

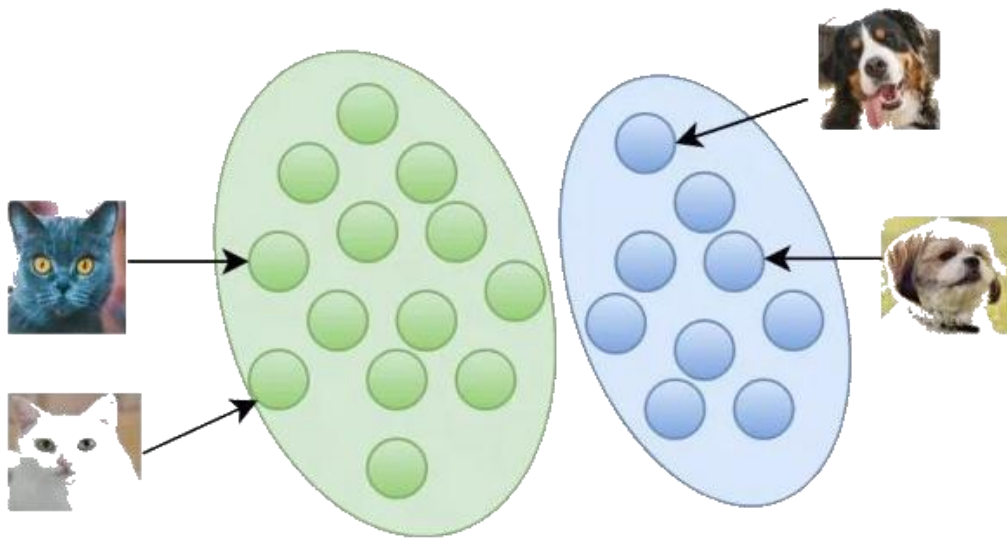


ИТМО

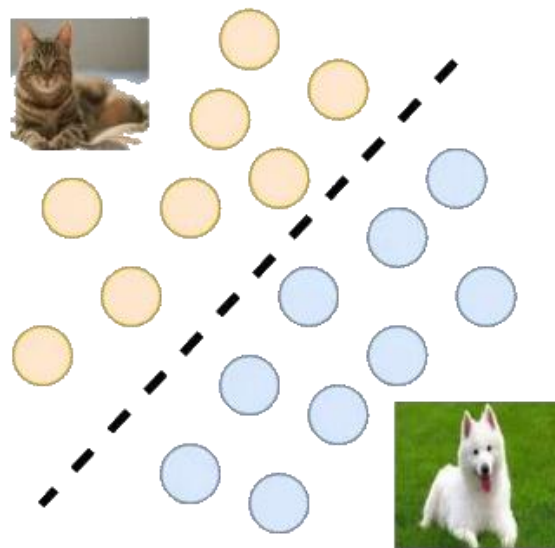
Convolutional Neural Networks (CNNs)

Современные архитектуры нейронных сетей

Дескриминативные vs генеративные задачи



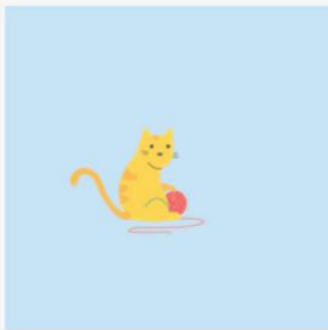
генеративная



дискриминативная

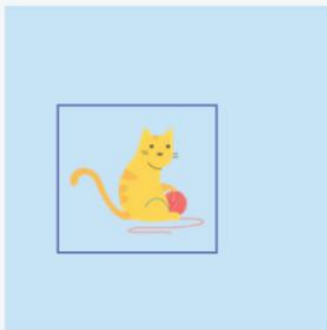
Computer Vision Problem Types

Classification



CAT

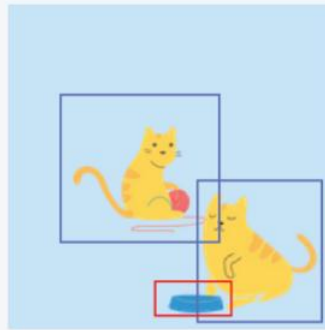
Classification
+ Localization



CAT

Single Object

Object Detection



CAT, CAT, BOWL

Multiple Objects

Semantic
Segmentation

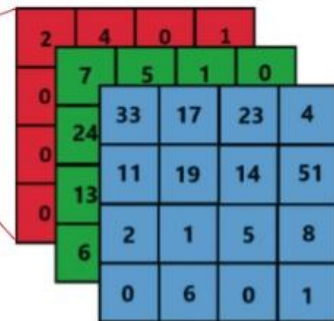
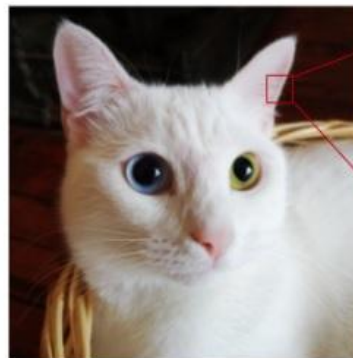
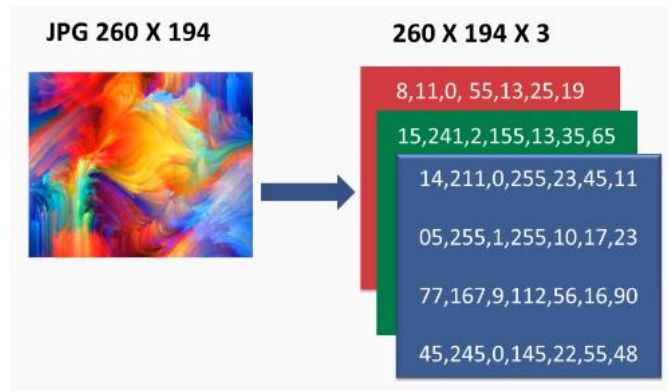


CAT, CAT, BOWL

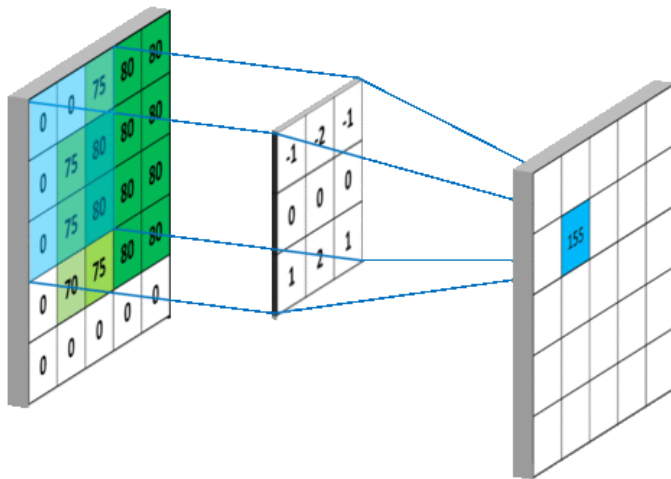
Что такое изображение?

Изображение как **тензор** размера $(h \times w \times c)$.

- h — высота, w — ширина, c — каналы (RGB).
- Каждый канал — интенсивность красного, зелёного, синего.

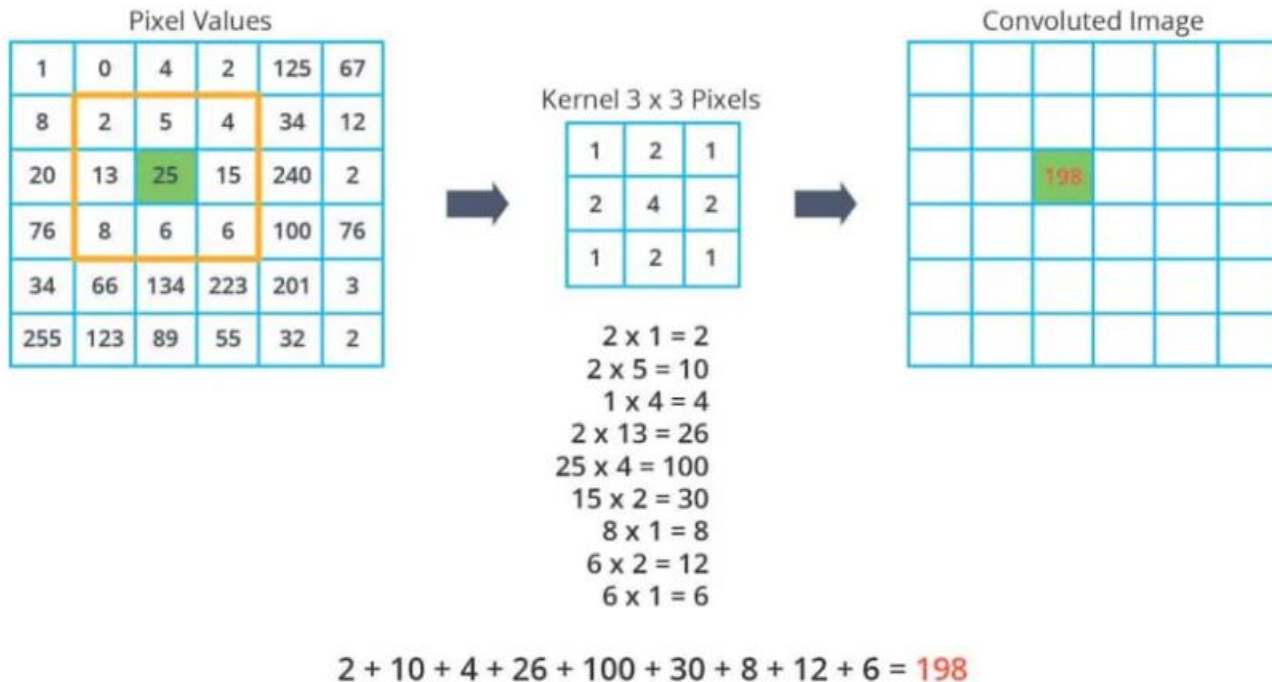


Что такое свертка?

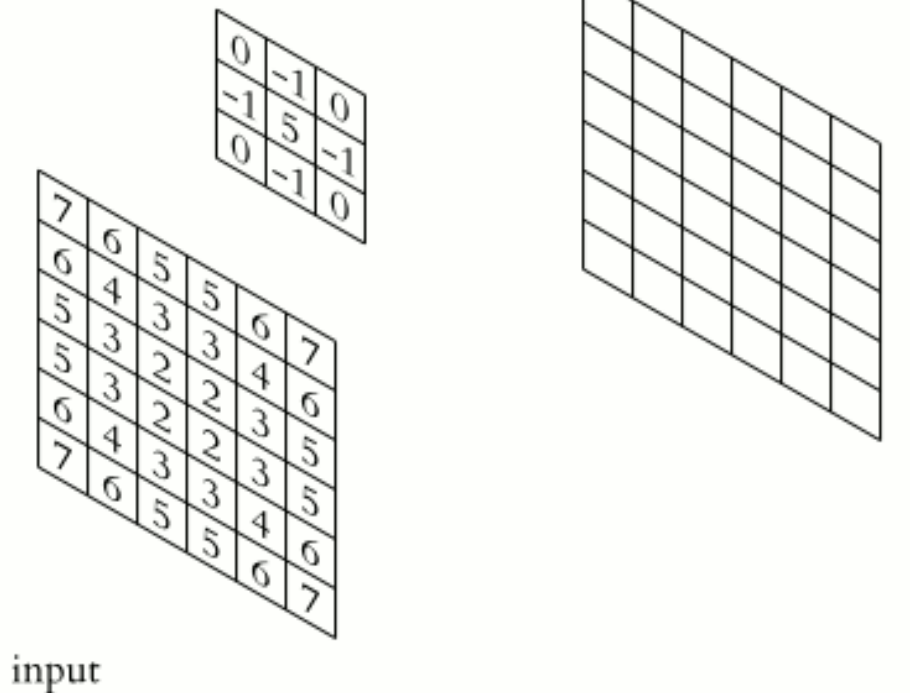


Свёртка (convolution) в компьютерном зрении — это операция, при которой маленькое **ядро (kernel, фильтр)** размера $k \times k$ «скользит» по изображению и в каждой позиции считает скалярное произведение между ядром и соответствующим фрагментом изображения.

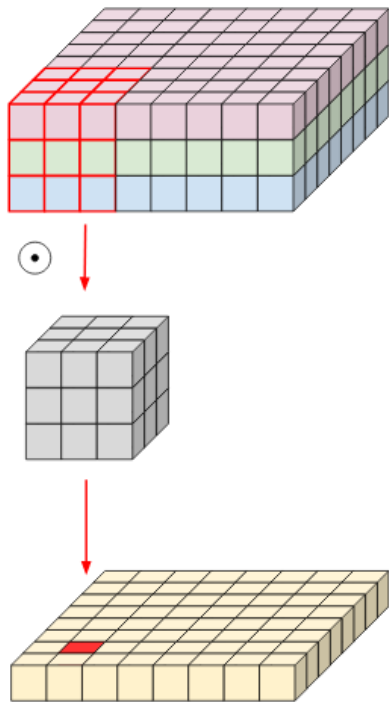
Как работает свертка?



Как работает свертка?



Как работает свертка?



Важно:

Параметров у **одного** фильтра свертки

- $k * k * C_{input}$

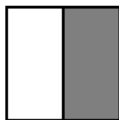
Если применить **C_{output}** фильтров свертки

- $C_{output} * k * k * C_{input}$

Для чего они нужны?

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

6 x 6



*

1	0	-1
1	0	-1
1	0	-1

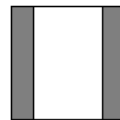
3 x 3



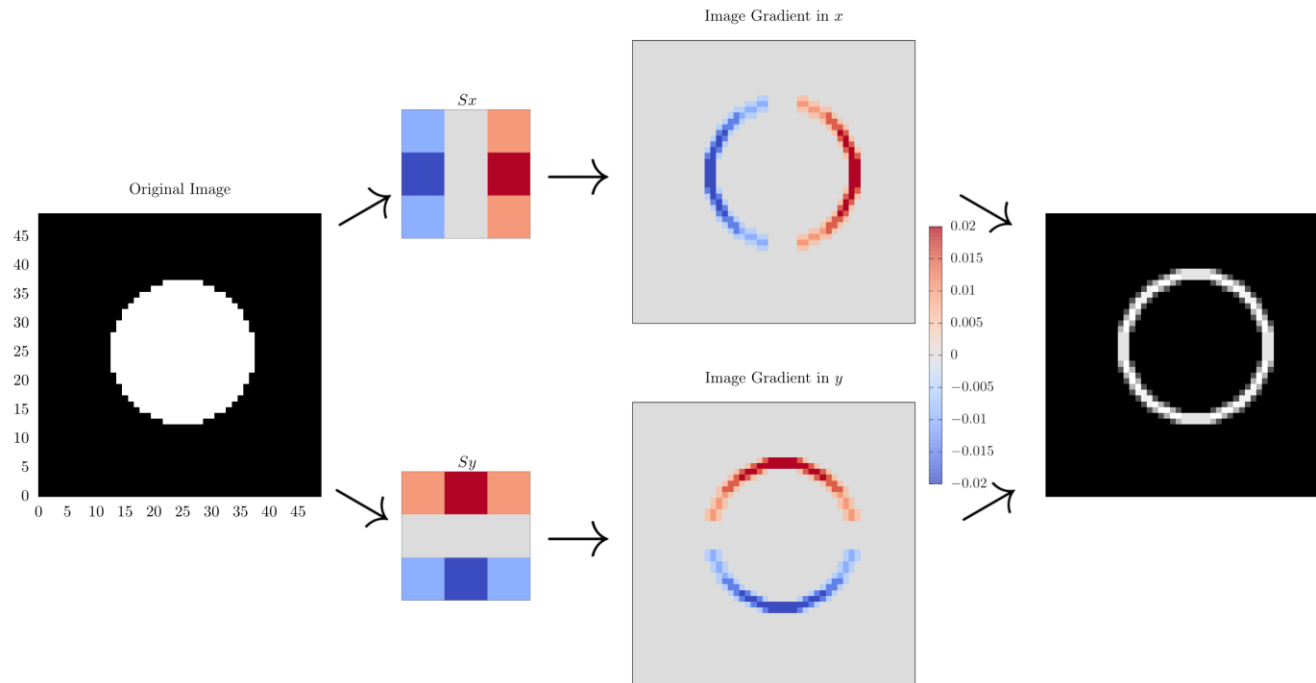
=

-0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

4 x 4

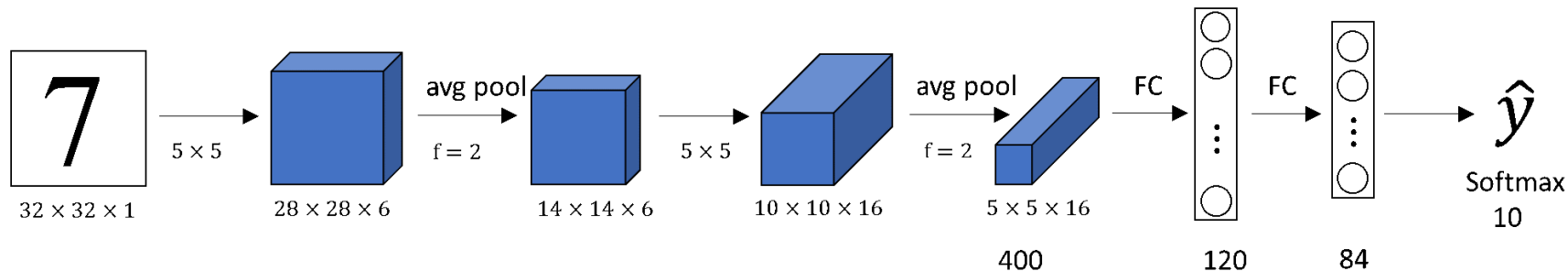


Для чего они нужны?

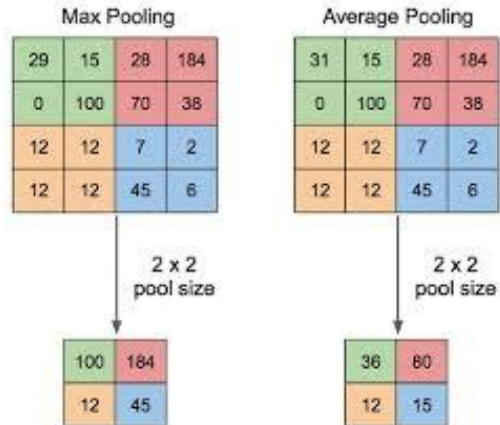


Свертки нужны для нахождения паттернов на изображении

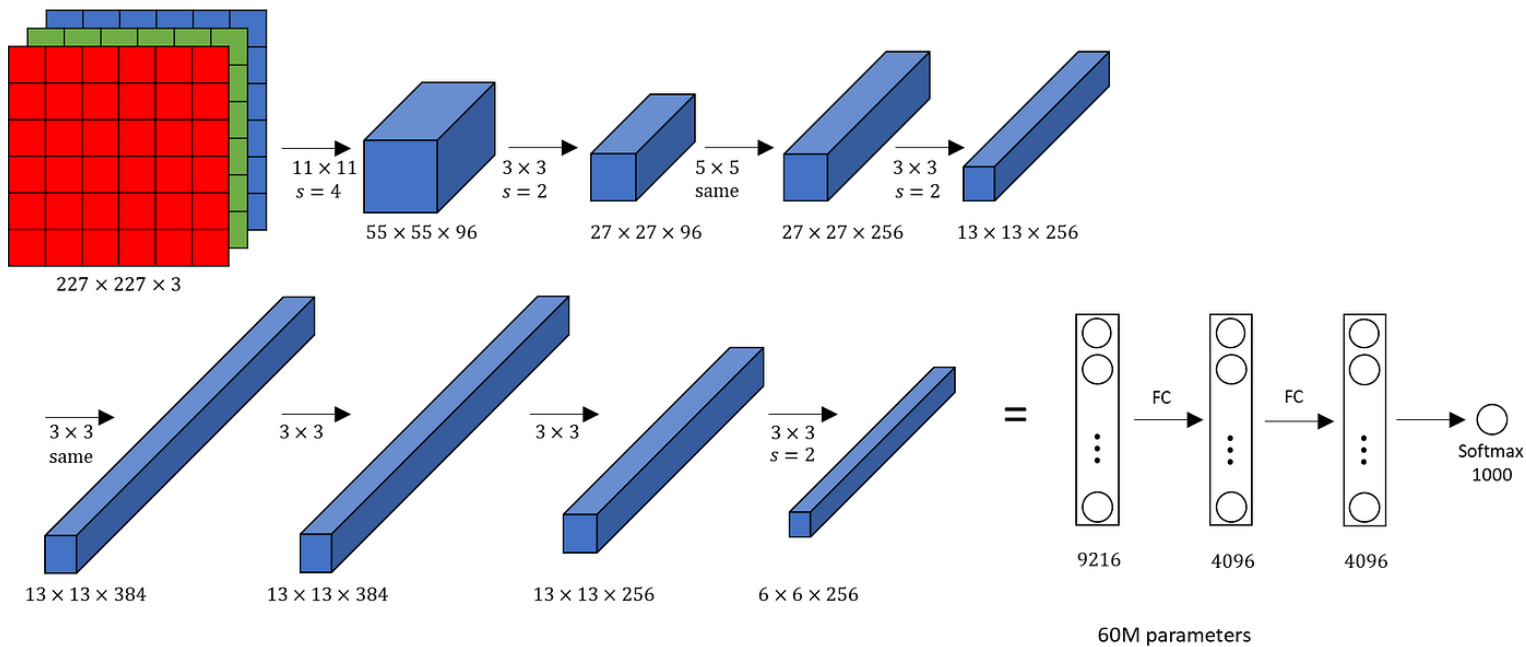
Первая сверточная сеть LeNet (1990-е)



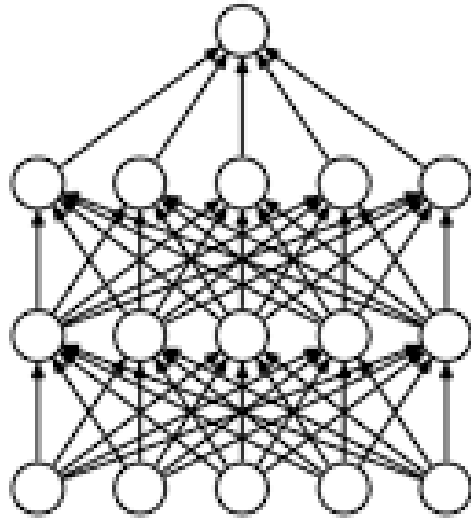
Average and Max Pooling



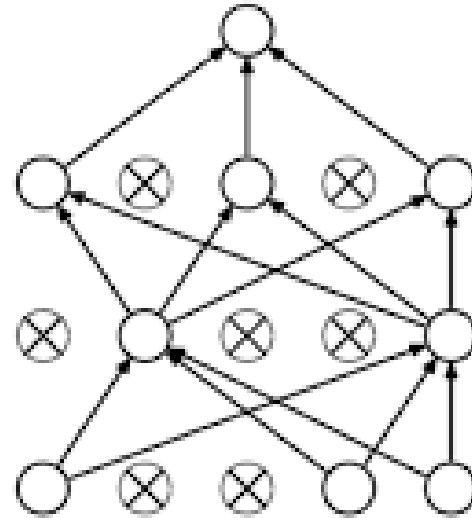
AlexNet (2012)



Dropout



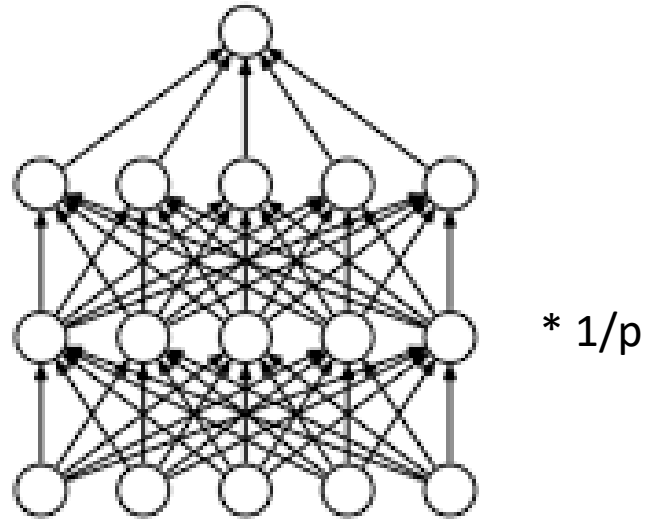
(a) Standard Neural Net



(b) After applying dropout.

Dropout($p=0.5$)

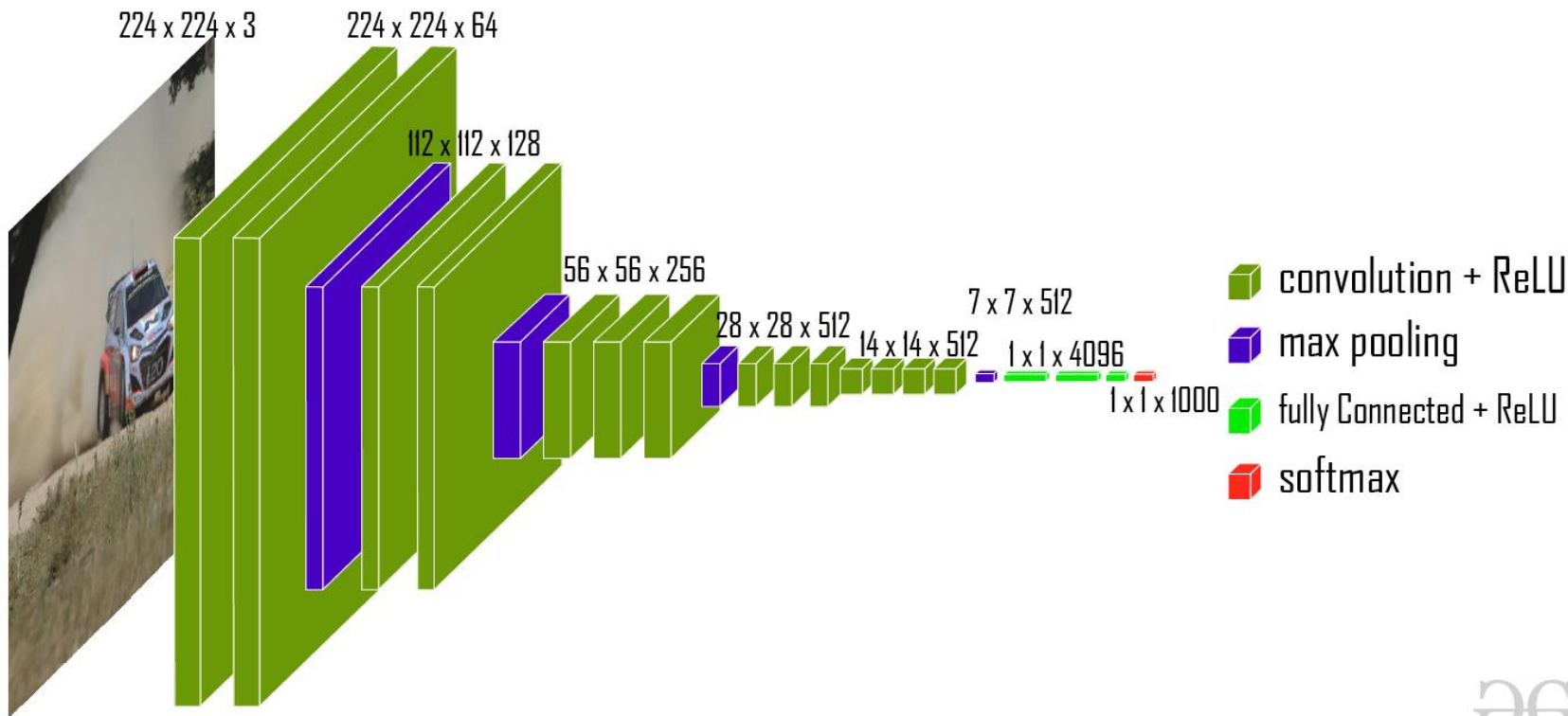
Dropout inference



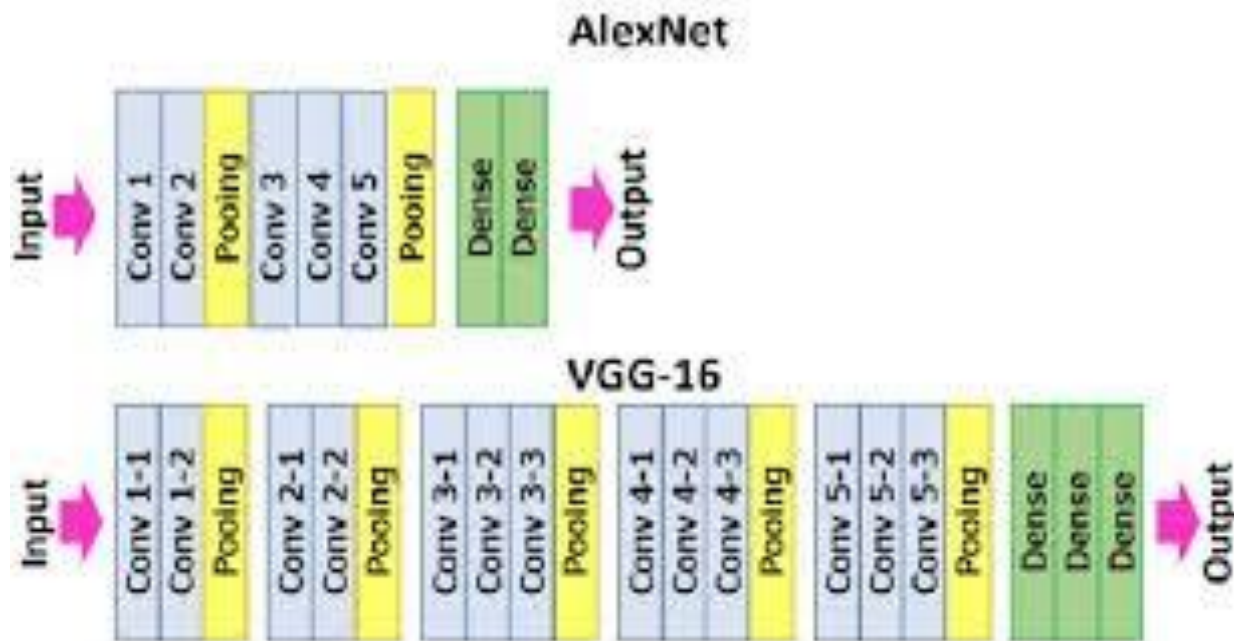
(a) Standard Neural Net

Dropout($p=0.5$)

VGG-16(2014)



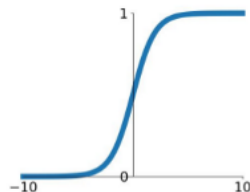
VGG-16 vs AlexNet



Activation Functions

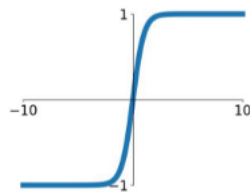
Sigmoid

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



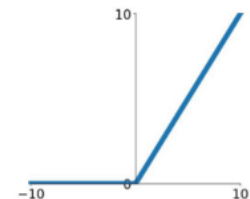
tanh

$$\tanh(x)$$



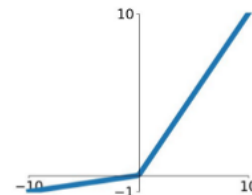
ReLU

$$\max(0, x)$$



Leaky ReLU

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

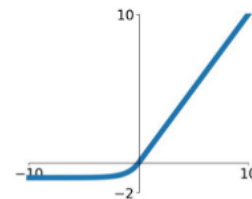


Maxout

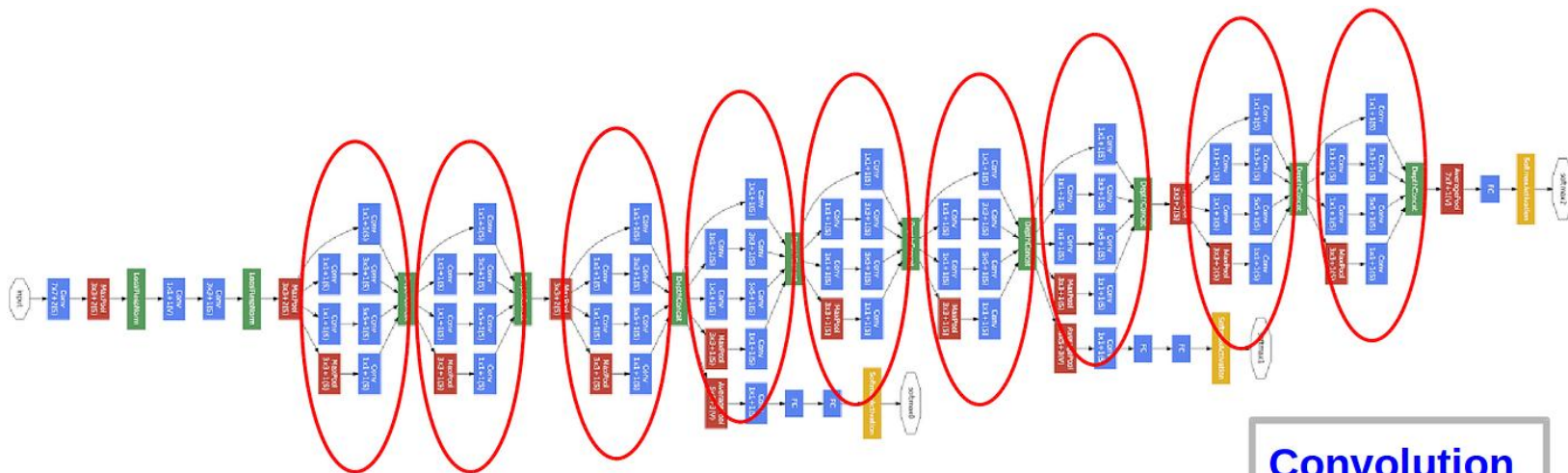
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

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

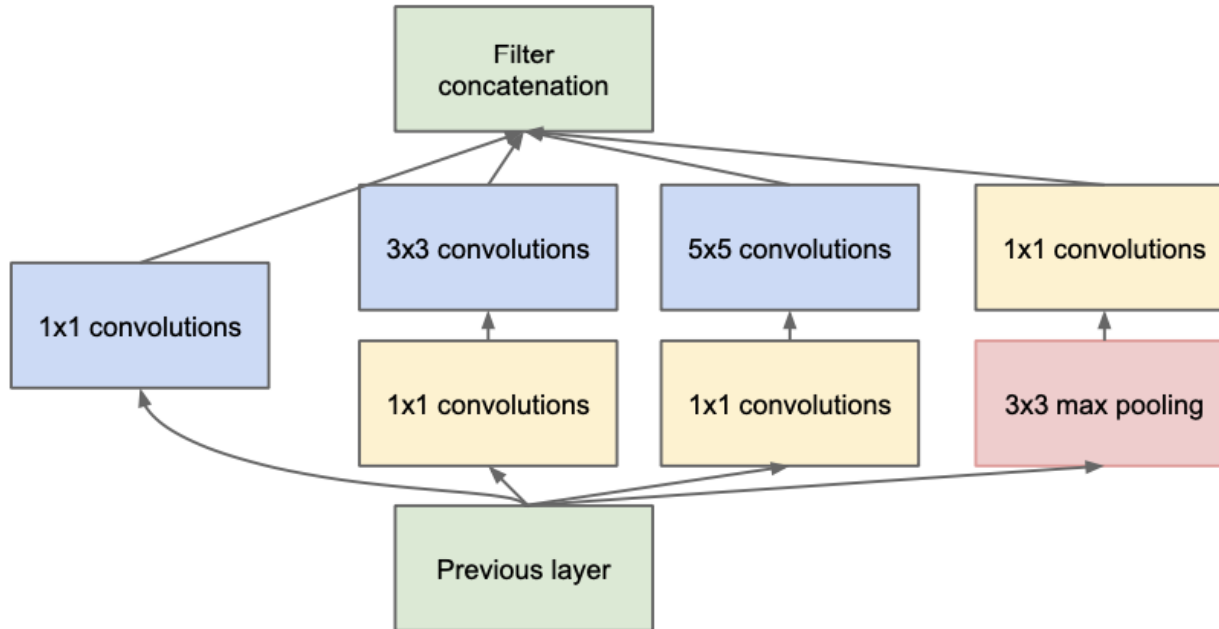


GoogleNet(2014)

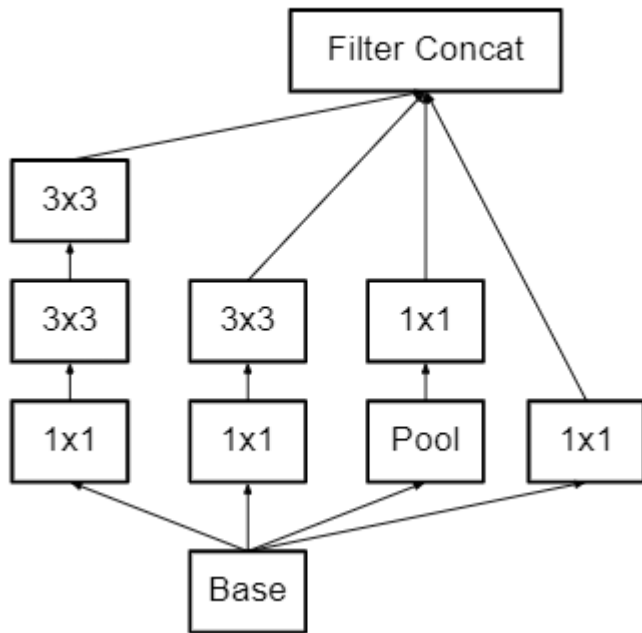


Convolution
Pooling
Softmax
Concat/Normalize

Inception



Inception v2 (2015)

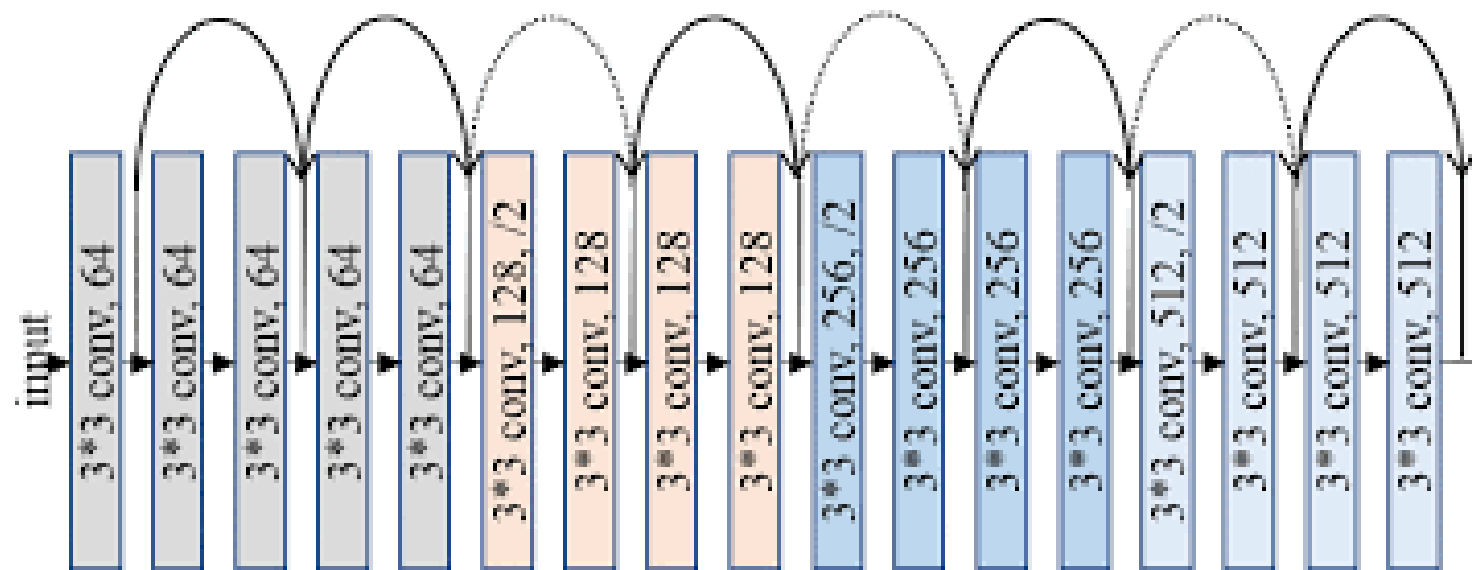


Convolution 5x5 vs 3x3-twice

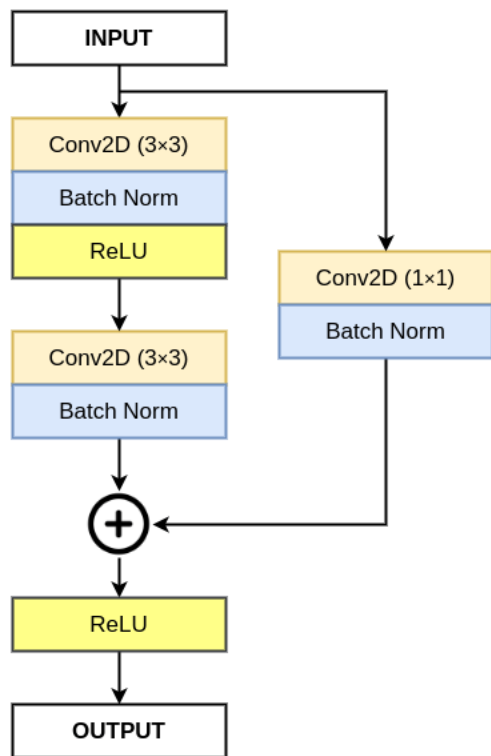
$$C_{\text{input}} * 5 * 5 * C_{\text{output}} = \\ \mathbf{25 * C_{\text{input}} * \text{output}}$$

$$2 * (C_{\text{input}} * 3 * 3 * \\ C_{\text{output}}) = \\ \mathbf{18 * C_{\text{input}} * \text{output}}$$

ResNet (2015)



Residual connection and Batch Norm



Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;

Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

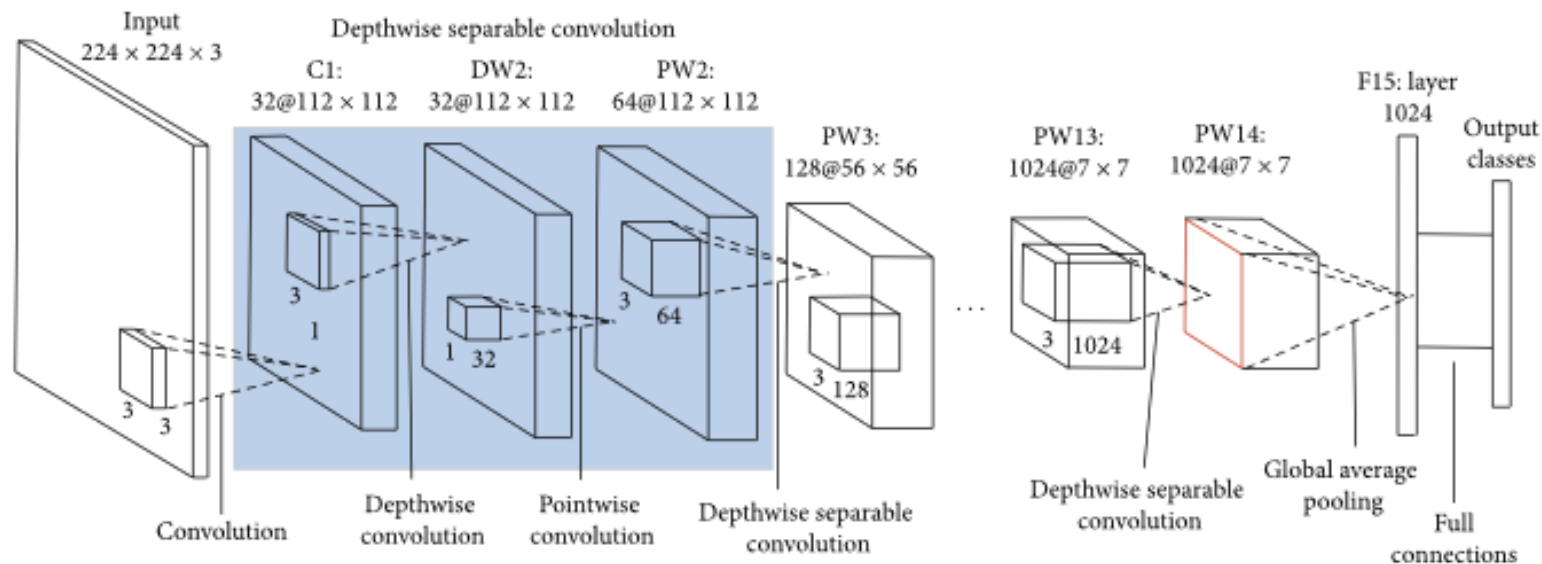
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

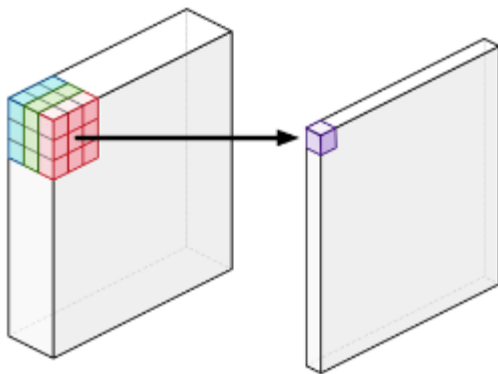
$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

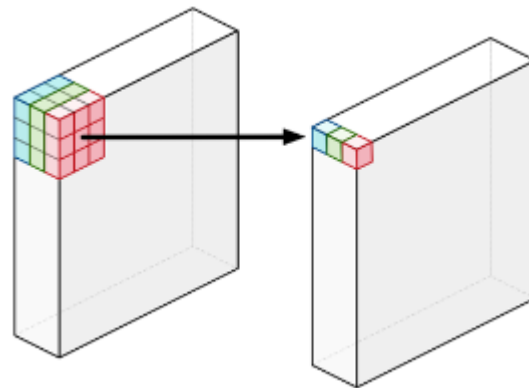
MobileNet (2016)



Depthwise convolution

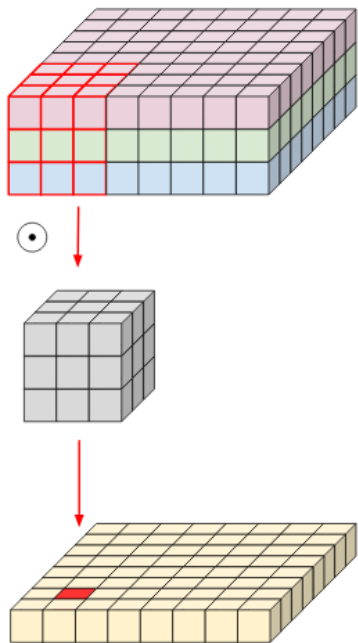


Normal convolution

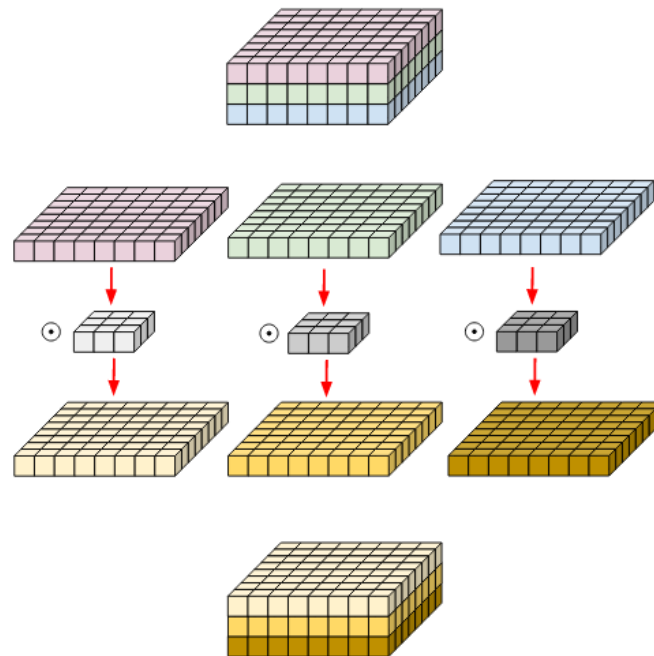


Depthwise convolution

Depthwise convolution

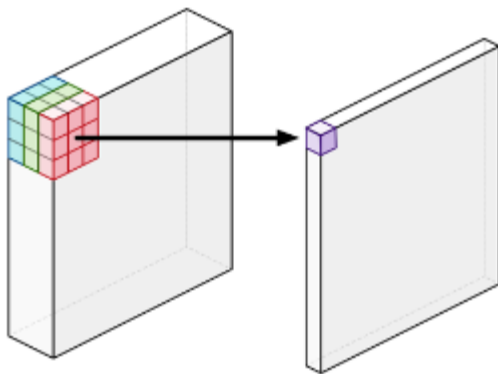


Normal convolution

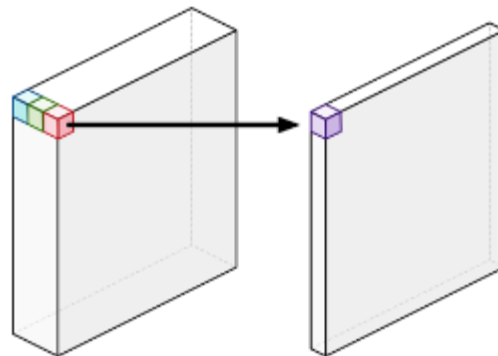


Depthwise convolution

Pointwise convolution

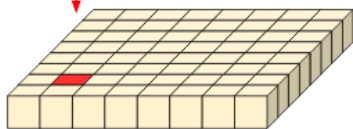
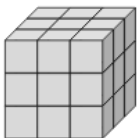
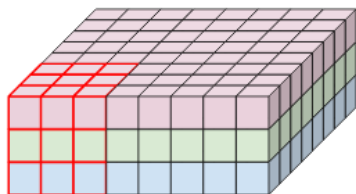


Normal convolution

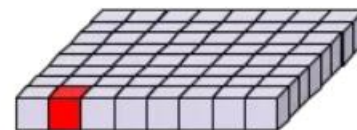
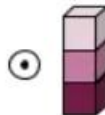
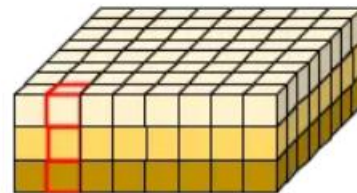


Pointwise convolution

Pointwise convolution

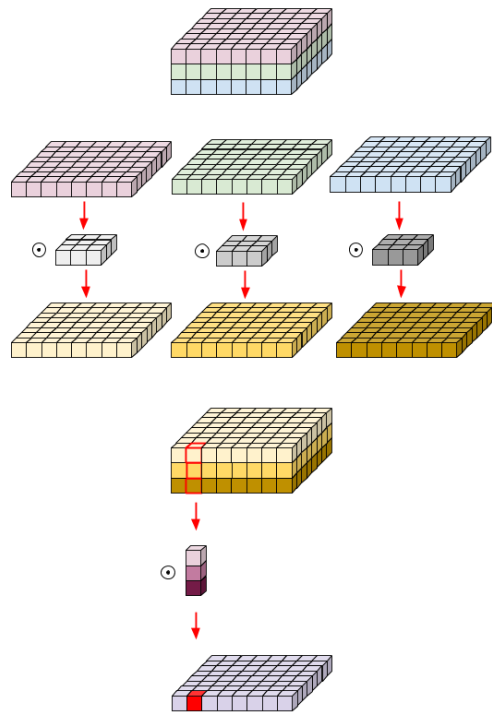


Normal convolution



Pointwise convolution

Depth-wise separable convolution



Параметры и число операций



Вход: $H * W * C_{input}$

Выход: $H * W * C_{output}$

Параметры:

$$K \times K \times C_{in}$$

Операции:

$$H \times W \times K \times K \times C_{in}$$

Количество параметров:

$$K \times K \times C_{in} \times C_{out}$$

Количество операций (примерно):

$$H \times W \times K \times K \times C_{in} \times C_{out}$$

Normal convolution

Deepwise convolution

Параметры:

$$C_{in} \times C_{out}$$

Операции:

$$H \times W \times C_{in} \times C_{out}$$

Pointwise convolution

Параметры и число операций



Вход: $H * W * C_{input}$

Выход: $H * W * C_{output}$

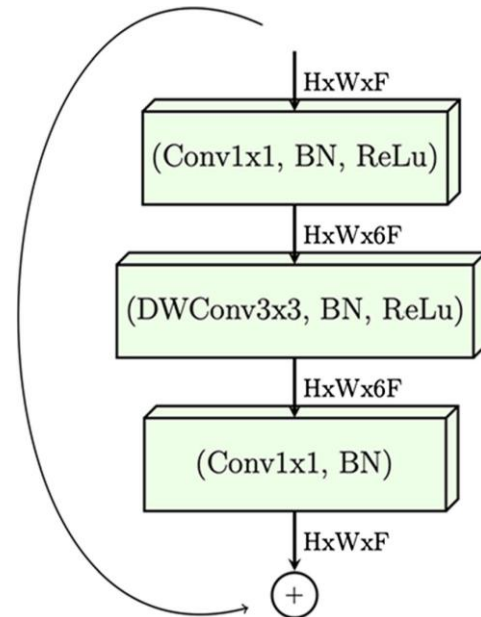
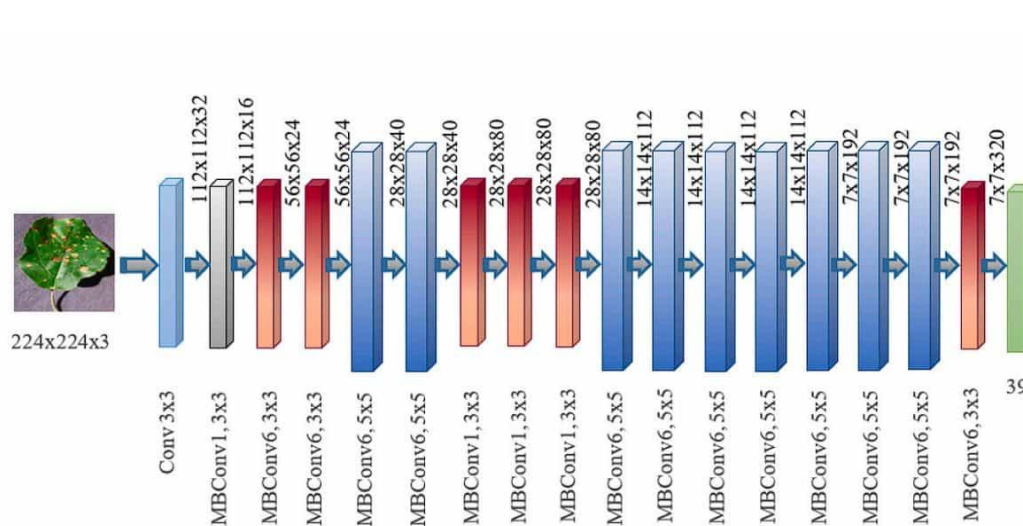
$$\text{Params} = K^2 \cdot C_{in} \cdot C_{out}$$

Normal convolution

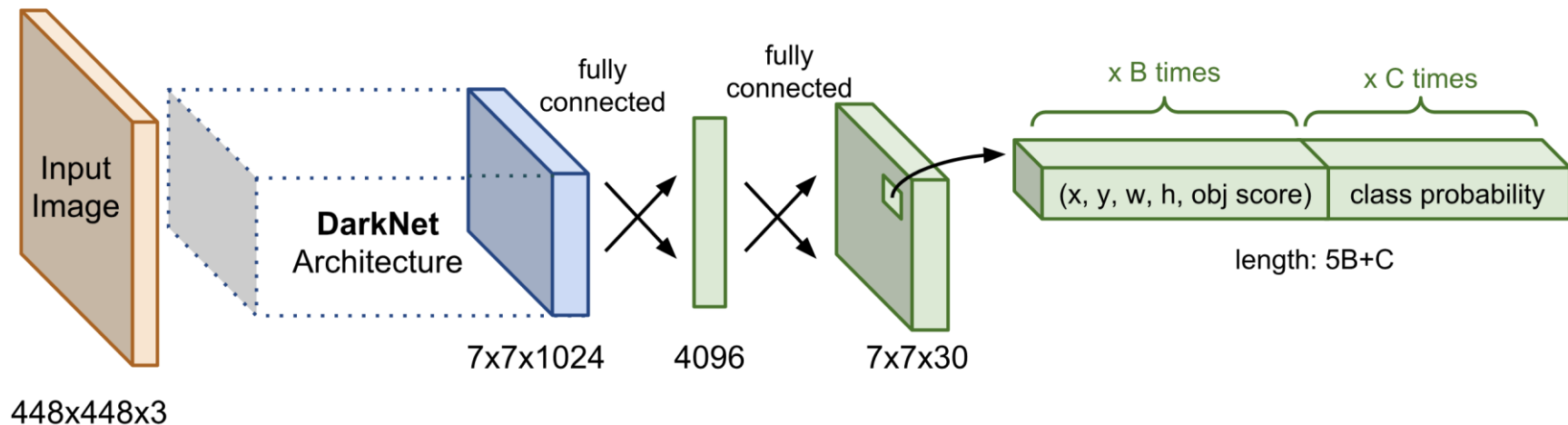
$$\text{Params} = K^2 \cdot C_{in} + C_{in} \cdot C_{out}$$

Depth-wise separable convolution

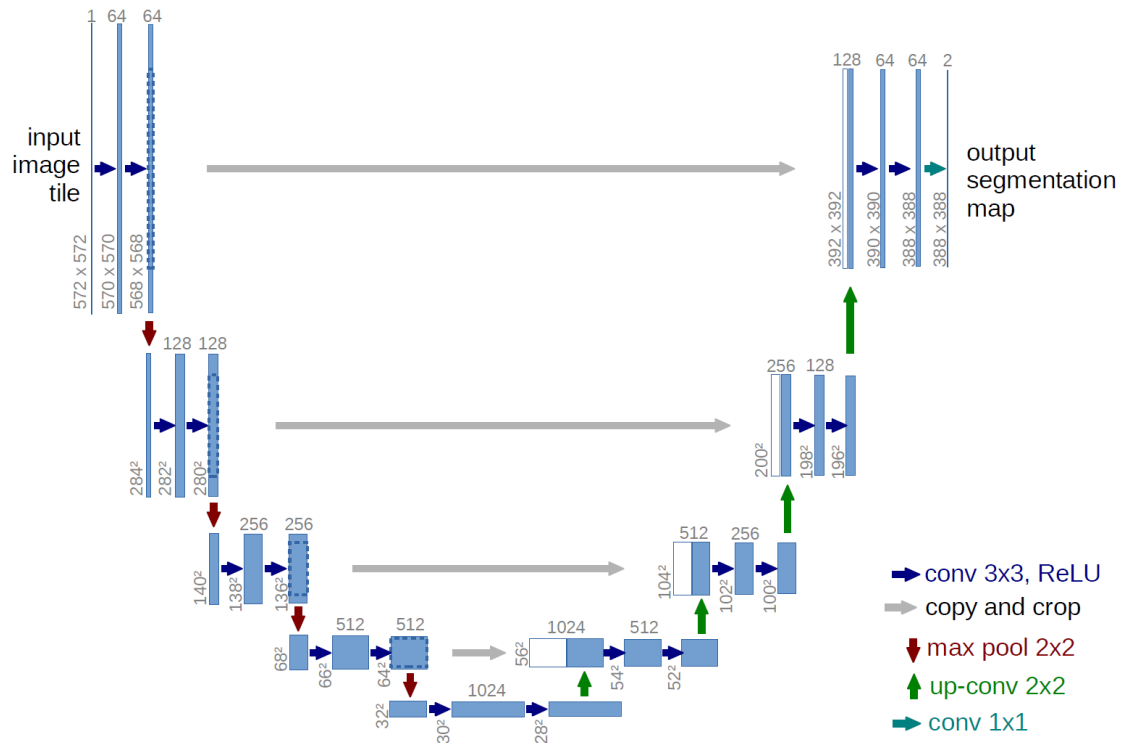
EfficientNet (2019)



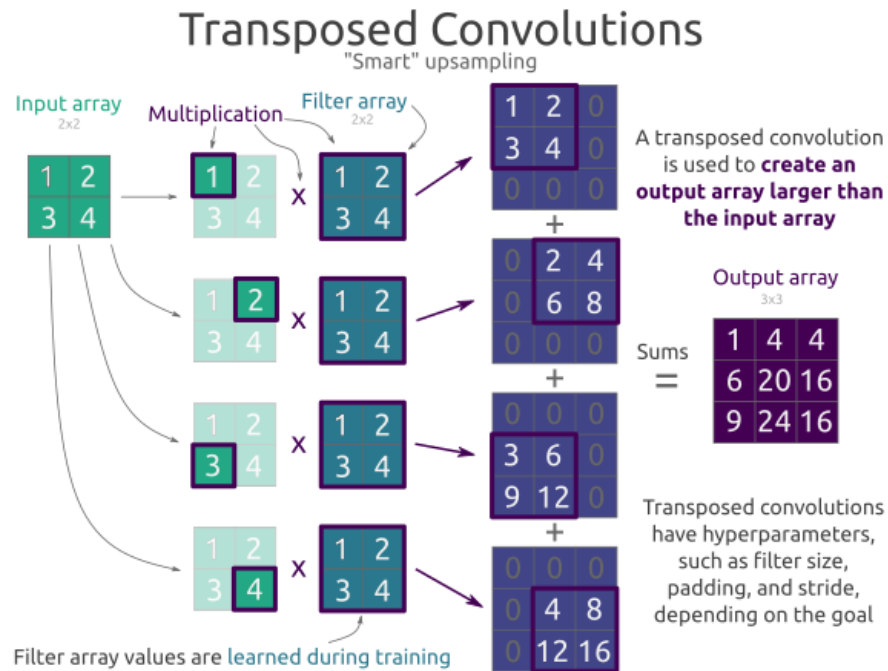
Object detection



Segmentation problem (Unet)



Transposed convolution



**Спасибо
за внимание!**

itMO *re than a*
UNIVERSITY

Ваши контакты