

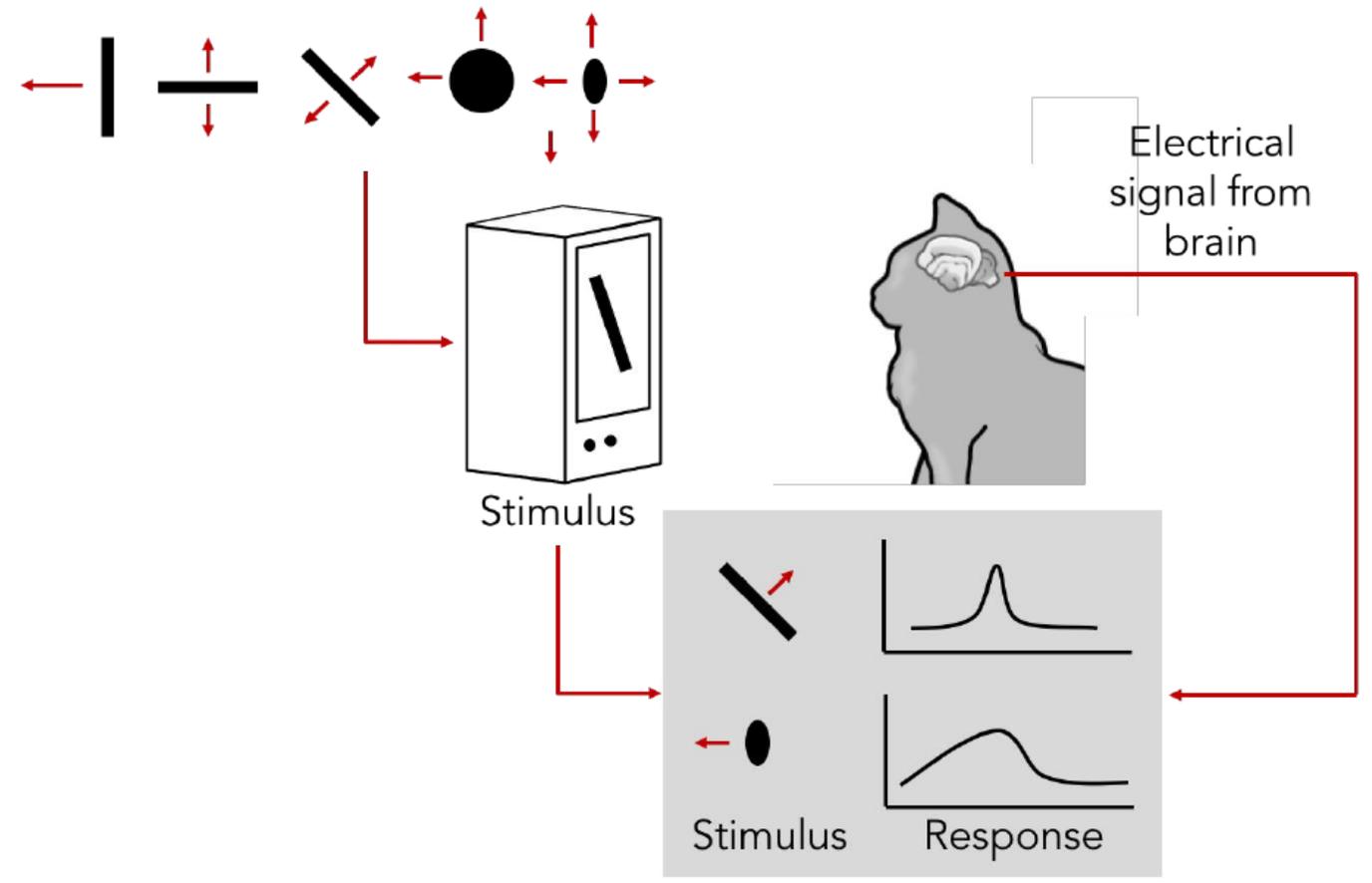
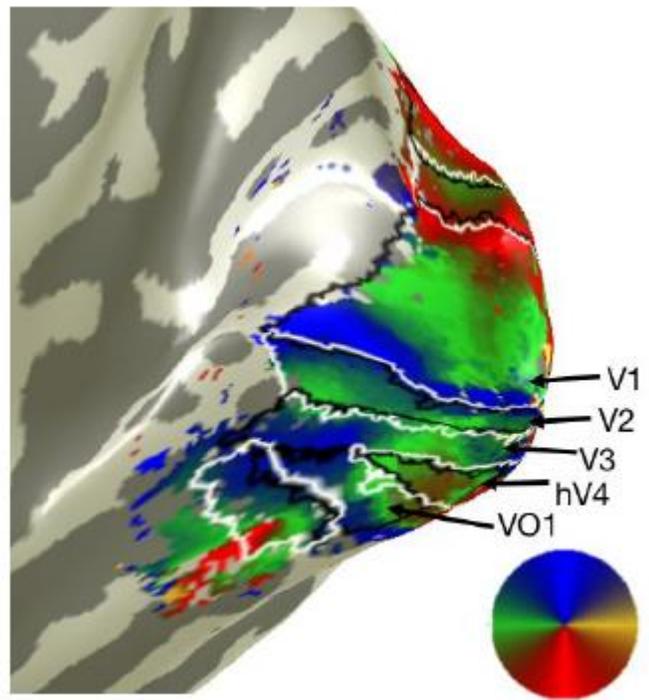
Сверточные нейронные сети

- Принцип работы сверточных нейронных сетей
- Где и как можно найти подходящую модель для решения задачи
- Задание 3

Немного истории

1959 Hubel & Wiesel

Реакции отдельных нейронов зрительной коры на раздражители



[Cat image](#) by CNX OpenStax is licensed under CC BY 4.0; changes made

Иерархическая организация зрительной системы

Простые нейроны:

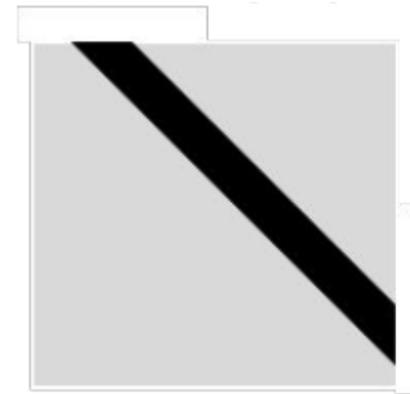
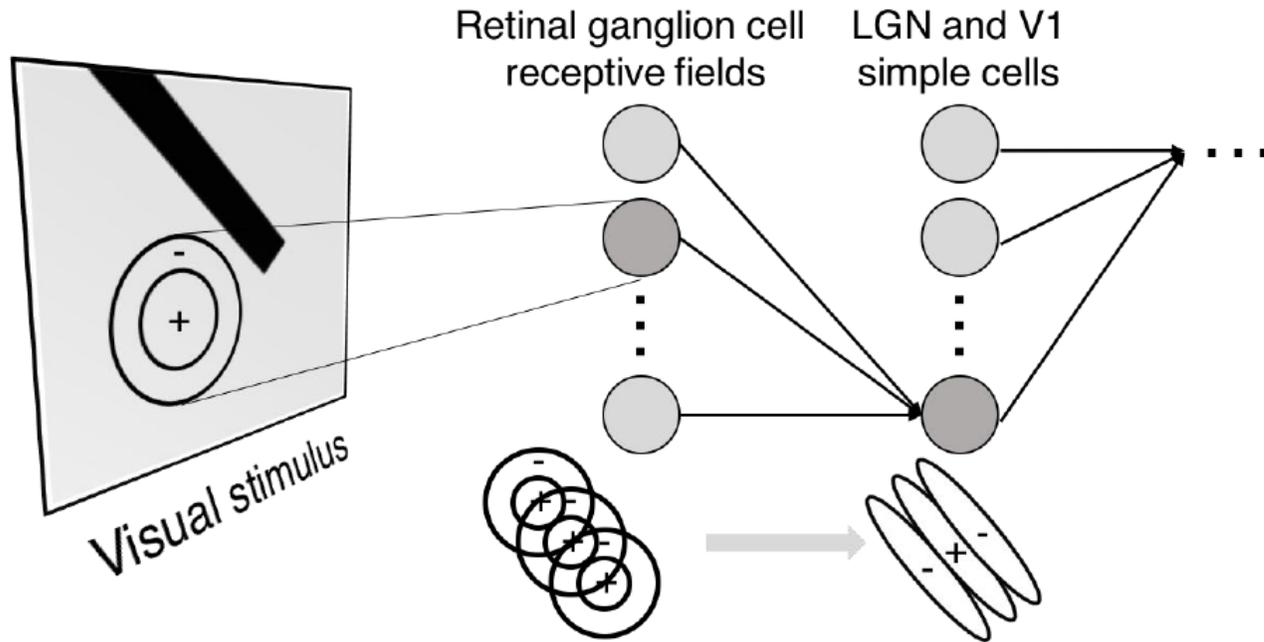
Реагируют на пятна света и грани изображения

Комплексы нейронов:

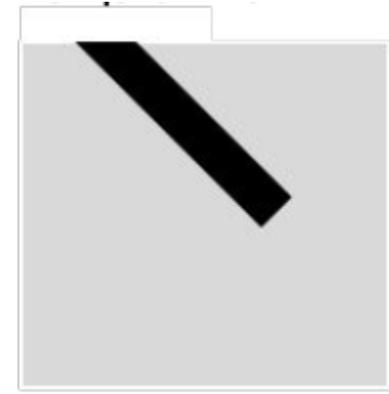
Реагируют на направление света и движения

Сложные комплексы нейронов:

Реагируют на движения «концевых элементов»

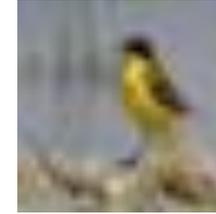


Нет реакции

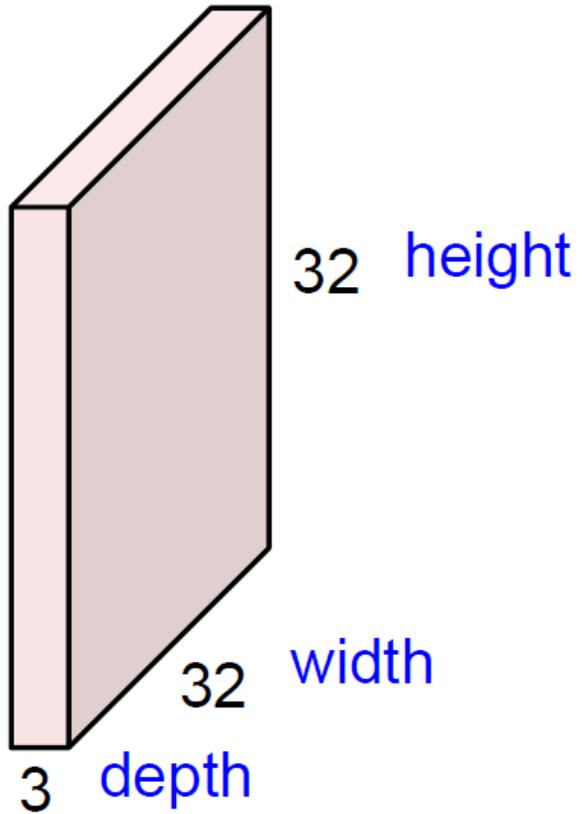


Есть реакция

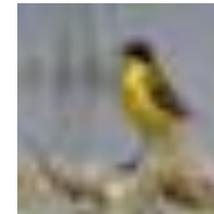
Сверточный слой



Дано изображение 32x32x3 (ширина 32 пикселя, высота 32 пикселя, 3 пикселя RGB)

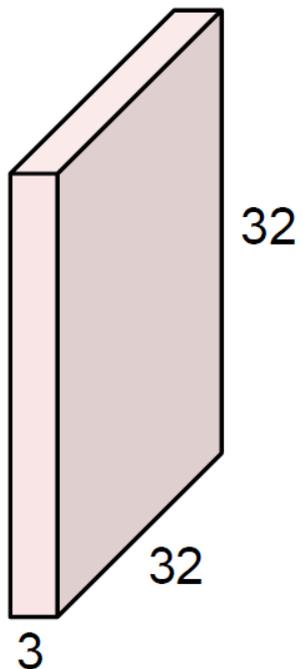


Сверточный слой



Дано изображение 32x32x3 (ширина 32 пикселя, высота 32 пикселя, 3 пикселя RGB)

32x32x3 image



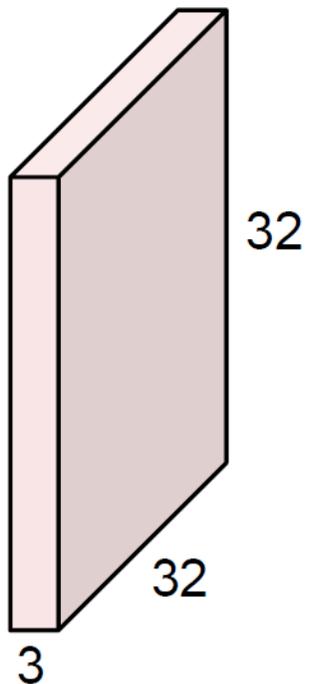
5x5x3 filter



Объединяем фильтр и изображения:
берем область изображения и
находим скалярное произведение
этой области на фильтр

Сверточный слой

32x32x3 image



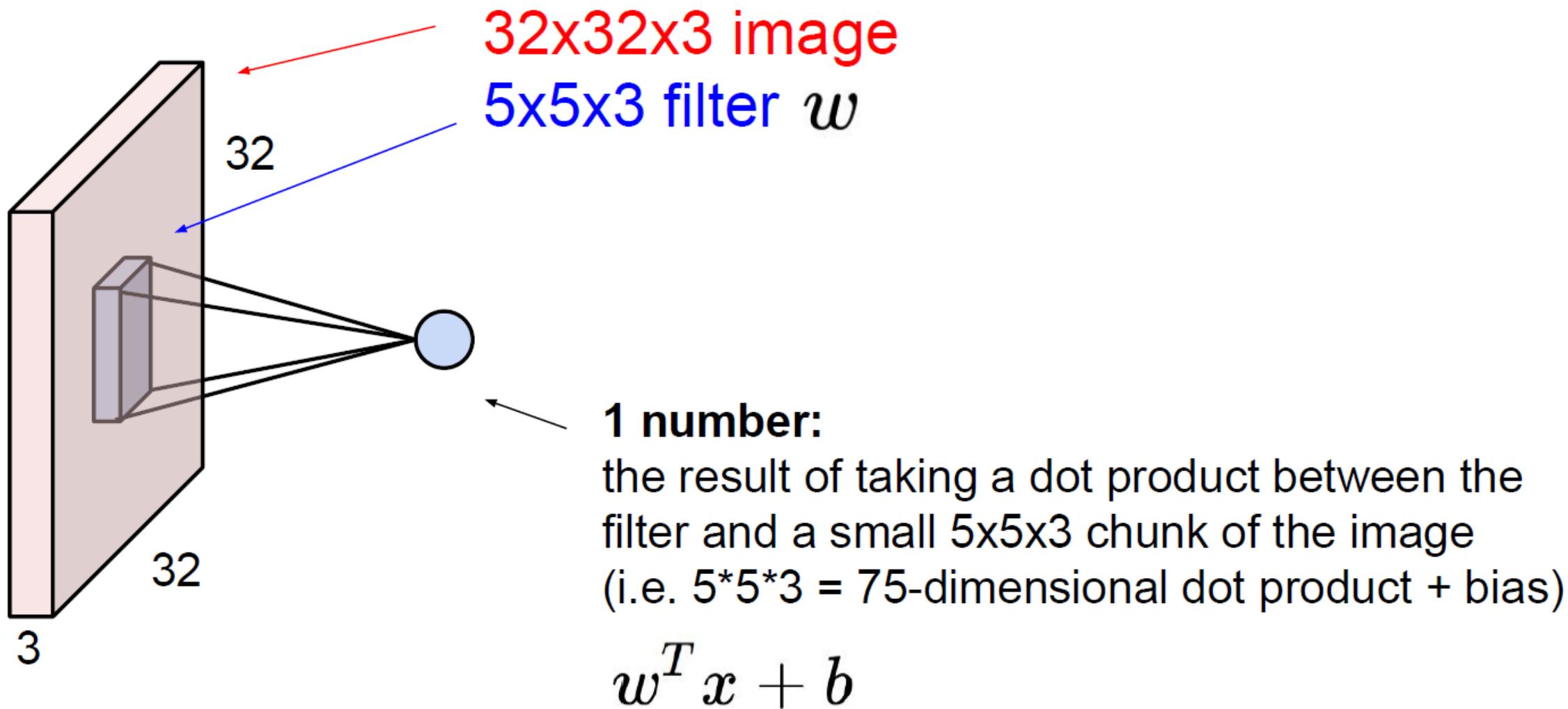
5x5x3 filter



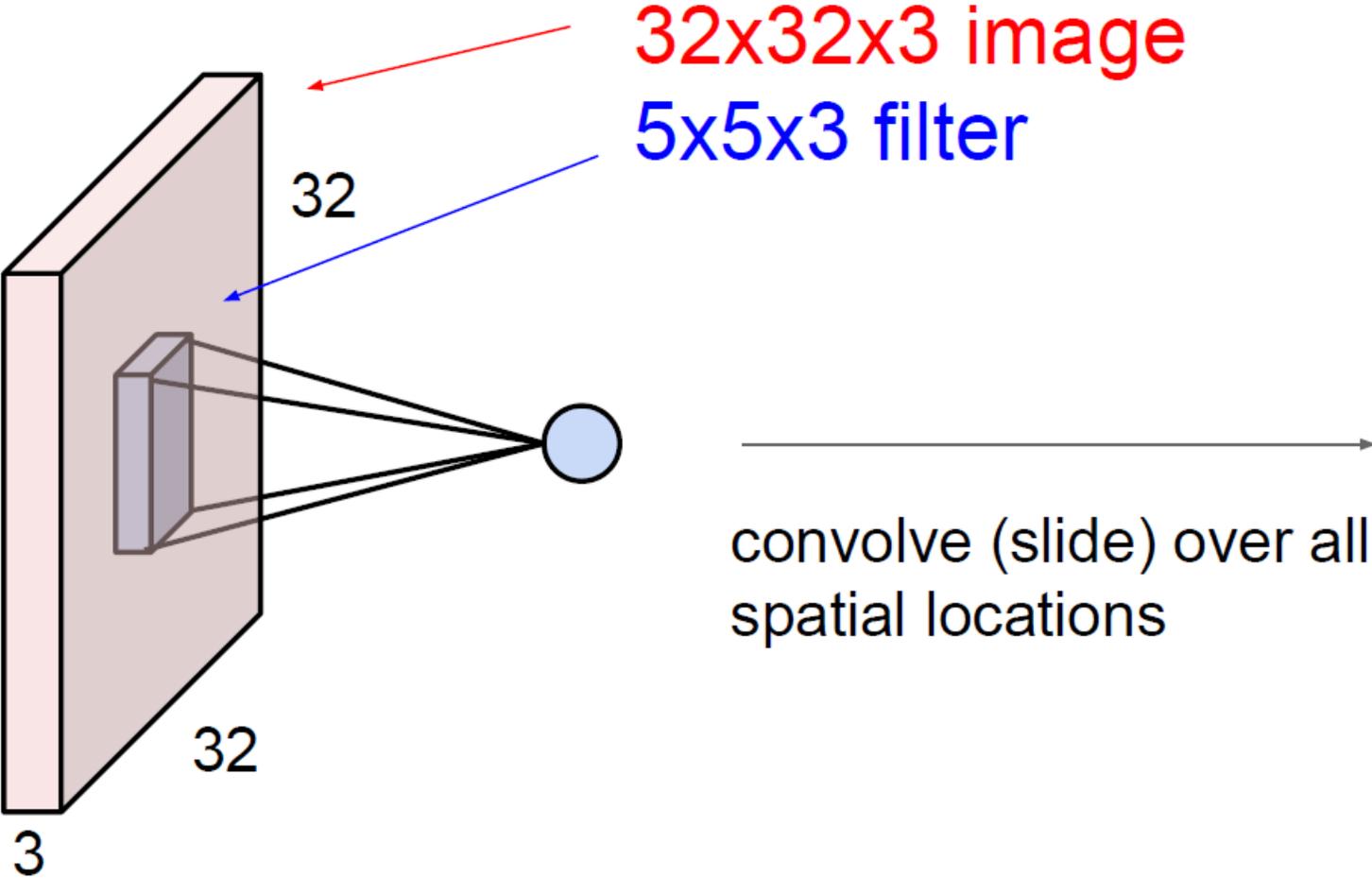
Глубина фильтра всегда совпадает с глубиной входящего изображения

Объединяем фильтр и изображения:
берем область изображения и
находим скалярное произведение
этой области на фильтр

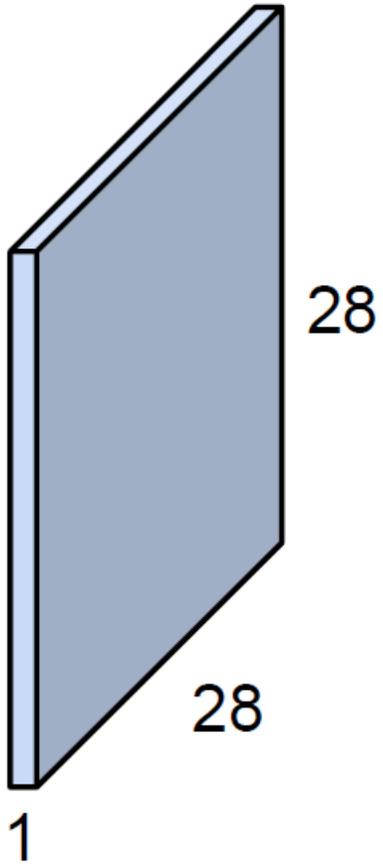
Сверточный слой



Сверточный слой

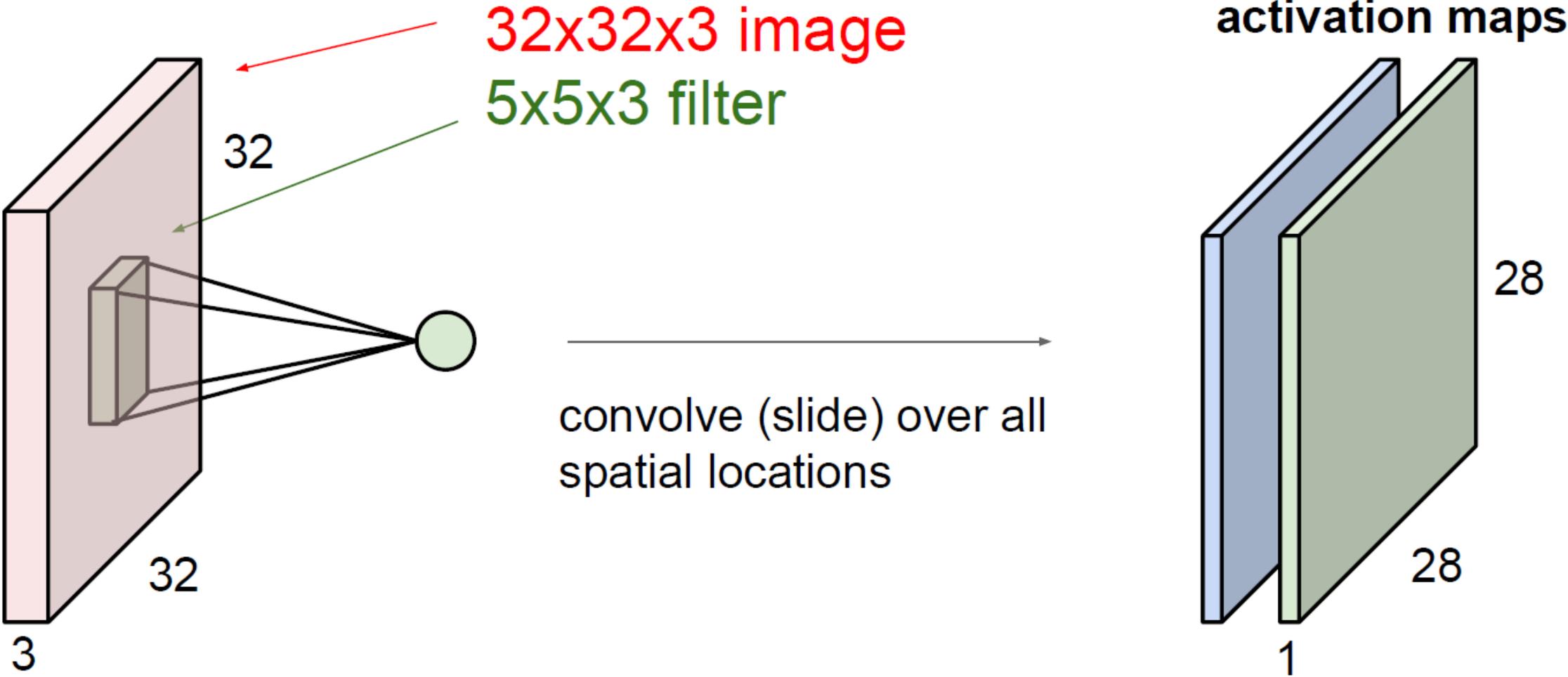


activation map



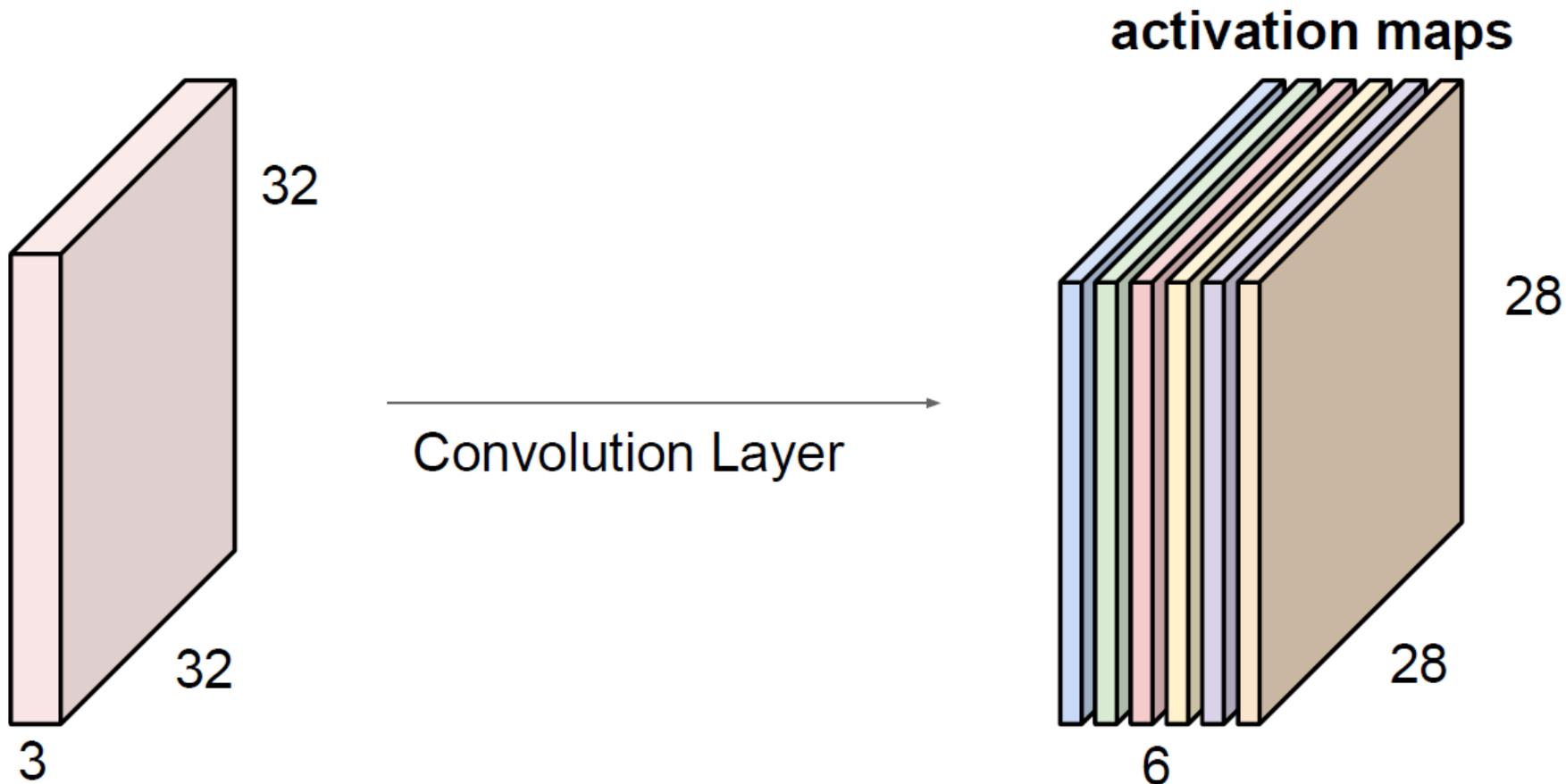
Сверточный слой

Возьмем второй фильтр и обозначим его зеленым



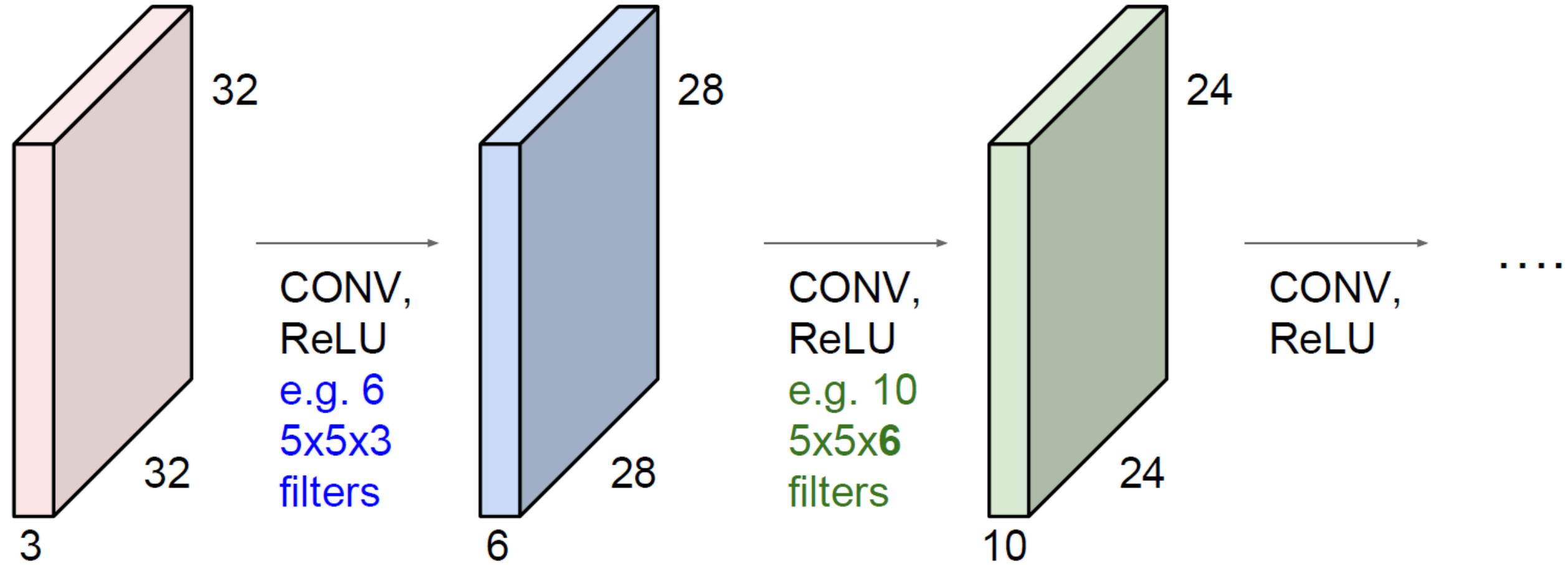
Сверточный слой

Затем возьмем 6 фильтров размера $5 \times 5 \times 3$, и в итоге получим 6 отдельных карт активации



Сложив «стопкой» эти карты, мы получаем новое «изображение» размера $28 \times 28 \times 6$

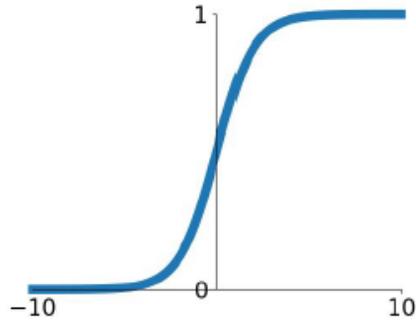
Таким образом, сверточная сеть – это чередование сверточных слоев и функций активации



Функция активации

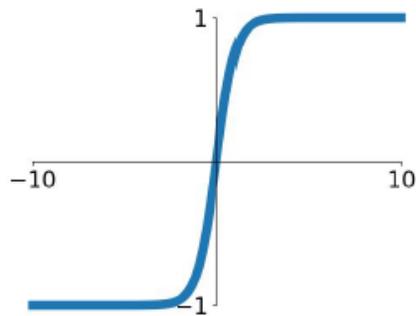
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



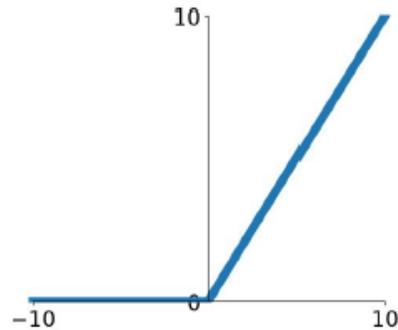
tanh

$$\tanh(x)$$



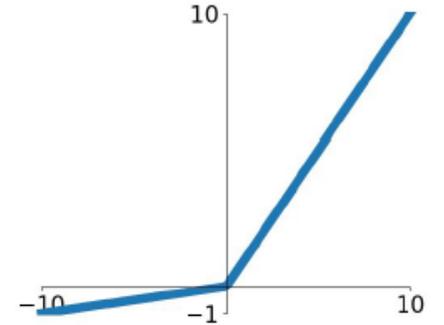
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

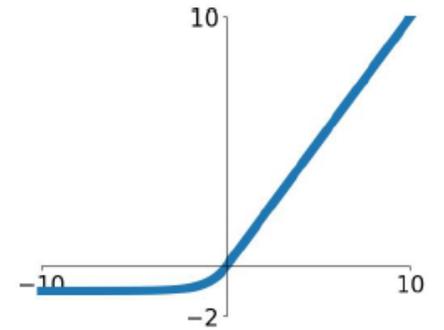


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Preview

[Zeiler and Fergus 2013]

Visualization of VGG-16 by Lane McIntosh. VGG-16 architecture from [Simonyan and Zisserman 2014].

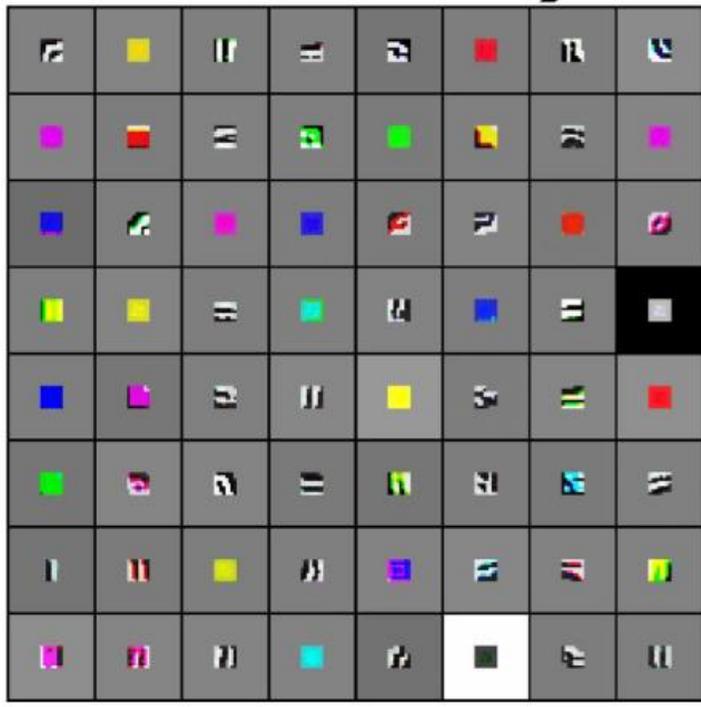


Low-level features

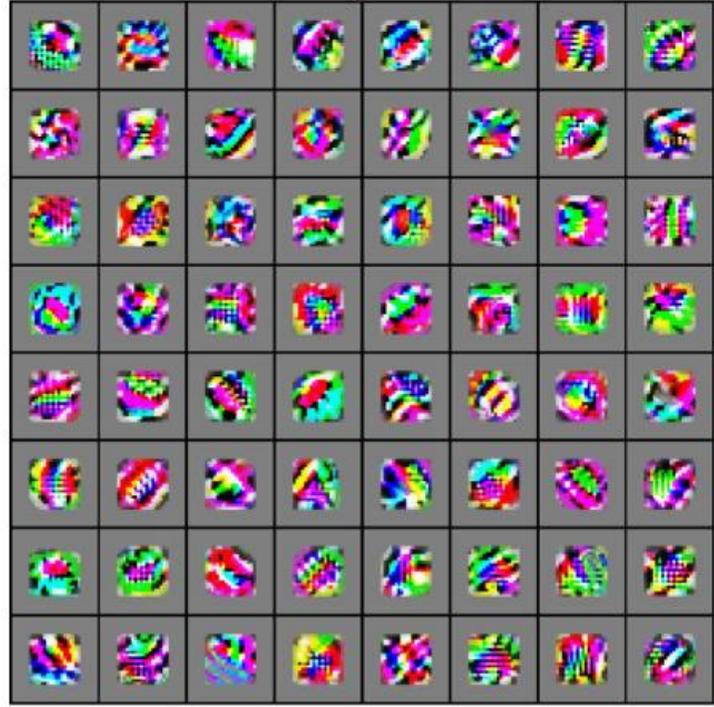
Mid-level features

High-level features

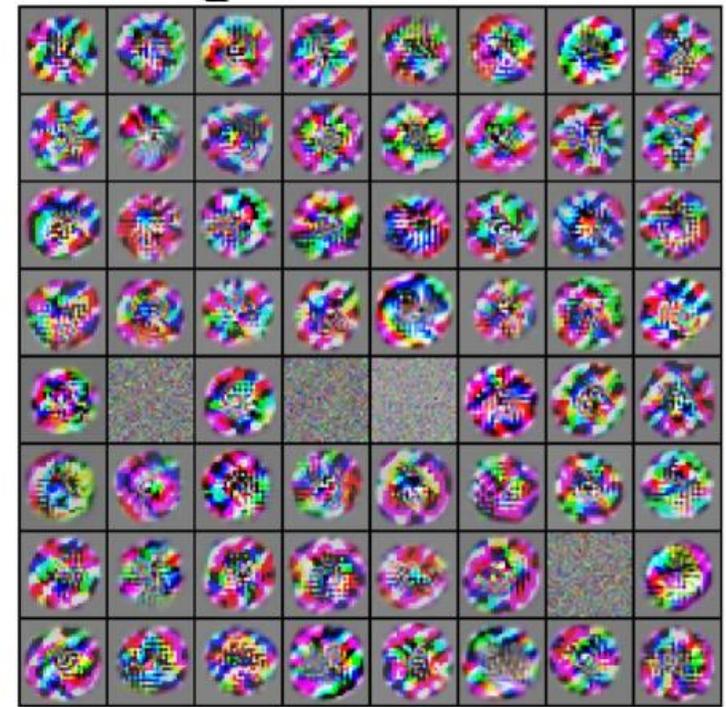
Linearly separable classifier



VGG-16 Conv1_1



VGG-16 Conv3_2



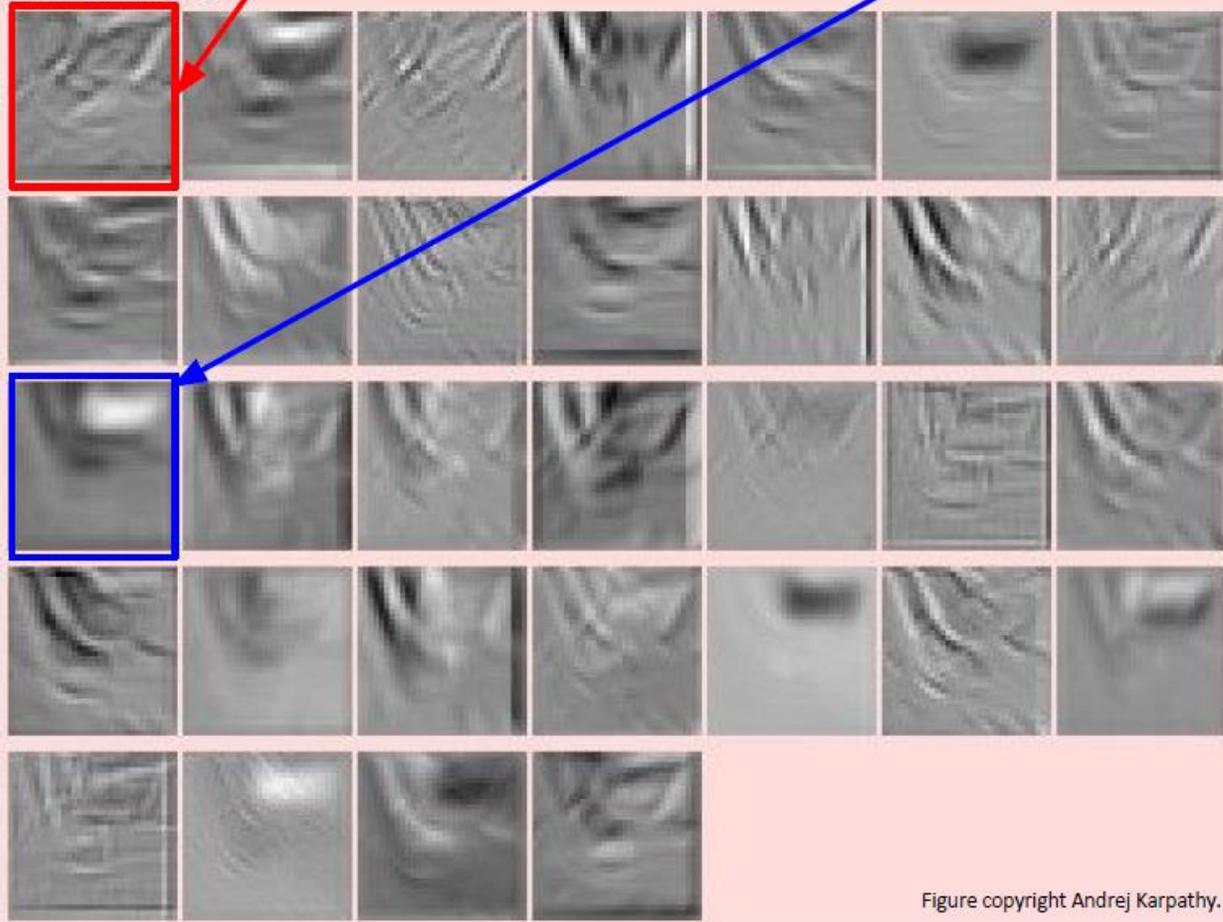
VGG-16 Conv5_3



one filter =>
one activation map

example 5x5 filters
(32 total)

Activations:



We call the layer convolutional because it is related to convolution of two signals:

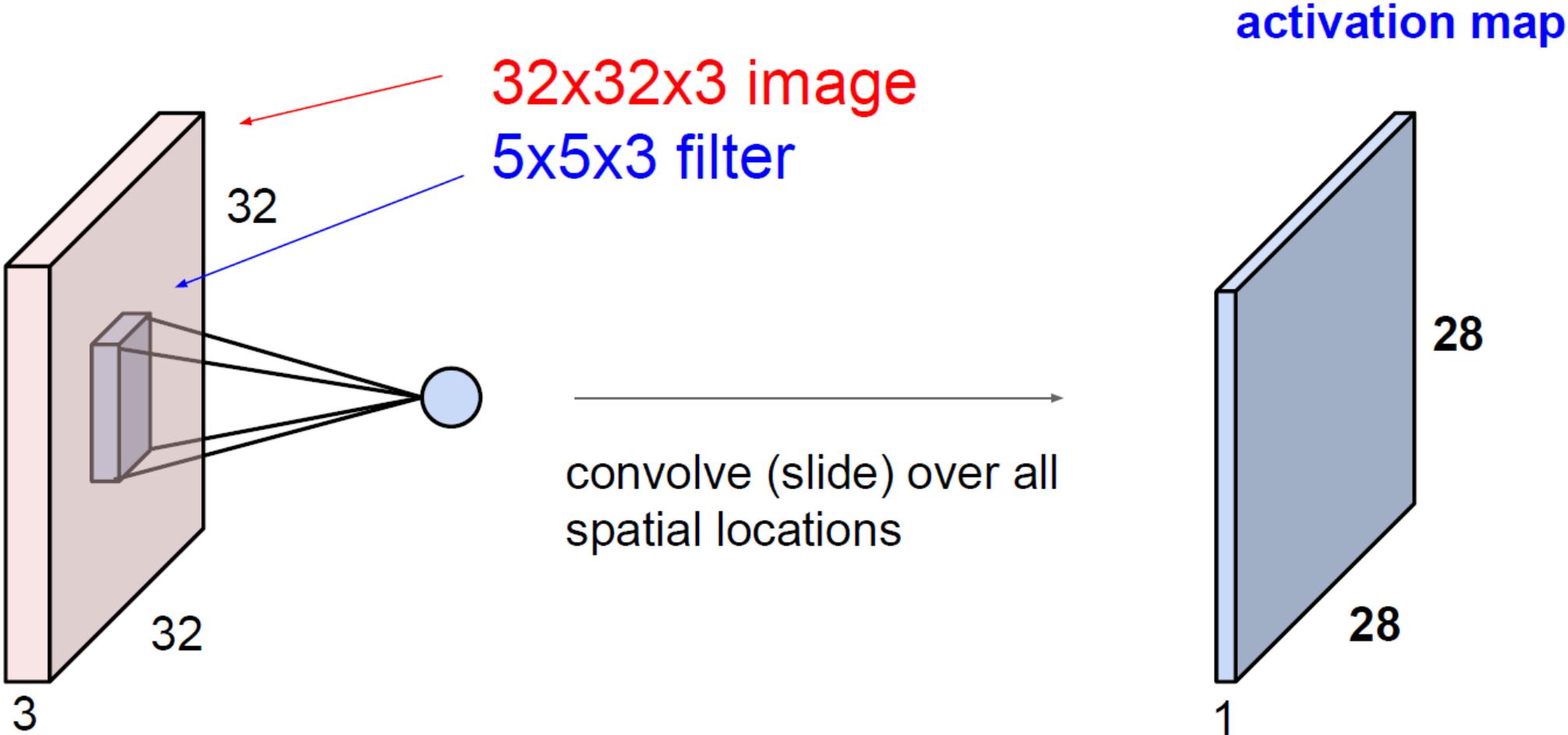
$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1,y-n_2]$$



elementwise multiplication and sum of a filter and the signal (image)

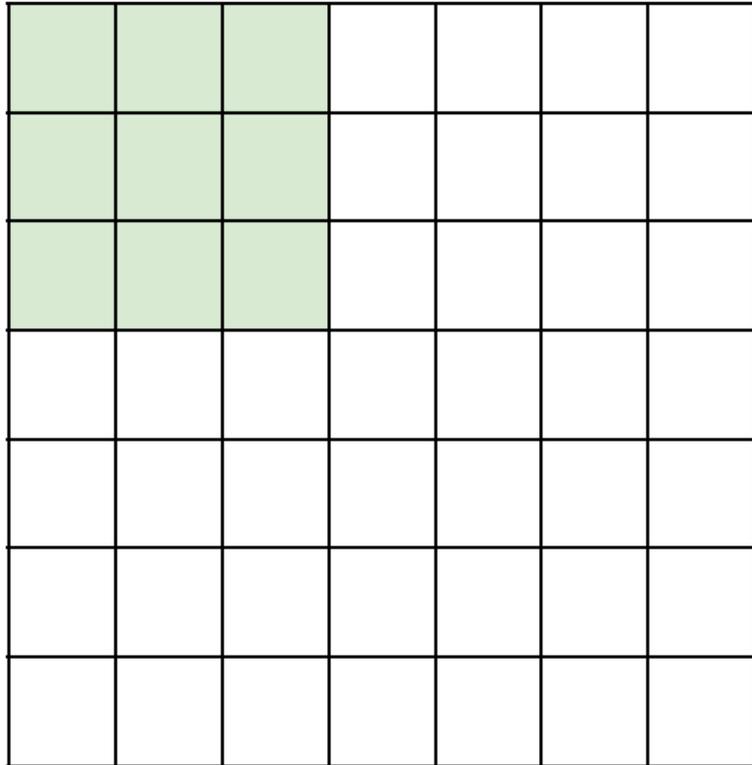
Figure copyright Andrej Karpathy.

Рассмотрим размерность полученных в результате свертки изображений



Рассмотрим размерность полученных в результате свертки изображений

7

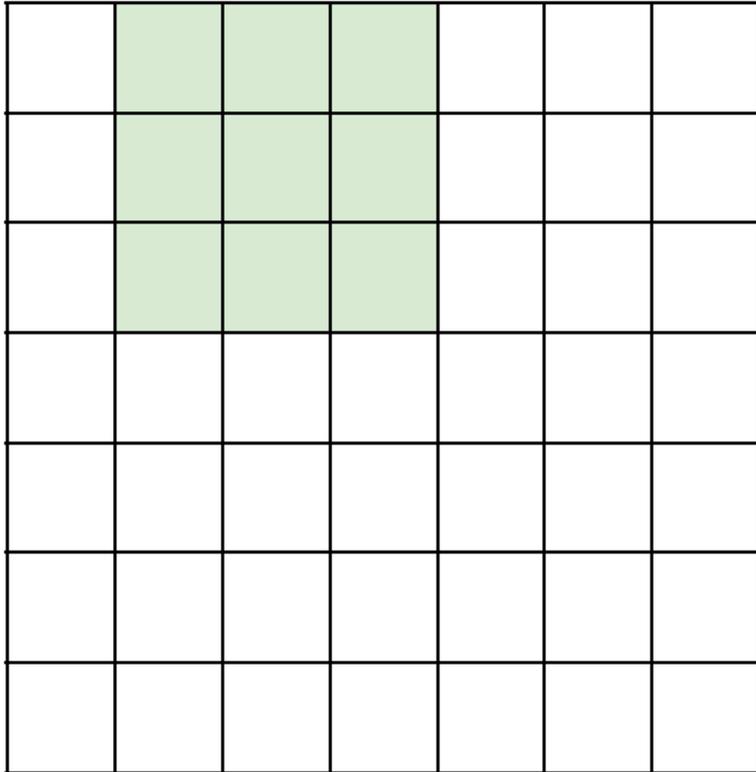


7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 1

Рассмотрим размерность полученных в результате свертки изображений

7

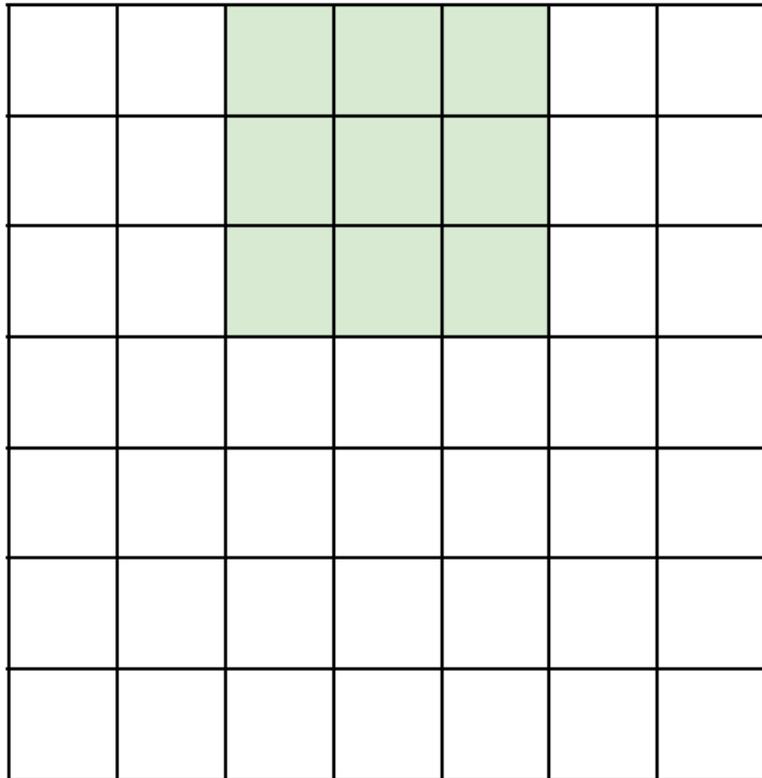


7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 1

Рассмотрим размерность полученных в результате свертки изображений

7

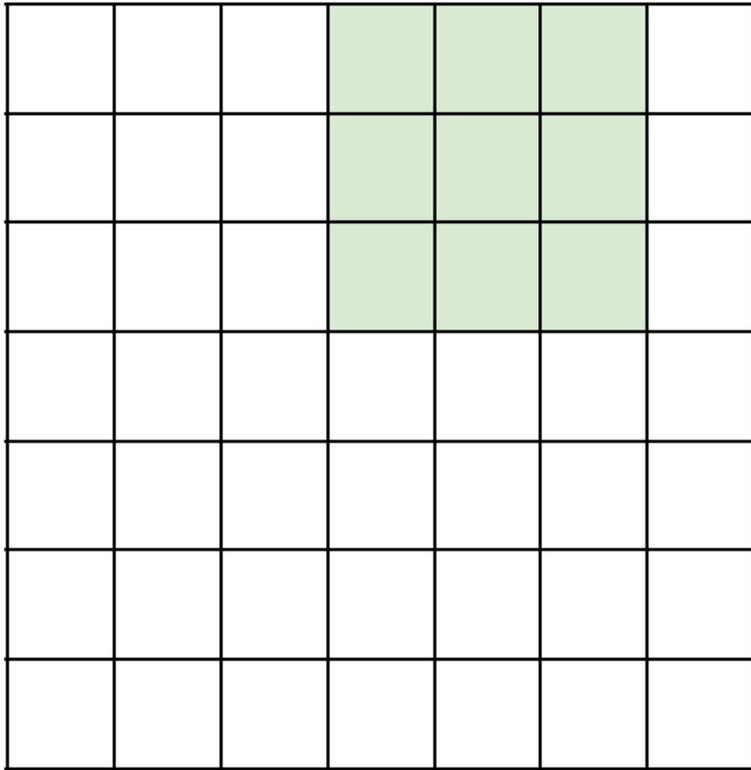


7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 1

Рассмотрим размерность полученных в результате свертки изображений

7

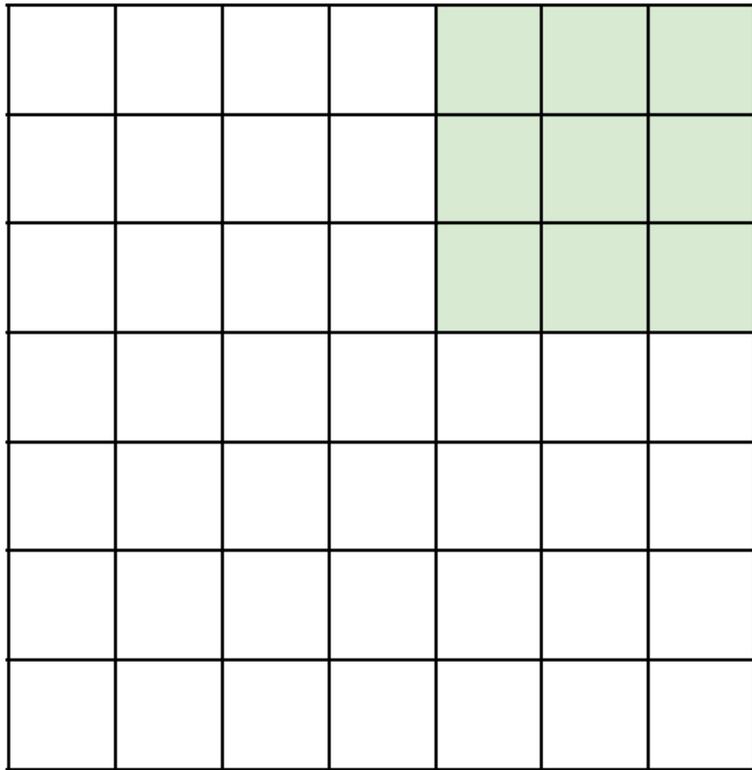


7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 1

Рассмотрим размерность полученных в результате свертки изображений

7



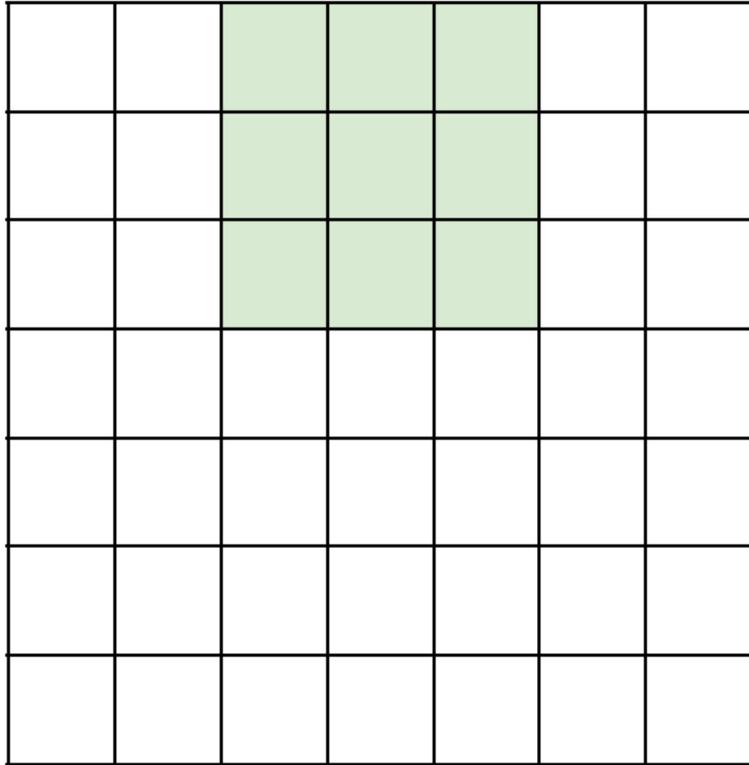
7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 1

Карта активации будет 5x5 пикселей

Рассмотрим размерность полученных в результате свертки изображений

7

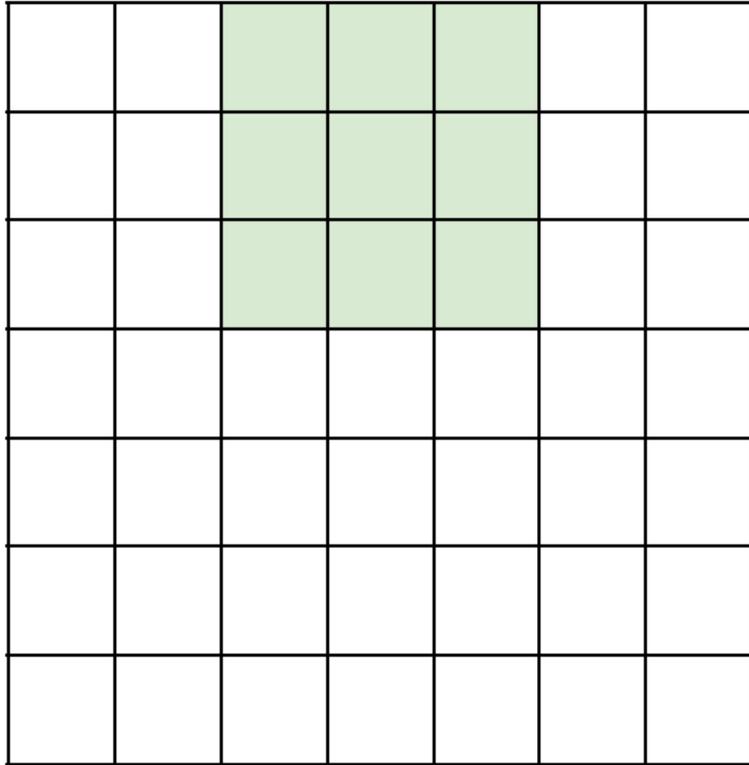


7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 2

Рассмотрим размерность полученных в результате свертки изображений

7



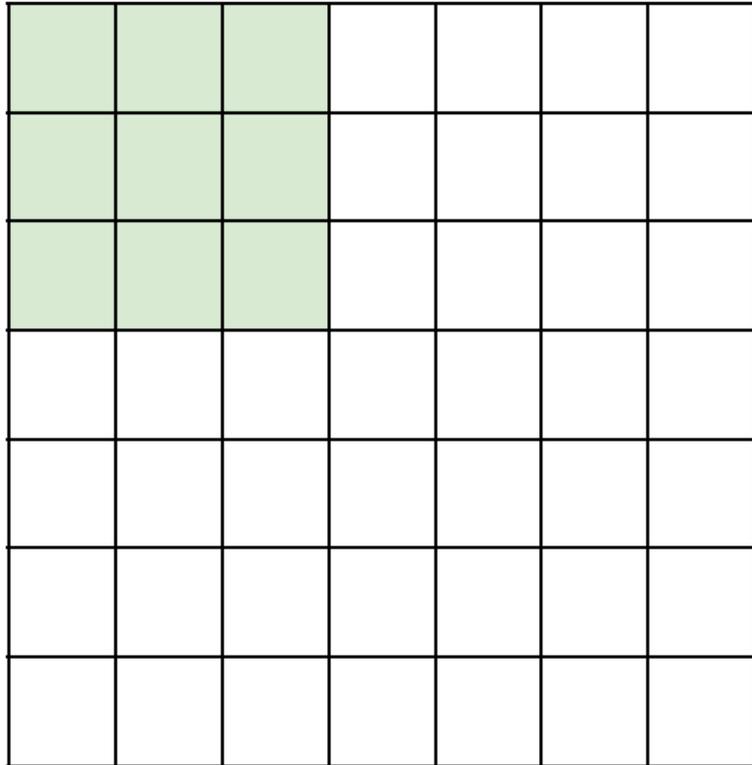
7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 2

Карта активации будет 3x3 пикселей

Рассмотрим размерность полученных в результате свертки изображений

7

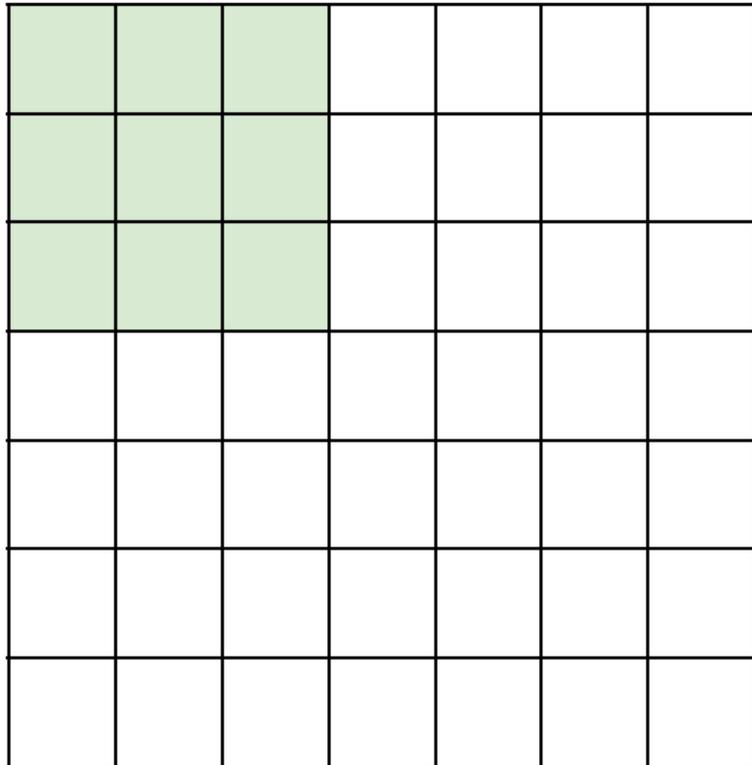


7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 3

Рассмотрим размерность полученных в результате свертки изображений

7



7

Изображение 7x7
пикселей, фильтр 3x3
с шагом 3

Фильтр применить нельзя

N

			F			
	F					

N

Размер карты активации:
 $(N - F) / \text{шаг} + 1$

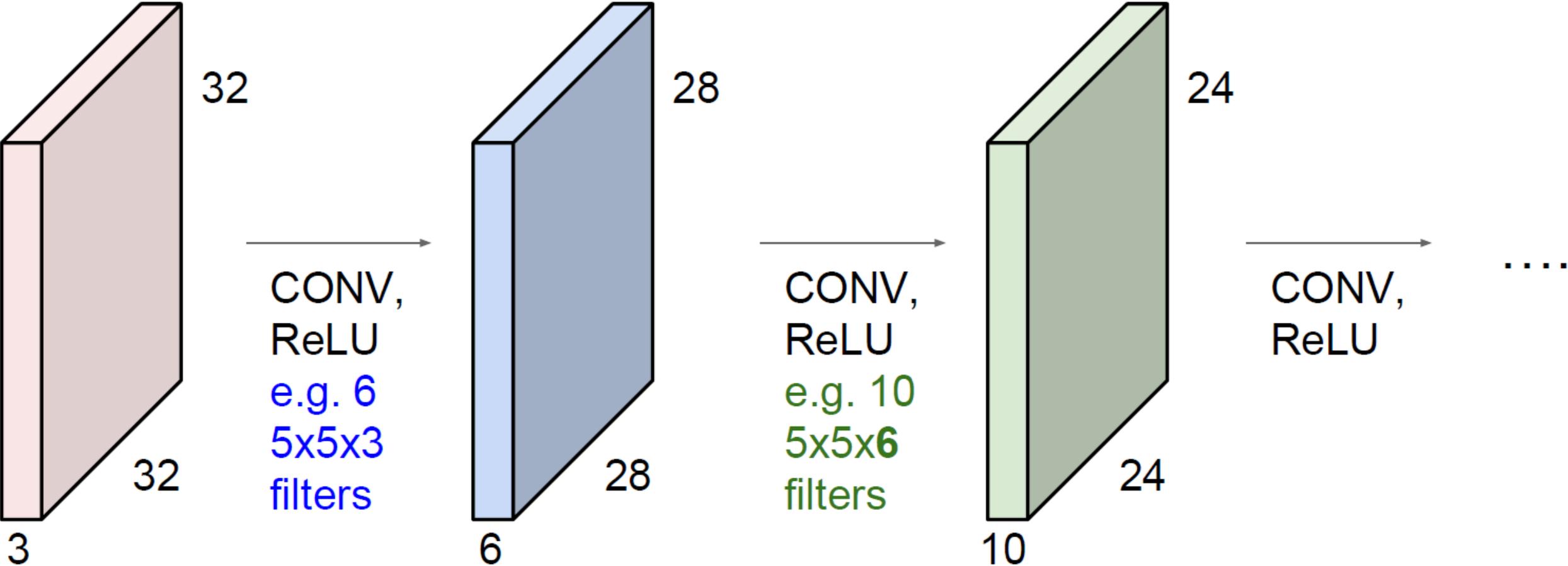
Например, при $N=7$, $F=3$:

Шаг 1: $(7 - 3) / 1 + 1 = 5$

Шаг 2: $(7 - 3) / 2 + 1 = 5$

Шаг 3: $(7 - 3) / 3 + 1 = 2,33$

Еще раз взглянем на размерность карт активации. Например, если к изображению 32x32 пикселя последовательно применять сверточные фильтры размера 5x5, то карты активации будут последовательно сжиматься, и это вызывает проблемы, так как теряется информация



На практике перед применением сверток к изображениям добавляют поля, заполненные нулями

0	0	0	0	0	0			
0								
0								
0								
0								

Например, чтобы сохранить размерность при использовании фильтра 3x3 пикселя, нужно добавить поле шириной в 1 пиксель

При использовании фильтра с шагом 1 размера FxF, размер поля рассчитывается по формуле: $(F - 1) / 2$

Например:

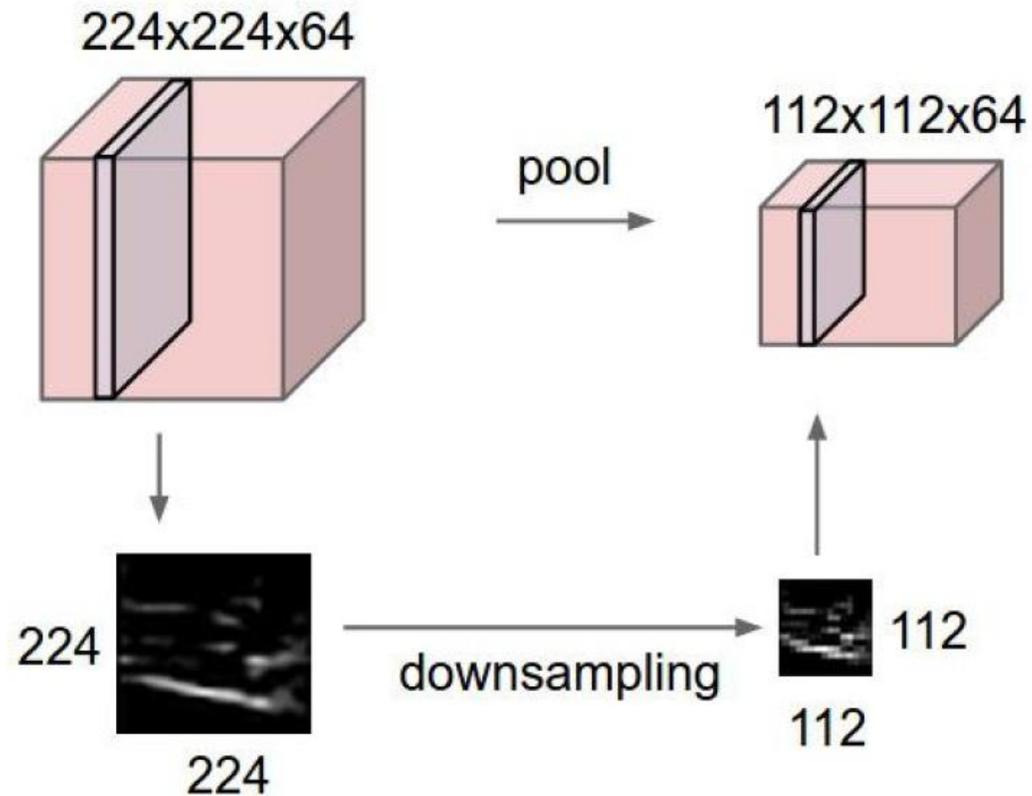
F = 3: поле 1 пиксель

F = 5: поле 2 пикселя

F = 7: поле 3 пикселя

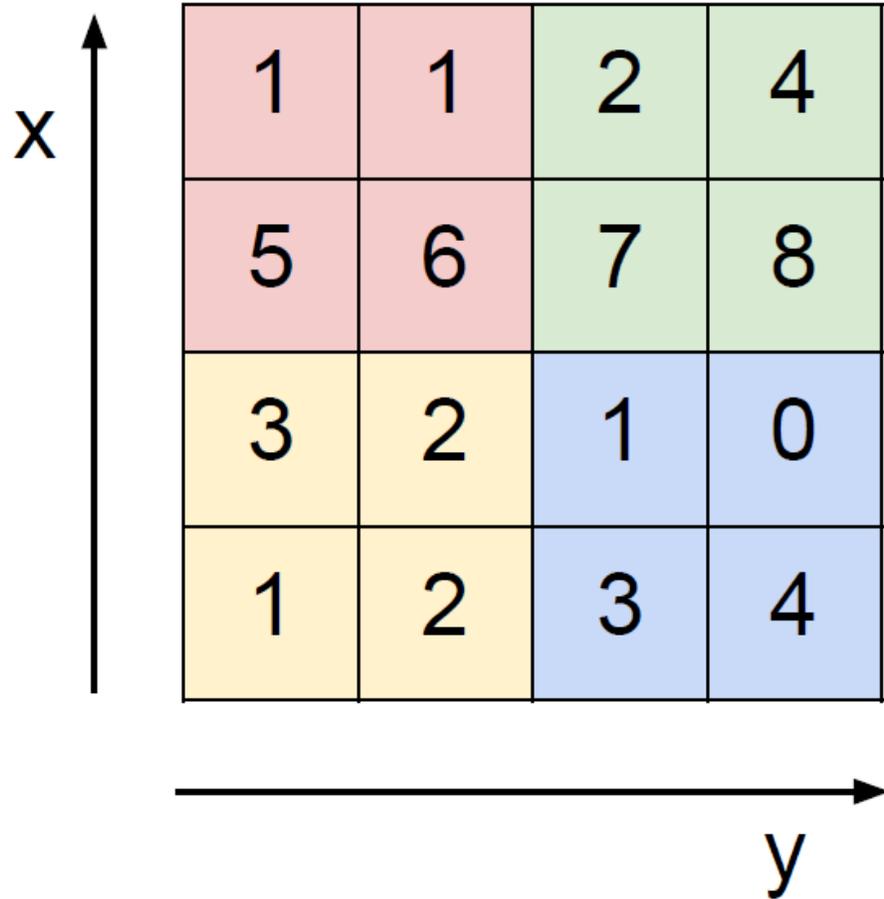
Слой пулинга

- Сжимают карты активации, передавая следующему слою наиболее значимые значения, тем самым делают представления изображений компактнее
- Применяются отдельно к каждой карте активации

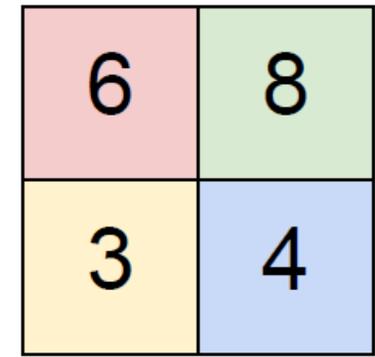


MAX POOLING

Single depth slice

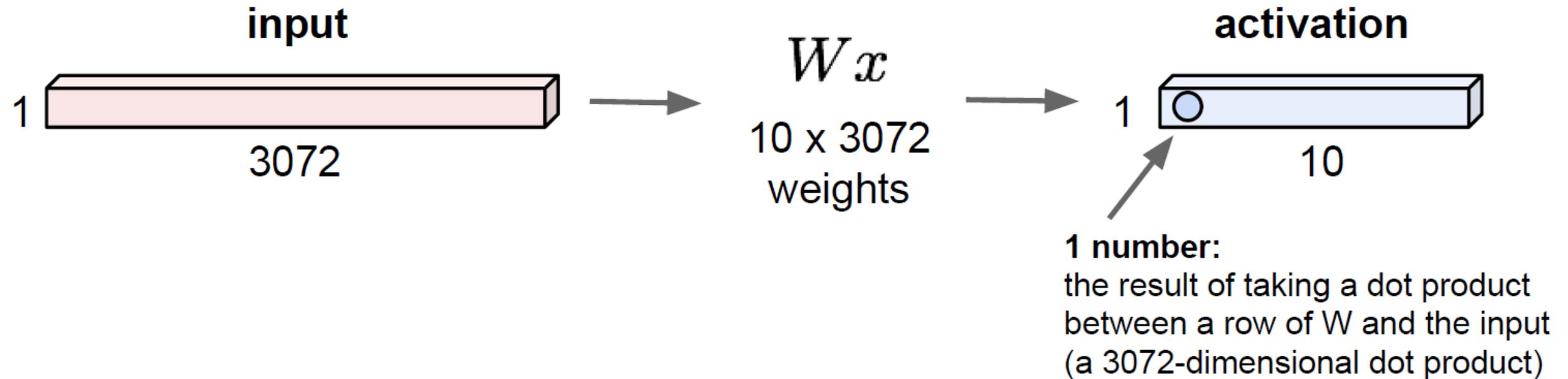


max pool with 2x2 filters
and stride 2



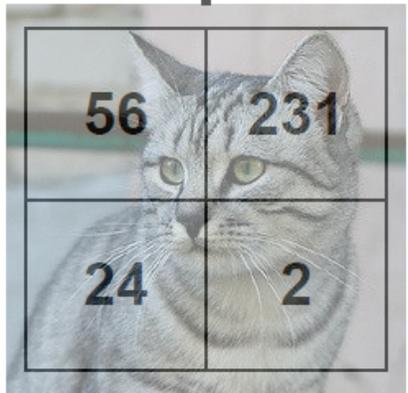
Полносвязный слой

32x32x3 image -> stretch to 3072 x 1



Полносвязный слой

Stretch pixels into column



Input image

0.2	-0.5	0.1	2.0
1.5	1.3	2.1	0.0
0	0.25	0.2	-0.3

W

56
231
24
2

+

1.1
3.2
-1.2

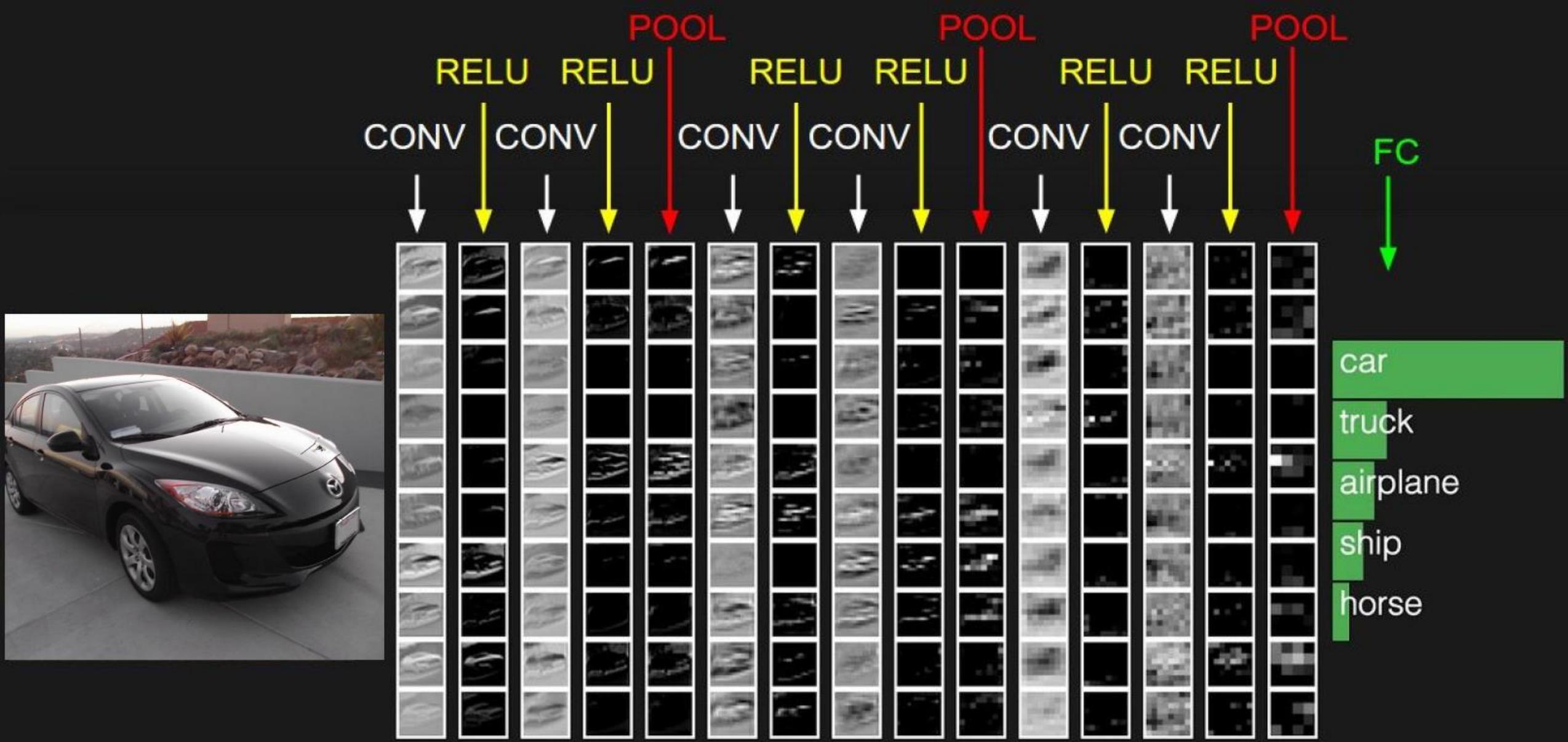
=

-96.8
437.9
61.95

Cat score

Dog score

Ship score



Пример взят с <http://cs231n.stanford.edu/>

Поиск решения

На что нужно обратить внимание при поиске модели

- решаемая задача: модель должна решать поставленную нами задачу (классификации, классификации и локализации)
- требования к вычислительным мощностям
- точность модели (по результатам экспериментальных данных)
- при выборе модели нужно искать компромисс и выбирать те модели, которые дают наибольшую точность при наименьших системных требованиях

Готовые и обученные модели



<https://github.com/CMU-Perceptual-Computing-Lab/openpose>

Готовые модели, требующие обучения или переобучения



Cat



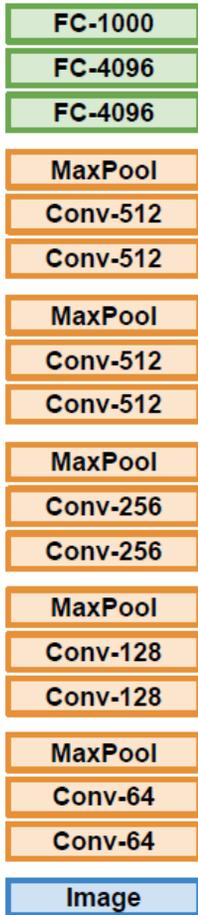
Dog



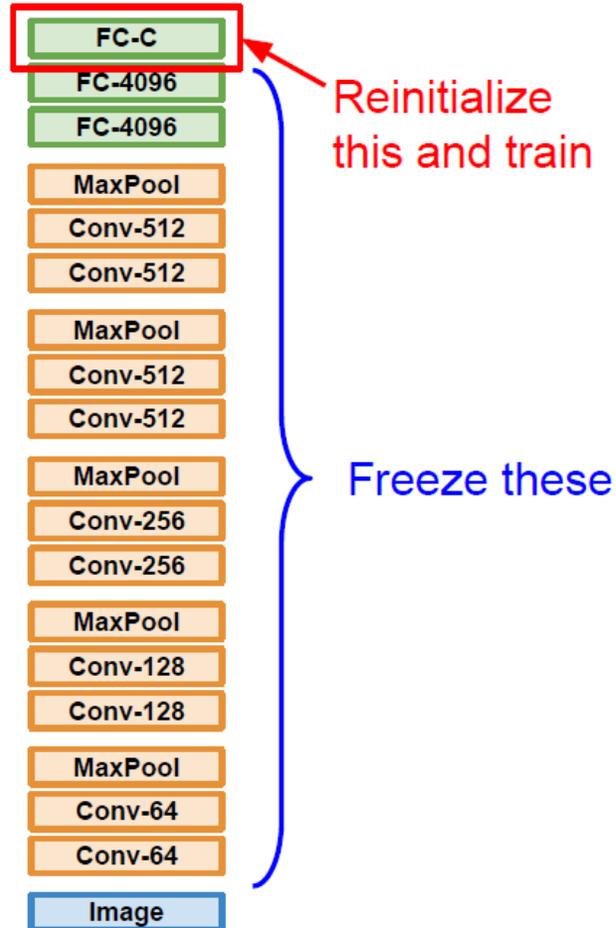
???

Transfer learning

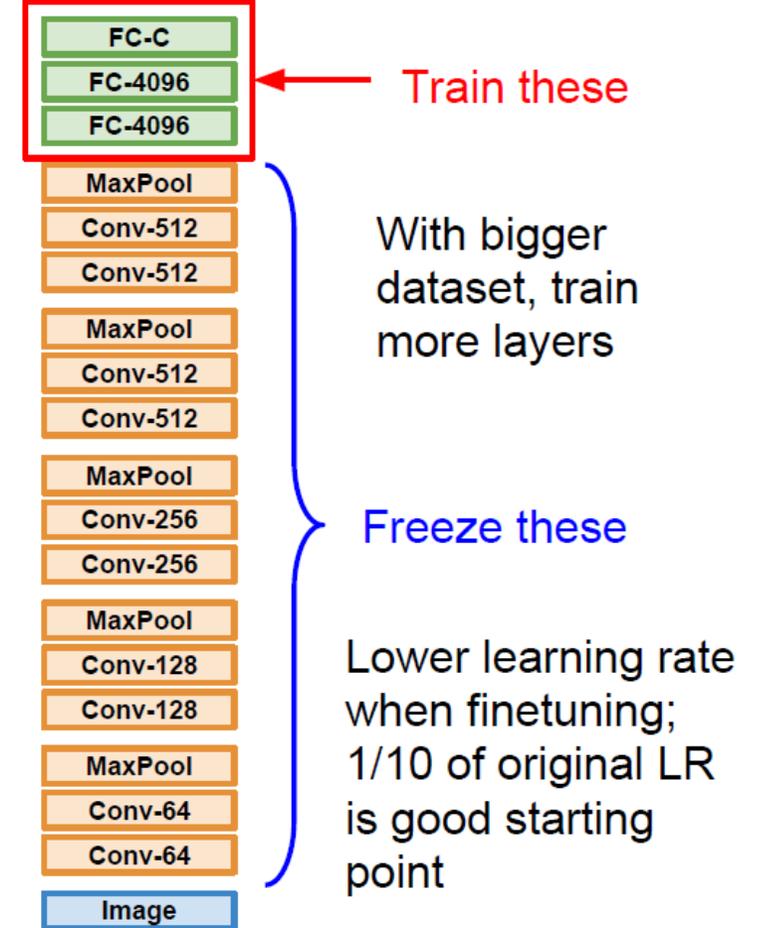
1. Train on Imagenet



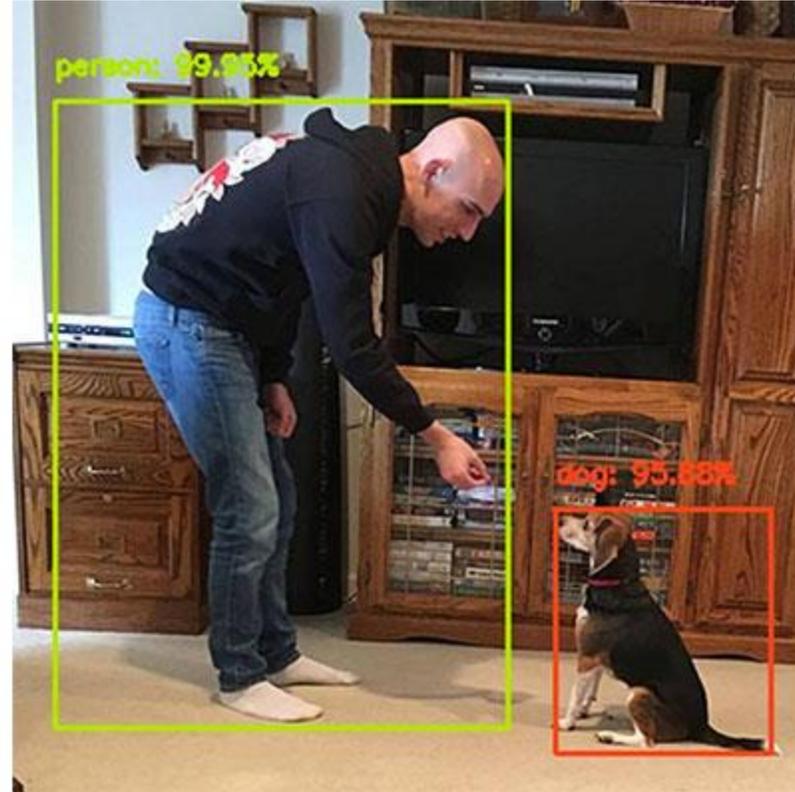
2. Small Dataset (C classes)



3. Bigger dataset



Transfer learning



<https://www.pyimagesearch.com/2020/06/22/turning-any-cnn-image-classifier-into-an-object-detector-with-keras-tensorflow-and-opencv/>

