

Глубинное обучение

Лекция 5

Изображения. Операция свертки.

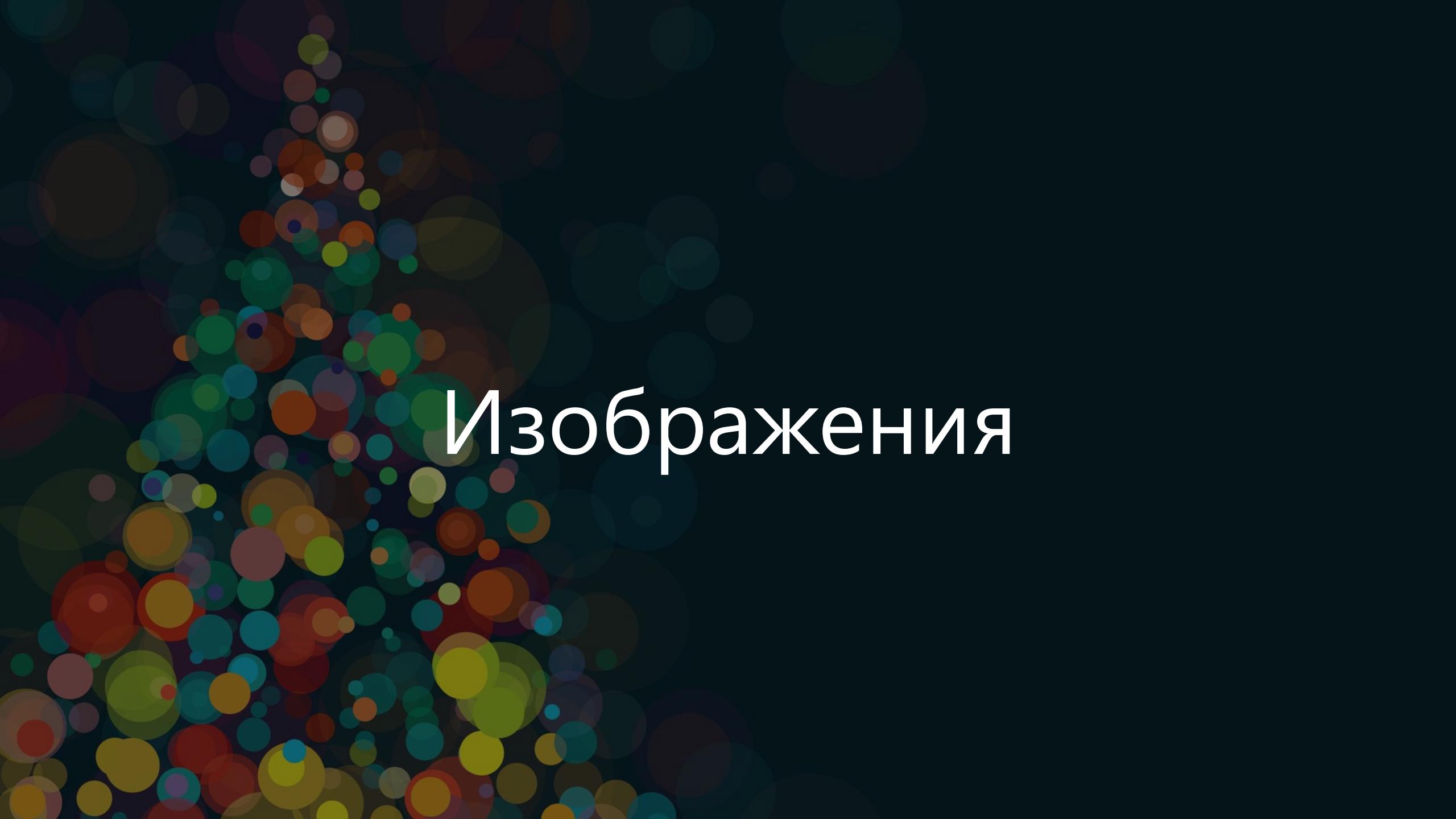
Михаил Гущин

mhushchyn@hse.ru

НИУ ВШЭ, 2024



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

The background of the slide features a dark, semi-transparent overlay of numerous small, semi-transparent colored circles in various sizes and colors, including shades of red, orange, yellow, green, blue, and purple, creating a bokeh effect.

Изображения

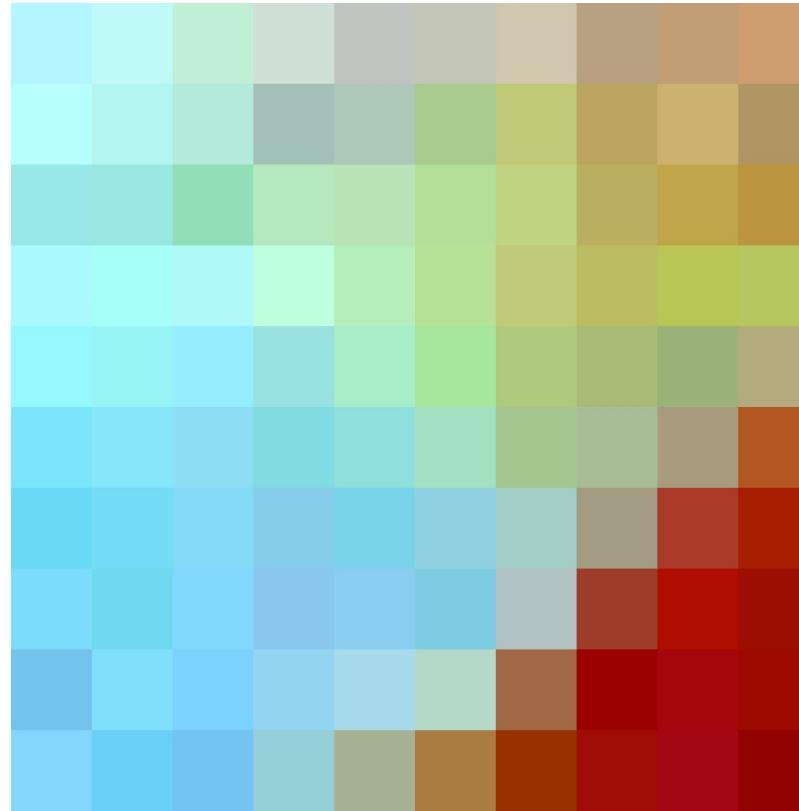
Изображение



Пиксели

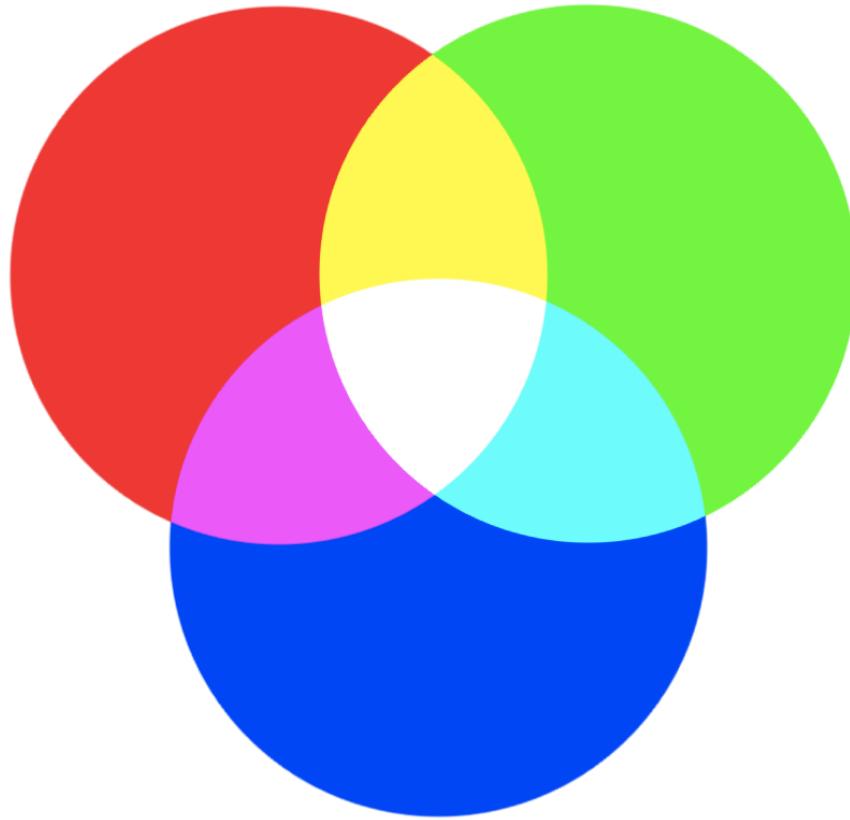


Пиксели



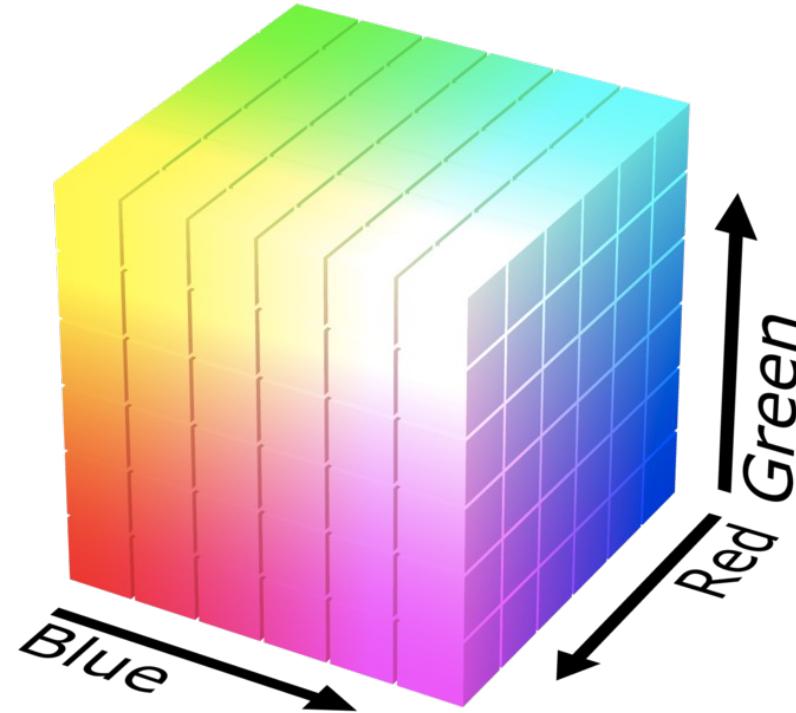
- ▶ Изображение состоит из матрицы пикселей
- ▶ Каждый пиксель имеет свой цвет

Цветовая модель



- ▶ Цвета пикселей представляются в виде комбинации красного (**R**), зеленого (**G**) и синего (**B**) цветов
- ▶ Такой способ называется **RGB цветовой моделью**

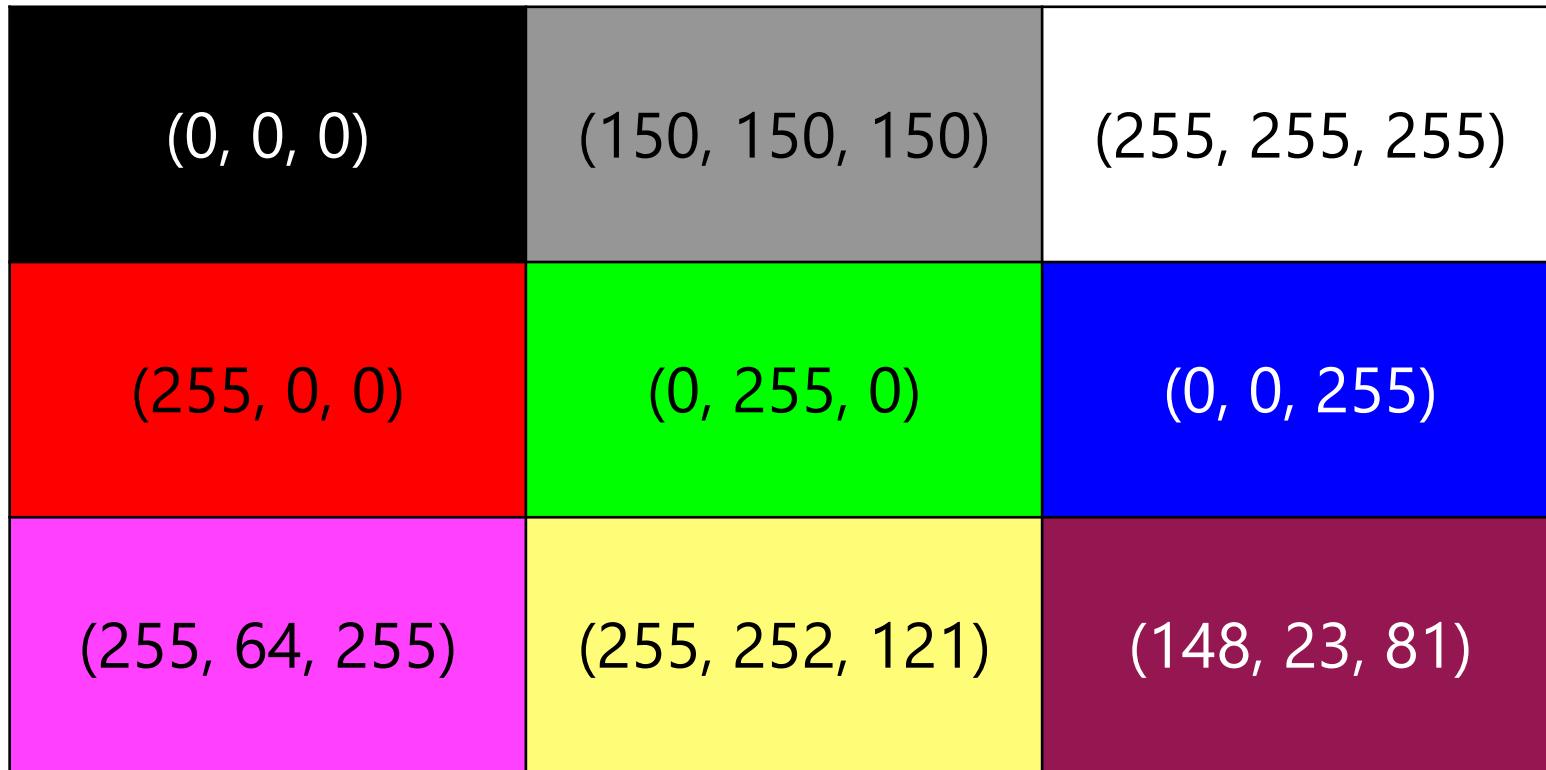
Цветовая модель



- ▶ Любой цвет задается тремя числами от 0 до 255:

$$(R, G, B)$$

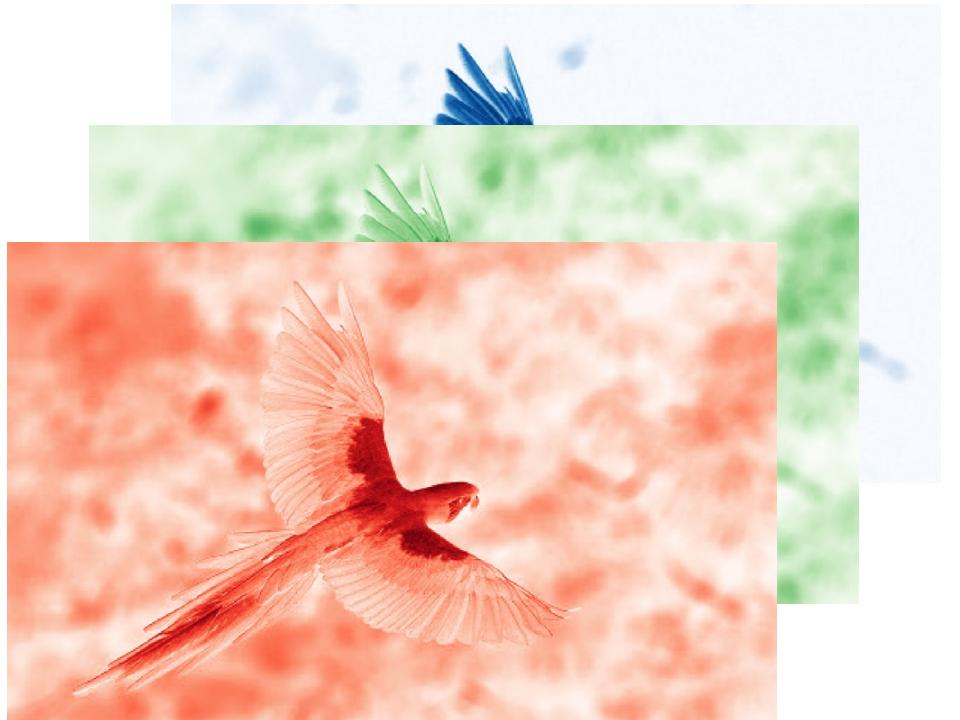
Примеры



Разложение изображений



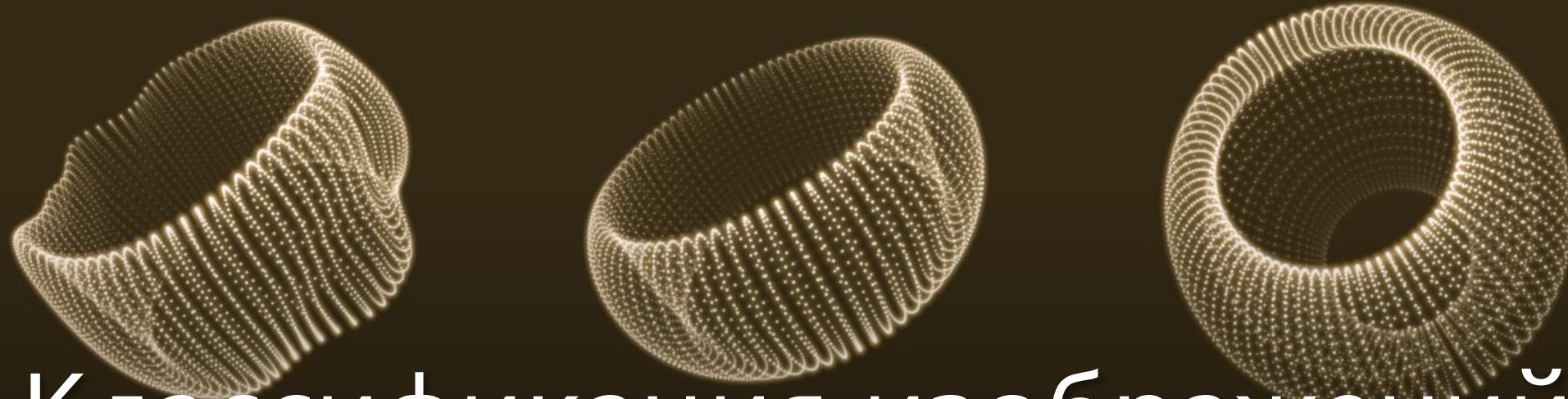
Каналы изображений



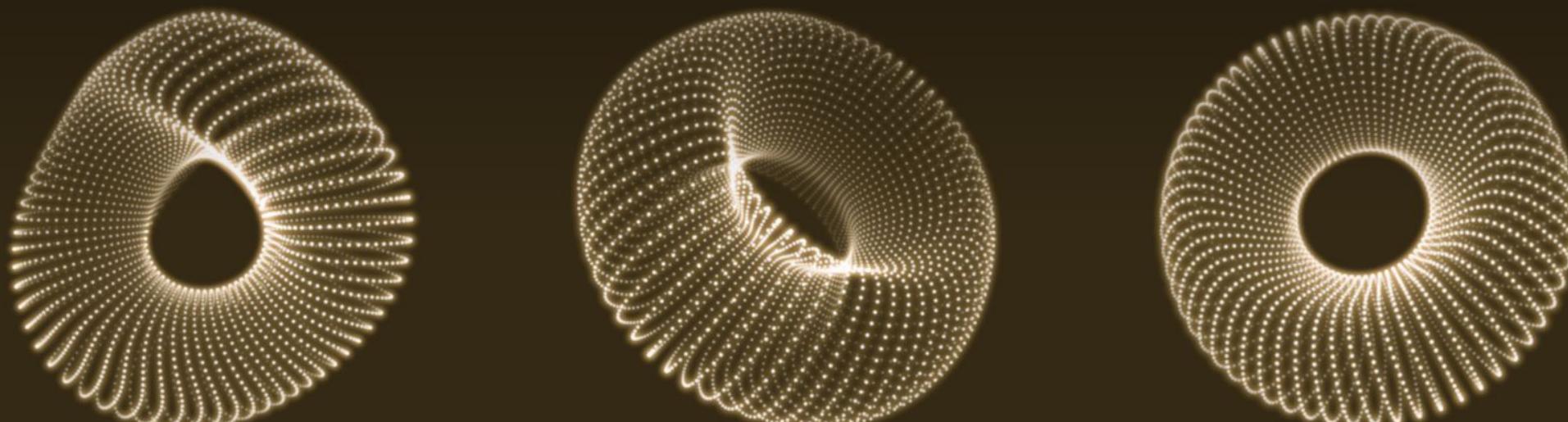
=

88	31	1	22	0
42	195	13	28	1
130	85	43	203	4
18	75	122	187	7
241	61	7	91	8
65	27	155	101	

- ▶ Изображение представляется в виде трех матриц
- ▶ Каждая матрица соответствует одному из цветовых **каналов**

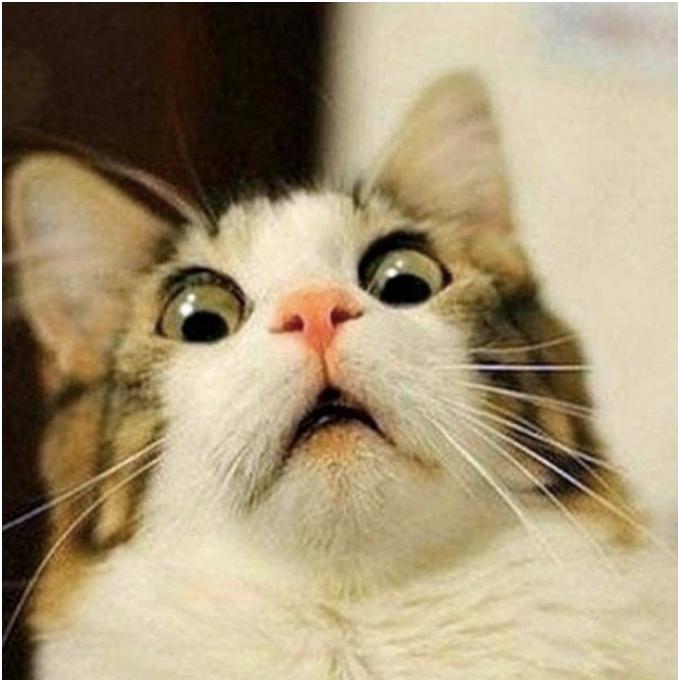


Классификация изображений

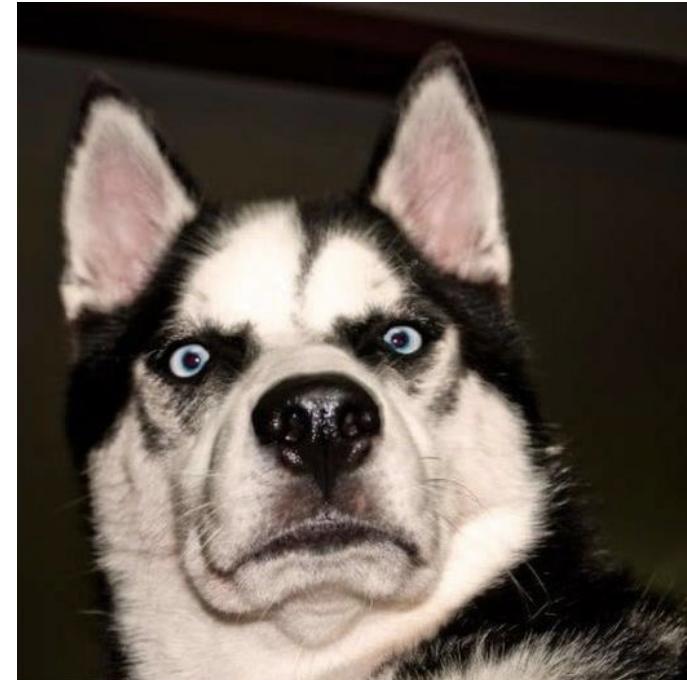


Задача

- ▶ Рассмотрим задачу бинарной классификации изображений
- ▶ Как подать изображение на вход нейронной сети?



VS



Пример

- ▶ Рассмотрим изображение с одним каналом



=

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

Входное изображение

- ▶ **Значения пикселей** в каждом канале изображения — входные признаки для классификатора

Изображение

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

(3, 3, 1)

Выпрямление изображения (flattening)

- Развернем значения пикселей во всех каналах в вектор

Изображение

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

$(3, 3, 1)$

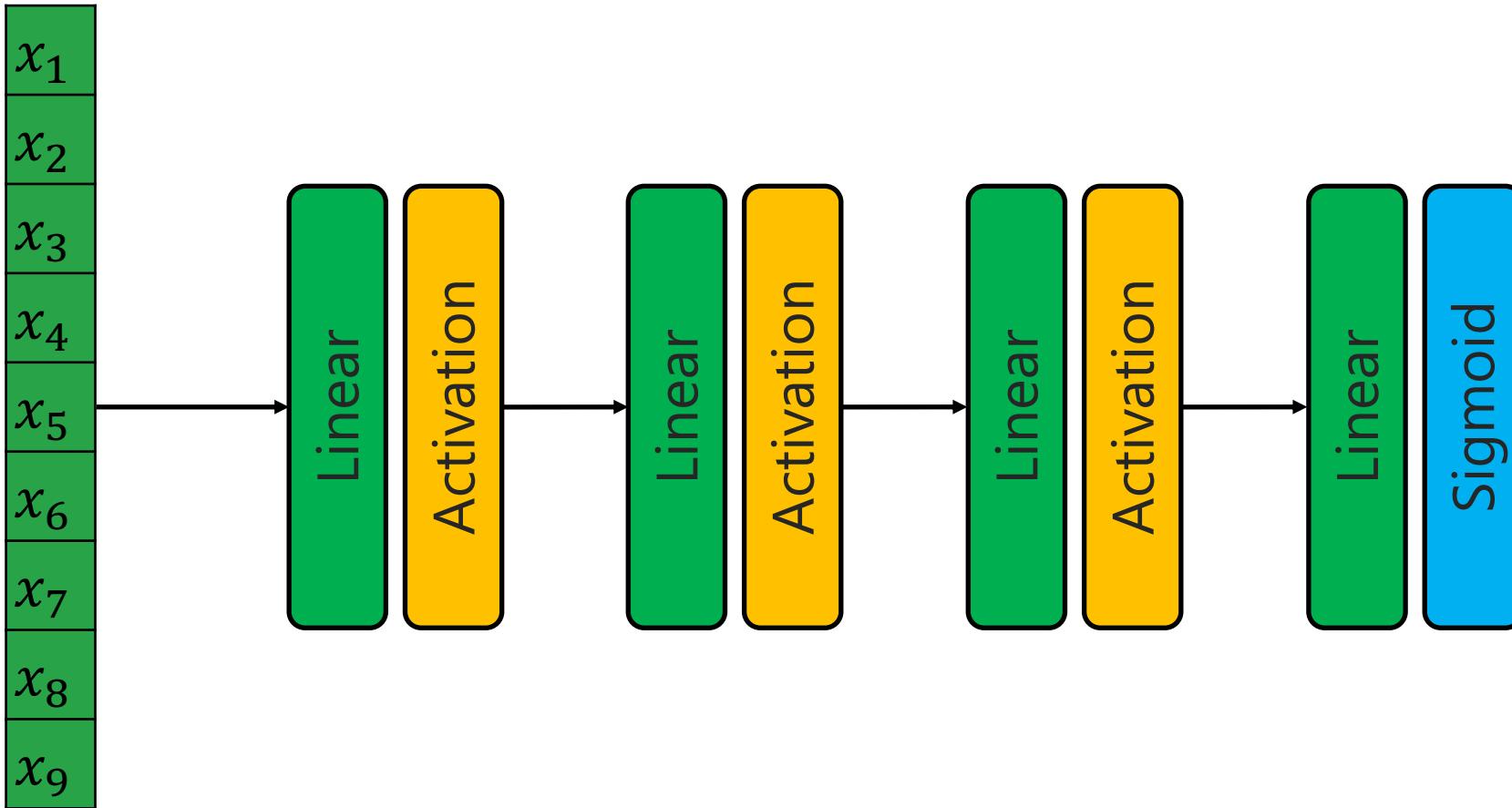
Flattening



x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9

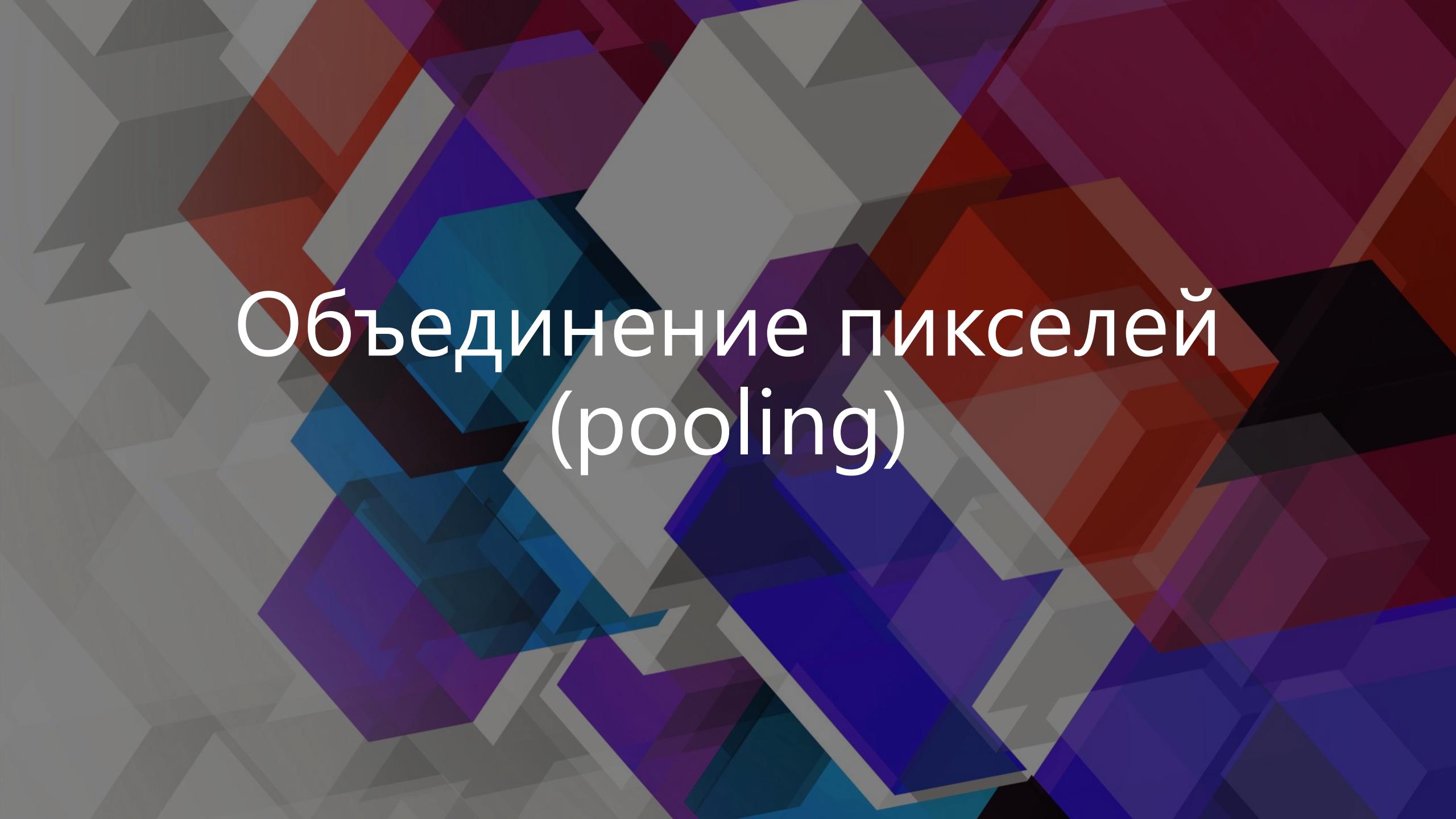
Вектор

Полносвязная нейронная сеть



Выводы

- ▶ Какие преимущества в такой архитектуре?
- ▶ Какие недостатки этого подхода?



Объединение пикселей
(pooling)

Мотивация

- ▶ Изображение в HD разрешении имеет 1280 x 720 пикселей
- ▶ Каждый пиксель имеет 3 цветовых канала
- ▶ Общее количество признаков N :

$$N = 1280 * 720 * 3 = 2\ 764\ 800$$

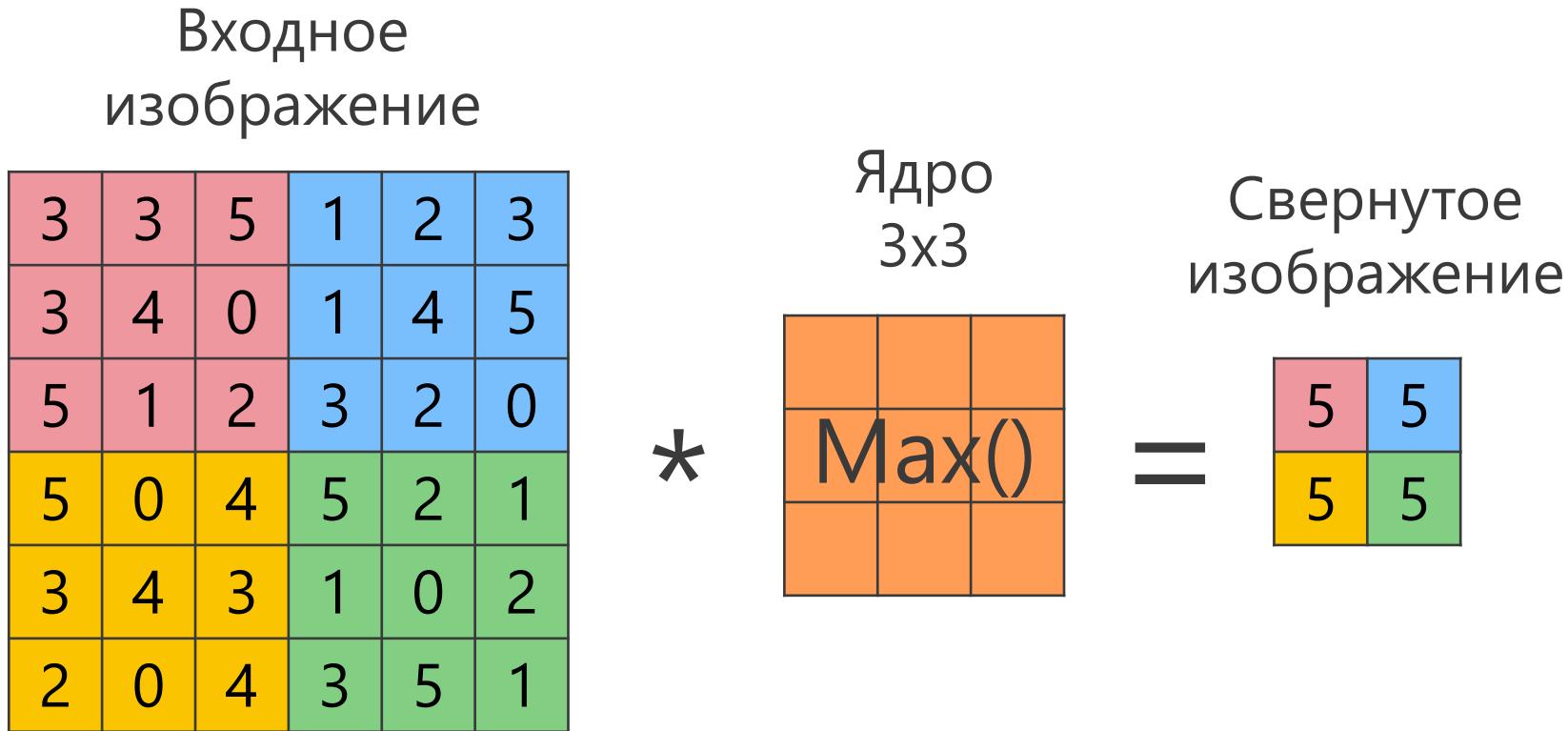
- ▶ **Слишком большая** размерность для полносвязной сети

Мотивация

- ▶ Корреляция между близкими пикселями больше
- ▶ Связь между далекими пикселями меньше

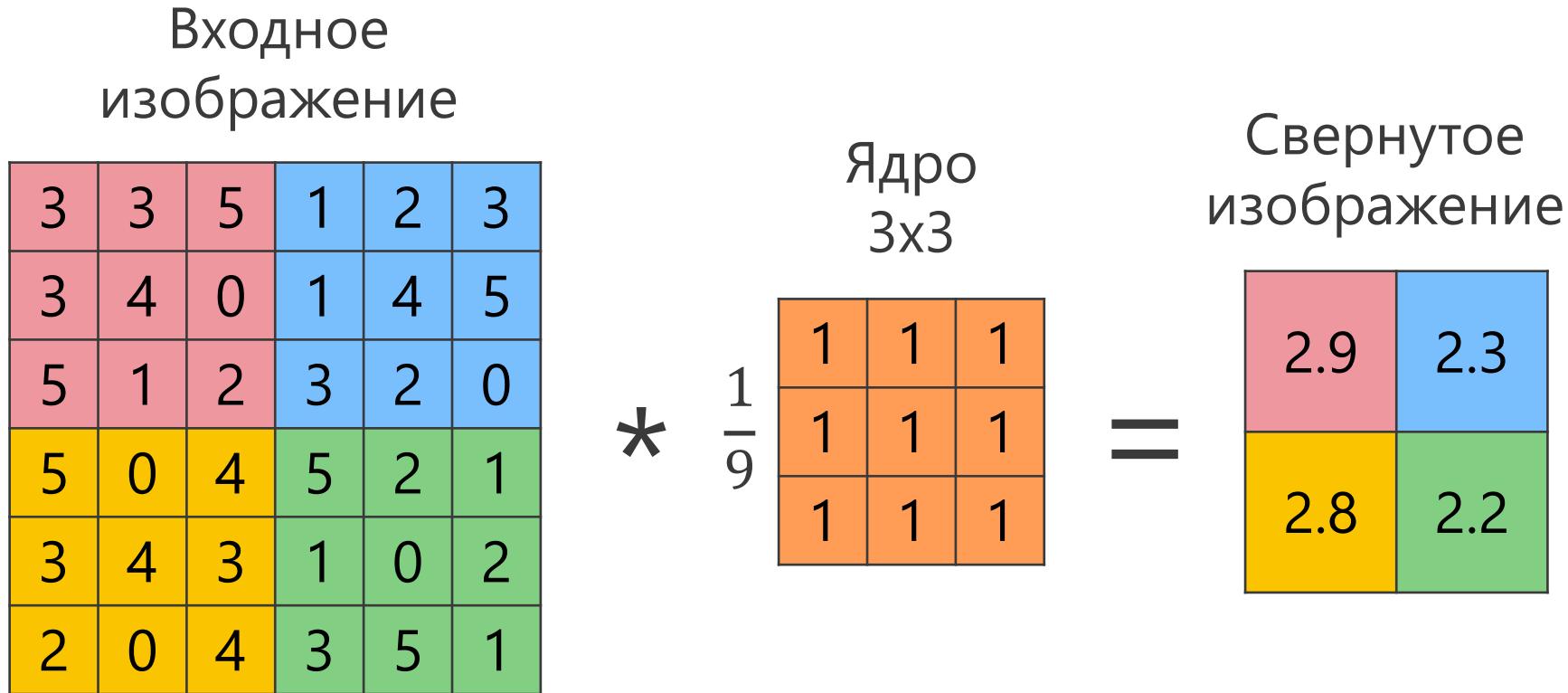


MaxPooling



- ▶ **Объединение по максимуму (MaxPooling)** возвращает максимальное значение соседних пикселей
- ▶ Выбирают шаг так, чтобы ядра не пересекались

AvgPooling



- ▶ **Объединение по среднему (AvgPooling)** возвращает среднее значение соседних пикселей
- ▶ Выбирают шаг так, чтобы ядра не пересекались

Свёртки изображений

Пример

- ▶ Рассмотрим изображение с одним каналом
- ▶ Как можем учесть локальную связь пикселей?



=

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

Операция свертки

Входное изображение

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

Ядро
3x3

0	1	0
-1	2	-1
0	1	0

*

=

Свернутое изображение

9			

$$\begin{aligned} & 3 * 0 + 3 * 1 + 5 * 0 - \\ & 3 * 1 + 4 * 2 - 0 * 1 + \\ & 5 * 0 + 1 * 1 + 2 * 0 = 9 \end{aligned}$$

Операция свертки

Входное изображение

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

Ядро
3x3

0	1	0
-1	2	-1
0	1	0

*

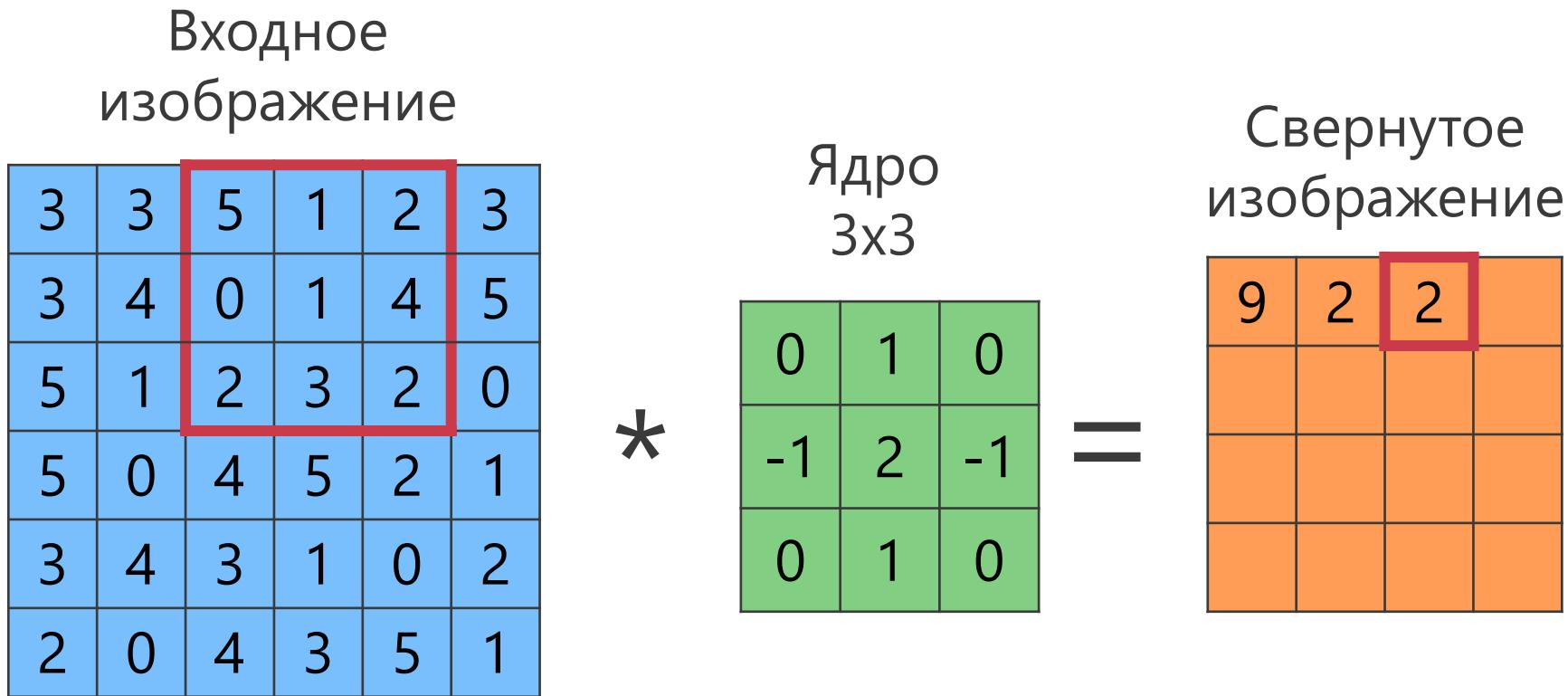
=

Свернутое изображение

9	2		

$$\begin{aligned} & 3 * 0 + 5 * 1 + 1 * 0 - \\ & 4 * 1 + 0 * 2 - 1 * 1 + \\ & 1 * 0 + 2 * 1 + 3 * 0 = 2 \end{aligned}$$

Операция свертки



$$\begin{aligned} & 5 * 0 + 1 * 1 + 2 * 0 - \\ & 0 * 1 + 1 * 2 - 4 * 1 + \\ & 2 * 0 + 3 * 1 + 2 * 0 = 2 \end{aligned}$$

Операция свертки

Входное изображение

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

*

Ядро
3x3

0	1	0
-1	2	-1
0	1	0

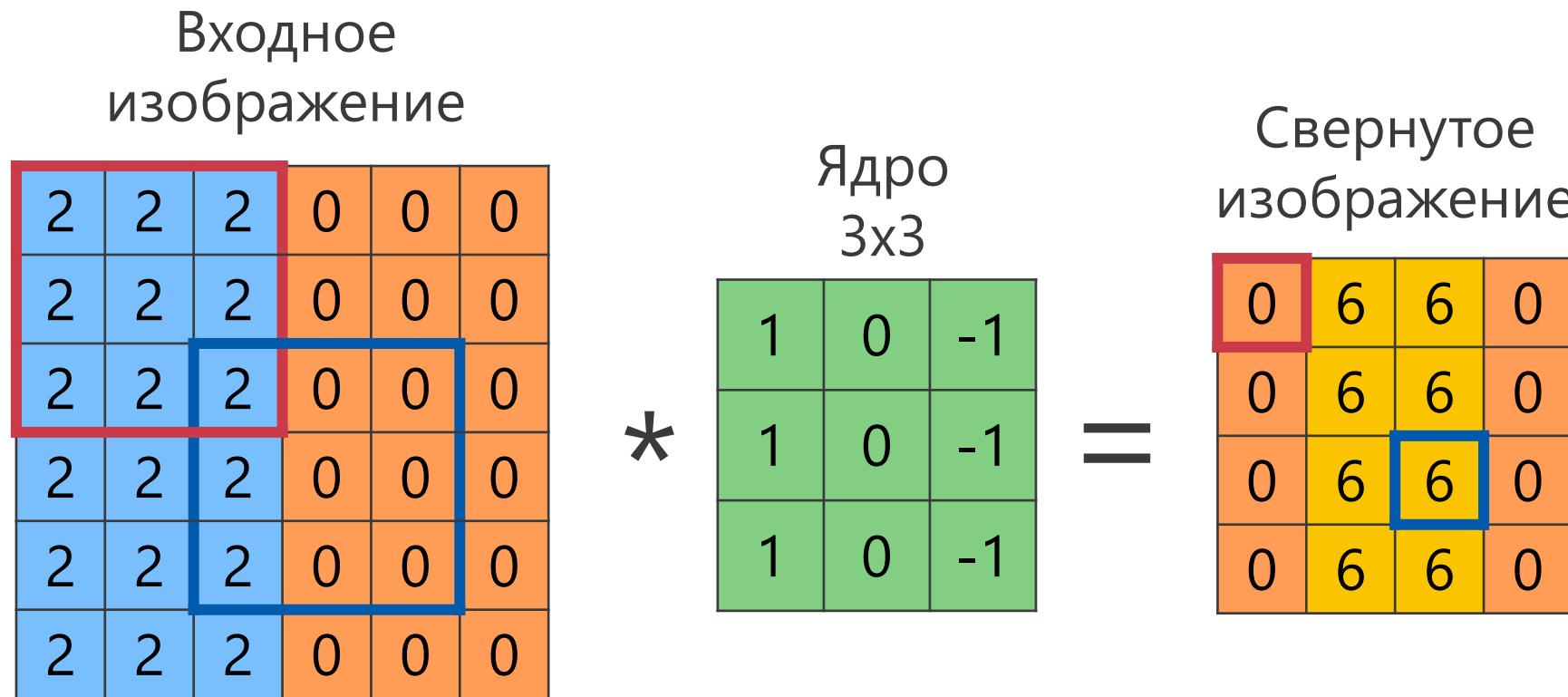
=

Свернутое изображение

9	2	2	6
-1	4	8	7
-4	8	8	0
2	9	7	4

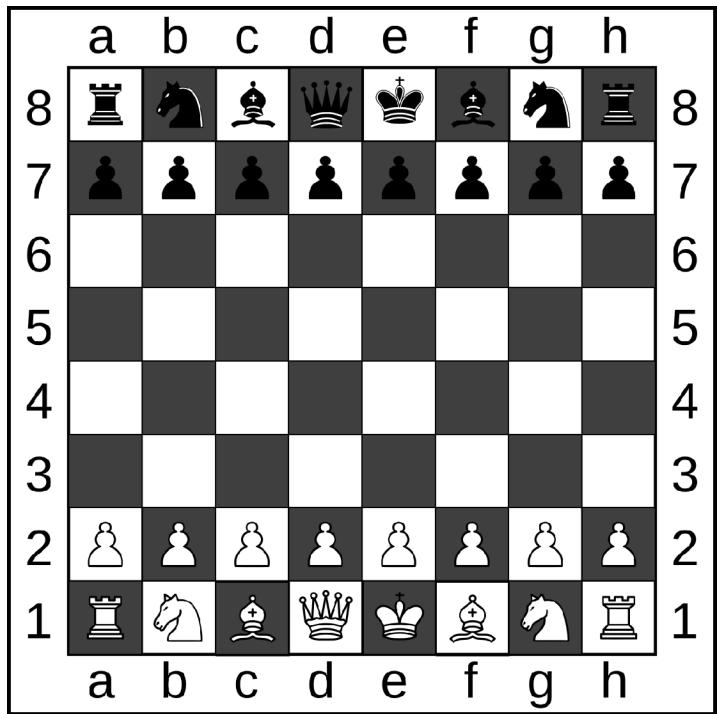
- ▶ **Свертка** — это процесс сложения соседних элементов изображения, взвешиваемых ядром

Пример: детектор вертикальных границ



- ▶ Эта свертка аналогична подсчету первой производной от изображения вдоль горизонтальной оси

Пример: детектор вертикальных границ



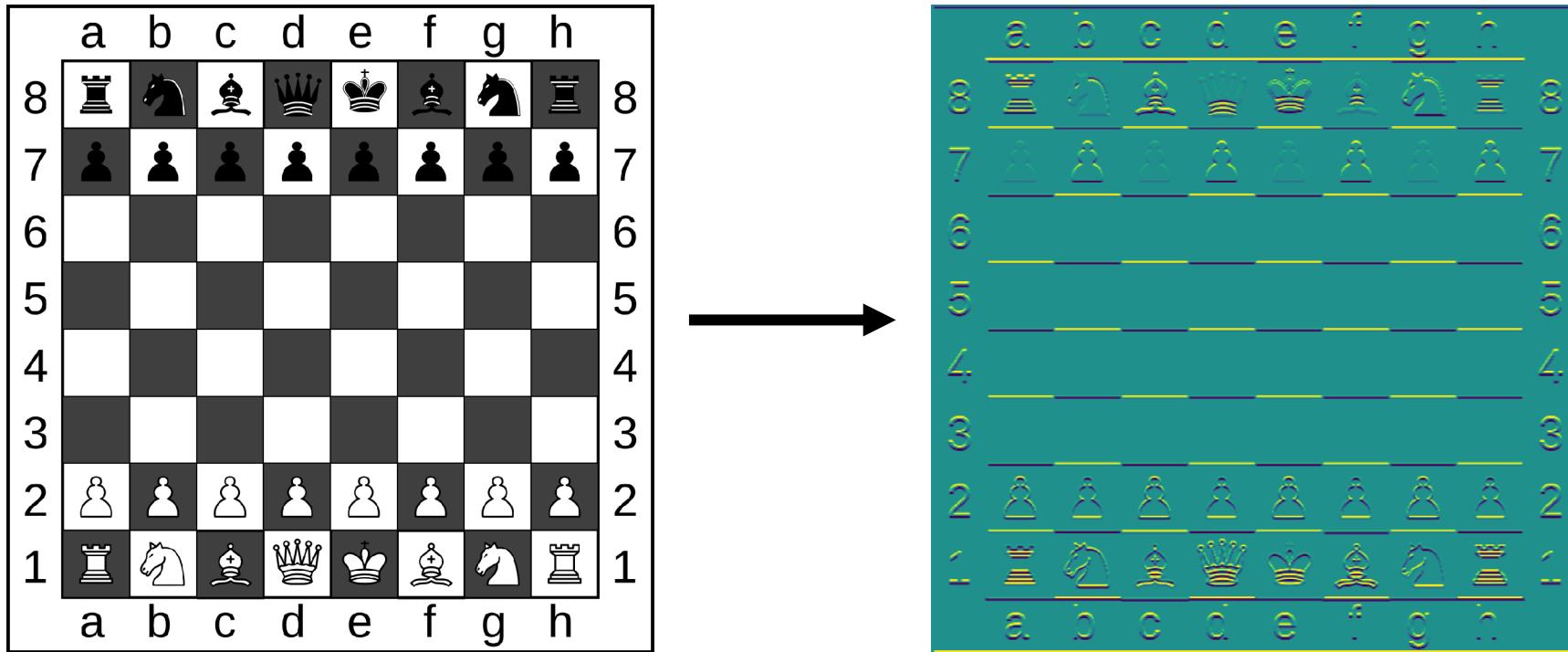
Оператор
Собеля

*

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

- ▶ Применим оператор Собеля для обнаружения горизонтальных линий

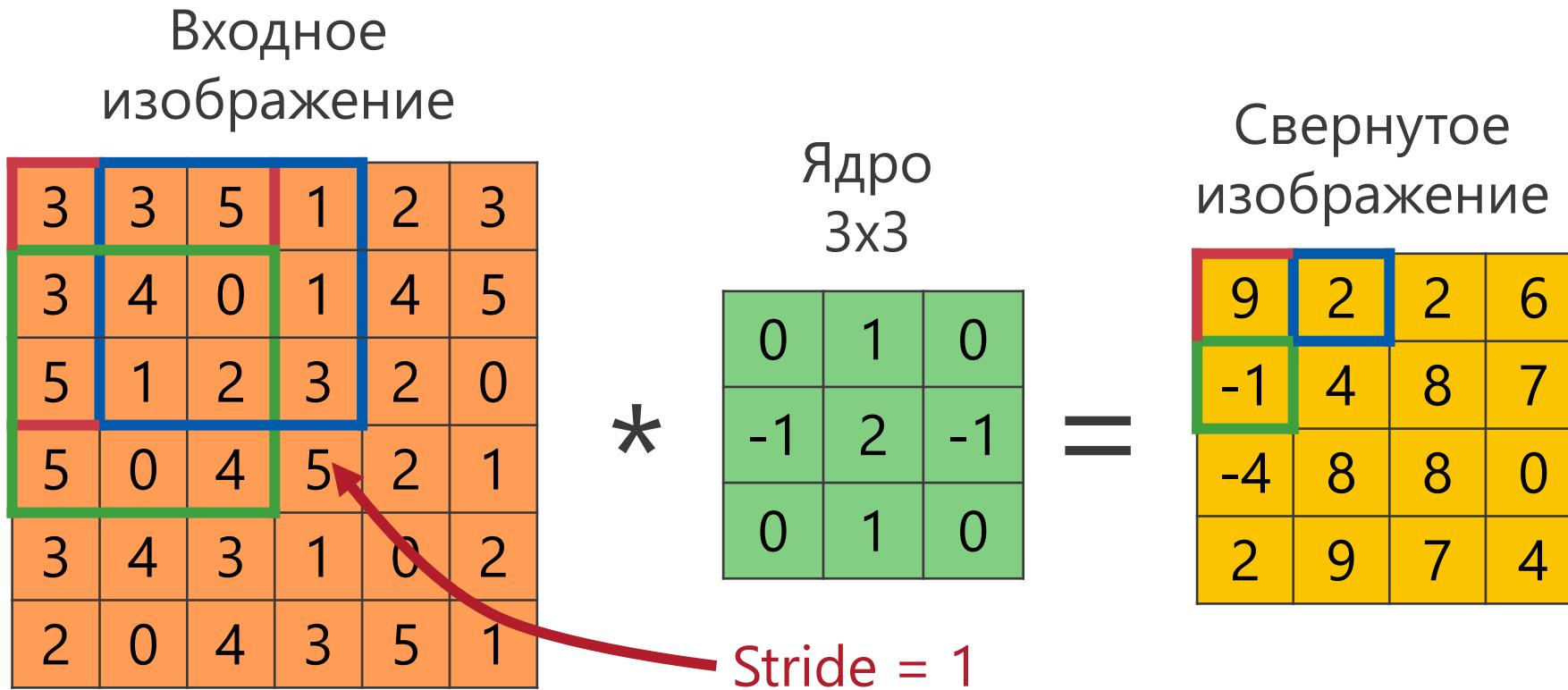
Пример: детектор вертикальных границ



Параметры свёрток

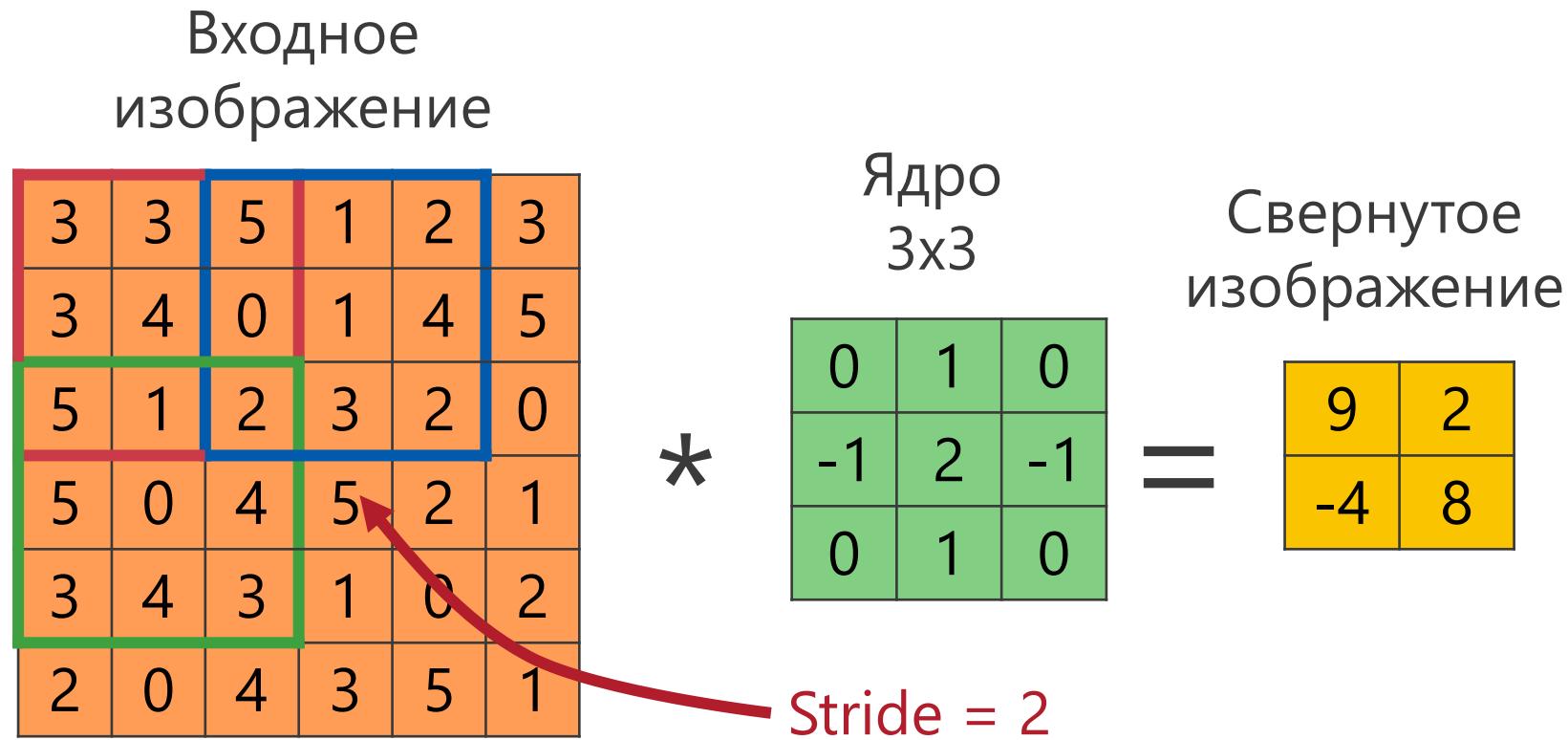
```
types.Operator):
    X mirror to the selected object.mirror_mirror_x"
    "mirror X"
```

Шаг свертки (stride)



- ▶ **Шаг свертки (stride)** — это величина сдвига ядра между соседними операциями свертки изображения

Шаг свертки (stride)



Шаг свертки (stride)

Входное изображение

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

(H, W)

Ядро
 3×3

0	1	0
-1	2	-1
0	1	0

*

(h, w)

Свернутое изображение

9	2
-4	8

$$\left(\left\lceil \frac{H - h}{\text{stride}} + 1 \right\rceil, \left\lceil \frac{W - w}{\text{stride}} + 1 \right\rceil \right)$$

Дополнение изображения (padding)

Входное изображение

4	4	3	2
2	3	4	4
1	1	3	2
1	2	2	2

Ядро
3x3

0	1	0
-1	2	-1
0	1	0

Свернутое изображение

5	7
3	9

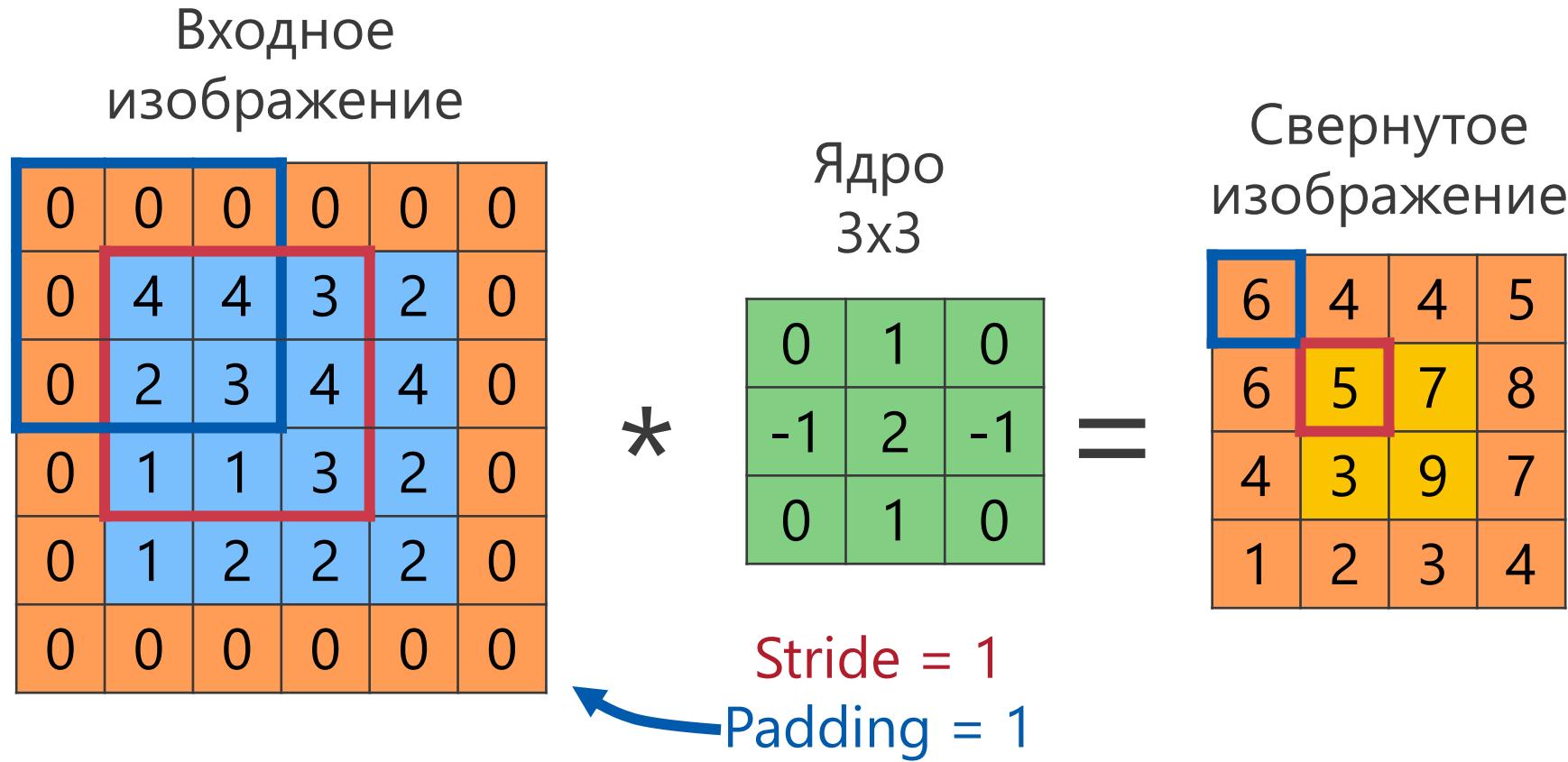
*

=

Stride = 1

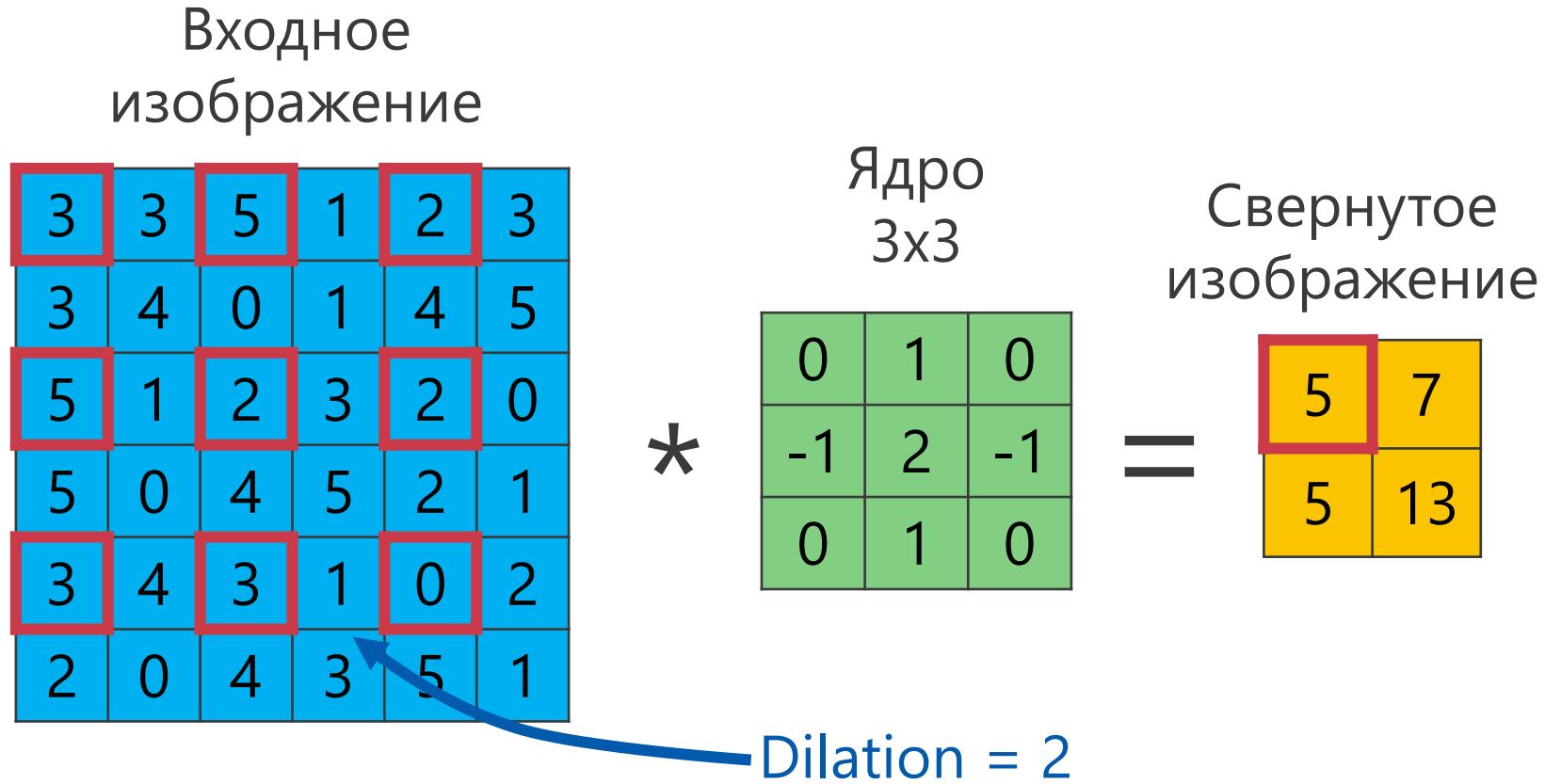
- ▶ Свертка изображения **всегда** уменьшает его размер

Дополнение изображения (padding)



- ▶ **Дополнение изображения (padding)** — это искусственное расширение изображения по краям

Расширение ядра (dilation)

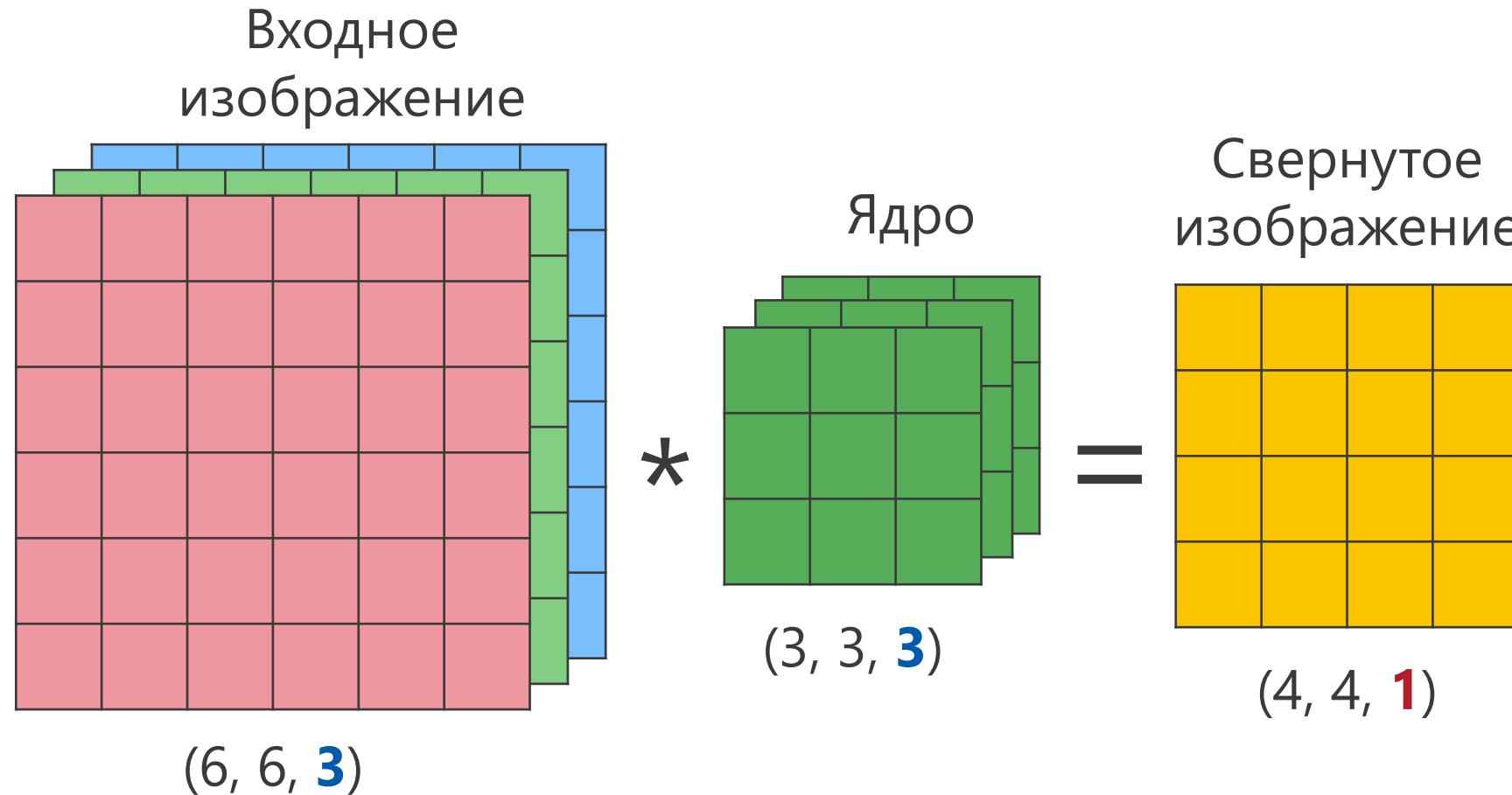


- ▶ **Расширение ядра (dilation)** — это расстояние между соседними элементами ядра

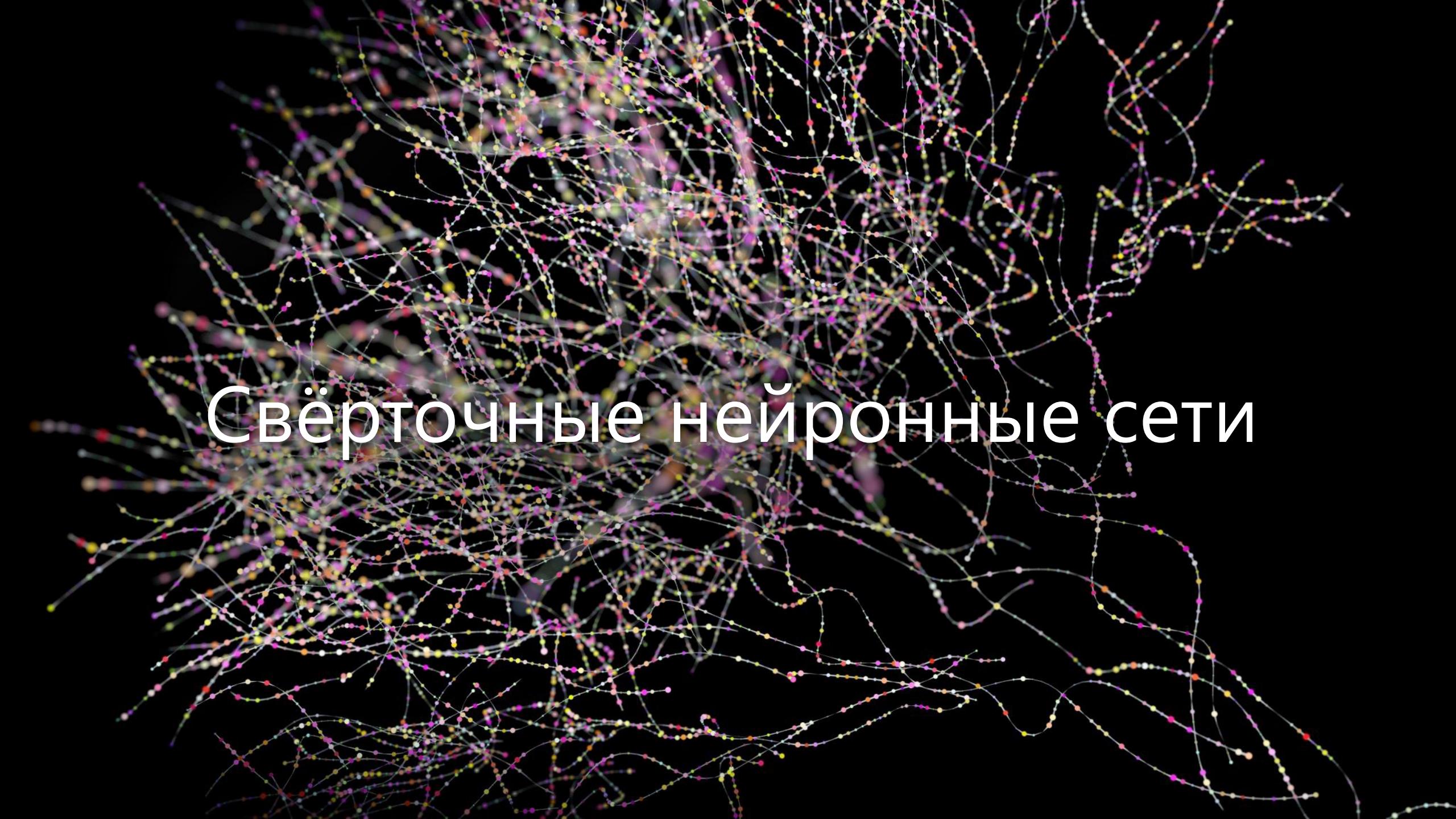
Параметры сверток

- ▶ Размер ядра свертки
- ▶ Шаг свертки (stride)
- ▶ Дополнение изображения (padding)
- ▶ Расширение ядра (dilation)

Свертка многоканального изображения



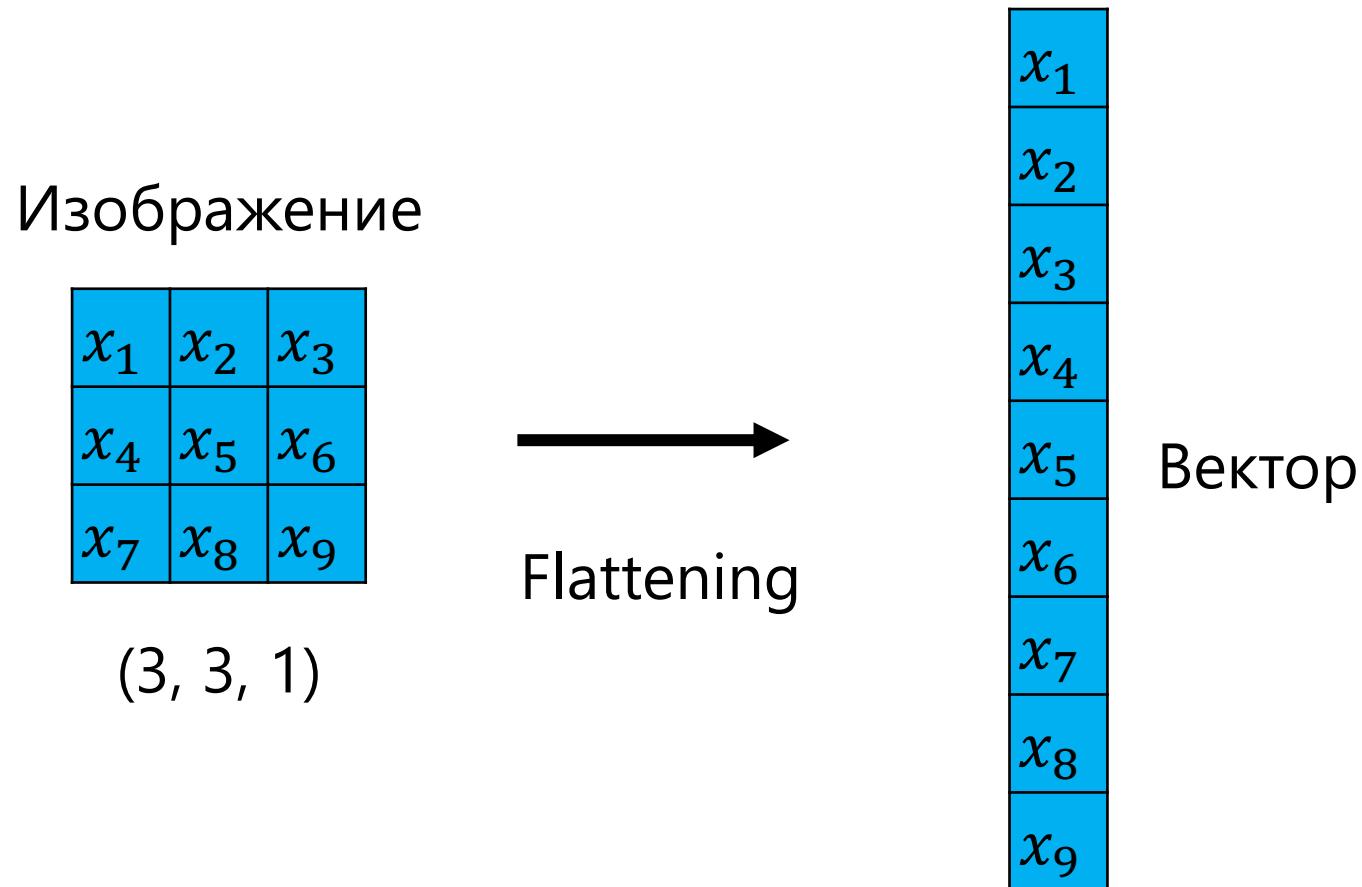
Результат свертки — одноканальное изображение



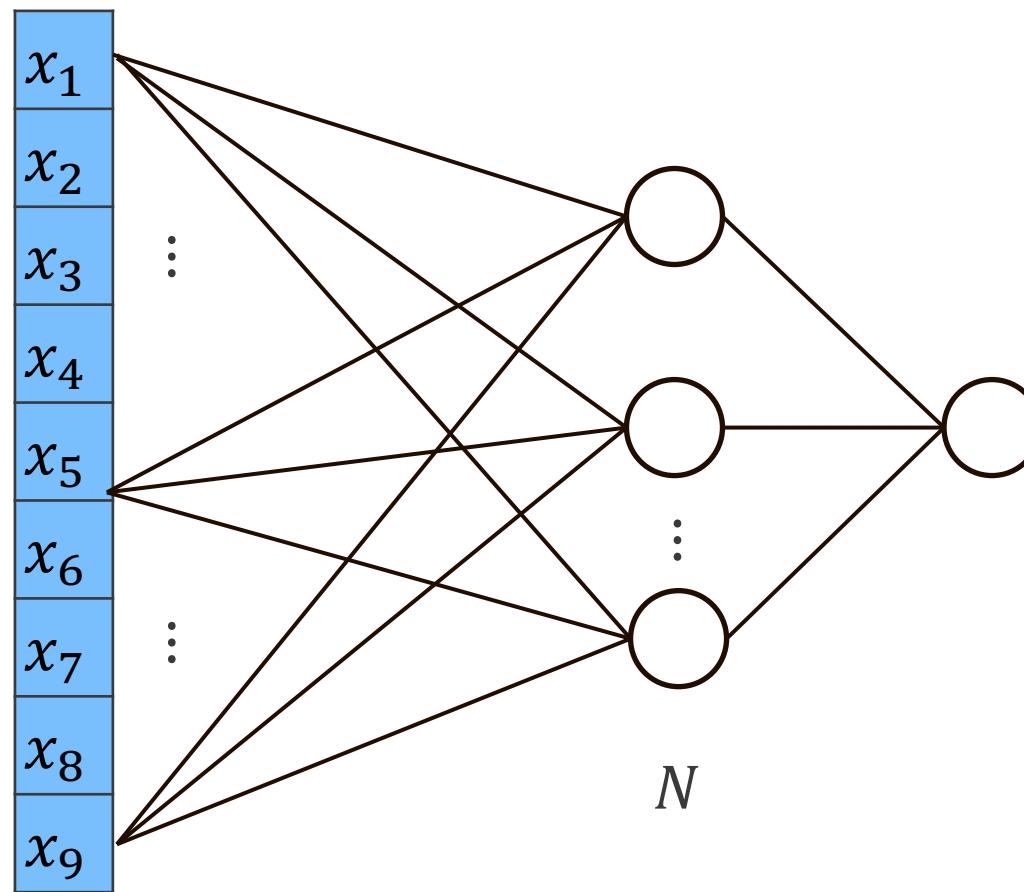
Свёрточные нейронные сети

Выпрямление изображения (flattening)

- ▶ Развернем значения пикселей во всех каналах в вектор



Полносвязная нейронная сеть



Количество весов сети:

$$9 * N + N = 10 * N$$

Свертка изображения

Изображение

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

(3, 3, 1)

*

Ядро

w_1	w_2
w_3	w_4

(2, 2, 1)

Свернутое
изображение

z_1	z_2
z_3	z_4

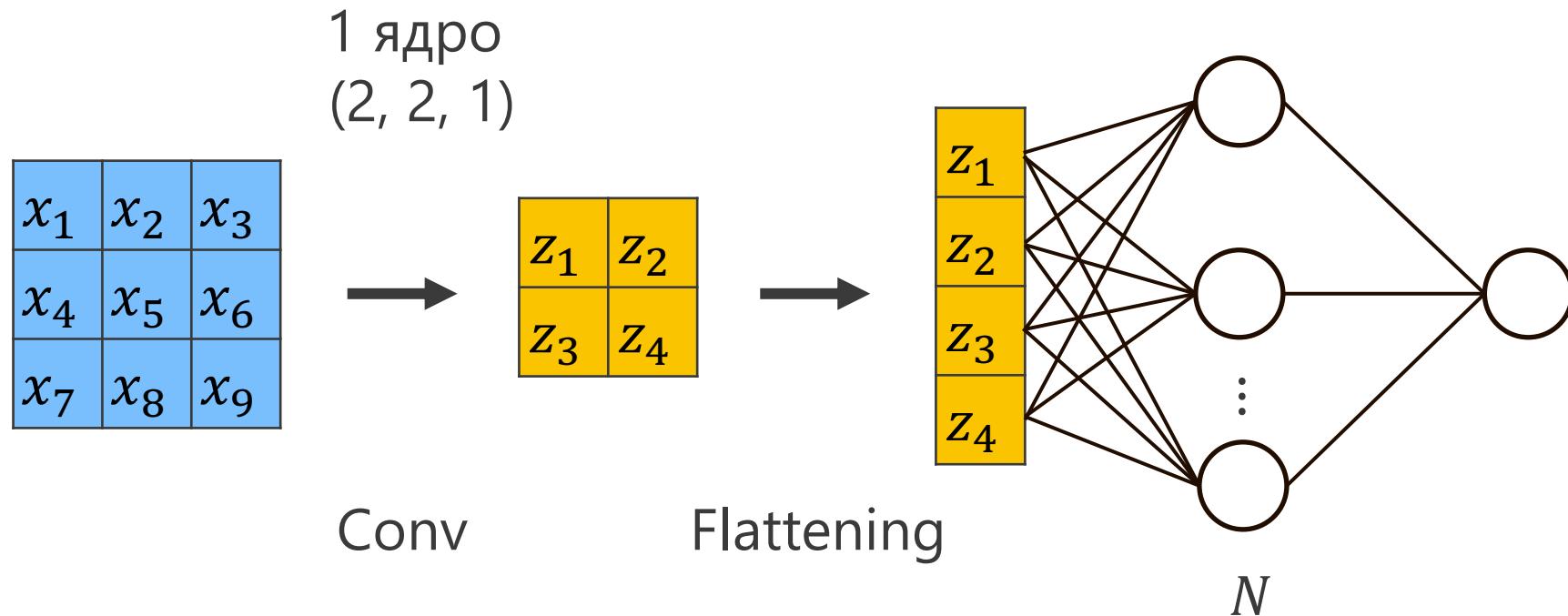
(2, 2, 1)

Ядро свертки как нейрон:

$$z_1 = f(w_1 x_1 + w_2 x_2 + w_3 x_4 + w_4 x_5)$$

Веса w_i получим в процессе обучения

Простая свёрточная сеть



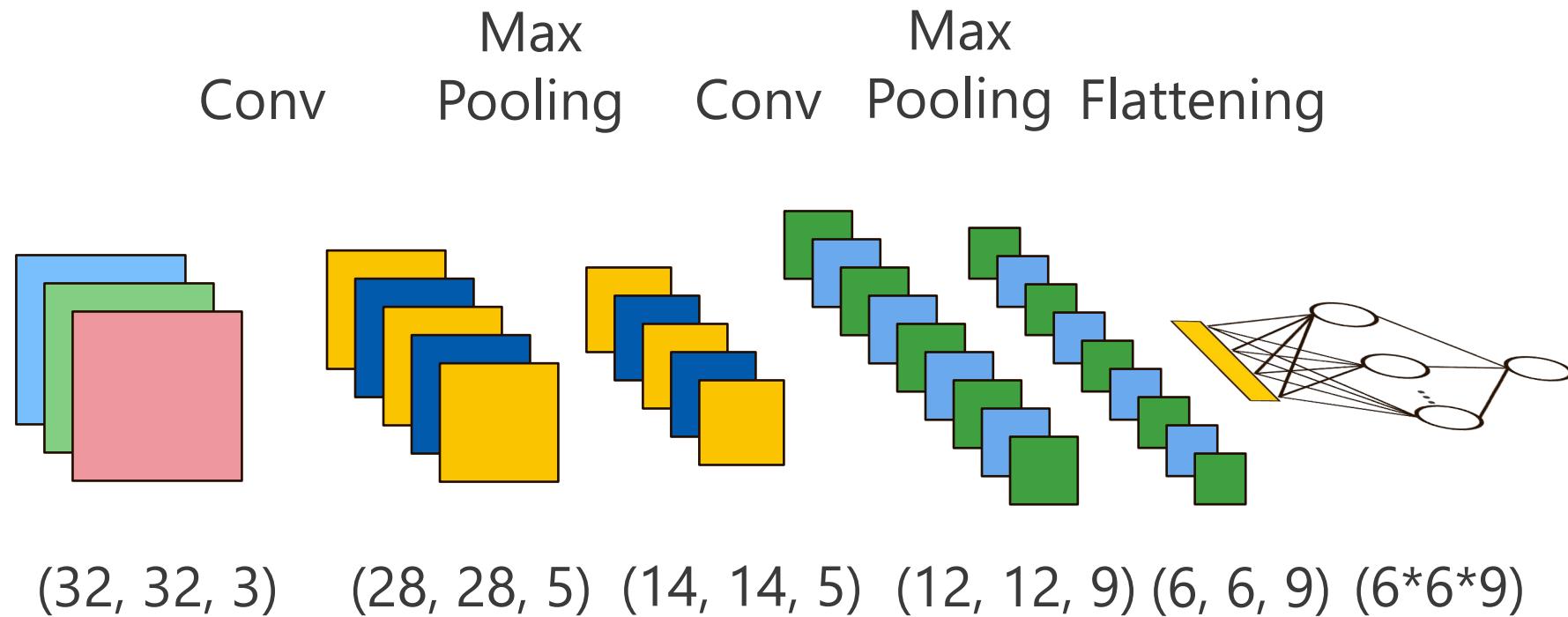
Количество весов сети:

$$4 + 4 * N + N = 5 * N + 4$$

Преимущества сверток

- ▶ Значительно уменьшают количество весов сети
- ▶ Сети быстрее обучаются
- ▶ Требуют меньше данных
- ▶ Достигают лучшего качества

Сверточная сеть



Типичная архитектура

- ▶ Последовательное применение комбинаций
свёрточный слой → нелинейность (ReLU) → pooling
- ▶ Выпрямление (flattening) выхода последней комбинации
- ▶ Серия полно связных слоёв