

Основы глубинного обучения

Лекция 3

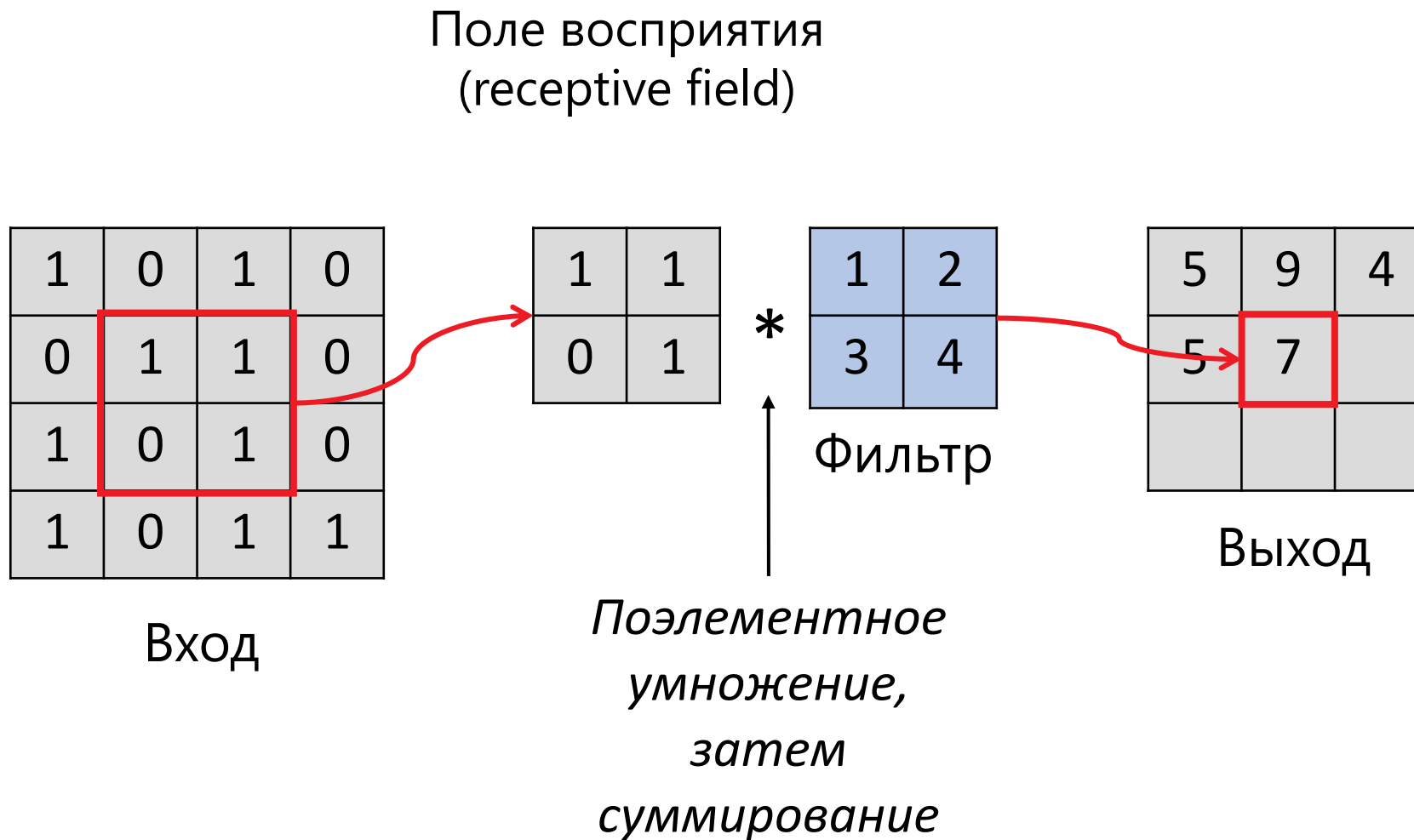
Свёрточные сети

Евгений Соколов

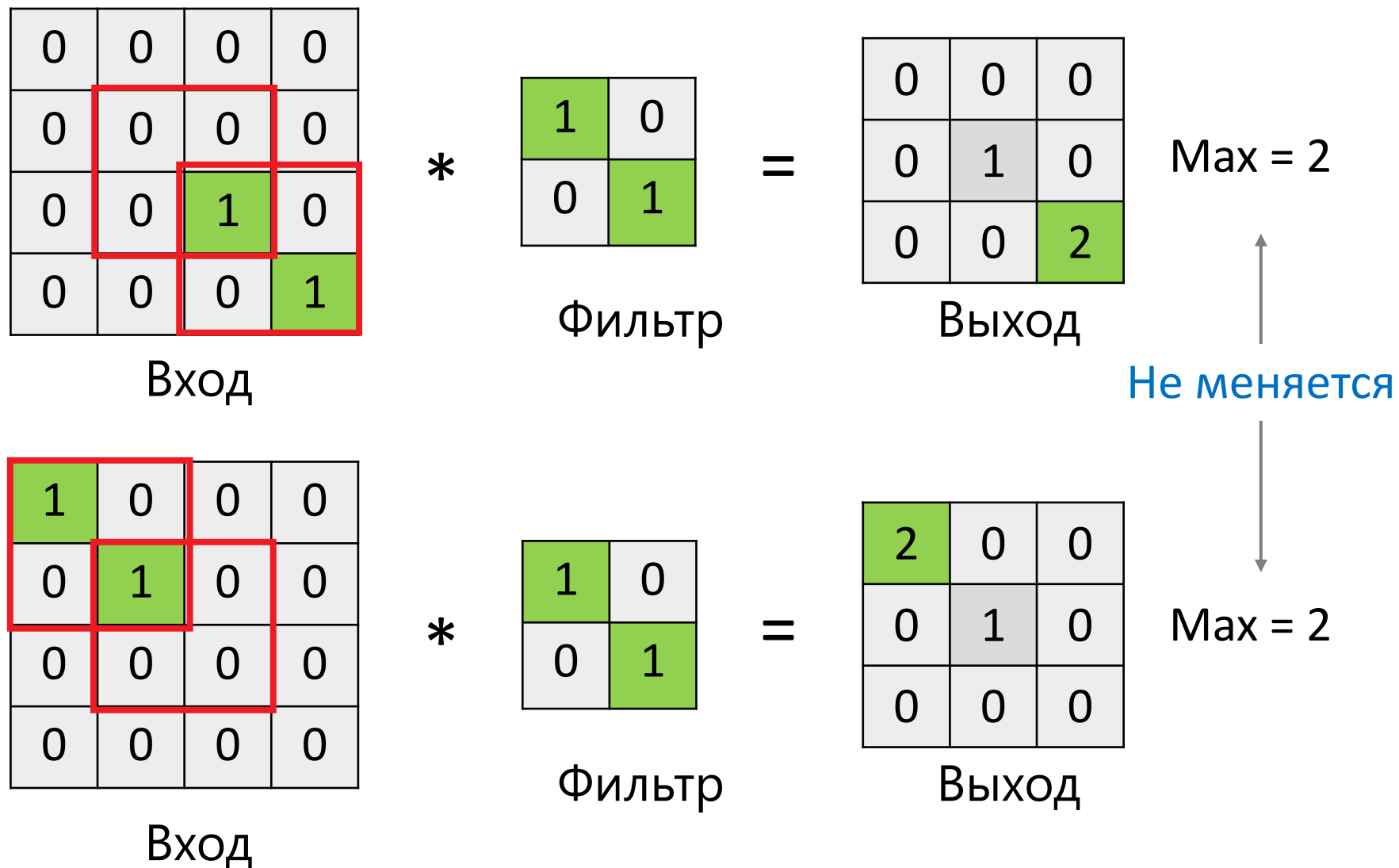
esokolov@hse.ru

НИУ ВШЭ, 2021

Свёртка



Свёртка инвариантна к сдвигам

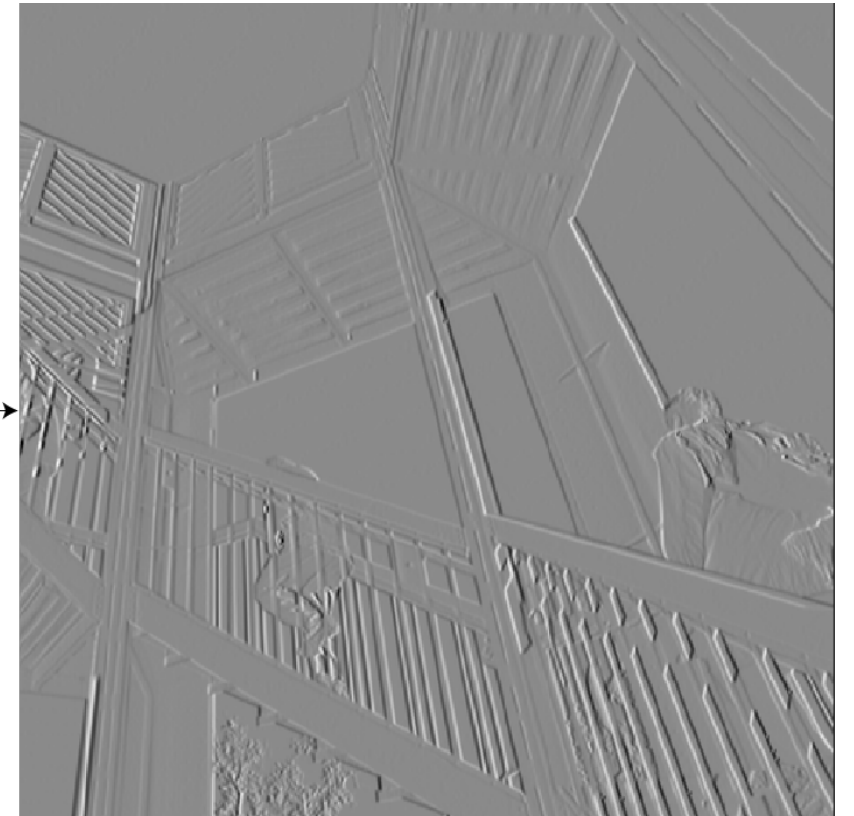


Свёртки в компьютерном зрении



$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

Horizontal Sobel kernel



Свёртка

$$\text{Im}^{out}(x, y) = \sum_{i=-d}^d \sum_{j=-d}^d K(i, j) \text{Im}^{in}(x + i, y + j)$$

Свёртка

$$\text{Im}^{out}(x, y) = \sum_{i=-d}^d \sum_{j=-d}^d K(i, j) \text{Im}^{in}(x + i, y + j)$$

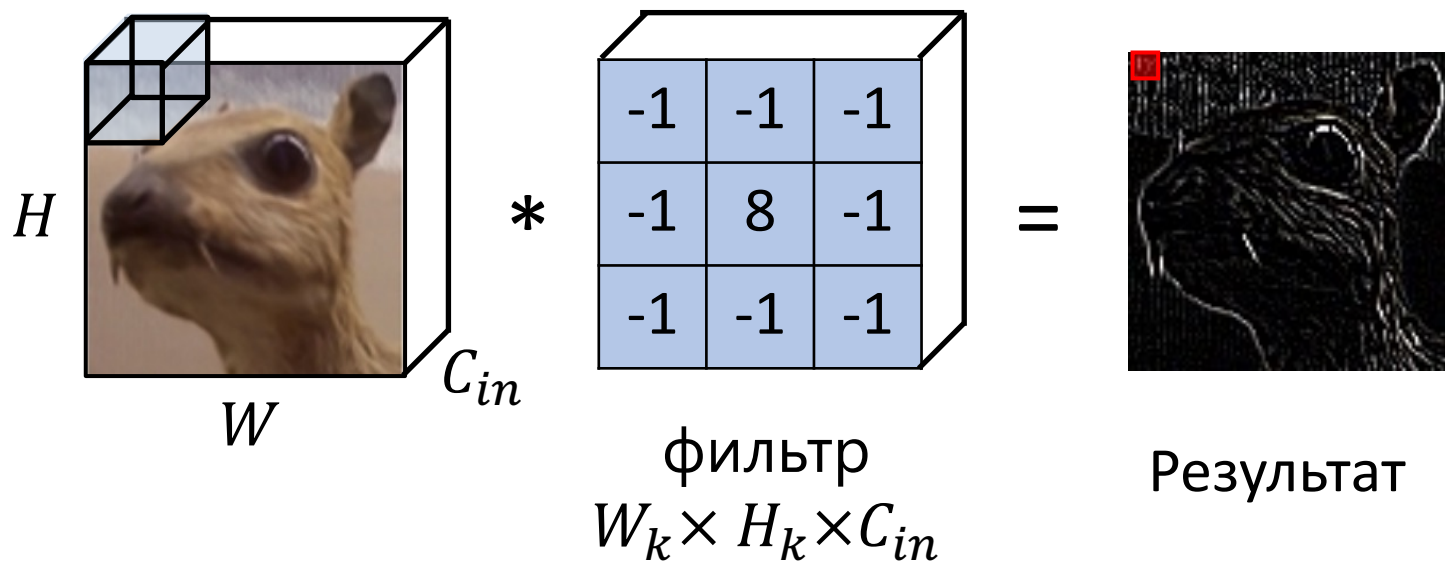
- Пиксель в результирующем изображении зависит только от небольшого участка исходного изображения (local connectivity)
- Веса одни и те же для всех пикселей результирующего изображения (shared weights)

Свёртка

- Обычно исходное изображение цветное!
- Это означает, что в нём несколько каналов (R, G, B)
- Учтём в формуле:

$$\text{Im}^{out}(x, y) = \sum_{i=-d}^d \sum_{j=-d}^d \sum_{c=1}^C K(i, j, c) \text{Im}^{in}(x + i, y + j, c)$$

Свёртка



Свёрточный слой

- Одна свёртка выделяет конкретный паттерн на изображении
- Нам интересно искать много паттернов
- Сделаем результат трёхмерным:

$$\text{Im}^{out}(x, y, t) = \sum_{i=-d}^d \sum_{j=-d}^d \sum_{c=1}^C K_t(i, j, c) \text{Im}^{in}(x + i, y + j, c)$$

Число параметров

$$\text{Im}^{out}(x, y, t) = \sum_{i=-d}^d \sum_{j=-d}^d \sum_{c=1}^C K_t(i, j, c) \text{Im}^{in}(x + i, y + j, c)$$

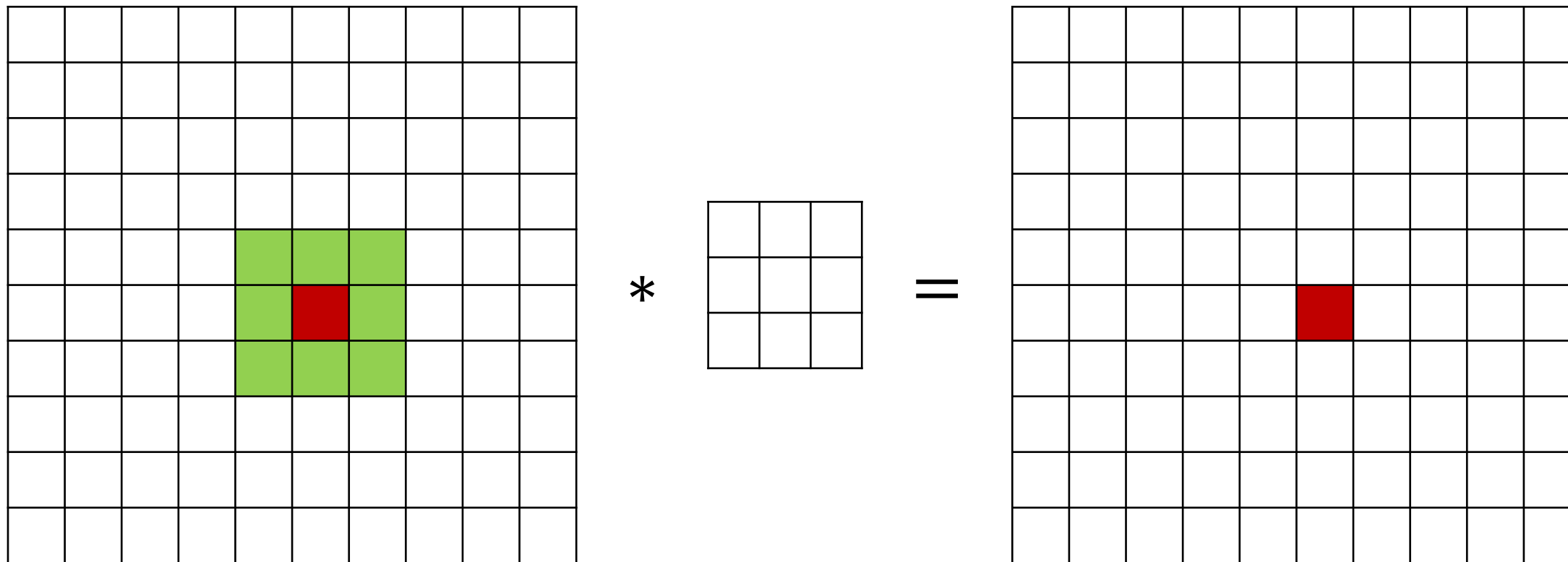
- Обучается только фильтр
- $(2d + 1) * C * T$ обучаемых параметров
- Как сделать из этого модель — обсудим позже

Receptive field

Receptive field

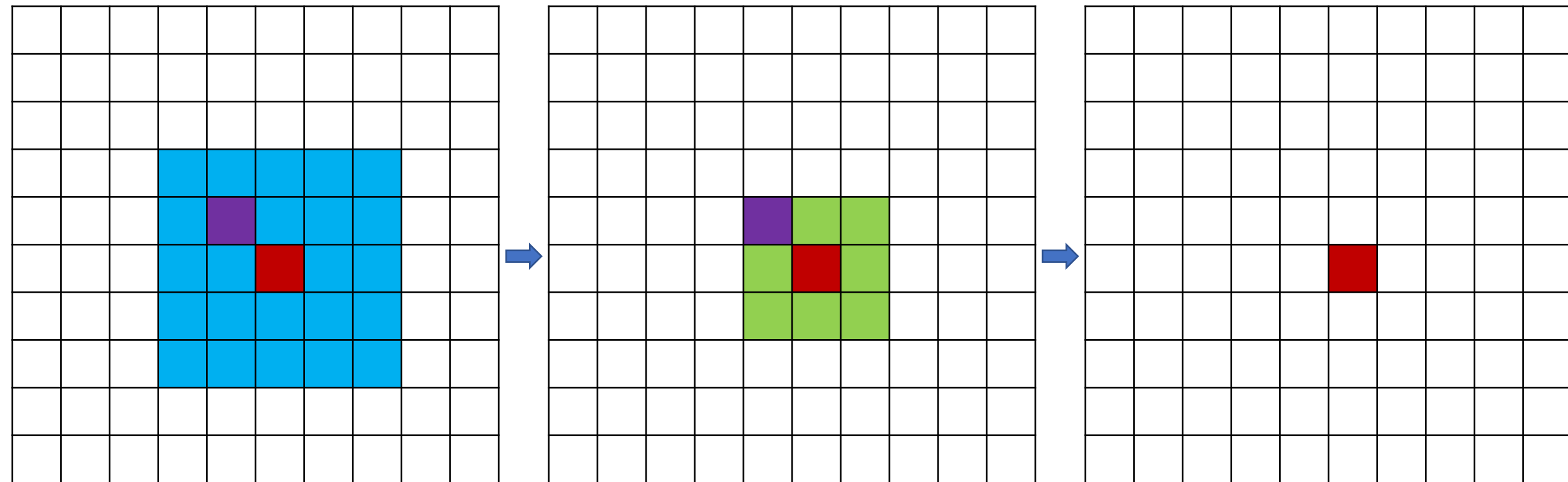
- Возьмём пиксель в итоговом изображении (после свёрточных слоёв)
- От какой части входного изображения зависит значение в этом пикселе?

Receptive field



Поле восприятия: 3 x 3

Receptive field



Поле восприятия: 5 x 5

Receptive field

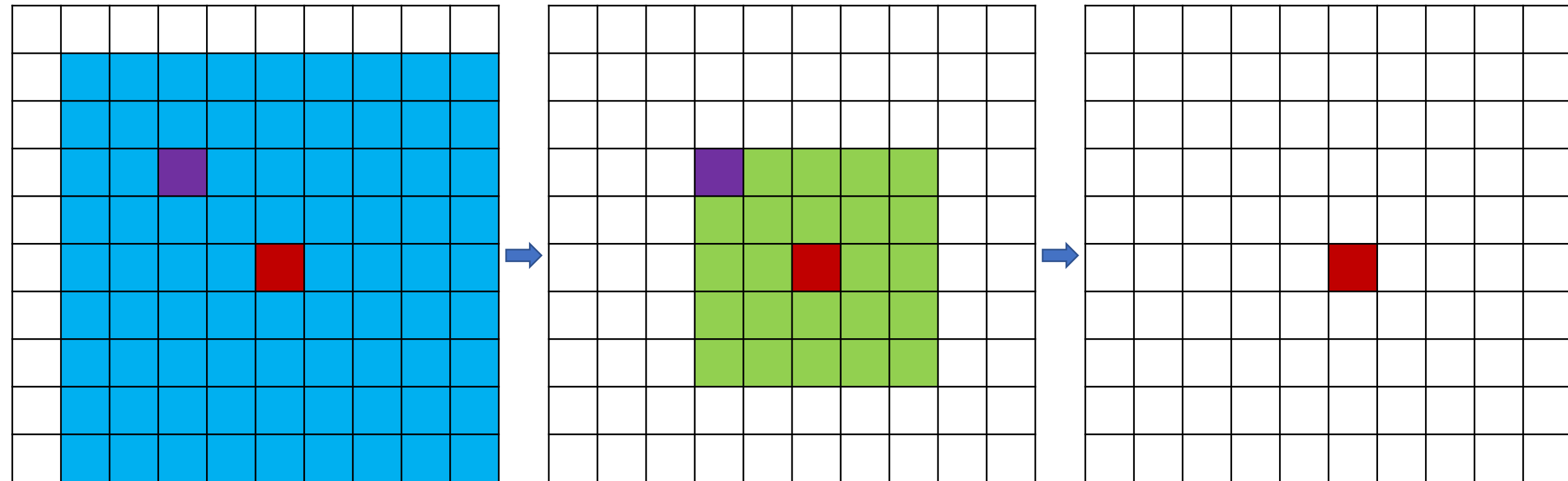
Поле восприятия для свёртки 3 x 3:

- После 1 свёрточного слоя: 3 x 3
- После 2 свёрточных слоев: 5 x 5
- После 3 свёрточных слоёв: 7 x 7

Receptive field

Поле восприятия для свёртки 5 x 5:

Receptive field



Поле восприятия: 5 x 5

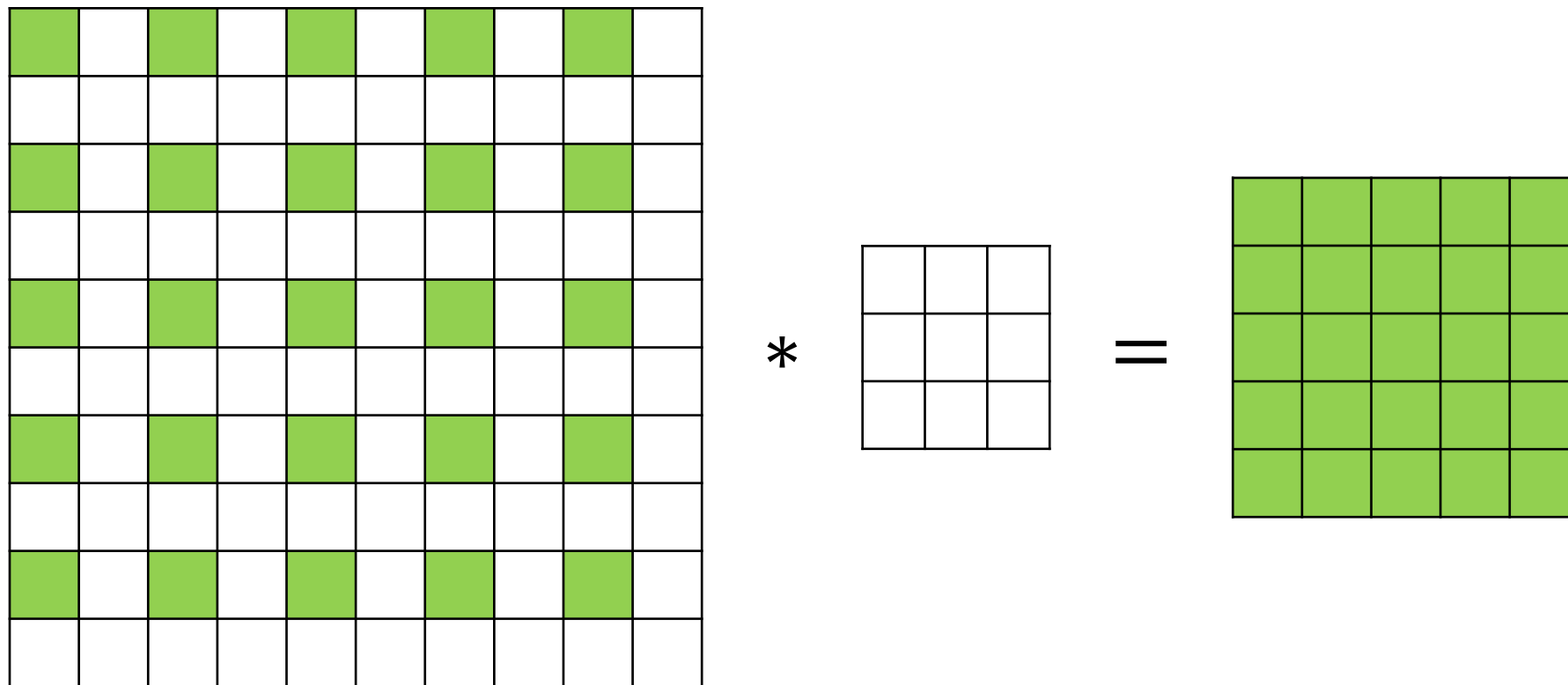
Receptive field

Поле восприятия для свёртки 5 x 5:

- После 1 свёрточного слоя: 5 x 5
- После 2 свёрточных слоев: 9 x 9
- После 3 свёрточных слоёв: 13 x 13

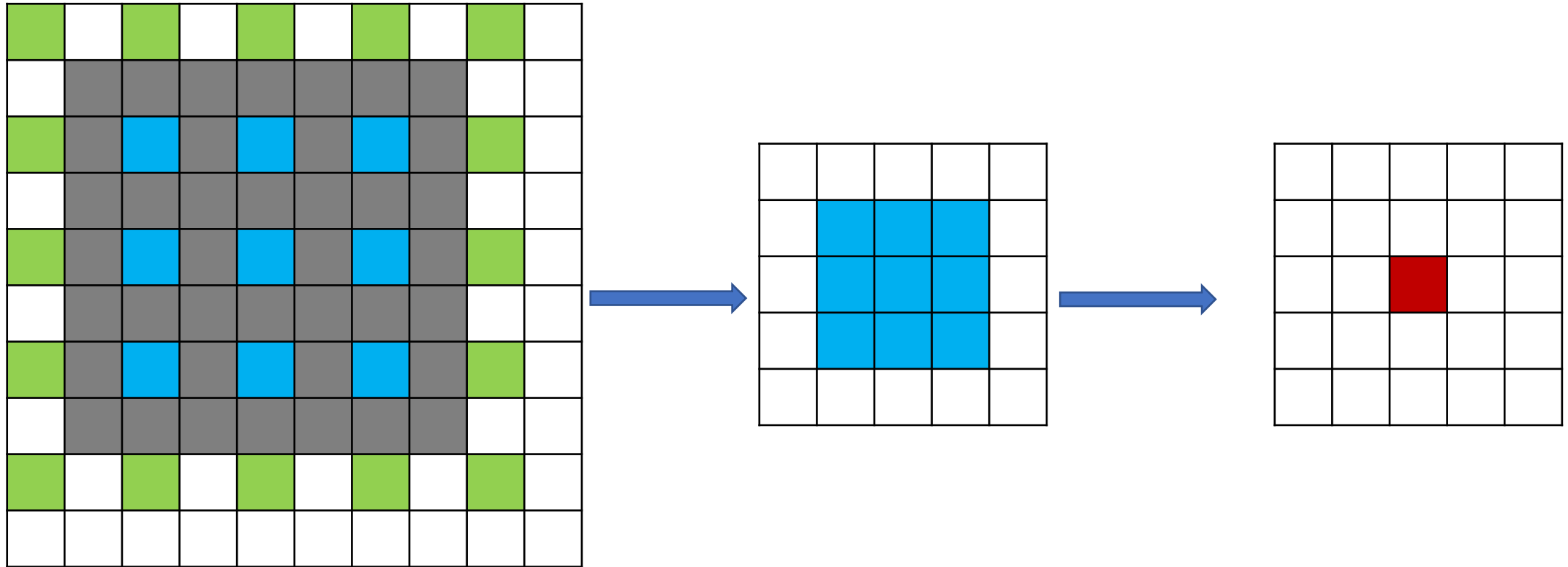
Нужно очень много слоёв, если изображение размера 512 x 512

Свёртки с пропусками (strides)



$$s = 2$$

Свёртки с пропусками (strides)



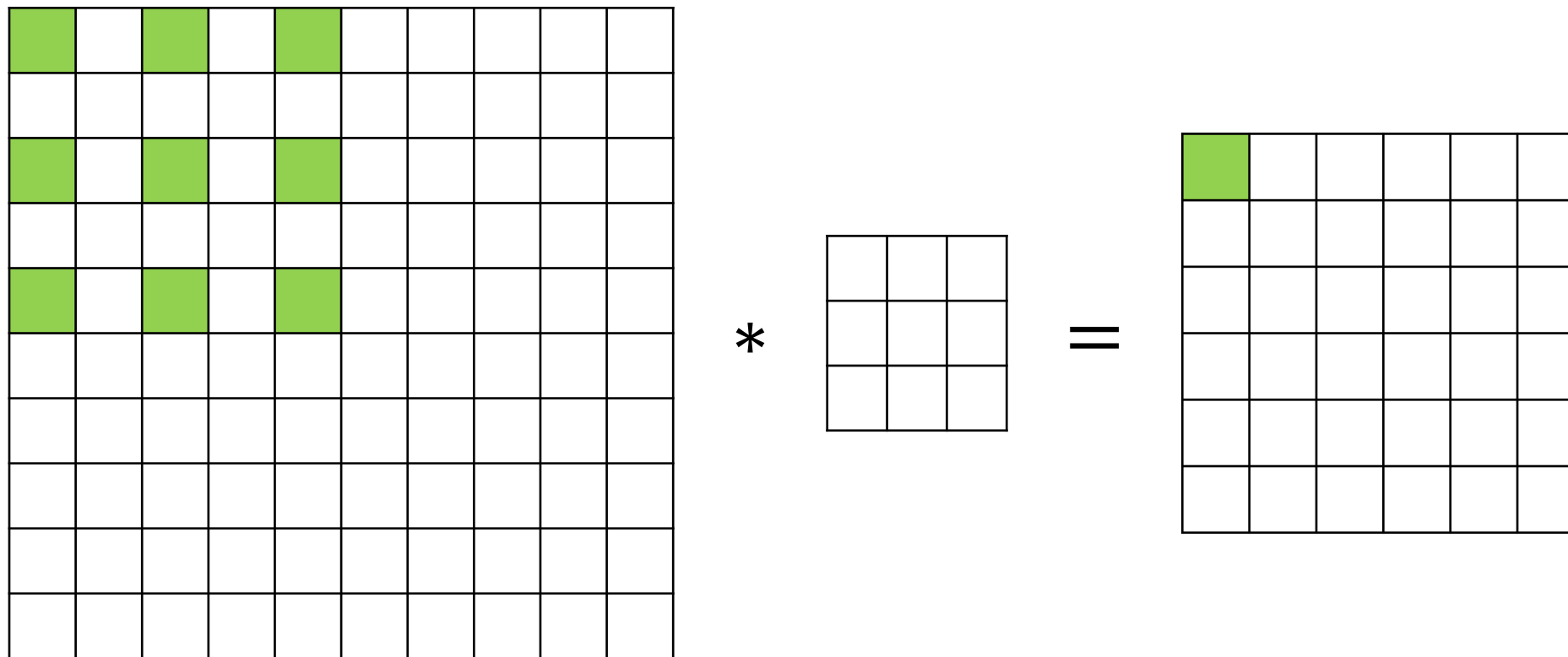
Поле восприятия: 7 x 7

Свёртки с пропусками (strides)

Подробности про подсчёт размера поля:

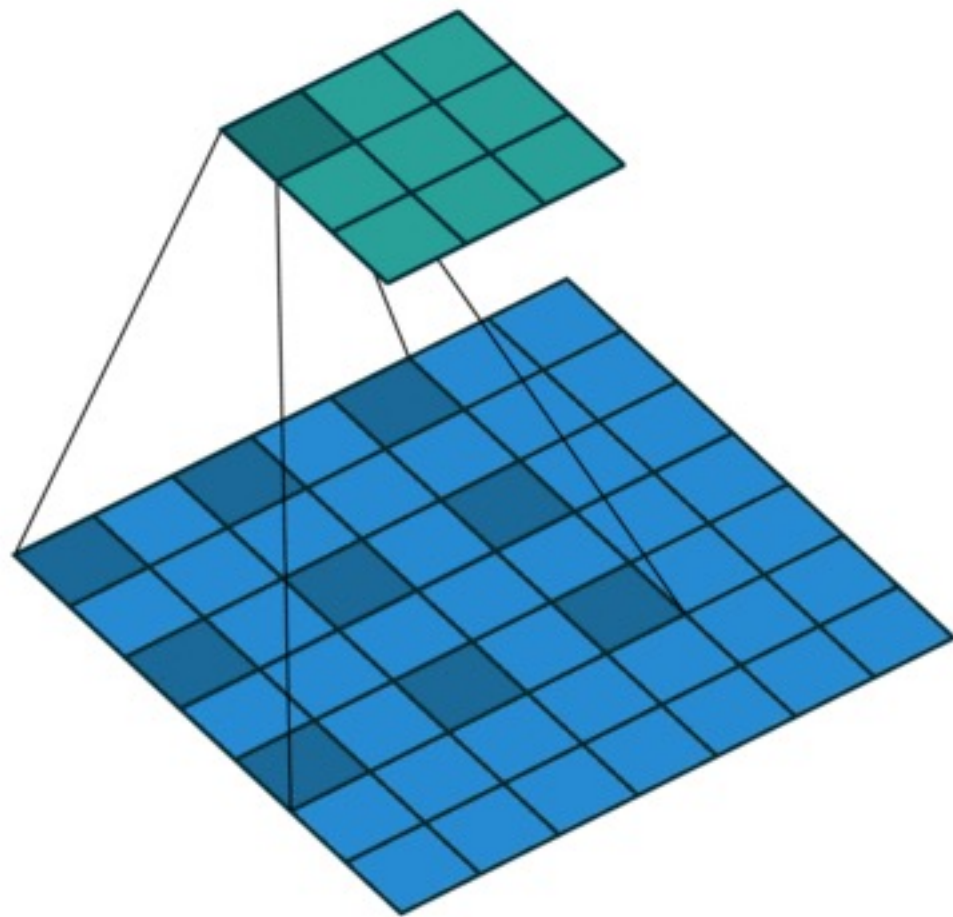
<https://distill.pub/2019/computing-receptive-fields/>

Dilated convolutions («раздутые» свёртки)

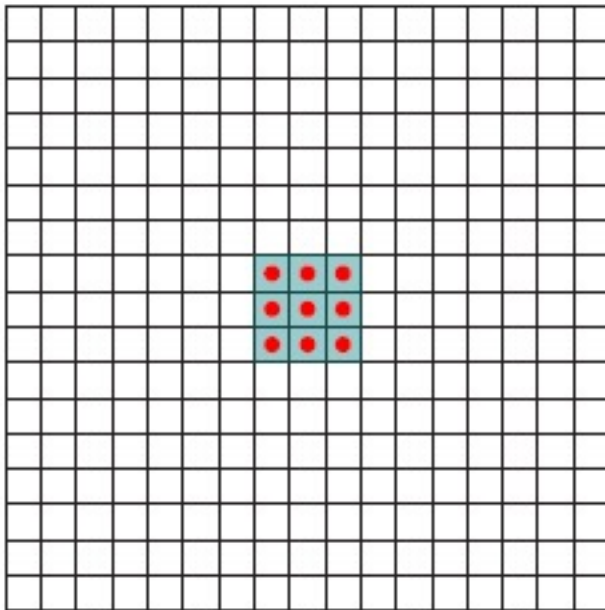


$$l = 2$$

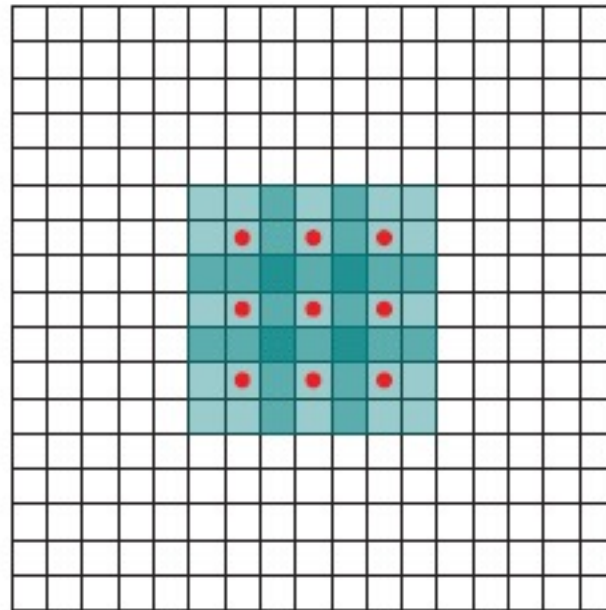
Dilated convolutions («раздутые» свёртки)



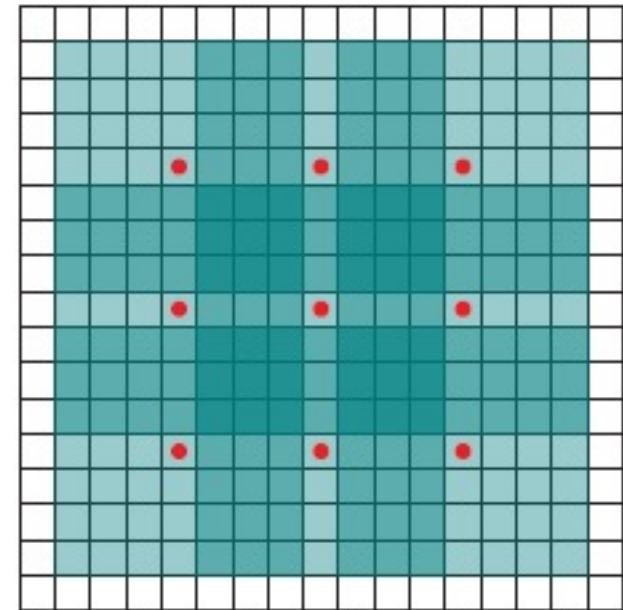
Dilated convolutions («раздутые» свёртки)



$l = 1$



$l = 2$



$l = 4$

Pooling

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



1	3	2

Max-pooling с фильтром 2x2

Pooling

- Разбивает изображение на участки $n \times m$ и считает некоторую статистику в каждом участке (обычно максимум)
- Существенно сокращает размер изображения (значит, увеличивает поле восприятия следующих слоёв)
- Не имеет параметров

Зачем это всё?

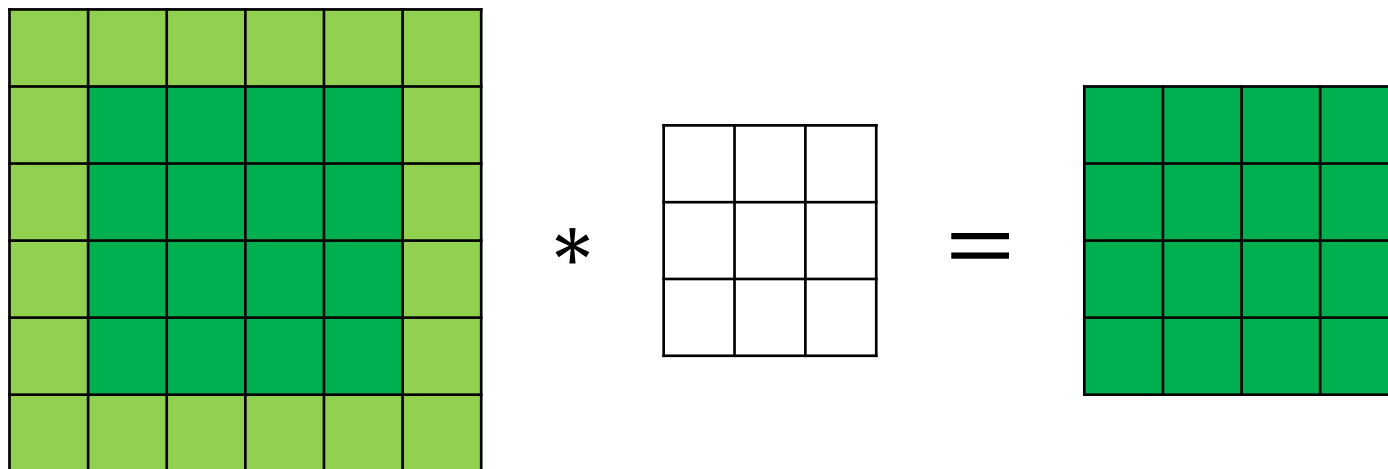
- Важно следить за тем, чтобы последние свёрточные слои имели размер поля восприятия, сравнимый со всей картинкой

Padding

Свёртки

- Если применять свёртку по формуле, то выходное изображение будет меньше входного

Свёртки



Valid mode

- При честном подсчёте свёрток пиксели на краях не оказывают большого влияния на результат

Не увидим, что фильтр имеет хороший отклик при помещении центра в этот пиксель

0	1				
1	1				

*

0	0	1
0	0	1
1	1	1

Zero padding

0	0	0	0	0	0	0	0
0							0
0							0
0							0
0							0
0							0
0							0
0	0	0	0	0	0	0	0

*

[illegible]

Zero padding

- Добавляем по границам нули так, чтобы посчитанная после этого свёртка в `valid mode` давала изображение такого же размера, как исходное
- Есть риск, что модель научится понимать, где на изображении края — можем потерять инвариантность

Reflection padding

[illegible]

*

[illegible]

Reflection padding

- Не получится легко находить края изображения
- Но теперь модель может начать находить зеркальные отражения и подбирать фильтры под них

Replication padding

[illegible]

*

[illegible]

Replication padding

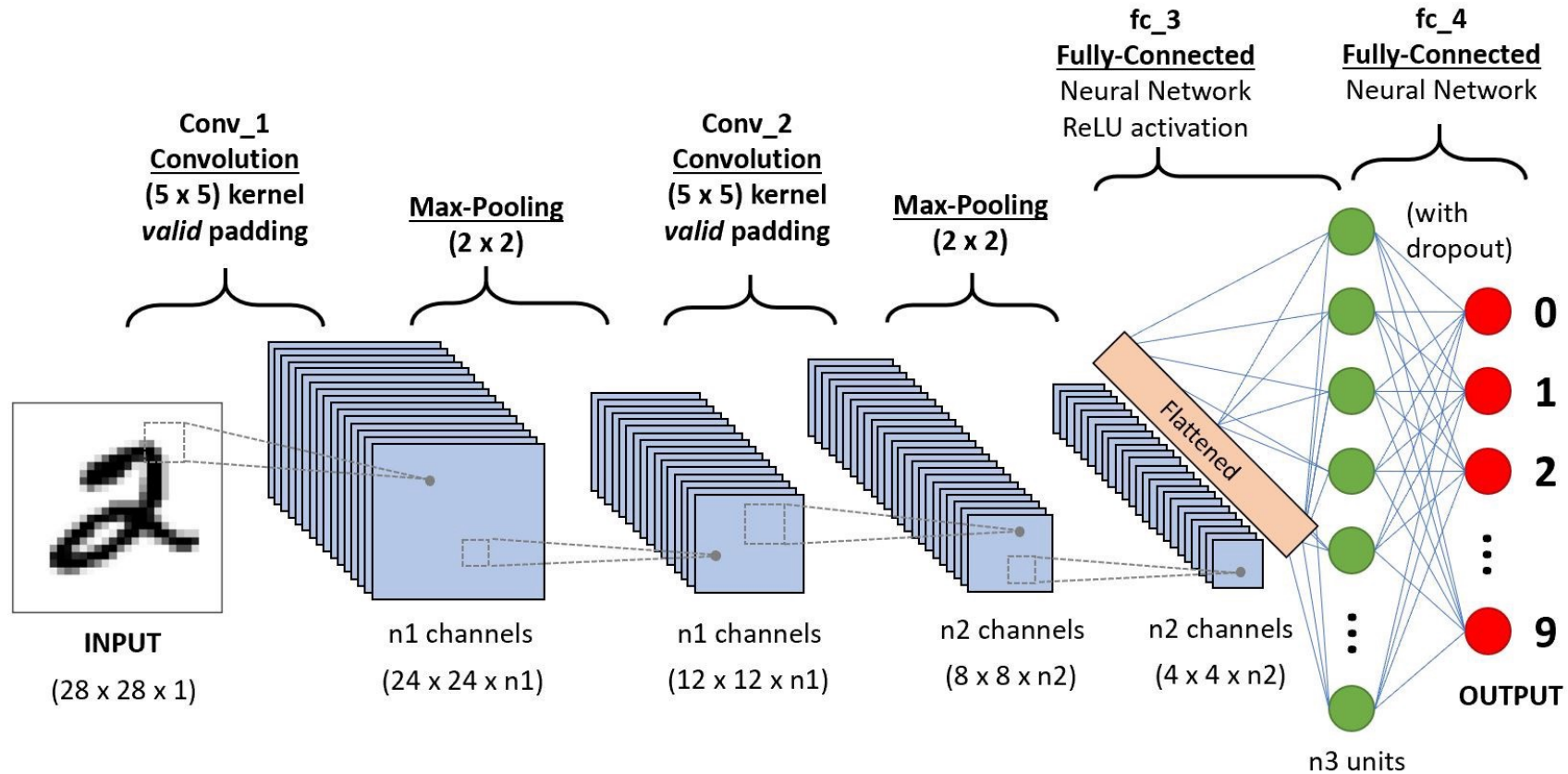
- Пиксель на границе равен ближайшему пикселю из изображения
- Модель всё ещё может настроиться под паттерны, которые возникают из-за такого паддинга

Резюме

- Паддинг позволяет контролировать размер выходных изображений
- Паддинг позволяет учитывать даже объекты на краях
- Разные типа паддингов допускают разные способы переобучения под края

Структура свёрточных сетей

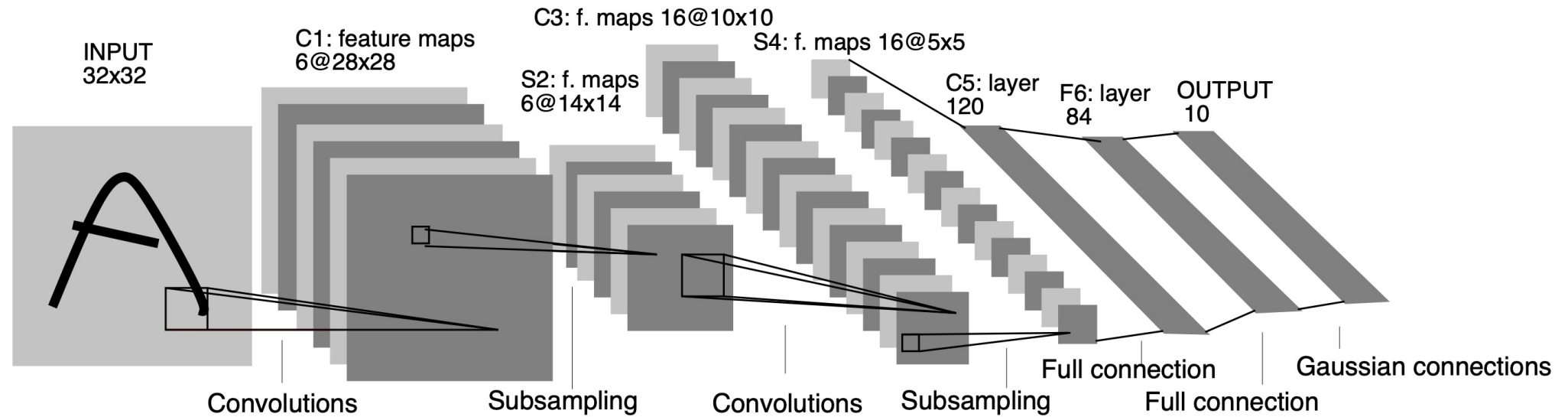
Типичная архитектура



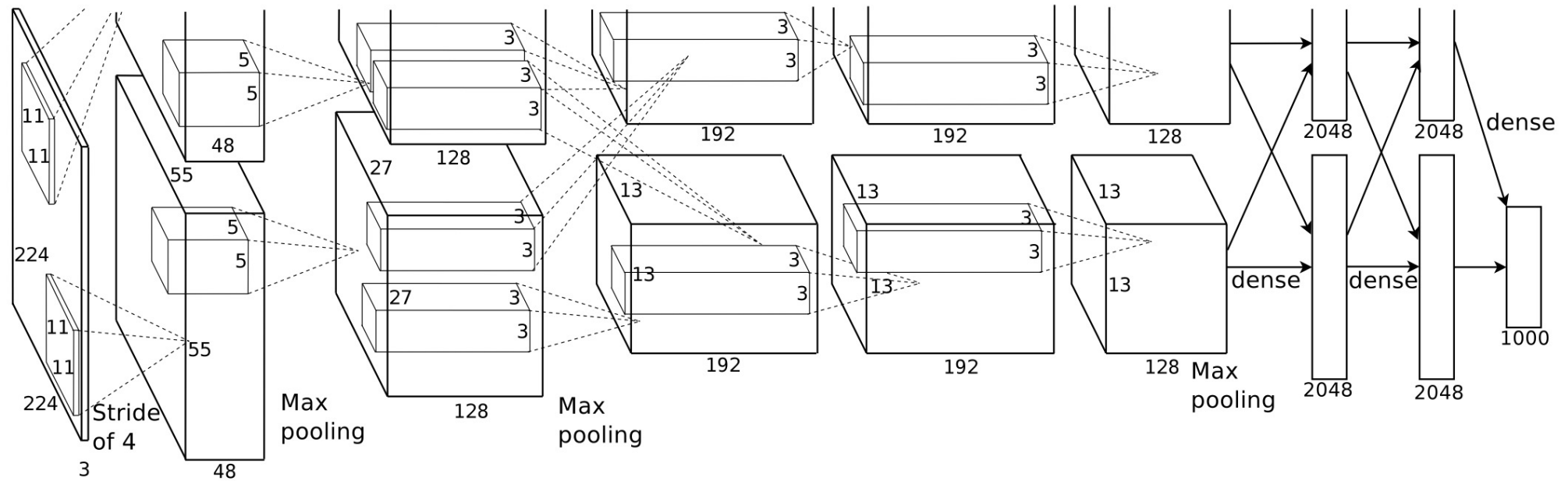
Типичная архитектура

- Последовательное применение комбинаций вида «свёрточный слой -> нелинейность -> pooling» или «свёрточный слой -> нелинейность»
- Выпрямление (flattening) выхода очередного слоя
- Серия полносвязных слоёв

LeNet



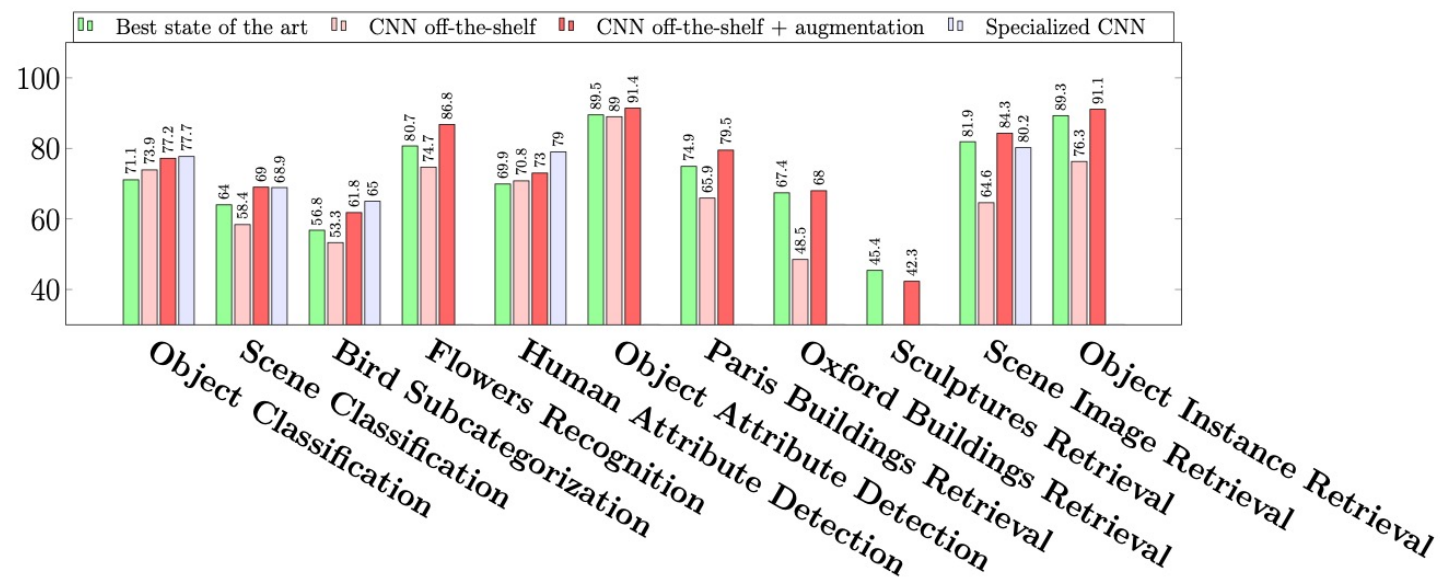
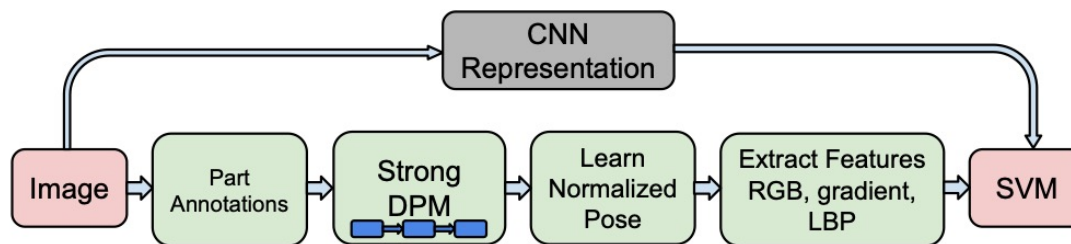
AlexNet



Представления с последних слоёв

- Важное наблюдение: выходы полносвязных слоёв являются хорошими признаковыми описаниями изображений
- Полезны во многих задачах
- Например, поиск похожих изображений

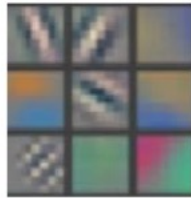
Представления с последних слоёв



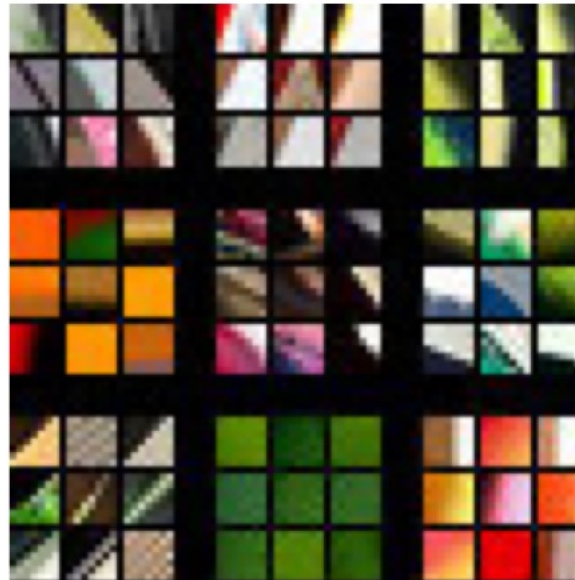
Представления с последних слоёв

- Не интерпретируется (в отличие от классического компьютерного зрения)
- По смыслу — «индикаторы» наличия каких-то паттернов

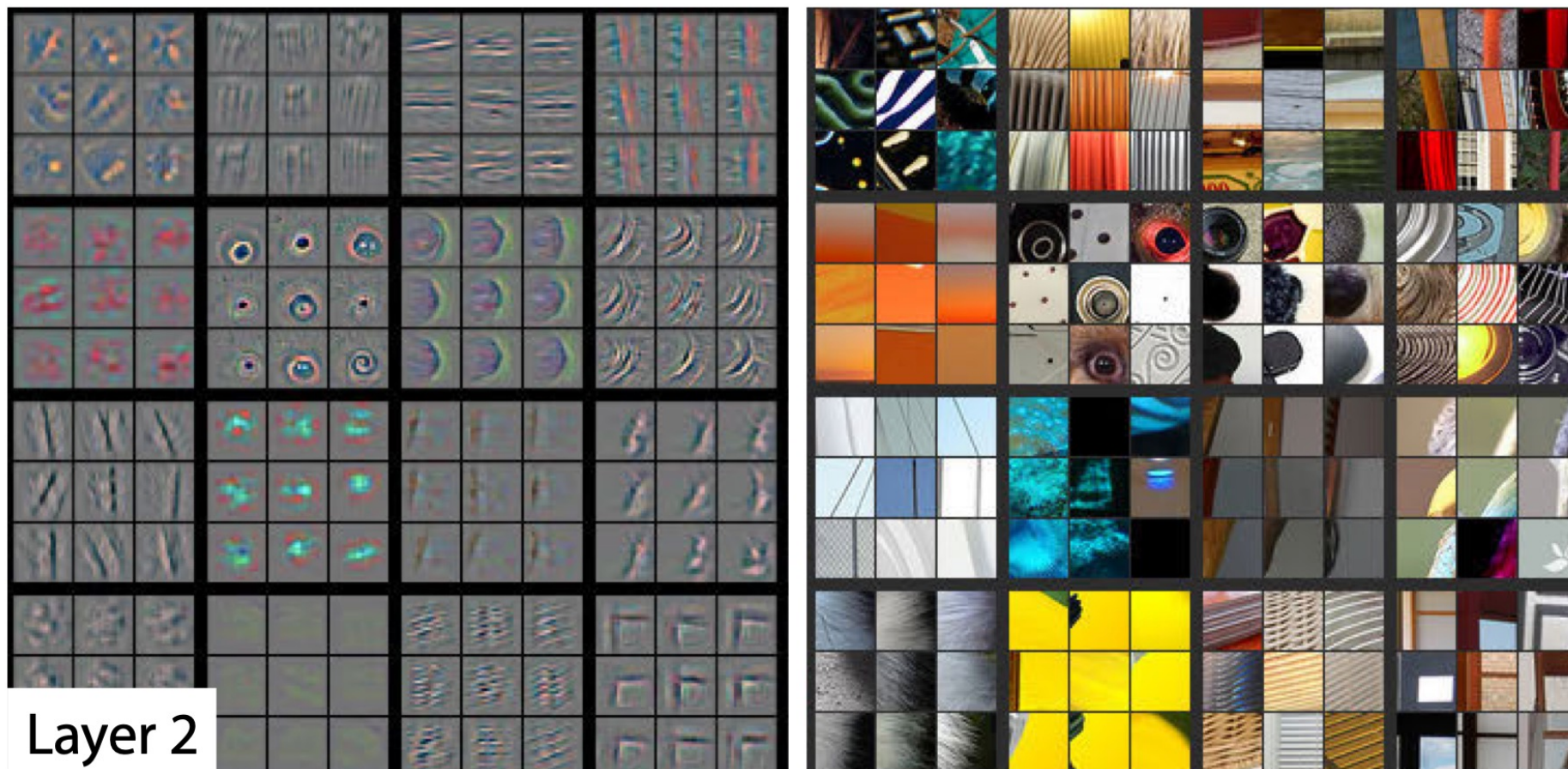
Представления с последних слоёв



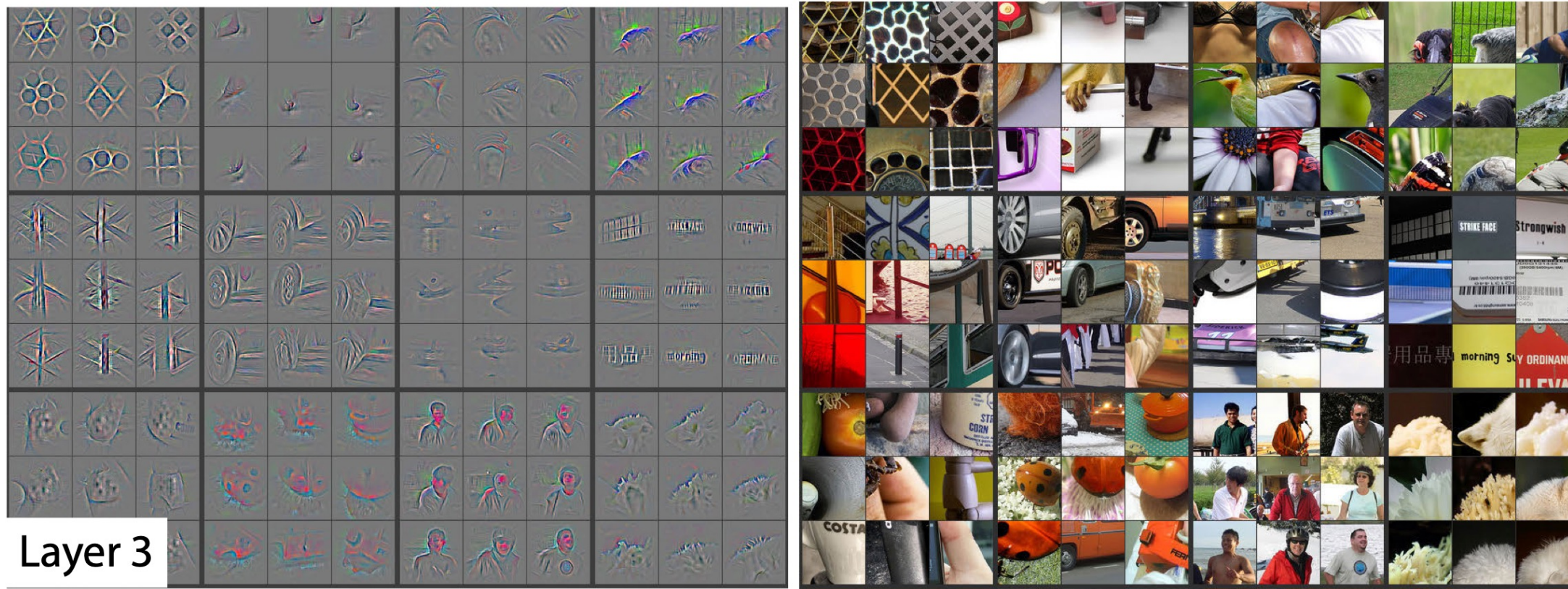
Layer 1



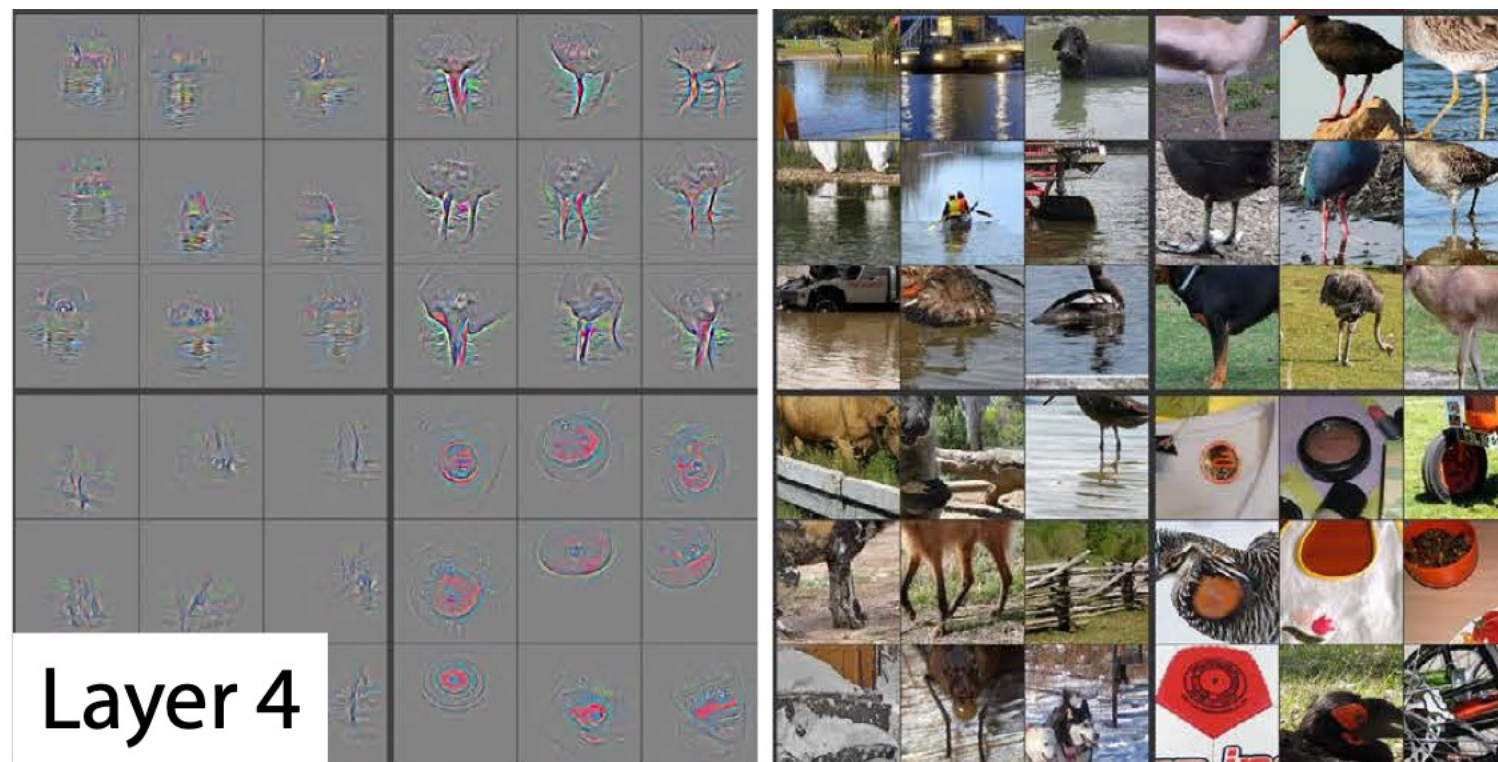
Представления с последних слоёв



Представления с последних слоёв



Представления с последних слоёв



Представления с последних слоёв

