

Глубинное обучение

Лекция 6

Свёрточные нейронные сети. Интерпретация моделей.

Михаил Гущин

mhushchyn@hse.ru

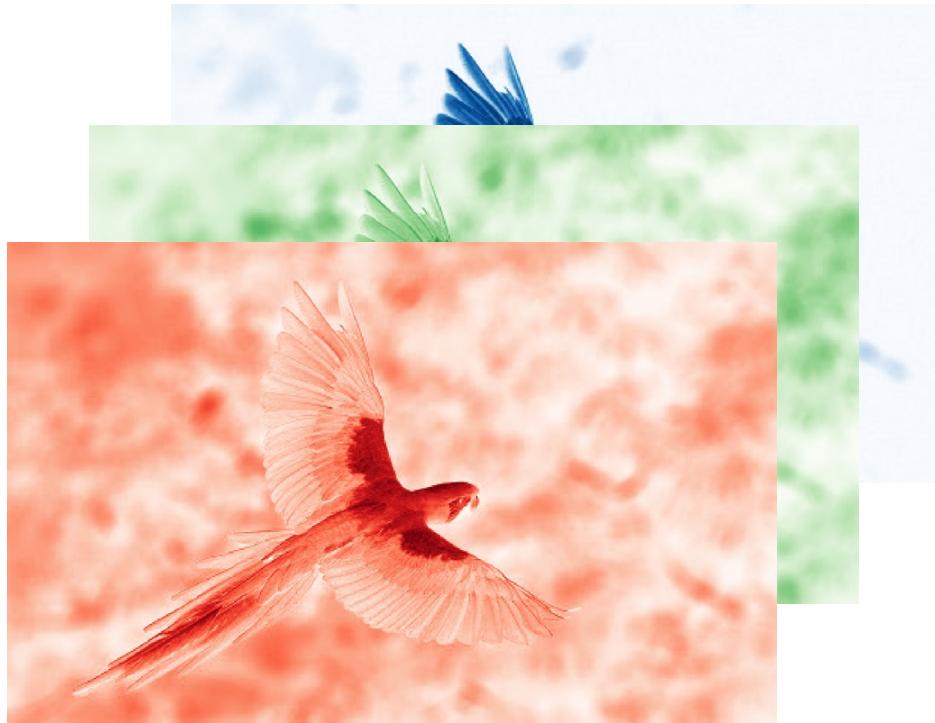
НИУ ВШЭ, 2024



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

В предыдущей серии

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**

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

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

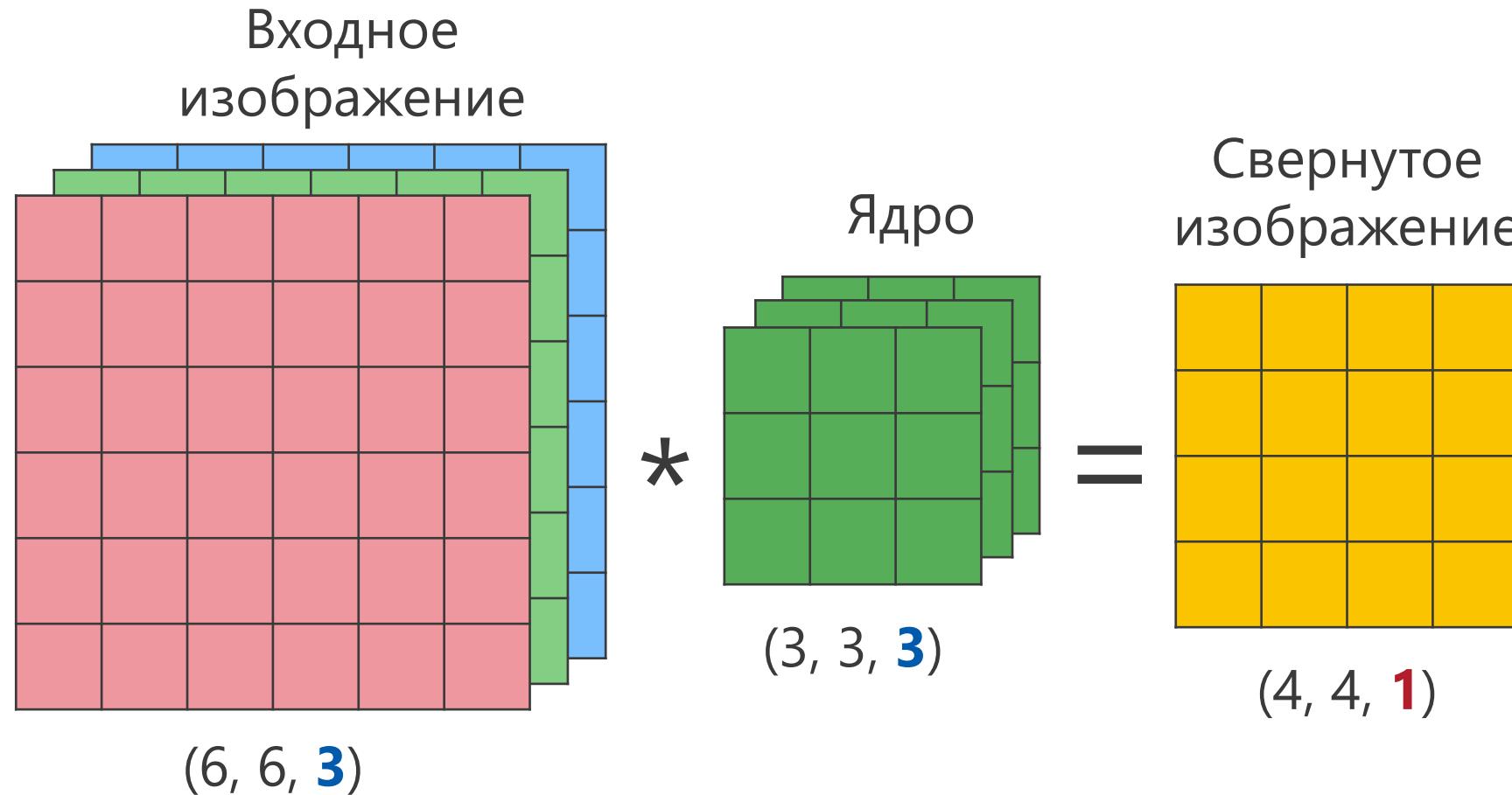
(2, 2, 1)

Ядро свертки как нейрон:

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

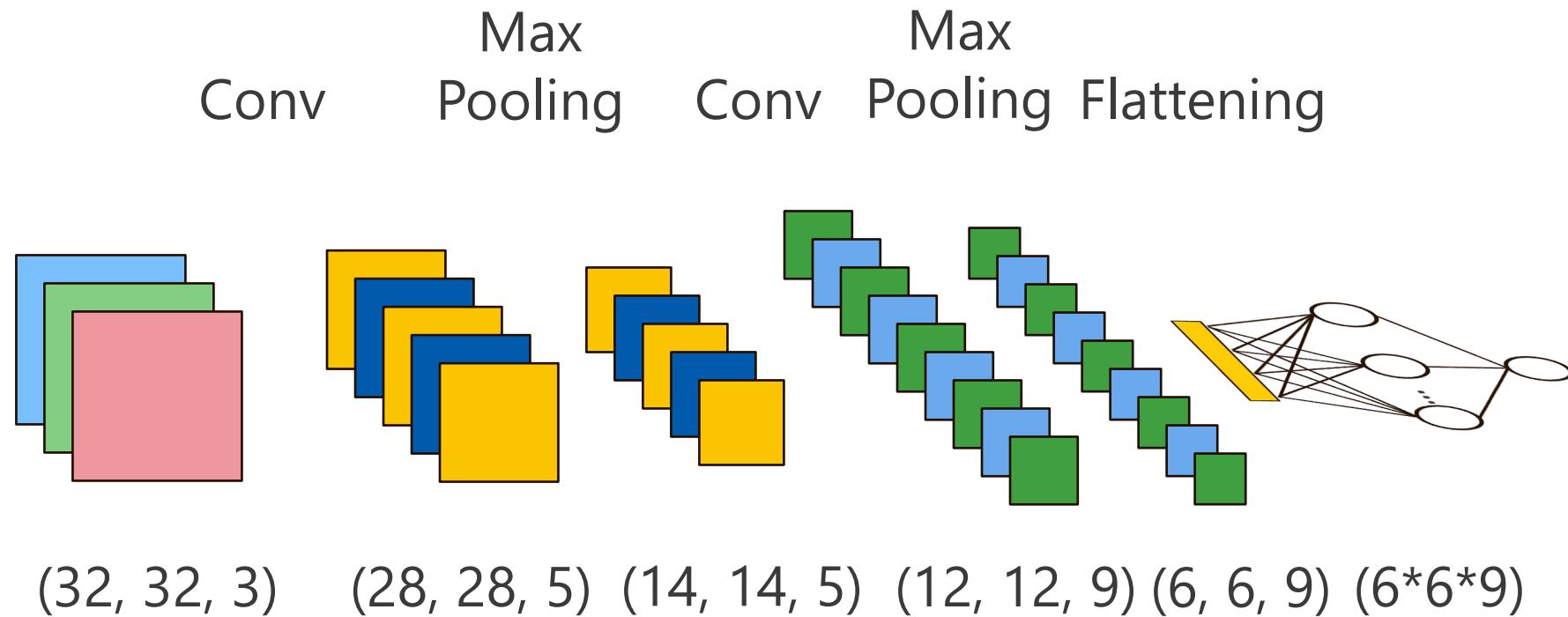
Веса w_i получим в процессе обучения

Свертка многоканального изображения



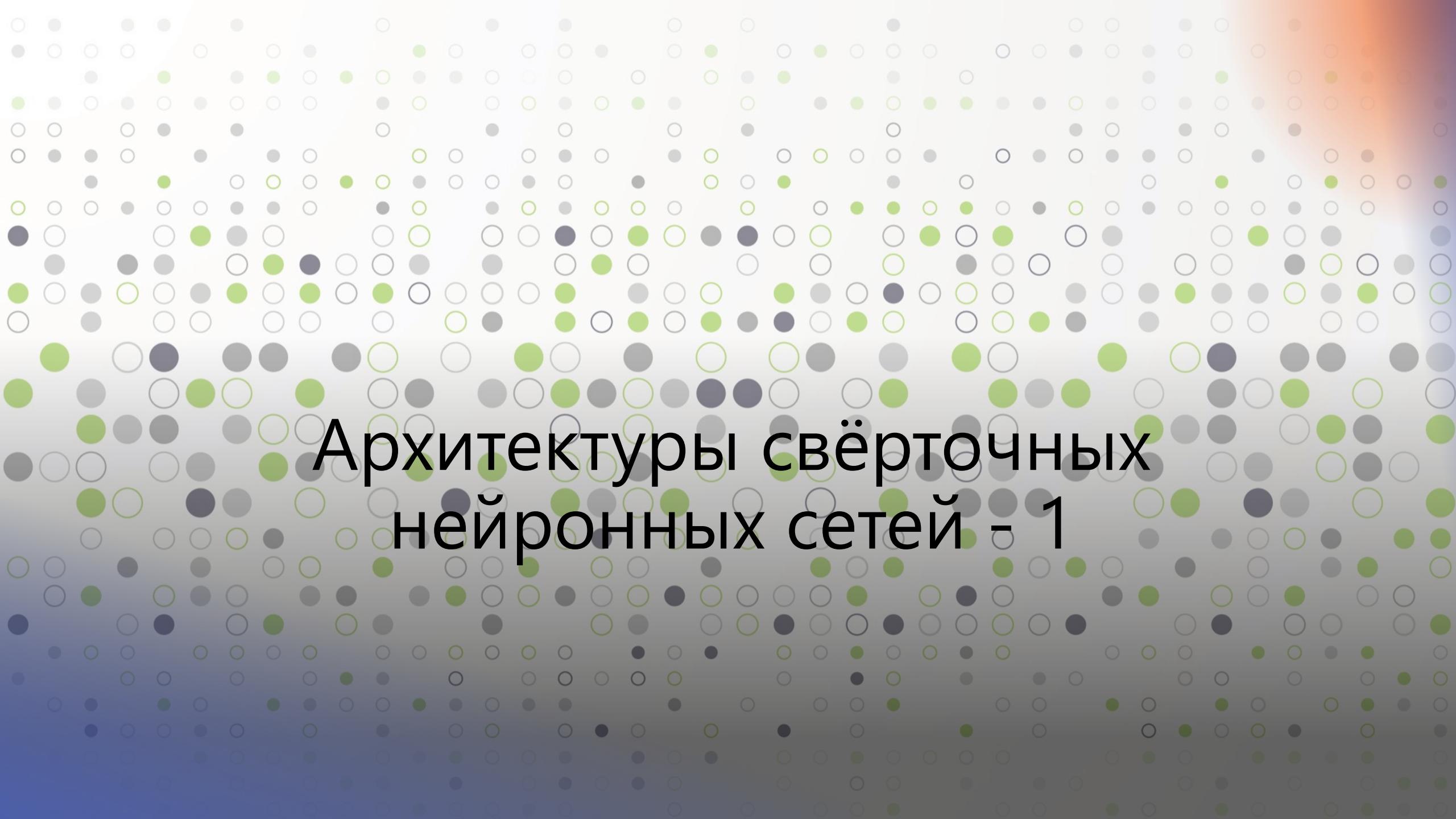
Результат свертки — одноканальное изображение

Сверточная сеть



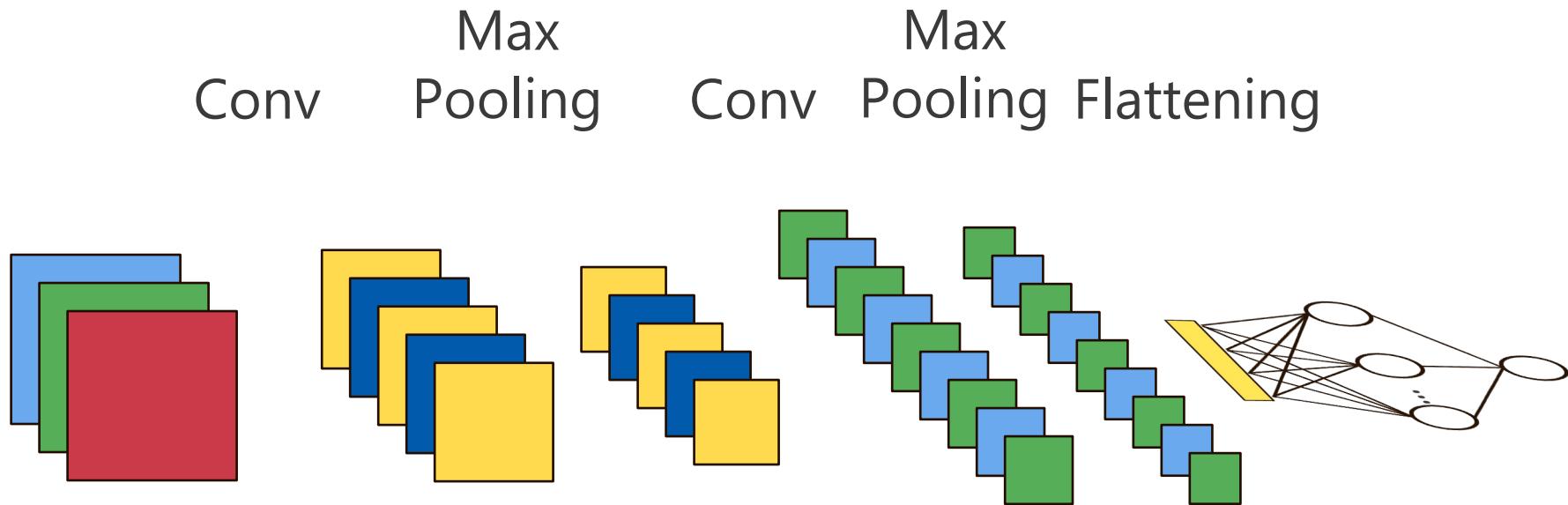
Типичная архитектура

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



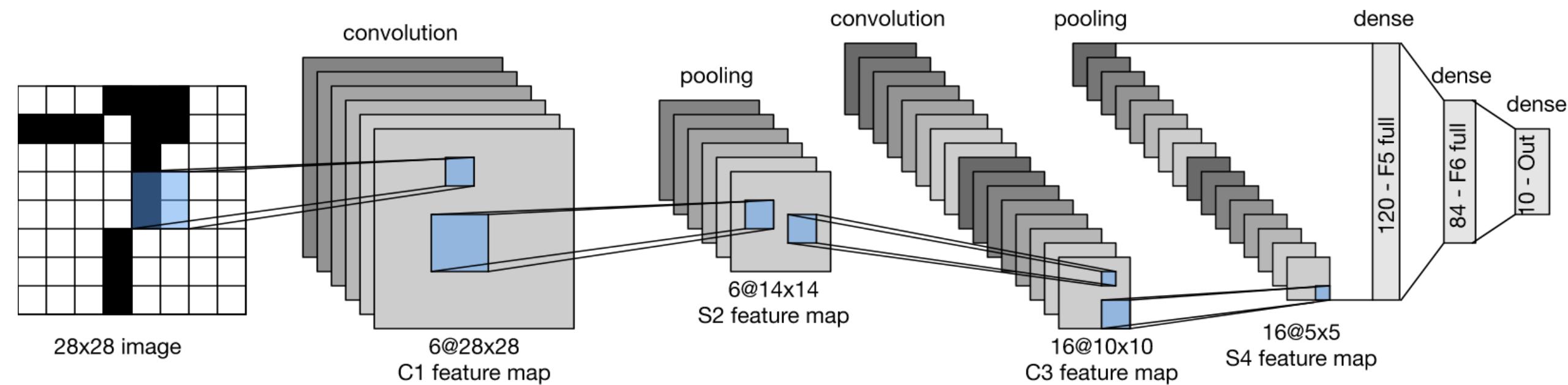
Архитектуры свёрточных нейронных сетей - 1

Типичная архитектура CNN



- ▶ Сколько слоев взять?
- ▶ Какой размер ядер сверток?

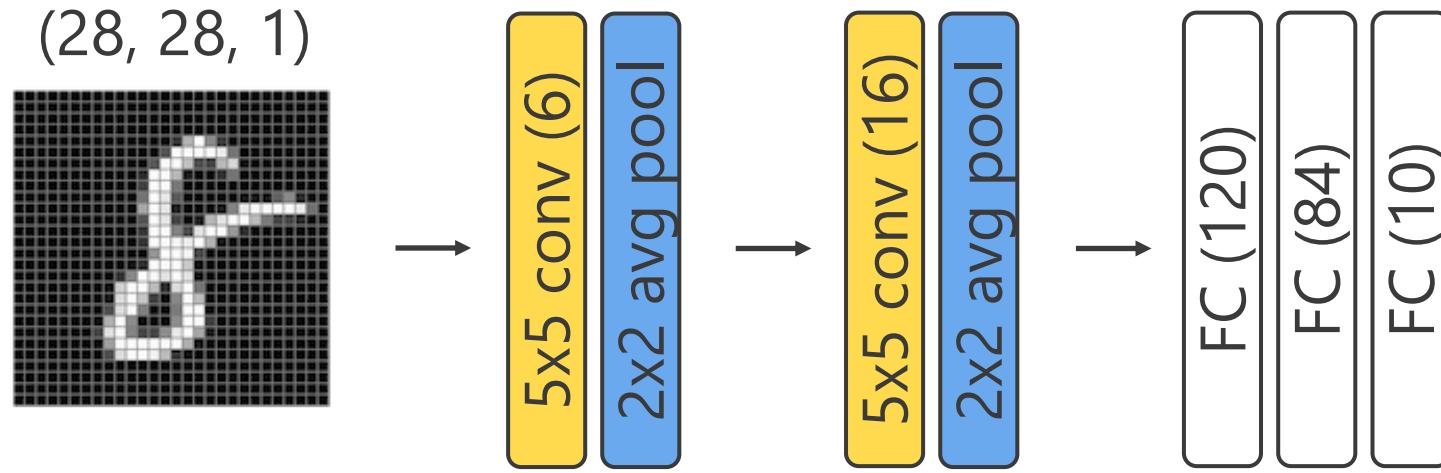
LeNet



Опубликована в 1998 г. для классификации рукописных цифр MNIST

Подробнее: Y. LeCun et al. Gradient-based learning applied to document recognition, IEEE 1998 (URL: <https://doi.org/10.1109/5.726791>)

LeNet



- ▶ Одноканальное изображение на входе
- ▶ Сигмоида (sigmoid) функция активации после сверточных и полносвязанных (FC) слоев
- ▶ Softmax функция активации выходного слоя

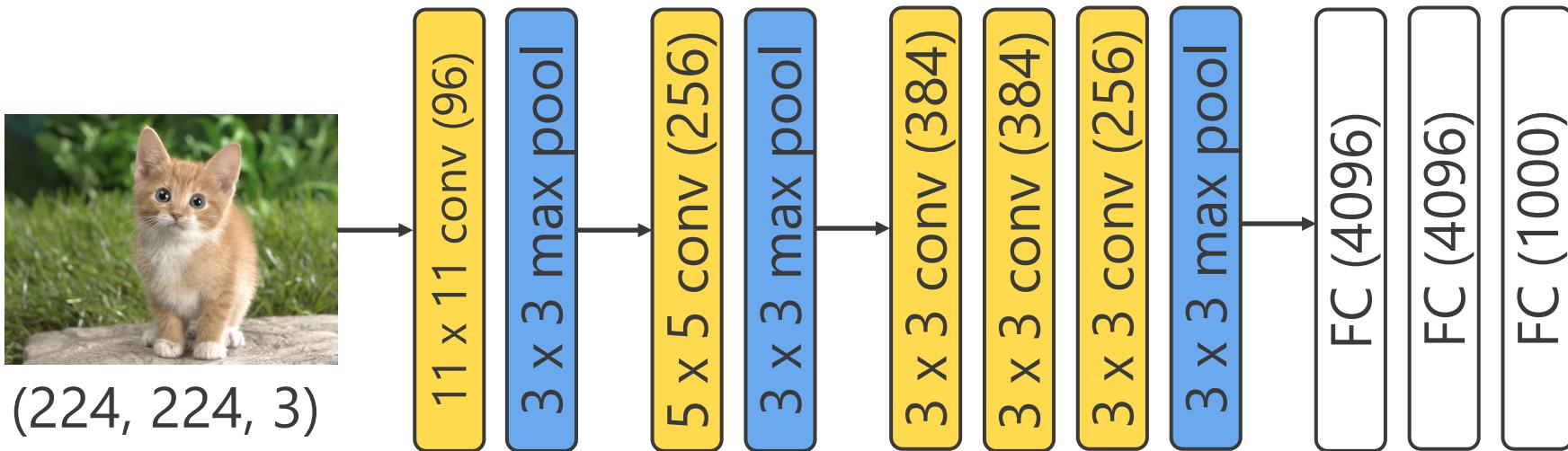
ImageNet



Подробнее: <https://image-net.org/challenges/LSVRC>

- ▶ ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
- ▶ 1000 классов изображений
- ▶ Более 1 000 000 изображений

AlexNet



- ▶ Больше слоев и нейронов
- ▶ ReLU функция активации во всех слоях
- ▶ Softmax функция активации выходного слоя

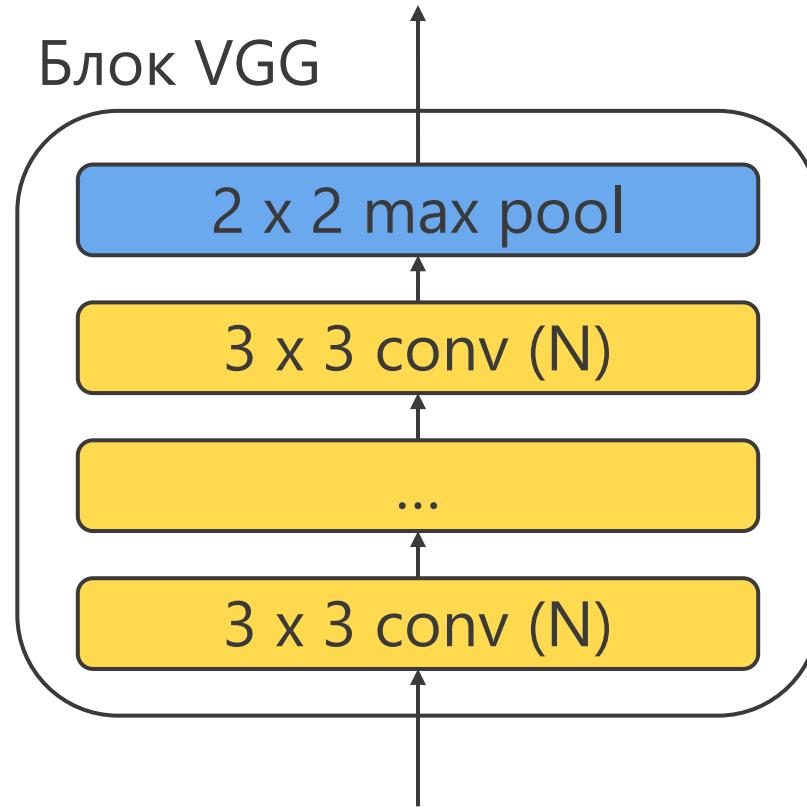
Подробнее: A. Krizhevsky et al. ImageNet classification with deep convolutional neural networks,
NIPS 2012 (URL: <https://doi.org/10.1145/3065386>)

Visual Geometry Group (VGG)

- ▶ Участник ImageNet 2014 года
- ▶ Имеет блочную структуру
- ▶ Блок аналогичен одному слою сети
- ▶ Каждый блок состоит из набора свёрточных слоев
- ▶ Маленькие свёртки 3×3

Подробнее: K. Simonyan et al. Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015
(URL: <https://arxiv.org/abs/1409.1556>)

Блок VGG

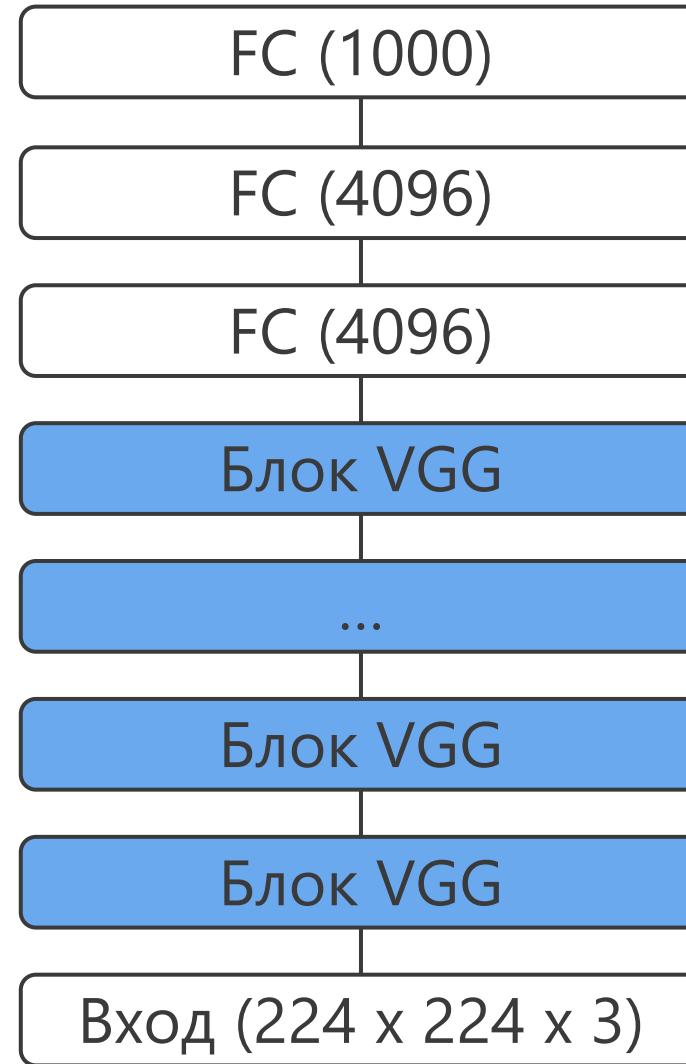


$(N_{\text{входных каналов}}, N_{\text{выходных каналов}}, N_{\text{сверточных слоев}})$

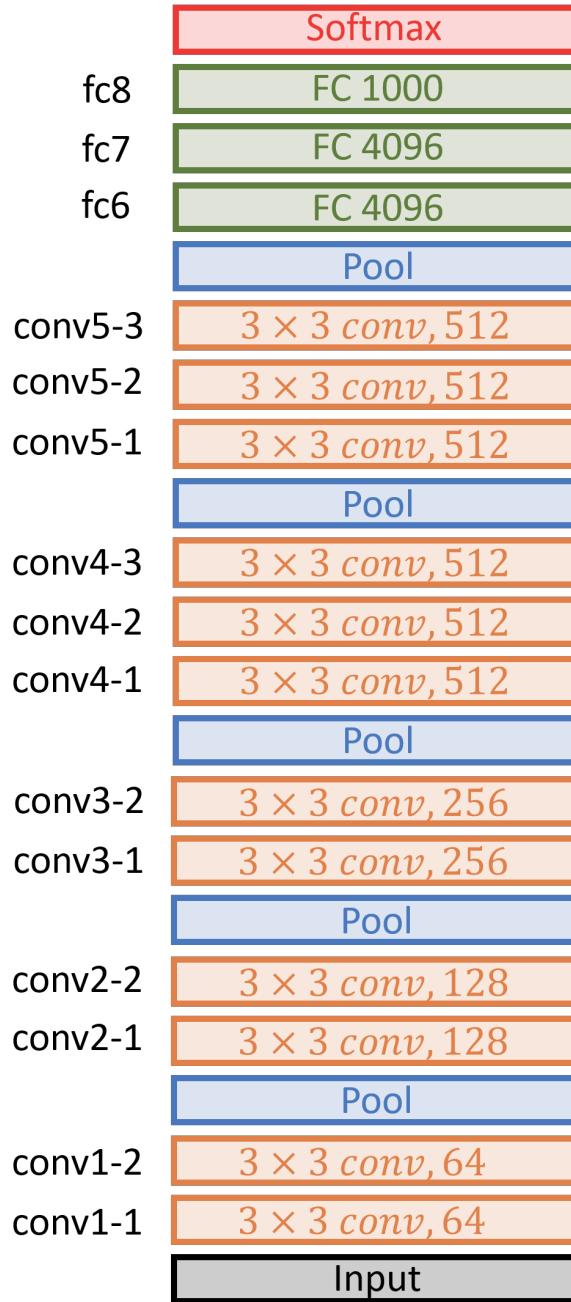
Блок VGG

- ▶ Свертки размером 3×3
- ▶ Свертки сохраняют размер изображения:
 - 3×3 , padding 1
- ▶ Однаковое число выходных каналов на всех слоях
- ▶ ReLU функция активации на всех сверточных слоях
- ▶ Изображение уменьшается в два раза
 - 2×2 max pooling

Архитектура VGG

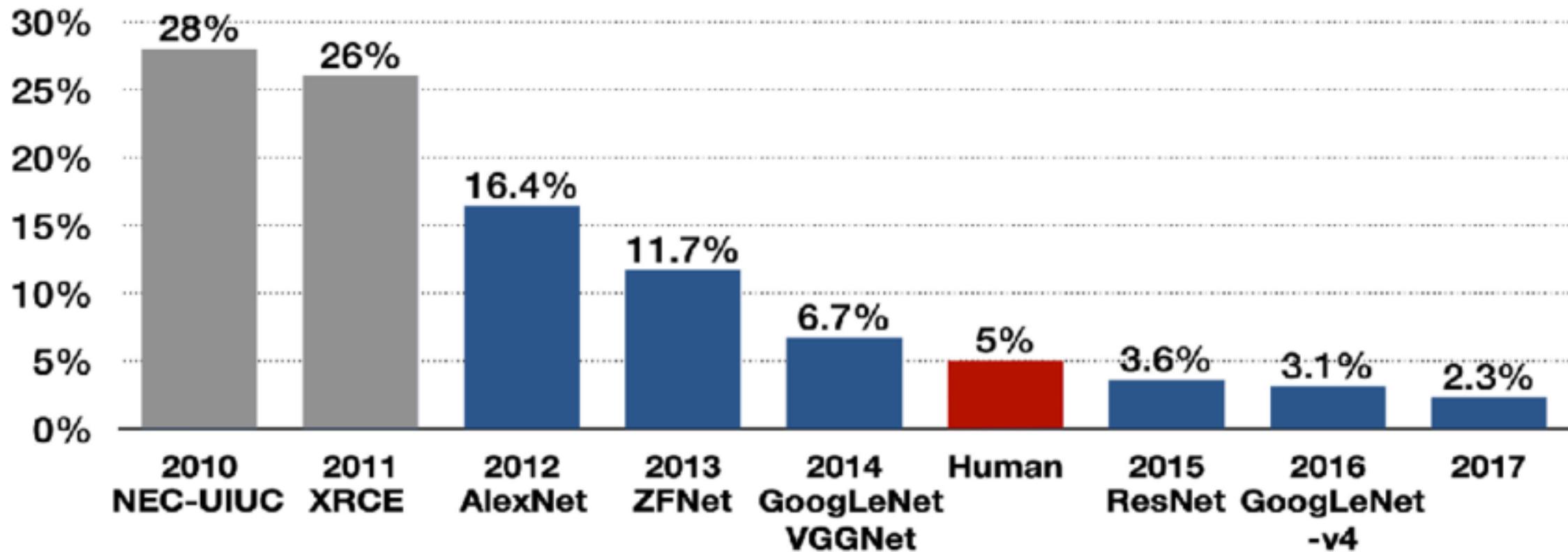


VGG-16

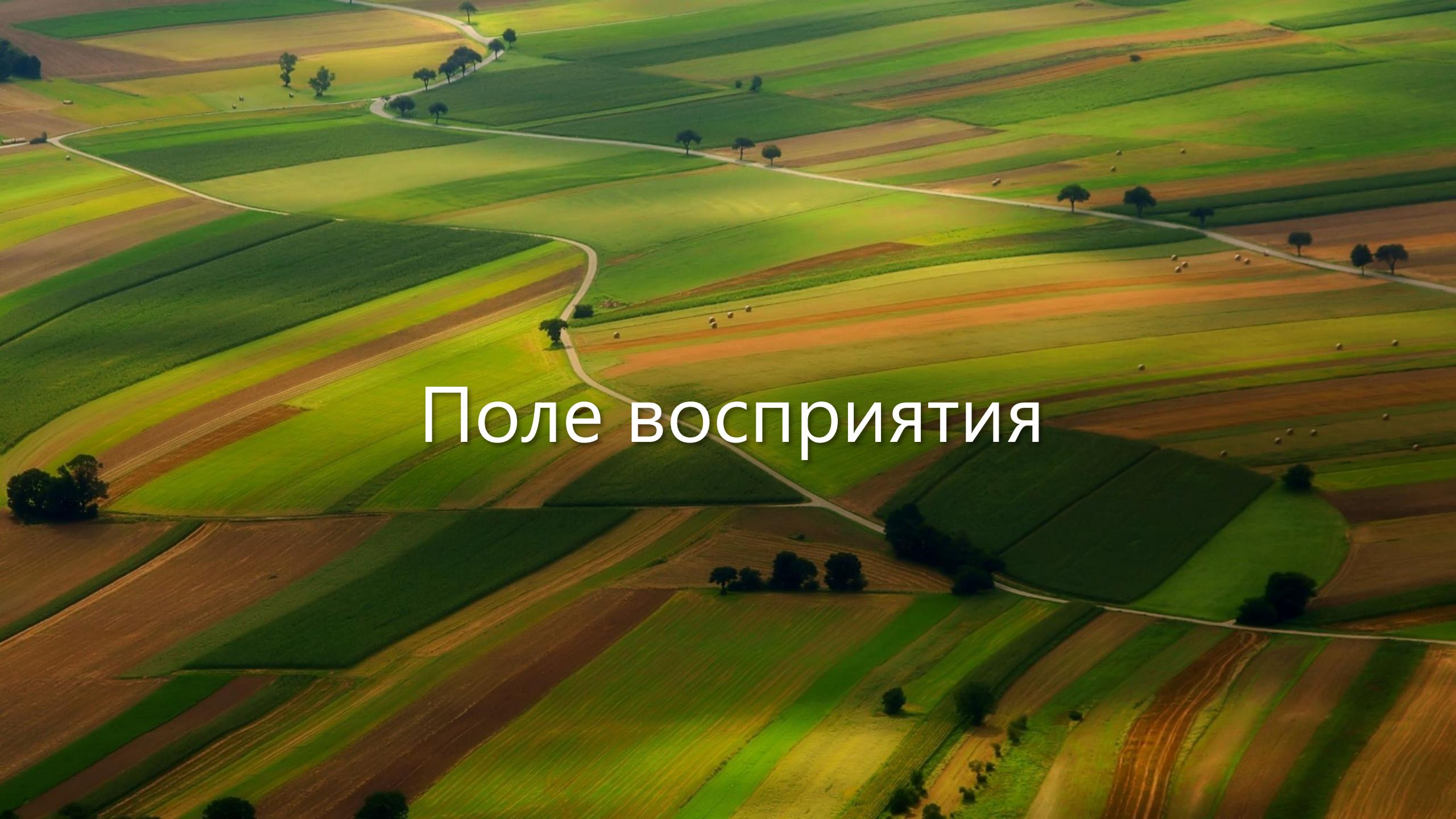


Результаты ImageNet

Top-5 error



- ▶ Зачем нужны VGG блоки?
- ▶ Почему используются маленькие свертки?
- ▶ Нельзя ли заменить их большими свертками 11×11 ?

An aerial photograph of a rural landscape featuring numerous agricultural fields in various stages of cultivation. The fields are a patchwork of green, yellow, and brown, separated by a network of white roads and small clusters of trees. The terrain is slightly hilly, creating a textured pattern of geometric shapes.

Поле восприятия

Поле восприятия

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

Свертка 1

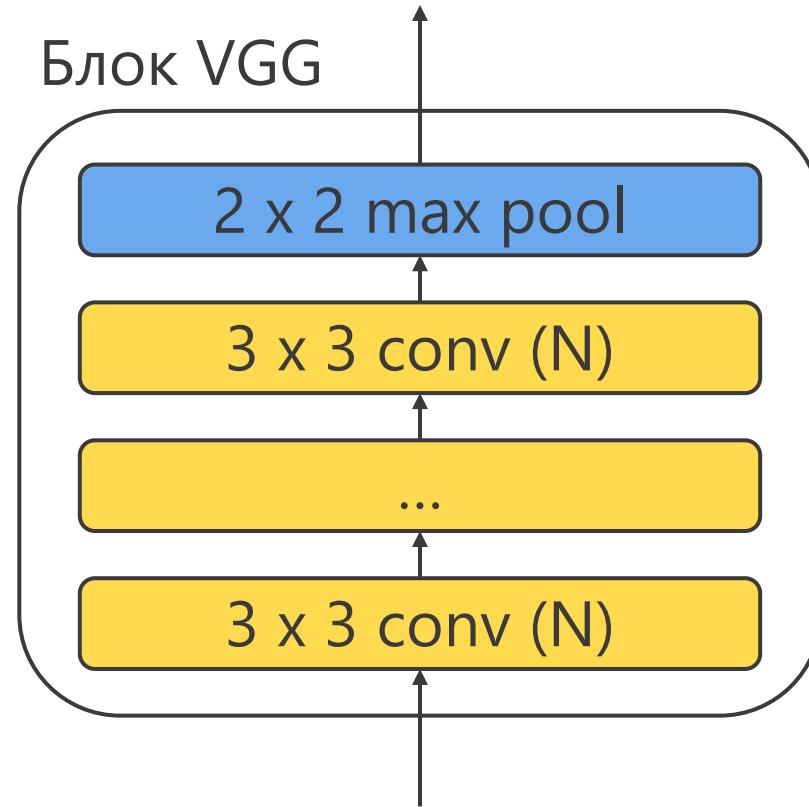
9	2	2	6
-1	4	8	7
-4	8	8	0
2	9	7	4

Свертка 2

11	15
25	23

- ▶ Поле восприятия увеличивается с каждым новым свёрточным слоем
- ▶ Поле восприятия **зеленого пикселя** – 5 x 5 пикселей входного изображения

Блок VGG



$(N_{\text{входных каналов}}, N_{\text{выходных каналов}}, N_{\text{сверточных слоев}})$

Поле восприятия

Свойства VGG блока:

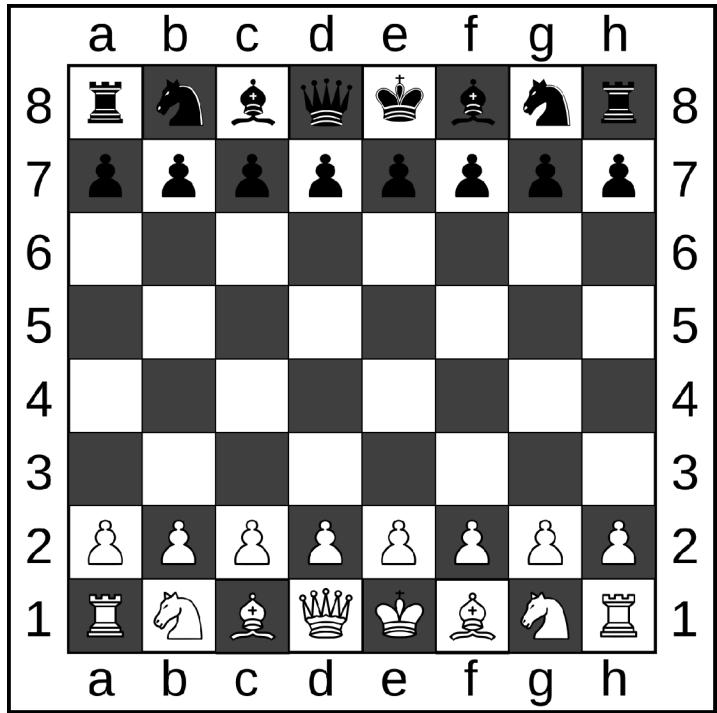
- ▶ Поле восприятия 3 последовательных сверток 3×3 эквивалентно свертке 7×7
- ▶ Но использование функций активации на каждом слое делает зависимости пикселей нелинейными
- ▶ Меньше параметров для обучения: $(3*3)*3$ против $(7*7)$

Рекомендации по архитектуре

- ▶ Берем больше свёрточных слоев
- ▶ Лучше использовать маленькие свертки
 - $1 \times 1, 3 \times 3, 5 \times 5$
- ▶ Увеличиваем число каналов с каждым свёрточным слоем
- ▶ Используем ReLU функцию активации после каждой свертки (но до pooling)

Интерпретация моделей

Пример: детектор вертикальных границ



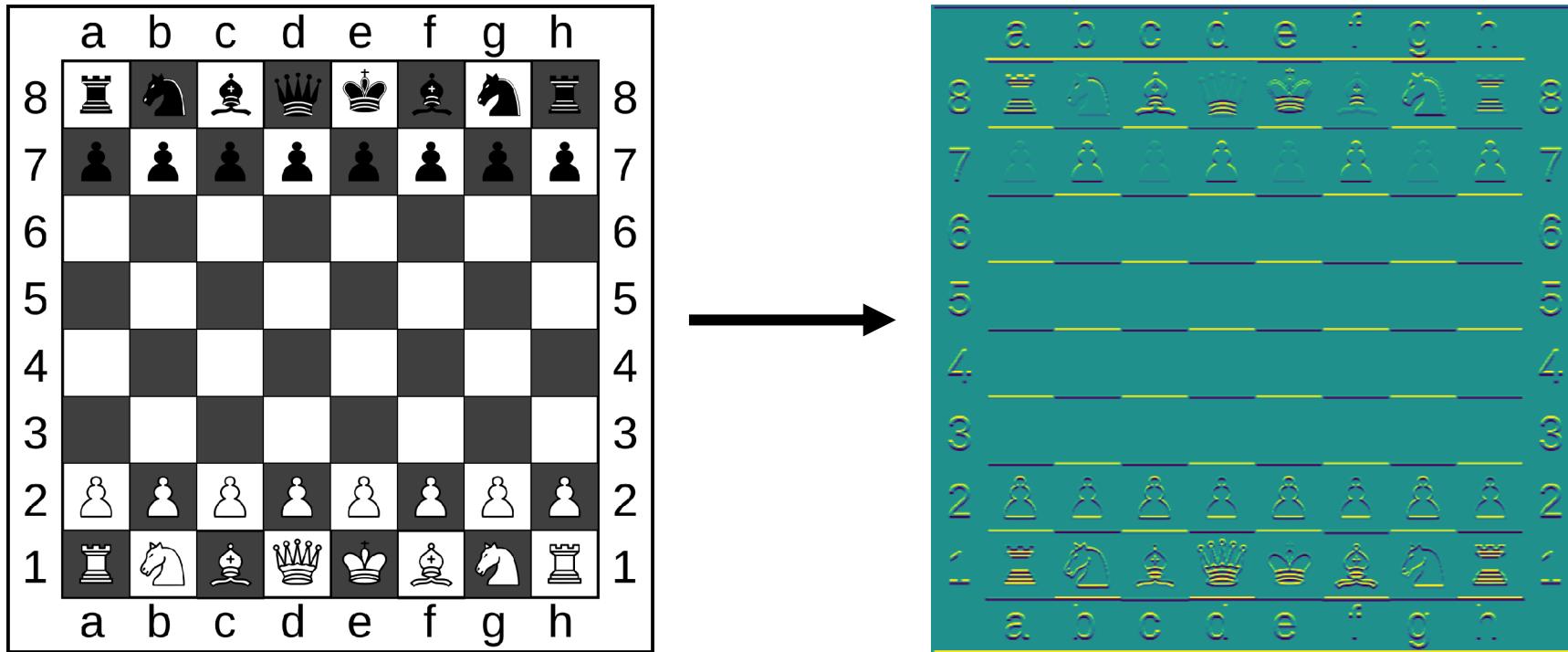
Оператор
Собеля

*

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

- ▶ Применим оператор Собеля для обнаружения горизонтальных линий

Пример: детектор вертикальных границ



Интерпретация моделей

- ▶ Каждое ядро свертки модели реагирует на определенный шаблон в изображении
- ▶ Как узнать, на какие именно шаблоны реагирует модель?

Интерпретация моделей

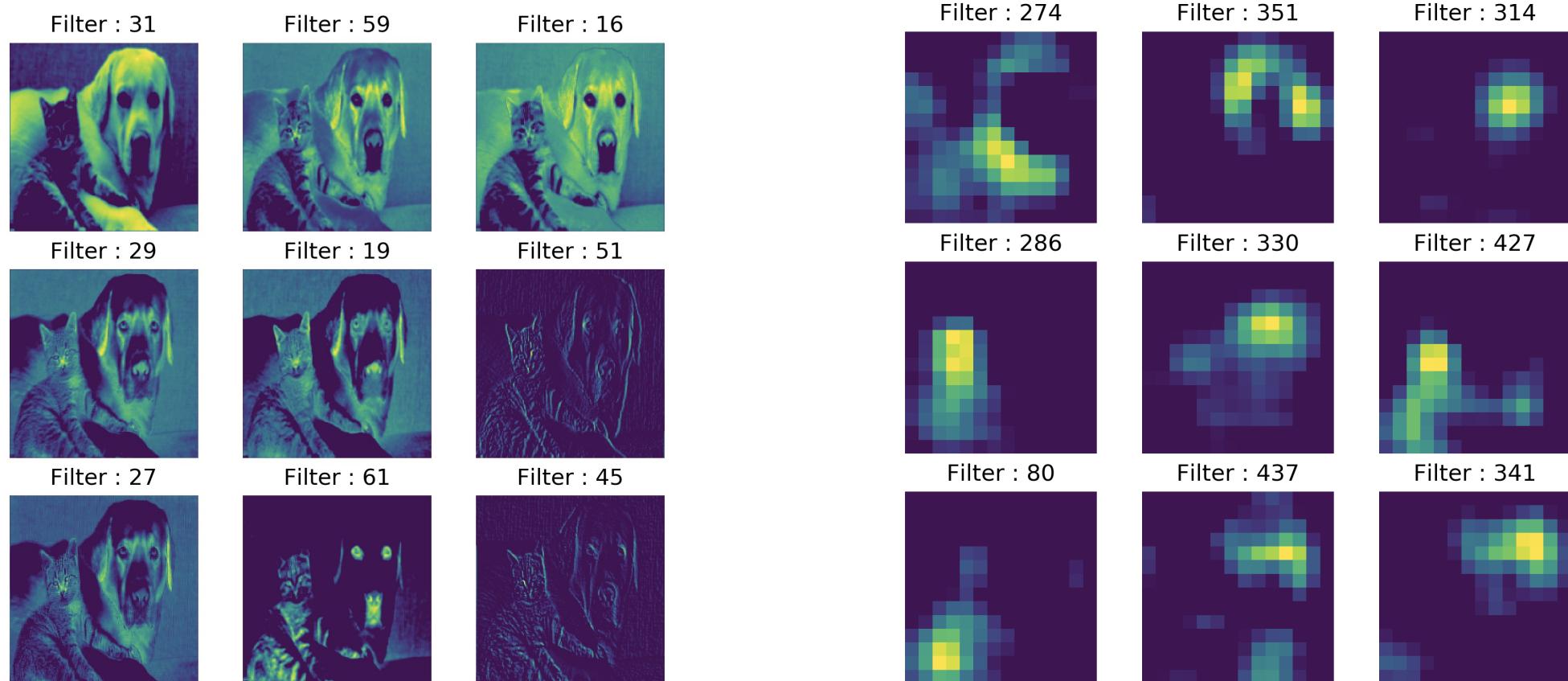
- ▶ Возьмем обученную на ImageNet сеть VGG-16
- ▶ Подадим ей на вход изображение
- ▶ Построим результат свертки (после функции активации) этого изображения для разных фильтров на разных слоях сети

Визуализация активаций VGG-16



Входное изображение

Визуализация активаций VGG-16



Блок 1, слой 1

Блок 5, слой 3

Источник: <https://mydeeplearningnb.wordpress.com/2019/07/24/visualizing-cnns-using-tensorflow/>

Максимизация активации

Maximally Activated Patch:

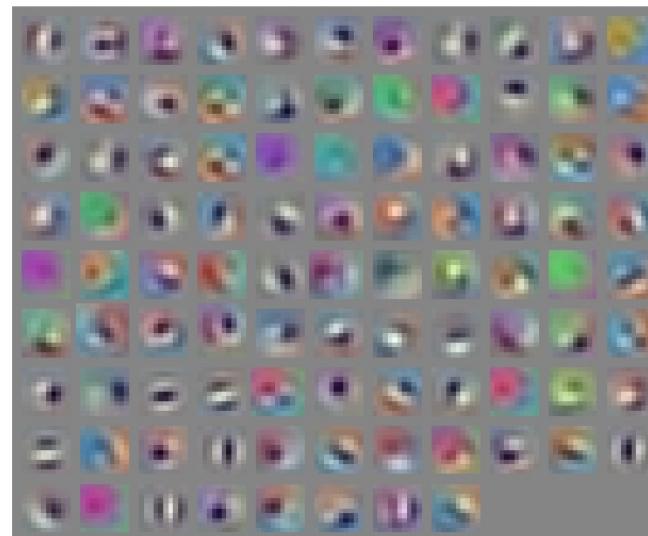
- ▶ Возьмем набор разных кусочков картинок
- ▶ Пропустим их через обученную сеть
- ▶ Выберем фильтр в каком-то слое сети
- ▶ Выберем те кусочки картинок, которые максимизируют активацию выбранного фильтра свертки

Максимизация активации

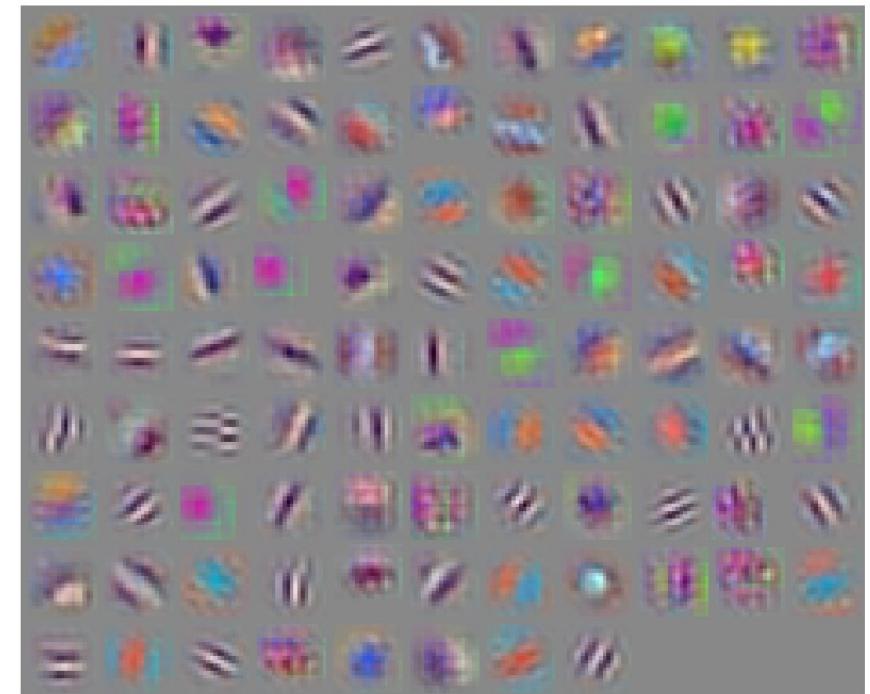
conv1



conv2



conv3



Каждая картинка – максимальная активация
каждого фильтра в слое

Источник: <https://arxiv.org/pdf/1412.6806.pdf>

Максимизация активации

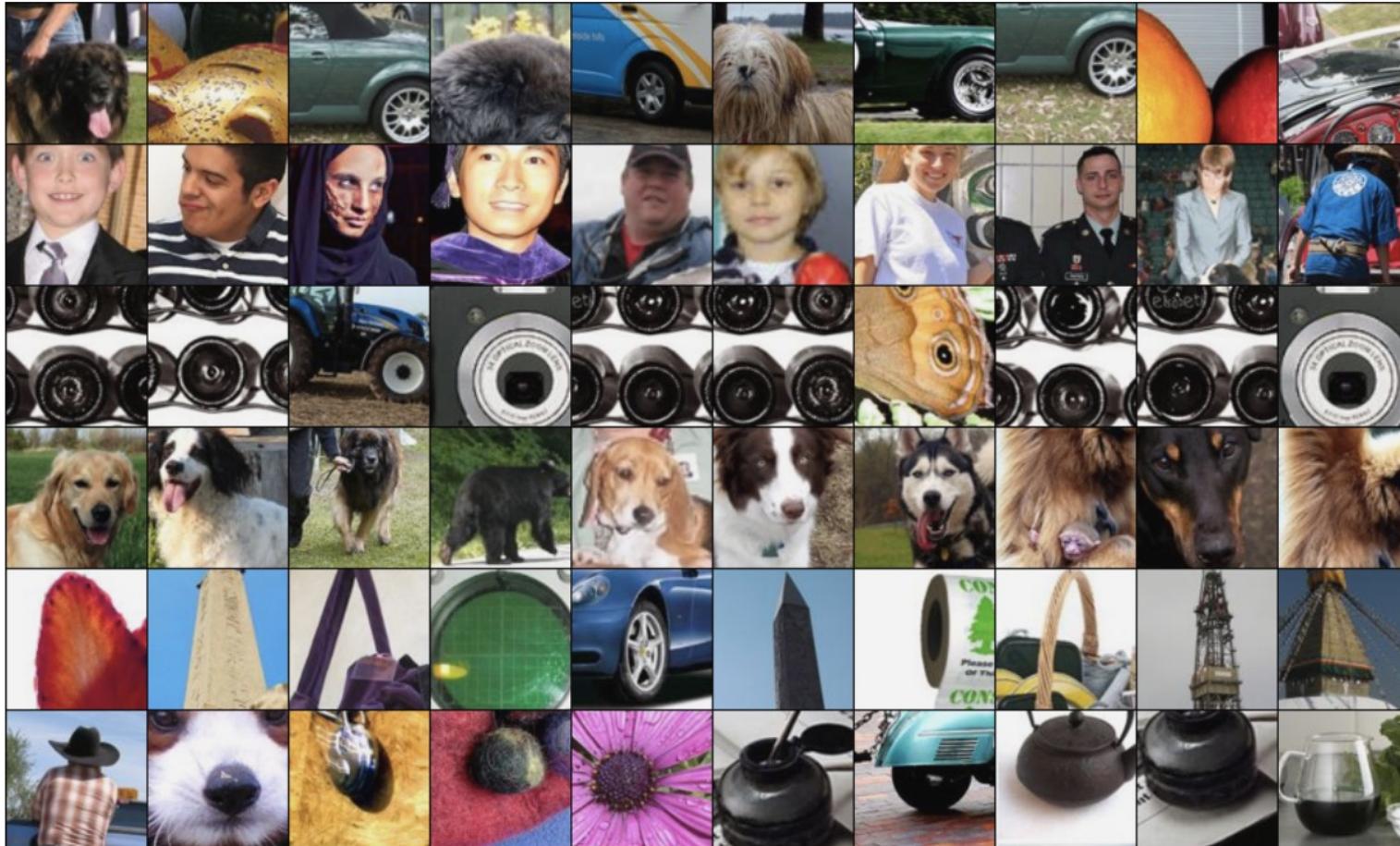
Каждая строка – один фильтр 6-го слоя сети:



Источник: <https://arxiv.org/pdf/1412.6806.pdf>

Максимизация активации

Каждая строка – один фильтр 9-го слоя сети:



Источник: <https://arxiv.org/pdf/1412.6806.pdf>

Максимизация активации

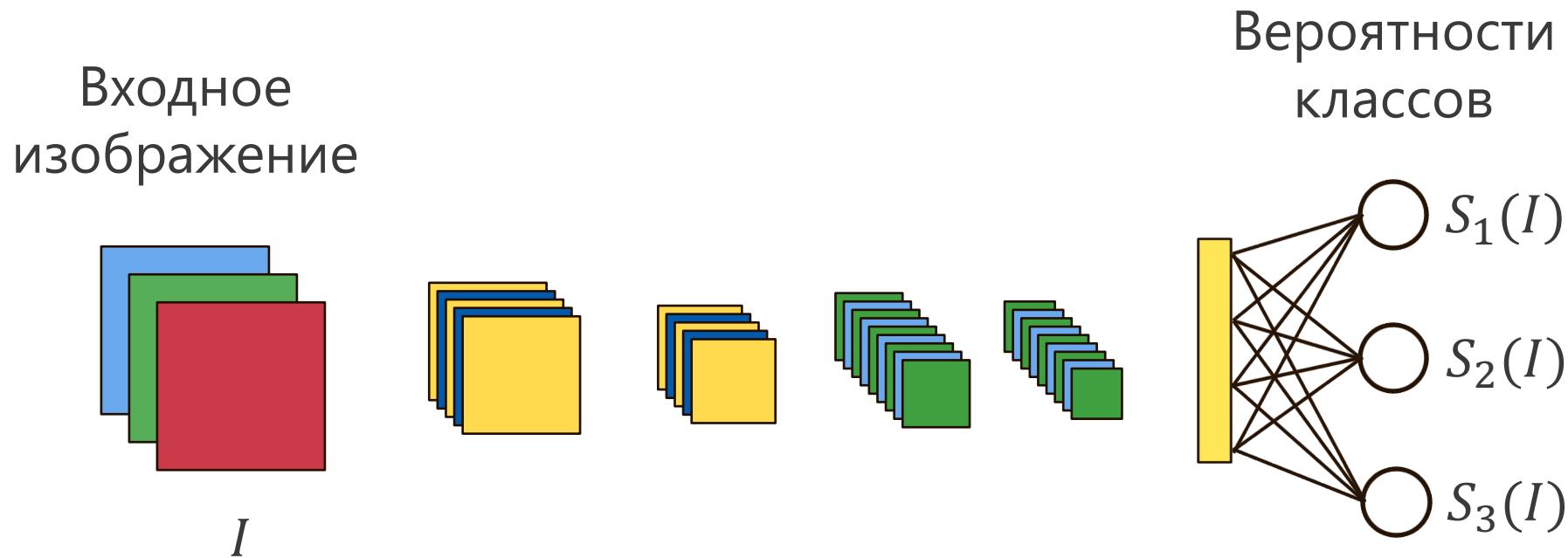
- ▶ На первых слоях сеть находит простые шаблоны
- ▶ С каждым новым слоем поле восприятия растет
- ▶ На глубоких слоях нейроны реагируют на сложные шаблоны

Интерпретация моделей - 2

Максимизация вероятности класса

- ▶ Возьмем модель обученную на ImageNet
- ▶ Выберем класс изображений
- ▶ Найдем такое изображение, которое максимизирует вероятность выбранного класса

Максимизация вероятности класса



- ▶ Изображение I инициализируем случайным шумом
- ▶ «Вероятности» классов $S_i(I)$ берем до Softmax

Максимизация вероятности класса

Находим оптимальную картинку градиентным спуском:

$$I^* = \arg \max_I (\mathbf{S}_c(\mathbf{I}) - \lambda \|I\|_2^2)$$

c — выбранный класс изображения

$\|I\|_2^2$ — регуляризация изображения

λ — настраиваемый параметр

Максимизация вероятности класса



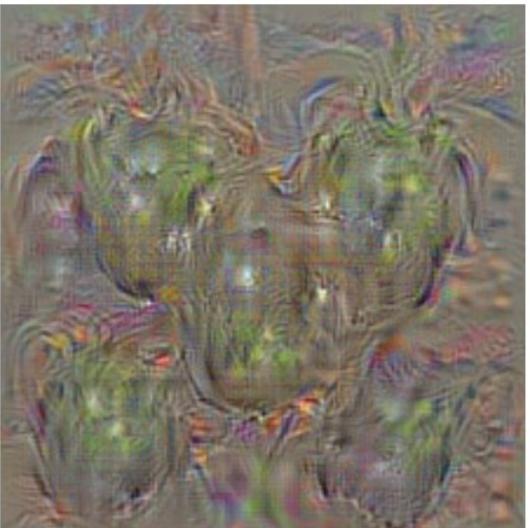
dumbbell



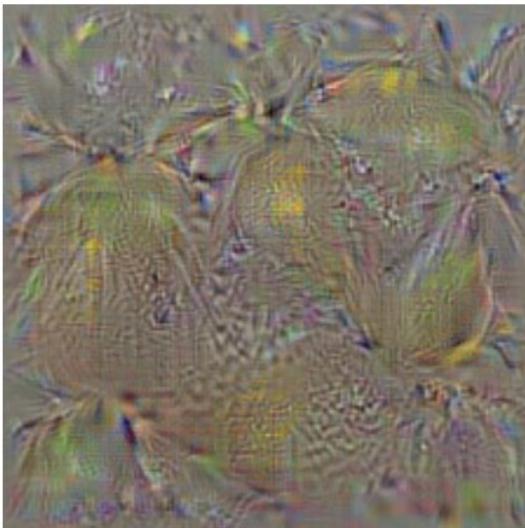
cup



dalmatian



bell pepper



lemon



husky

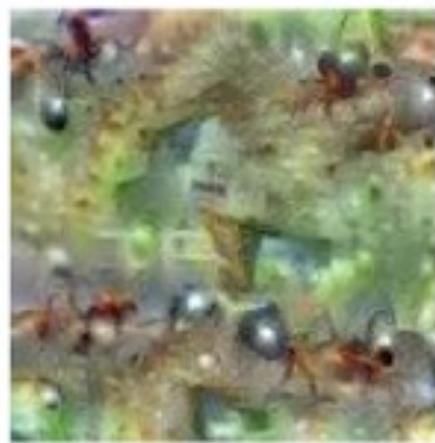
Максимизация вероятности класса



Hartebeest



Measuring Cup



Ant



Starfish



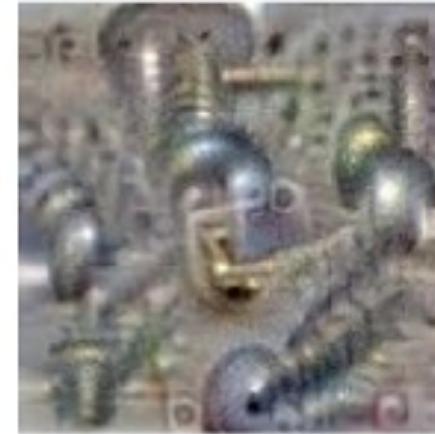
Anemone Fish



Banana



Parachute



Screw

Максимизация активации нейрона

По аналогии:

- ▶ Возьмем модель обученную на ImageNet
- ▶ Выберем нейрон из произвольного слоя
- ▶ Найдем такое изображение, которое максимизирует активацию выбранного нейрона

Максимизация активации нейрона

Находим оптимальную картинку градиентным спуском:

$$I^* = \arg \max_I (\mathbf{a}_i(\mathbf{I}) - \lambda \|I\|_2^2)$$

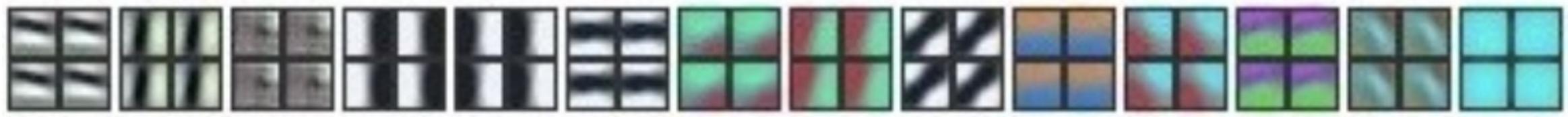
$a_i(I)$ — активация выбранного нейрона

$\|I\|_2^2$ — регуляризация изображения

λ — настраиваемый параметр

Максимизация активации нейрона

1 слой

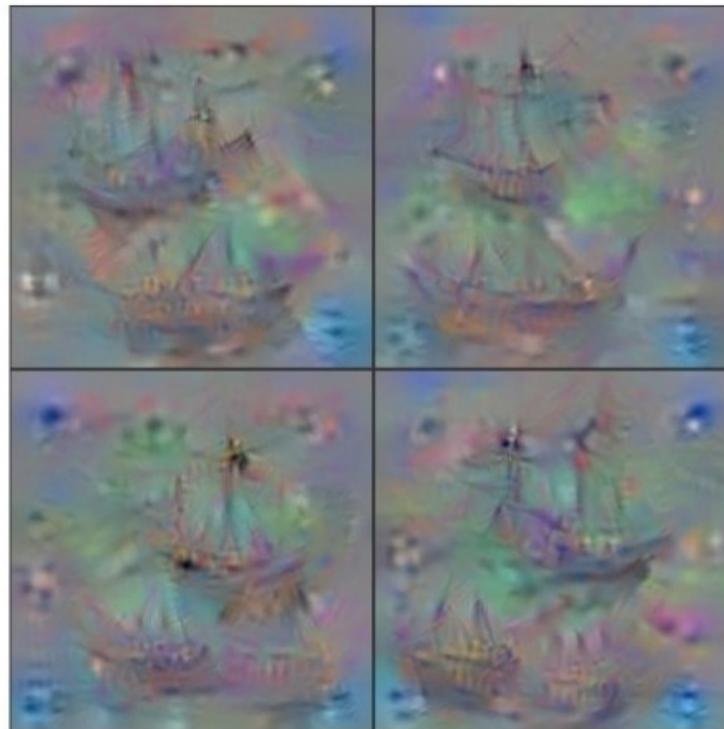


2 слой



Максимизация активации нейрона

8 слой



Выводы

- ▶ Каждое ядро свертки находит определенный шаблон
- ▶ На первых слоях находятся простые шаблоны
- ▶ На глубоких слоях — более сложные
- ▶ Шаблон каждого нейрона можно найти максимизацией его активации по входной картинке методом градиентного спуска

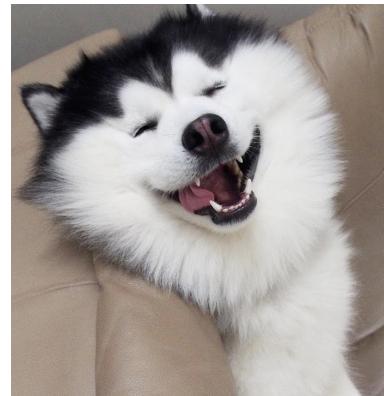
Аугментация данных

Мотивация

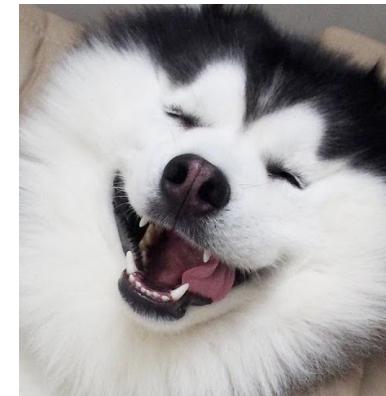
- ▶ Аугментация данных (Augmentation) — искусство увеличение объема данных, используя существующие данные
- ▶ Уменьшает переобучение нейронных сетей
- ▶ Увеличивает качество моделей

Аугментация данных (augmentation)

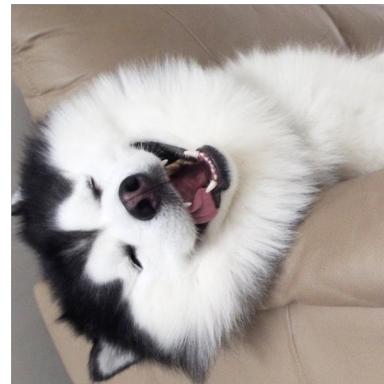
Исходное
изображение



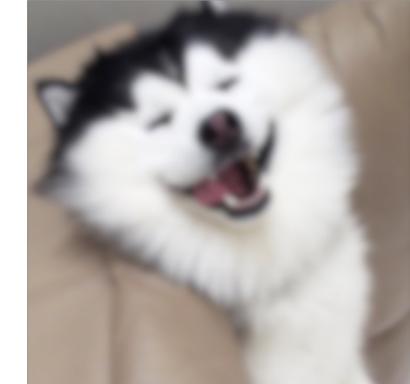
Отражение



Увеличение



Повороты



Размытие

Методы

- ▶ Зеркальные отражения
- ▶ Повороты и сдвиги
- ▶ Масштабирование
- ▶ Изменение цвета и яркости
- ▶ Визуальные эффекты
- ▶ Добавление шума (не только картинки)

Применение

- ▶ Аугментацию заново применяют к порции данных на каждой итерации обучения сети
- ▶ В результате, объекты данных всегда разные
- ▶ Это уменьшает переобучение и повышает качество сети