

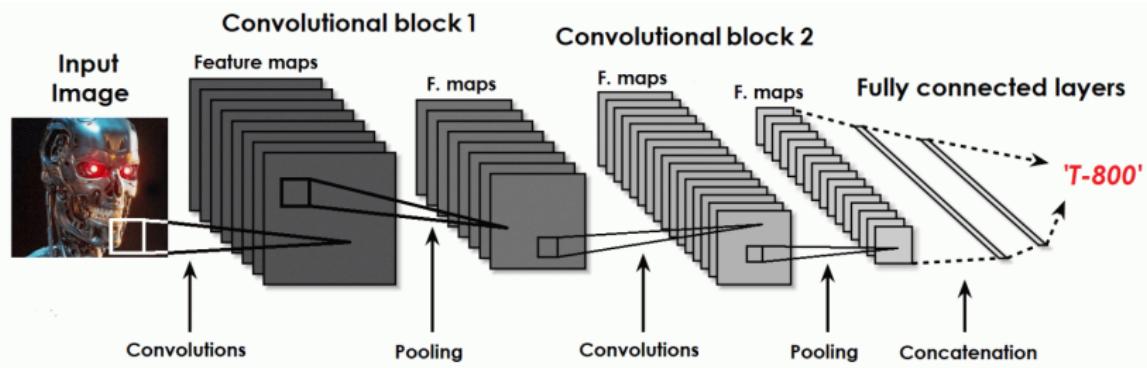
# Свёрточные нейронные сети для обработки изображений: классификация и сегментация

Санкт-Петербургский государственный университет  
Кафедра статистического моделирования

28 октября 2025

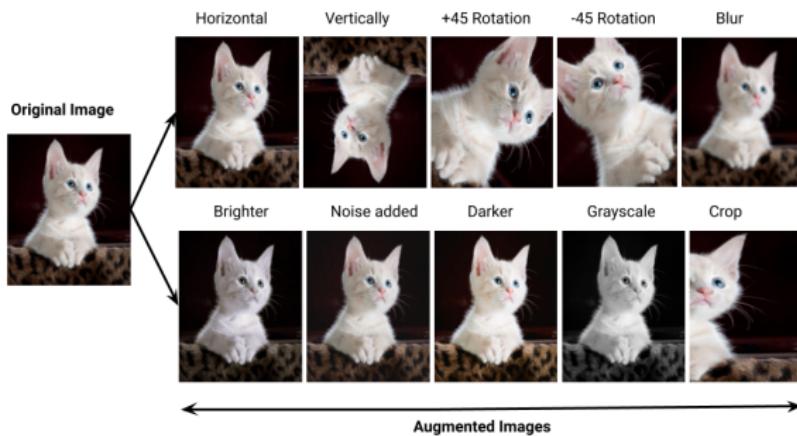
# Свёрточные нейронные сети (CNN)

- Класс нейронных сетей, эффективно работающих с изображениями (и другими объектами, в которых важна пространственная связь);
- Используют свёртки (свёрточные слои) для извлечения признаков;
- Основные применения: классификация, сегментация, детекция объектов.

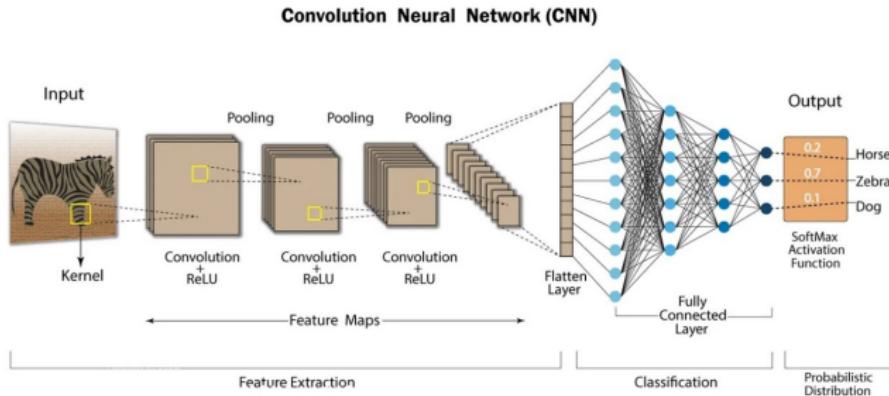


# Проблемы обработки изображений полносвязными сетями

- Изображения содержат пространственную структуру, которую теряют обычные полносвязные сети;
- Если изображение имеет высокое разрешение, то полносвязная сеть содержит слишком много параметров;
- При небольшом изменении изображения (сдвиг, поворот) входной слой обычной сети полностью меняется, хотя суть осталась прежней.



# Основные компоненты CNN

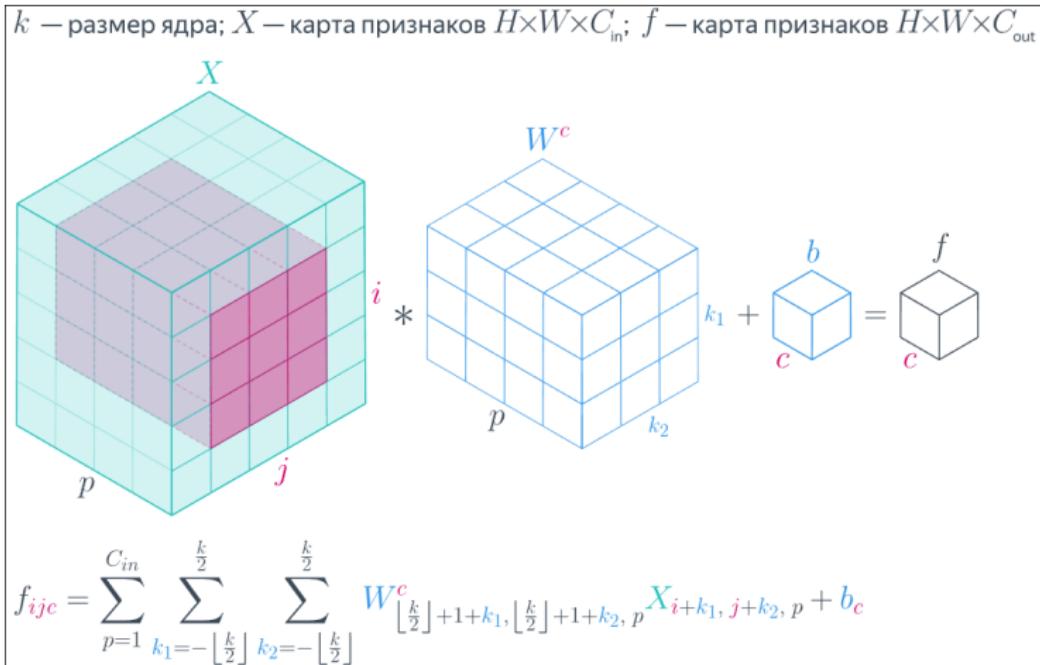


- Свёрточные слои (Conv);
- Пулинг (Pooling, часто MaxPooling или AvgPooling);
- Нелинейности (ReLU, sigmoid, tanh и т.д.);
- Полносвязные слои.

С помощью свёрточных слоёв и пулинга изображение сводится до вектора размерности 1, то есть формируется вход для полносвязной сети.

# Свёрточная операция

- Свертка — это скользящее умножение ядра на участок изображения.
- Позволяет выделить локальные признаки (границы, текстуры).



# Пример свёрток

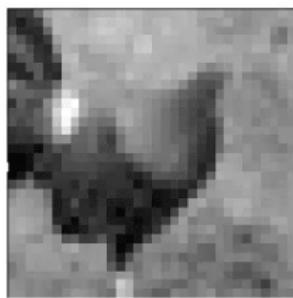
$$B_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Усредняет все пиксели размывая изображение.

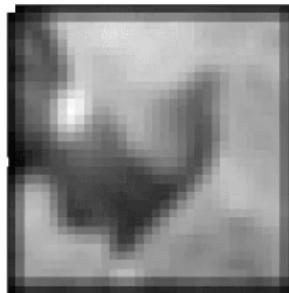
$$B_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Смысл: пиксели из однородных участков изображения слабеют, тогда как контрастные точки, напротив, усиливаются.

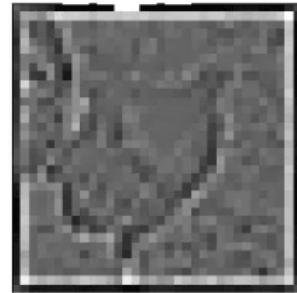
image



image\*B<sub>1</sub>



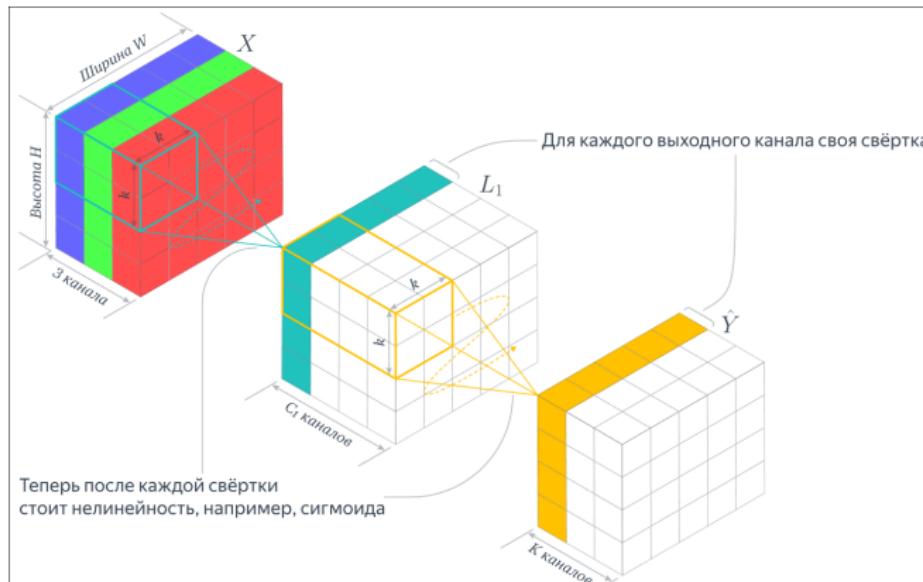
image\*B<sub>2</sub>



Заметим, что свёртки можно воспринимать, как фильтры, применяемые последовательно к изображению.

# Каналы и свёртки

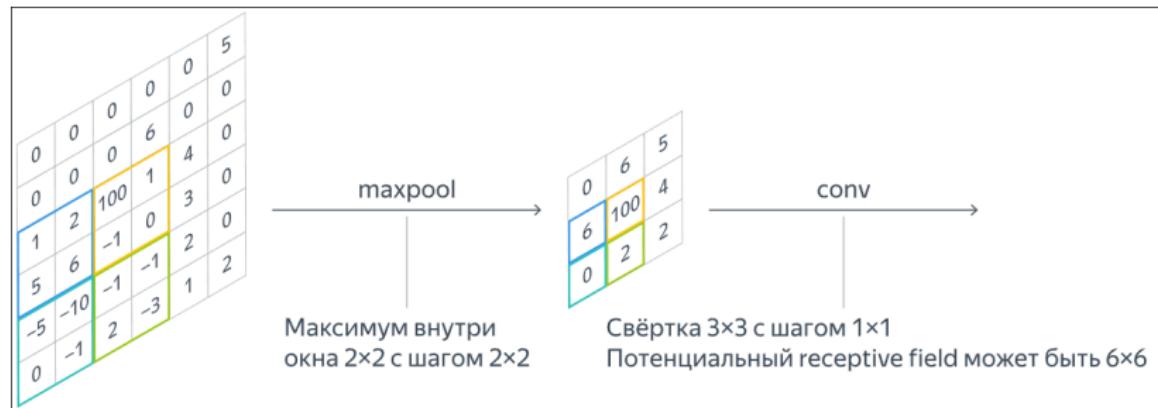
- Изначально в цветном изображении содержится три канала для каждого цвета;
- С каждой свёрткой мы уменьшаем размер изображения, но увеличиваем количество каналов;
- Каждый канал может отвечать за какой-то признак изображения (границы, определённые объекты и т.д.).



# Pooling — уменьшение размерности

Хотим повысить количество каналов в каждой свёртке, чтобы выделять больше признаков. Проблема: слишком много параметров, решение: уменьшить разрешение каждого канала.

- Max Pooling — берёт максимум из области (сохраняет наиболее важные признаки);
- Average Pooling — усреднение значений;
- Снижает вычислительные затраты и переобучение.



- ➊ После нескольких слоёв свёрток, ReLU и Max Pooling: либо тензор  $1 \times 1 \times C$ , либо  $n \times m \times C$ ;
- ➋ В первом случае просто вытягиваем в вектор;
- ➌ Во втором: можем применить Global average pool — пулинг по каналам;
- ➍ Далее применяется обычная полносвязная нейронная сеть.

Для проблем с затуханием используем Residual connection.

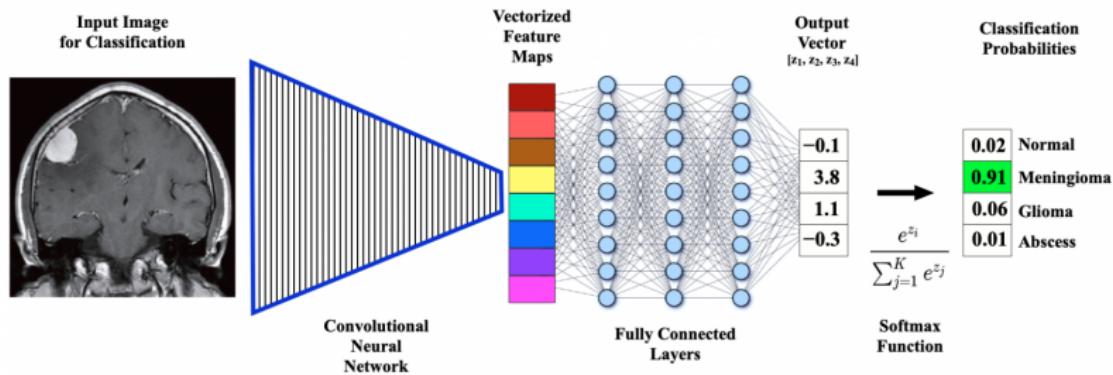
## Регуляризация

- L1/L2, DropOut;
- Аугментация — применение операций сдвига, поворота и т.д.;
- Label smoothing — предсказание сглаженного варианта One hot метки;
- Mixup — выпуклая комбинация двух картинок и их лейблов.

# Задача классификации

Свёрточные нейронные сети хороши тем, что они end-to-end.

- Вход: изображение в виде двухмерных каналов цветов;
- Выход: метка класса (двуклассовая и многоклассовая классификация);
- Пример: распознавание животных, предметов, сцен и т.д.



# Функция потерь и обучение

- Для задачи классификации используется кросс-энтропия (Cross-Entropy Loss):

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c})$$

где  $N$  — размер батча,  $C$  — число классов,  $y_{i,c} \in \{0, 1\}$  — истинная метка (one-hot),  $\hat{y}_{i,c}$  — вероятность, выданная softmax:

$$\hat{y}_{i,c} = \frac{e^{z_{i,c}}}{\sum_{k=1}^C e^{z_{i,k}}}$$

- Используемые оптимизаторы: SGD, Adam
- Backpropagation.

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial \theta}$$

— вычисляется градиент по параметрам сети.

# Backpropagation в свёрточном слое

Обратный проход в свёрточном слое имеет несколько иную форму.

*Градиент по входу:*

$$\frac{\partial \mathcal{L}}{\partial x_{i,j,c}} = \sum_{k=0}^{C_{out}-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{\partial \mathcal{L}}{\partial z_{i-m,j-n,k}} w_{m,n,c,k},$$

— это свёртка ошибки с перевёрнутым фильтром.  $z_{i,j,k}$  — значения следующего слоя.

*Градиент по весам:*

$$\frac{\partial \mathcal{L}}{\partial w_{m,n,c,k}} = \sum_{i,j} x_{i+m,j+n,c} \frac{\partial \mathcal{L}}{\partial z_{i,j,k}}.$$

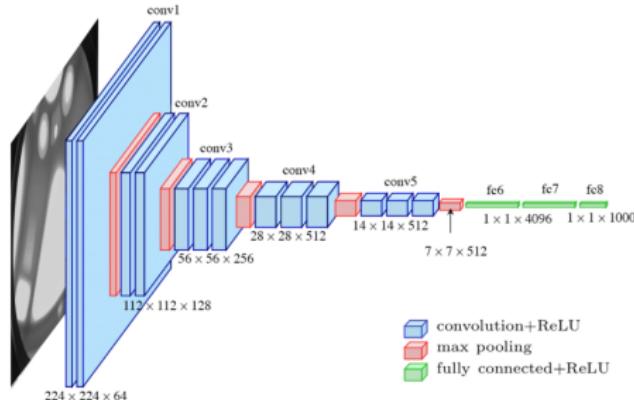
Вес обновляется пропорционально корреляции входа и ошибки

*Градиент по смещению:*

$$\frac{\partial \mathcal{L}}{\partial b_k} = \sum_{i,j} \frac{\partial \mathcal{L}}{\partial z_{i,j,k}}.$$

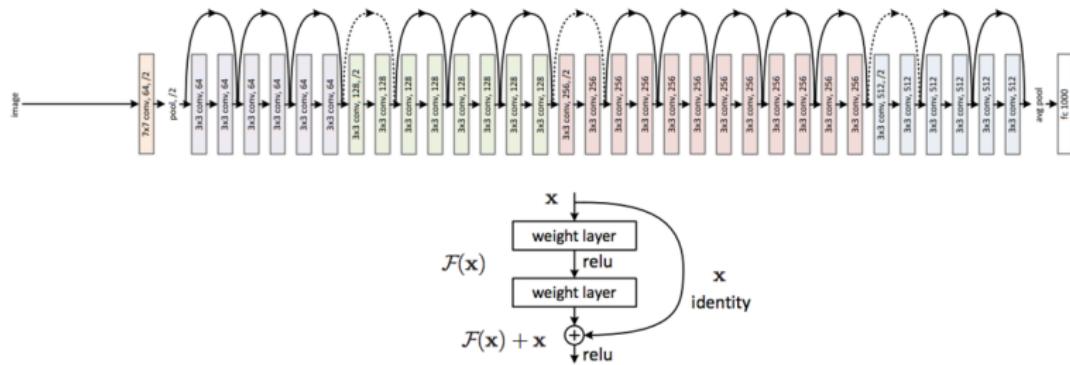
# Популярные архитектуры для классификации

- LeNet (1998) (7 слоёв) — первая свёрточная нейронная сеть, показавшая SOTA результаты на MNIST. Свёрточные слои с ядром  $5 \times 5$ . Активация —  $\tanh$ , вместо max-pool — average;
- AlexNet (2012) (11 слоёв) — первая CNN, взявшая imagenet. ReLU вместо сигмоид, max-pool вместо average. Обучение на двух GPU;
- VGGNet (2014) (19 слоёв) — ввели стандарт свёрток  $3 \times 3$  и последовательное выполнение их с нелинейностями;



ResNet (2015) (152 слоя) — нынешний baseline, ввели residual connections и свёртки  $1 \times 1$ , которые сокращают каналы и добавляют обобщение признаков.

## Residual Networks (ResNet50)

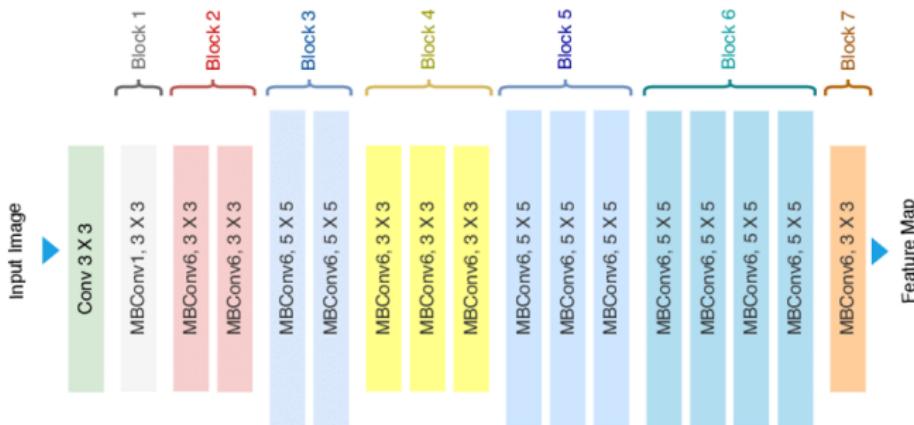


Выше представлен residual block, который не даёт градиентам затухать.

EfficientNet (2019) — серия нейронных сетей (B0-B7, V2), в которых применена идея автоматического поиска архитектуры NAS.

Имеет большую производительность при меньшем количестве параметров по сравнению с ResNet.

Подходит для промышленного использования, когда важны баланс между точностью и ресурсами.



# Что такое сегментация

**Задача:** разделить изображение на осмысленные области (сегменты), соответствующие различным объектам или частям сцены.

- Пиксельная классификация изображения;
- Цель: выделить объекты на уровне пикселей;
- Применения: медицина, автономные автомобили, спутниковые снимки.



# Формальное определение

Пусть у нас есть изображение:

$$\mathbf{I} \in \mathbb{R}^{H \times W \times C},$$

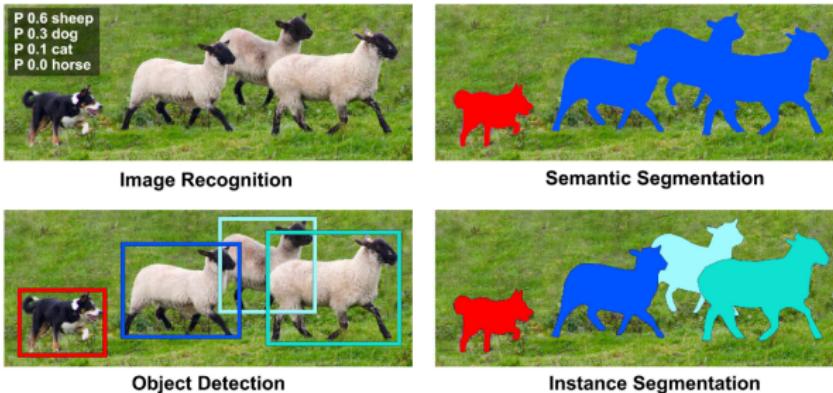
где  $H$  — высота изображения,  $W$  — ширина,  $C$  — число каналов (3 в RGB).

Задача сегментации — найти отображение:

$$f_\theta : \mathbb{R}^{H \times W \times C} \rightarrow \{1, \dots, K\}^{H \times W},$$

где  $K$  — количество классов, а  $f_\theta$  — обучаемая модель, которая выдаёт карту классов для каждого пикселя.

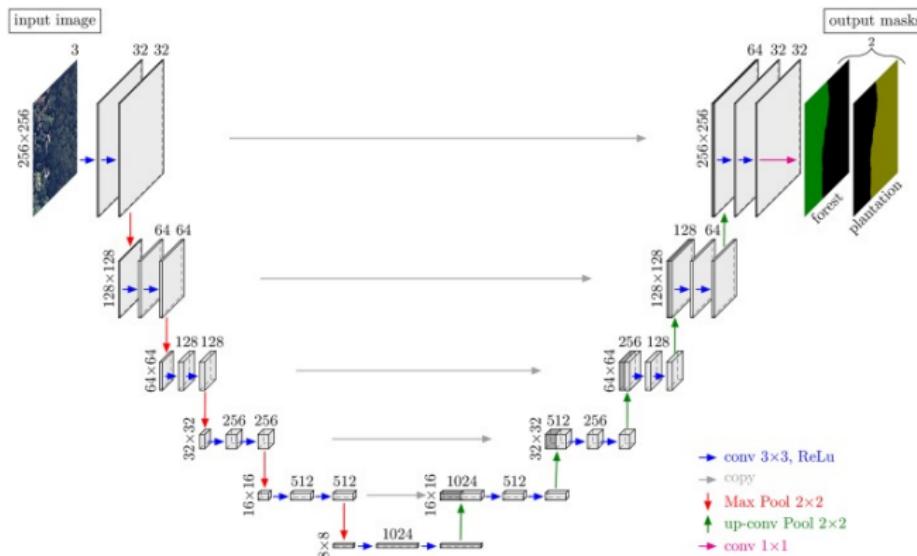
# Виды сегментации



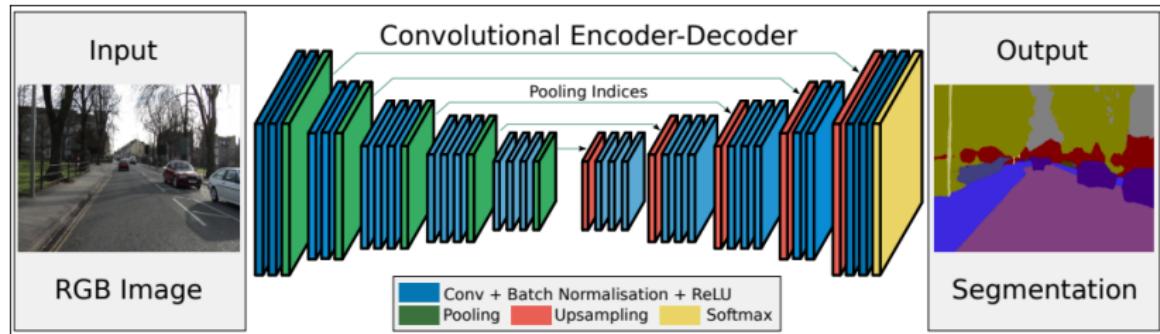
- Семантическая сегментация — классифицирует каждый пиксель, не различая экземпляры (все кошки = один класс);
- Инстанс-сегментация — различает отдельные экземпляры объектов одного класса (каждая кошка — отдельный объект);
- Детекция — выделение объектов прямоугольниками с классификацией;
- Паноптическая сегментация — объединяет семантическую и инстанс-сегментацию (сцена со всеми типами объектов).

# U-Net (2015)

- Симметричная encoder-decoder архитектура, decoder использует транспонированные свёртки (upsampling);
- Skip-соединения между слоями (копирует признаки с энкодера и объединяет с декодером);
- Широко применяется в медицине.

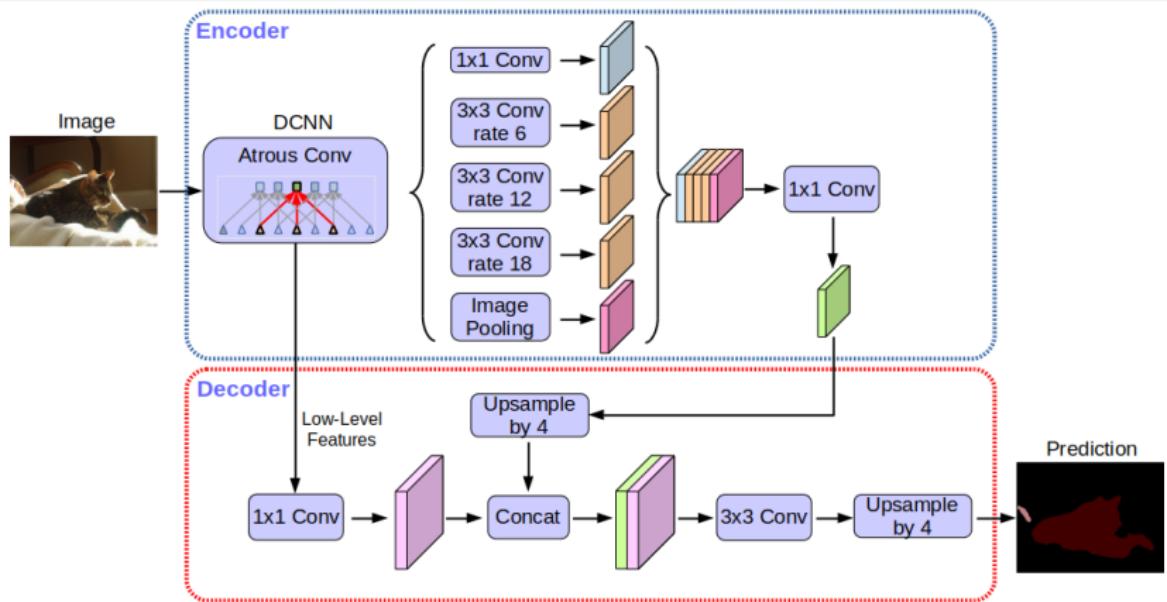


# SegNet (2016)



- Использует индексы max pooling для восстановления формы;
- Нет skip-соединений, меньше параметров;
- Подходит для сегментации сцен в реальном времени.

# DeepLab (v1–v3+) (2017–2018)



- Использует dilated convolutions (увеличение receptive field без потери разрешения);
- Применяет CRF (разновидность марковского случайного поля) для уточнения границ объектов;
- Высокая точность на сложных сценах.

Модель детекции объектов на изображении в реальном времени.

- Изображение делится на сетку  $S \times S$ ;
- В каждой ячейке предсказывается  $B$  ограничивающих прямоугольников, вероятность объекта  $P(\text{obj})$  и распределение вероятностей классов  $P(\text{class}|\text{obj})$ ;
- Используется составной loss:

$$\begin{aligned}\mathcal{L} = & \lambda_{\text{coord}} \sum (x - \hat{x})^2 + (y - \hat{y})^2 + \\ & + \lambda_{\text{coord}} \sum (\sqrt{w} - \sqrt{\hat{w}})^2 + (\sqrt{h} - \sqrt{\hat{h}})^2 +; \\ & + \sum (p_{\text{obj}} - \hat{p}_{\text{obj}})^2 + \sum (p_{\text{cls}} - \hat{p}_{\text{cls}})^2\end{aligned}$$

- Очень высокая скорость и компактность.

- 1 Стандартные оценки для классификации: accuracy, precision, recall, f1 и т.д. Здесь TP — пиксели, правильно отнесённые к классу объекта, TN — пиксели фона, правильно распознаные, FP — пиксели фона, ошибочно отнесённые к объекту, FN — пиксели объекта, пропущенные моделью.
- 2 **Intersection over Union (IoU)** — пересечение предсказанных и истинных областей:

$$\text{IoU} = \frac{TP}{TP + FP + FN}, \quad \text{mIoU} = \frac{1}{K} \sum_{k=1}^K \text{IoU}_k$$

- 3 **Dice Coefficient** используется для несбалансированных данных:

$$\text{Dice} = \frac{2TP}{2TP + FP + FN}, \quad \text{Dice Loss} = 1 - \text{Dice}$$

- Обучение модели с нуля не всегда бывает оправданным: на это требуется много времени и ресурсов;
- Чтобы избежать этого следует использовать предобученные сети, которые обучались на большом объёме общих данных и уже имеют некоторое понятие „о мире“;
- Если мы имеем дело со специфичной задачей, то предобученную сеть можно „дообучить“ (fine-tuning) на нашей задаче, используя наш датасет и применяя оптимизацию параметров на нём;
- Дообучение претрейн модели происходит быстрее, так как модель уже выучила общие зависимости и осталось только подогнать параметры под конкретную задачу.