

# Глубинное обучение

## Свёрточные нейронные сети

Даниил Водолазский

ВШЭ

21 июля 2021 г.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Содержание

## 1 Компьютерное зрение

Что это такое

История развития CV

Представление изображений в компьютере

FFNN vs. CNN

## 2 Свёртки

Введение

Двумерная свёртка

Обработка краёв

Градиенты

Свёрточный слой

Рецептивное поле

## 3 Примеры свёрточных сетей

## 4 Интерпретация

## 5 Что узнали

## Определение

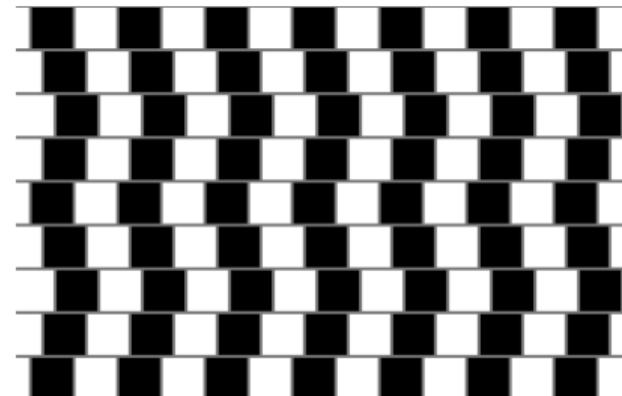
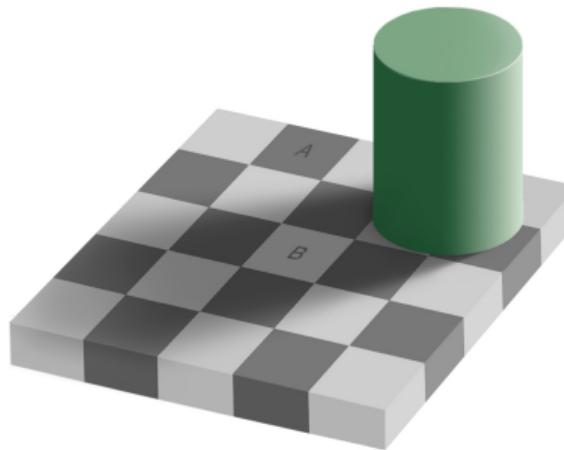
**Цель компьютерного зрения** — дать компьютерам схожее с человеческим визуальное восприятие мира.

<sup>1</sup>[https://en.wikipedia.org/wiki/Optical\\_illusion](https://en.wikipedia.org/wiki/Optical_illusion)

## Определение

**Цель компьютерного зрения** — дать компьютерам схожее с человеческим визуальное восприятие мира.

В общем случае компьютерное зрение **не работает**<sup>1</sup>.



<sup>1</sup>[https://en.wikipedia.org/wiki/Optical\\_illusion](https://en.wikipedia.org/wiki/Optical_illusion)

# История развития CV: изучение зрительной коры (1959)

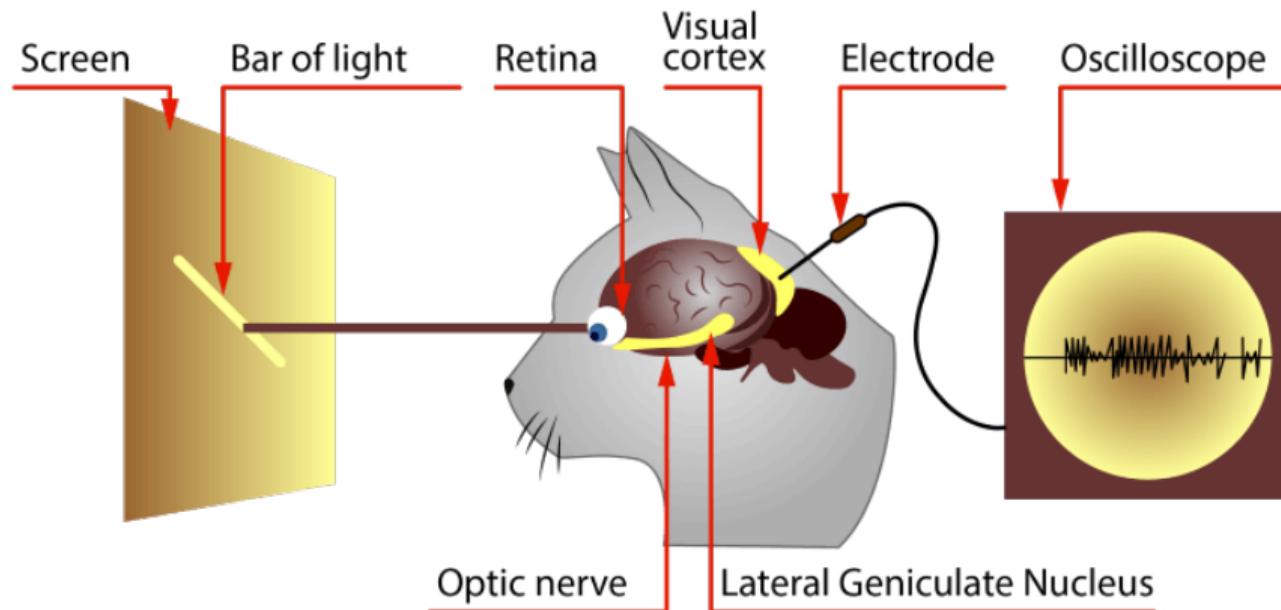


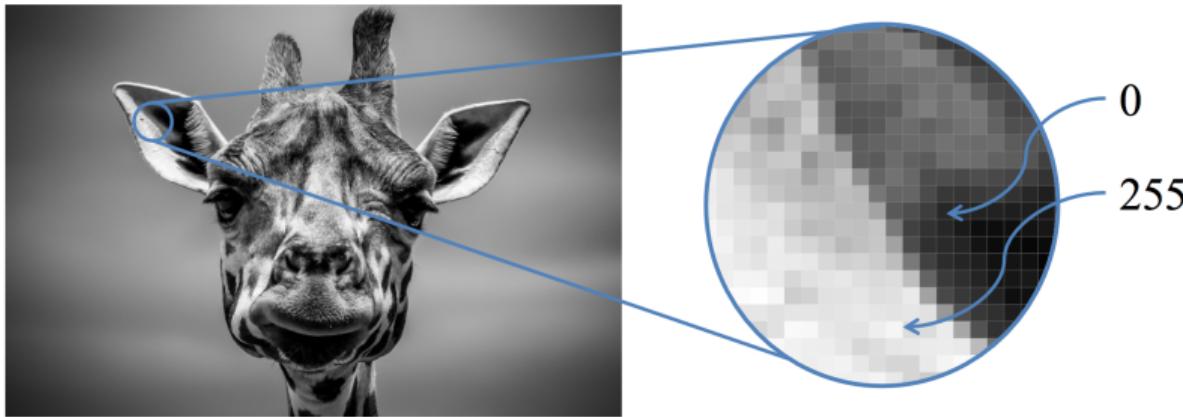
Схема эксперимента нейрофизиологов Дэвида Хубеля и Торстена Визеля<sup>2</sup>

<sup>2</sup><https://distillery.com/blog/implementing-human-brain-exploring-potential-convolutional-neural-networks/>

- 
- A vertical timeline chart showing the evolution of visual processing research from 1959 to 1999. Each year is marked by a blue dot on a vertical axis, followed by a horizontal line and a text description of a significant achievement in the field.
- 1959 Изучение зрительной коры
  - 1963 Реконструкция формы 3D-моделей
  - 1966 The Summer Vision Project
  - 1981 Оценка optical flow
  - 1986 Детектор границ Кэнни
  - 1996 Скорость работы зрительной коры человека 150 мс
  - 1998 LeCun et al. CNN для классификации MNIST
  - 1999 Single object recognition - Scale Invariant Feature Transform (SIFT)

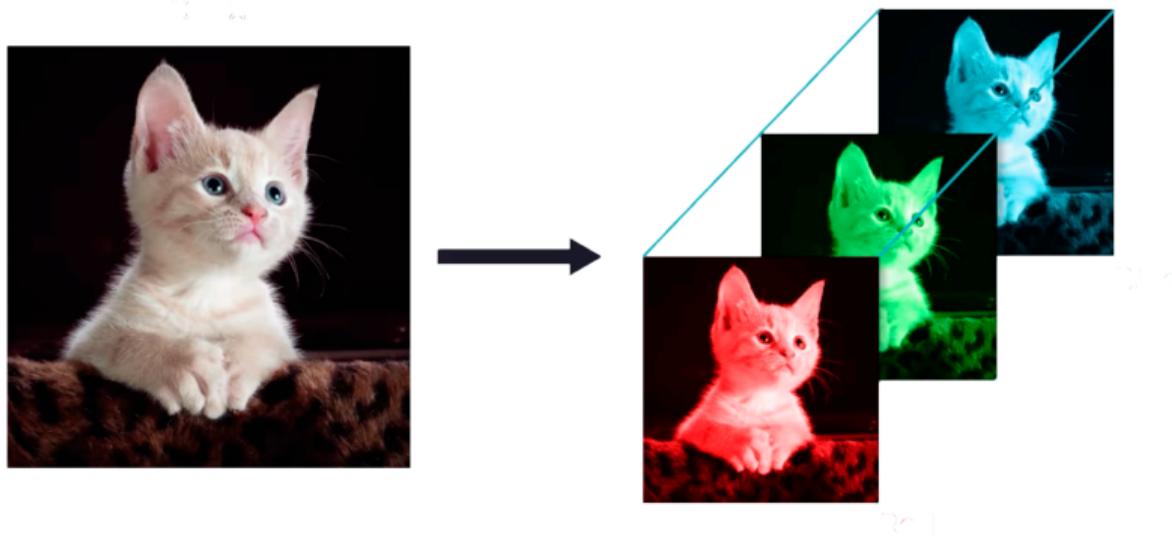
# Представление изображений в компьютере

- Каждая картинка — это матрица  $h \times w$  из пикселей.
- Каждый пиксель обладает яркостью по шкале от 0 до 255.

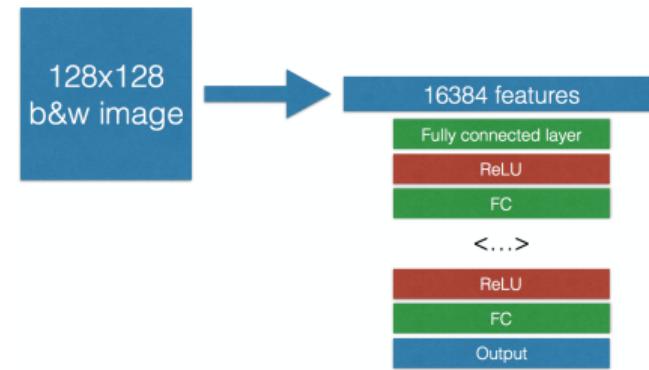
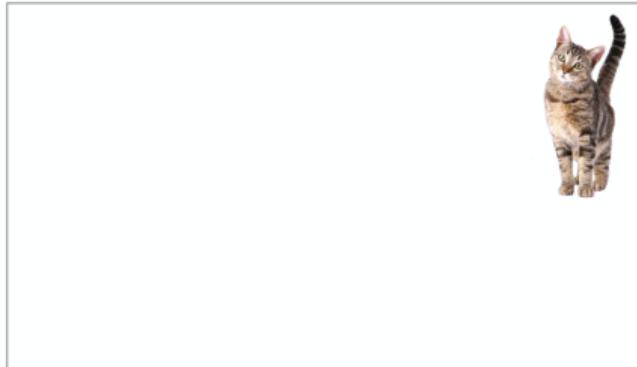


# Представление изображений в компьютере

- Каждая картинка — это матрица  $h \times w$  из пикселей.
- Каждый пиксель обладает яркостью по шкале от 0 до 255 (8-битные числа).
- Цветное изображение имеет три канала пикселей: красный, зелёный и синий (RGB), размерность изображения  $h \times w \times 3$ .



# Проблемы «обычных» сетей»



- Очень много весов.
- Теряется информация о взаимном расположении пикселей.
- Изображение в разных местах картинки даёт разные веса.

- Хотим, чтобы информация не терялась.
- Хотим, чтобы сетка была устойчива к положению картинки.
- Хотим одинаковые веса ⇒ **свёртка**.

# Содержание

## 1 Компьютерное зрение

Что это такое

История развития CV

Представление изображений в компьютере

FFNN vs. CNN

## 2 Свёртки

Введение

Двумерная свёртка

Обработка краёв

Градиенты

Свёрточный слой

Рецептивное поле

## 3 Примеры свёрточных сетей

## 4 Интерпретация

## 5 Что узнали

# Введение

Вспомним, как умножаются многочлены.

$$A(x) = \sum_{k=0}^n a_k x^k = a_n x^n + a^{n-1} x^{n-1} + \cdots + a_1 x + a_0;$$

$$B(x) = \sum_{l=0}^m b_l x^l = b_m x^m + b^{m-1} x^{m-1} + \cdots + b_1 x + b_0;$$

$$C(x) = A(x)B(x) = \sum_{s=0}^{m+n} c_s x^s, \quad c_s = \sum_{t=0}^s a_t b_{s-t}.$$

$\sum_{t=0}^s a_t b_{s-t}$  — это не что иное, как свёртка (одномерная).

$$A = [a_n, a_{n-1}, \dots, a_1, a_0];$$

$$B = [b_m, b_{m-1}, \dots, b_1, b_0];$$

$$C = A * B.$$

# Двумерная свёртка

## Определение

Свёртка в общем виде задается следующим выражением:

$$g(x, y) = \omega * f(x, y) = \sum_{dx=-a}^a \sum_{dy=-b}^b \omega(dx, dy) f(x - dx, y - dy),$$

где  $g(x, y)$  — результат свёртки,  $f(x, y)$  — исходное изображение,  $\omega$  — **фильтр / ядро свёртки / матрица свёртки**.

Свёртки — один из важнейших инструментов в обработке изображений.

# Двумерная свёртка

3 <sub>0</sub>	3 <sub>1</sub>	2 <sub>2</sub>	1	0
0 <sub>2</sub>	0 <sub>2</sub>	1 <sub>0</sub>	3	1
3 <sub>0</sub>	1 <sub>1</sub>	2 <sub>2</sub>	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3 <sub>0</sub>	2 <sub>1</sub>	1 <sub>2</sub>	0
0	0 <sub>2</sub>	1 <sub>2</sub>	3 <sub>0</sub>	1
3	1 <sub>0</sub>	2 <sub>1</sub>	2 <sub>2</sub>	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2 <sub>0</sub>	1 <sub>1</sub>	0 <sub>2</sub>
0	0	1 <sub>2</sub>	3 <sub>2</sub>	1 <sub>0</sub>
3	1	2 <sub>0</sub>	2 <sub>1</sub>	3 <sub>2</sub>
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0 <sub>0</sub>	0 <sub>1</sub>	1 <sub>2</sub>	3	1
3 <sub>2</sub>	1 <sub>2</sub>	2 <sub>0</sub>	2	3
2 <sub>0</sub>	0 <sub>1</sub>	0 <sub>2</sub>	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0 <sub>0</sub>	1 <sub>1</sub>	3 <sub>2</sub>	1
3	1 <sub>2</sub>	2 <sub>2</sub>	2 <sub>0</sub>	3
2	0 <sub>0</sub>	0 <sub>1</sub>	2 <sub>2</sub>	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0	1 <sub>0</sub>	3 <sub>1</sub>	1 <sub>2</sub>
3	1	2 <sub>2</sub>	2 <sub>2</sub>	3 <sub>0</sub>
2	0	0 <sub>0</sub>	2 <sub>1</sub>	2 <sub>2</sub>
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0	1	3	1
3 <sub>0</sub>	1 <sub>1</sub>	2 <sub>2</sub>	2	3
2 <sub>2</sub>	0 <sub>2</sub>	0 <sub>0</sub>	2	2
2 <sub>0</sub>	0 <sub>1</sub>	0 <sub>2</sub>	0	1

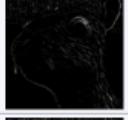
12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

3	3	2	1	0
0	0	1	3	1
3 <sub>1</sub>	1 <sub>0</sub>	2 <sub>1</sub>	2 <sub>2</sub>	3
2	0 <sub>2</sub>	0 <sub>2</sub>	2 <sub>0</sub>	2
2	0 <sub>0</sub>	0 <sub>1</sub>	0 <sub>2</sub>	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

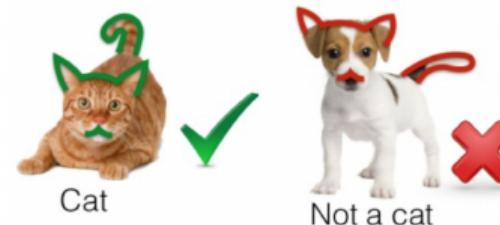
3	3	2	1	0
0	0	1	3	1
3	1	2 <sub>0</sub>	2 <sub>1</sub>	3 <sub>2</sub>
2	0	0 <sub>2</sub>	2 <sub>2</sub>	2 <sub>0</sub>
2	0	0 <sub>0</sub>	0 <sub>1</sub>	1 <sub>2</sub>

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Operation	Kernel $\omega$	Image result $g(x,y)$
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur $3 \times 3$ (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

# Двумерная свёртка

- Разные ядра помогают накладывать на картинку различные эффекты.
- Какие-то ядра помогают искать границы.
- **Идея:** возможно, с помощью некоторых ядер можно искать разные образы...

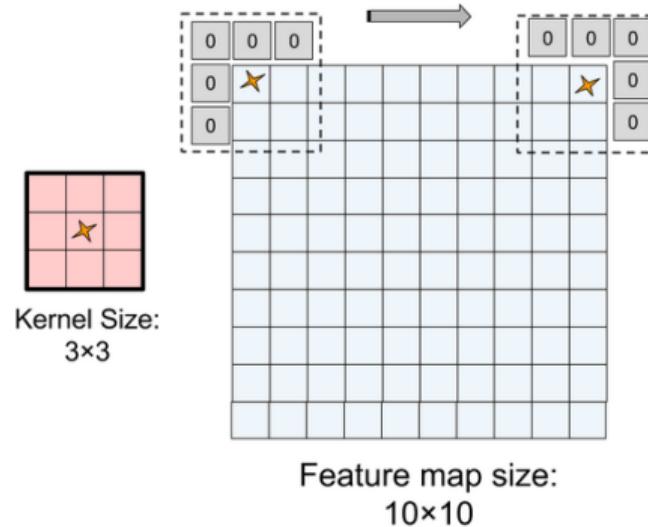


# Обработка краёв

Ядерная свёртка обычно требует значения из пикселей за пределами границ изображения. Существует множество методов обработки краёв изображений.

- **Extend.** The nearest border pixels are conceptually extended as far as necessary to provide values for the convolution. Corner pixels are extended in 90° wedges. Other edge pixels are extended in lines.
- **Wrap.** The image is conceptually wrapped (or tiled) and values are taken from the opposite edge or corner.
- **Mirror** The image is conceptually mirrored at the edges. For example, attempting to read a pixel 3 units outside an edge reads one 3 units inside the edge instead.
- **Crop.** Any pixel in the output image which would require values from beyond the edge is skipped. This method can result in the output image being slightly smaller, with the edges having been cropped.
- **Kernel Crop.** Any pixel in the kernel that extends past the input image isn't used and the normalizing is adjusted to compensate.

# Обработка краёв. Паддинг



**Дополнение (padding)** используют, чтобы пространственная размерность картинки не уменьшалась после применения свёртки, это помогает не терять информацию на краях и избежать проблем с размерностями.

# Обработка краёв. Паддинг

0 <sub>2</sub>	0 <sub>0</sub>	0 <sub>1</sub>	0	0	0	0
0 <sub>1</sub>	2 <sub>0</sub>	2 <sub>0</sub>	3	3	3	0
0 <sub>0</sub>	0 <sub>1</sub>	1 <sub>1</sub>	3	0	3	0
0	2	3	0	1	3	0
0	3	3	2	1	2	0
0	3	3	0	2	3	0
0	0	0	0	0	0	0

1	6	5
7	10	9
7	10	8

## Backpropagation in a Convolutional Layer of a CNN

Finding the gradients:

$$\frac{\partial L}{\partial F} = \text{Convolution} \left( \text{Input } X, \text{ Loss gradient } \frac{\partial L}{\partial O} \right)$$

$$\frac{\partial L}{\partial X} = \text{Full Convolution} \left( \text{180}^\circ \text{rotated Filter } F, \text{ Loss Gradient } \frac{\partial L}{\partial O} \right)$$

# Свёрточный слой

- Слой действует одинаково для каждого участка картинки, в отличие от полносвязного.
- Нужно оценивать меньшее количество параметров.
- Слой учитывает взаимное расположение пикселей.
- Можно учить тем же самым backpropagation.
- Свёрточный слой — это полносвязный слой с ограничениями, **попробуйте нарисовать свёрточный слой так же, как мы рисовали полносвязный.**

# Свёрточный слой

- Слой действует одинаково для каждого участка картинки, в отличие от полносвязного.
- Нужно оценивать меньшее количество параметров.
- Слой учитывает взаимное расположение пикселей.
- Можно учить тем же самым backpropagation.
- Свёрточный слой — это полносвязный слой с ограничениями, **попробуйте нарисовать свёрточный слой так же, как мы рисовали полносвязный.**

Но есть и другие вызовы.

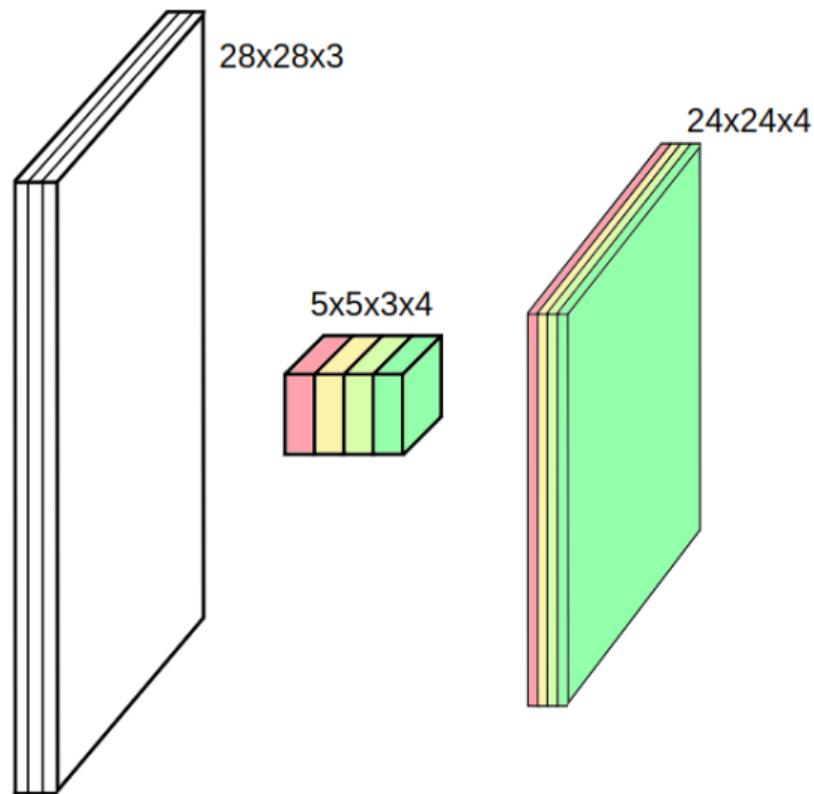
- Свёртки не инвариантны к поворотам, масштабированию, сжатиям и растяжениям.
- Для того чтобы свёрточные модели были более устойчивыми, при обучении используют **аугментации** — искусственные преобразования данных для обогащения тренировочной выборки.

## Определение

**Рецептивное поле** сенсорного нейрона — участок с рецепторами, которые при воздействии на них определённого стимула приводят к изменению возбуждения этого нейрона.

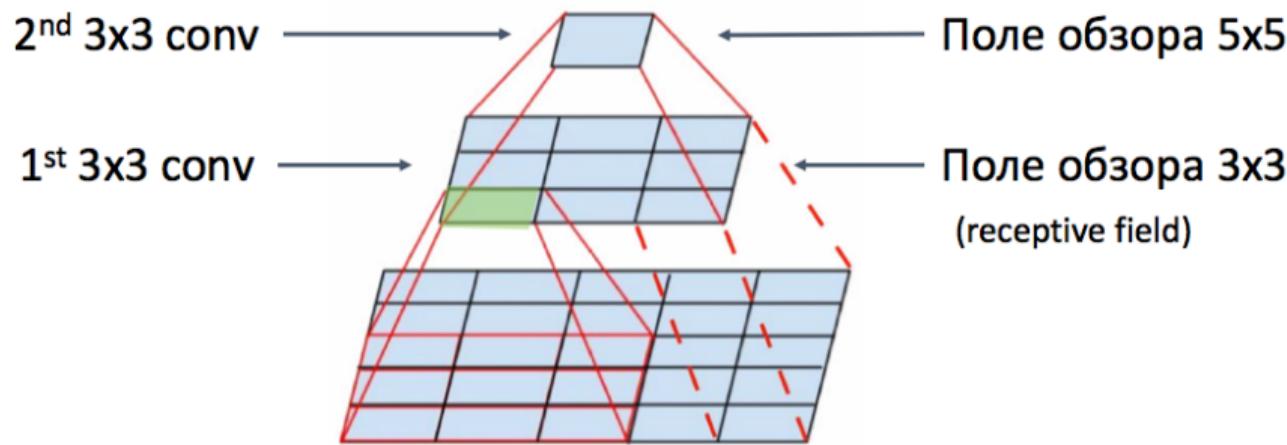
— Википедия

# Одного ядра недостаточно



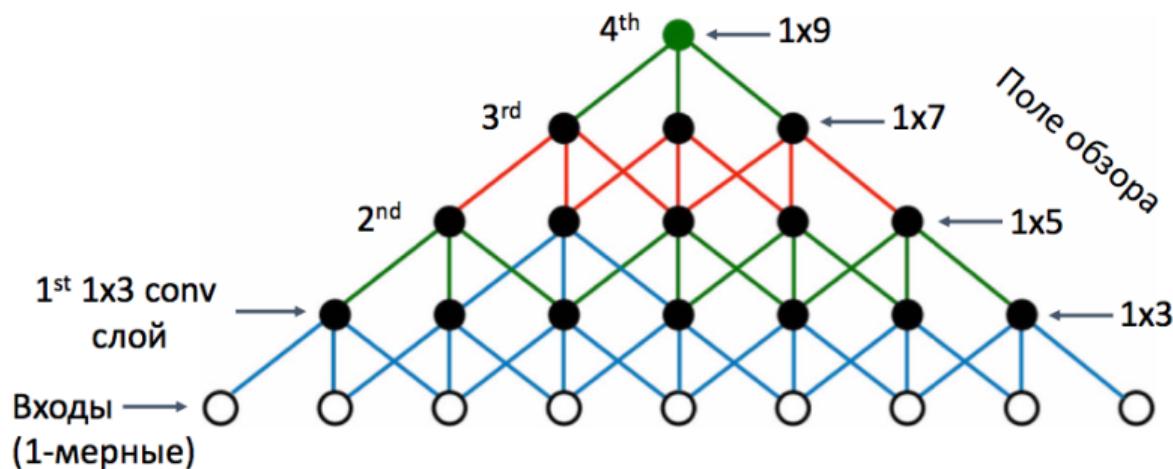
# Одного свёрточного слоя недостаточно

- Нейроны первого слоя смотрят на поле  $3 \times 3$
- Если интересующий нас объект больше, нам нужна вторая свёртка



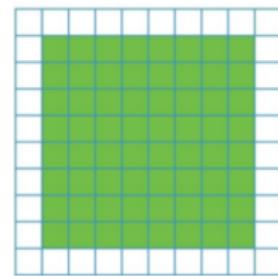
# Одного свёрточного слоя недостаточно

- $N$  слоёв со свёртками  $3 \times 3$ .
- На  $N$ -м слое поле обзора  $(2N + 1) \times (2N + 1)$ .
- Если наш объект размера 300, надо 150 слоёв... ⇒ **нужно растить поле обзора быстрее.**

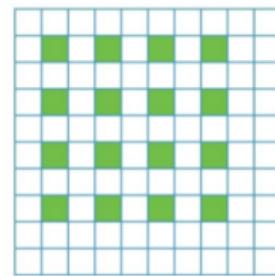


# Рецептивное поле. Шаг свёртки

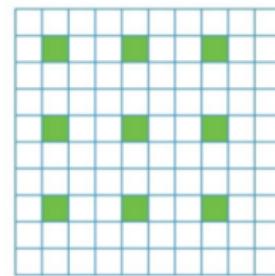
- Пиксели локально скоррелированы — соседние пиксели, как правило, не сильно отличаются друг от друга.
- Если будем делать свёртку с каким-то шагом, сэкономим мощности компьютера и не потеряем в информации.



Stride = 1



Stride = 2

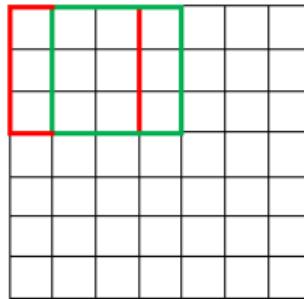


Stride = 3

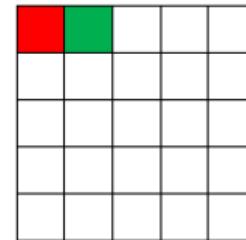
- Это очень агрессивная стратегия снижения размерности изображения.

# Рецептивное поле. Шаг свёртки

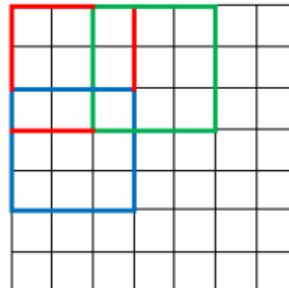
7 x 7 Input Volume



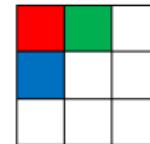
5 x 5 Output Volume



7 x 7 Input Volume



3 x 3 Output Volume



# Рецептивное поле. Пулинг

- Будем считать внутри какого-то окна максимум или среднее и сворачивать размерность, пользуясь локальной коррелированностью.

2	4	5	7	3	-2
-2	0	0	4	9	9
1	0	-1	2	1	1
1	1	6	3	7	2
3	4	0	-2	3	0
3	0	5	1	0	0

Feature Map

2	4	5	7	3	-2
-2	0	0	4	9	9
1	0	-1	2	1	1
1	1	6	3	7	2
3	4	0	-2	3	0
3	0	5	1	0	0

Pool size=2

Max  
Pooling

4	7	9
1	6	7
4	5	3

Average  
Pooling

1	4	4,8
0,8	2,5	2,8
2,5	1	0,8

# Содержание

## 1 Компьютерное зрение

Что это такое

История развития CV

Представление изображений в компьютере

FFNN vs. CNN

## 2 Свёртки

Введение

Двумерная свёртка

Обработка краёв

Градиенты

Свёрточный слой

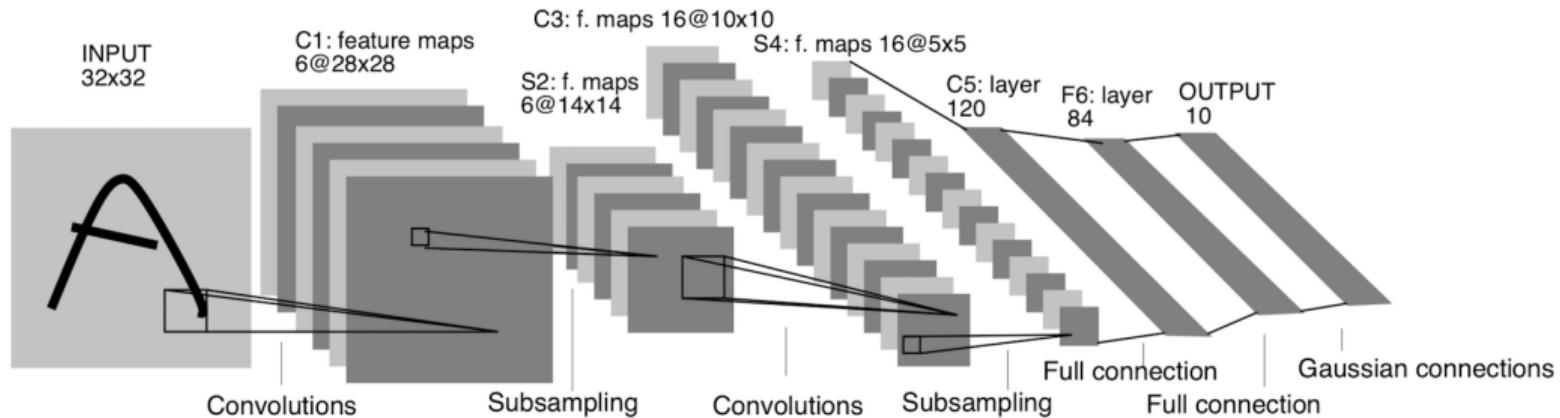
Рецептивное поле

## 3 Примеры свёрточных сетей

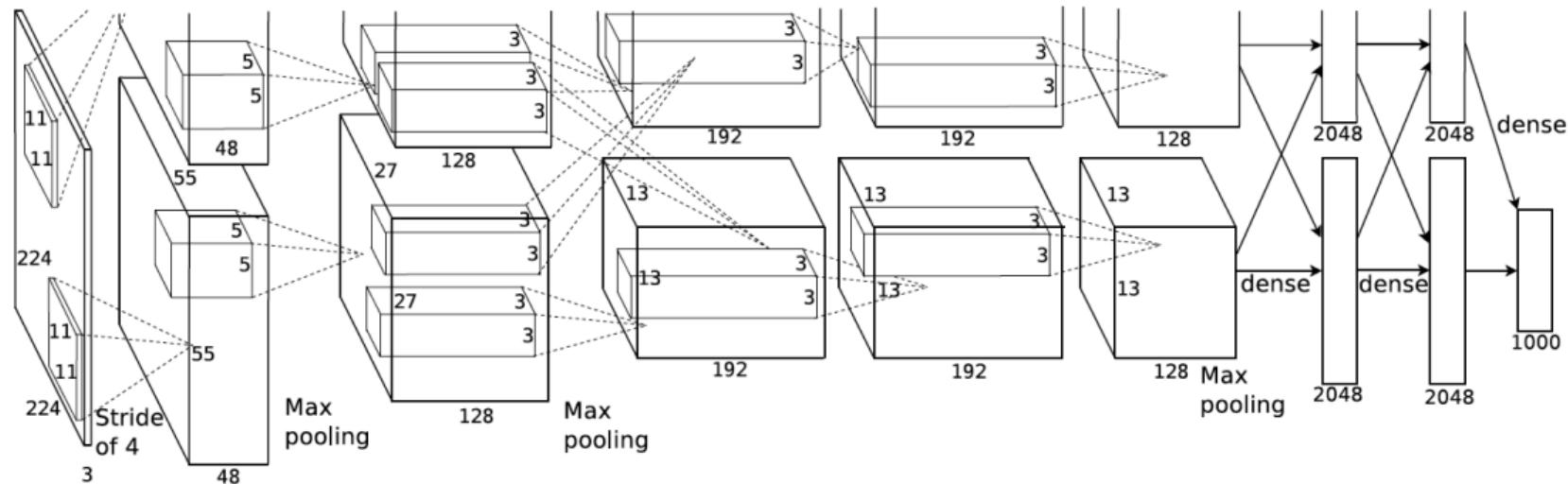
## 4 Интерпретация

## 5 Что узнали

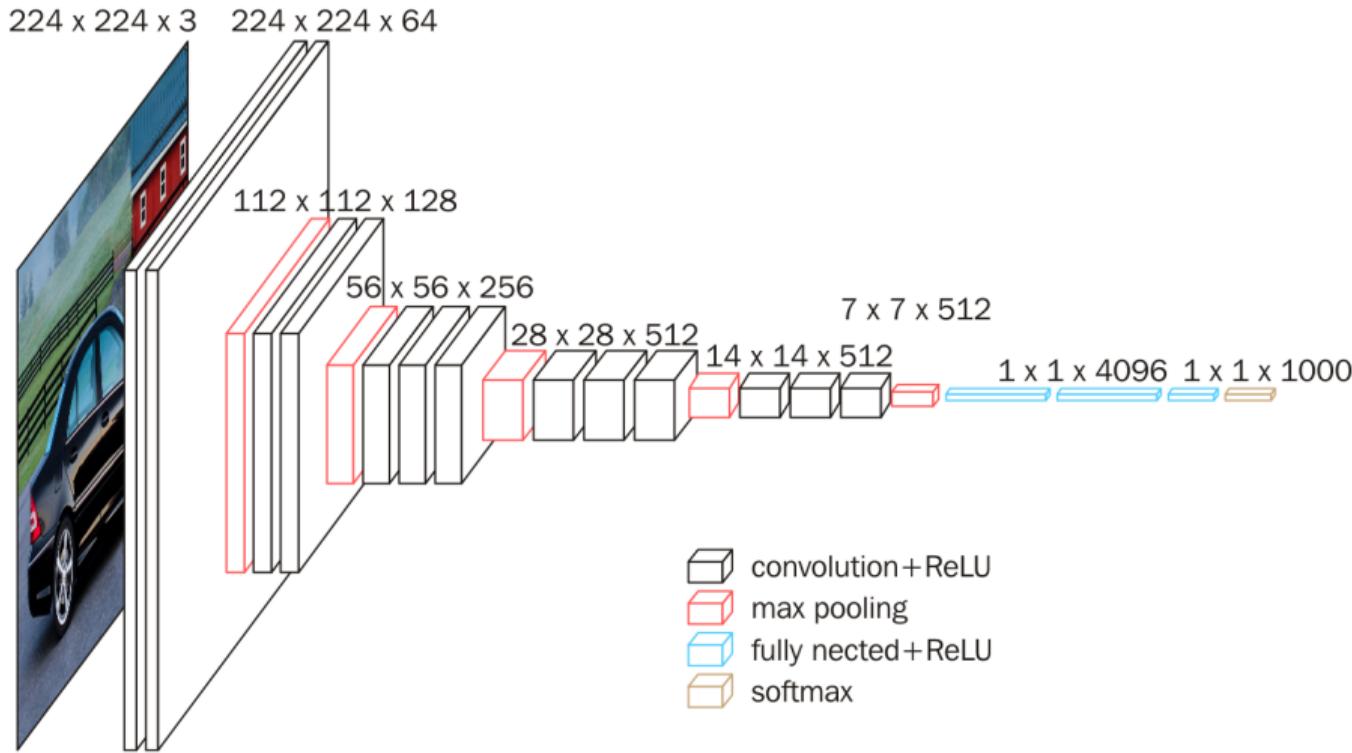
# Примеры свёрточных сетей. LeNet (1998)



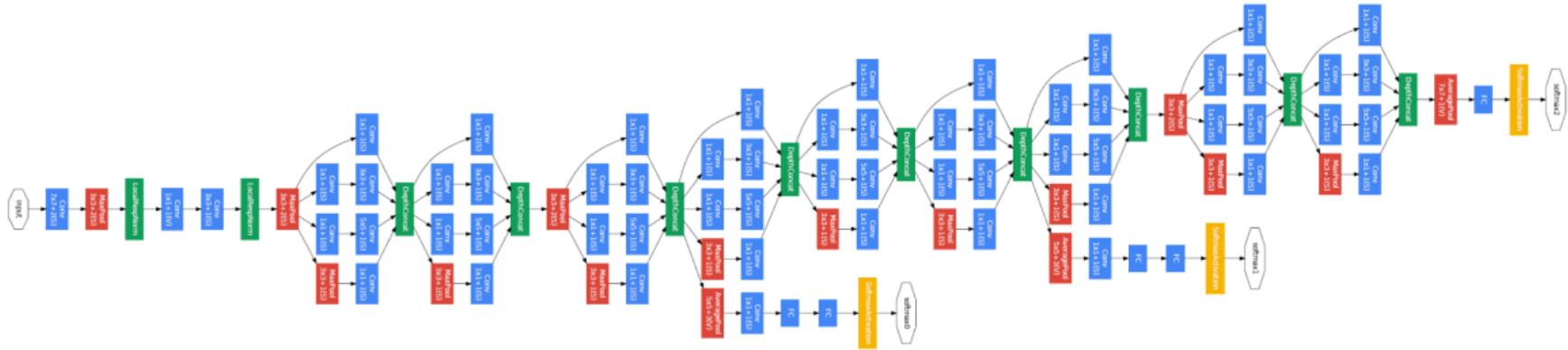
# Примеры свёрточных сетей. AlexNet (2012)



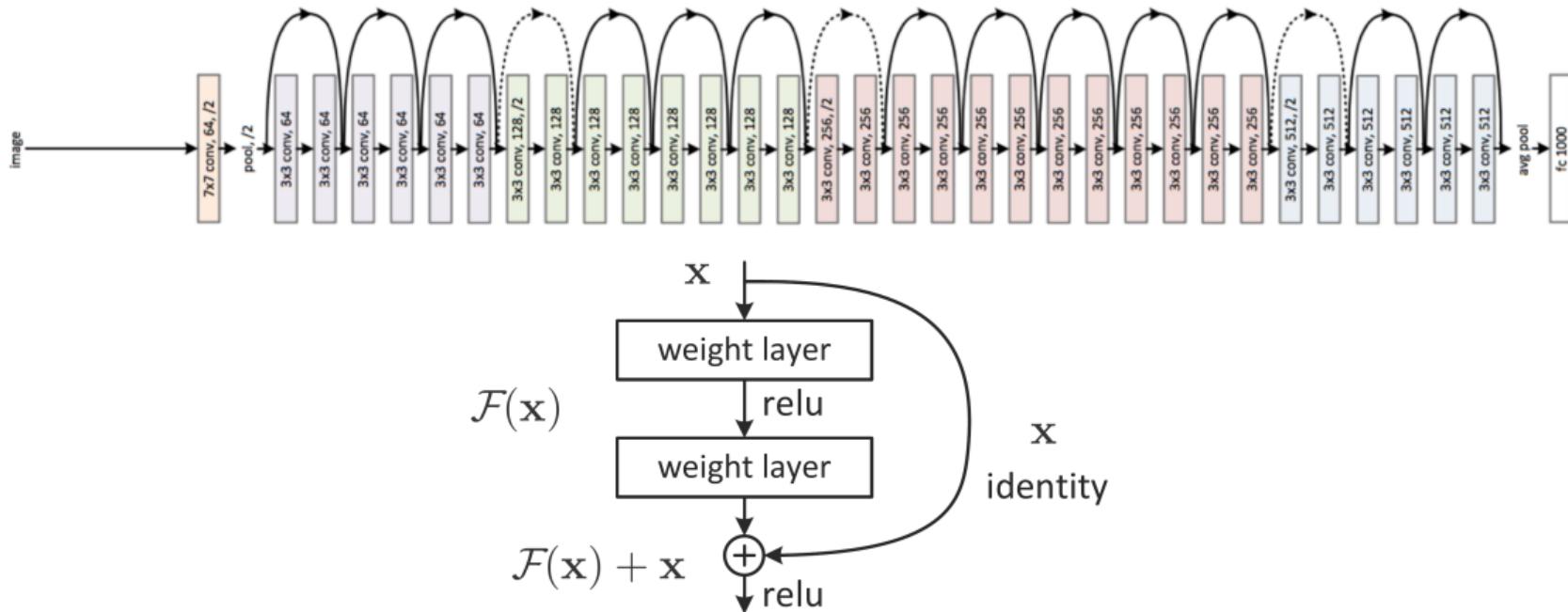
# Примеры свёрточных сетей. VGG (2014)



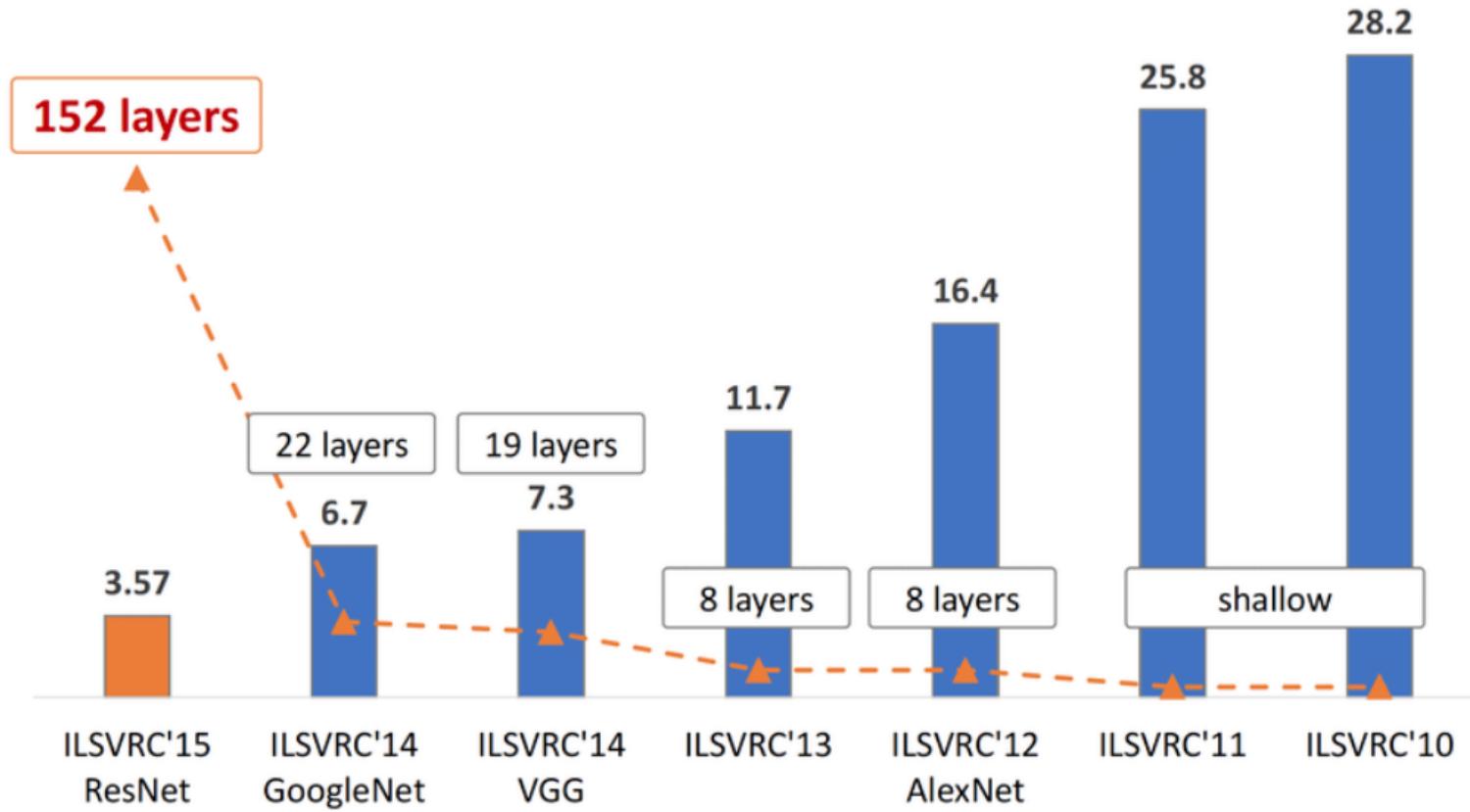
# Примеры свёрточных сетей. GoogleNet (2014)



# Примеры свёрточных сетей. ResNet (2015)



# Результаты классификации на ImageNet



# Содержание

## 1 Компьютерное зрение

Что это такое

История развития CV

Представление изображений в компьютере

FFNN vs. CNN

## 2 Свёртки

Введение

Двумерная свёртка

Обработка краёв

Градиенты

Свёрточный слой

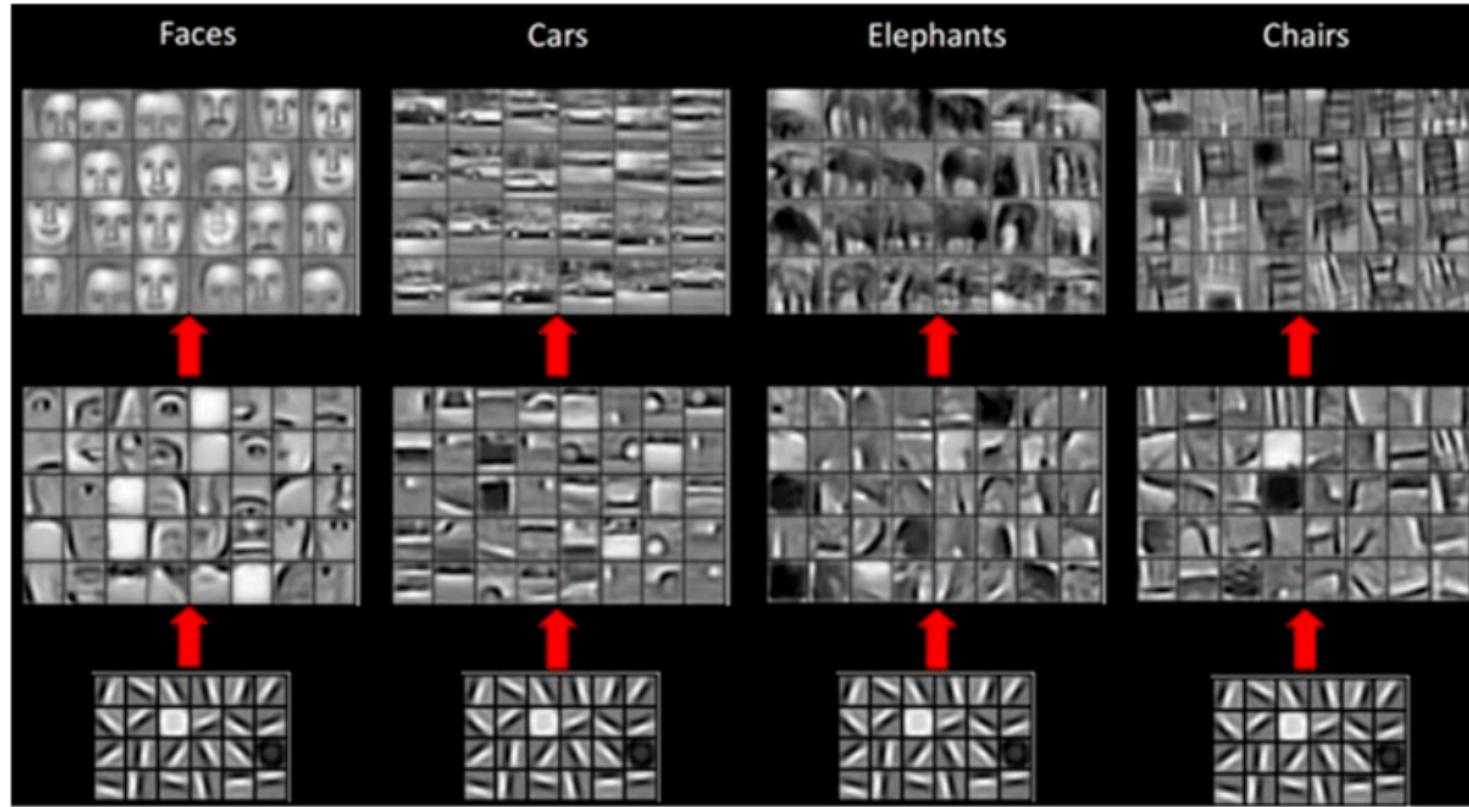
Рецептивное поле

## 3 Примеры свёрточных сетей

## 4 Интерпретация

## 5 Что узнали

# Интерпретация



# Содержание

## 1 Компьютерное зрение

Что это такое

История развития CV

Представление изображений в компьютере

FFNN vs. CNN

## 2 Свёртки

Введение

Двумерная свёртка

Обработка краёв

Градиенты

Свёрточный слой

Рецептивное поле

## 3 Примеры свёрточных сетей

## 4 Интерпретация

## 5 Что узнали

# Что узнали

- Чем занимается компьютерное зрение.
- Для чего нужны свёрточные нейронные сети.
- Как обрабатывать изображения при помощи свёрточных сетей.
- Современные глубокие архитектуры.
- Интерпретация позволяет лучше понять модель.