} = w_k^t - \eta \frac{\partial \mathcal{L}}{\partial w_k}$$

$$\frac{\partial \mathcal{L}}{\partial w_3} =$$

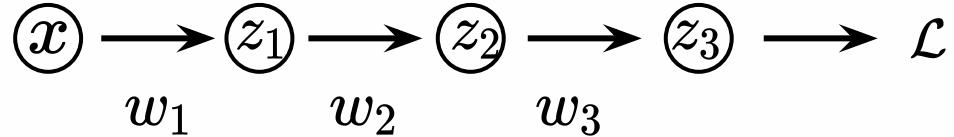
$$\frac{\partial \mathcal{L}}{\partial w_3} = \frac{\partial \mathcal{L}}{\partial z_3} \frac{\partial z_3}{\partial w_3} = (z_3 - t) f'(w_3 z_2 + b_3) z_2$$

$$\frac{\partial \mathcal{L}}{\partial w_3} = \frac{\partial \mathcal{L}}{\partial z_3} \frac{\partial z_3}{\partial w_3} = (z_3 - t) f'(w_3 z_2 + b_3) z_2 \leq (z_3 - t) \frac{1}{4} z_2$$

Vanishing Gradients Problem

$$f(x) = \sigma(x)$$

$$z_1 = f(w_1 x + b_1)$$



$$z_2 = f(w_2 z_1 + b_2)$$

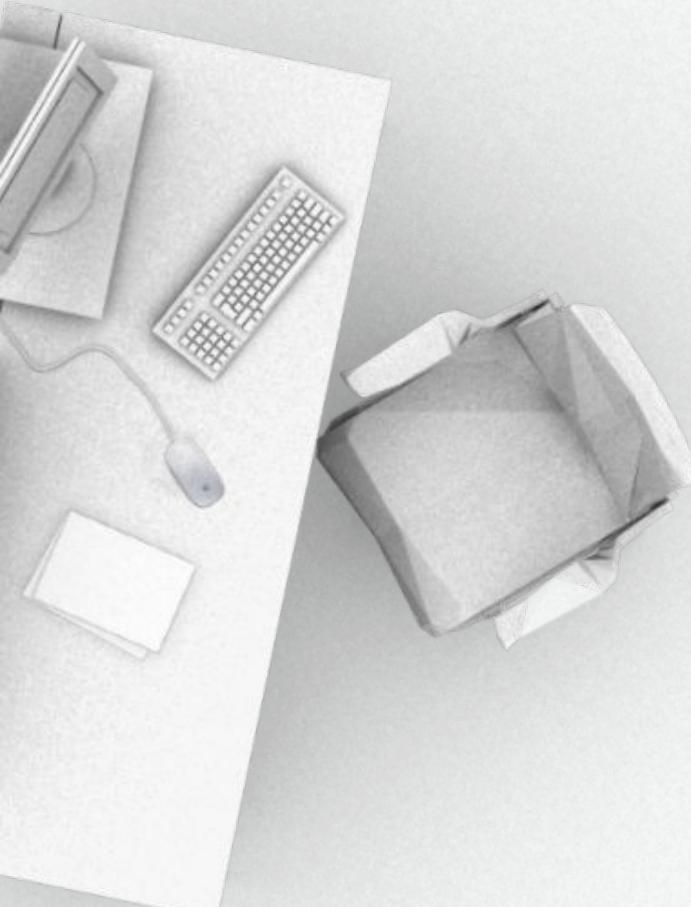
$$z_3 = f(w_3 z_2 + b_3)$$

$$\mathcal{L} = \frac{1}{2}(z_3 - t)^2 \quad w_k^{t+1} = w_k^t - \eta \frac{\partial \mathcal{L}}{\partial w_k}$$

$$\frac{\partial \mathcal{L}}{\partial w_3} = \frac{\partial \mathcal{L}}{\partial z_3} \frac{\partial z_3}{\partial w_3} = (z_3 - t) f'(w_3 z_2 + b_3) z_2$$

$$\frac{\partial \mathcal{L}}{\partial w_3} = \frac{\partial \mathcal{L}}{\partial z_3} \frac{\partial z_3}{\partial w_3} = (z_3 - t) f'(w_3 z_2 + b_3) z_2 \leq (z_3 - t) \frac{1}{4} z_2$$

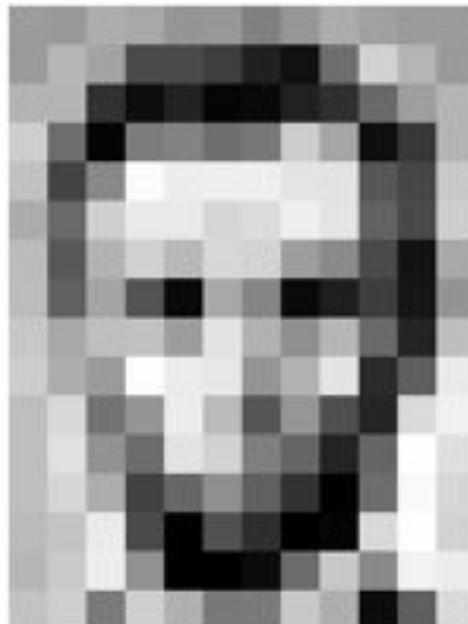
$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{\partial \mathcal{L}}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial w_1} = (z_3 - t) f'(w_3 z_2 + b_3) w_3 f'(w_2 z_1 + b_2) w_2 f'(w_1 x + b_1) x \leq (z_3 - t) (\frac{1}{4})^3 w_3 w_2 x$$



Представление изображений

Представление изображений: ЧБ

Pixel = Picture Element

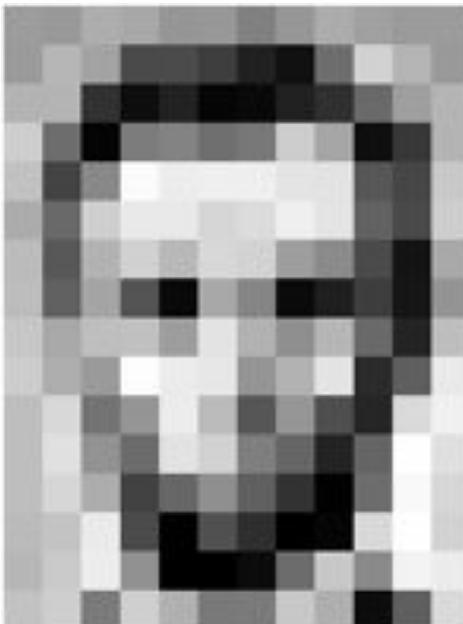


#020

https://openframeworks.cc/ofBook/chapters/image_processing_computer_vision.html

Представление изображений: ЧБ

Pixel = Picture Element



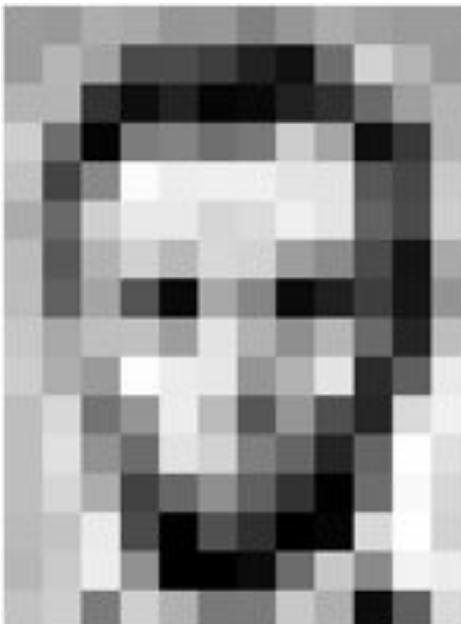
187	188	174	168	160	162	129	181	172	140	168	188
186	180	169	74	76	62	33	17	118	210	180	154
180	180	80	14	34	4	10	33	48	194	169	181
204	198	5	134	197	111	100	204	186	15	54	180
194	68	182	251	237	239	239	238	237	67	71	201
172	138	207	233	233	214	220	236	238	86	14	206
148	84	179	309	186	215	211	184	199	78	20	149
189	97	186	84	10	168	184	91	20	42	22	146
198	186	191	193	186	227	178	149	182	78	38	190
205	174	186	252	236	281	149	178	238	49	95	234
196	214	218	149	236	187	86	150	79	38	258	241
198	234	147	188	227	210	127	102	36	181	268	234
196	214	173	45	188	149	96	56	2	198	249	215
187	186	235	73	1	81	47	0	6	217	258	211
183	202	237	148	6	8	12	188	200	188	243	236
195	206	123	267	177	121	129	206	176	18	96	218

#021

https://openframeworks.cc/ofBook/chapters/image_processing_computer_vision.html

Представление изображений: ЧБ

Pixel = Picture Element



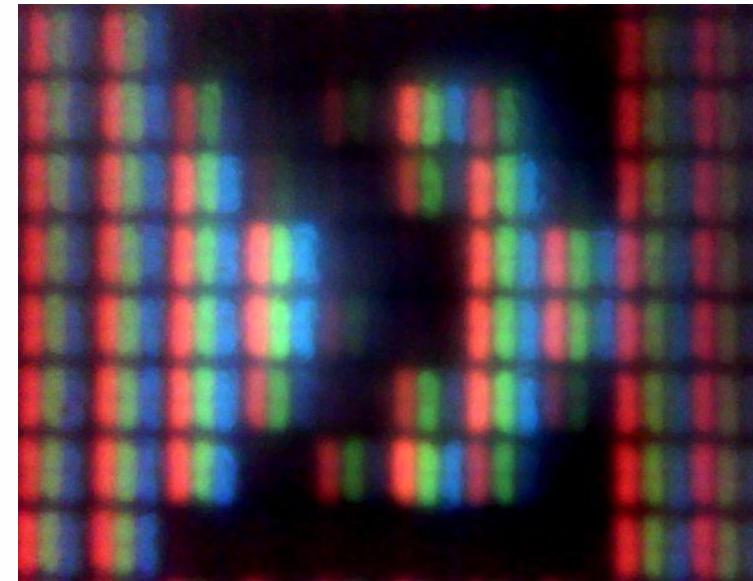
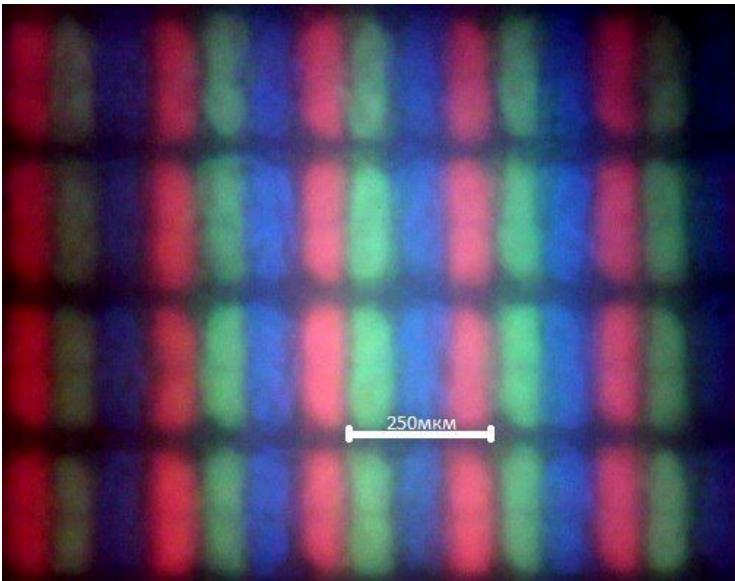
187	188	174	168	160	162	129	181	172	140	168	188
186	180	60	14	34	4	10	33	48	104	169	181
204	198	5	134	197	111	100	204	186	15	54	180
194	68	182	251	237	239	239	238	237	57	71	201
172	138	267	233	233	214	220	236	238	86	14	206
148	84	179	309	186	215	211	184	199	78	20	148
189	97	186	84	16	168	184	91	20	43	22	146
198	148	193	193	227	178	149	182	786	38	190	
205	174	188	252	296	281	349	178	238	49	95	234
196	218	218	149	236	187	86	150	79	38	258	241
198	234	147	188	227	210	127	102	36	181	268	234
196	214	173	45	188	149	96	56	2	198	249	216
187	186	235	73	1	81	47	0	6	217	258	211
183	202	237	148	6	8	12	188	200	188	243	236
195	206	123	267	177	121	129	206	118	18	96	218

187	188	174	168	180	162	129	181	172	161	188	188
185	182	163	74	75	62	33	17	110	218	180	181
180	180	58	14	34	6	10	33	48	126	159	181
206	108	8	124	131	111	120	204	186	16	56	180
194	68	137	251	237	239	239	238	237	87	71	201
172	106	207	239	239	214	220	236	238	98	74	206
188	88	179	209	186	216	211	188	139	75	20	149
189	97	185	64	10	168	184	11	31	62	32	148
196	168	191	183	198	227	179	143	182	126	36	190
205	174	188	252	236	231	149	178	239	43	95	234
190	216	114	149	236	187	86	150	79	36	218	241
190	234	147	198	227	210	127	102	36	121	256	238
190	214	173	68	188	143	98	56	2	108	249	215
187	196	235	79	1	81	47	0	6	217	258	211
183	202	237	148	6	8	12	188	200	188	243	236
195	206	123	267	177	121	129	206	118	18	96	218

#022

https://openframeworks.cc/ofBook/chapters/image_processing_computer_vision.html

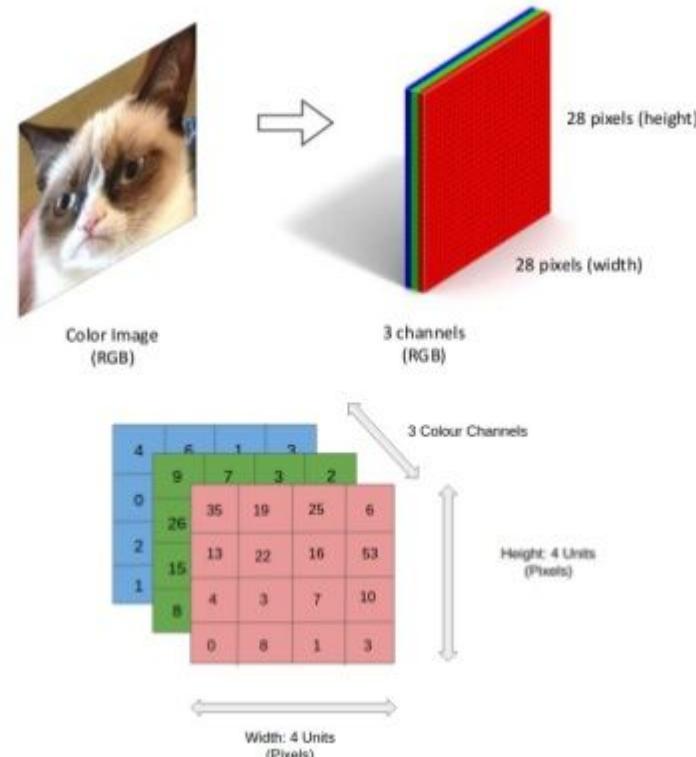
Пиксели дисплея



#023

Цветное изображение

color image is 3rd-order tensor



#024

<https://www.slideshare.net/BertonEarnshaw/a-brief-survey-of-tensors>

Цветное изображение



Мифы

- До CNN не было Computer Vision
<https://www.youtube.com/watch?v=zISjTXwmvFQ>
- После CNN классические методы перестали использоваться
<https://www.youtube.com/watch?v=UqMxCzwxM2o>



ImageNet

ImageNet



Large Scale Visual Recognition Challenge

Одна из задач: классификация изображений на 1000 классов

<https://arxiv.org/pdf/1409.0575.pdf>

#028

Image Sample



mite

container ship

motor scooter

leopard

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



grille

mushroom

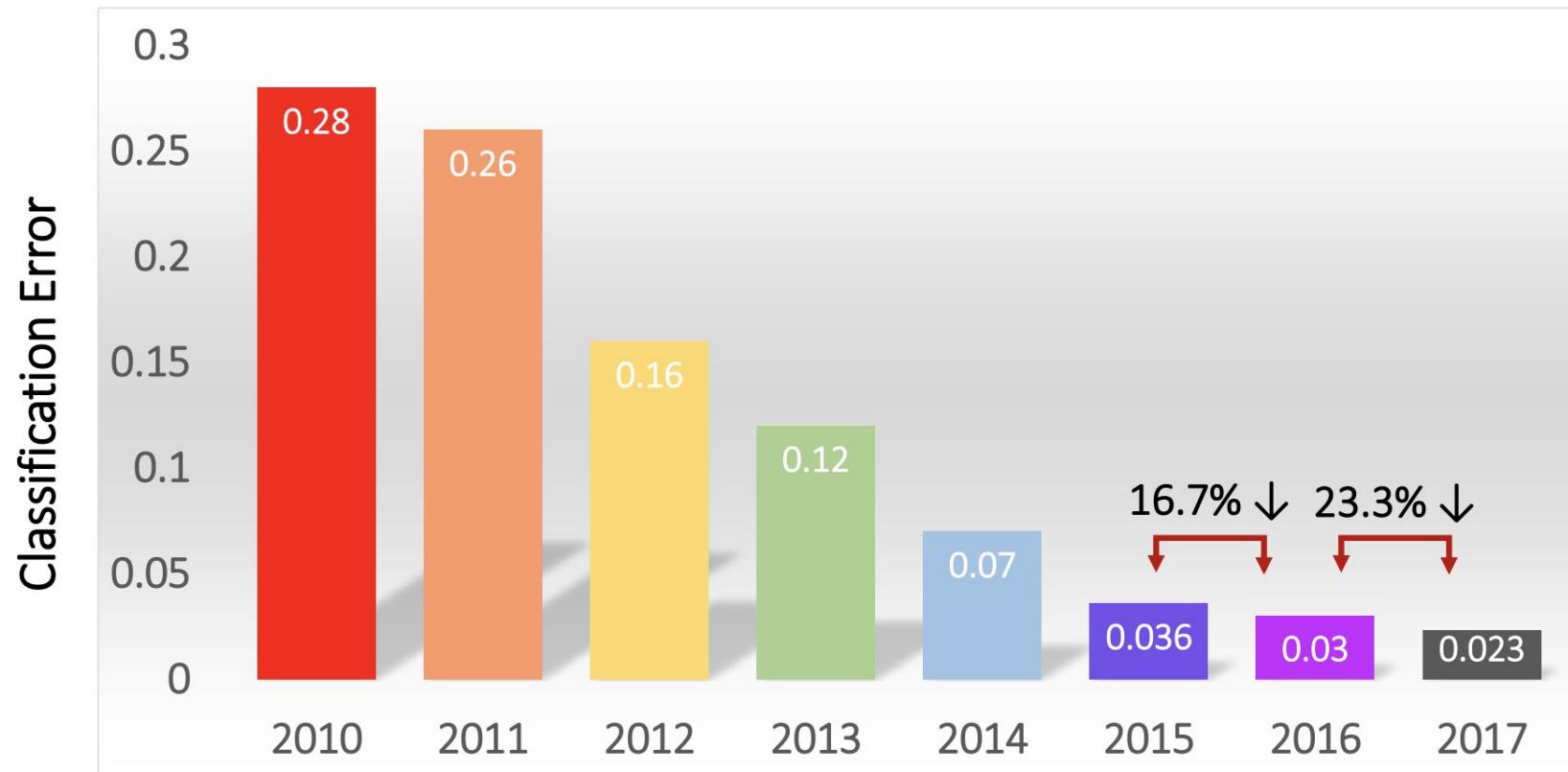
cherry

Madagascar cat

convertible	agaric	dalmatian	squirrel monkey
grille	mushroom	grape	spider monkey
pickup	jelly fungus	elderberry	titi
beach wagon	gill fungus	ffordshire bullterrier	indri
fire engine	dead-man's-fingers	currant	howler monkey

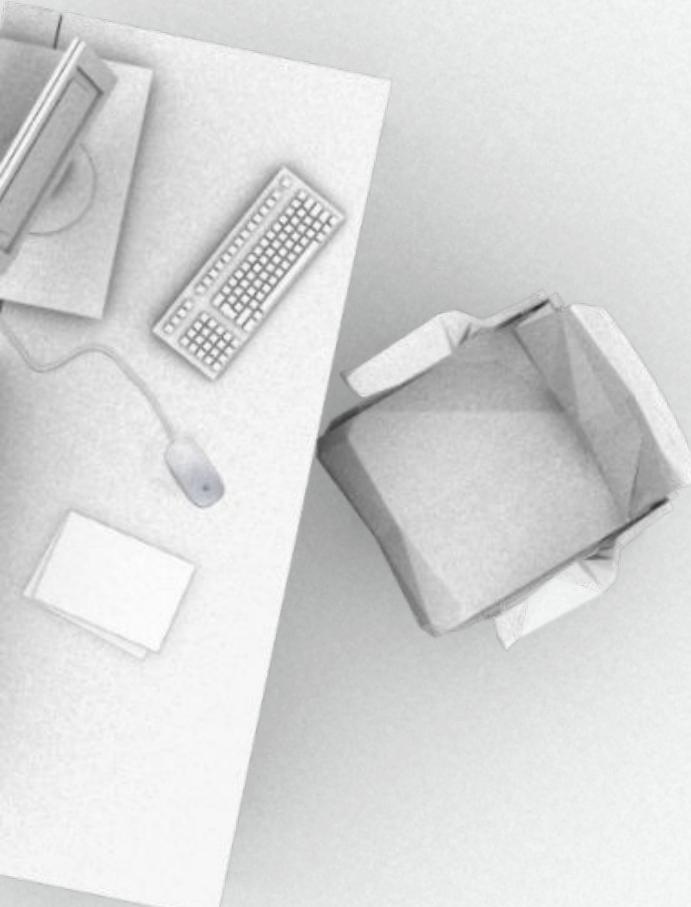
#029

Top-5 Classification Error



http://image-net.org/challenges/talks_2017/ILSVRC2017_overview.pdf

#030



Convolution

Convolution

В Deep Learning под сверткой понимается кросс-корреляция

A — исходный массив

w — ядро свертки

K — размер ядра

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

$$A = \begin{array}{|c|c|c|c|c|c|} \hline 3 & 1 & 4 & 1 & 5 & 9 \\ \hline \end{array}$$
$$w = \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline \end{array}$$
$$\xrightarrow{\hspace{2cm}} K$$

Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$

$$A = \begin{array}{|c|c|c|c|c|c|} \hline 3 & 1 & 4 & 1 & 5 & 9 \\ \hline \end{array}$$

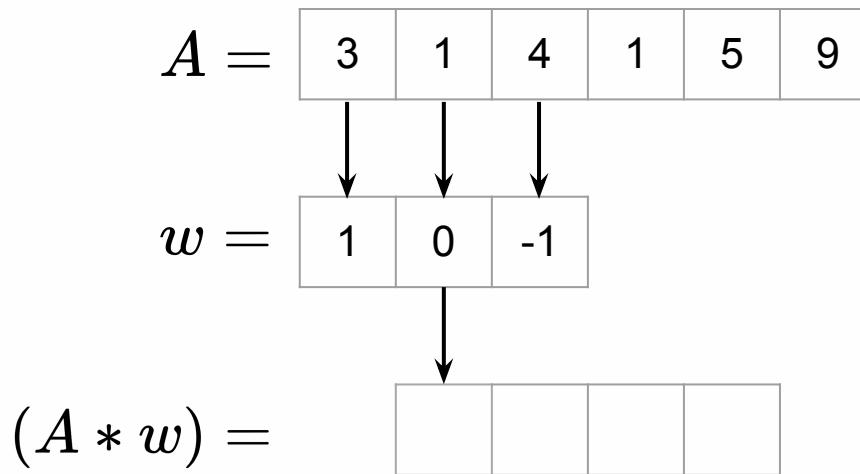
$$w = \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|c|c|} \hline \quad & \quad & \quad & \quad \\ \hline \end{array}$$

Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

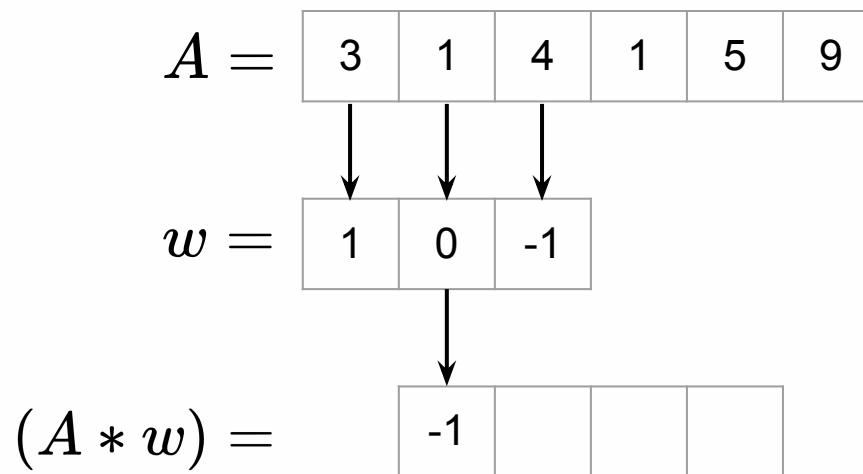
$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$



Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

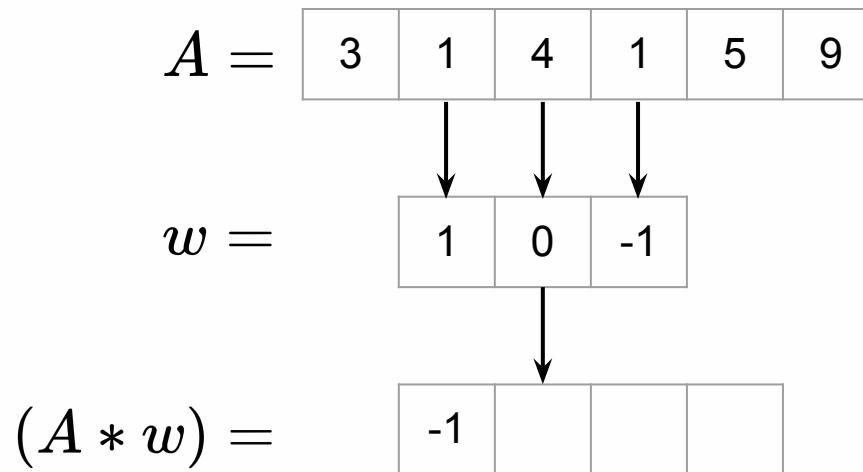
$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$



Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

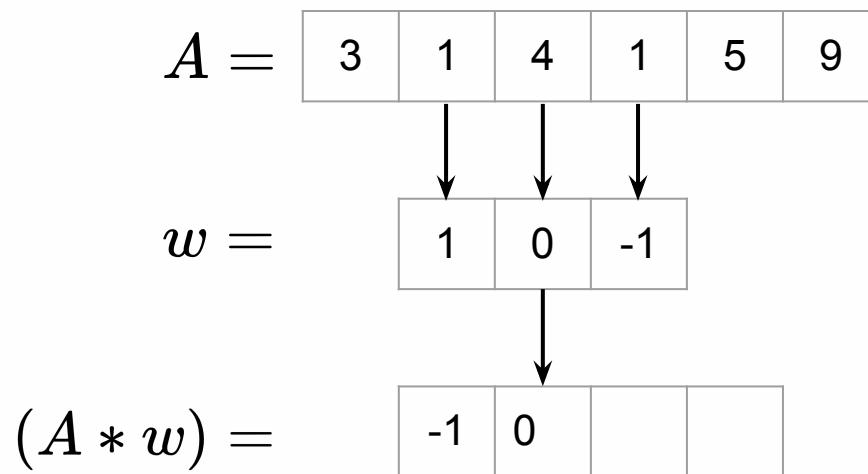
$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$



Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

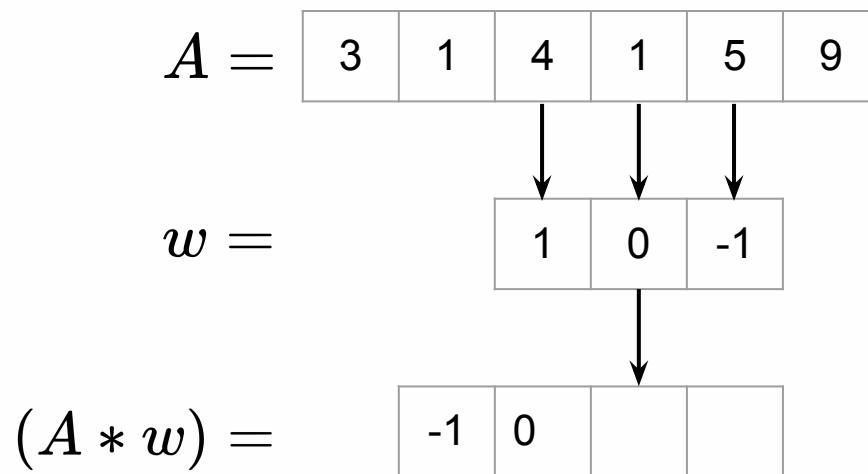
$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$



Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

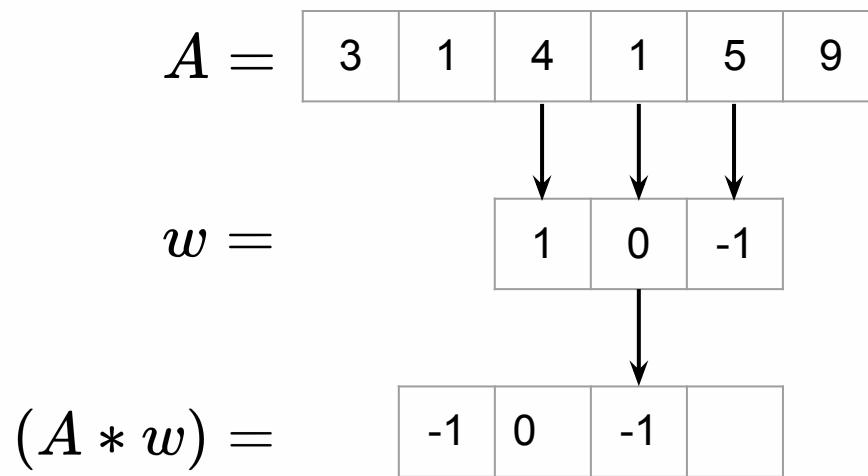
$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$



Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

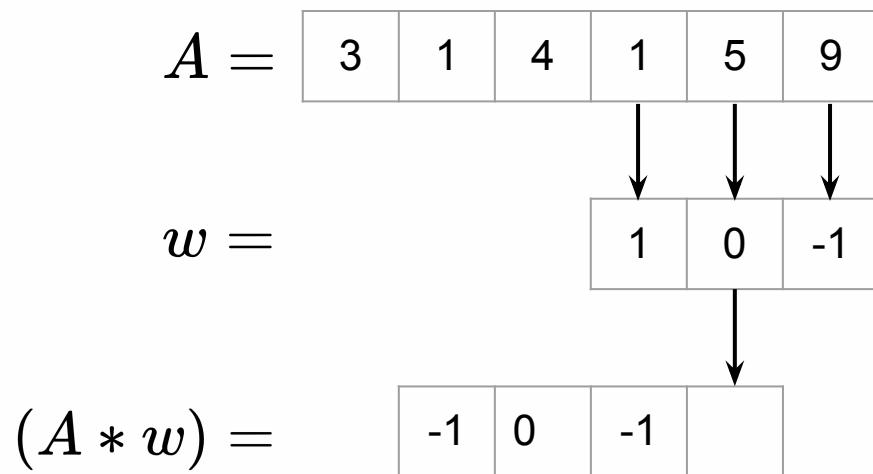
$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$



Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

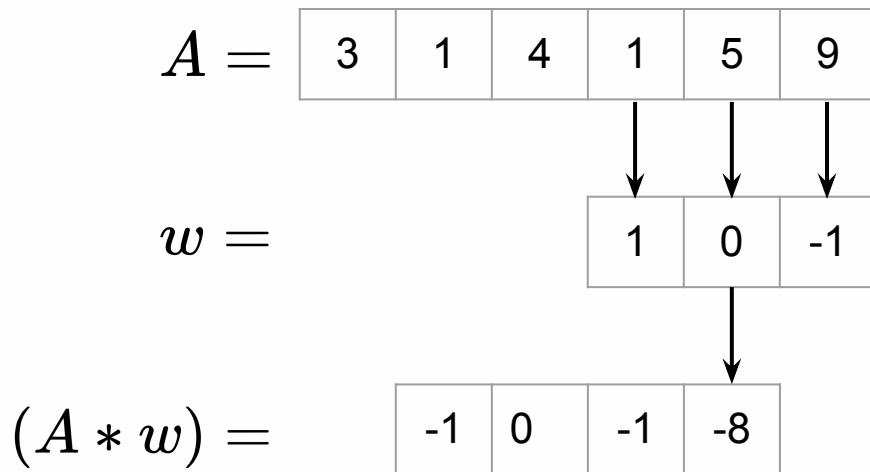
$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$



Convolution

$$(A * w)_i = \sum_{i'=0}^{K-1} A(i + i')w(i')$$

$$(A * w)_0 = A(0) \cdot w(0) + A(1) \cdot w(1) + A(2) \cdot w(2)$$



Convolution

Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$
$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$
$$(A * w) = \begin{array}{|c|c|} \hline & \\ \hline & \\ \hline \end{array}$$

Convolution

Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline \boxed{ } & \boxed{ } \\ \hline \boxed{ } & \boxed{ } \\ \hline \end{array}$$

Convolution

Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline -2 & \\ \hline & \\ \hline \end{array}$$

Convolution

Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline -2 & \boxed{} \\ \hline \end{array}$$

Convolution

Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline -2 & -8 \\ \hline \end{array}$$

Convolution

Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline -2 & -8 \\ \hline \boxed{} & \boxed{} \\ \hline \end{array}$$

Convolution

Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline -2 & -8 \\ \hline \boxed{-1} & \\ \hline \end{array}$$

Convolution

Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline -2 & -8 \\ \hline -1 & \boxed{} \\ \hline \end{array}$$

Convolution

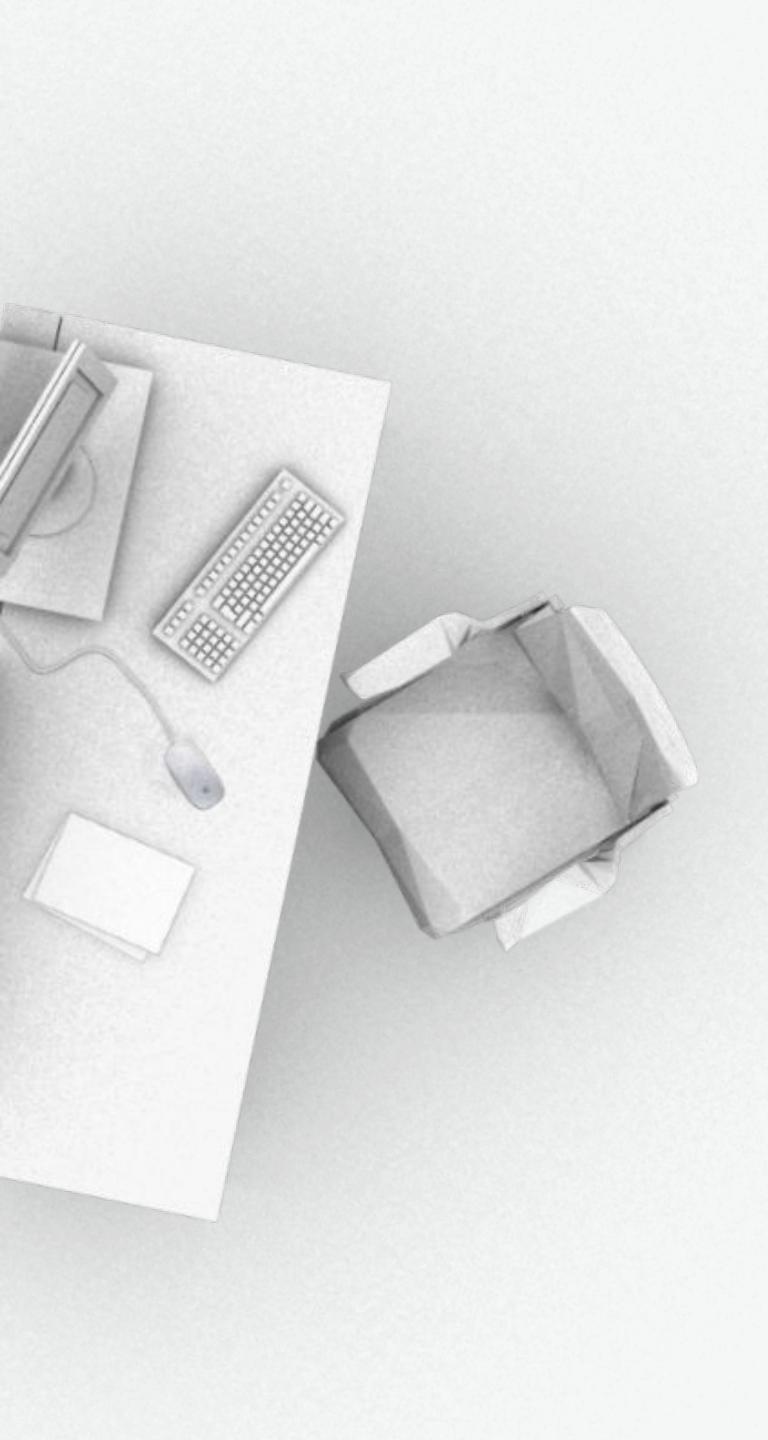
Двумерный случай:

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline -2 & -8 \\ \hline -1 & 0 \\ \hline \end{array}$$

A grayscale illustration of a computer setup on a desk. It includes a monitor displaying a grid pattern, a keyboard, a mouse, and some papers. A large, open book or folder is prominently displayed in the foreground.

Convolution: Feature Transformation

Feature Maps



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Feature Maps

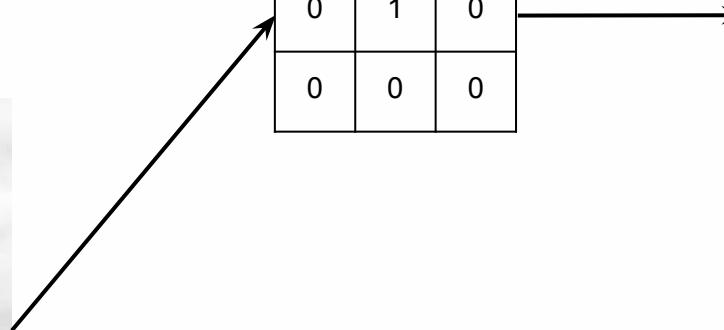


0	0	0
0	1	0
0	0	0

Feature Maps



0	0	0
0	1	0
0	0	0



Feature Maps



0	0	0
0	1	0
0	0	0



1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Feature Maps



0	0	0
0	1	0
0	0	0



1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



Feature Maps

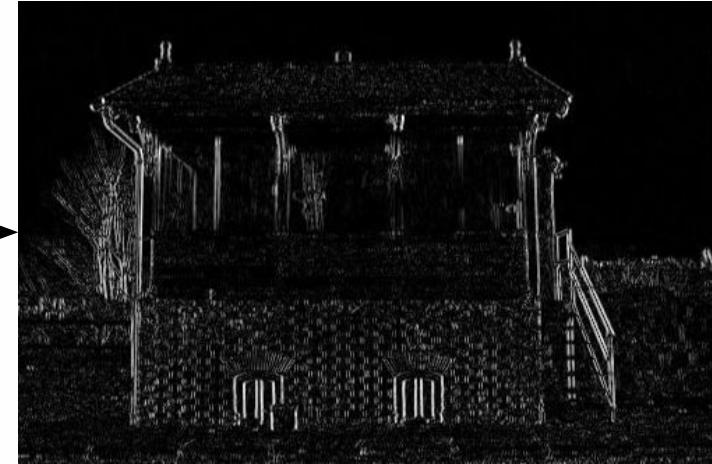


1	0	-1
2	0	-2
1	0	-1

Feature Maps



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

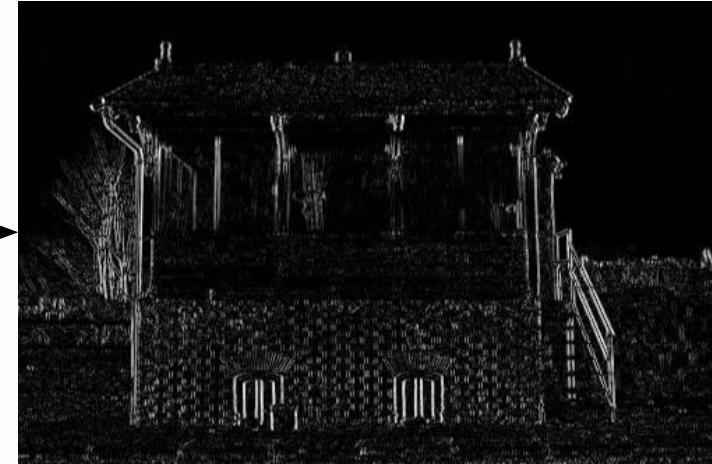


Feature Maps



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

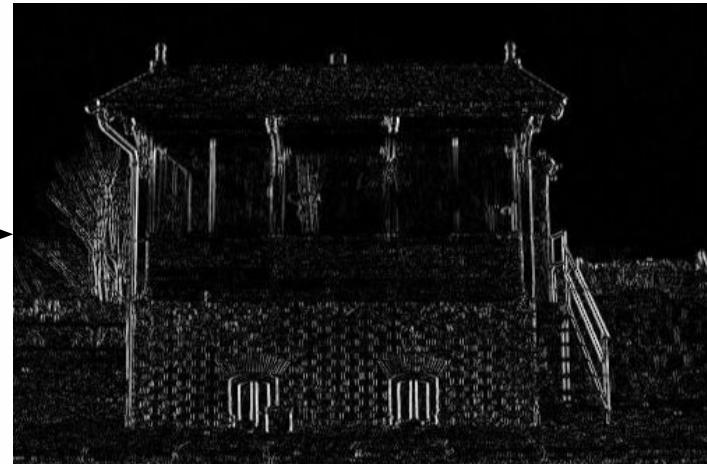
1	2	1
0	0	0
-1	-2	-1



Feature Maps



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

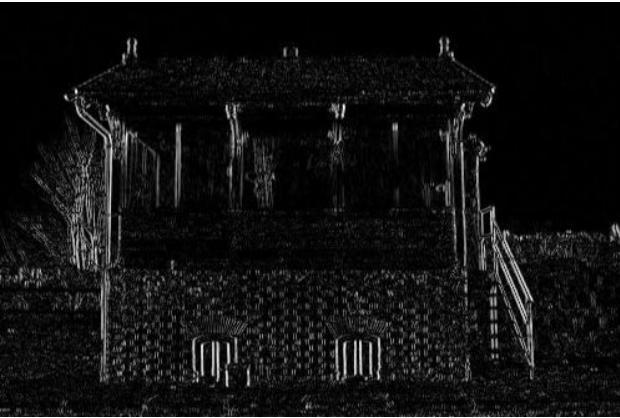


1	2	1
0	0	0
-1	-2	-1



Feature Maps

$$L_x =$$



$$L_y =$$



$$\sqrt{L_x^2 + L_y^2} =$$

Feature Maps

$$L_x =$$



$$L_y =$$



$$\sqrt{L_x^2 + L_y^2} =$$



Convolution as Feature Transformation



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



Convolution as Feature Transformation



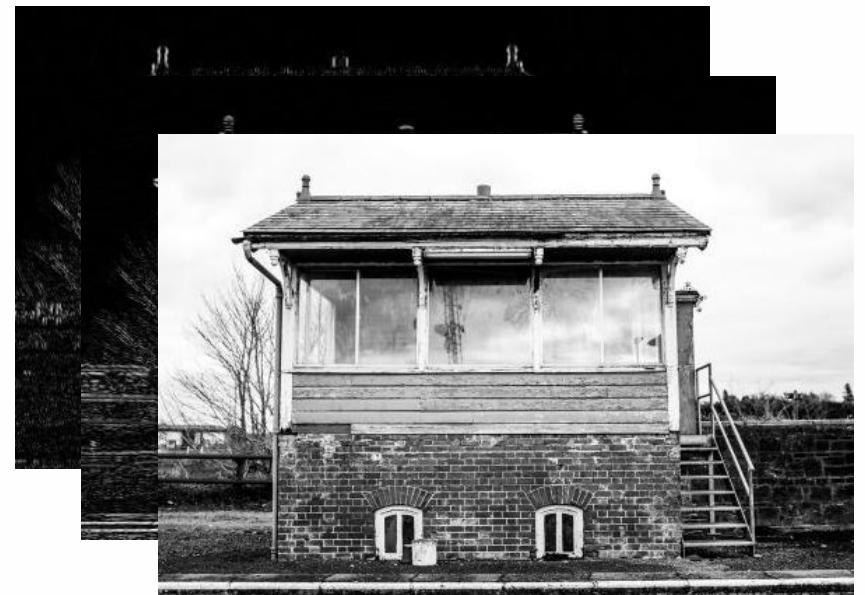
1	0	-1	
2	1	2	1
0	0	0	
1	-1	-2	-1



Convolution as Feature Transformation



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



Convolution as Feature Transformation



1	0	-1
1	2	1
2	0	1/9 1/9 1/9
1	-1	1/9 1/9 1/9
0	0	1/9 1/9 1/9



Convolution as Feature Transformation

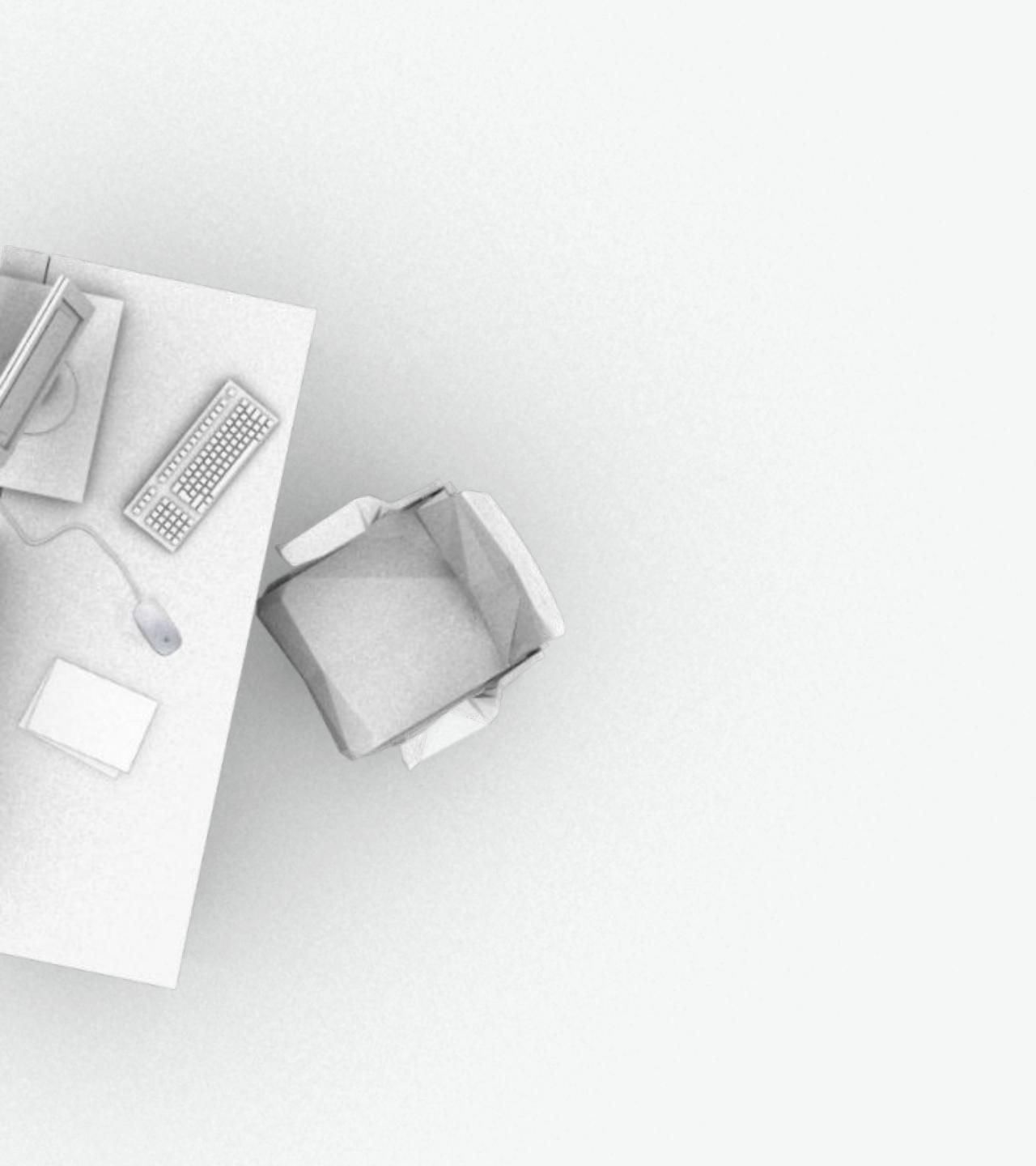


1	0	-1
1	2	1
2	0	1/9 1/9 1/9
1	-1	1/9 1/9 1/9
0	0	1/9 1/9 1/9

filter bank



feature maps



Convolution: Pattern Finding

Similarity Measures

$a =$

1
0
3
0
-1

$b =$

2
0
6
0
-2

Similarity Measures

$$a = \begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline 3 \\ \hline 0 \\ \hline -1 \\ \hline \end{array}$$
$$b = \begin{array}{|c|} \hline 2 \\ \hline 0 \\ \hline 6 \\ \hline 0 \\ \hline -2 \\ \hline \end{array}$$

$$a^T b = \sum_i a_i b_i = |a||b| \cos \theta$$

$$\cos \theta = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2} \sqrt{\sum_i b_i^2}}$$

$$\rho = \frac{\sum_i (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_i (a_i - \bar{a})^2} \sqrt{\sum_i (b_i - \bar{b})^2}}$$

Вычисление свертки

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|} \hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline \end{array}$$

$$w = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$(A * w) = \begin{array}{|c|c|} \hline -2 & -8 \\ \hline -1 & 0 \\ \hline \end{array}$$

Вычисление свертки

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|}\hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline\end{array}$$
$$w = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline\end{array}$$
$$(A * w) = \begin{array}{|c|c|}\hline -2 & -8 \\ \hline -1 & 0 \\ \hline\end{array}$$
$$\begin{array}{|c|c|c|c|c|}\hline 1 & 2 & -1 & 0 & 1 \\ \hline -1 & 1 & 3 & 4 & 4 \\ \hline 2 & -3 & 4 & 3 & -4 \\ \hline -3 & 2 & 0 & -1 & 0 \\ \hline 5 & 2 & -4 & 0 & 1 \\ \hline\end{array}$$
 $*$
$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|c|}\hline 0 & -1 & -5 & -4 \\ \hline 2 & -3 & 0 & 8 \\ \hline 0 & -3 & 5 & 3 \\ \hline -5 & 6 & 0 & -1 \\ \hline\end{array}$$

Вычисление свертки

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|}\hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline\end{array}$$

$$w = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline\end{array}$$

$$(A * w) = \begin{array}{|c|c|}\hline -2 & -8 \\ \hline -1 & 0 \\ \hline\end{array}$$

1	2	-1	0	1
-1	1	3	4	4
2	-3	4	3	-4
-3	2	0	-1	0
5	2	-4	0	1

*

$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline\end{array}$$



0	-1	-5	-4
2	-3	0	8
0	-3	5	3
-5	6	0	-1

Вычисление свертки

$$(A * w)_{ij} = \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j') w(i', j')$$

$$A = \begin{array}{|c|c|c|}\hline 3 & 1 & 4 \\ \hline 1 & 5 & 9 \\ \hline 2 & 6 & 5 \\ \hline\end{array}$$

$$w = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline\end{array}$$

$$(A * w) = \begin{array}{|c|c|}\hline -2 & -8 \\ \hline -1 & 0 \\ \hline\end{array}$$

1	2	-1	0	1
-1	1	3	4	4
2	-3	4	3	-4
-3	2	0	-1	0
5	2	-4	0	1

*

$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline\end{array}$$



0	-1	-5	-4
2	-3	0	8
0	-3	5	3
-5	6	0	-1

Pattern Finding



0	1	0
0	1	0
0	1	0

0	0	0
1	1	1
0	0	0

Pattern Finding



0	1	0
0	1	0
0	1	0



0	0	0
1	1	1
0	0	0



Pattern Finding



0	1	0
0	1	0
0	1	0



0	0	0
1	1	1
0	0	0



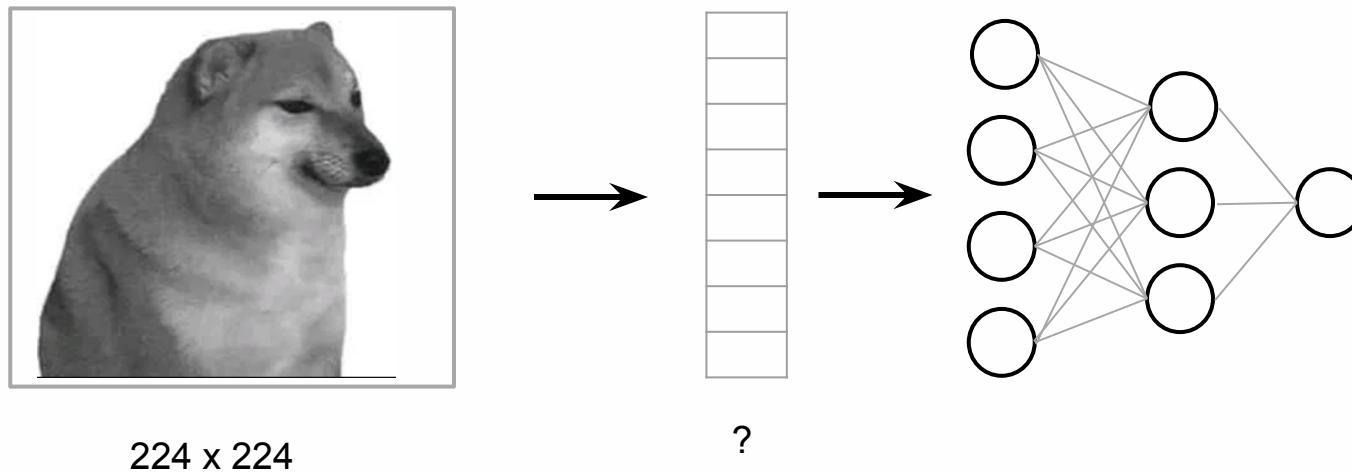


Convolution: Weight Sharing

Weight Sharing: мотивация

Задача: Cat vs Dog

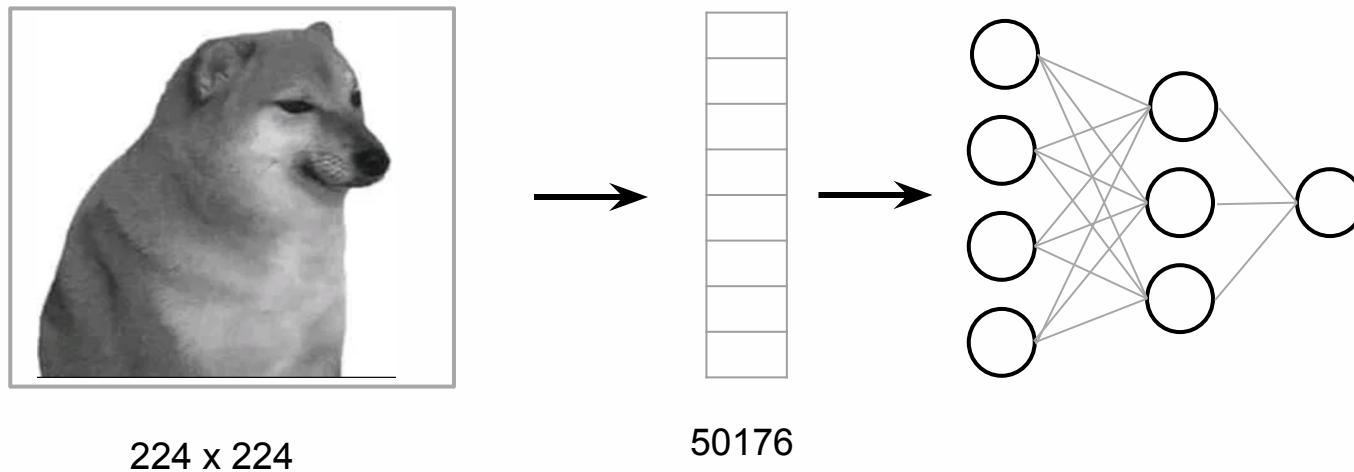
Решаем задачу с помощью полносвязной нейронной сети



Weight Sharing: мотивация

Задача: Cat vs Dog

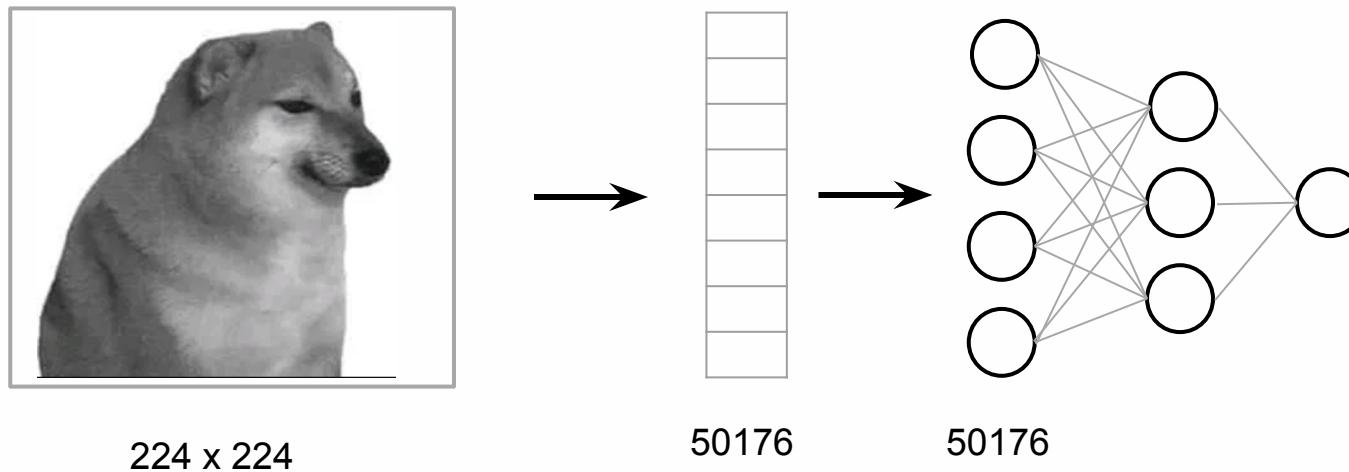
Решаем задачу с помощью полносвязной нейронной сети



Weight Sharing: мотивация

Задача: Cat vs Dog

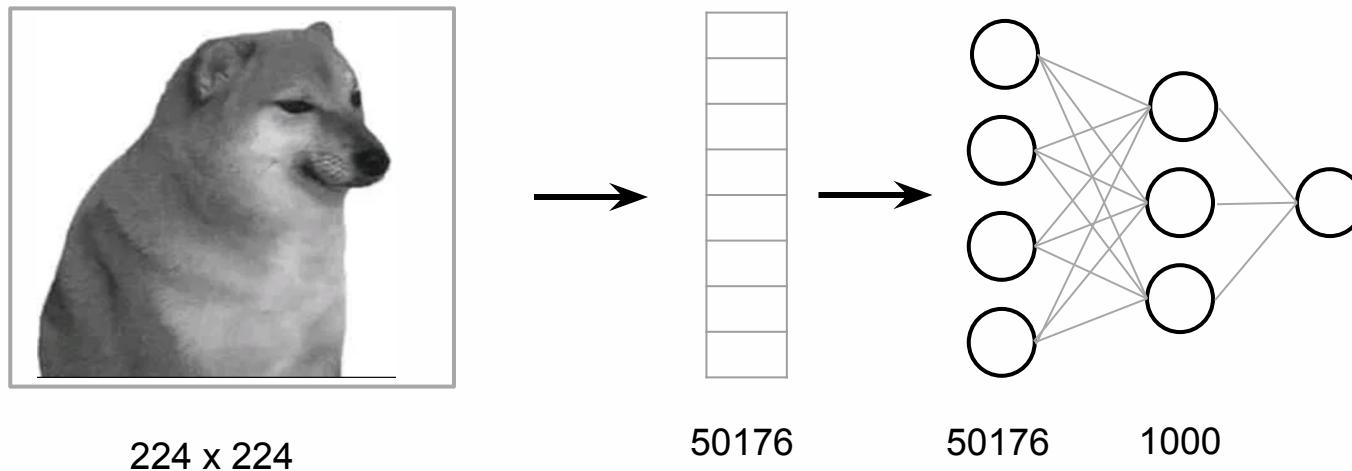
Решаем задачу с помощью полносвязной нейронной сети



Weight Sharing: мотивация

Задача: Cat vs Dog

Решаем задачу с помощью полносвязной нейронной сети

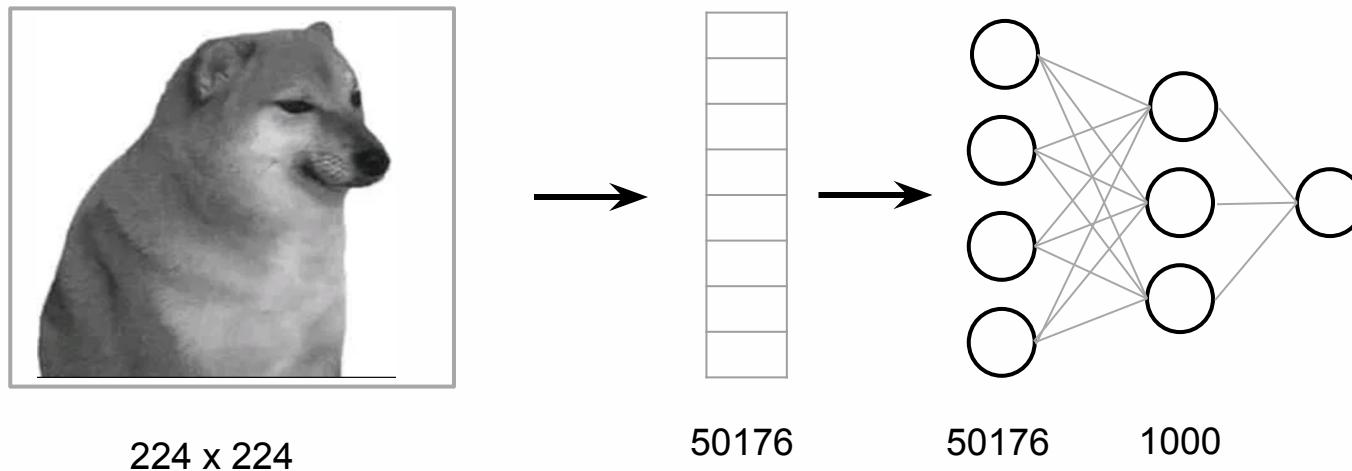


Хотим из входного вектора получить 1000 признаков после первого слоя. Сколько нужно параметров?

Weight Sharing: мотивация

Задача: Cat vs Dog

Решаем задачу с помощью полносвязной нейронной сети

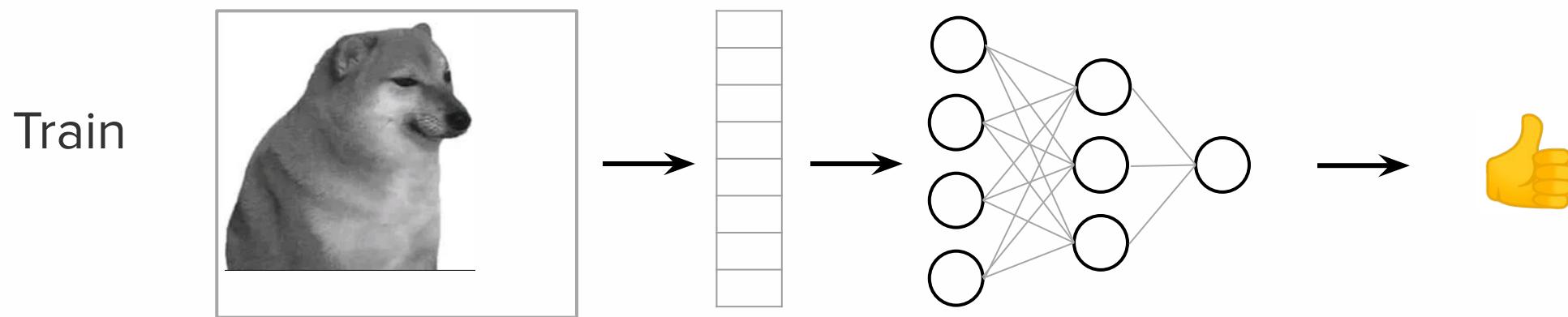


Хотим из входного вектора получить 1000 признаков после первого слоя. Сколько нужно параметров:

$$50176 \times 1000 + 1000 \sim 50\text{e}6$$

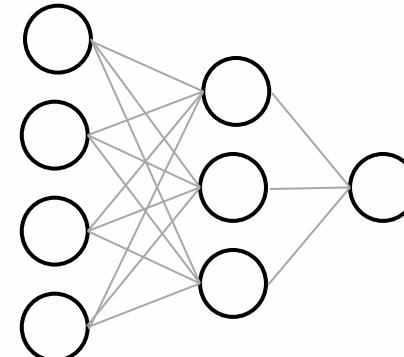
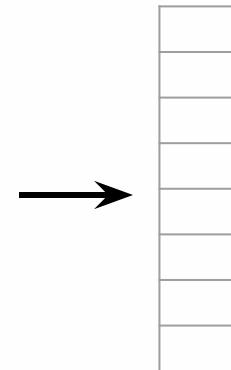
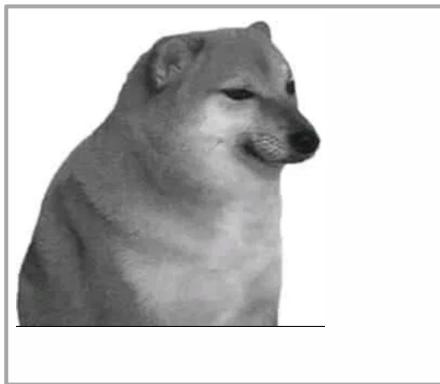
С чем можем столкнуться?

Weight Sharing: мотивация



Weight Sharing: мотивация

Train

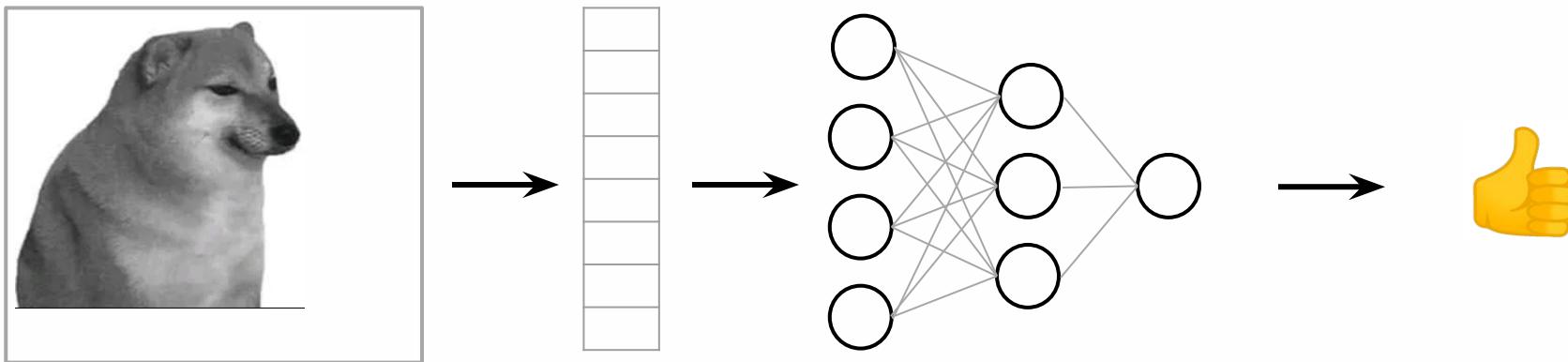


Test

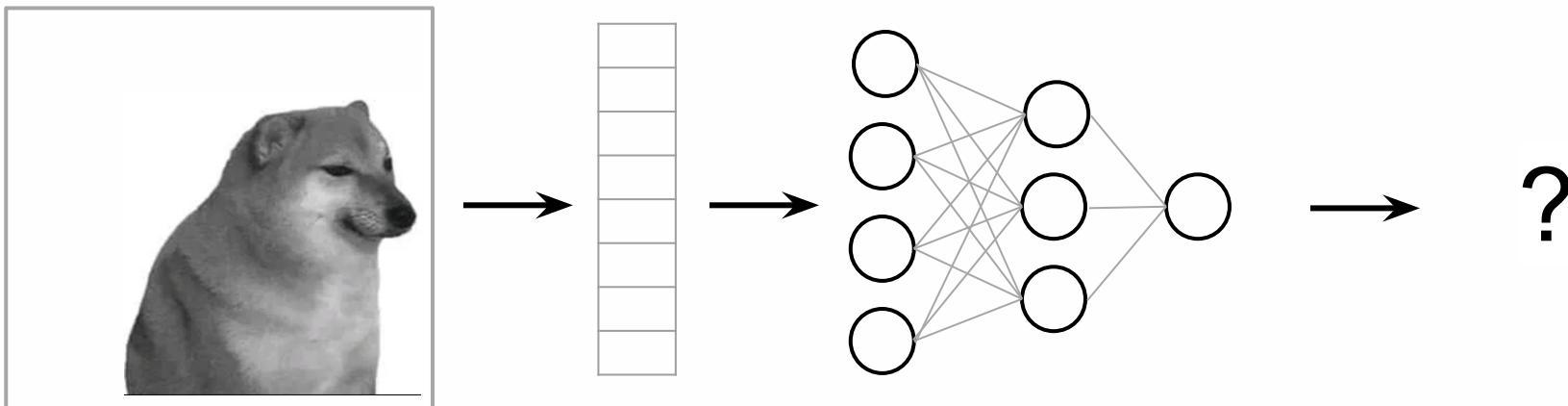


Weight Sharing: мотивация

Train

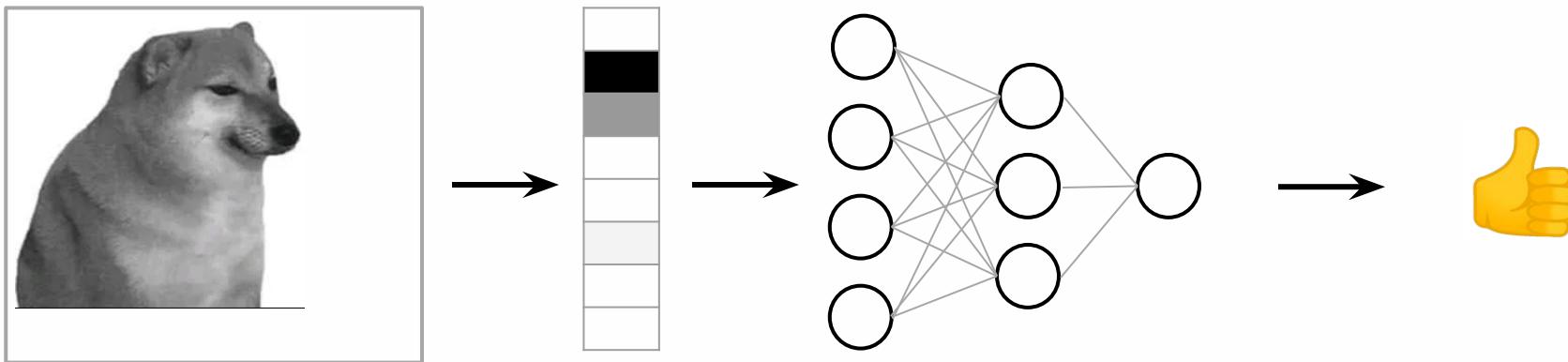


Test

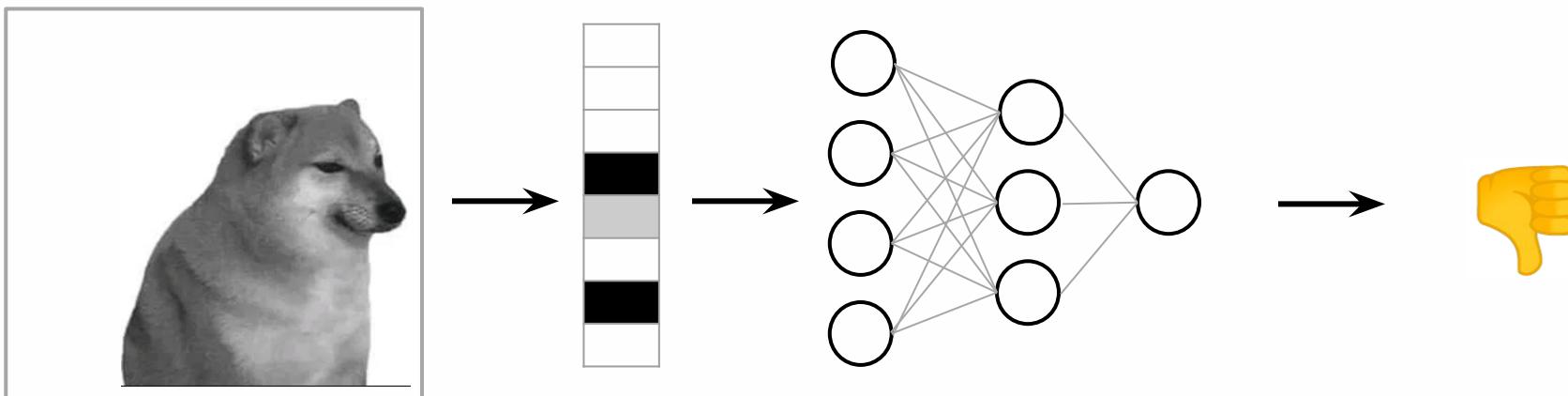


Weight Sharing: мотивация

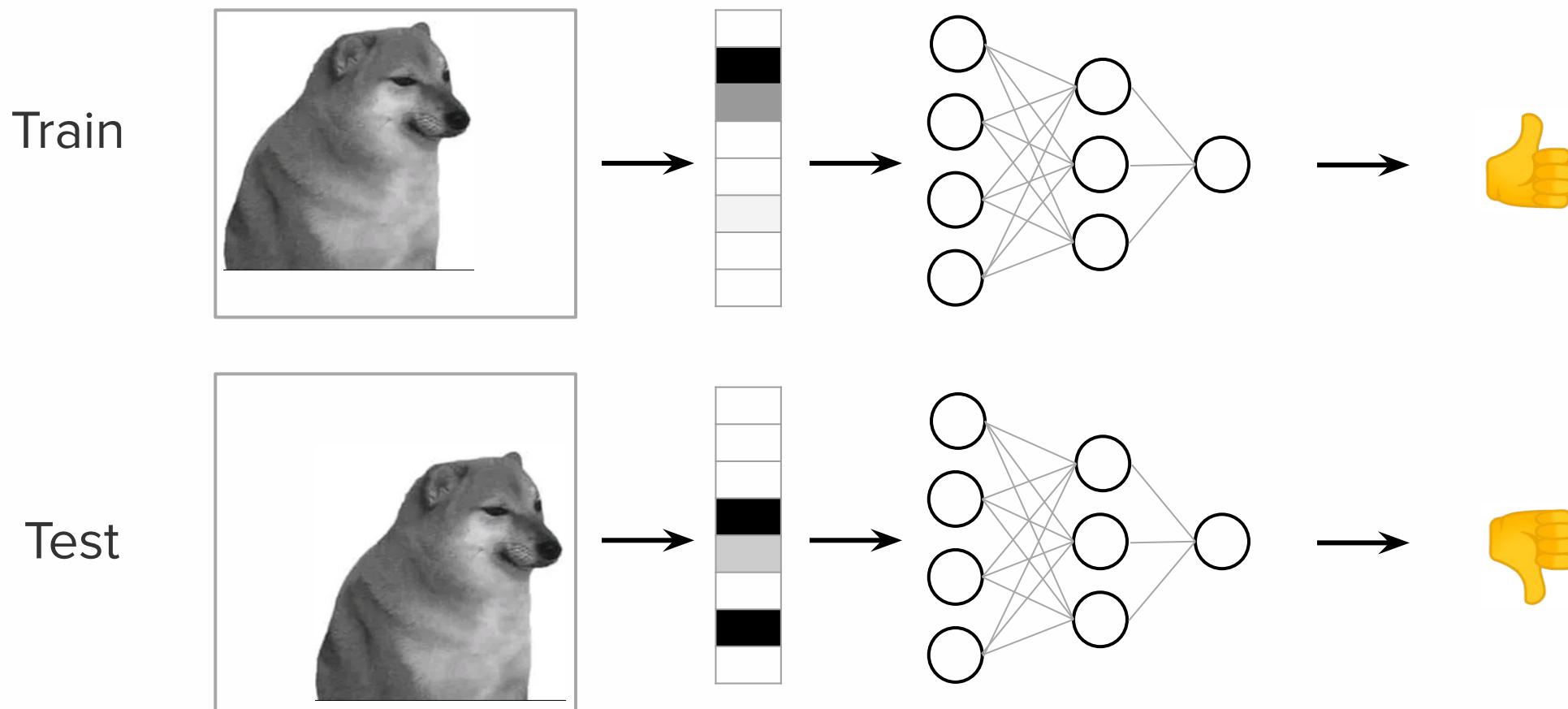
Train



Test



Weight Sharing: мотивация



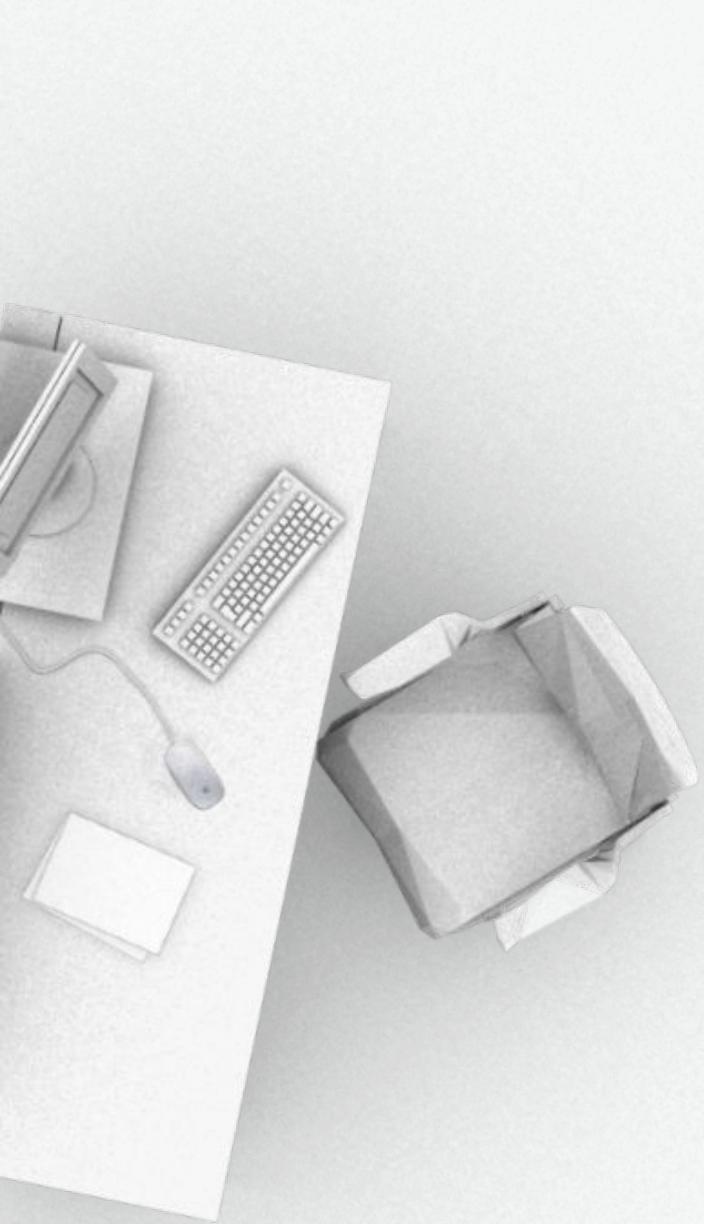
Разное положение объекта => Разный вектор признаков на входе
Модель к такому не готова. Нет трансляционной инвариантности

Weight Sharing

Замеченные проблемы:

- количество параметров полносвязной нейросети
- отсутствие трансляционной инвариантности

Обе решаются с помощью Weight/Parameter Sharing: одно и то же ядро применяется ко всем областям изображения



Pooling

Сжатие изображения



Сжатие изображения



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25



Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2×2
pool size

36	80
12	15



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25



Pooling

Pooling



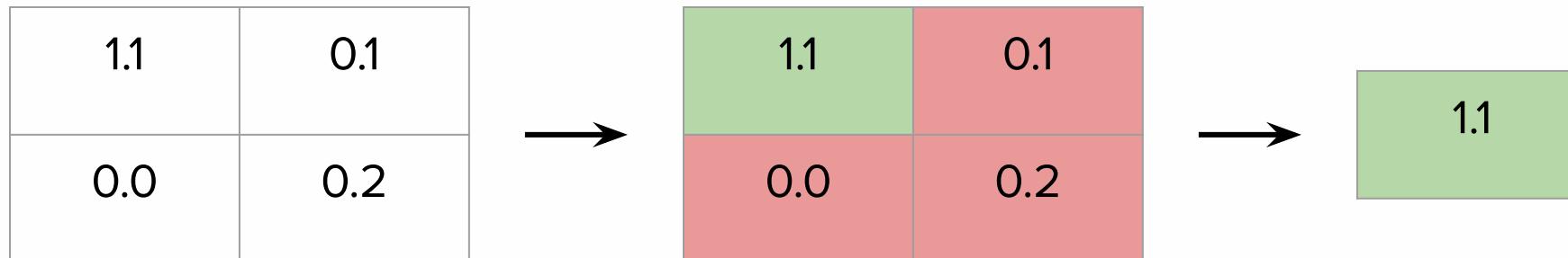
Pooling

- Требует функции агрегации. Наиболее распространенная: **max**
- downsampling: меньше размер изображения => меньше данных для обработки
- не добавляет обучаемых параметров!

Pooling

- Требует функции агрегации. Наиболее распространенная: **max**
- downsampling: меньше размер изображения => меньше данных для обработки
- не добавляет обучаемых параметров!

- мы видели, что свертка позволяет находить паттерны
- больше значение активации => больше соответствие фрагмента тензора и ядра свертки
- pooling позволяет “прокинуть” информацию о найденном шаблоне в следующие слои





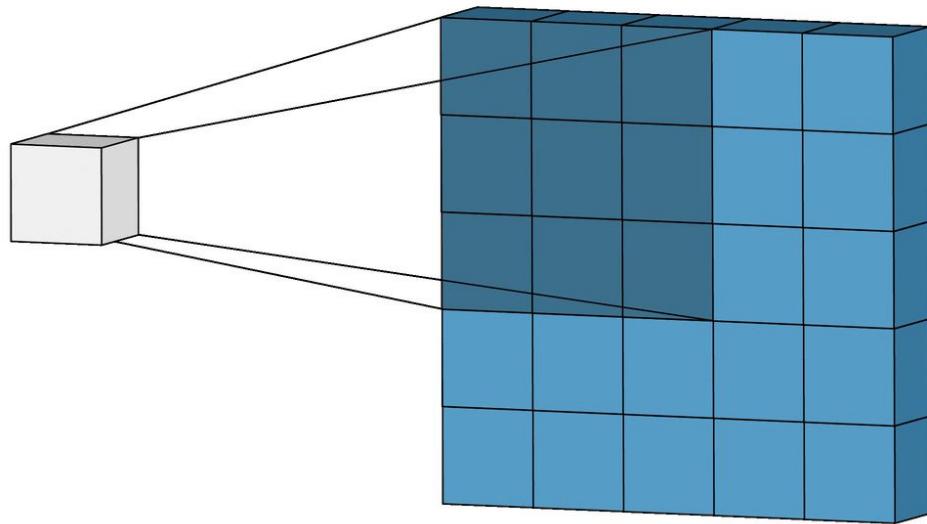
Convolution: Keras

Convolution Layer

```
tf.keras.layers.Conv2D(  
    filters, kernel_size, strides=(1, 1), padding='valid', data_format=None,  
    dilation_rate=(1, 1), groups=1, activation=None, use_bias=True,  
    kernel_initializer='glorot_uniform', bias_initializer='zeros',  
    kernel_regularizer=None, bias_regularizer=None, activity_regularizer=None,  
    kernel_constraint=None, bias_constraint=None, **kwargs  
)
```

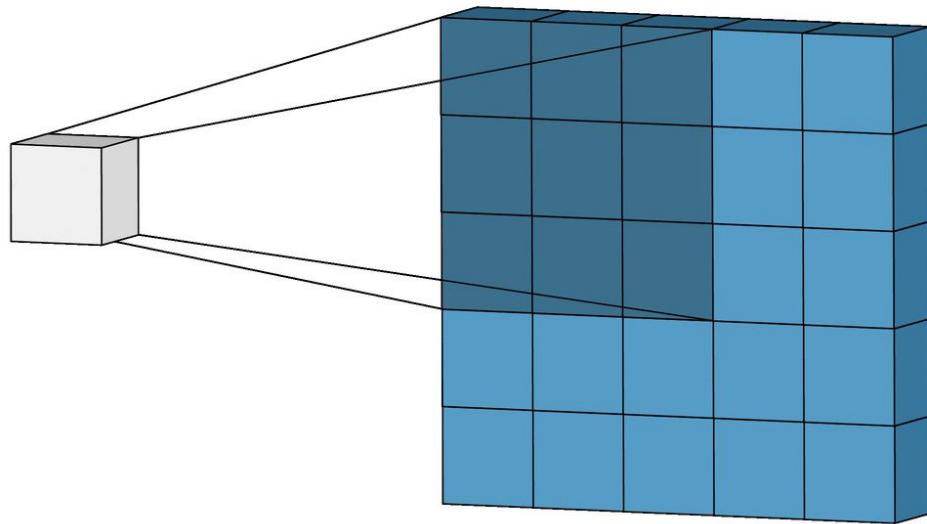
- **filters** — размер банка фильтров
- **kernel_size** — пространственный размер ядра (3x3, 5x5, ...)
- **data_format** — по умолчанию batch_size x height x width x channels (32, 224, 224, 3)
- **activation** — sigmoid, tanh, relu, ...

Convolution Layer: Padding

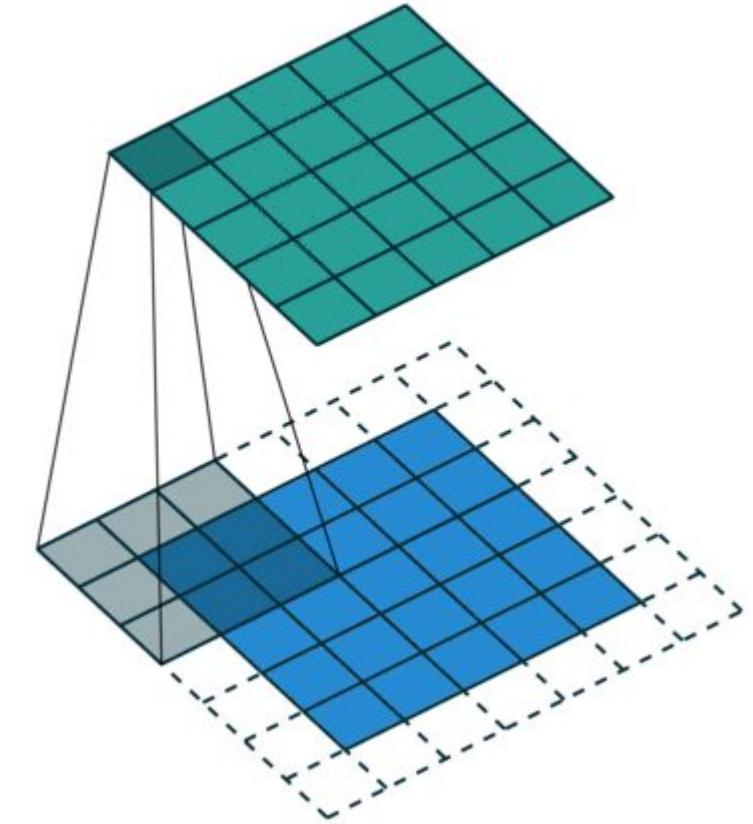


padding='valid'

Convolution Layer: Padding

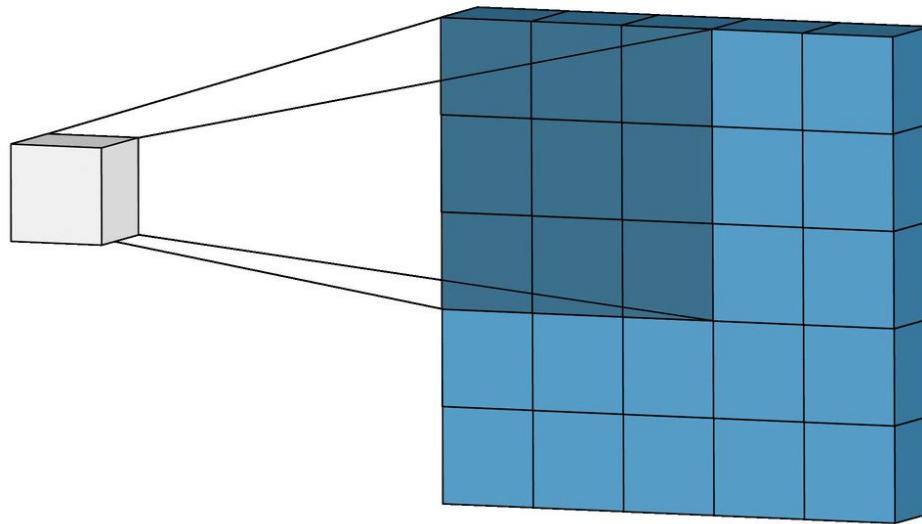


padding='valid'



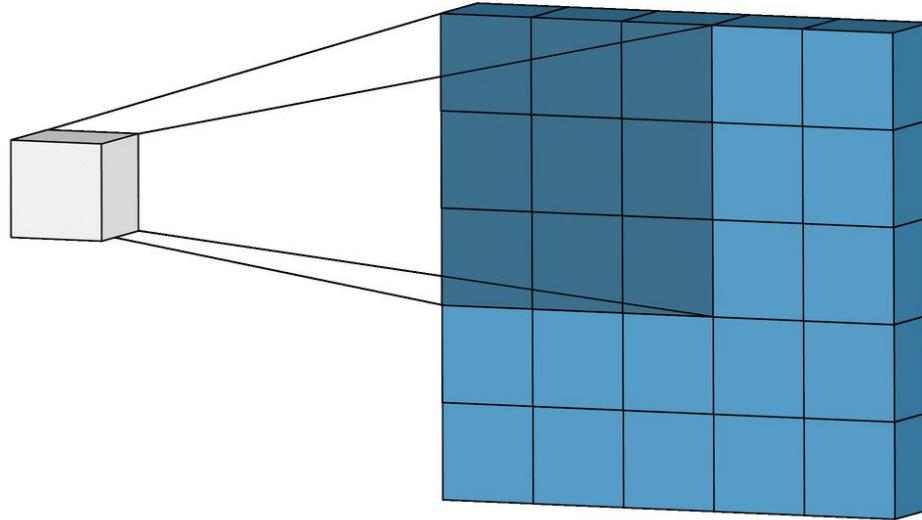
padding='same'

Convolution Layer: Stride

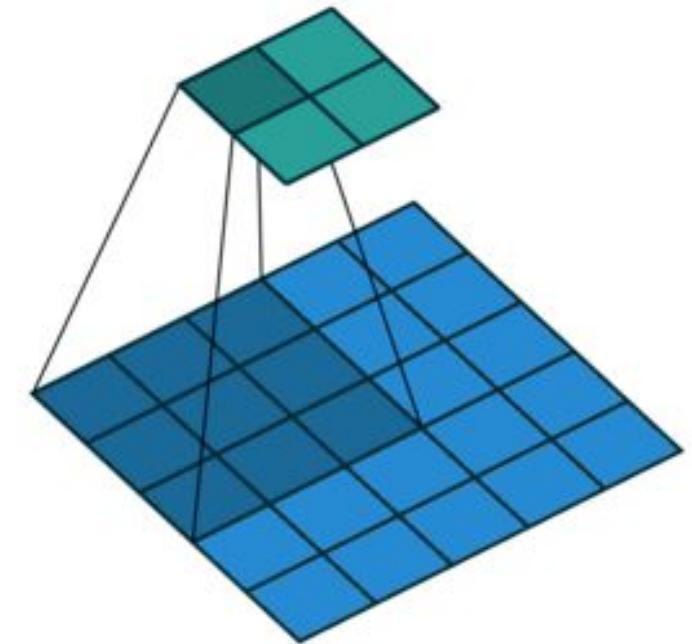


`strides=(1, 1)`

Convolution Layer: Stride



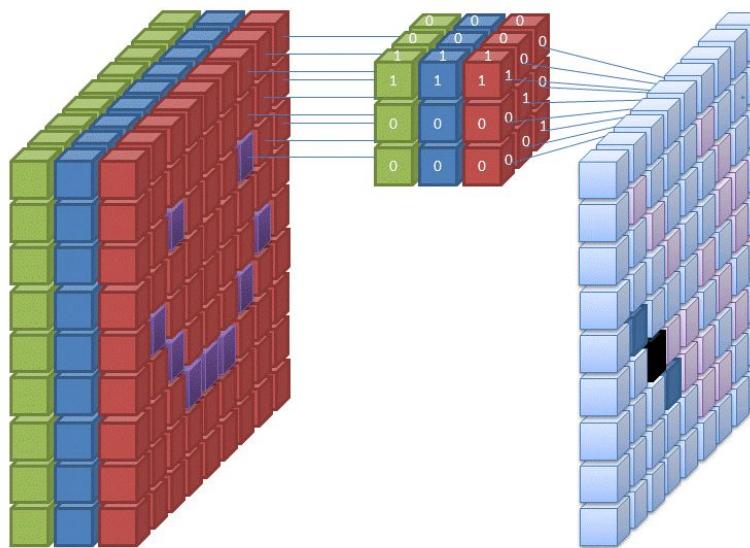
strides=(1, 1)



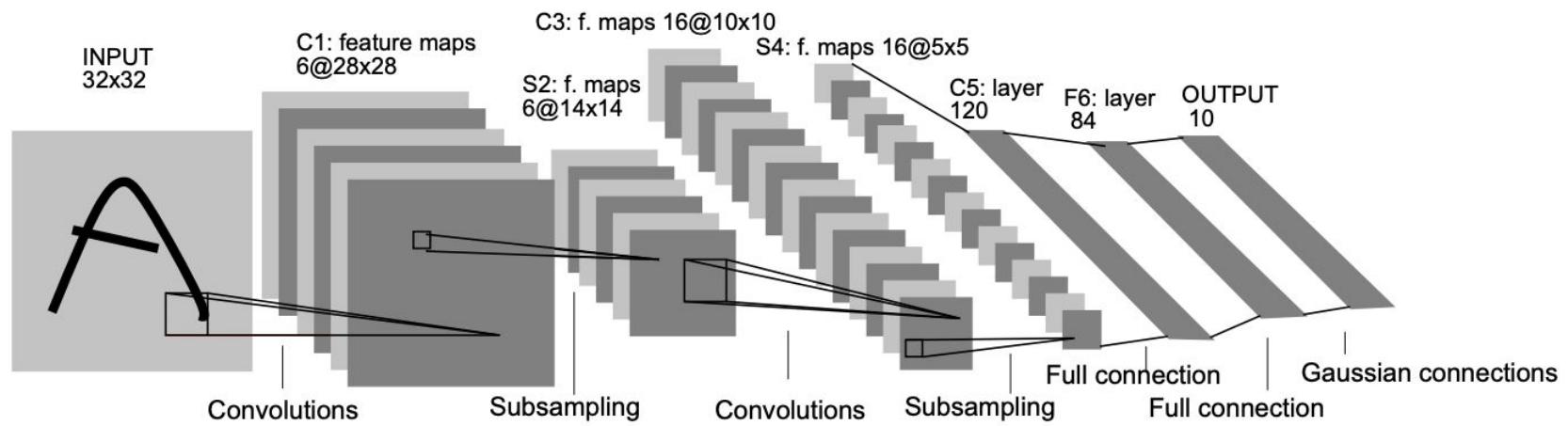
strides=(2, 2)

Свертки над многоканальными изображениями

$$(A * w)_{ij} = \sum_{k=0}^2 \sum_{i'=0}^{K-1} \sum_{j'=0}^{K-1} A(i + i', j + j', k) w(i', j', k)$$



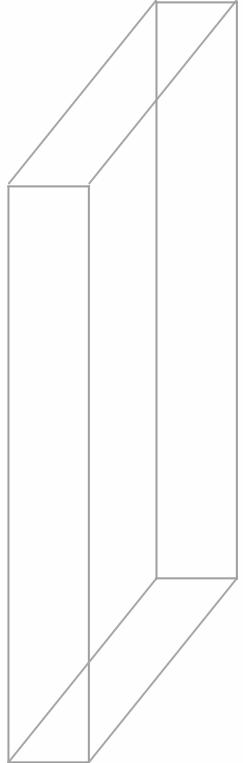
Пример CNN: LeNet-5 (1998)





Convolution: Receptive Field

Receptive Field



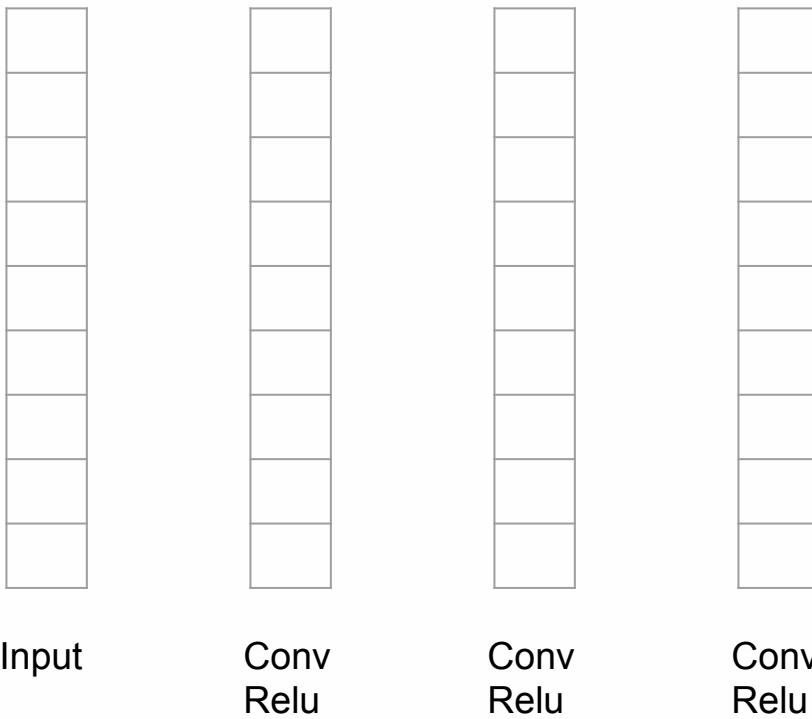
Input

Receptive Field

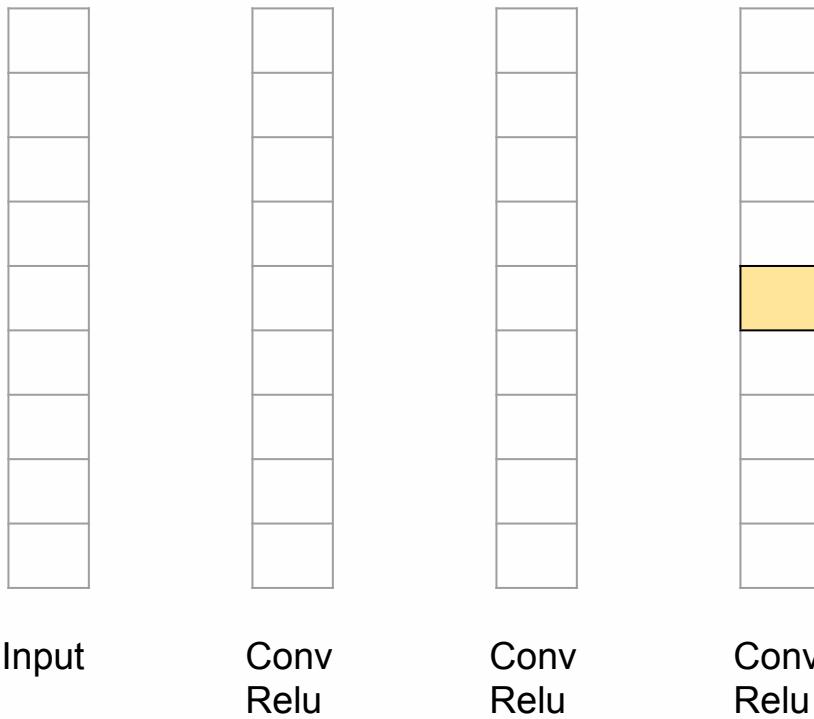


Input

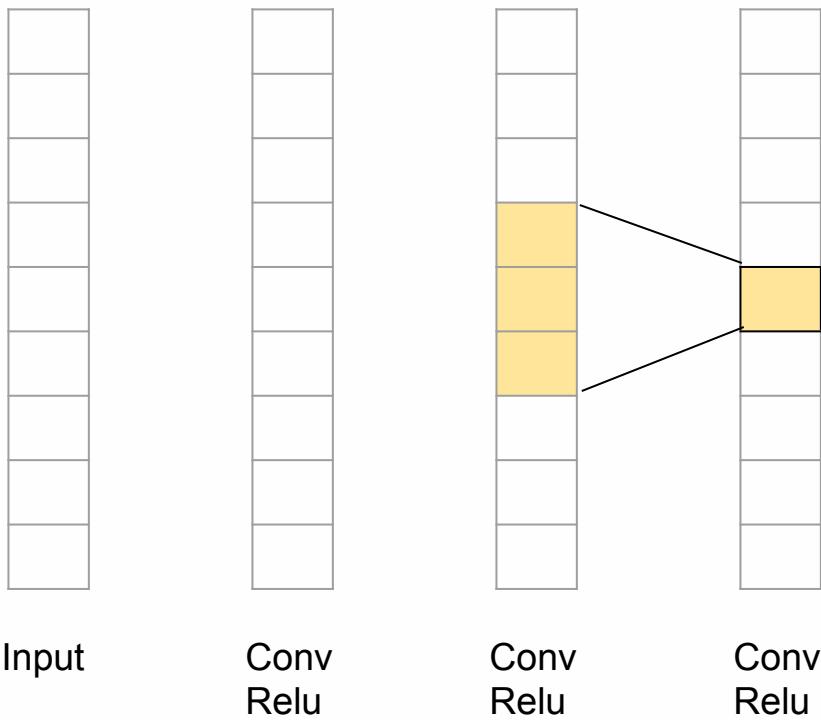
Receptive Field



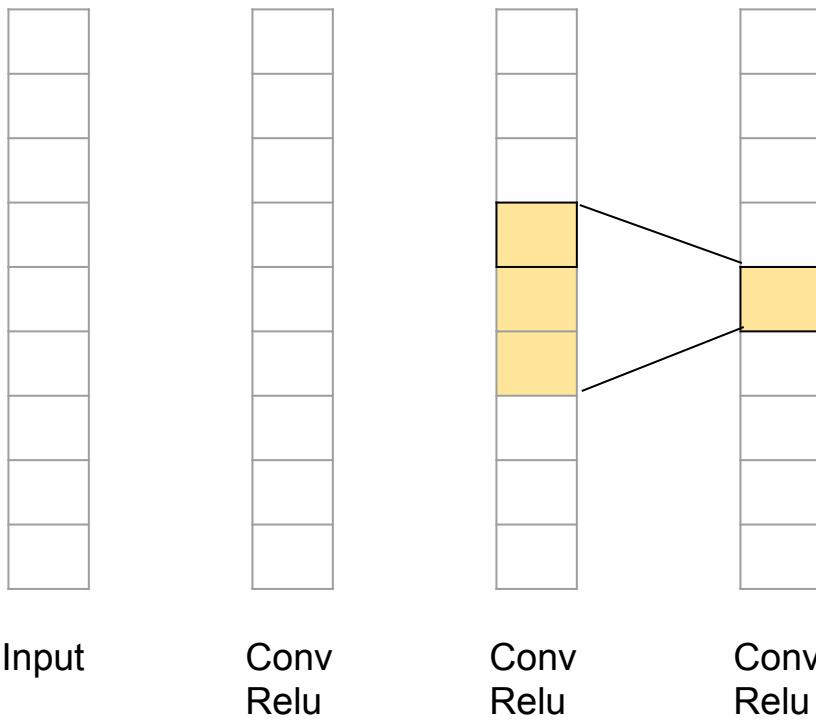
Receptive Field



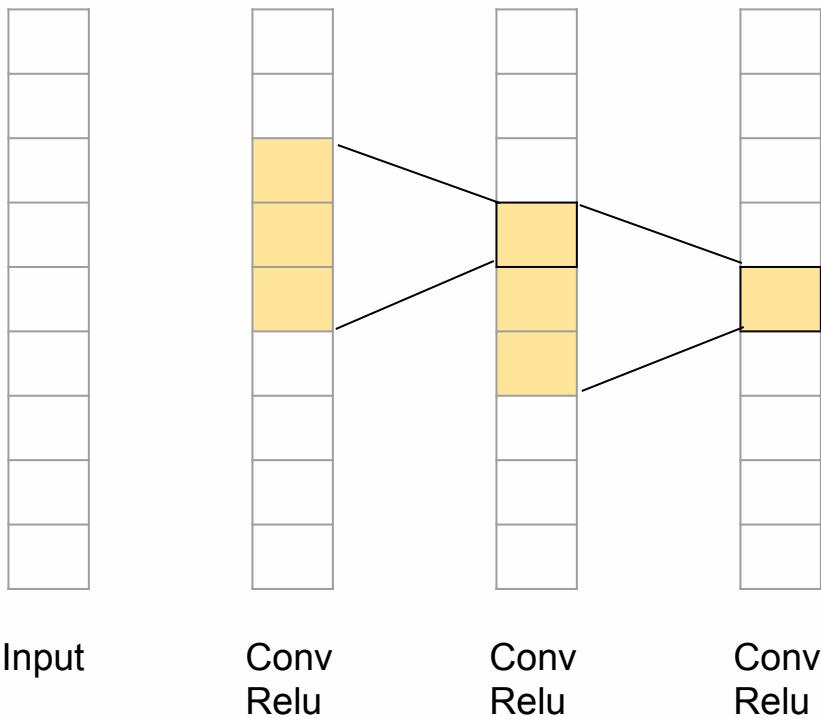
Receptive Field



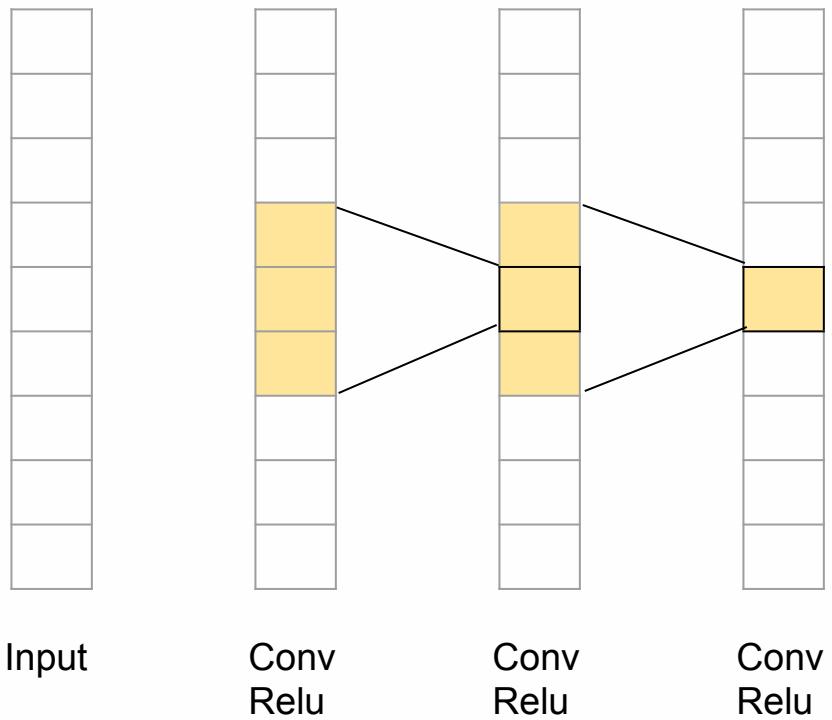
Receptive Field



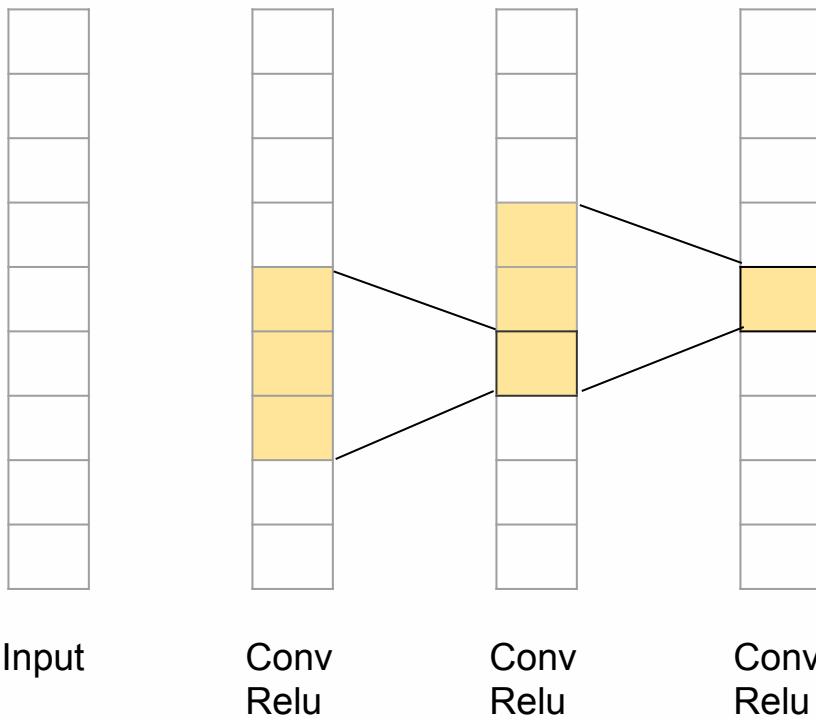
Receptive Field



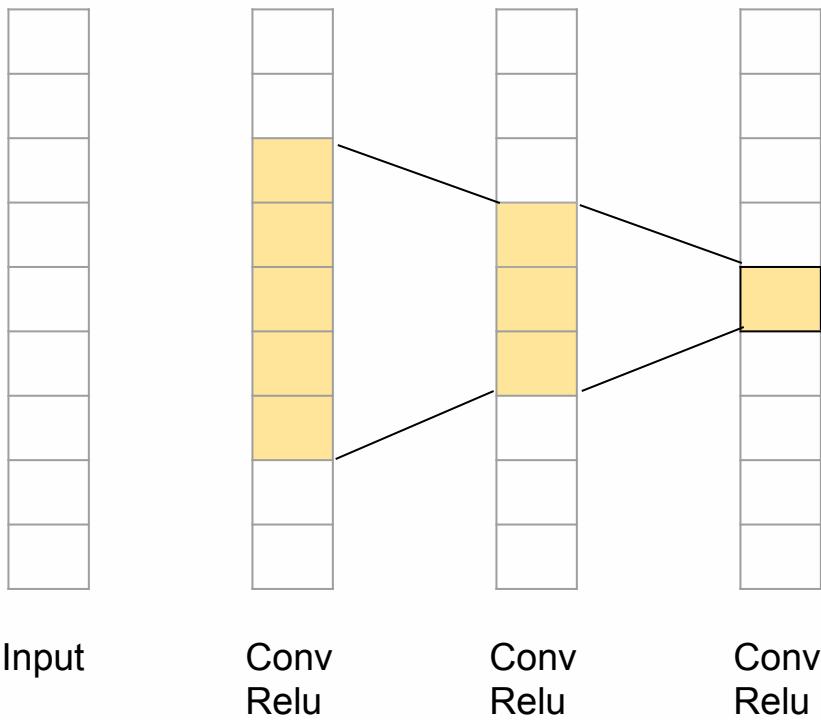
Receptive Field



Receptive Field

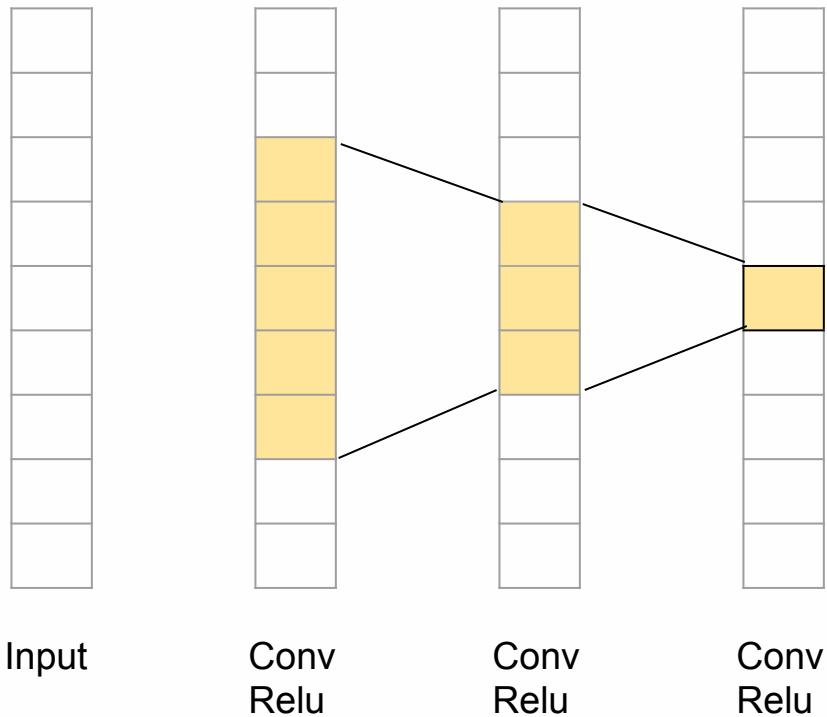


Receptive Field

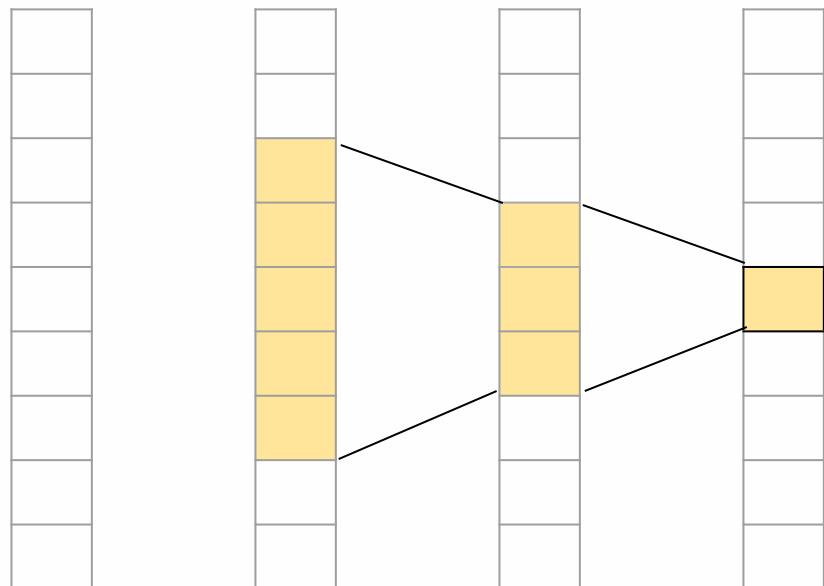


Receptive Field

У двух последовательных сверток с ядром 3x3 receptive область такая, как у одной свертки с ядром 5x5



Receptive Field



Input

Conv
Relu

Conv
Relu

Conv
Relu

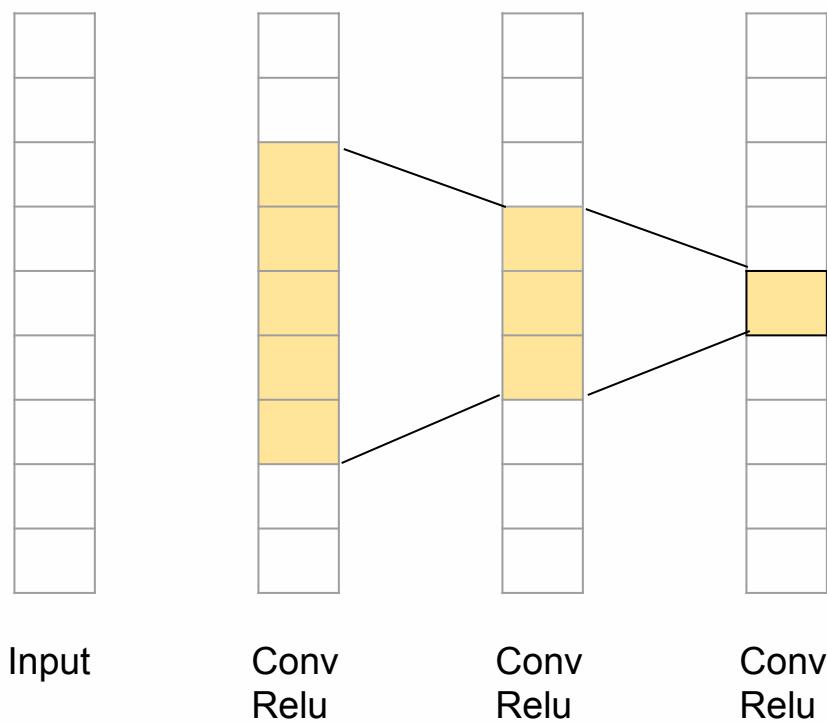
У двух последовательных сверток с ядром 3x3 рецептивная область такая, как у одной свертки с ядром 5x5

1) кол-во параметров

Кол-во параметров у свертки 5x5:
 $K \times 5 \times 5 \times K = 25K^2$

Кол-во параметров у сверток 3x3:
 $2 \times (K \times 3 \times 3 \times K) = 18K^2$

Receptive Field



У двух последовательных сверток с ядром 3x3 рецептивная область такая, как у одной свертки с ядром 5x5

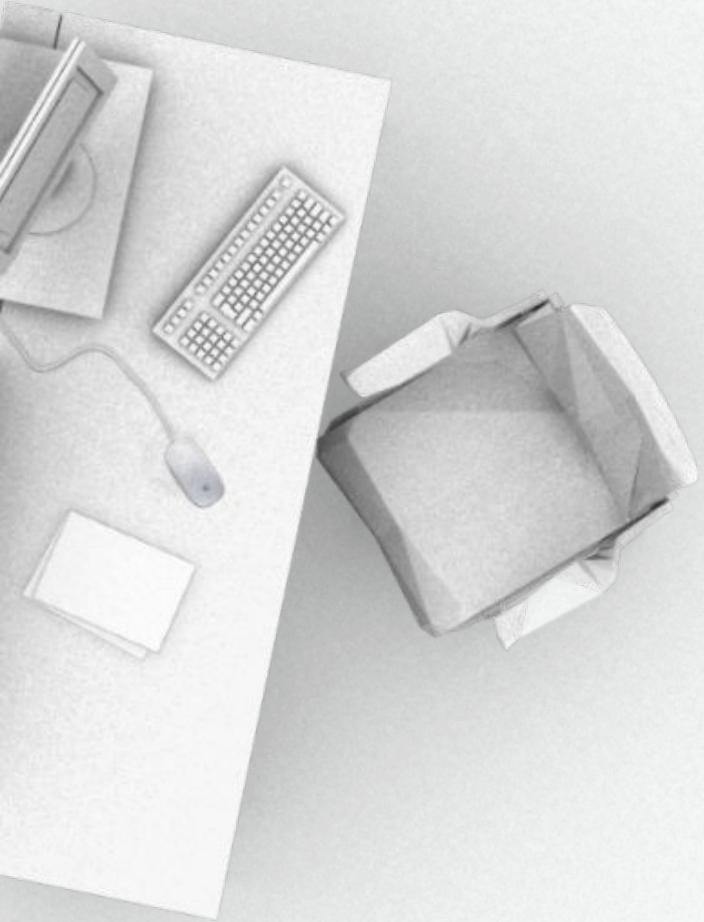
1) кол-во параметров

Кол-во параметров у свертки 5x5:
 $K \times 5 \times 5 \times K = 25K^2$

Кол-во параметров у сверток 3x3:
 $2 \times (K \times 3 \times 3 \times K) = 18K^2$

2) выразительность признаков

между последовательными свертками 3x3 присутствуют нелинейности => более выразительные признаки



Overfitting

Методы борьбы с переобучением



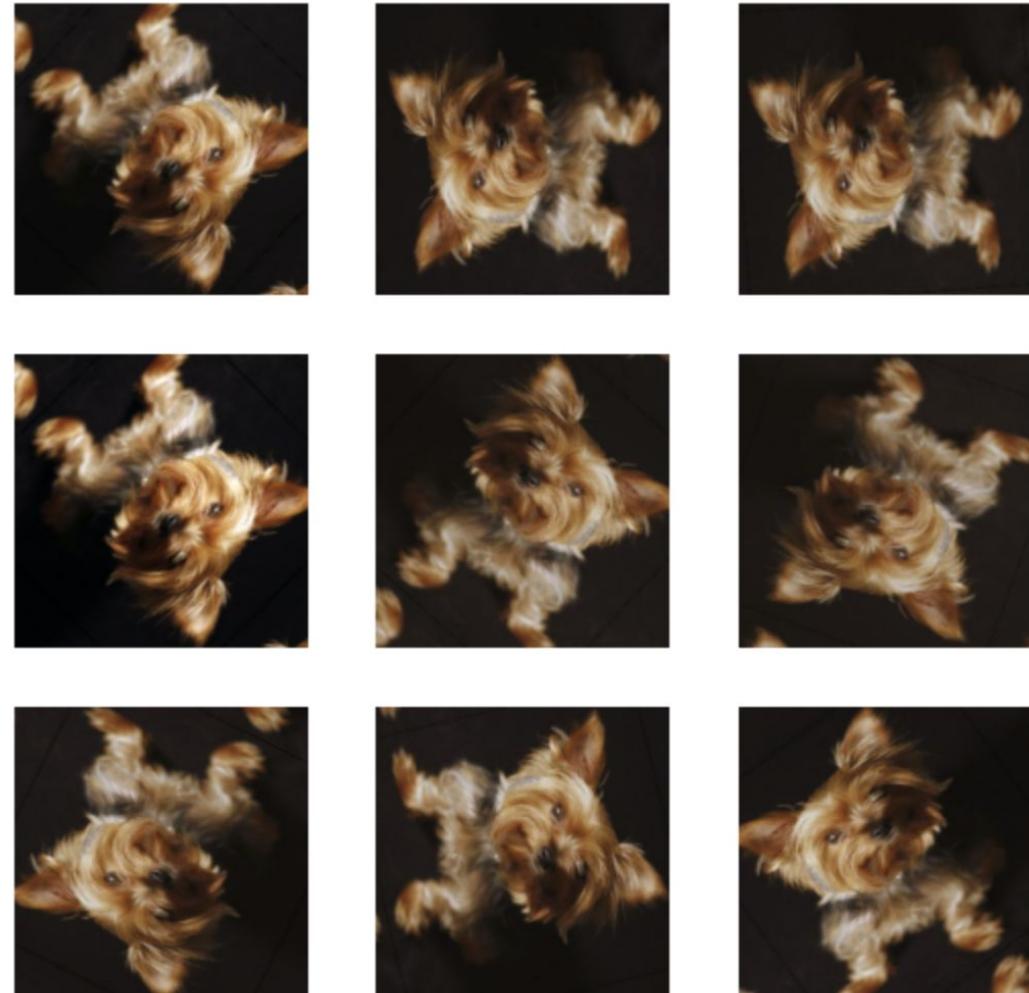
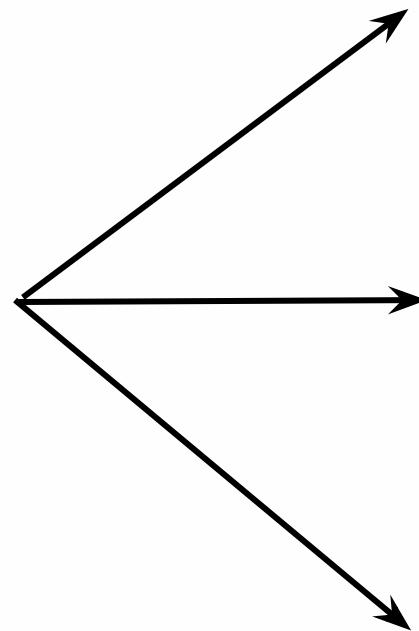
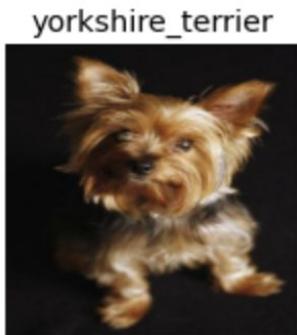
Методы борьбы с переобучением

- уменьшение количества параметров модели
- методы регуляризации из классического ML: L1 / L2
- специфичные для NN методы регуляризации
 - DropOut
 - DropConnect
 - ...

Методы борьбы с переобучением

- уменьшение количества параметров модели
- методы регуляризации из классического ML: L1 / L2
- специфичные для NN методы регуляризации
 - DropOut
 - DropConnect
 - ...
- сбор дополнительных данных для обучения
- аугментации

Аугментации



#0123

Аугментации

- повороты
- зеркальные отражения
- яркость
- сдвиги
- кропы
- контрастность
- ...

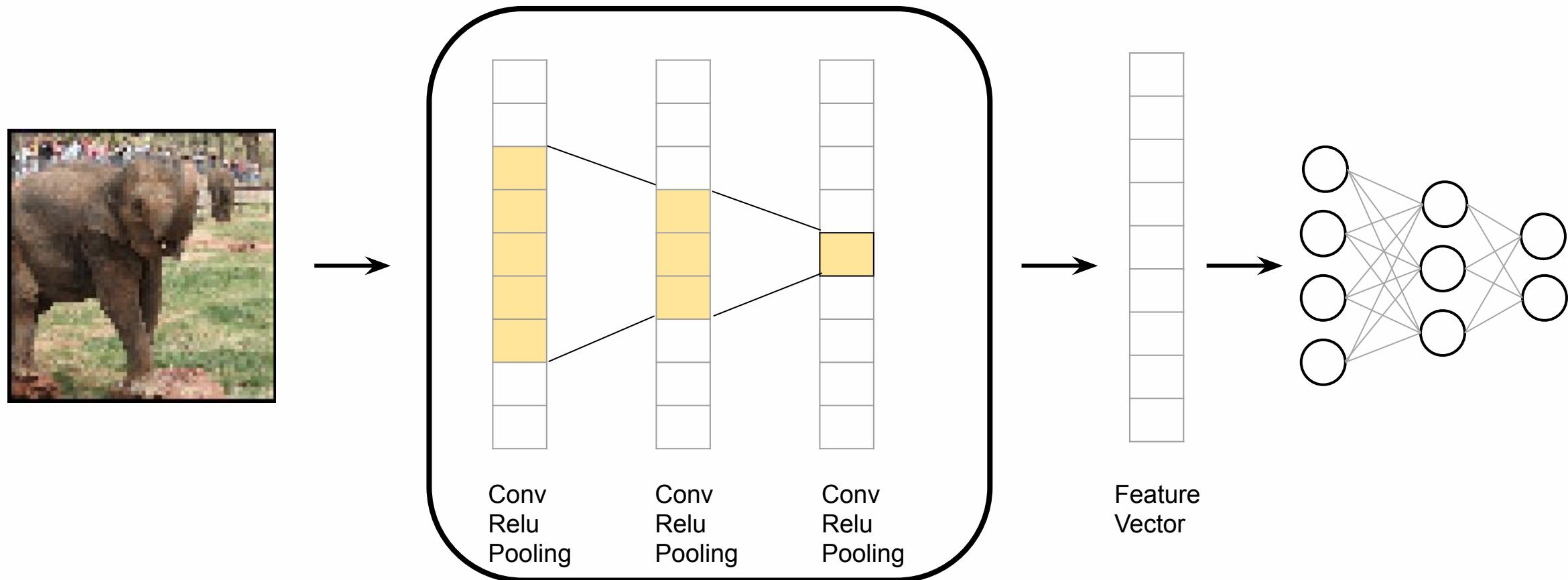
<https://synthesis.ai/2020/11/11/driving-model-performance-with-synthetic-data-i-augmentation-s-in-computer-vision/>

<https://synthesis.ai/2020/12/02/driving-model-performance-with-synthetic-data-ii-smart-augmentations/>

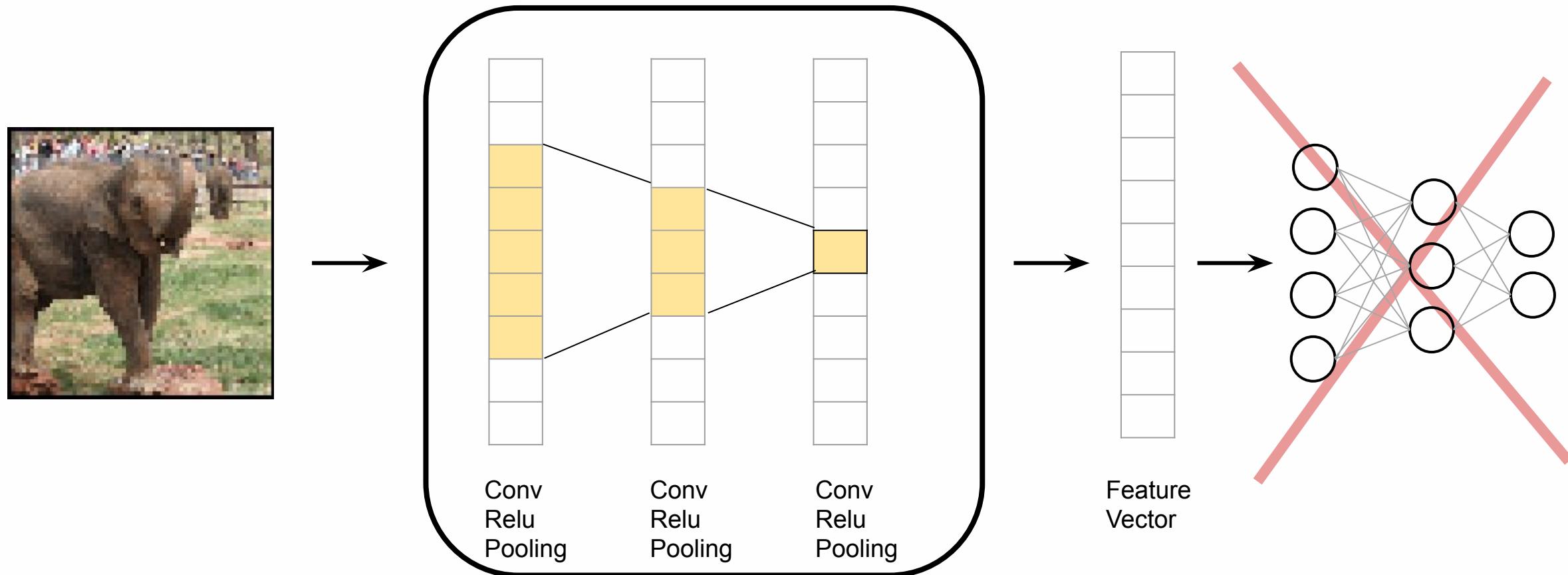
A grayscale illustration of a computer setup on a desk. It includes a monitor, a keyboard, a mouse, and a large stack of papers. The monitor is on the left, the keyboard is below it, and the mouse is to the right. A large, crumpled piece of paper is positioned in front of the keyboard.

Transfer Learning

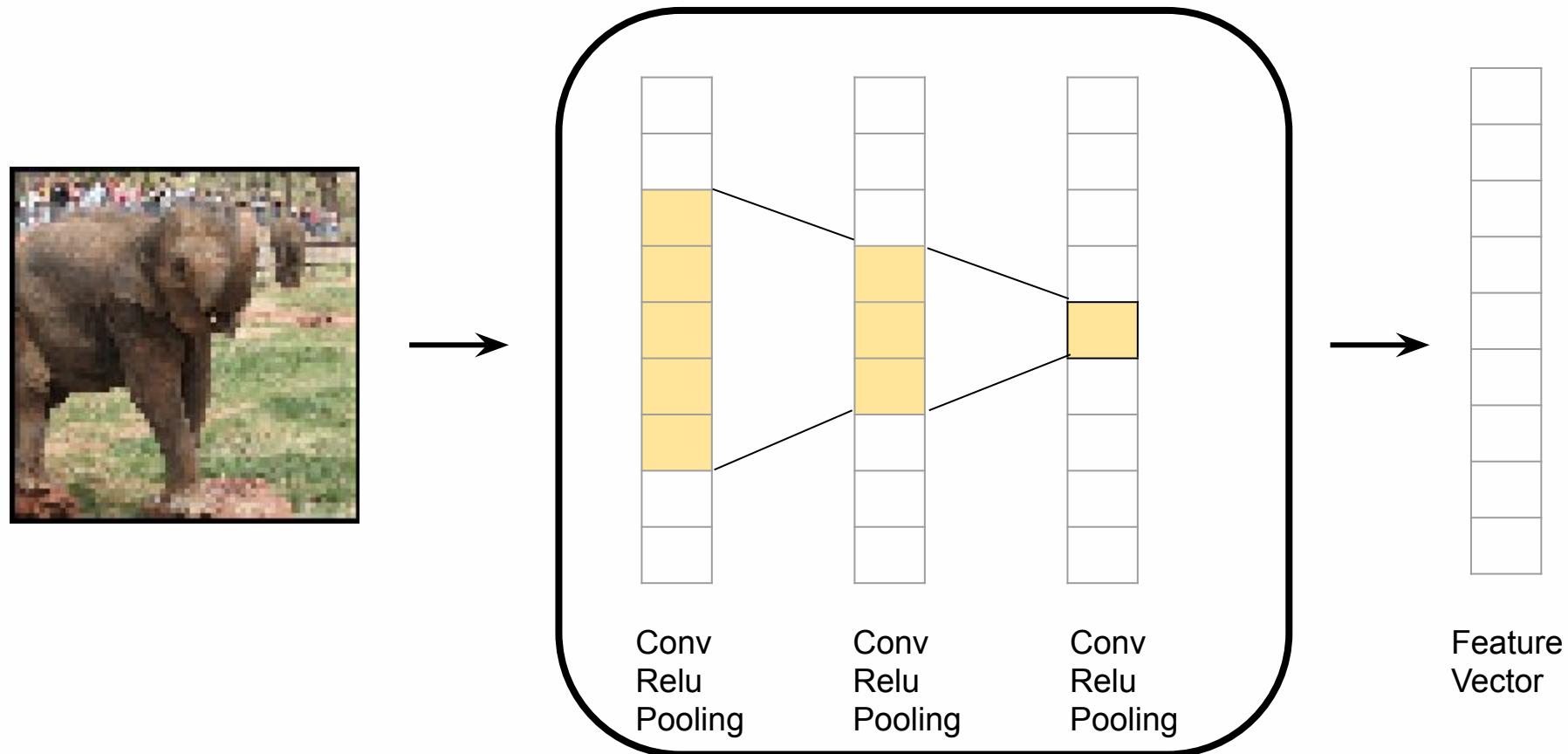
CNN Architecture



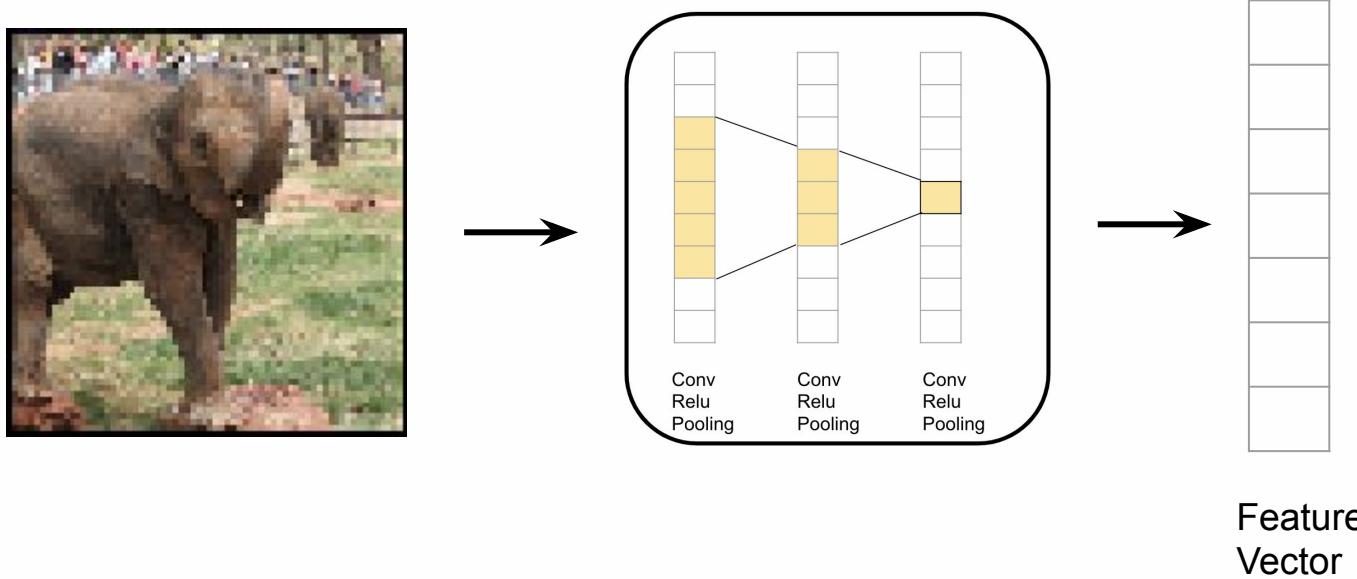
CNN Architecture: Image Embedding



CNN Architecture: Image Embedding

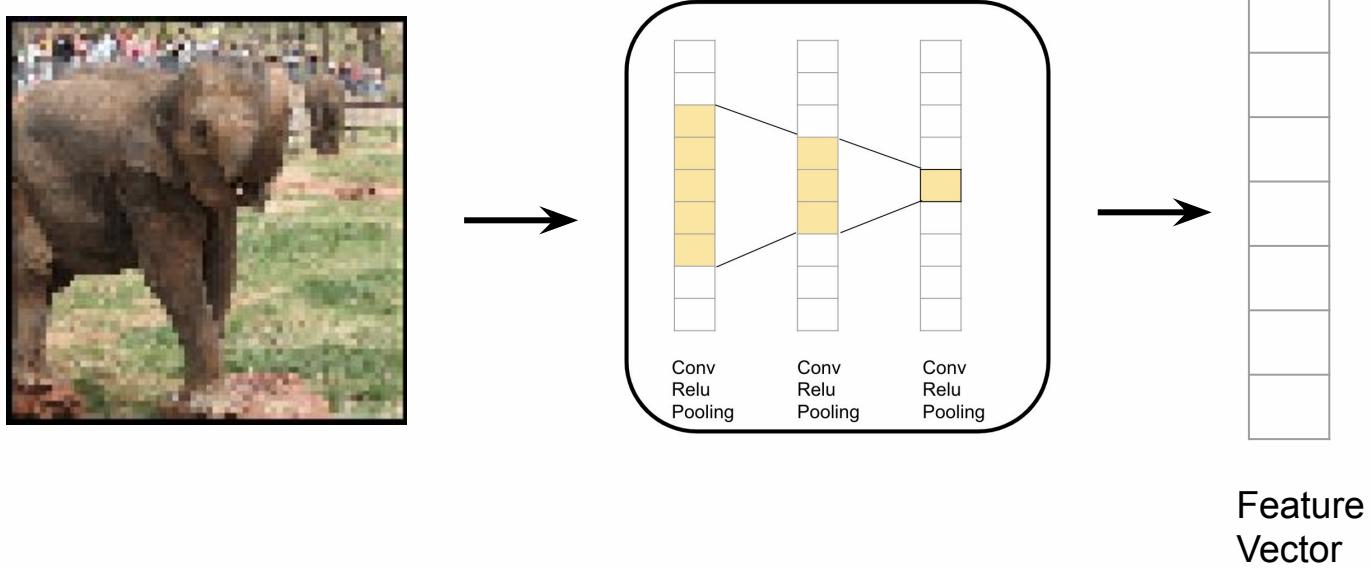


CNN Architecture: Image Embedding



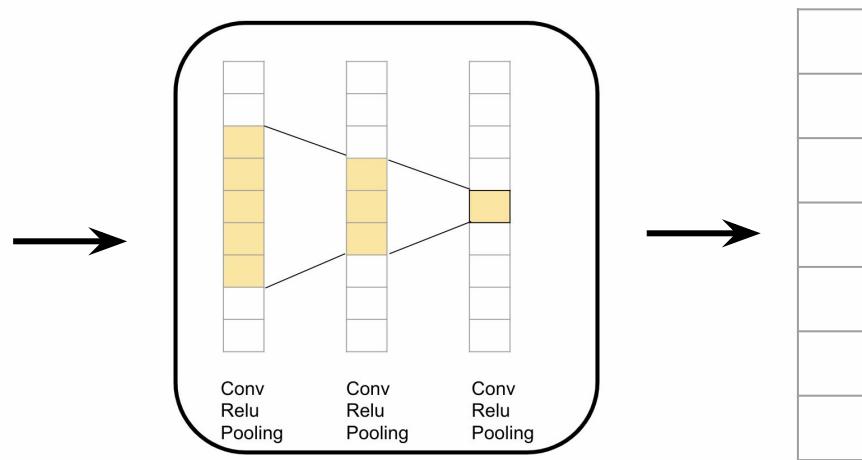
#0129

CNN Architecture: Image Embedding



Для интерпретации полученного после CNN-слоев вектора можно искать ближайших соседей с помощью L2-нормы / cosine similarity

CNN Architecture: Image Embedding



Для интерпретации полученного после CNN-слоев вектора можно искать ближайших соседей с помощью L2-нормы / cosine similarity

Ближайшие соседи:



#0131

CNN Architecture: Image Embedding

Query:



#0132

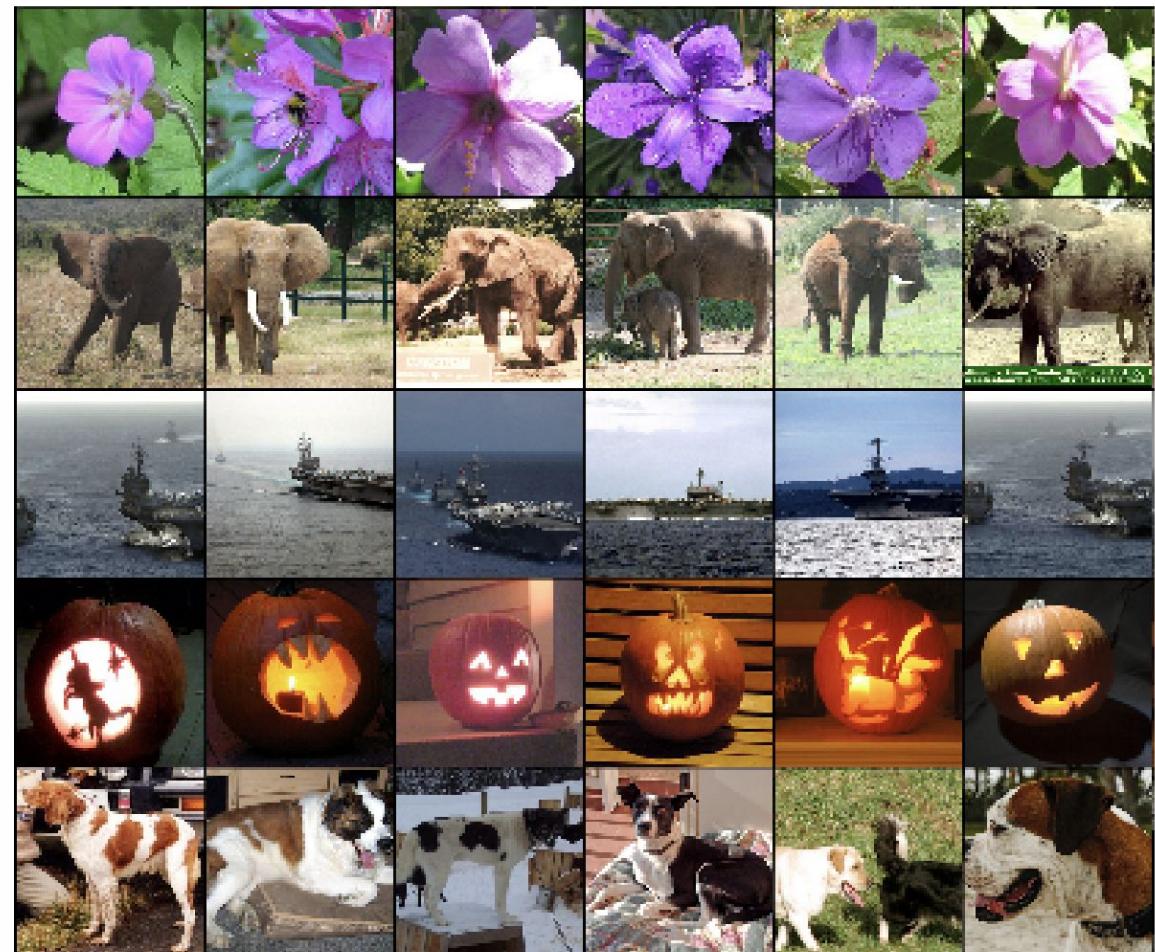
<http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf>

CNN Architecture: Image Embedding

Query:



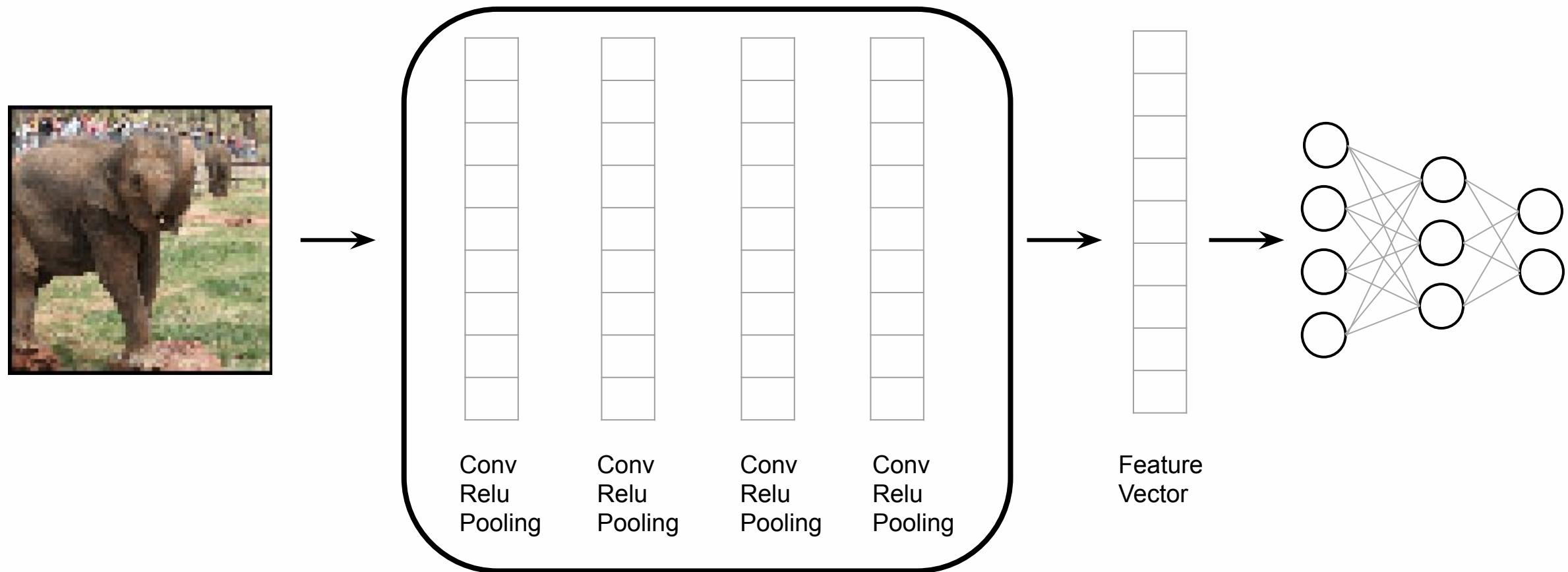
Neighbors:



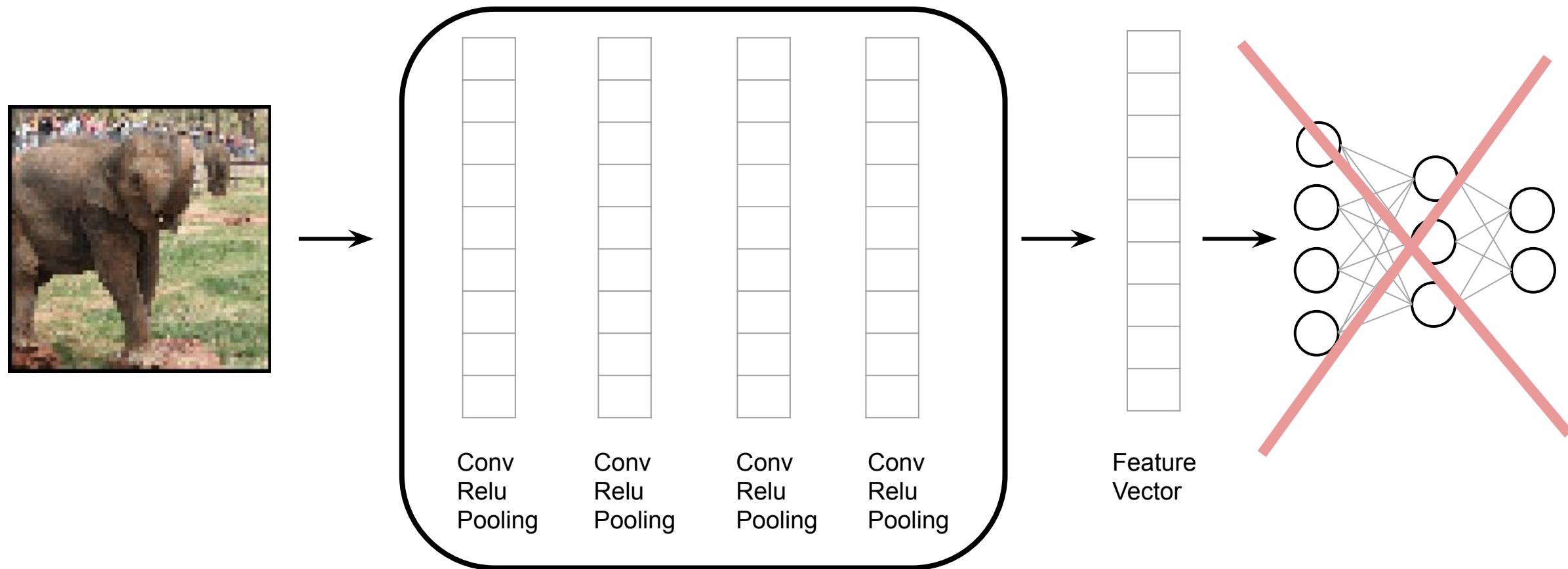
#0133

<http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf>

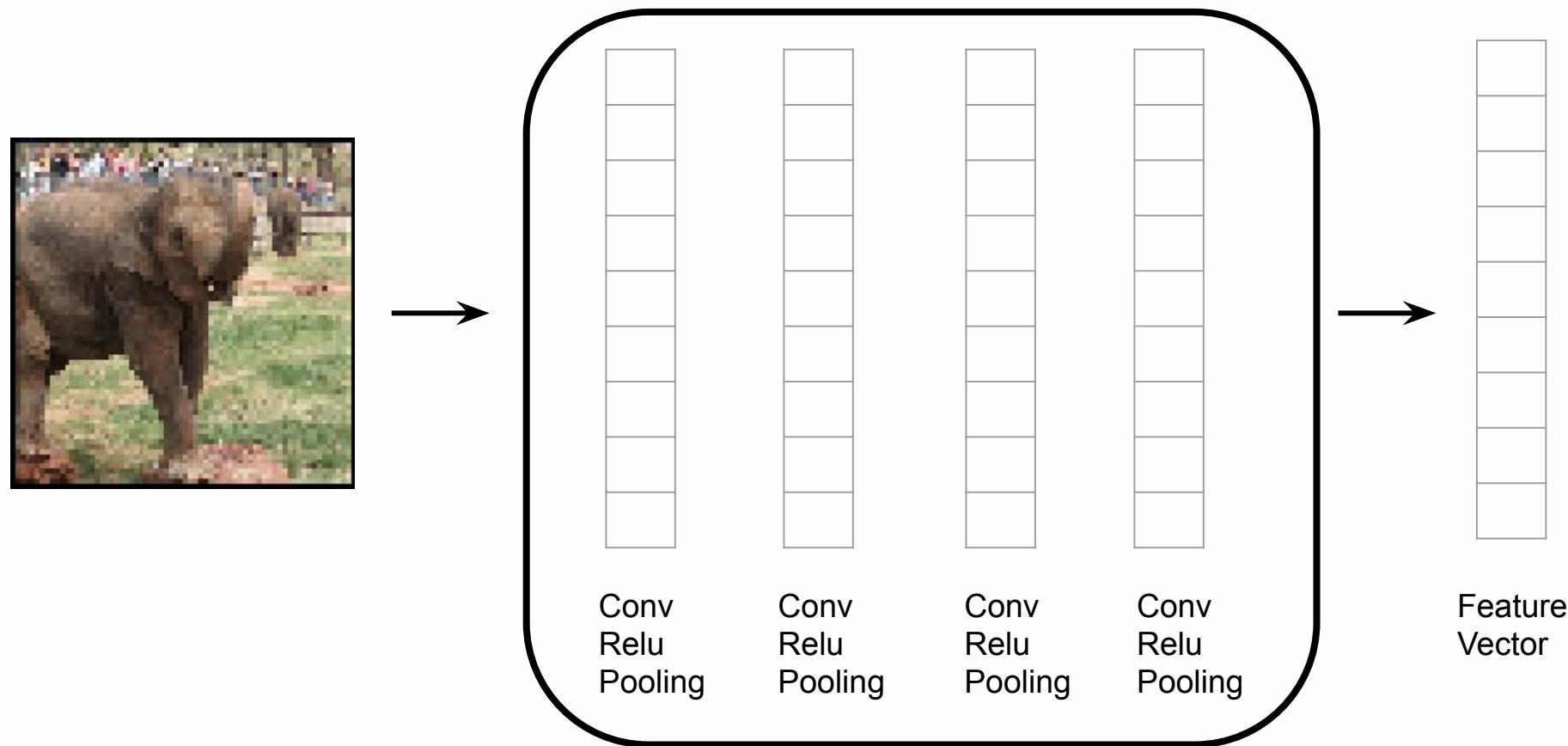
CNN Architecture: Feature Extraction



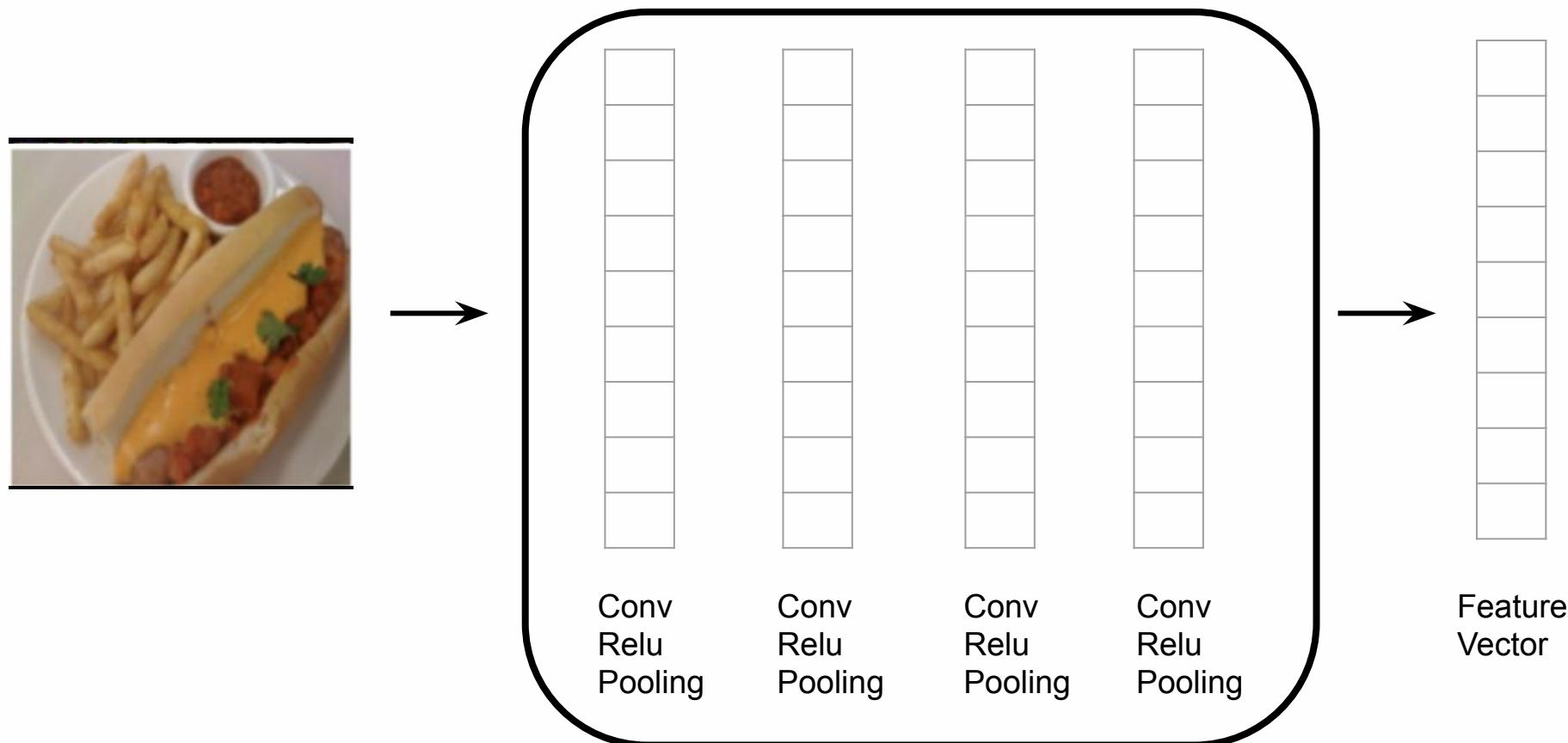
CNN Architecture: Feature Extraction



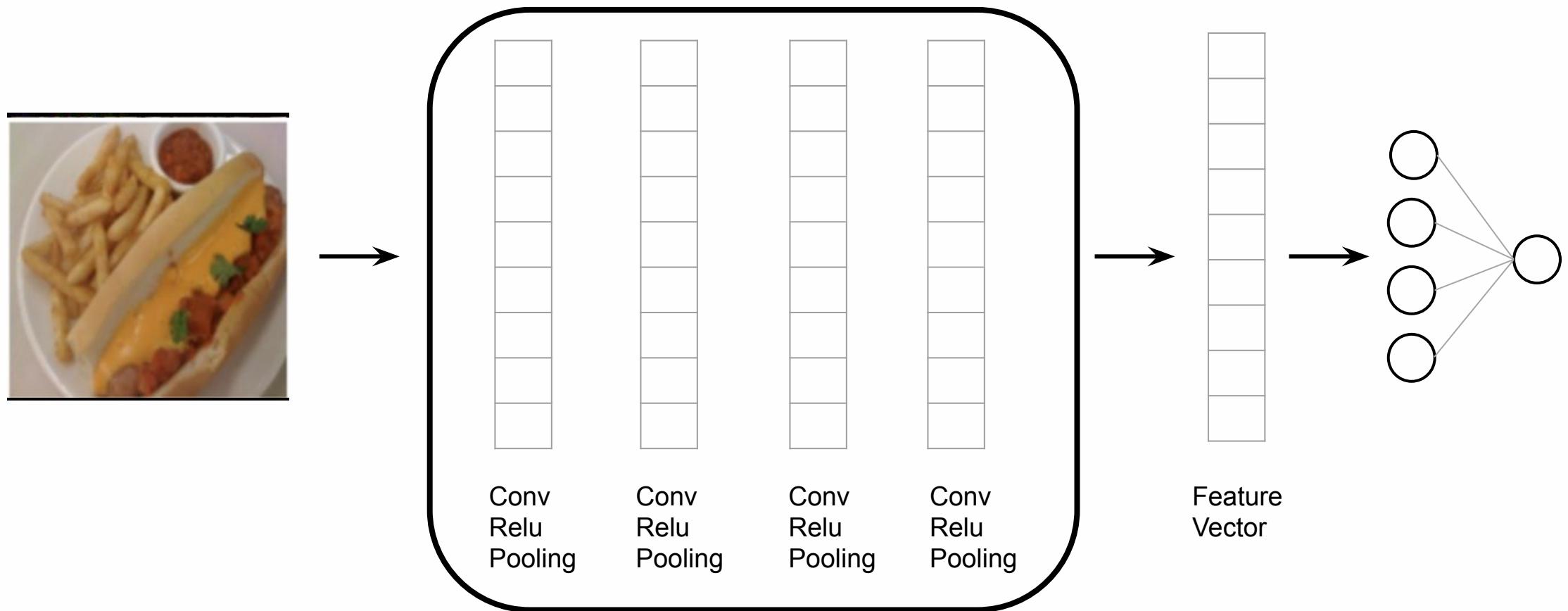
CNN Architecture: Feature Extraction



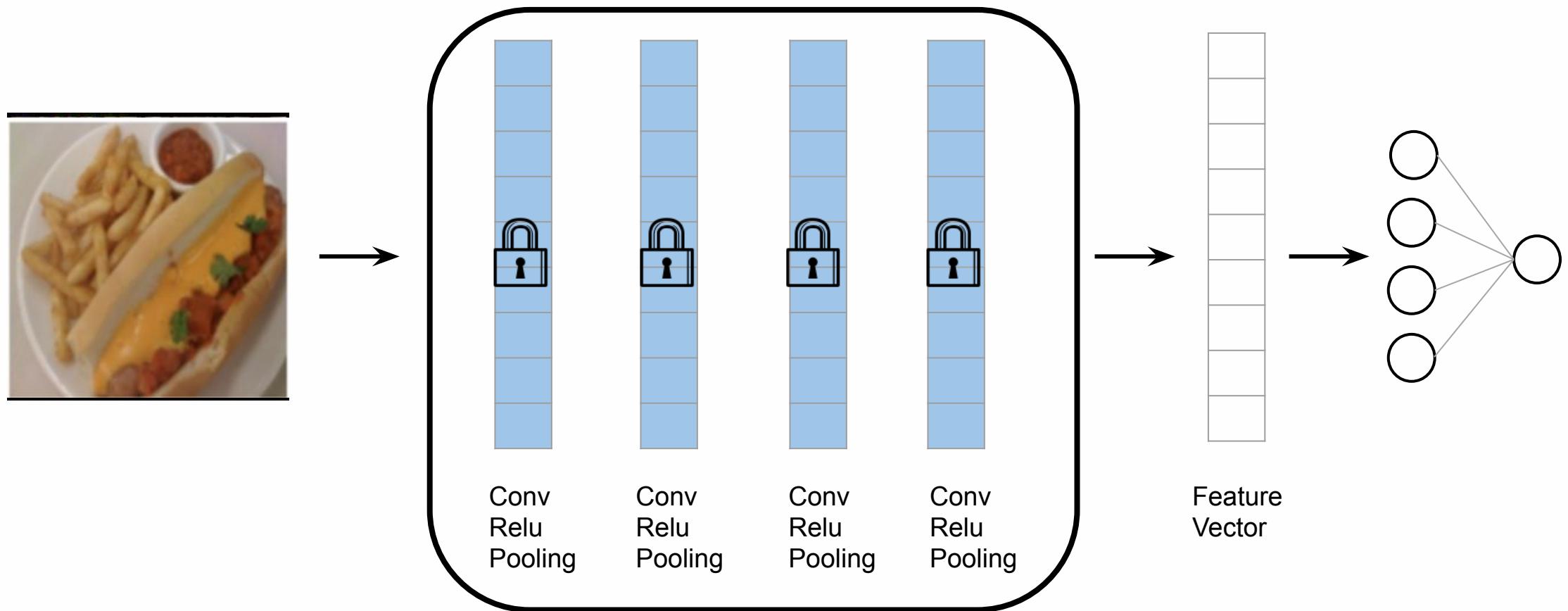
CNN Architecture: Feature Extraction



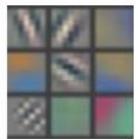
CNN Architecture: Feature Extraction



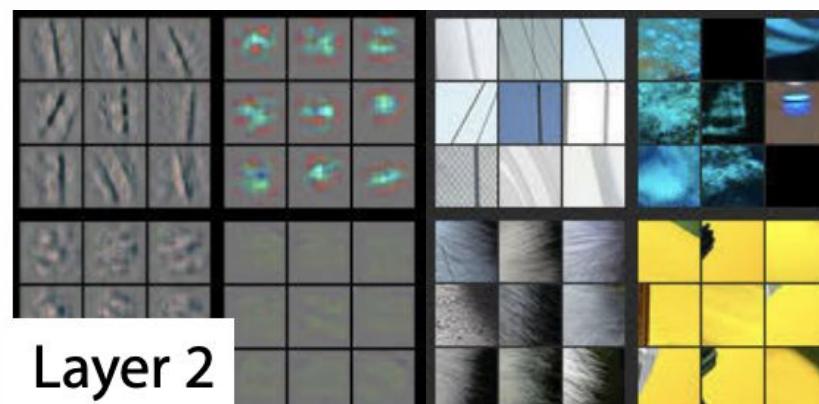
CNN Architecture: Feature Extraction



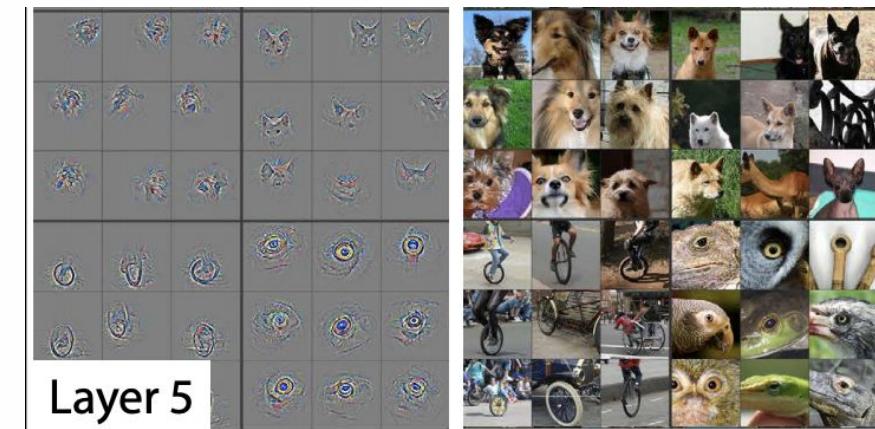
CNN Architecture: Learned Features



Layer 1



Layer 2



Layer 5



CNN Architecture: Learned Features

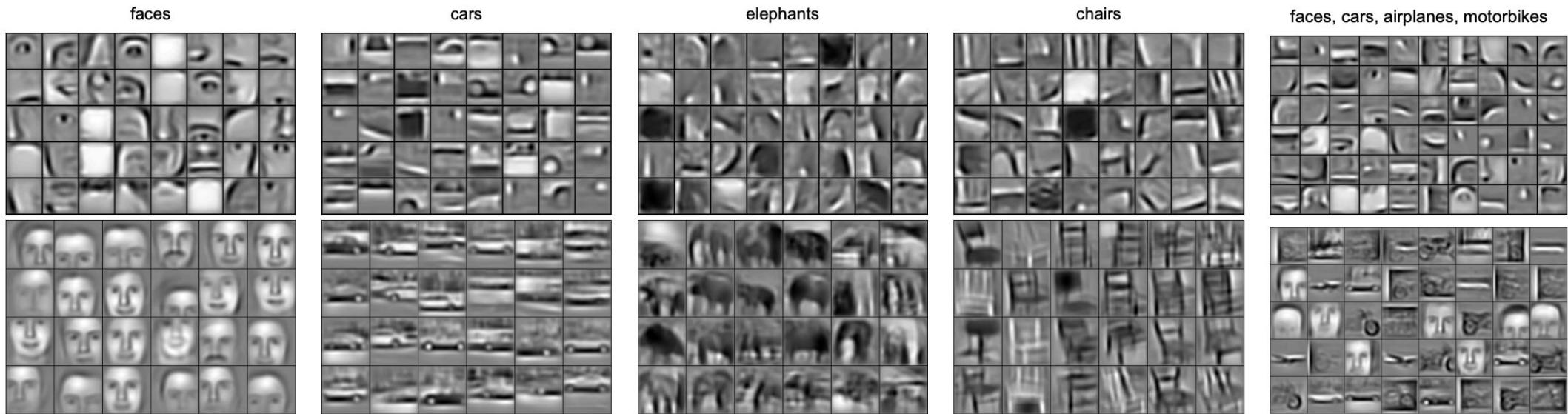
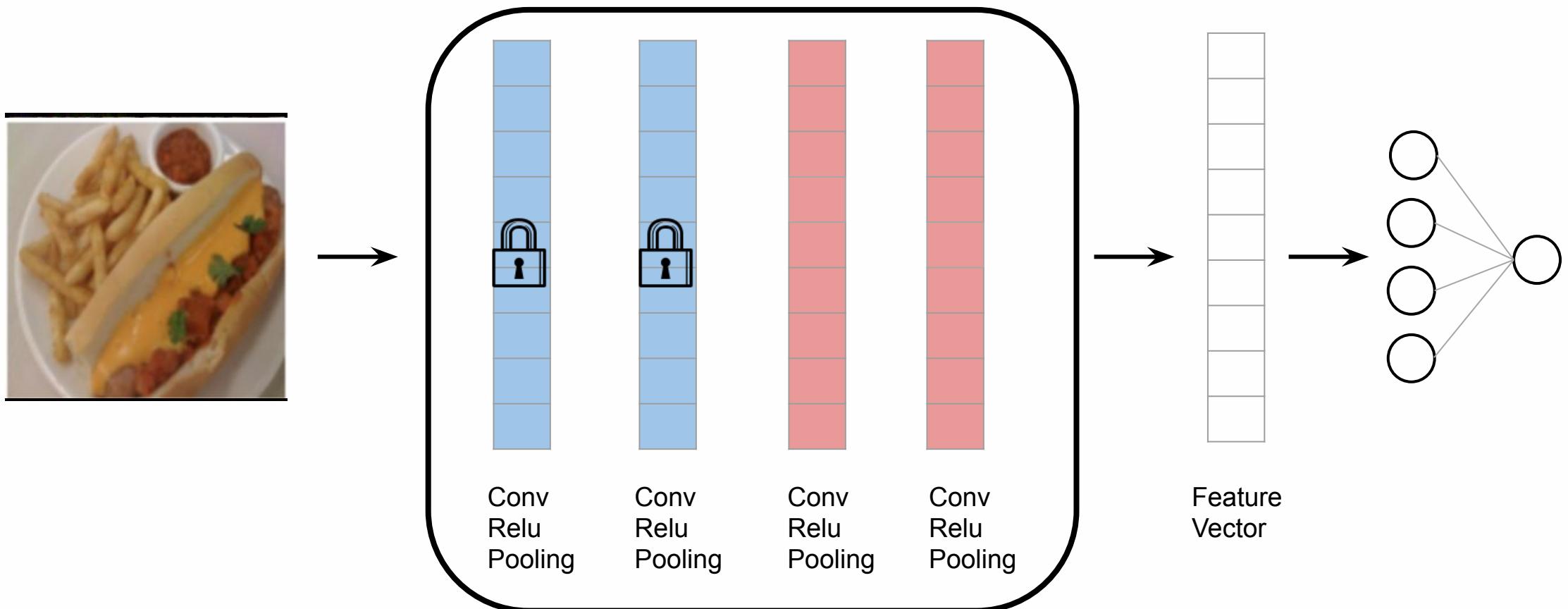


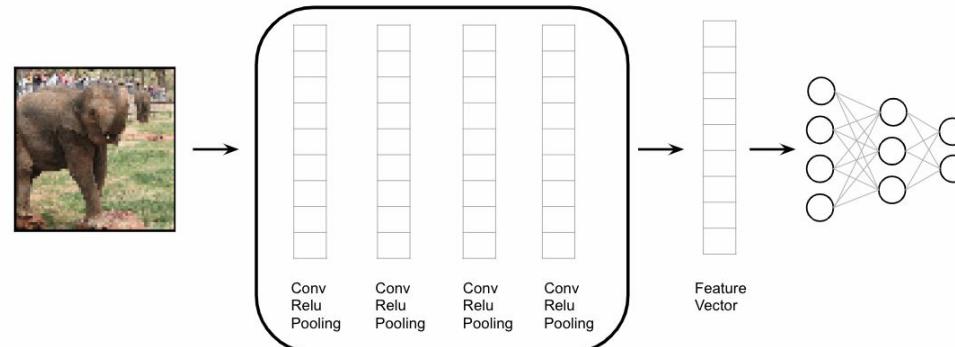
Figure 3. Columns 1-4: the second layer bases (top) and the third layer bases (bottom) learned from specific object categories. Column 5: the second layer bases (top) and the third layer bases (bottom) learned from a mixture of four object categories (faces, cars, airplanes, motorbikes).

CNN Architecture: Fine-Tuning



Transfer Learning

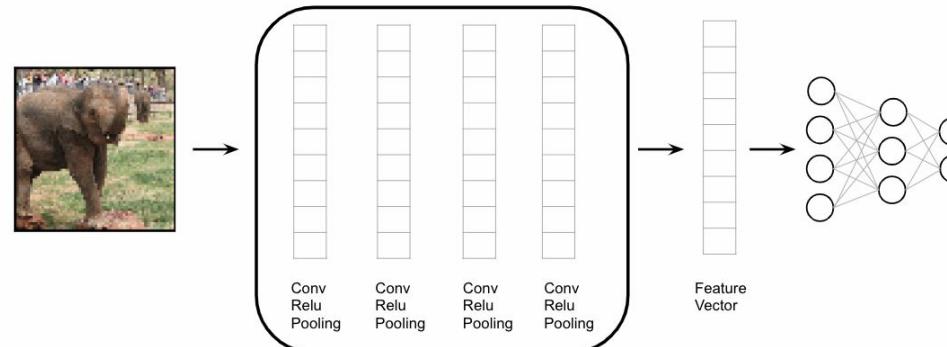
Исходная нейронная сеть,
обученная на ImageNet



на выходе: softmax для 1000 классов

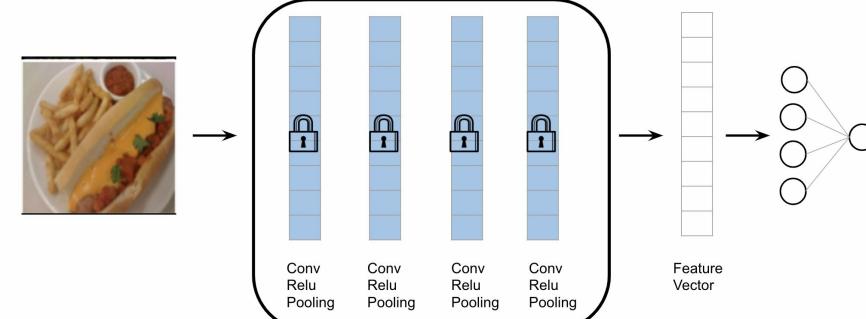
Transfer Learning

Исходная нейронная сеть,
обученная на ImageNet



на выходе: softmax для 1000 классов

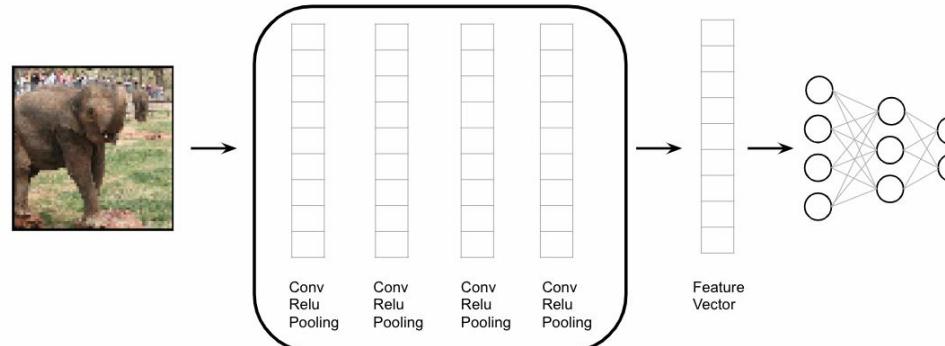
Feature Extraction: замораживаем
сверточные слои, меняем “голову”



Решаем задачу бинарной классификации
(есть ли на фото хот-дог) => 1 выход

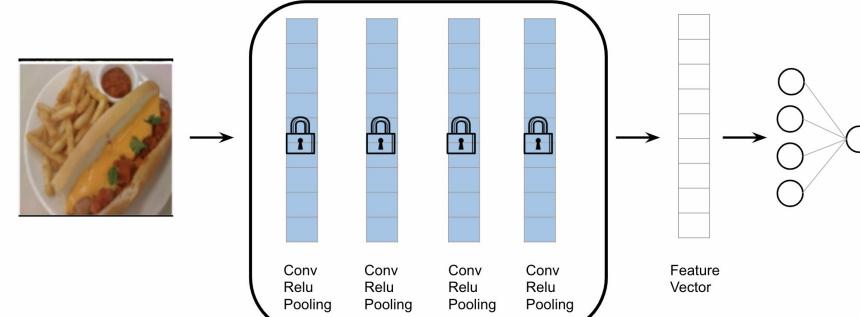
Transfer Learning

Исходная нейронная сеть,
обученная на ImageNet



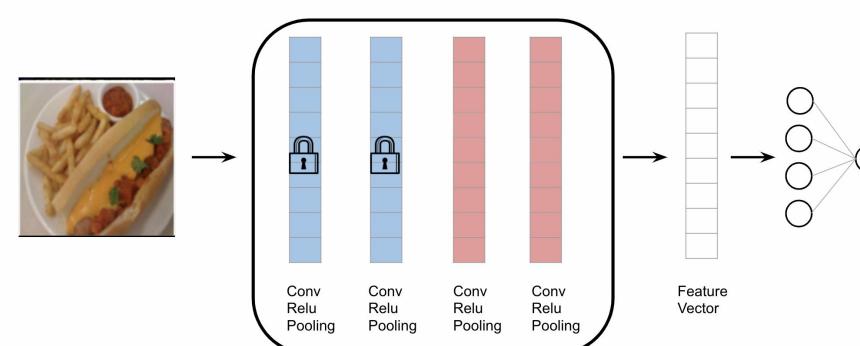
на выходе: softmax для 1000 классов

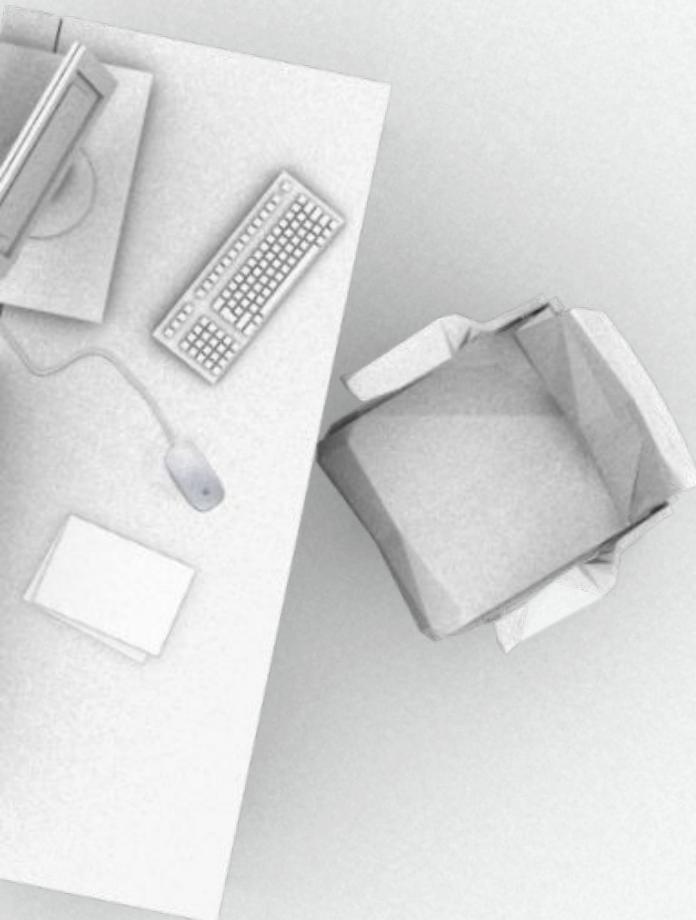
Feature Extraction: замораживаем
сверточные слои, меняем “голову”



Решаем задачу бинарной классификации
(есть ли на фото хот-дог) => 1 выход

Fine tuning: размораживаем
несколько последних слоев
Feature Extactor'a, поскольку они
отвечают за высокоуровневые
признаки и дообучаем модель с
небольшой скоростью обучения

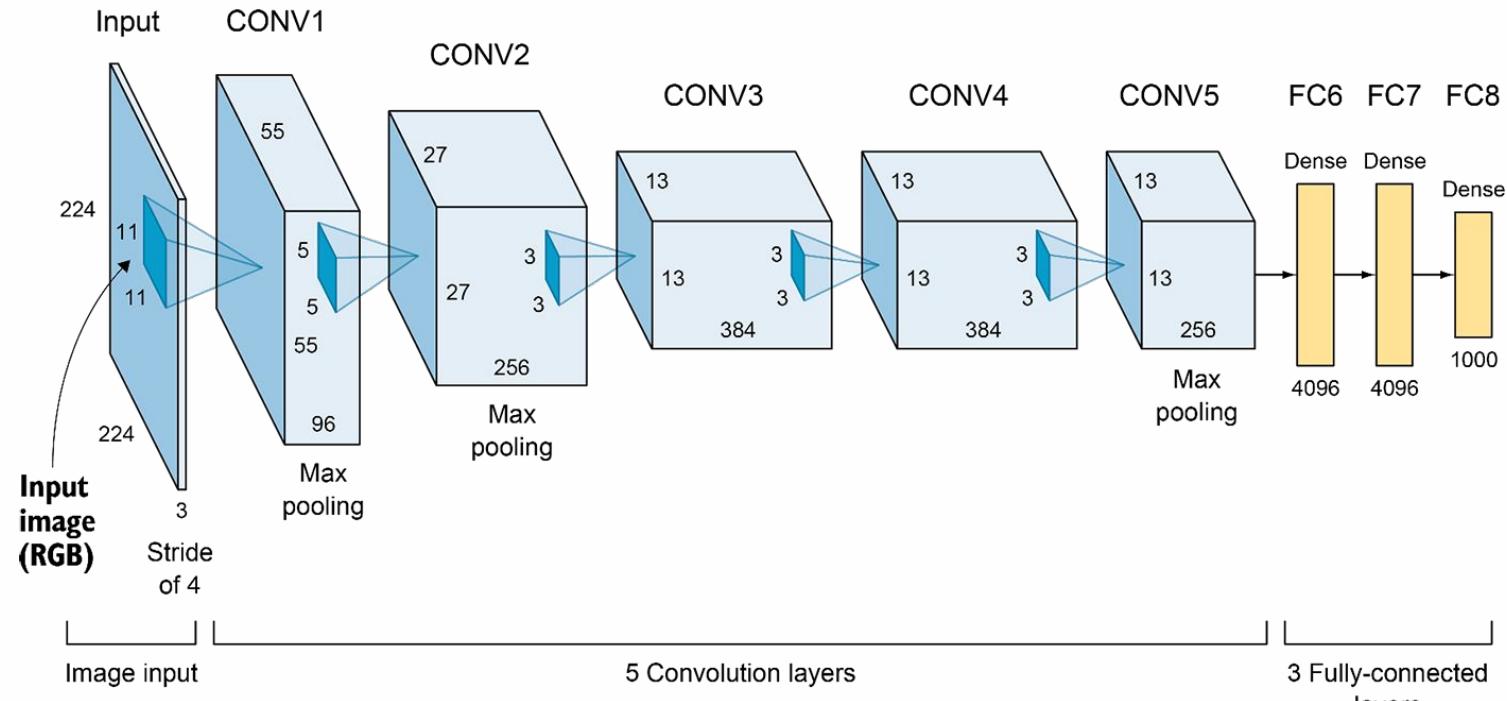
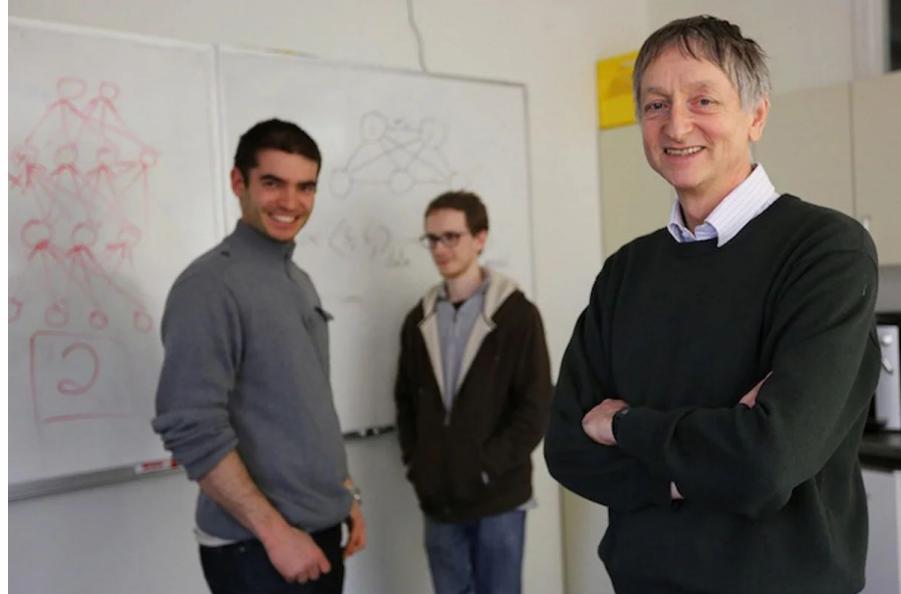




CNN Architectures

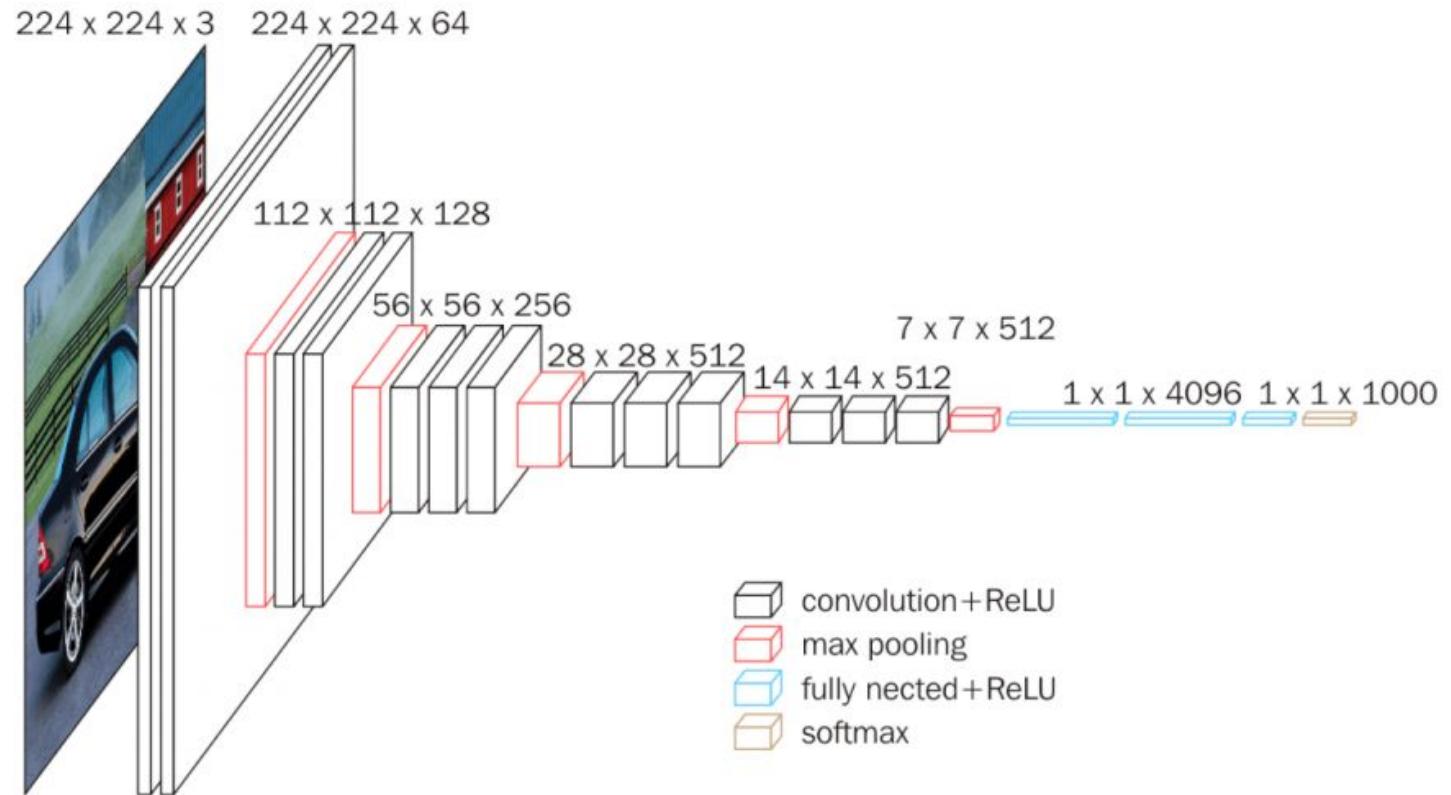
<https://josephpcohen.com/w/visualizing-cnn-architectures-side-by-side-with-mxnet/>

AlexNet (2012)



- ReLU
- Аугментации

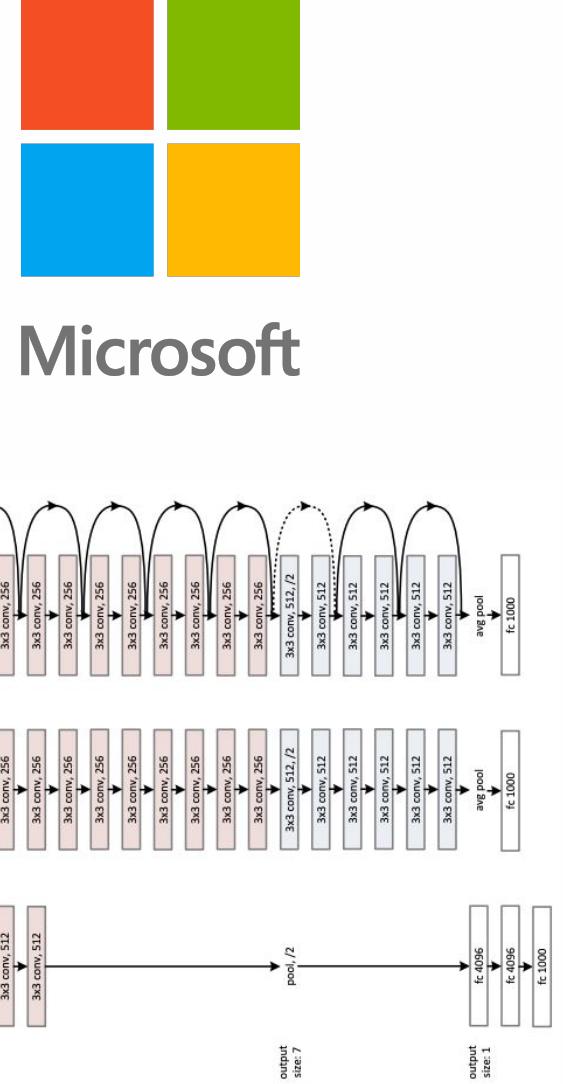
VGG (2014)



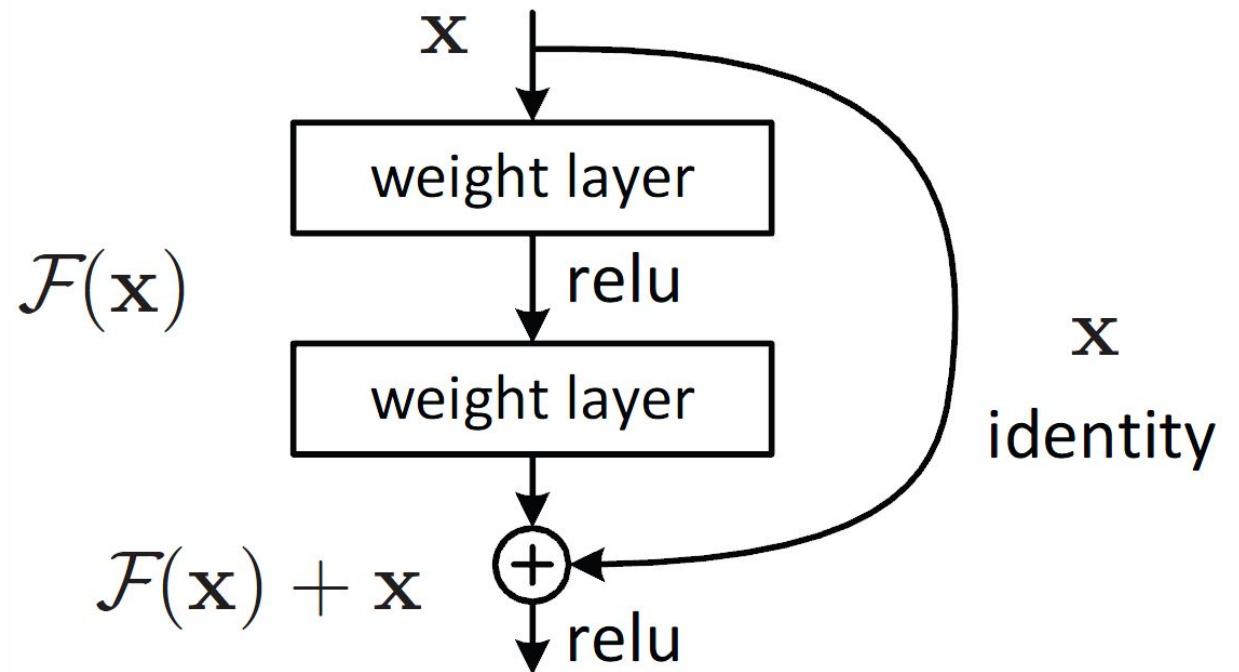
- 3x3
- 11 слоев -> 13 слоев -> ...

ResNet (2015)

- Average Pooling
- от 18 до 1024 слоев
- Residual Block



Residual Block (Skip Connection)



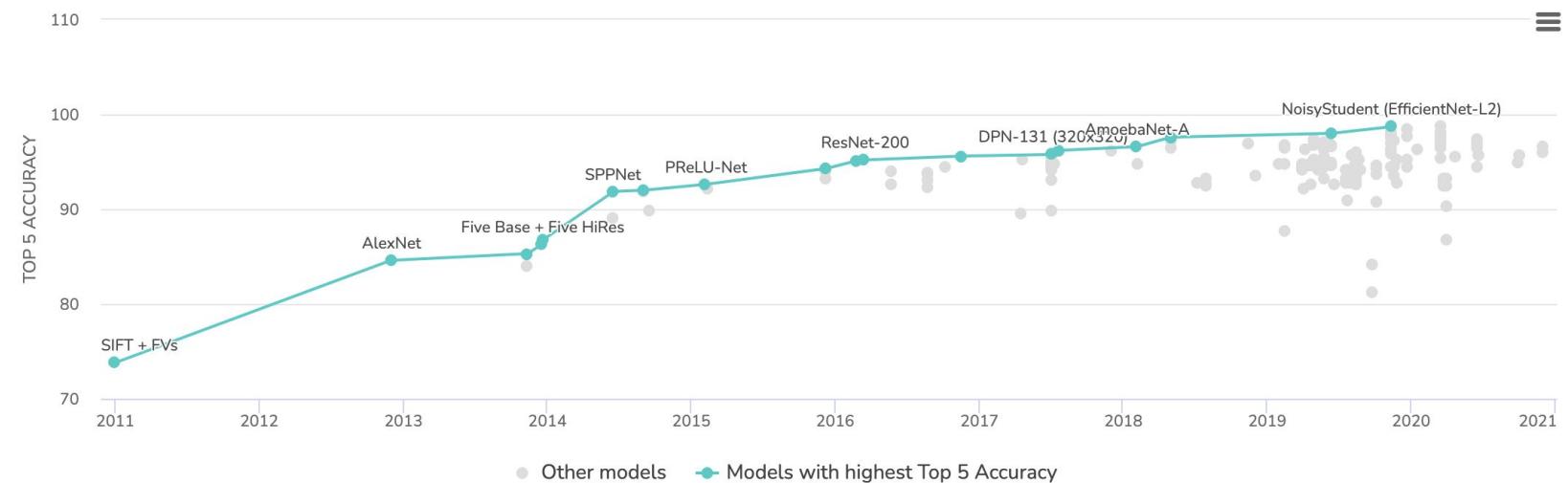
#0150

State Of The Art

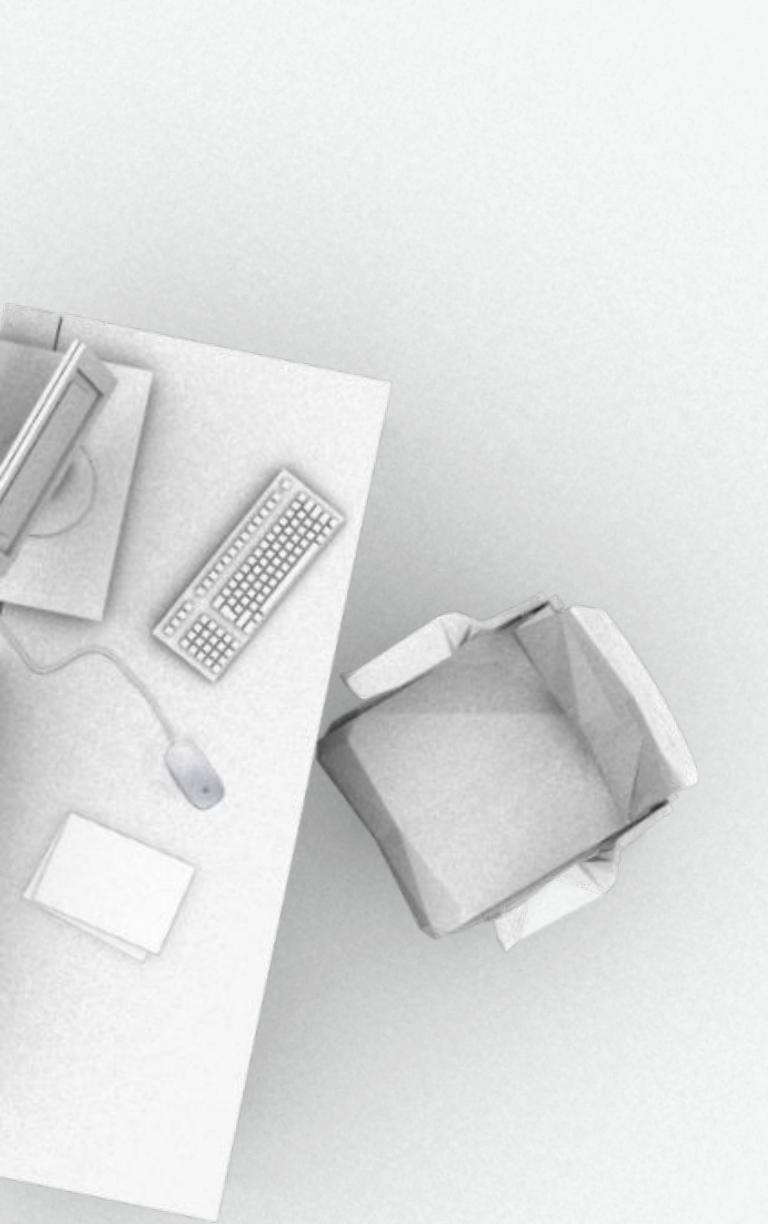
<https://paperswithcode.com/sota>

<https://paperswithcode.com/sota/image-classification-on-imagenet>

Image Classification on ImageNet



#0151



Домашнее задание

Классификация изображений: архитектурные стили

ДЗ №11

7

Баллов
за задание

15.12.20

Срок
сдачи

[https://www.kaggle.com/t/b9828436db
e84d9aad90a16b73f9ff5b](https://www.kaggle.com/t/b9828436db
e84d9aad90a16b73f9ff5b)

Спасибо за внимание!

Господинов Георгий

✉ g.gospodinov@corp.mail.ru

👉 georgygospodinov

