Свёрточные нейронные сети

Степанов Иван

Московский физико-технический институт

Москва, 11 марта 2025 г.

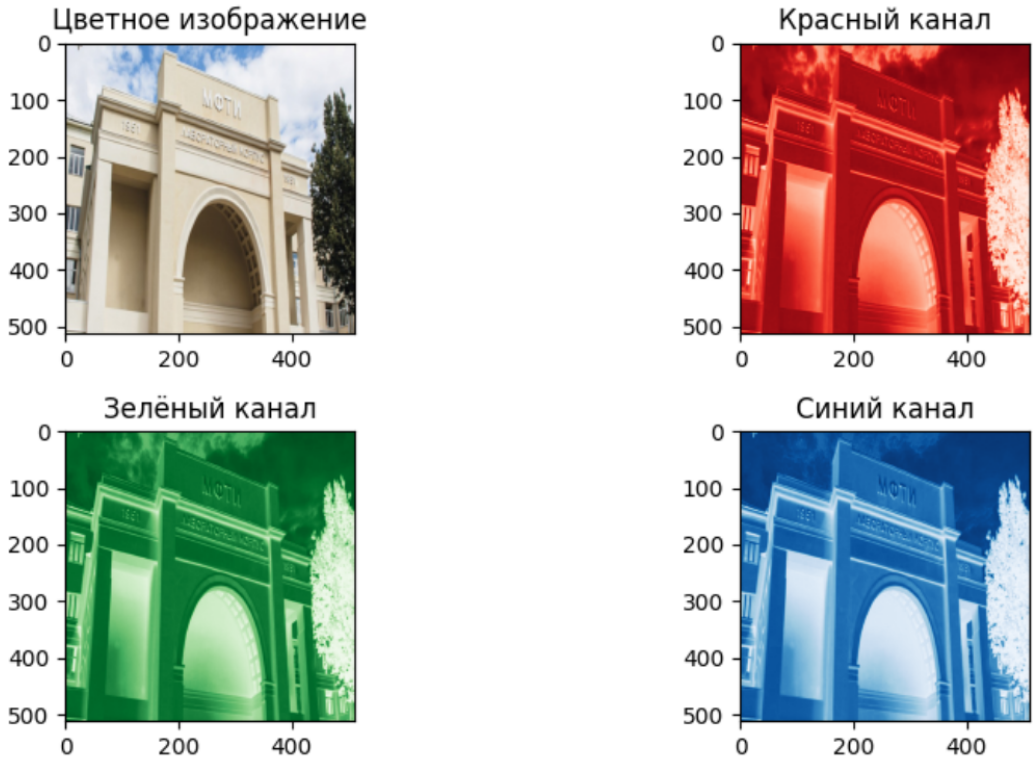
План лекции

- Структура изображения
- Свёртка
- Pooling
- LeNet
- Аугментация

В каком виде хранятся изображения?

Тензор (массив):

- Черно-белое изображение - представлено в виде 2D-матрицы.
- Цветное изображение - 3D-тензор.



Задача1

Предположим мы хотим решить задачу классификации картинок с помощью полносвязной нейронной сети.

- Как будем решать данную задачу?
- В чём недостатке данного метода?

Свёртка (Convolution) — ключевая операция в сверточных нейронных сетях (CNN), которая позволяет **извлекать** локальные признаки из изображений (края, текстуры, объекты).

свёртка

- **ядро:** небольшая матрица, которая скользит по изображению.
- **операция свёртки**
- **сдвиг (stride):** шаг, с которым фильтр перемещается по изображению.

формула для размера выхода

$$Output = \frac{Input - kernel_{size}}{Stride} + 1$$

Входное изображение

$$I = \begin{bmatrix} 255 & 255 & 100 & 0 \\ 255 & 100 & 0 & 0 \\ 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Ядро

$$K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$ **stride=1**

Результат свёртки

$$\begin{bmatrix} 510 & 355 \\ 355 & 100 \end{bmatrix}$$

Padding — добавление дополнительных пикселей (обычно нулевых) вокруг границ входного изображения перед применением свёртки.

Входное изображение

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Padding

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 0 \\ 0 & 5 & 6 & 7 & 8 & 0 \\ 0 & 9 & 10 & 11 & 12 & 0 \\ 0 & 13 & 14 & 15 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

формула для размера выхода

$$Output = \frac{Input - kernel_{size} + 2padding}{Stride} + 1$$

Применение различных свёрток

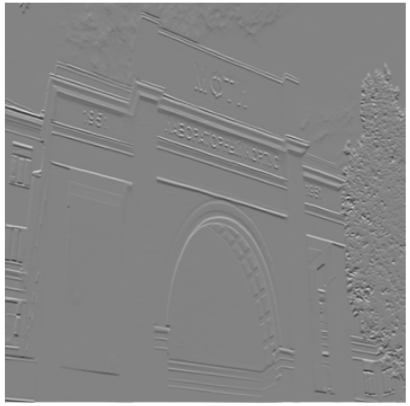
Original



Vertical Edges



Horizontal Edges



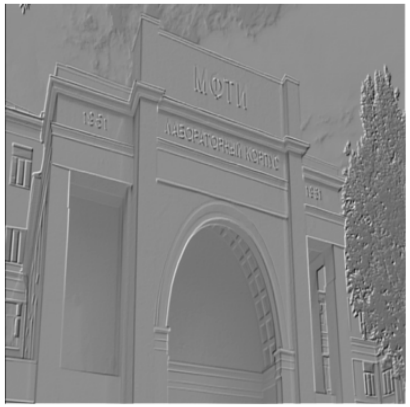
Blur (3x3)



Sharpen



Emboss



Применение различных свёрток

Вертикальные границы (Vertical Edges)

$$K_{\text{vertical}} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Горизонтальные границы (Horizontal Edges)

$$K_{\text{horizontal}} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Размытие (Blur)

$$K_{\text{blur}} = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Повышение резкости (Sharpen

$$K_{\text{sharpen}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Тиснение (Emboss)

$$K_{\text{emboss}} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

Pooling — операция в сверточных нейронных сетях (CNN), которая снижает пространственный размер карт признаков.

Максимальный пулинг

Берётся **максимальное значение** из локальной области (например, 2×2).

Средний пулинг

Берётся **средние значение** из локальной области (например, 2×2).

Формула для выходного размера

$$\text{Выходной размер} = \left\lfloor \frac{input - pooling_{size}}{stride} \right\rfloor + 1$$

Входная карта признаков (4×4)

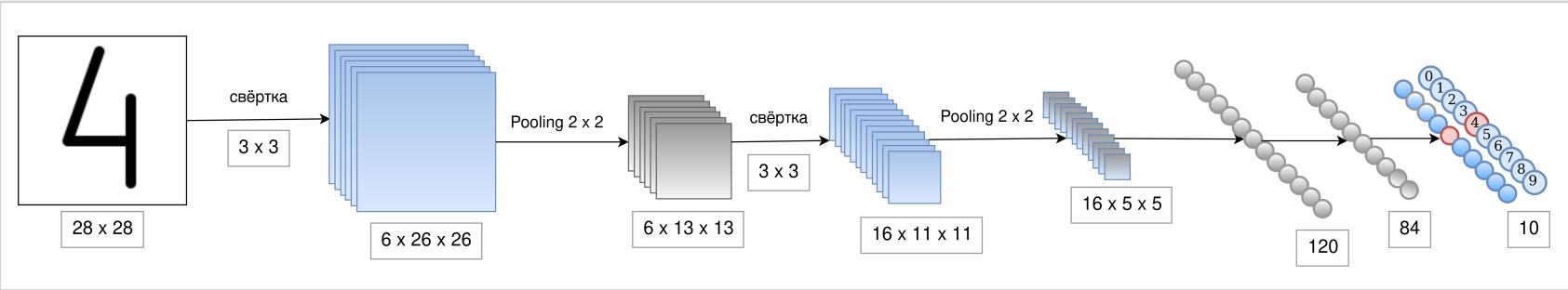
$$\begin{bmatrix} 1 & 3 & 0 & 5 \\ 4 & 2 & 1 & 0 \\ 3 & 7 & 2 & 1 \\ 6 & 5 & 4 & 3 \end{bmatrix}$$

$$\begin{bmatrix} \max(1,3,4,2) & \max(0,5,1,0) \\ \max(3,7,6,5) & \max(2,1,4,3) \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 5 \\ 7 & 4 \end{bmatrix}$$

Входная карта признаков (4×4)

$$\begin{bmatrix} 1 & 3 & 0 & 5 \\ 4 & 2 & 1 & 0 \\ 3 & 7 & 2 & 1 \\ 6 & 5 & 4 & 3 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1+3+4+2}{4} & \frac{0+5+1+0}{4} \\ \frac{3+7+6+5}{4} & \frac{2+1+4+3}{4} \end{bmatrix} \rightarrow \begin{bmatrix} 2.5 & 1.5 \\ 5.25 & 2.5 \end{bmatrix}$$



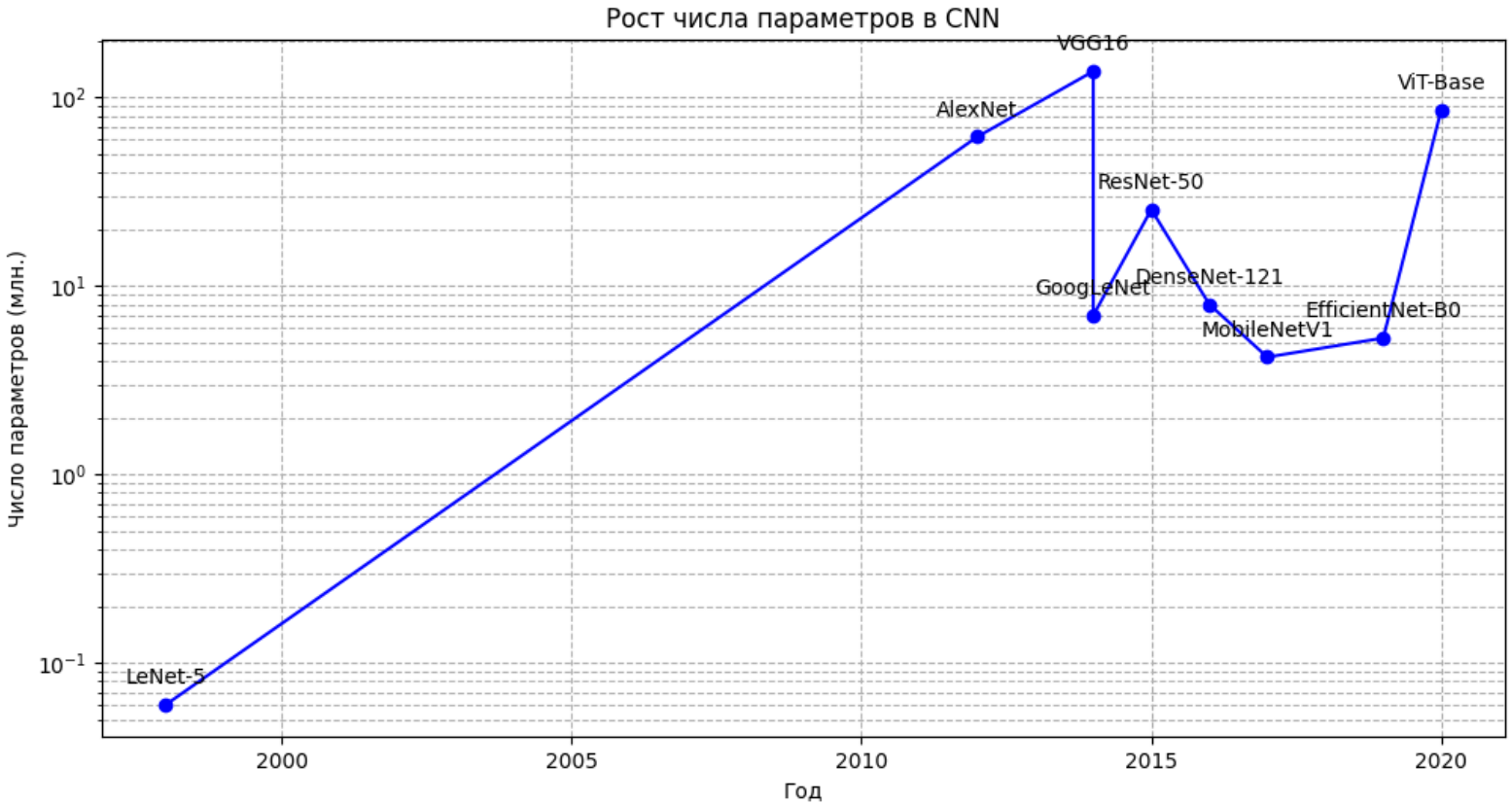
Задача2

Какое количество обучаемых параметров модели?

Год	Архитектура	Параметры	Особенности
1998	LeNet-5	60 тыс.	Первая CNN для MNIST,5×5 свёртки, avg-pooling
2012	AlexNet	62 млн.	Победа на ImageNet,ReLU, dropout, 5 свёрточных слоёв
2014	VGG16	138 млн.	Блоки 3×3 свёрток,глубина 16 слоёв
2014	GoogLeNet	7 млн.	Inception-модули
2015	ResNet-50	25.5 млн.	Остаточные связи,решение проблемы градиентов
2016	DenseNet-121	8 млн.	Прямые связи между слоями
2017	MobileNetV1	4.2 млн.	Depthwise separable свёртки
2019	EfficientNet-B0	5.3 млн.	Баланс глубины/ширины/разрешения
2020	Vision Transformer	86 млн.	Трансформер для CV,патчи как токены

Эволюция CNN-архитектур

Ключевые архитектуры



Аугментация данных (data augmentation) — метод увеличения разнообразия обучающей выборки за счёт искусственного создания новых примеров из существующих. Позволяет:

- **Снизить переобучение** при малом объёме данных.
- **Улучшить обобщающую способность** модели.
- **Сделать сеть инвариантной** к преобразованиям (например, поворотам, масштабированию).

Поворот (Rotation)

Изображение поворачивается на угол θ с помощью матрицы поворота:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Новые координаты пикселя (x', y'):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R(\theta) \cdot \begin{bmatrix} x - c_x \\ y - c_y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix},$$

где (c_x, c_y) – центр изображения.

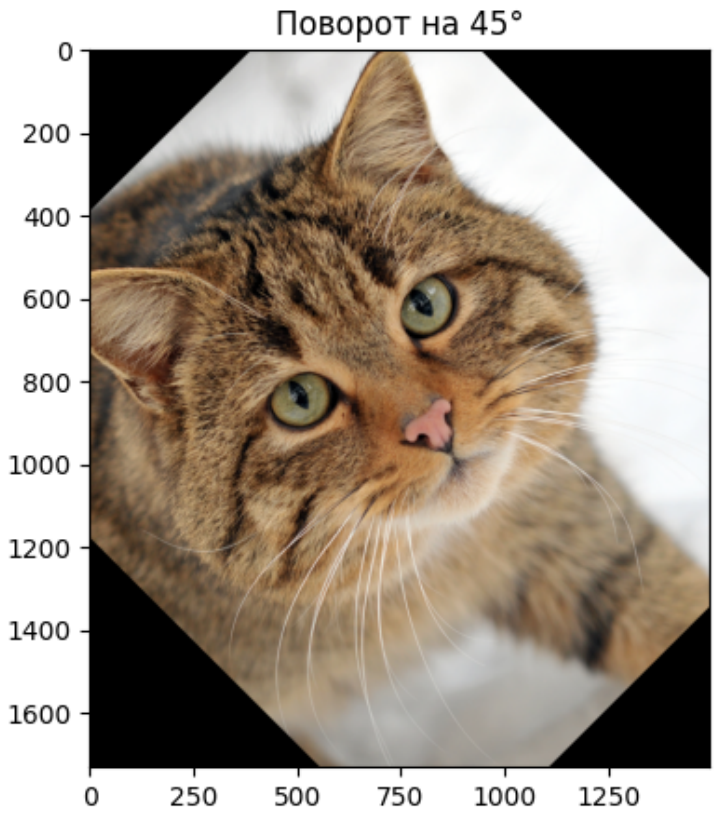
Сдвиг (Translation)

Сдвиг на вектор $(\Delta x, \Delta y)$:

$$x' = x + \Delta x, \quad y' = y + \Delta y.$$

Аугментация (Геометрические преобразования)





Изменение яркости

Добавление константы b к значениям пикселей:

$$I'(x, y) = \text{clip}(I(x, y) + b, 0, 255).$$

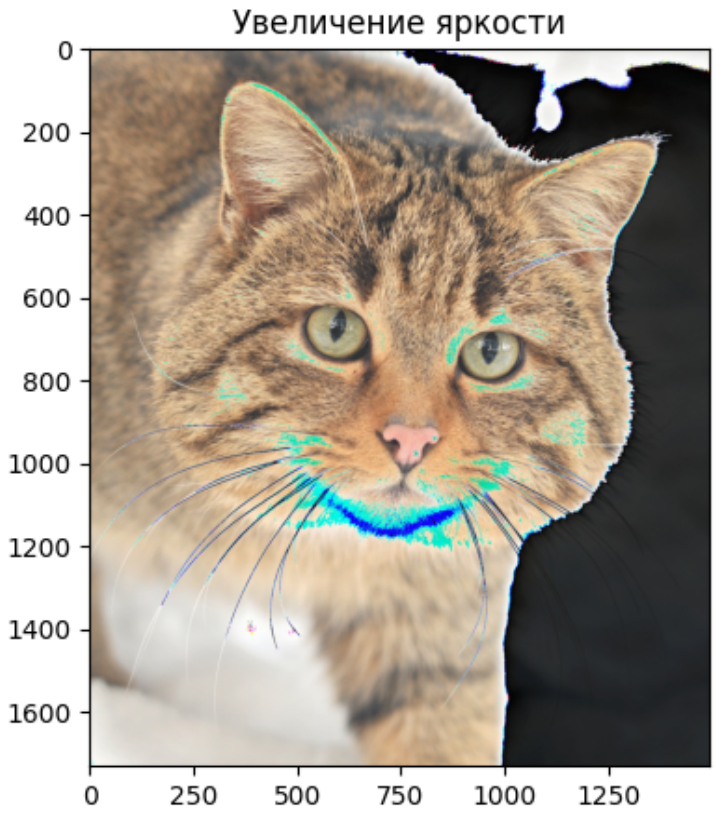
Изменение контраста

Добавление константы b к значениям пикселей:

$$I'(x, y) = \text{clip}(c \cdot (I(x, y) - \mu) + \mu, 0, 255).$$

Аугментация (Цветовые преобразования)





Добавление шума

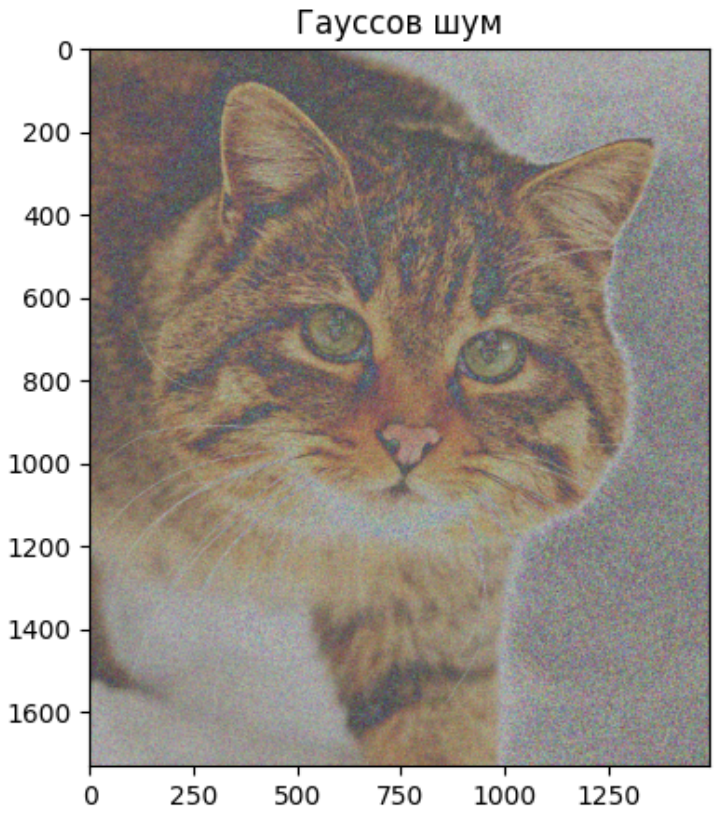
$$I'(x,y) = \text{clip}(I(x,y) + \mathcal{N}(0,\sigma^2), 0, 255),$$

Случайная замена пикселей на 0 или 255 с вероятностью p:

$$I'(x,y) = \begin{cases} 0, & \text{с вероятностью } p/2, \\ 255, & \text{с вероятностью } p/2, \\ I(x,y), & \text{иначе.} \end{cases}$$

Аугментация (добавление шума)





Линейная комбинация двух изображений I_1 и I_2 с коэффициентом $\lambda \in [0, 1]$:

$$I' = \lambda I_1 + (1 - \lambda) I_2, \quad y' = \lambda y_1 + (1 - \lambda) y_2,$$

где y_1, y_2 – метки классов.

Аугментация (добавление шума)



Вопросы?