

Методы искусственного интеллекта

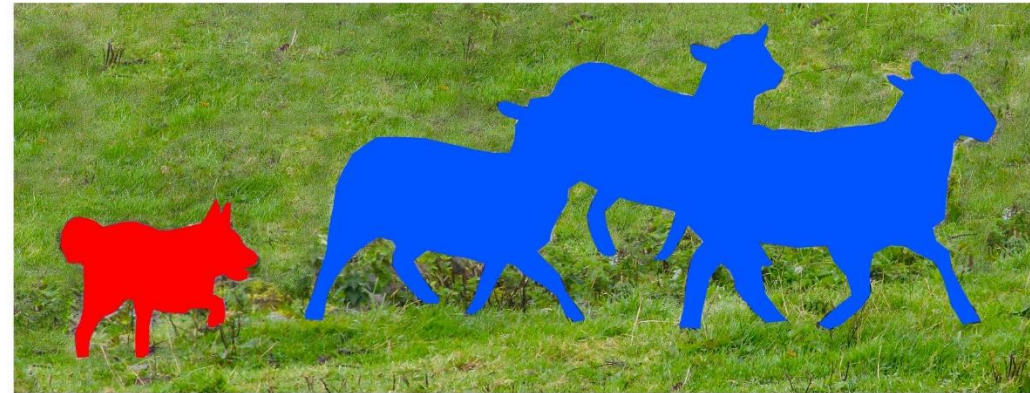
Лекция 5. Компьютерное зрение

Тема 11. Компьютерное зрение

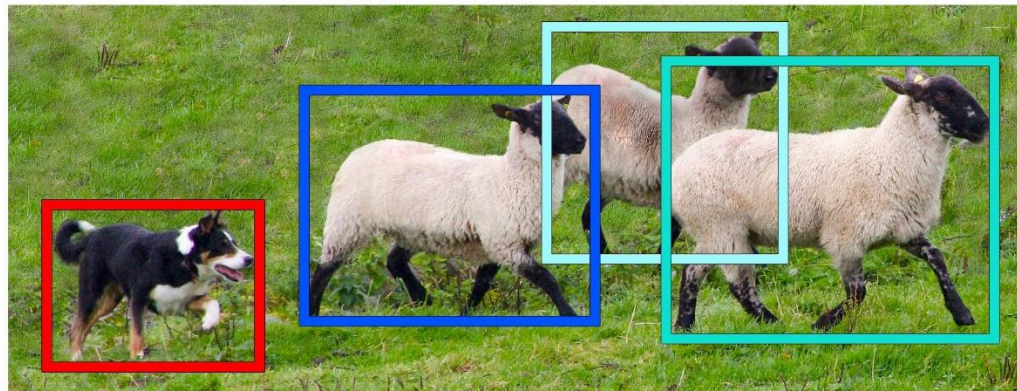
Основные разновидности задач компьютерного зрения



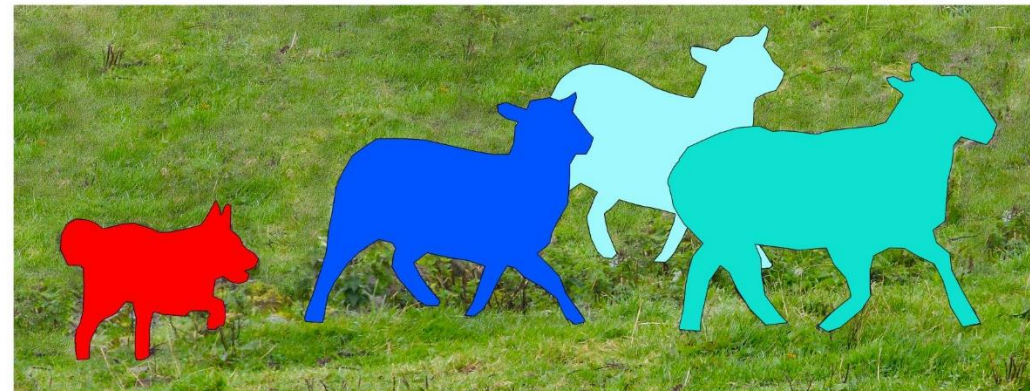
Image Recognition



Semantic Segmentation



Object Detection



Instance Segmentation

Фильтры и свёртка

Линейный фильтр – матрица размером $r \times c$.

- Как правило, квадратная, но не обязательно.

2	4	1
---	---	---

1 x 3

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

3 x 3

1	1	1	1	1
1	2	2	2	1
1	2	4	2	1
1	2	2	2	1
1	1	1	1	1

5 x 5

Фильтры и свёртка

Исходное изображение

90	90	0	0	0	0
0	90	0	0	0	0
0	0	90	90	0	0
0	0	90	90	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Фильтр
(kernel)

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

3 x 3

Результат

40			

$$\begin{aligned} &90 * 1/9 + 90 * 1/9 + 0 * 1/9 + \\ &+ 0 * 1/9 + 90 * 1/9 + 0 * 1/9 + \\ &+ 0 * 1/9 + 0 * 1/9 + 90 * 1/9 = 40 \end{aligned}$$

Фильтры и свёртка

Исходное изображение

90	90	0	0	0	0
0	90	0	0	0	0
0	0	90	90	0	0
0	0	90	90	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Фильтр
(kernel)

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

3 x 3

Результат

40	40		

$$\begin{aligned} &90 * 1/9 + 0 * 1/9 + 0 * 1/9 + \\ &+ 90 * 1/9 + 0 * 1/9 + 0 * 1/9 + \\ &+ 0 * 1/9 + 90 * 1/9 + 90 * 1/9 = 40 \end{aligned}$$

Фильтры и свёртка

Исходное изображение

90	90	0	0	0	0
0	90	0	0	0	0
0	0	90	90	0	0
0	0	90	90	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Фильтр
(kernel)

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

3 x 3

Результат

40	40	20	

$$\begin{aligned} &0 * 1/9 + 0 * 1/9 + 0 * 1/9 + \\ &+ 0 * 1/9 + 0 * 1/9 + 0 * 1/9 + \\ &+ 90 * 1/9 + 90 * 1/9 + 0 * 1/9 = 20 \end{aligned}$$

Фильтры и свёртка

Исходное изображение

90	90	0	0	0	0
0	90	0	0	0	0
0	0	90	90	0	0
0	0	90	90	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Фильтр
(kernel)

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

3 x 3

Результат

40	40	20	10

$$\begin{aligned} &0 * 1/9 + 0 * 1/9 + 0 * 1/9 + \\ &+ 0 * 1/9 + 0 * 1/9 + 0 * 1/9 + \\ &+ 90 * 1/9 + 0 * 1/9 + 0 * 1/9 = 10 \end{aligned}$$

Фильтры и свёртка

Исходное изображение

90	90	0	0	0	0
0	90	0	0	0	0
0	0	90	90	0	0
0	0	90	90	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Фильтр
(kernel)

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

3 x 3

Результат

40	40	20	10
30			

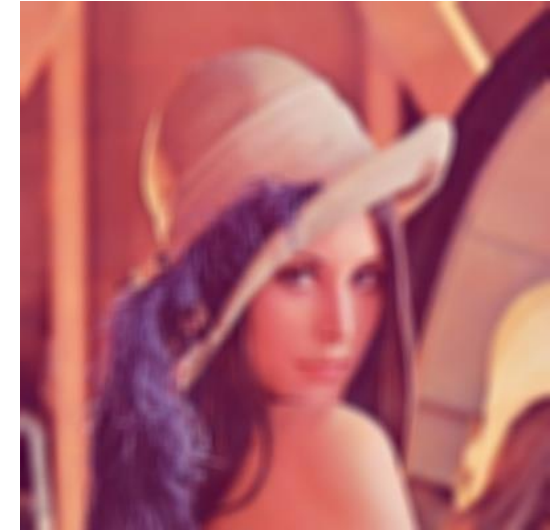
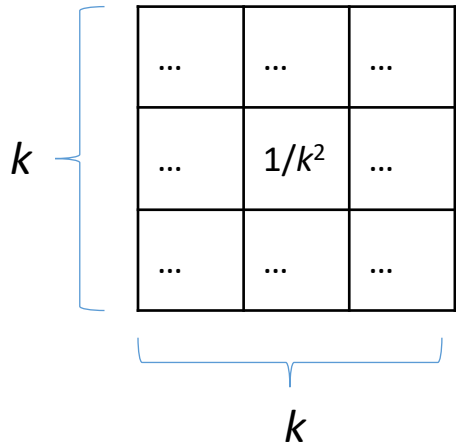
$$\begin{aligned} &0 * 1/9 + 90 * 1/9 + 0 * 1/9 + \\ &+ 0 * 1/9 + 0 * 1/9 + 90 * 1/9 + \\ &+ 0 * 1/9 + 0 * 1/9 + 90 * 1/9 = 30 \end{aligned}$$

Примеры фильтров. Размытие (blur)

Исходное
изображение

$k = 3$

$k = 10$



Примеры фильтров. Тиснение (emboss)

Исходное
изображение



Тиснение
(emboss)



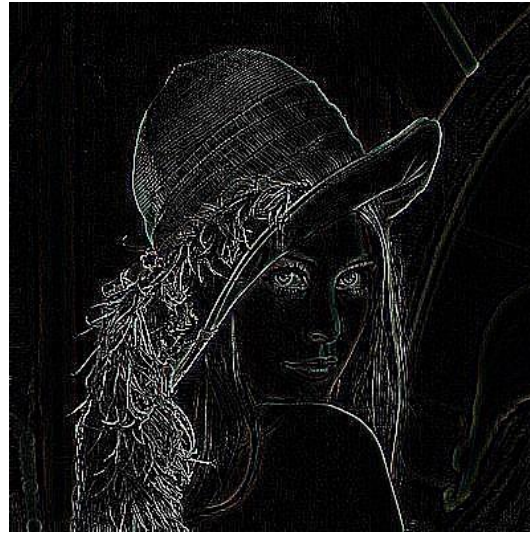
-2	-1	0
-1	1	1
0	1	2

Примеры фильтров. Выделение границ

Исходное
изображение



Выделение
контура



Фильтр Собеля
(снизу)



Фильтр Собеля
(слева)



-1	-1	-1
-1	8	-1
-1	-1	-1

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

1	0	-1
2	0	-2
1	0	1

Пример задачи. Подсчет прямоугольников

0	0	0	0	0	0
0	1	1	0	0	0
0	1	1	0	0	0
0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	1	1

Пример задачи. Подсчет прямоугольников. Свёртка

0	0	0	0	0	0
0	1	1	0	0	0
0	1	1	0	0	0
0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	1	1

*

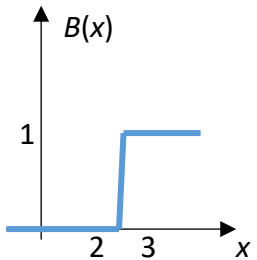
-1	-1
-1	3

=

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

Пример задачи. Подсчет прямоугольников. Свёртка + активация

$$B(\begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 \\ \hline \end{array} * \begin{array}{|c|c|} \hline -1 & -1 \\ \hline -1 & 3 \\ \hline \end{array}) = \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$



Пример задачи. Подсчет прямоугольников.
Свёртка + активация + сумма

$$\text{sum}(\begin{array}{|c|c|c|c|c|}\hline 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline\end{array}) = 2$$

Пример задачи. Подсчет прямоугольников

Итоговый алгоритм:

1. Осуществить свёртку с определенным фильтром.
2. Применить пороговую функцию активации.
3. Осуществить суммирование.

-1	-1
-1	3

Плюсы:

- Задача может быть сведена к последовательности свёрток, применений активации и арифметических операций (суммирование)

Проблемы:

- Как выбирать эти фильтры, пороги активации?
- А если задача сложнее и предполагает длинную цепочку преобразований?

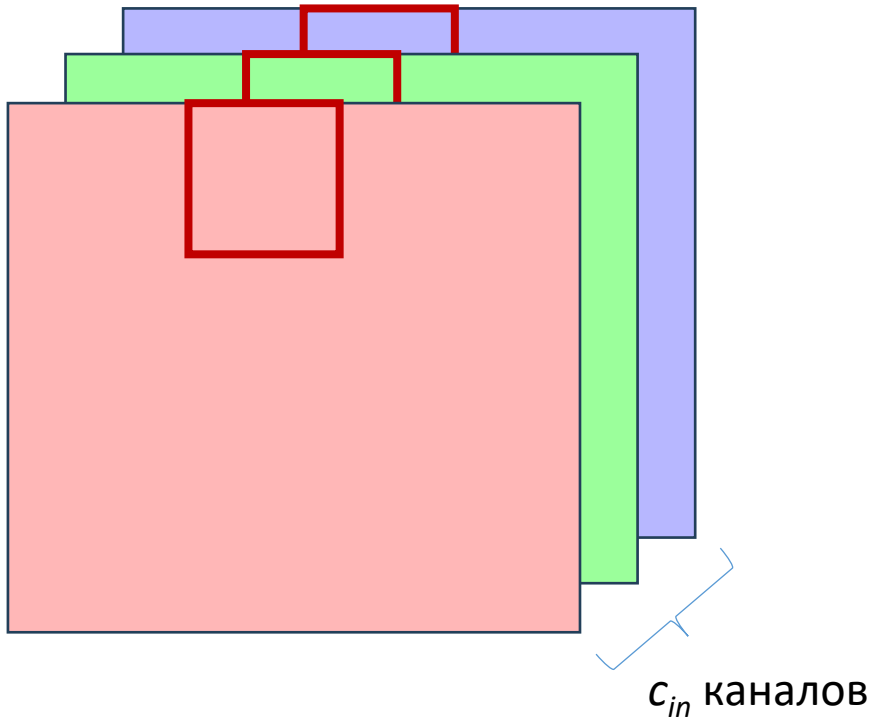
Обучаемые фильтры

- **Идея.** Почему бы не сделать элементы фильтра *настраиваемыми*, чтобы они *автоматически* подбирались таким образом, чтобы добиться наилучшего результата на заданном *наборе данных*?
 - Сверточные (конволюционные) слои (convolutional layers) – основа современных сетей для обработки изображений

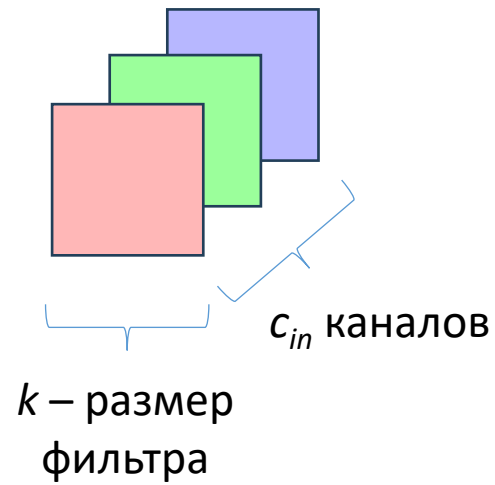
w_{11}	w_{12}	w_{13}
w_{21}	w_{22}	w_{23}
w_{31}	w_{32}	w_{33}

Свёрточные слои (Conv2D)

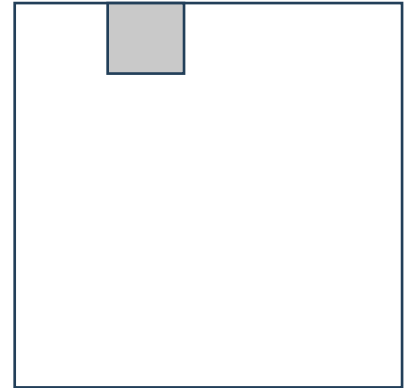
Входные
данные



Фильтр



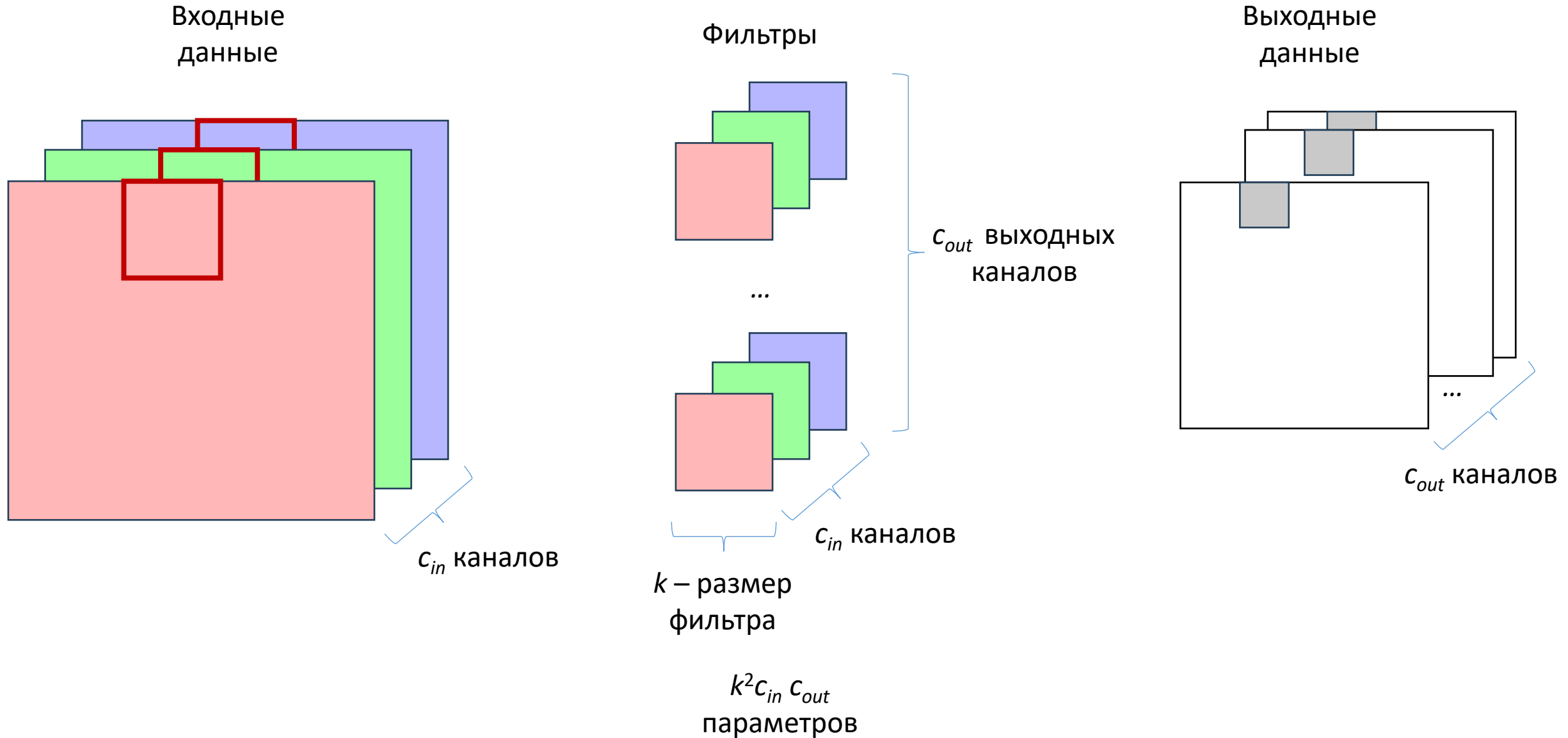
Выходные
данные



$$r_{u,v} = \sum_{c=1}^{c_{in}} \sum_{i=1}^k \sum_{j=1}^k x_{c,i,j} w_{c,u+i-1,v+j-1}$$

$k^2 c_{in}$ параметров

Свёрточные слои (Conv2D)



Свёрточные слои (Conv2D). Размер шага (stride)

Размер шага
(stride)

90	90	0	0	0	0
0	90	0	0	0	0
0	0	90	90	0	0
0	0	90	90	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Свёрточные слои (Conv2D). Дополнение (padding)

Valid padding

Используются только «настоящие» значения
входного изображения

42	13	10	8
14	3	11	43
5	7	12	42
7	11	42	1

Размер выходного изображения:

- при шаге 1: 2 x 2
- при шаге 2 и больше: 1 x 1

$$O = \left\lfloor \frac{I - K}{S} \right\rfloor + 1$$

Same padding

Дополняется нулями так, чтобы
при шаге 1 выходной размер был такой же, как и входной

0	0	0	0	0	0
0	42	13	10	8	0
0	14	3	11	43	0
0	5	7	12	42	0
0	7	11	42	1	0
0	0	0	0	0	0

Размер выходного изображения:

- при шаге 1: 4 x 4
- при шаге 2: 2 x 2

$$O = \left\lfloor \frac{I + P_L + P_R - K}{S} \right\rfloor + 1$$

Свёрточные слои. Оценка размера выходного изображения

- Пусть:
 - C – количество каналов
 - W – ширина
 - H – высота
 - Индексы:
 - I – вход, O – выход, K – фильтр, S – шаг, P – выравнивание (суммарное, с двух сторон)
- Количество параметров слоя:
$$W_K * H_K * C_I * C_O$$
- Размер выходного изображения:

$$W_O = \left\lfloor \frac{W_I + W_P - W_K}{W_S} \right\rfloor + 1$$

$$H_O = \left\lfloor \frac{H_I + H_P - H_K}{H_S} \right\rfloor + 1$$

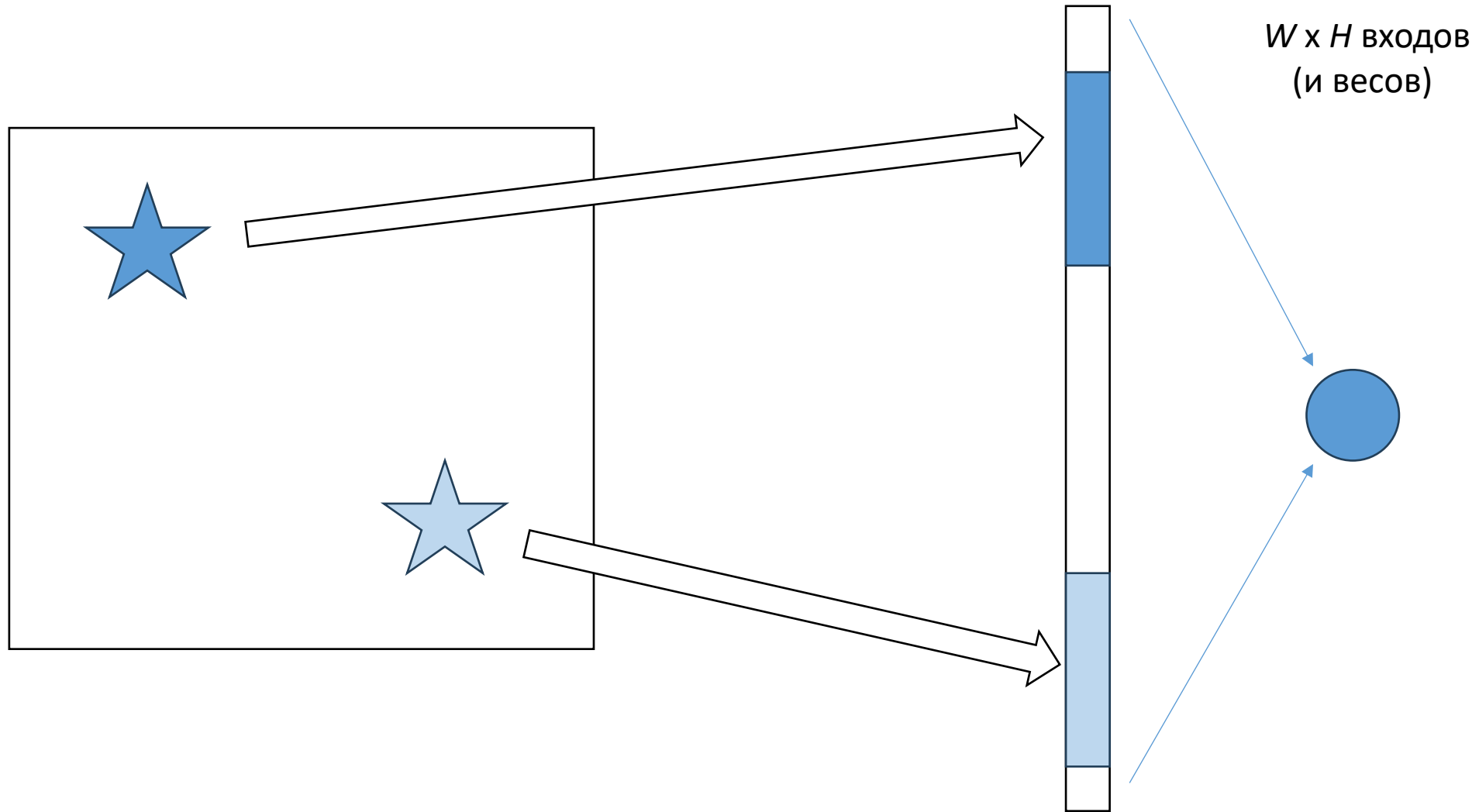
Свёрточные слои. Вопрос 1

- Входное изображение 100 x 100 x 3, свёртка с ядром размера 3 (valid padding, шаг 1), на выходе должно быть 10 каналов.
 - Сколько параметров?
 - Какой будет размер выходного изображения?

Свёрточные слои. Вопрос 2

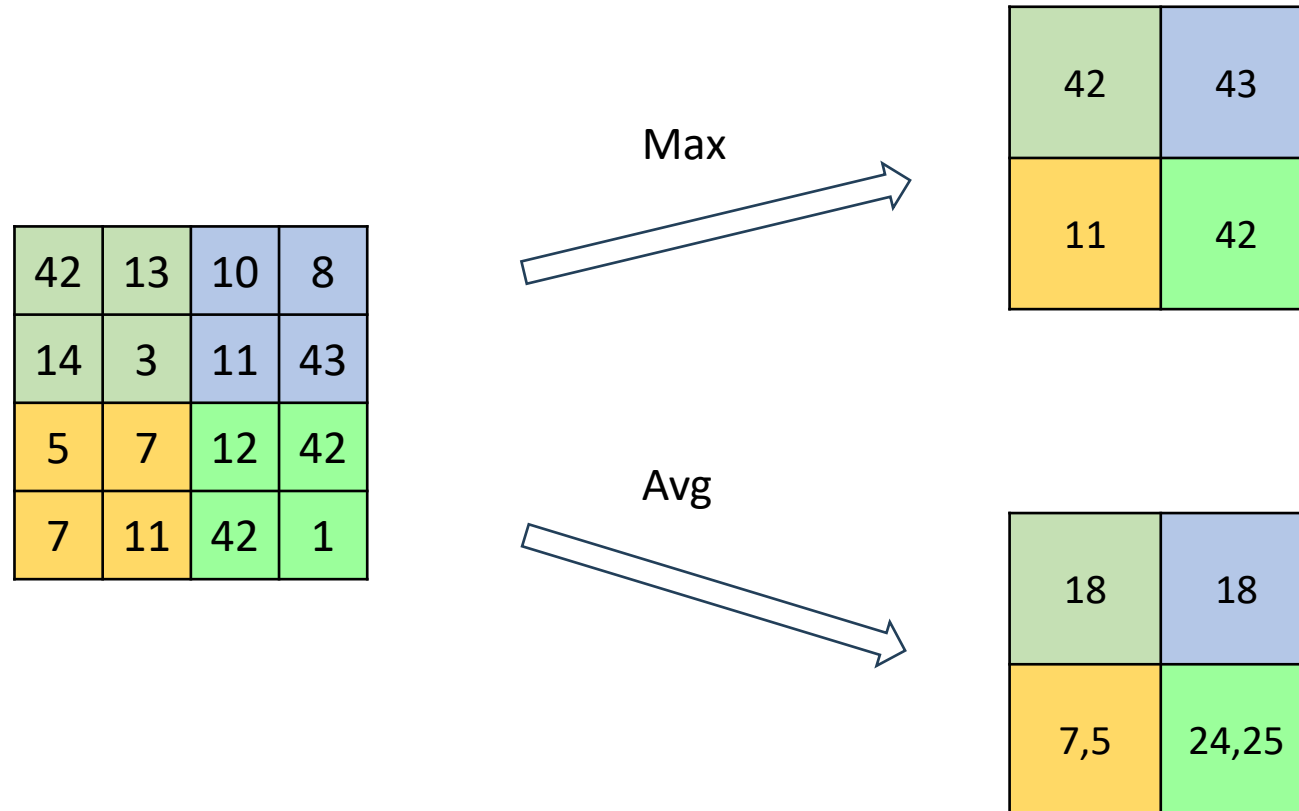
- Входное изображение 100 x 100 x 3, свёртка с ядром размера 3 (valid padding, шаг 3), на выходе должно быть 10 каналов.
 - Сколько параметров?
 - Какой будет размер выходного изображения?

Почему нельзя просто обычный полносвязный слой?



Подвыборка (пулинг). Max pooling, average pooling

С каждым каналом:

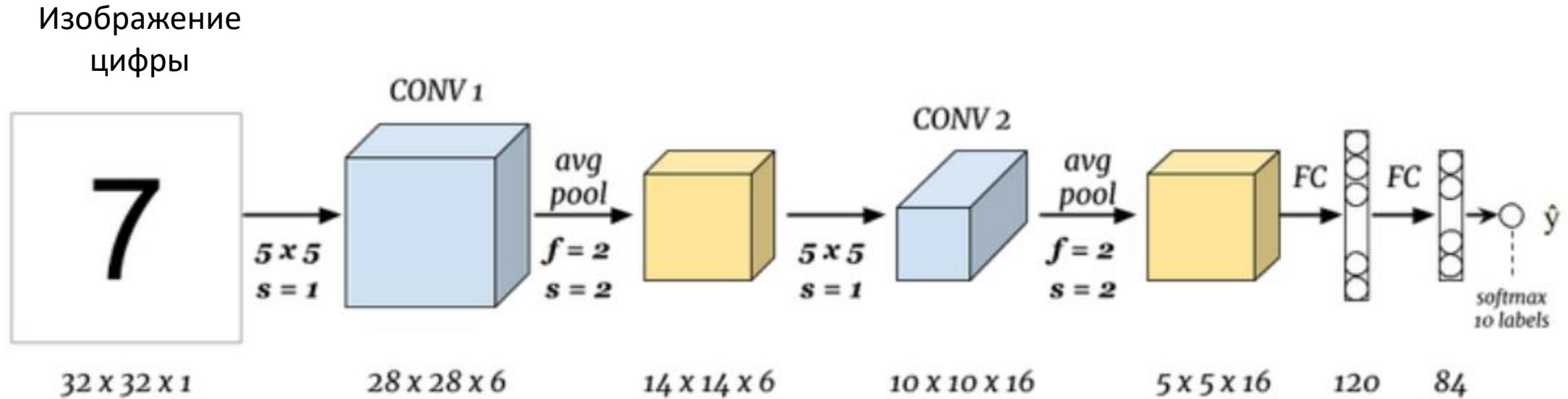


Размер фильтра (ядра) и шаг тоже могут быть определены, но, обычно, размер фильтра совпадает с шагом, а выравнивание valid.

LeNet-5

1998 год
~60 тыс. параметров

Активация: tanh



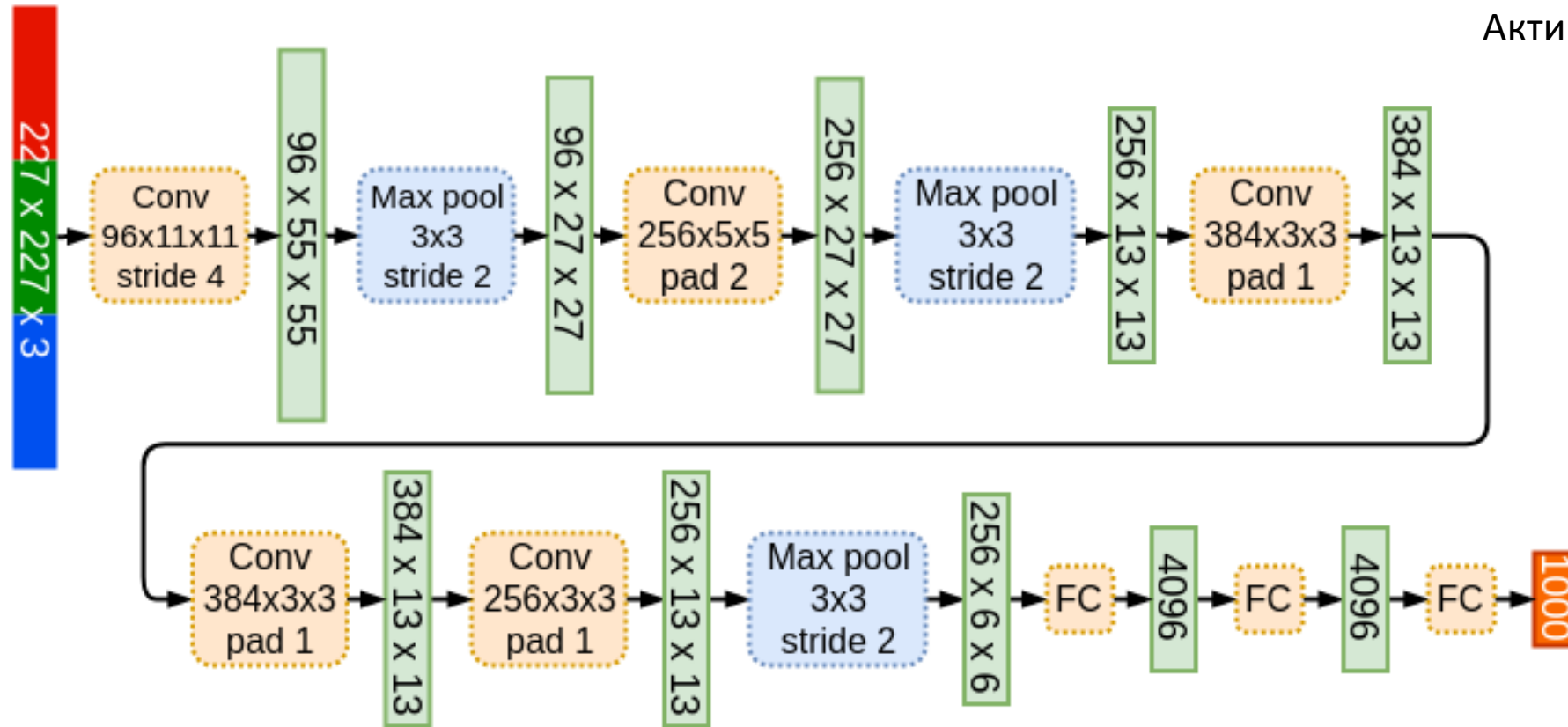
Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. (1998). "Gradient-based learning applied to document recognition".
Proceedings of the IEEE. 86 (11): 2278–2324. doi:10.1109/5.726791. S2CID 14542261.

Картинка: <https://www.philschmid.de/getting-started-with-cnn-by-calculating-lenet-layer-manually>

AlexNet (и ImageNet)

2012 год
~62 млн. параметров

Активация: ReLU



Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). "ImageNet classification with deep convolutional neural networks" (PDF). Communications of the ACM. 60 (6): 84–90.

Картинка: <https://www.oreilly.com/library/view/advanced-deep-learning/9781789956177/b2258aa6-2c18-449c-ac00-939e812f5a4a.xhtml>

AlexNet

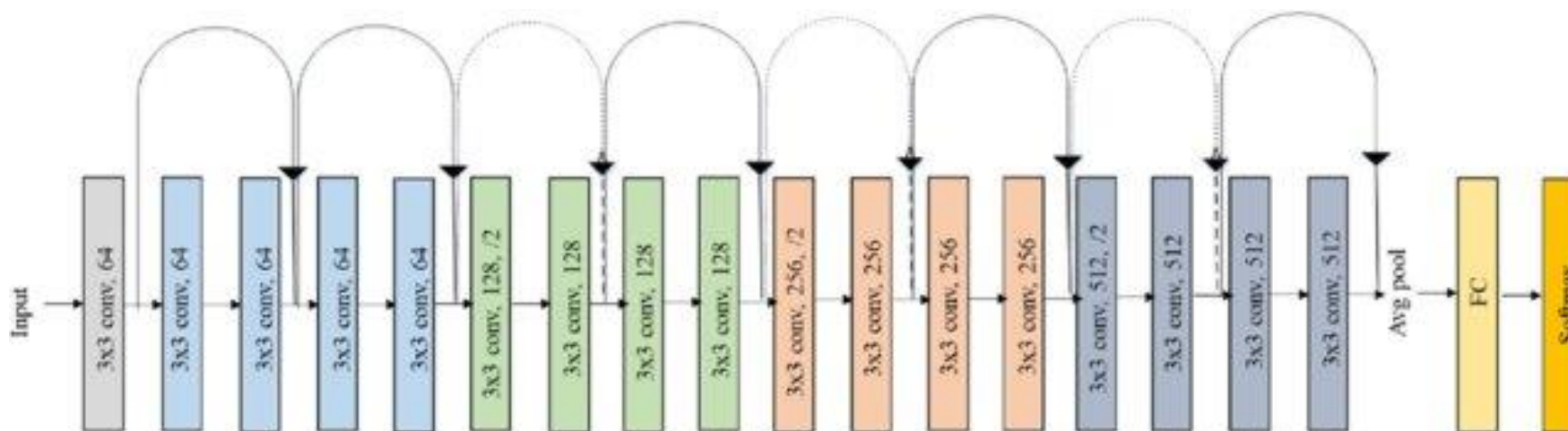
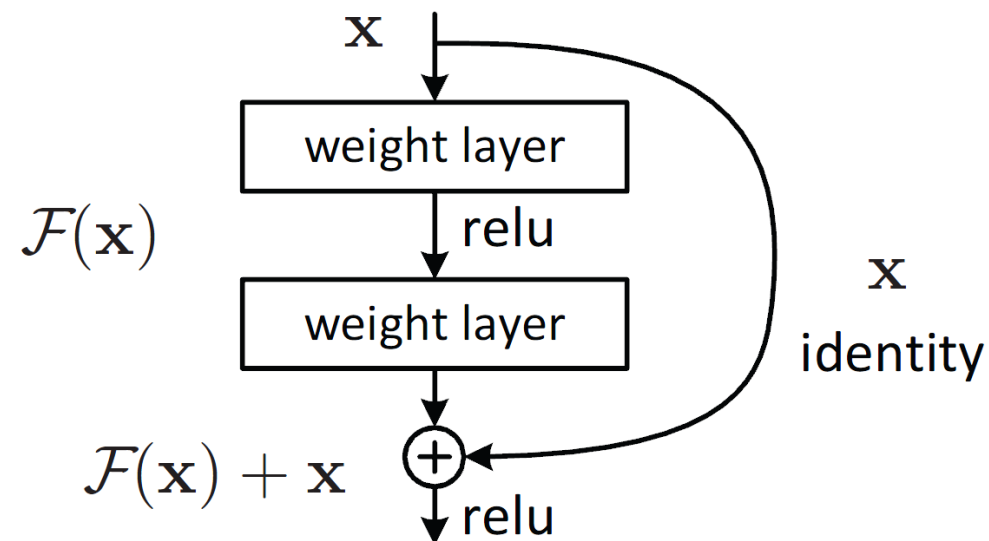


Figure 3: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images.

ResNet. «Остаточные» сети

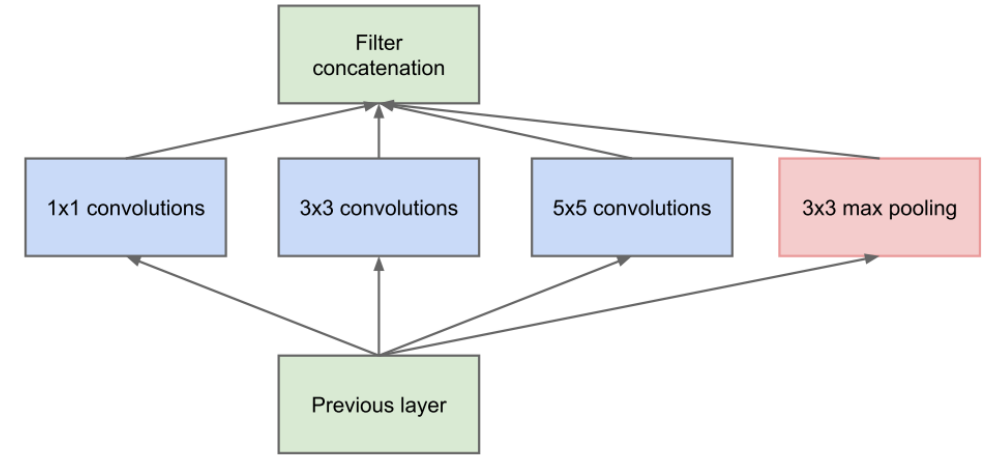
2016 год
~11-25 млн. параметров

- Проблемы с обучением очень глубоких сетей
- Предлагаемое решение:
 - Skip-connections («обходные» связи)
- Семейство: ResNet-18, ResNet-35, ResNet-50, ...

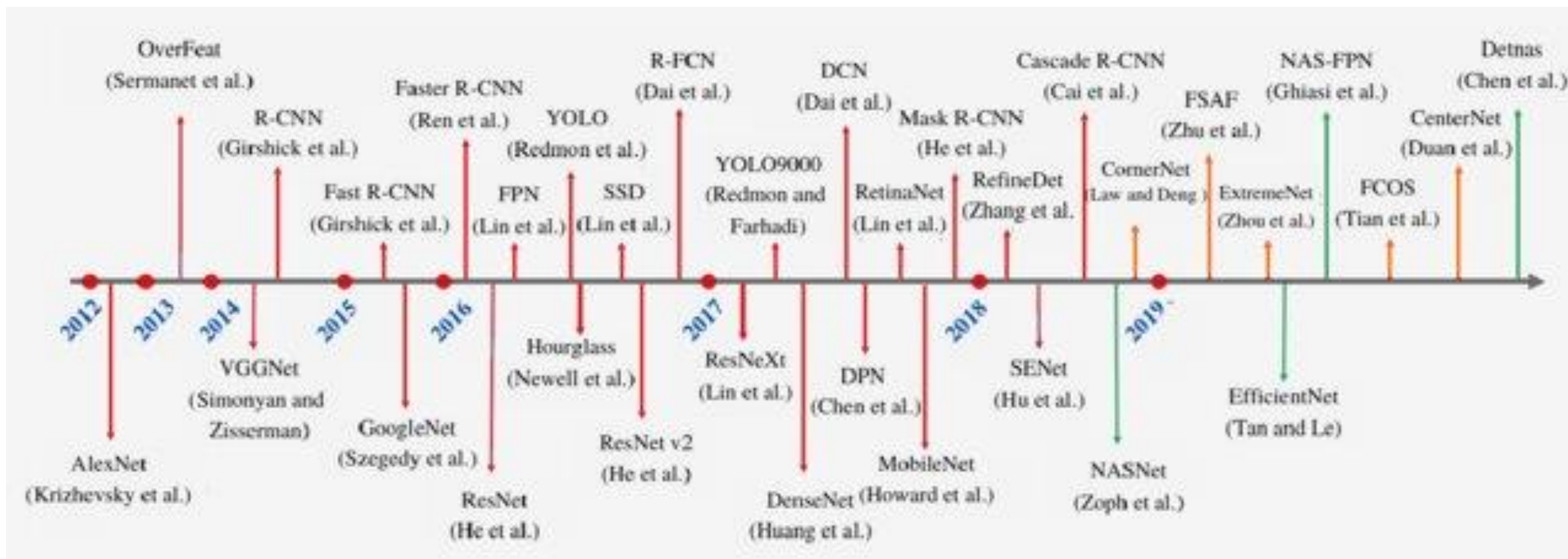


Другие направления улучшения

- Зачем выбирать размер фильтра – давайте сразу все!
 - Inception
- Упрощенные свёртки (depth-wise separable convolutions):
 - MobileNet, EfficientNet
- Пакетная нормализация:
 - Много где (те же ResNet, EfficientNet, MobileNet)

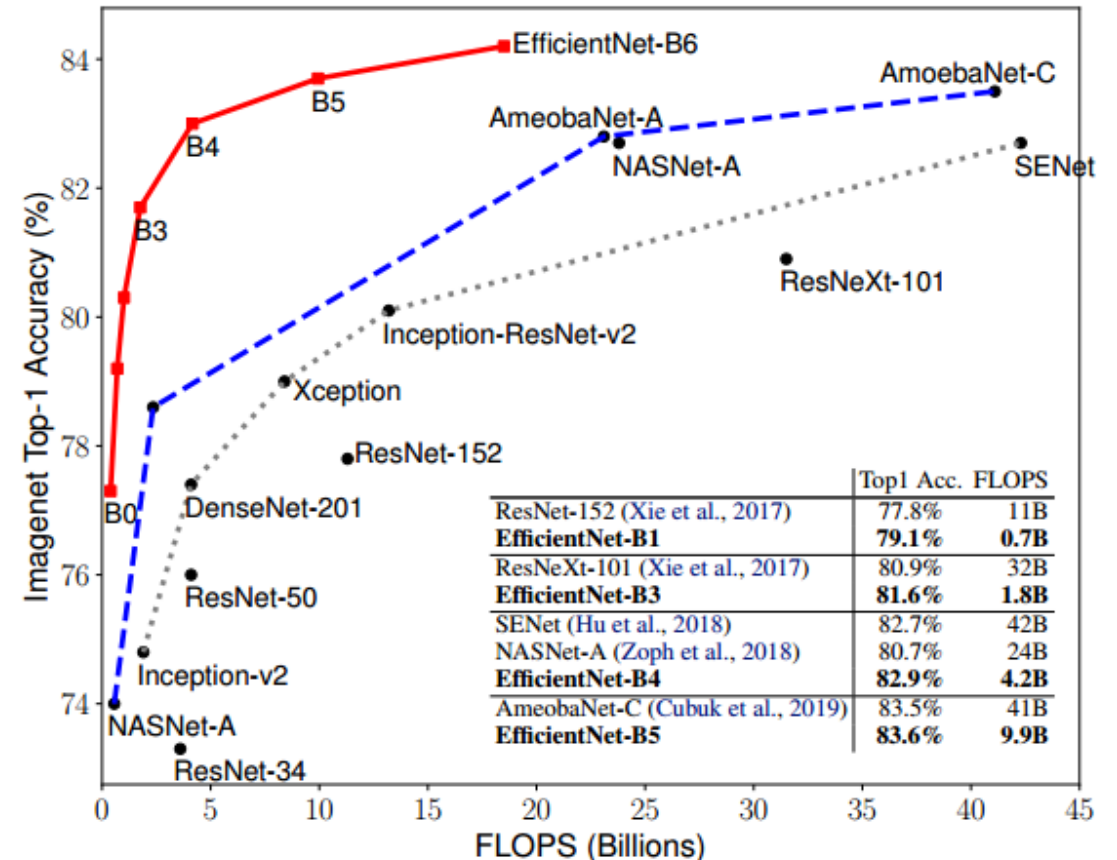
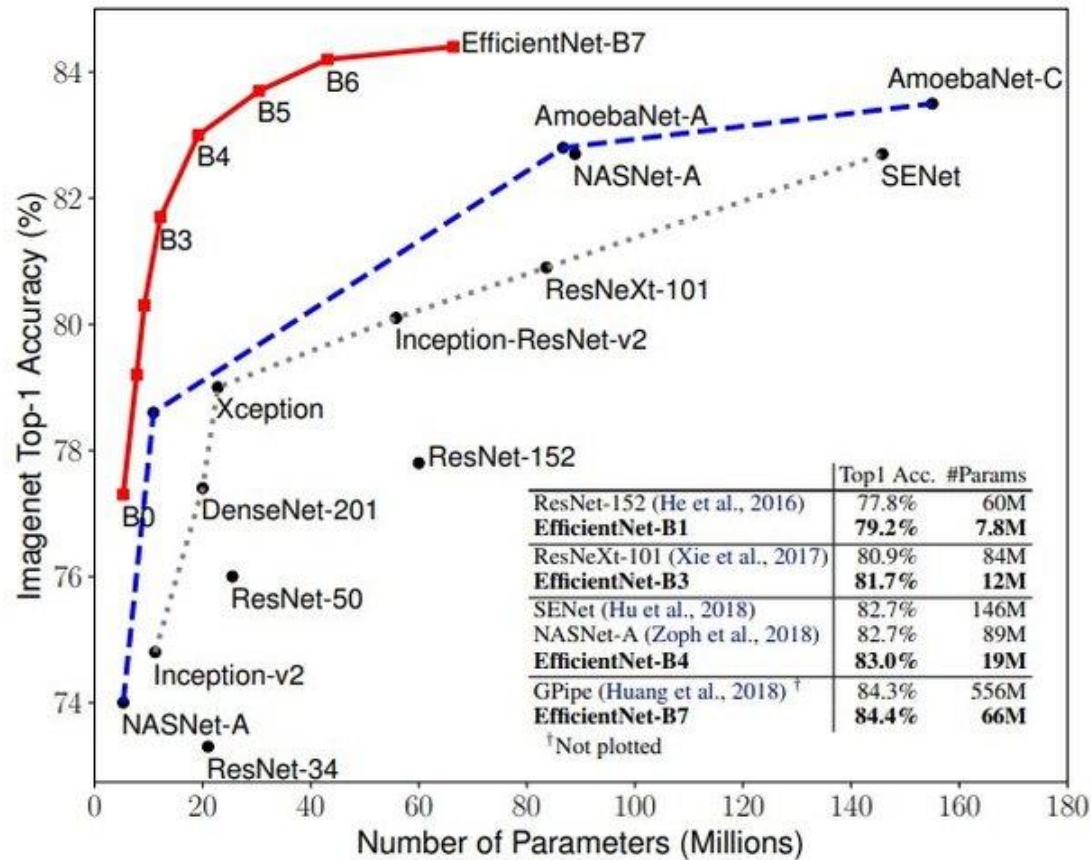


Краткая эволюция нейронных сетей для компьютерного зрения



Nain, Megha & Sharma, Shilpa & Sandeep, Chaurasia. (2021). Safety and Compliance Management System Using Computer Vision and Deep Learning. IOP Conference Series: Materials Science and Engineering. 1099. 012013. 10.1088/1757-899X/1099/1/012013..

Основные архитектуры по качеству и сложности (на 2019 год)



«Тонкости обучения». Аугментации

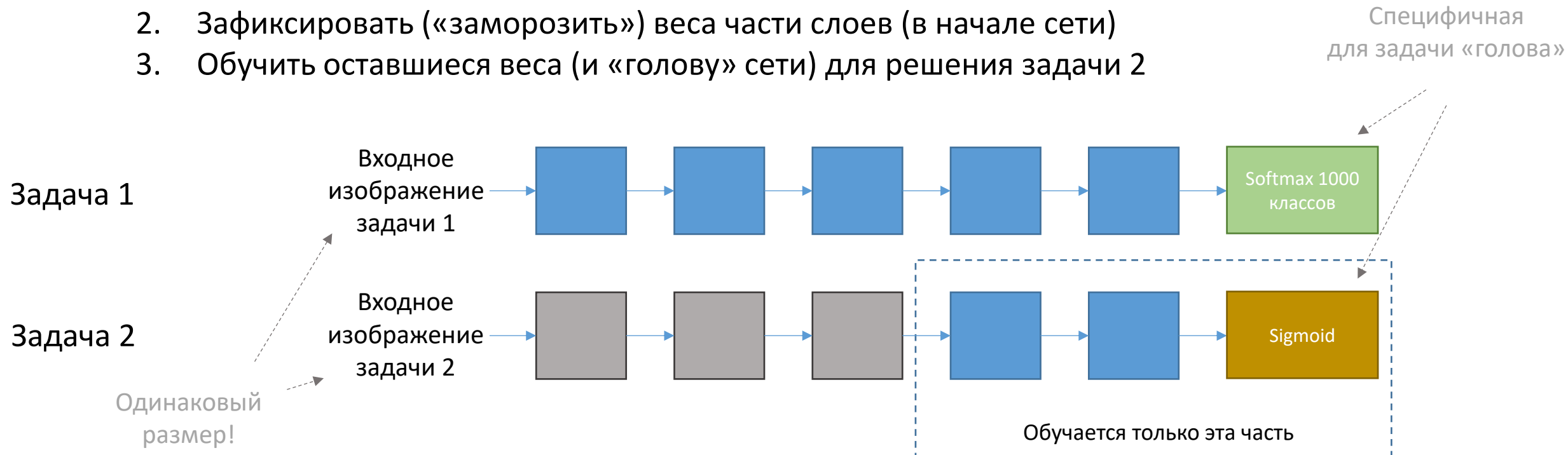
- Проблема:
 - Данных, как правило, мало
 - Модели сложно отличить важное от неважного (особенно, когда данных мало)
 - Шумы камеры, цветовая температура, наклон объекта и т.п.
- Потенциальное решение:
 - Расширять обучающую выборку с помощью преобразований, изменяющих внешний вид распознаваемых объектов, но не изменяющих их «суть»

«Тонкости обучения». Примеры аугментаций

- Геометрические:
 - Вращение (rotate)
 - Отражение (flip)
 - Обрезка (crop)
- Изменение гаммы (grayscale, color jitter)
- Шумы (noise)

«Тонкости обучения». Обучение с переносом (transfer learning)

- Ситуация:
 - Задача 1: Много данных, есть хорошие модели и архитектуры
 - Задача 2: Мало данных, задача специфична, но (в определенном смысле) близка задаче 1
- Общая идея решения:
 1. Обучить модель на данных задачи 1
 2. Зафиксировать («заморозить») веса части слоев (в начале сети)
 3. Обучить оставшиеся веса (и «голову» сети) для решения задачи 2



Сегментация. U-Net

2015 год

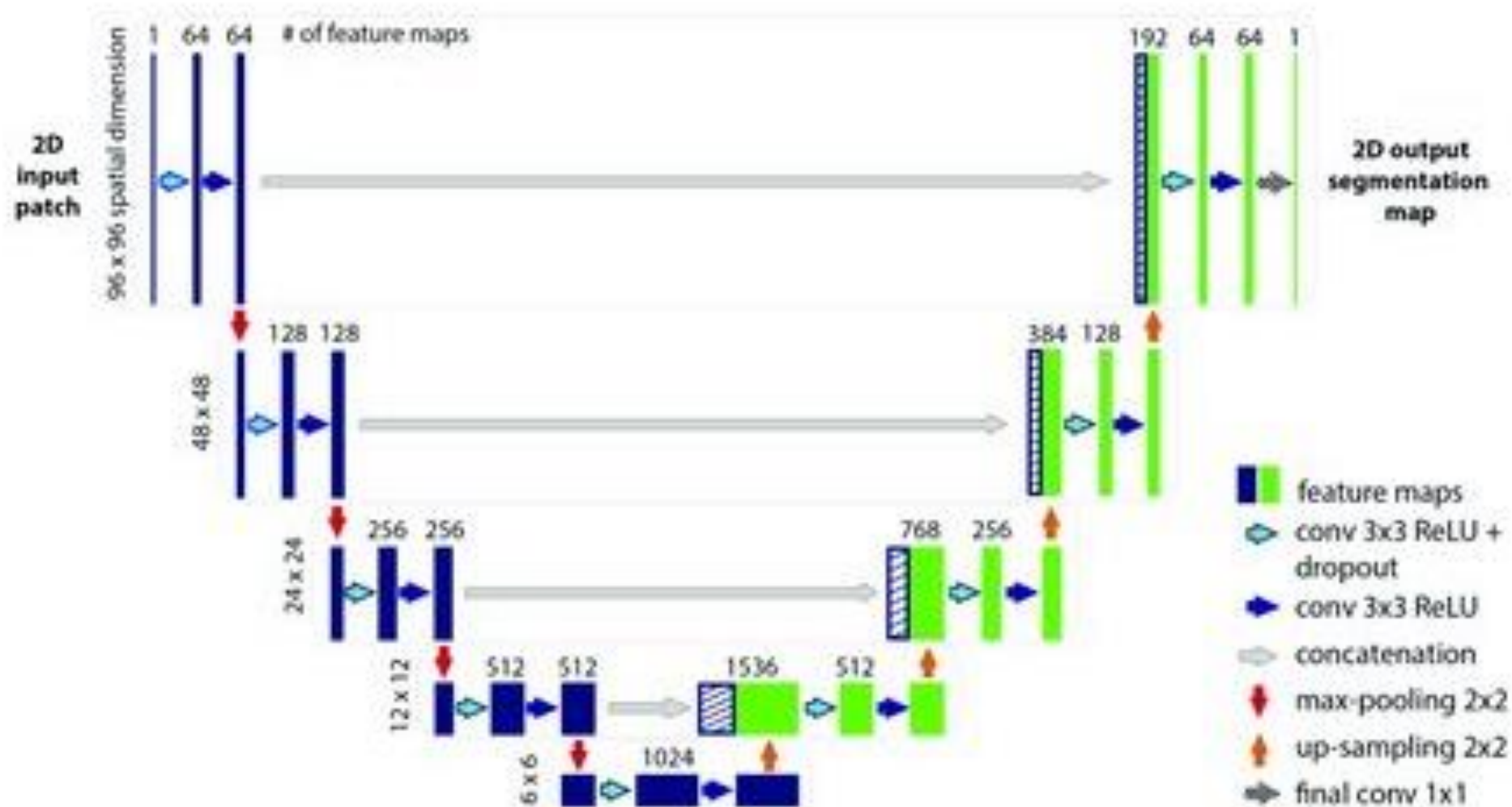


Рисунок: Livne, M., Rieger, J., Aydin, O. и др. (2019). A U-Net Deep Learning Framework for High Performance Vessel Segmentation in Patients With Cerebrovascular Disease. Frontiers in Neuroscience. 13. 10.3389/fnins.2019.00097

Резюме

- Свёртка с фильтром – важный строительный блок цифровой обработки изображений, в том числе, компьютерного зрения
 - Идея свёрточных нейронных сетей – обучаемые фильтры
- Основные строительные блоки свёрточных нейронных сетей:
 - Свёрточные слои:
 - Размер фильтра, количество выходных каналов, шаг, дополнение
 - Подвыборка (pooling)
 - Пакетная нормализация
 - ReLU активация
- Обучение:
 - Аугментации
 - Обучение с переносом (transfer learning)
- Архитектуры:
 - Классификация:
 - LeNet-5, AlexNet, ResNet
 - Сегментация:
 - U-Net

Литература

- Основное:
 - Dive into Deep Learning: <http://d2l.ai>
- Дополнительно:
 - Рассел С., Норвиг П. Искусственный интеллект. Современный подход. 4-е издание
 - Coursera/DeepLearning.AI: Deep Learning Specialization (Andrew Ng)
 - Масса источников, только вводи ключевые слова