

# Машинное обучение

## Часть II - Convolution Neural Network

*Власов Кирилл Вячеславович*



2020

# Histogram of oriented gradients

Input image



Histogram of Oriented Gradients

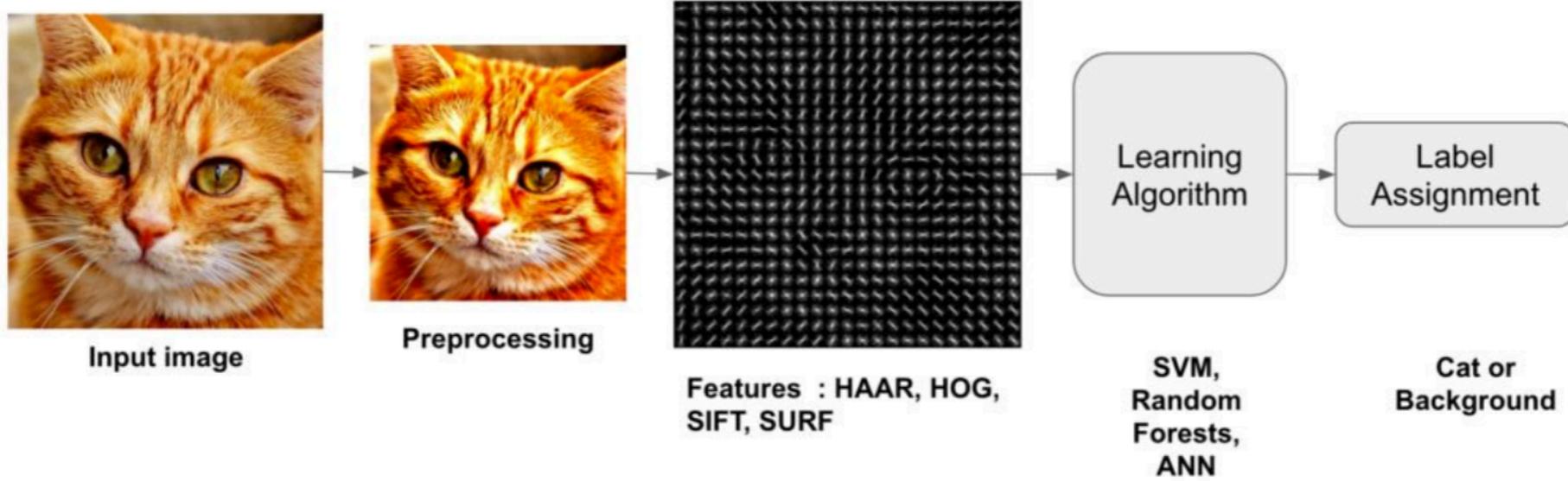


# Histogram of oriented gradients

Input image



Histogram of Oriented Gradients



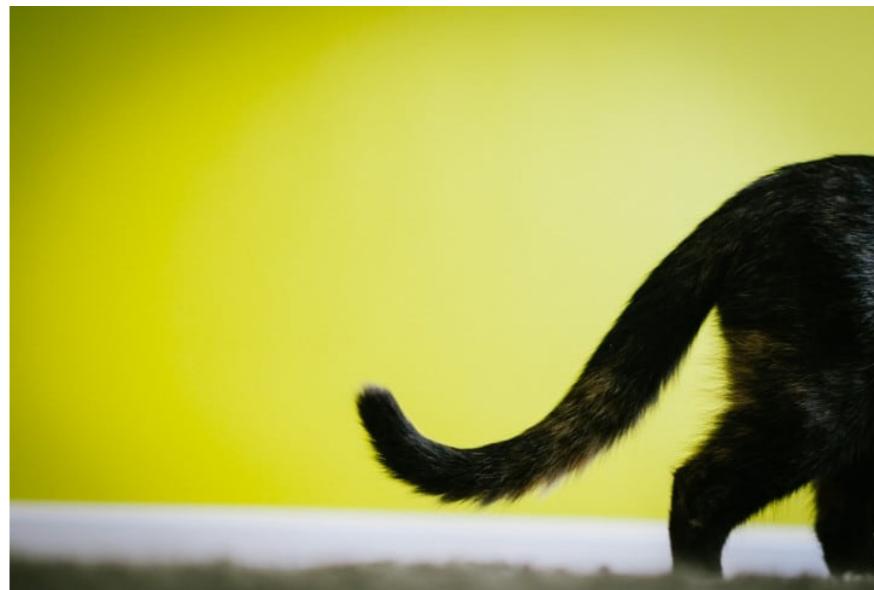
# Распознавание образов



# Распознавание образов



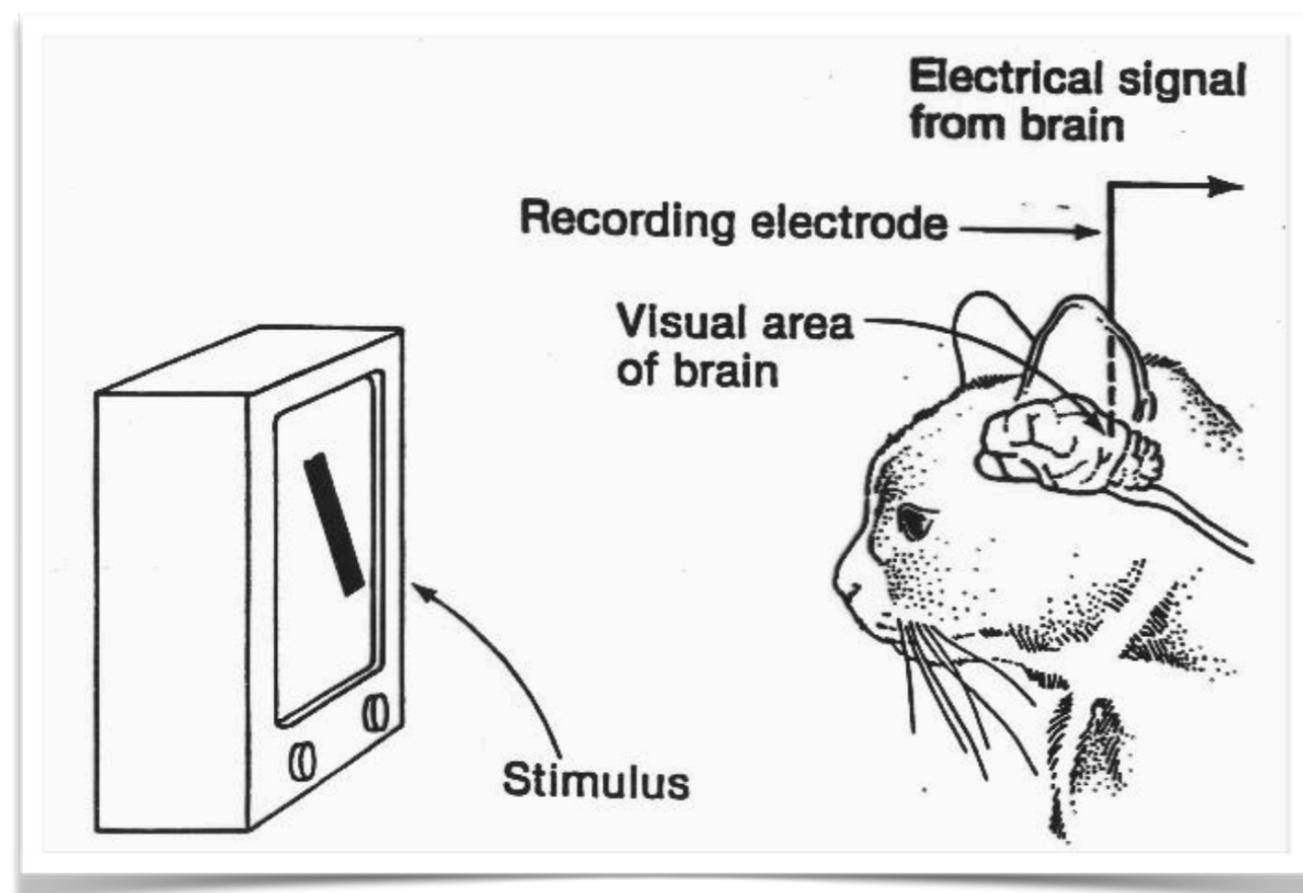
# Распознавание образов



# Биологическая предпосылка

The Nobel Prize in Physiology or Medicine 1981

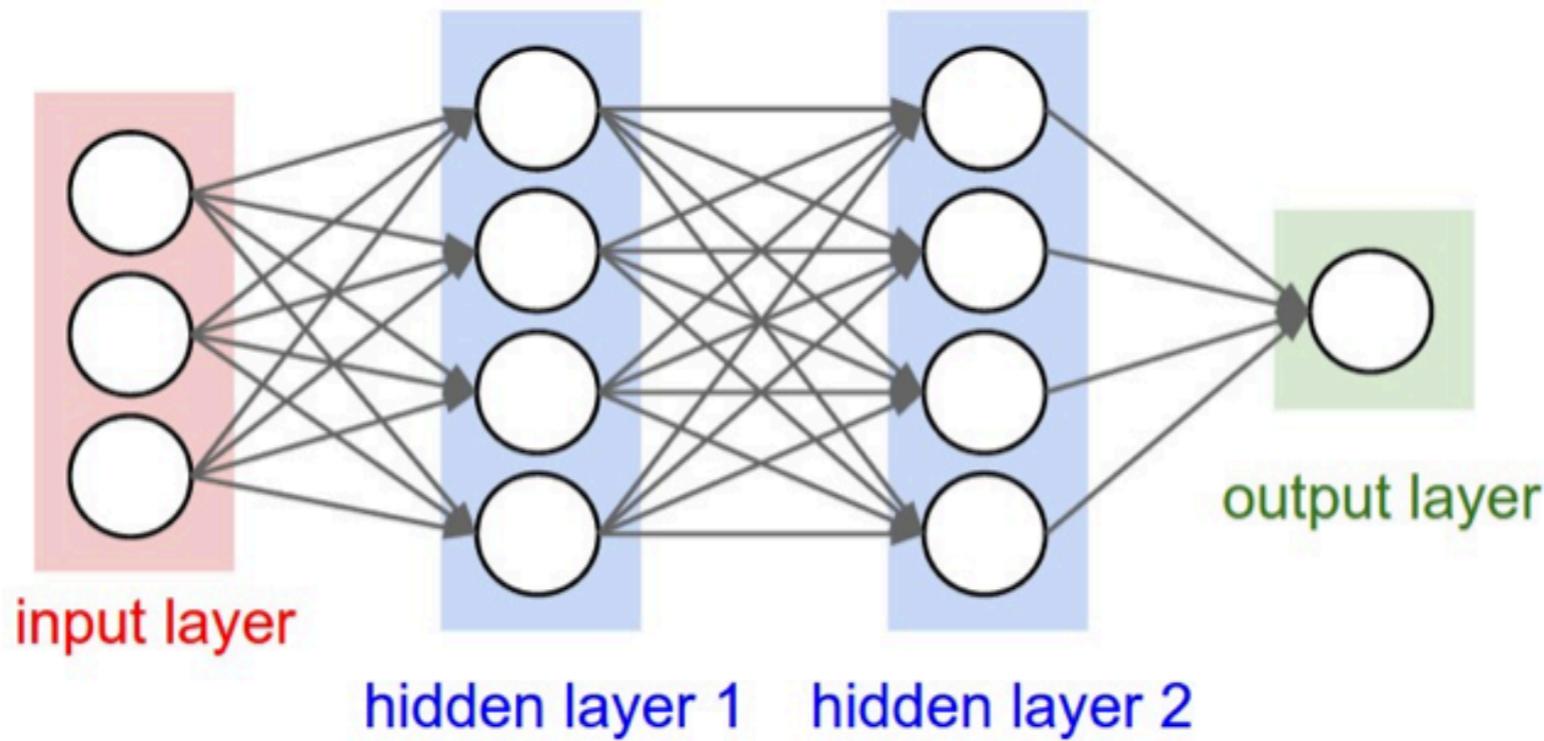
специализированные нейроны, которые реагируют только на определенную сенсорную информацию.



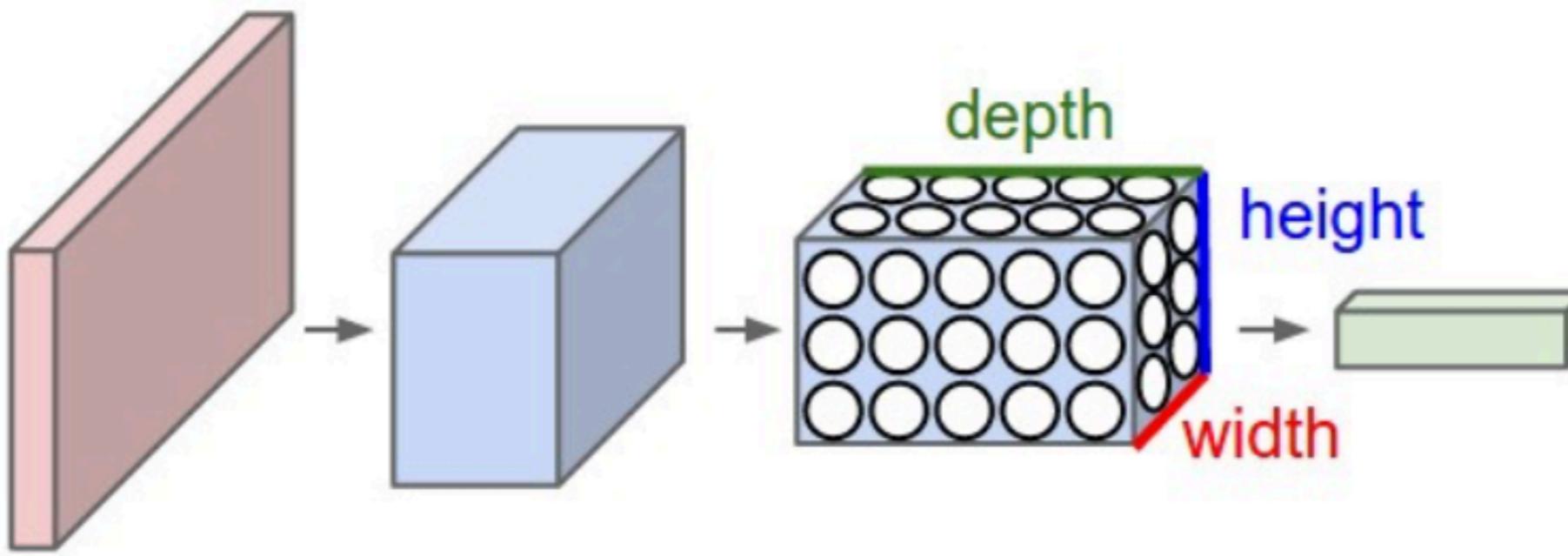
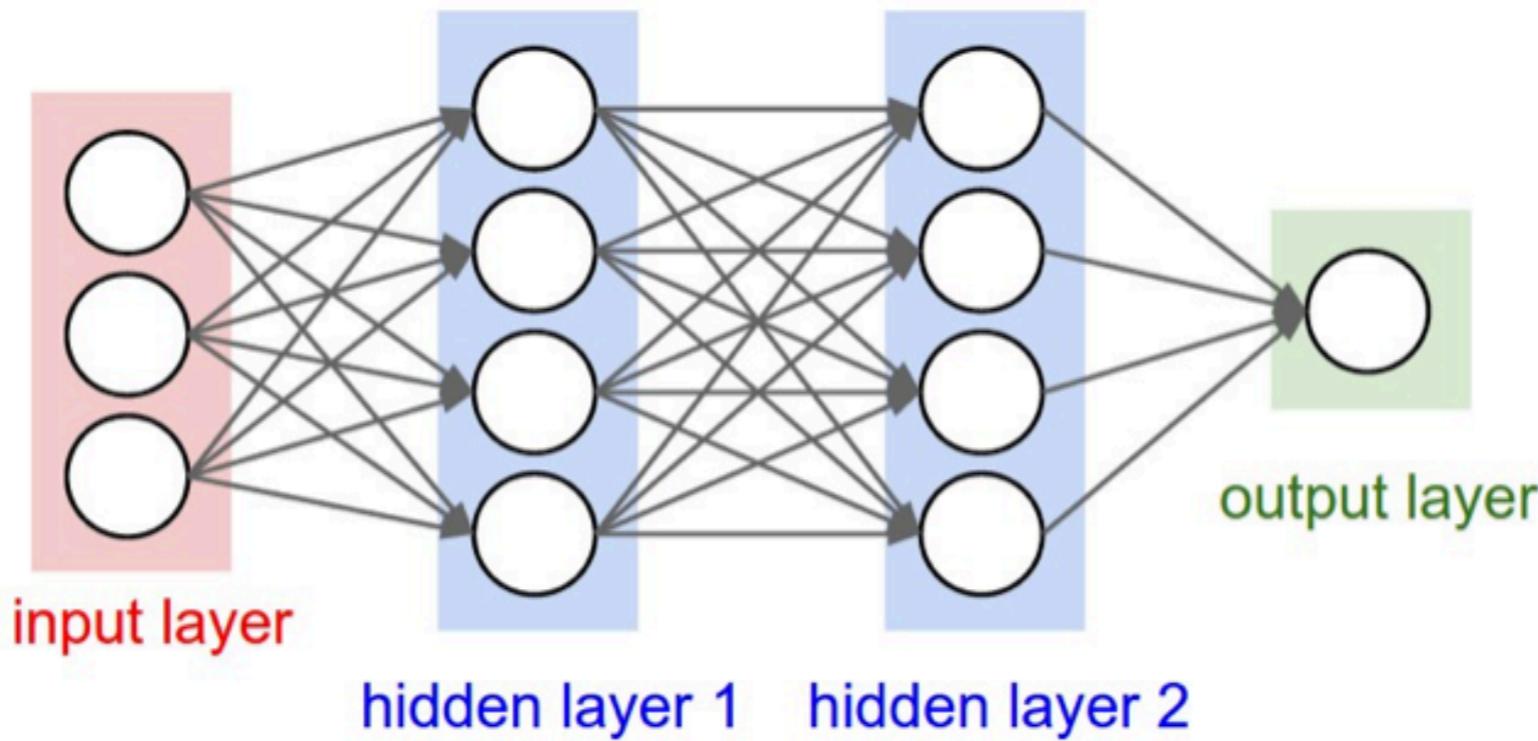
- соседние нейроны обрабатывают сигналы с соседних областей сетчатки;
- нейроны образуют иерархическую структуру (изображение ниже), где каждый следующий уровень выделяет все более и более высокоуровневые признаки;
- нейроны организованы в так называемые колонки – вычислительные блоки, которые трансформируют и передают информацию от уровня к уровню.

<https://www.youtube.com/watch?v=JOHayh06LJ4>

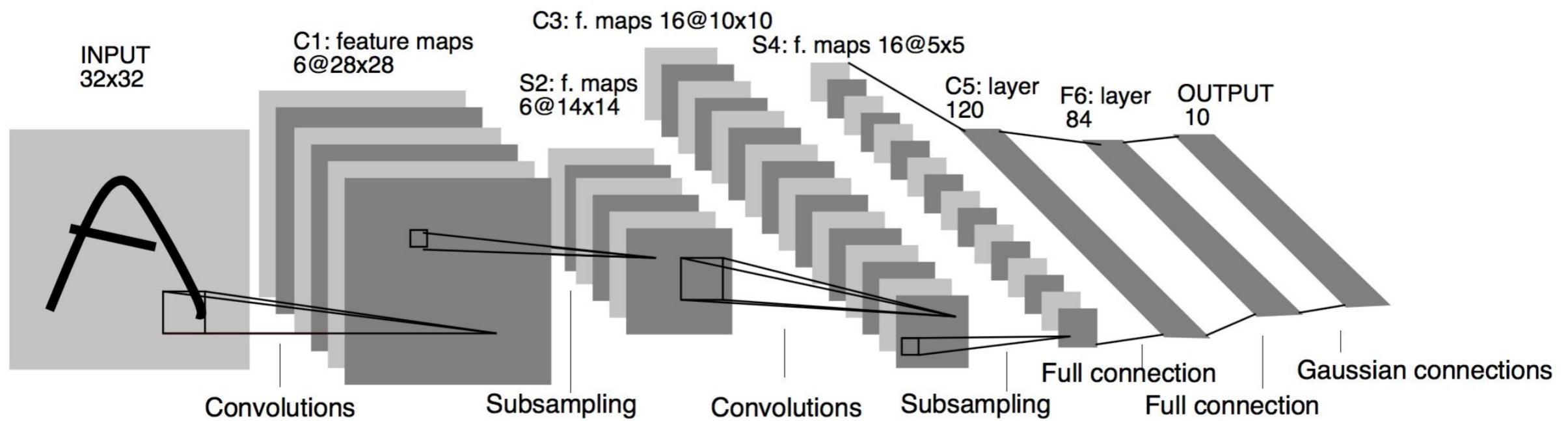
# Сверточные нейронные сети



# Сверточные нейронные сети



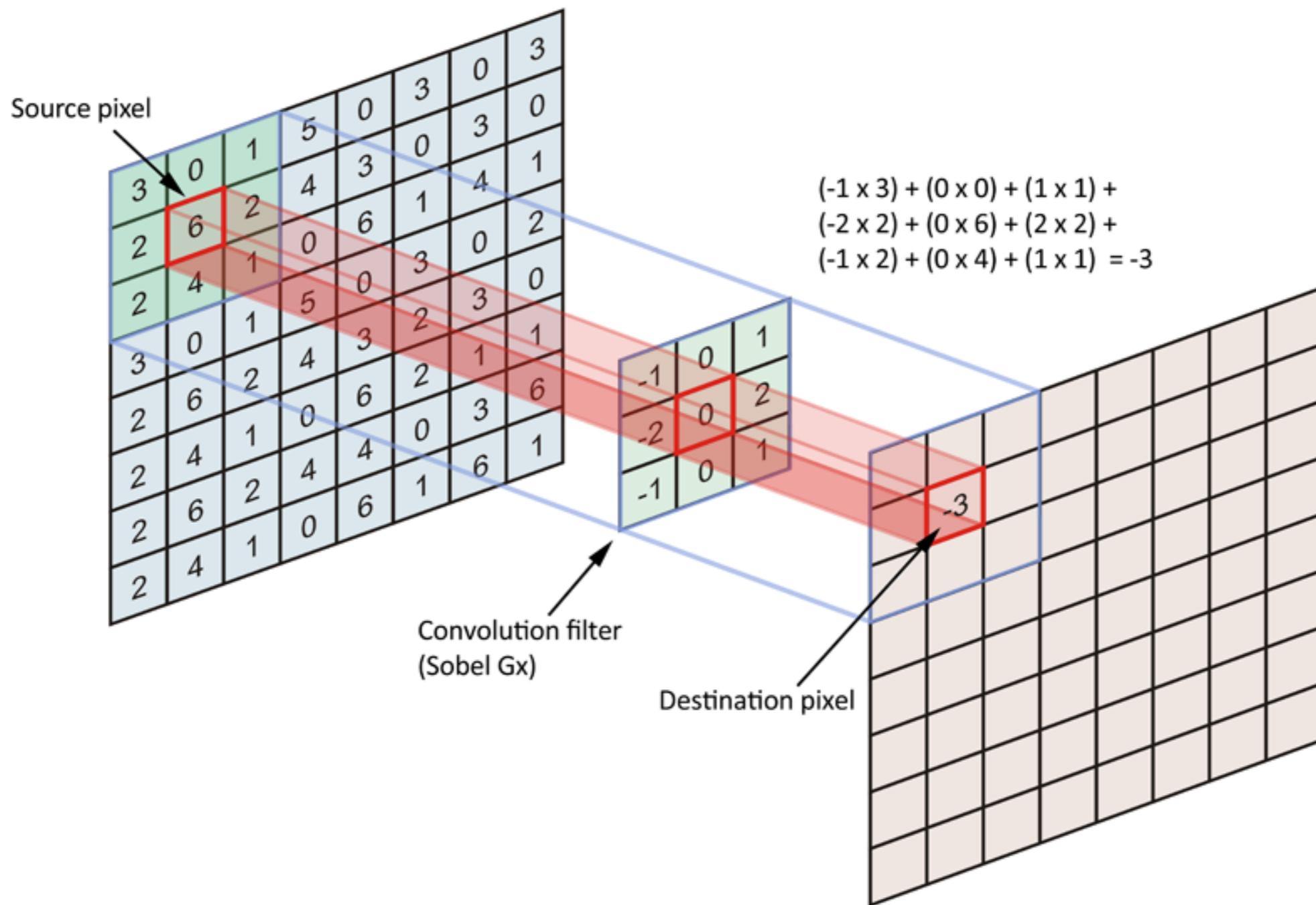
# Первая сверточная нейронная сеть



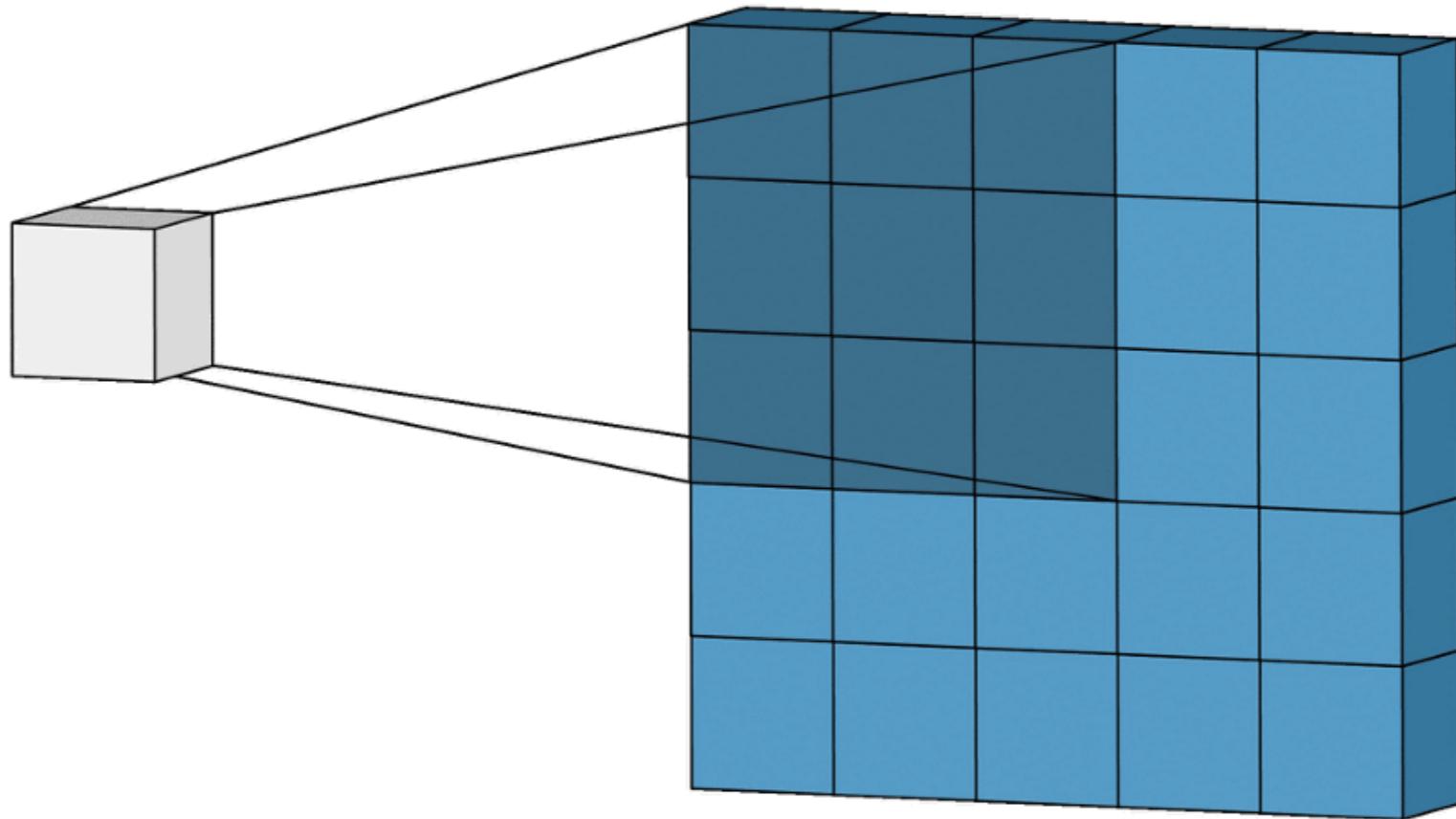
- Свертки  $5 \times 5$  со сдвигом 1 и пулинг  $2 \times 2$  со сдвигом 2
- 60 тысяч параметров
- Практическое применение: с помощью этой сети на почте США распознавали рукописные индексы

Оригинальная статья

# Операция Свертки



# Операция Свертки



<https://proglib.io/wp-content/uploads/2018/06/1.gif>

# Операция Свертки

## Lines detection

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal lines

-1	2	-1
-1	2	-1
-1	2	-1

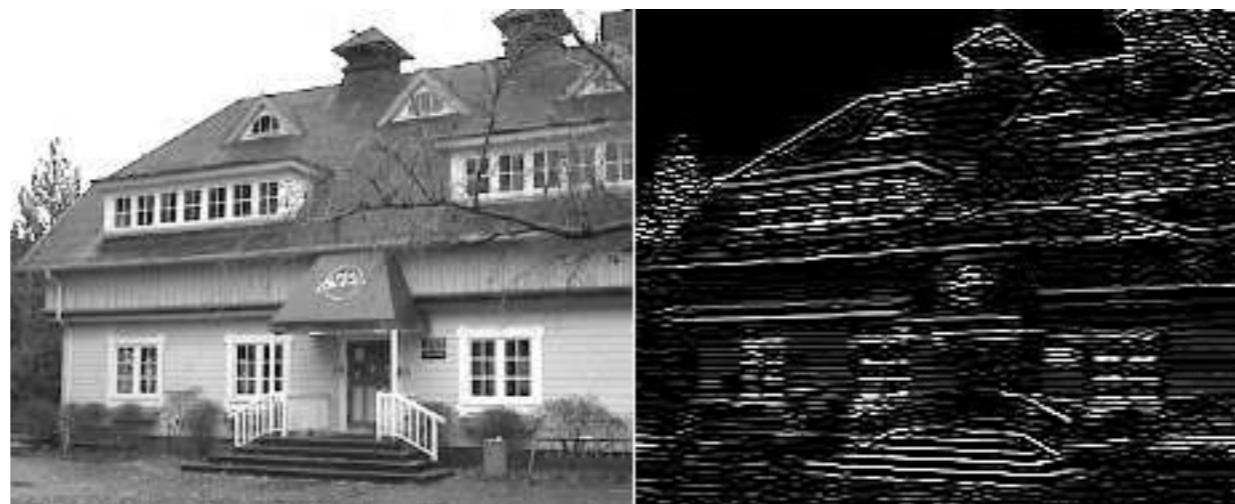
Vertical lines

-1	-1	2
-1	2	-1
2	-1	-1

45 degree lines

2	-1	-1
-1	2	-1
-1	-1	2

135 degree lines



## Edge detection

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



## Blur

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



<http://aishack.in/tutorials/image-convolution-examples/>

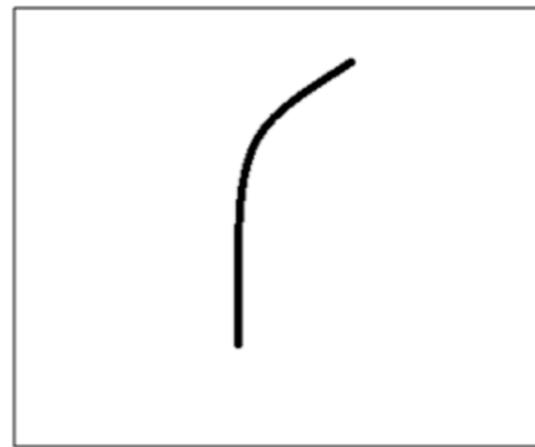
<https://habr.com/post/142818/>

# Операция Свертки

## Фильтры

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



Visualization of a curve detector filter

# Операция Свертки



Original image



Visualization of the filter on the image



Visualization of the receptive field

0	0	0	0	0	0	30	
0	0	0	0	50	50	50	
0	0	0	20	50	0	0	
0	0	0	50	50	0	0	
0	0	0	50	50	0	0	
0	0	0	50	50	0	0	
0	0	0	50	50	0	0	

Pixel representation of the receptive field

\*

0	0	0	0	0	0	30	0
0	0	0	0	0	30	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0

Pixel representation of filter

# Операция Свертки



Original image



Visualization of the filter on the image



Visualization of the receptive field

0	0	0	0	0	0	30	
0	0	0	0	50	50	50	
0	0	0	20	50	0	0	
0	0	0	50	50	0	0	
0	0	0	50	50	0	0	
0	0	0	50	50	0	0	
0	0	0	50	50	0	0	

Pixel representation of the receptive field

\*

0	0	0	0	0	0	30	0
0	0	0	0	0	30	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation =  $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$  (A large number!)

# Операция Свертки



Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

\*

0	0	0	0	0	0	30	0
0	0	0	0	0	30	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	0	0	0	0	0

Pixel representation of filter

# Операция Свертки



Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

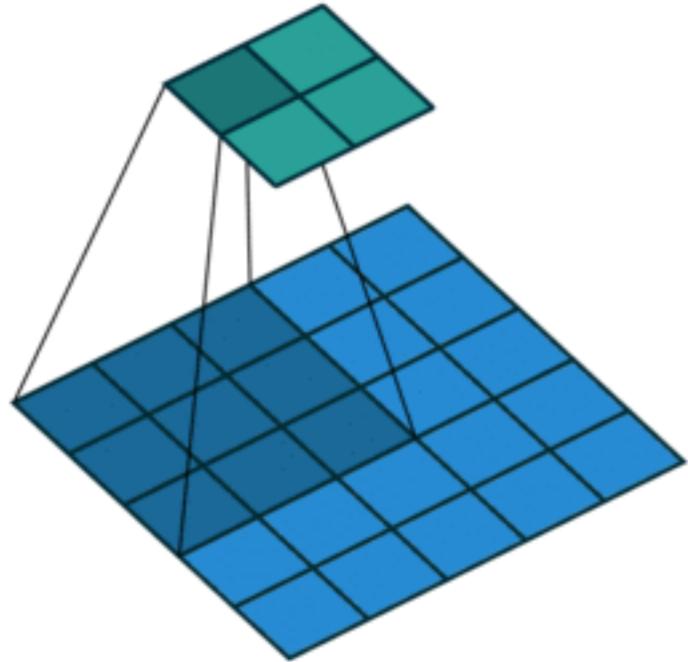
\*

0	0	0	0	0	0	30	0
0	0	0	0	0	30	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	0	0	0	0	0

Pixel representation of filter

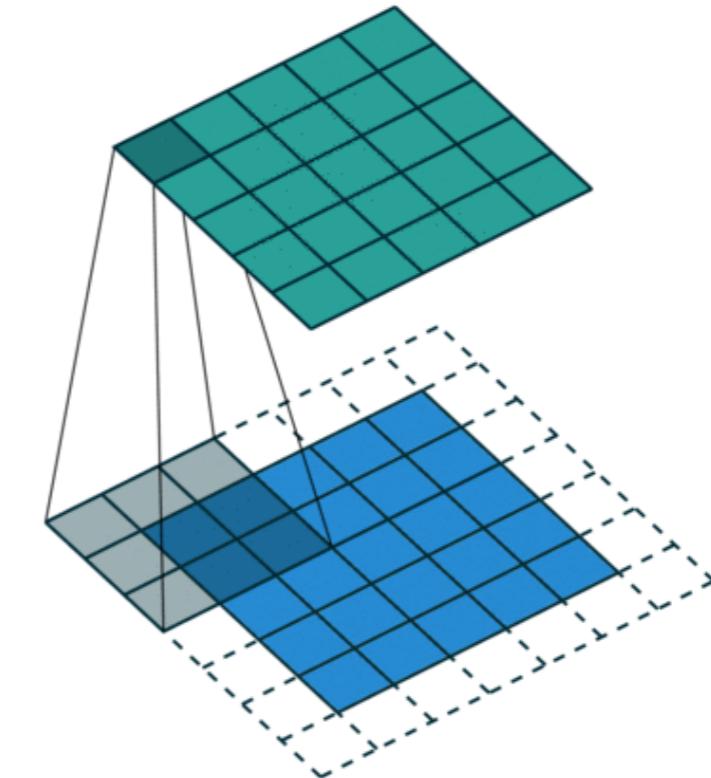
Multiplication and Summation = 0

# Операция Сдвига (stride) и дополнения (padding)



Stride снижает размерность (экономя тем самым вычислительные ресурсы), не теряя при этом много информации, поскольку изображения обладают свойством локальной коррелированности пикселей — соседние пиксели, как правило, не сильно отличаются друг от друга.

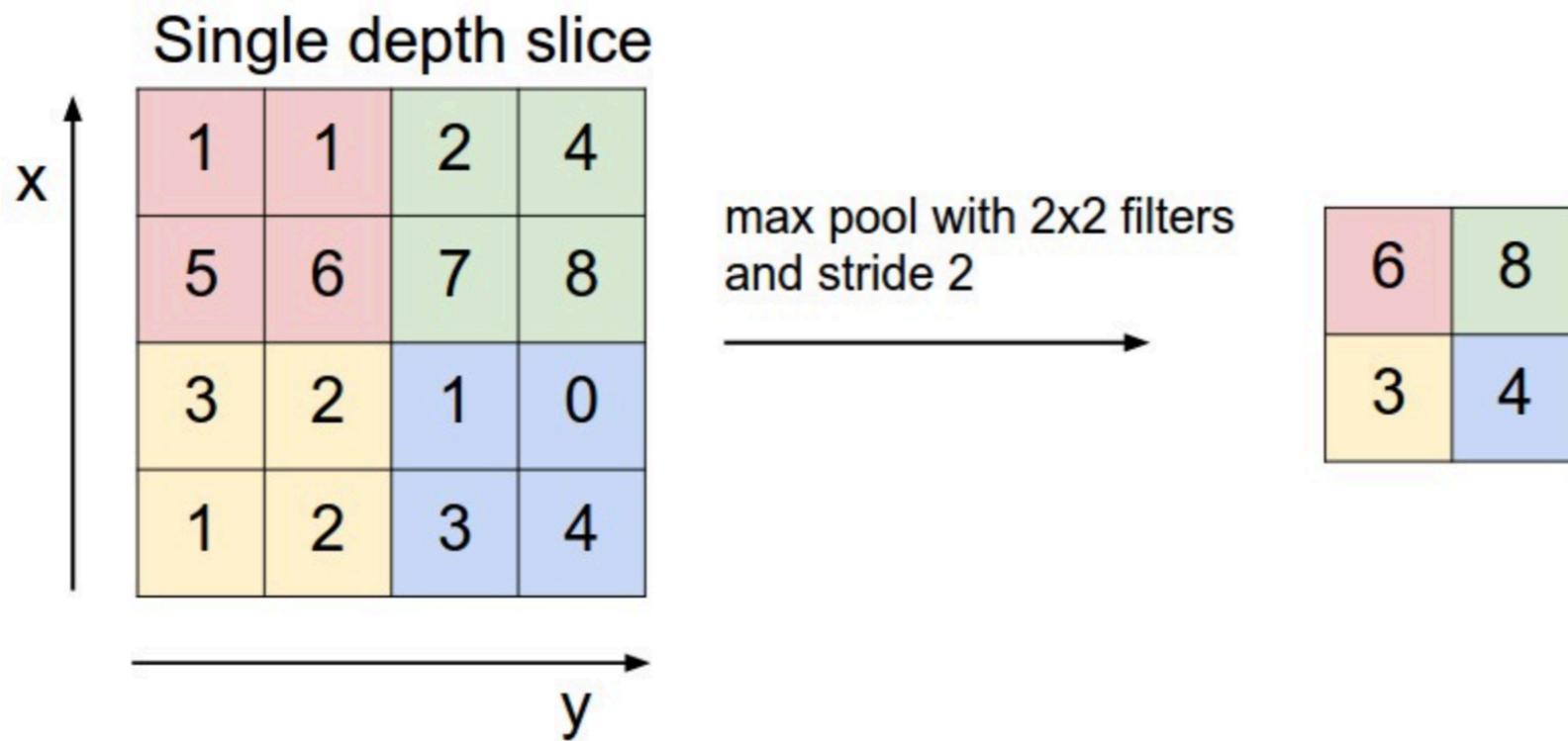
[https://raw.githubusercontent.com/Theano/Theano/master/doc/tutorial/conv\\_arithmetic\\_figures/no\\_padding\\_strides.gif](https://raw.githubusercontent.com/Theano/Theano/master/doc/tutorial/conv_arithmetic_figures/no_padding_strides.gif)



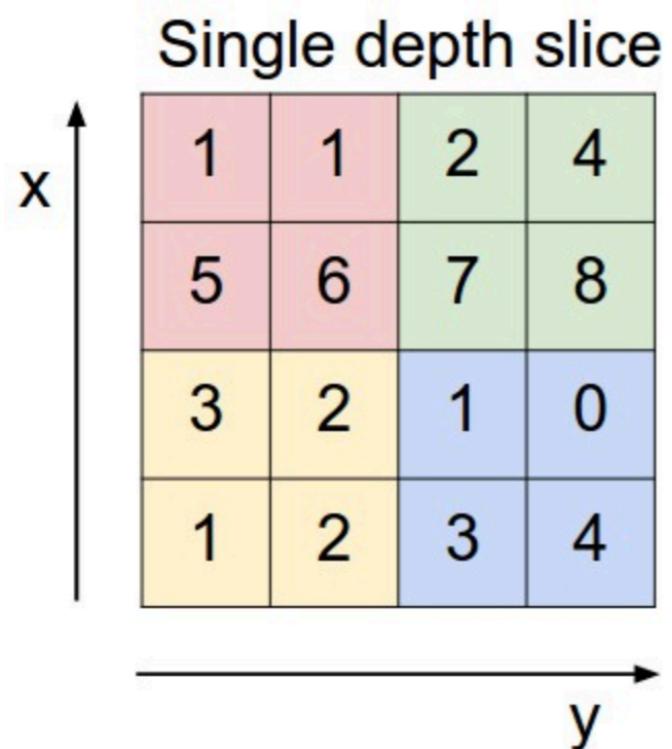
Padding используют, чтобы пространственная размерность картинки не уменьшалась после применения свертки или уменьшалась не так быстро

[https://raw.githubusercontent.com/Theano/Theano/master/doc/tutorial/conv\\_arithmetic\\_figures/same\\_padding\\_no\\_strides.gif](https://raw.githubusercontent.com/Theano/Theano/master/doc/tutorial/conv_arithmetic_figures/same_padding_no_strides.gif)

# Операция пулинга (pooling)

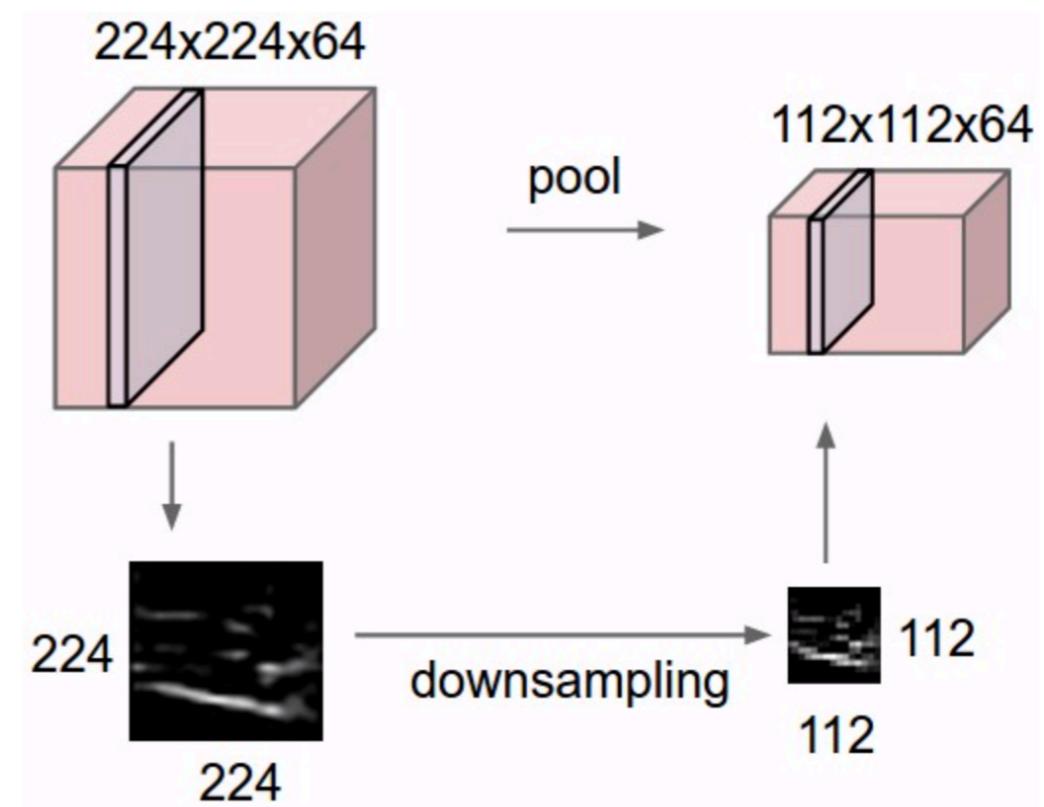


# Операция пулинга (pooling)

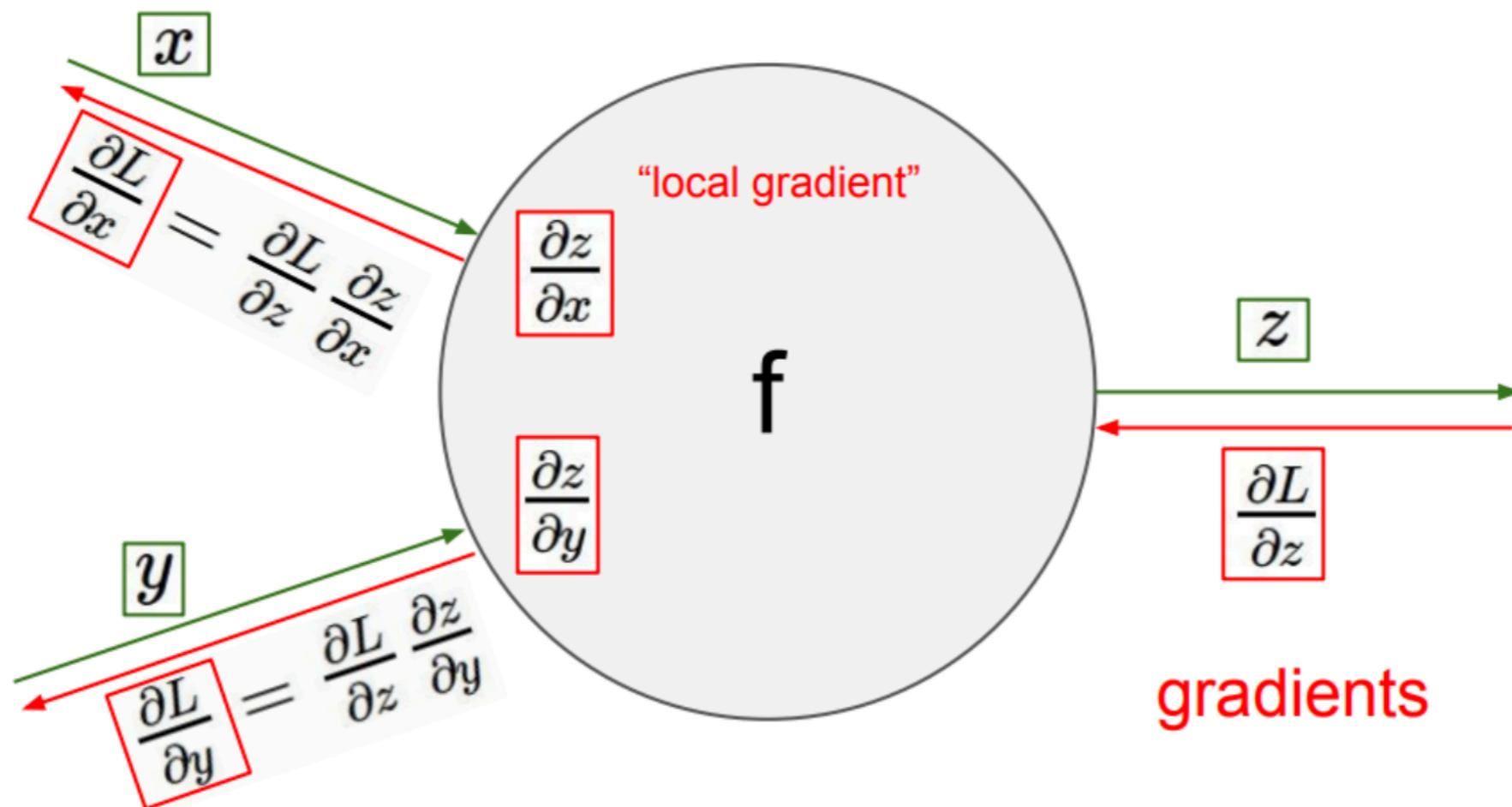


max pool with 2x2 filters  
and stride 2

6	8
3	4

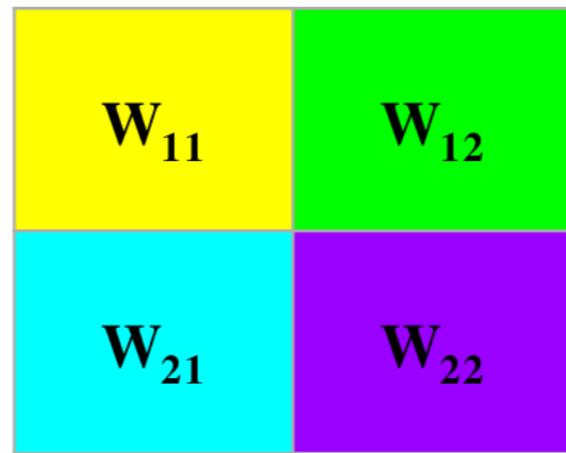


# Обучение нейронных сетей (reminder)



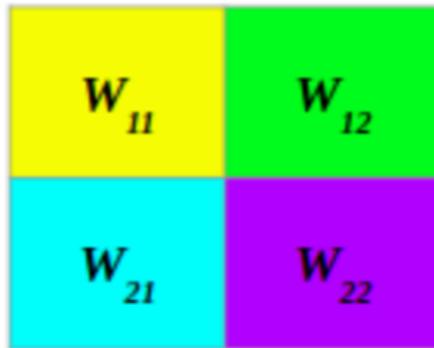
# Обучение CNN

$\mathbf{X}_{11}$	$\mathbf{X}_{12}$	$\mathbf{X}_{13}$
$\mathbf{X}_{21}$	$\mathbf{X}_{22}$	$\mathbf{X}_{23}$
$\mathbf{X}_{31}$	$\mathbf{X}_{32}$	$\mathbf{X}_{33}$



# Обучение CNN

$X_{11}$	$X_{12}$	$X_{13}$
$X_{21}$	$X_{22}$	$X_{23}$
$X_{31}$	$X_{32}$	$X_{33}$



$h_{11}$	$h_{12}$
$h_{21}$	$h_{22}$

Input Size : 3x3, Filter Size : 2x2, Output Size : 2x2

$$h_{11} = W_{11}X_{11} + W_{12}X_{12} + W_{21}X_{21} + W_{22}X_{22}$$

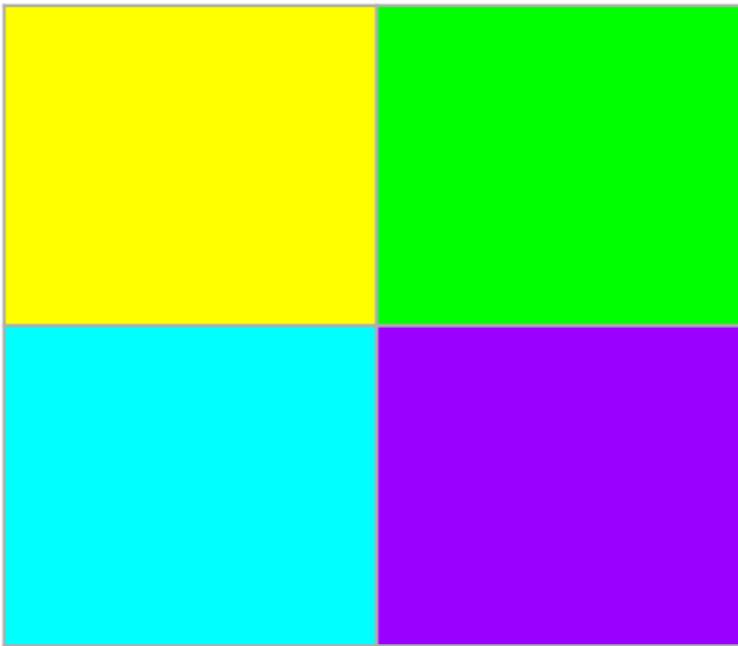
$$h_{12} = W_{11}X_{12} + W_{12}X_{13} + W_{21}X_{22} + W_{22}X_{23}$$

$$h_{21} = W_{11}X_{21} + W_{12}X_{22} + W_{21}X_{31} + W_{22}X_{32}$$

$$h_{22} = W_{11}X_{22} + W_{12}X_{23} + W_{21}X_{32} + W_{22}X_{33}$$

# Обучение CNN

$X_{11}$	$X_{12}$	$X_{13}$
$X_{21}$	$X_{22}$	$X_{23}$
$X_{31}$	$X_{32}$	$X_{33}$



$\partial h_{11}$	$\partial h_{12}$
$\partial h_{21}$	$\partial h_{22}$

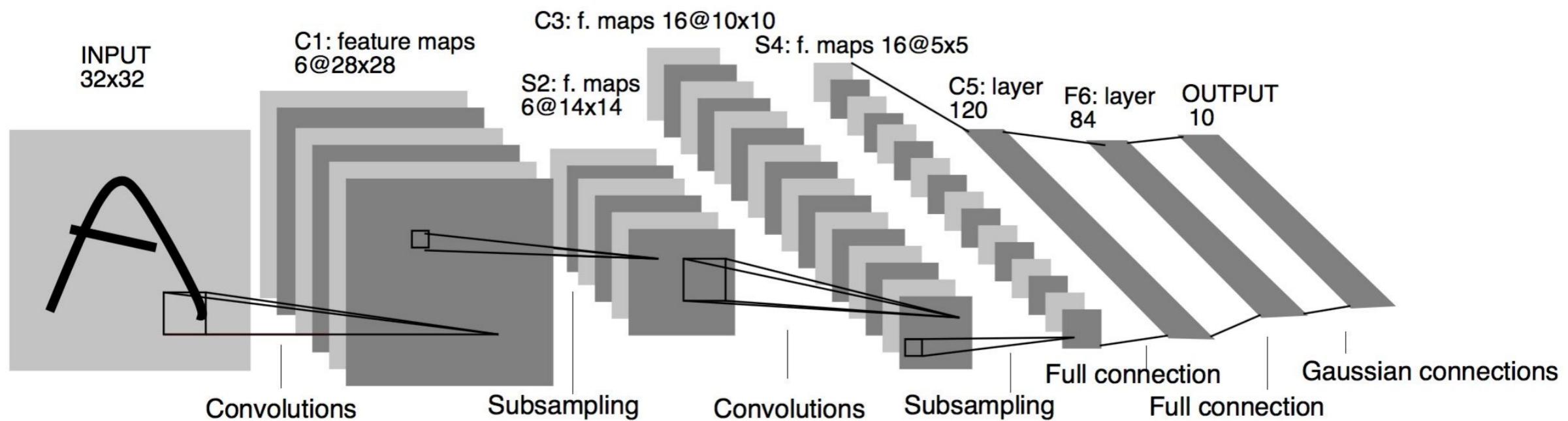
$$\partial W_{11} = X_{11} \partial h_{11} + X_{12} \partial h_{12} + X_{21} \partial h_{21} + X_{22} \partial h_{22}$$

$$\partial W_{12} = X_{12} \partial h_{11} + X_{13} \partial h_{12} + X_{22} \partial h_{21} + X_{23} \partial h_{22}$$

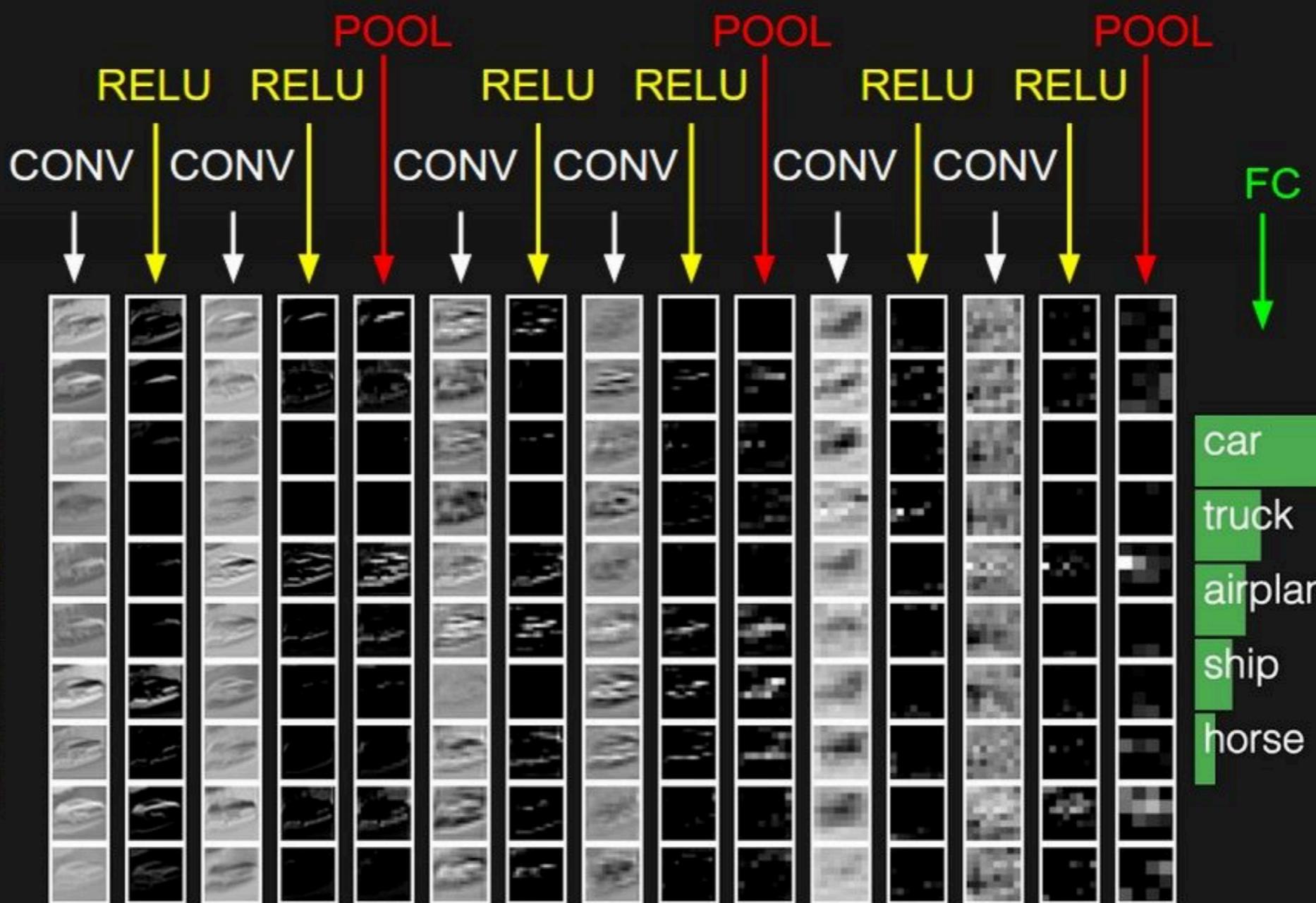
$$\partial W_{21} = X_{21} \partial h_{11} + X_{22} \partial h_{12} + X_{31} \partial h_{21} + X_{32} \partial h_{22}$$

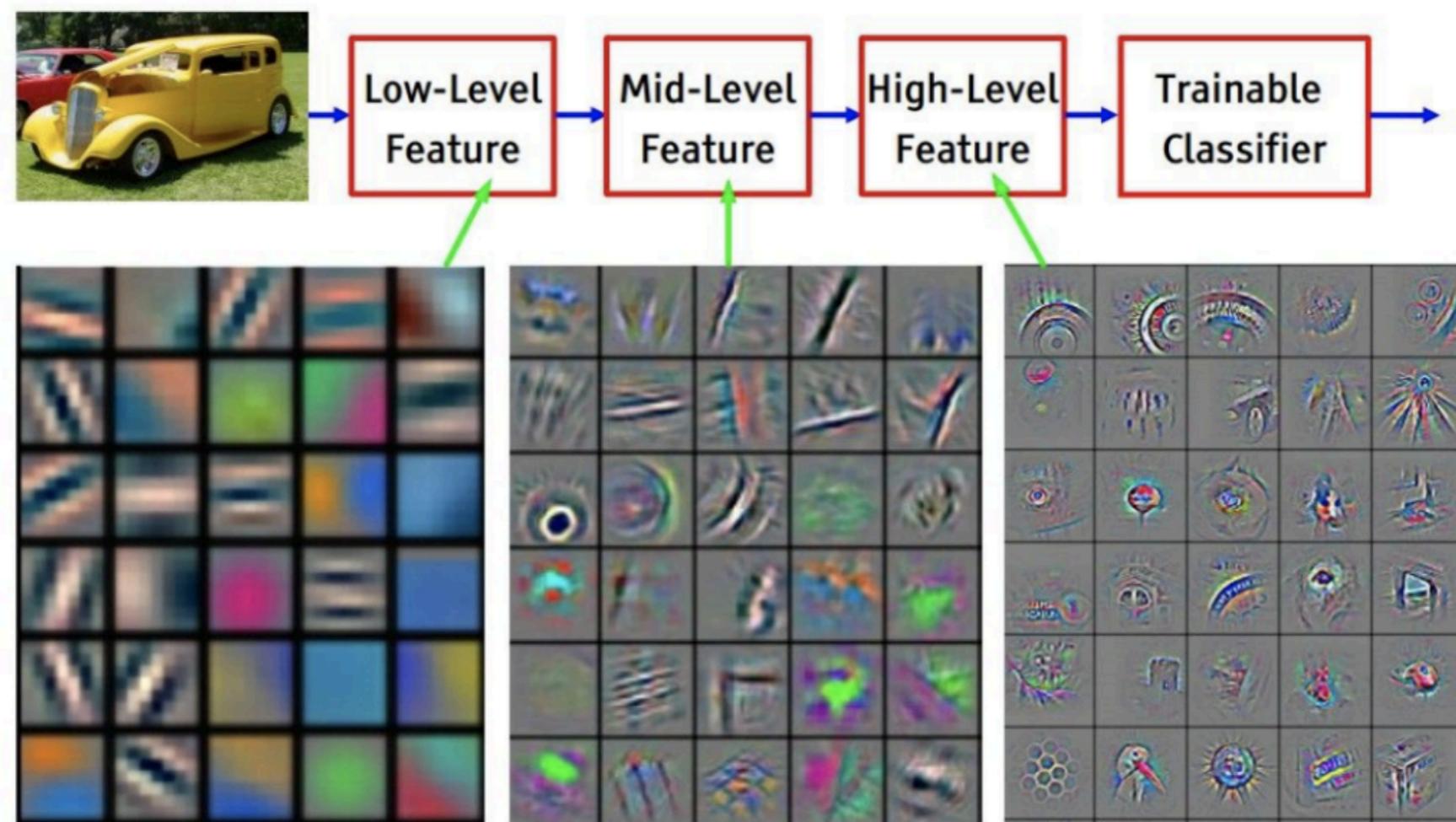
$$\partial W_{22} = X_{22} \partial h_{11} + X_{23} \partial h_{12} + X_{32} \partial h_{21} + X_{33} \partial h_{22}$$

# LeNet

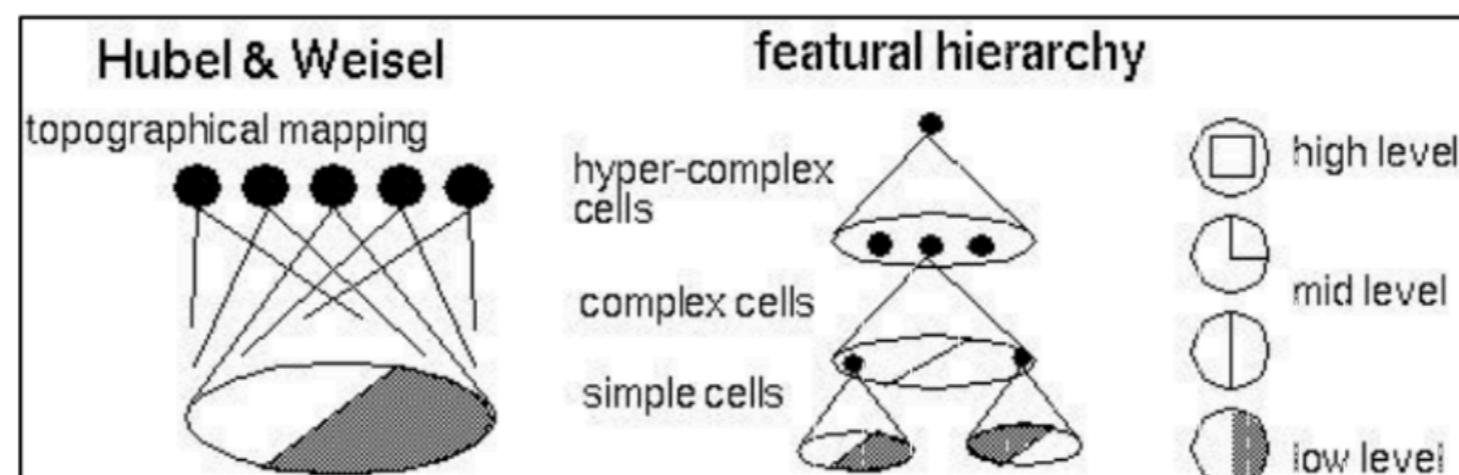


- Свертки  $5 \times 5$  со сдвигом 1 и пулинг  $2 \times 2$  со сдвигом 2
- 60 тысяч параметров
- Практическое применение: с помощью этой сети на почте США распознавали рукописные индексы

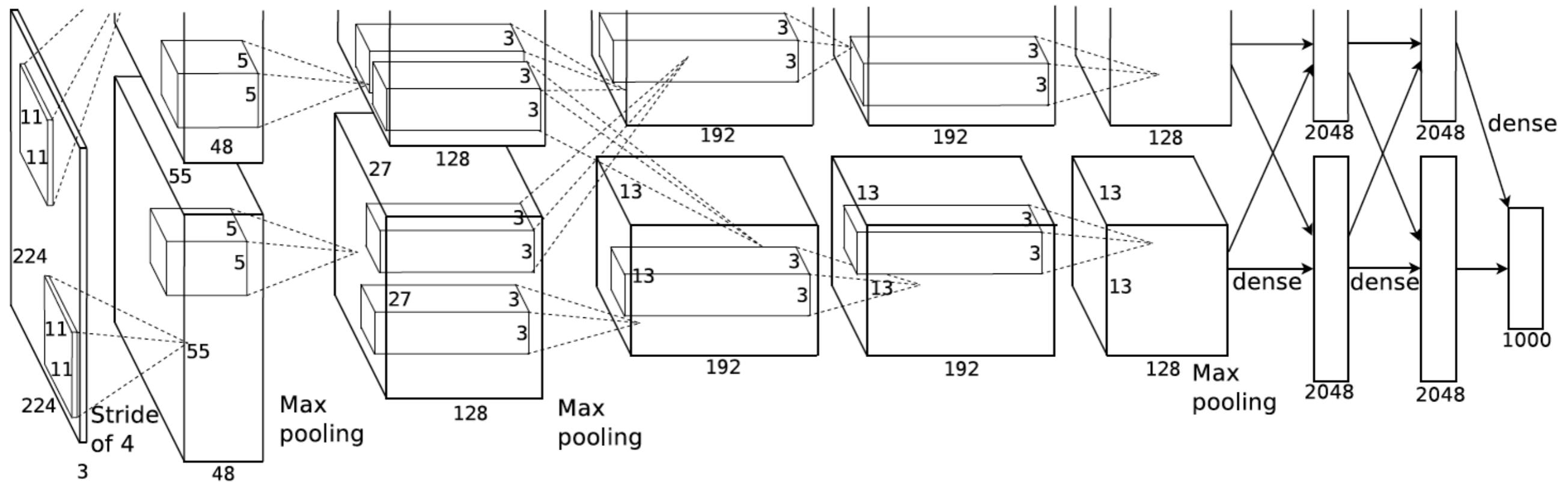




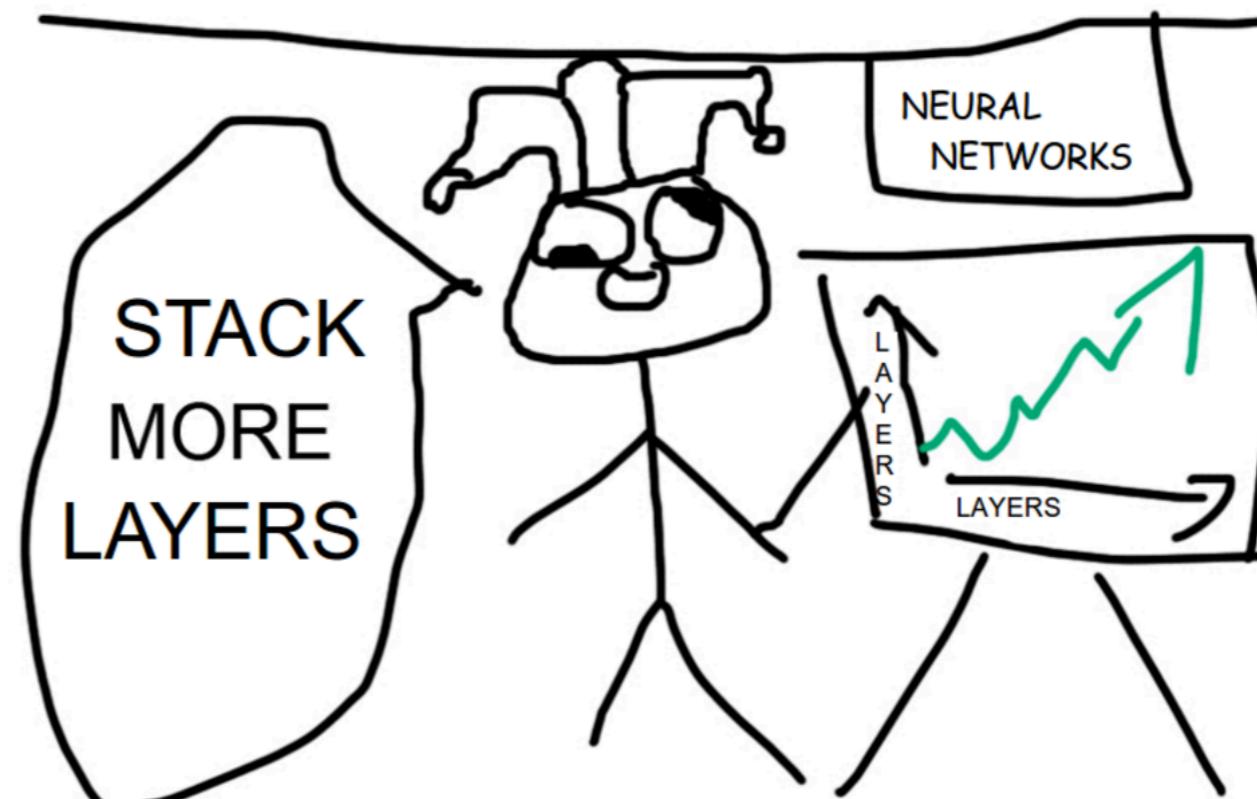
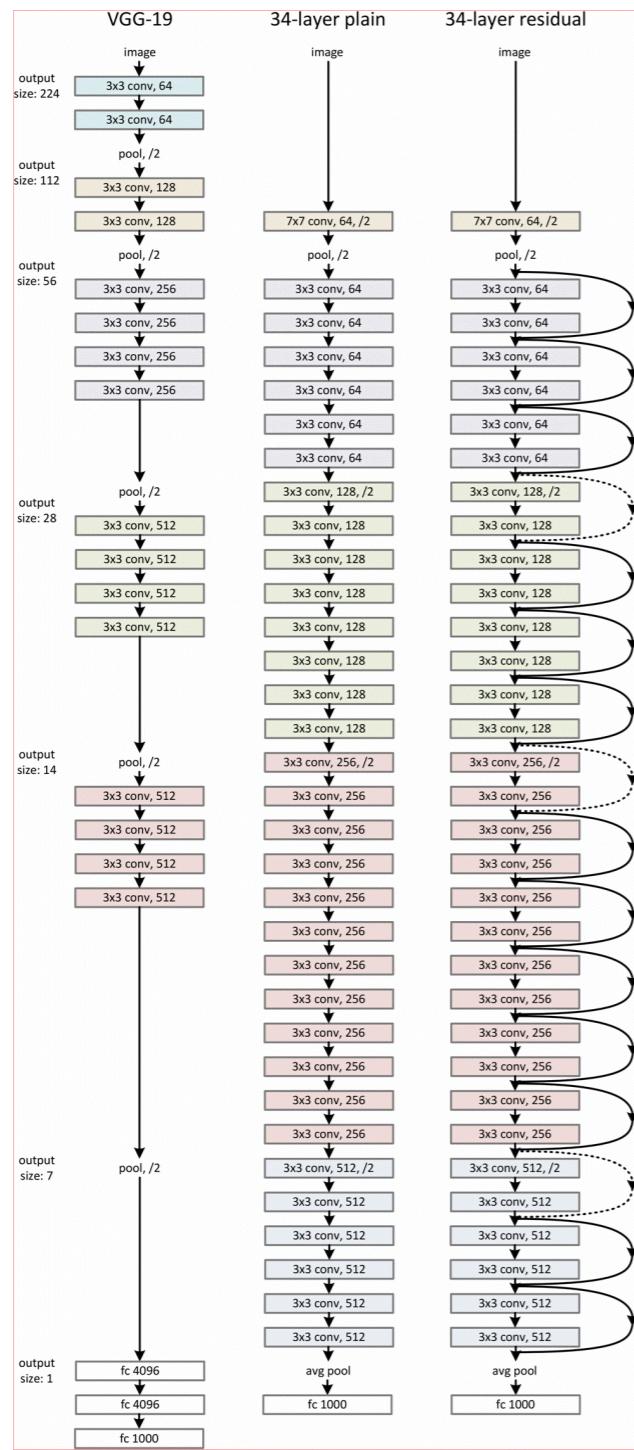
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]



# AlexNet

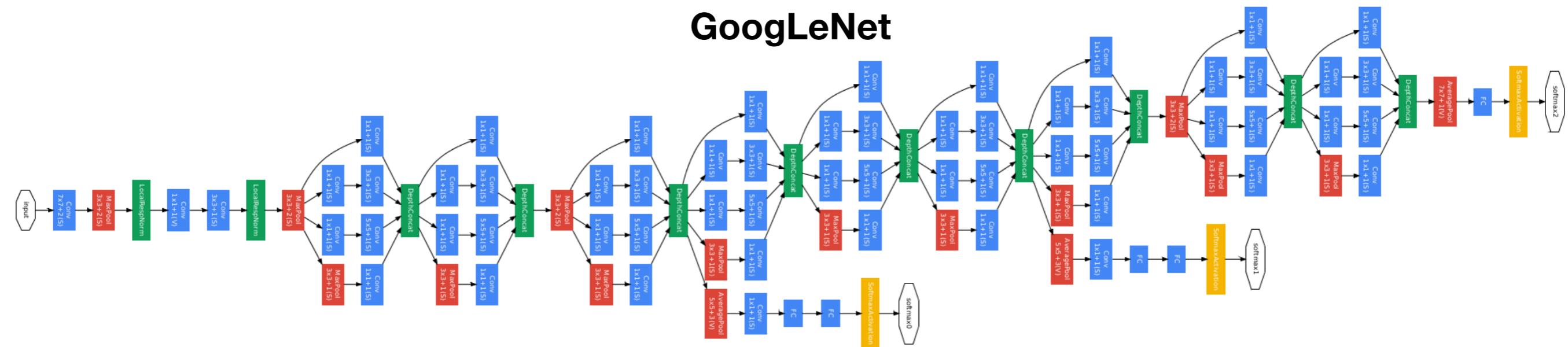


# Быстрее, выше, сильнее



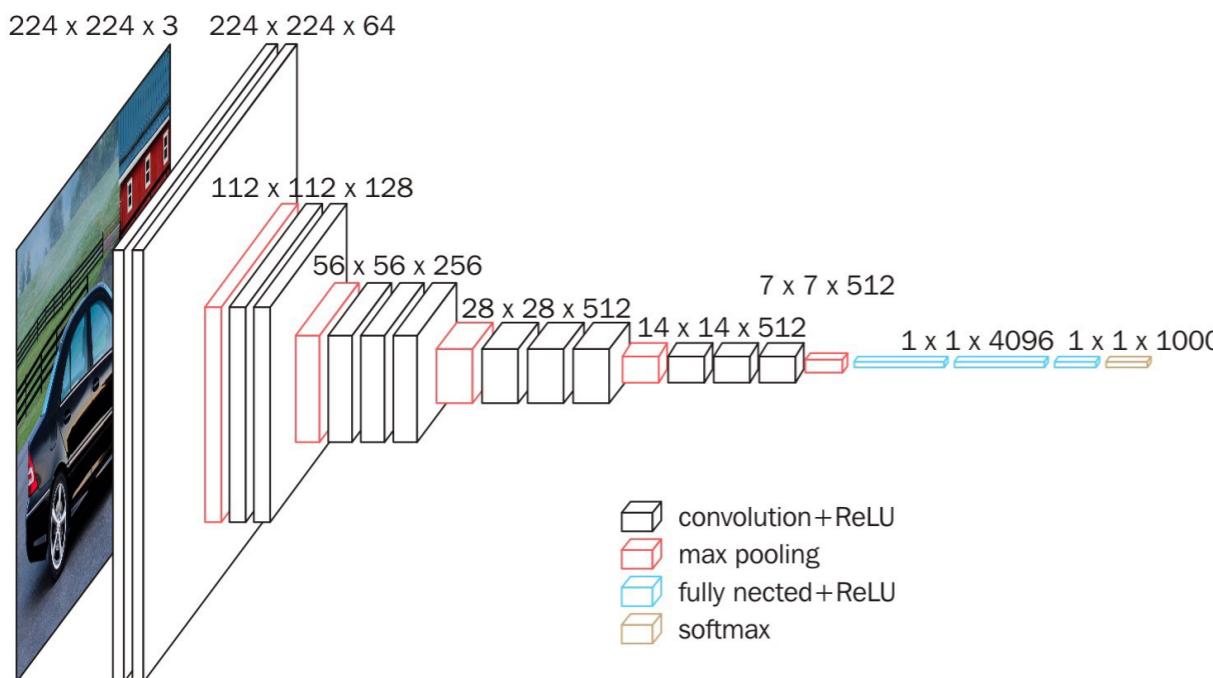
# Архитектуры CNN

## GoogLeNet



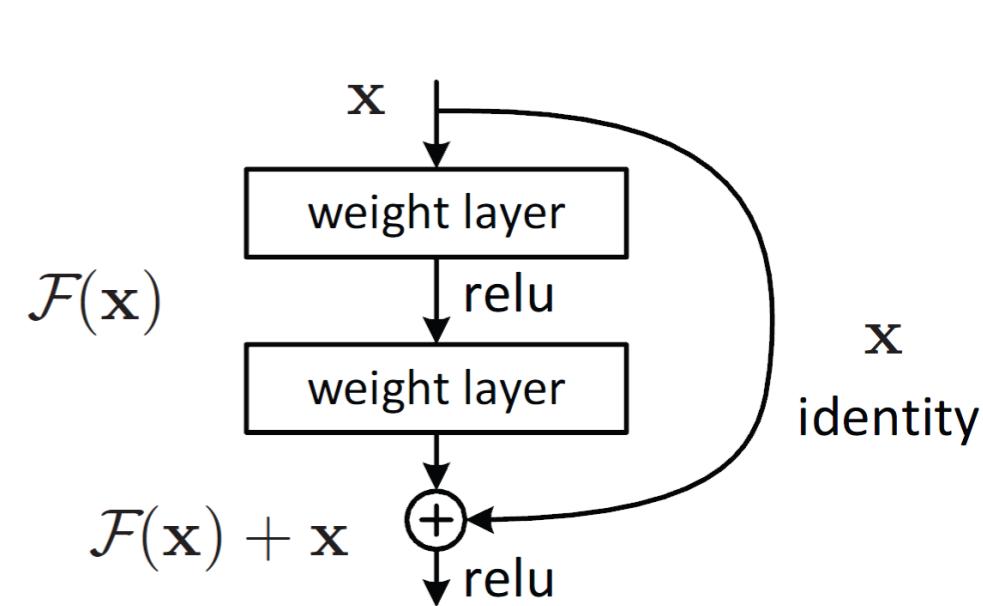
Подробнее: <https://habr.com/ru/post/301084/>

## VGG

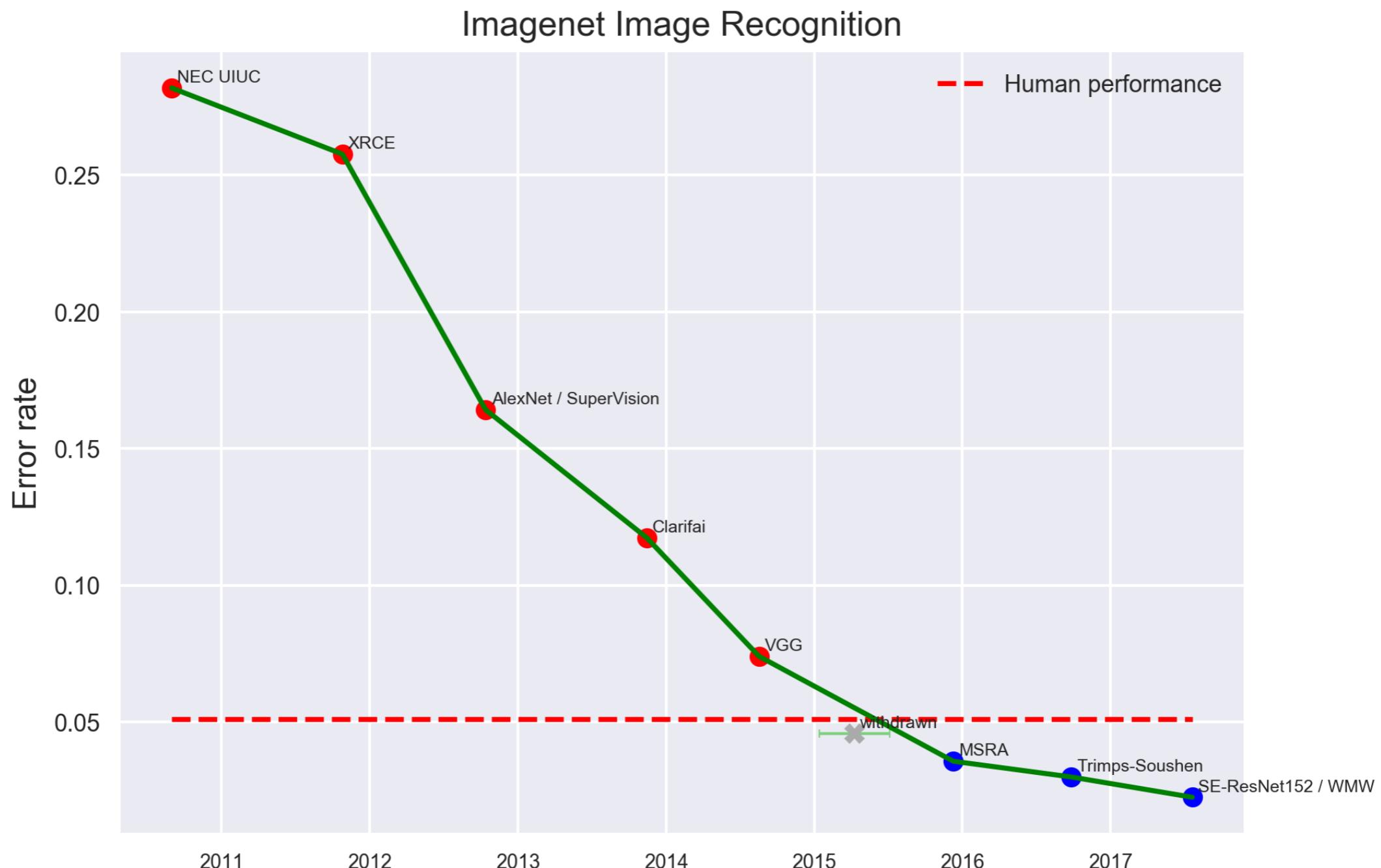


- convolution + ReLU
- max pooling
- fully connected + ReLU
- softmax

## ResNet



# Превосходство NN над человеком

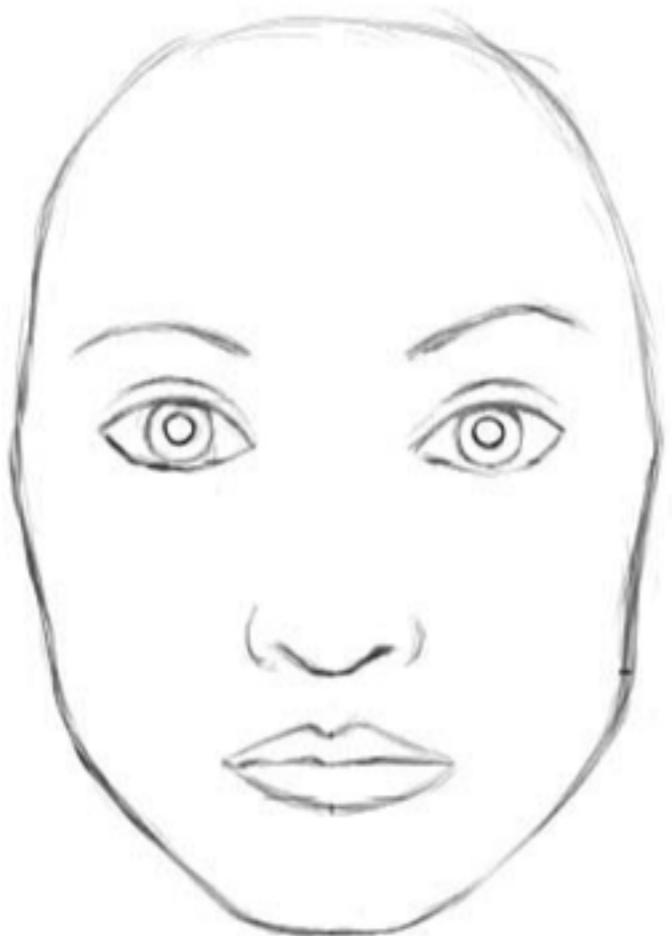


<https://www.eff.org/ai/metrics>

Быстрее, выше, сильнее

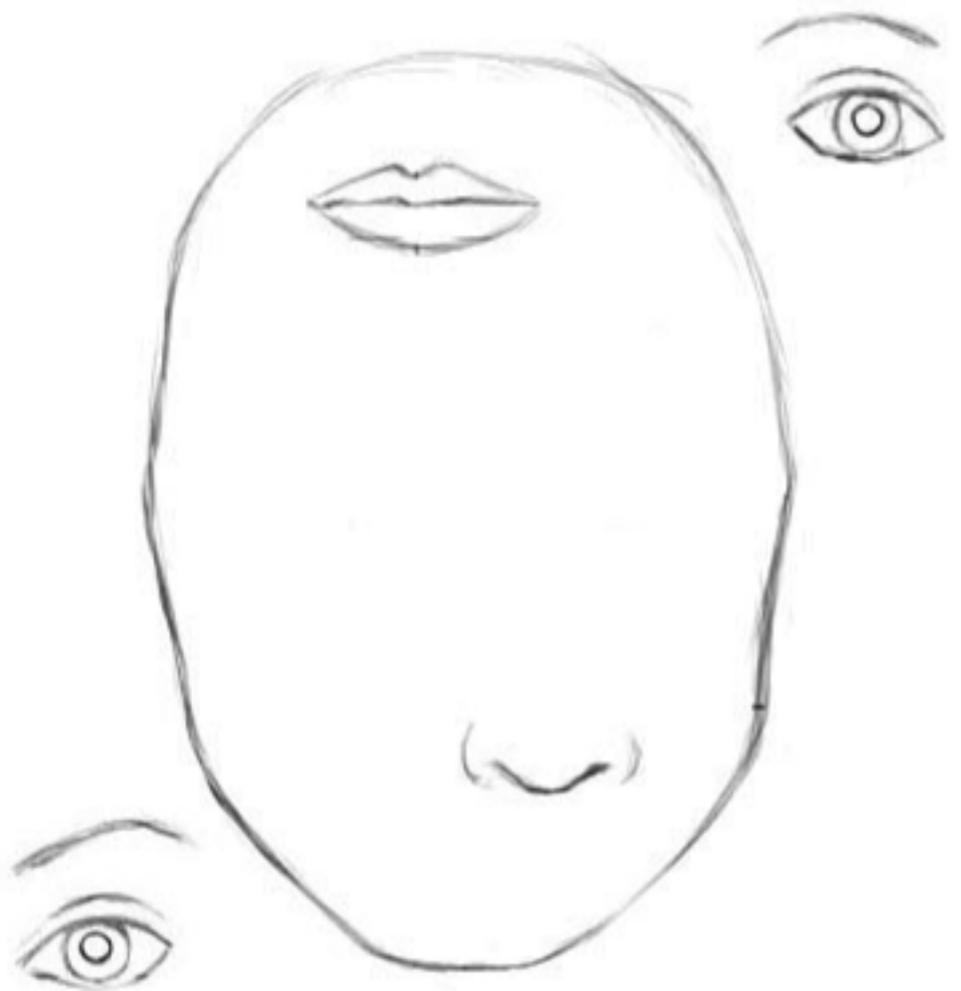


# Проблемы



*Определит ли нейросеть, что на изображении лицо?*

# Проблемы

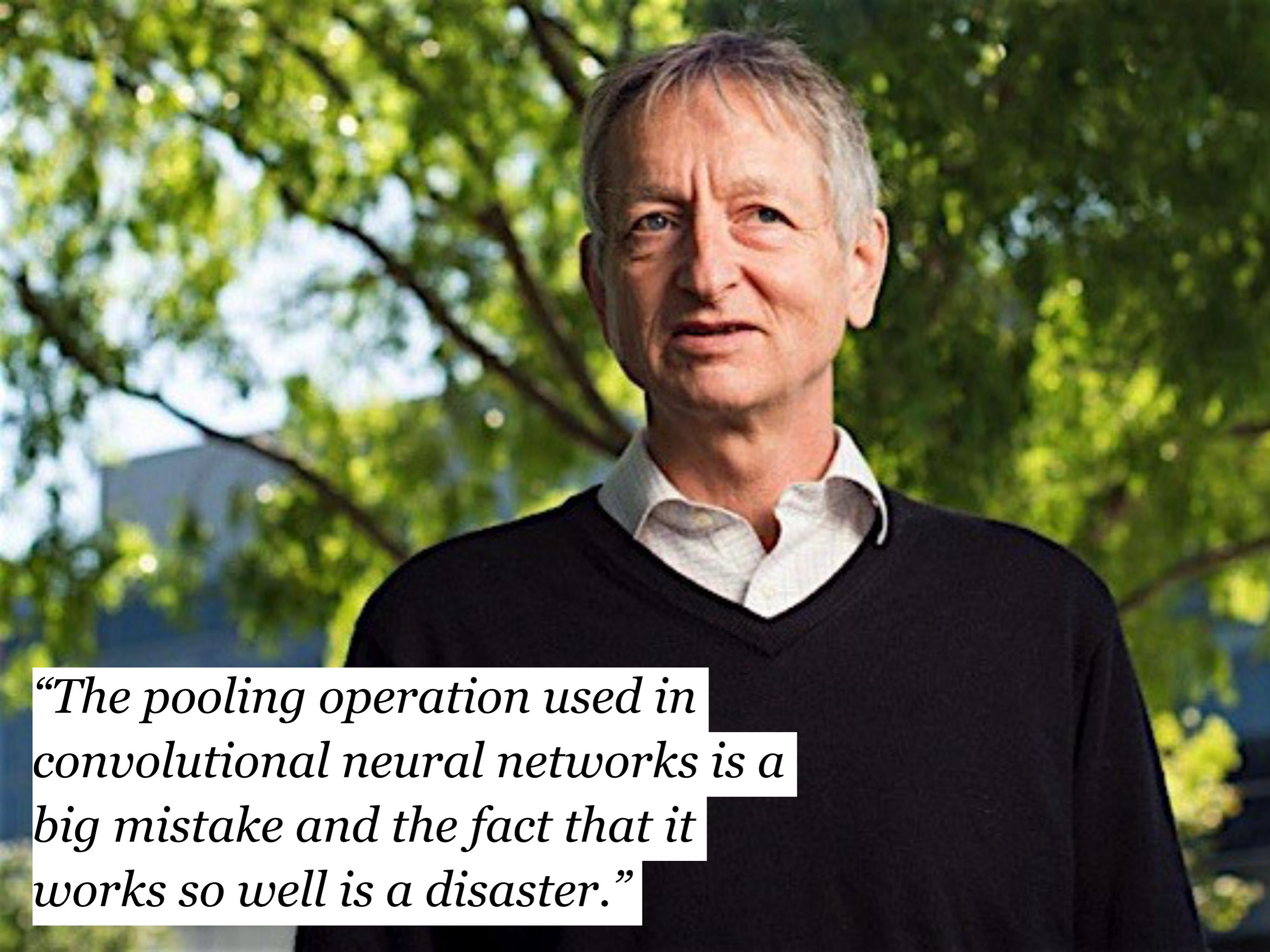


А так?

# Проблемы



Ваш мозг может легко распознать, что это один и тот же объект, даже если все фотографии сделаны под разными углами. CNNs не имеют такой возможности.



*“The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster.”*