



Национальный исследовательский  
Нижегородский государственный университет им. Н.И. Лобачевского  
Институт информационных технологий, математики и механики

# Введение в компьютерное зрение

Кустикова В.Д.,  
к.т.н., доцент каф. МОСТ ИИТММ  
ННГУ им. Н.И. Лобачевского

# Содержание (1)

---

- ❑ Общая информация
  - Что такое «зрение»?
  - Что такое «компьютерное зрение»?
  - В чем сложность задач компьютерного зрения?
  - Цифровые изображения
    - Бинарные изображения
    - Полутоновые изображения
    - Мультиспектральные изображения
  - Операции над бинарными изображениями
  - Операции с полутоновыми изображениями
  - Цветные изображения



# Содержание (2)

---

- ❑ Результаты решения классических задач компьютерного зрения с использованием глубокого обучения
  - Что такое «глубокое обучение»?
  - Классификация изображений
  - Детектирование объектов
  - Семантическая сегментация изображений
- ❑ Общая схема решения задач компьютерного зрения с помощью глубокого обучения



# ОБЩАЯ ИНФОРМАЦИЯ



# Что такое «зрение»?

- ❑ **Зрение** – одно из внешних чувств человека и животного, органом которого является глаз; способность видеть (*Толковый словарь Ожегова*)
- ❑ **Зрение** – восприятие организмом объектов внешнего мира посредством улавливания отражаемого или излучаемого объектами света (*Большой Энциклопедический словарь*)
- ❑ **Зрение** (visio, visus) – восприятие света органом зрения и зрительным анализатором, благодаря чему организм получает информацию об объектах (*Большой медицинский словарь*)

❑ ...



# Что такое «компьютерное зрение»?

---

- ❑ **Компьютерное зрение** – теория и технология создания машин, которые могут извлекать информацию из изображений и анализировать ее



# В чем сложность задач компьютерного зрения?

- ❑ Разное восприятие визуальной информации

**Человек видит картинку**



**Машина видит числа**

```
array([[173, 173, 173, ..., 166, 166, 166],  
       [173, 173, 173, ..., 166, 166, 166],  
       [173, 173, 173, ..., 166, 166, 166],  
       ...,  
       [168, 167, 166, ..., 173, 173, 173],  
       [168, 167, 166, ..., 173, 173, 173],  
       [169, 168, 167, ..., 173, 173, 173]], dtype=uint8)
```

- ❑ Неоднозначность интерпретации визуальной информации\*

\* Источник изображения [<https://www.adme.ru/svoboda-narodnoe-tvorchestvo/22-strannyh-proisshestiya-kotorye-sluchajno-popali-na-fotografii-1592515>]



**Камень летит в воздухе?  
Или это не небо?**



# Цифровые изображения

- ❑ **Цифровое изображение** – это двумерный массив  $I[r, c]$  элементов (пикселей), которые представляют собой одно или нескольких дискретных значений
  - **Бинарное изображение** – это цифровое изображение, пиксели которого принимают значения 0 или 1
  - **Полутонное (серое, монохромное, черно-белое) изображение** – это цифровое изображение, у которого каждому пикселю соответствует одно значение интенсивности (яркости)
  - **Мультиспектральное изображение** – это цифровое изображение, у которого каждому пикселю соответствует вектор значений (у цветных изображений размерность вектора равна 3)



\* Учебный курс «Разработка мультимедийных приложений с использованием библиотек OpenCV и IPP» [<http://www.hpcc.unn.ru/?doc=602>]. Лекция 1. Обзор задач и методов компьютерного зрения.



# Операции над бинарными изображениями.

## Морфологические операции

- ❑ **Эрозия.** Значение элемента заменяется на минимум из окрестности

$$dst[x, y] = \min_{(i,j), i,j \in \{-1,0,1\}} src[x + i, y + j]$$



- ❑ **Дилатация.** Значение элемента заменяется на максимум из окрестности

$$dst[x, y] = \max_{(i,j), i,j \in \{-1,0,1\}} src[x + i, y + j]$$



- ❑ **Размыкание.** Последовательное применение эрозии и дилатации

$$dst = open(src) = dilate(erode(src))$$



- ❑ **Замыкание.** Последовательное применение дилатации и эрозии

$$dst = close(src) = erode(dilate(src))$$



# Операции над бинарными изображениями.

## Выделение связных компонент

- ❑ **Выделение связных компонент** – присвоение уникального идентификатора каждой области (объекту) изображения
- ❑ **Пиксели**  $(x_1, y_1)$  и  $(x_2, y_2)$  называются **связными**, если они являются соседними (расстояние равно 1) по некоторой выбранной метрике
- ❑ **Связная компонента изображения** – это связное множество пикселей согласно выбранной метрике
- ❑ **Метрики:**

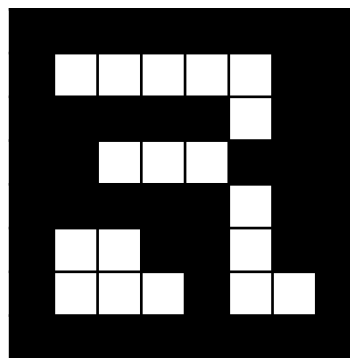
4-ех связная

	1	
2	*	3
	4	

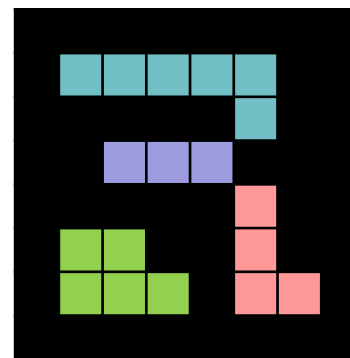
8-ми связная

1	2	3
4	*	5
6	7	8

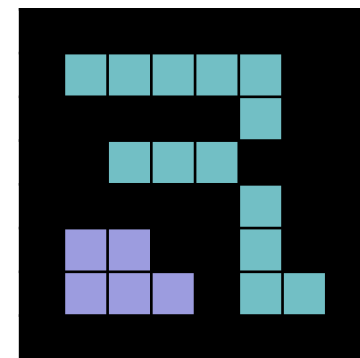
*Пример*



Исходное  
изображение



4-ех связные  
компоненты



8-ми связные  
компоненты

# Операции с полутоновыми изображениями.

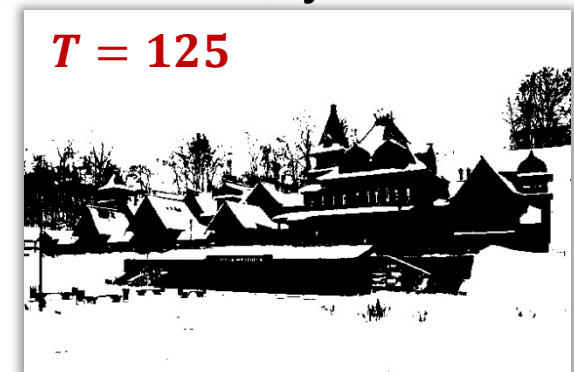
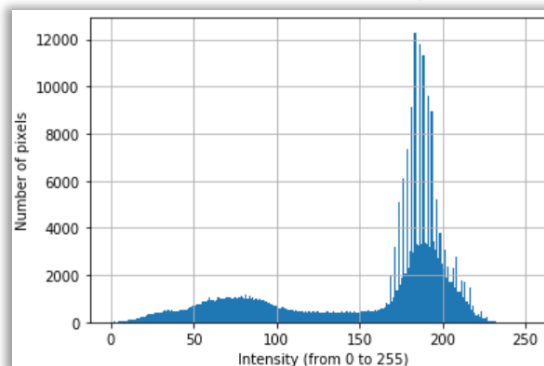
## Бинаризация полутоновых изображений (1)

### □ Пороговая бинаризация

$$I'[x, y] = \begin{cases} 1, & I[x, y] \geq T \\ 0, & \text{иначе} \end{cases}$$

- Для выбора порога строится гистограмма интенсивностей  $H[i] = |\{(x, y) | I[x, y] = i\}|$  (бин – число пикселей с заданной интенсивностью)

- При наличии хорошо разделимых объектов и фона гистограмма будет иметь два пика
- Порог – интенсивность, отвечающая впадине между пиками



# Операции с полутоновыми изображениями.

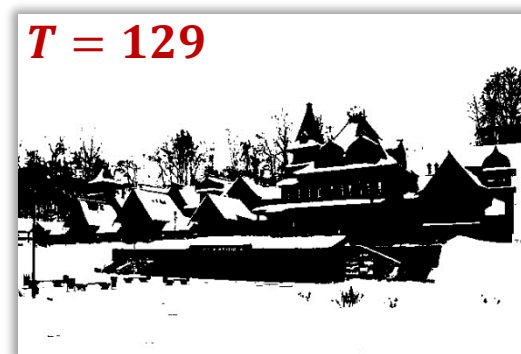
## Бинаризация полутоновых изображений (2.1)

### □ Пороговая бинаризация

$$I'[x, y] = \begin{cases} 1, & I[x, y] \geq T \\ 0, & \text{иначе} \end{cases}$$

- Для выбора порога строится гистограмма интенсивностей  $H[i] = |\{(x, y) | I[x, y] = i\}|$  (бин – число пикселей с заданной интенсивностью)

– Метод Отсу\* (Otsu) – метод подбора порога, основанный на минимизации внутриклассовой дисперсии



\* Otsu N. A threshold selection method from gray-level histograms // IEEE Transactions on Systems, Man, and Cybernetics. – 1979. – Vol. 9. – P. 62-66.

# Операции с полутоновыми изображениями.

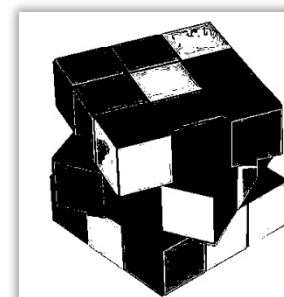
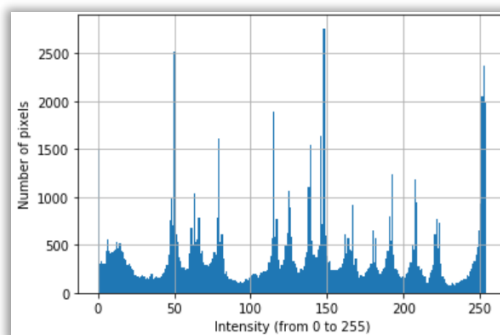
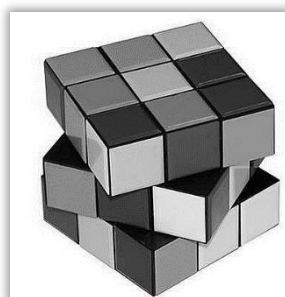
## Бинаризация полутоновых изображений (2.2)

### □ Пороговая бинаризация

$$I'[x, y] = \begin{cases} 1, & I[x, y] \geq T \\ 0, & \text{иначе} \end{cases}$$

- Для выбора порога строится гистограмма интенсивностей  $H[i] = |\{(x, y) | I[x, y] = i\}|$  (бин – число пикселей с заданной интенсивностью)

– Метод Отсу\* (Otsu) – метод подбора порога, основанный на минимизации внутриклассовой дисперсии



**$T = 160$**

\* Otsu N. A threshold selection method from gray-level histograms // IEEE Transactions on Systems, Man, and Cybernetics. – 1979. – Vol. 9. – P. 62-66.

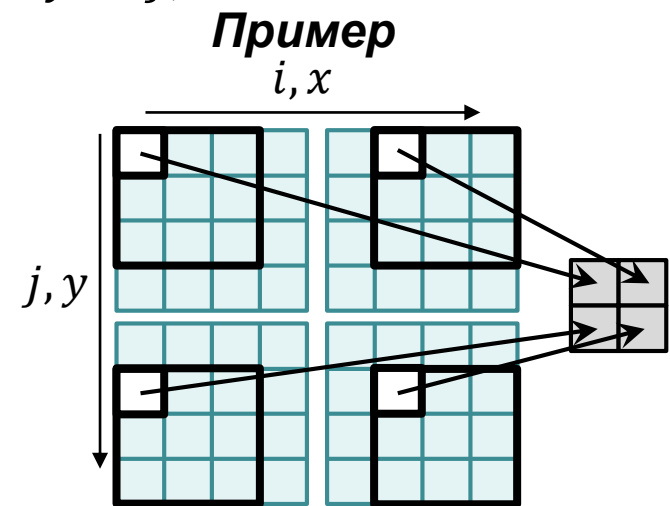
# Операции с полутоновыми изображениями.

## Линейная фильтрация (1)

- ❑  $I$  – полутоновое изображение
- ❑ Линейный фильтр определяется вещественнозначной функцией  $F$ , заданной на растре, которая называется **ядром фильтра** (kernel)
- ❑ Операция фильтрации выполняется посредством вычисления дискретной свертки изображения  $I$  с ядром  $F$ :

$$I'(x, y) = \sum_i \sum_j F(i, j) \cdot I(x + i, y + j)$$

- ❑ Окрестность называется **шаблоном** или **апертурой**
- ❑ Каждый пиксель совмещается с конкретной точкой шаблона – **ведущей позицией шаблона** (anchor)



# Операции с полутоновыми изображениями.

## Линейная фильтрация (2)

---

- ❑ Текущий пиксель находится на границе изображения?
  
- ❑ Возможные решения:
  - Обрезать края
  - Не учитывать в процессе суммирования пиксель, который реально не существует
  - Доопределить окрестности граничных пикселей посредством экстраполяции (например, простым дублированием граничных пикселей)
  - Доопределить окрестности граничных пикселей посредством зеркального отражения – завернуть изображение в тор



# Операции с полутоновыми изображениями.

## Линейная фильтрация (3)

*Исходное изображение*



*Отфильтрованное изображение*



$$\times \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



# Операции с полутоновыми изображениями.

## Выделение краев. Оператор Собеля

- ❑ **Оператор Собеля** – дискретный дифференциальный оператор, вычисляющий приближенные значения производных разного порядка для функции яркости пикселей
- ❑ Позволяет определить приближенное значение градиента функции интенсивности:
  - Вычисление сверток исходного изображения с ядрами  $G_x$  и  $G_y$  для определения первых производных по направлениям:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Магнитуда градиента:  $|G^{ij}| = \sqrt{(G_x^{ij})^2 + (G_y^{ij})^2} \approx |G_x^{ij}| + |G_y^{ij}|^*$
- Направление градиента:  $\theta^{ij} = \arctan(G_y^{ij} / G_x^{ij})$

\* Для снижения числа операций используются различные приближения (приведен один пример).



# Операции с полутоновыми изображениями.

## Выделение краев. Оператор Лапласа

- ❑ **Оператор Лапласа:**  $\Delta f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
- ❑ Дискретный аналог оператора Лапласа используется при обработке изображений
- ❑ Определение ребер объектов на изображении:
  - Ребра формируются из множества пикселей, в которых оператор Лапласа принимает нулевые значения
  - Нули вторых производных функции соответствуют экстремальным перепадам интенсивности
- ❑ В простейшем случае применяется апертура размером 3x3 и ядро представляется матрицей:

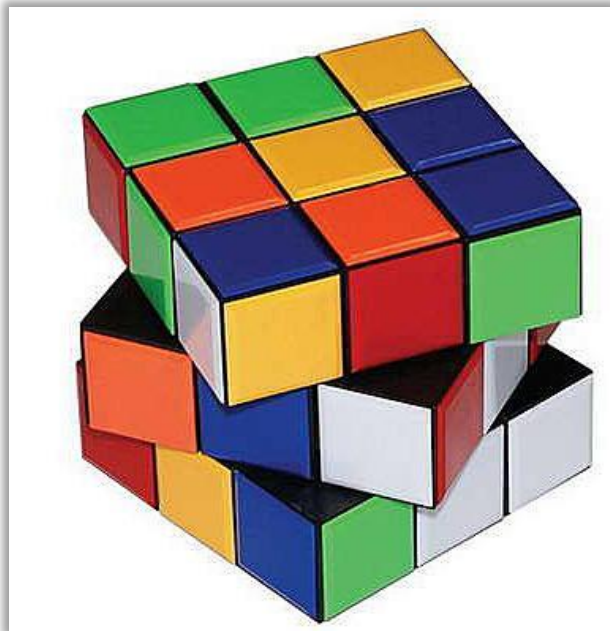
$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



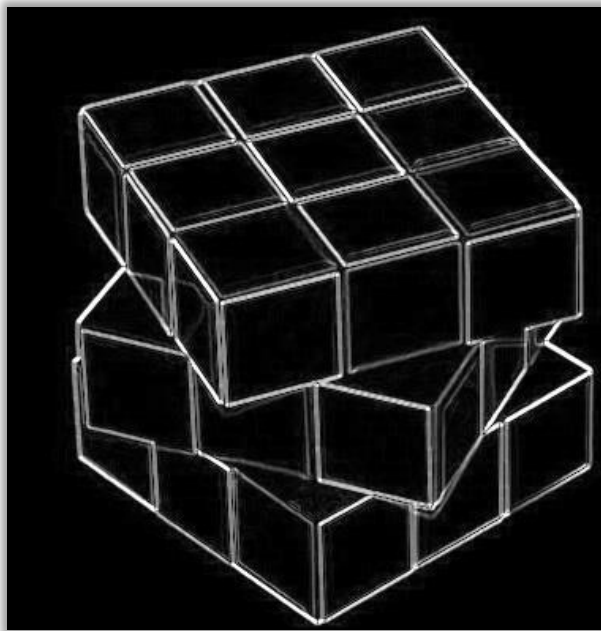
# Операции с полутоновыми изображениями.

## Выделение краев. Примеры

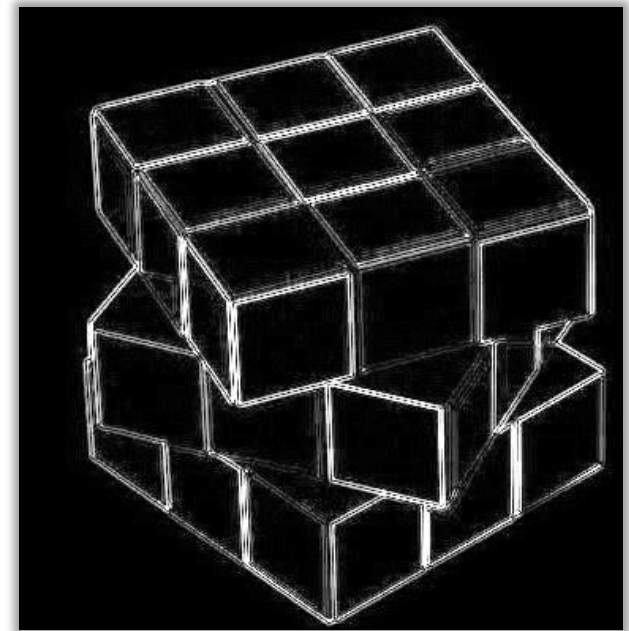
*Исходное  
изображение*



*Результат применения  
оператора Собеля*



*Результат применения  
оператора Лапласа*



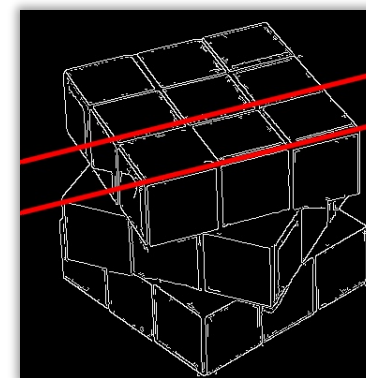
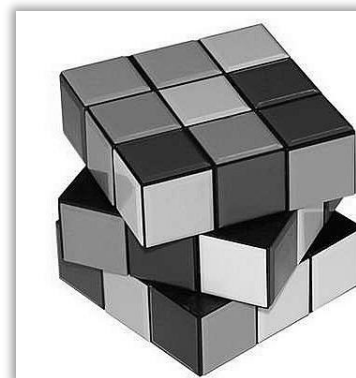
- ❑ **Примечание:** существуют продвинутые алгоритмы поиска краев, детектор Канни\* – один из таких алгоритмов

\* Canny G. A computational approach to edge detection // IEEE Trans. on PAMI. – Vol. PAMI-8, No. 6. – 1986.

# Операции с полутоновыми изображениями.

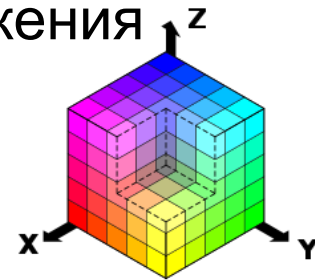
## Другие примеры операций

- ❑ Повышение контраста изображения (увеличение диапазона полутонов)
  - Линейная растяжка гистограммы
  - Нормализация гистограммы
  - Выравнивание (эквализация) гистограммы
- ❑ Поиск примитивов на изображениях
  - Преобразование Хафа (поиск линий, окружностей)
  - Преобразование Радона (поиск линий)
- ❑ ...



# Цветные изображения

- ❑ Современные устройства, как правило, выдают изображения в формате RGB (Red-Green-Blue). RGB в нормальном диапазоне:  $I[x, y] = (r, g, b)$ ,  $r, g, b \in [0, 255]$



- ❑ В телевидении, в алгоритмах сжатия и в некоторых видеосистемах используется пространство YUV (Y – яркость, U, V – цвет). Кодирование в телевизионных стандартах:

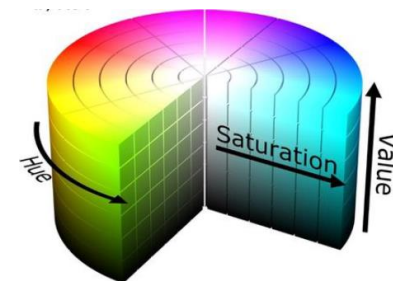
$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

$$U = -0.147 \times R - 0.289 \times G + 0.436 \times B$$

$$V = 0.615 \times R - 0.515 \times G - 0.100 \times B$$

- ❑ В графических редакторах используется нелинейное пространство HSV (Hue-Saturation-Value)

- $H \in [0, 360]$  – цветовой тон,  $S \in [0, 1]$  – насыщенность (чем ближе к нулю, тем ближе к нейтральному серому),  $V \in [0, 1]$  – яркость

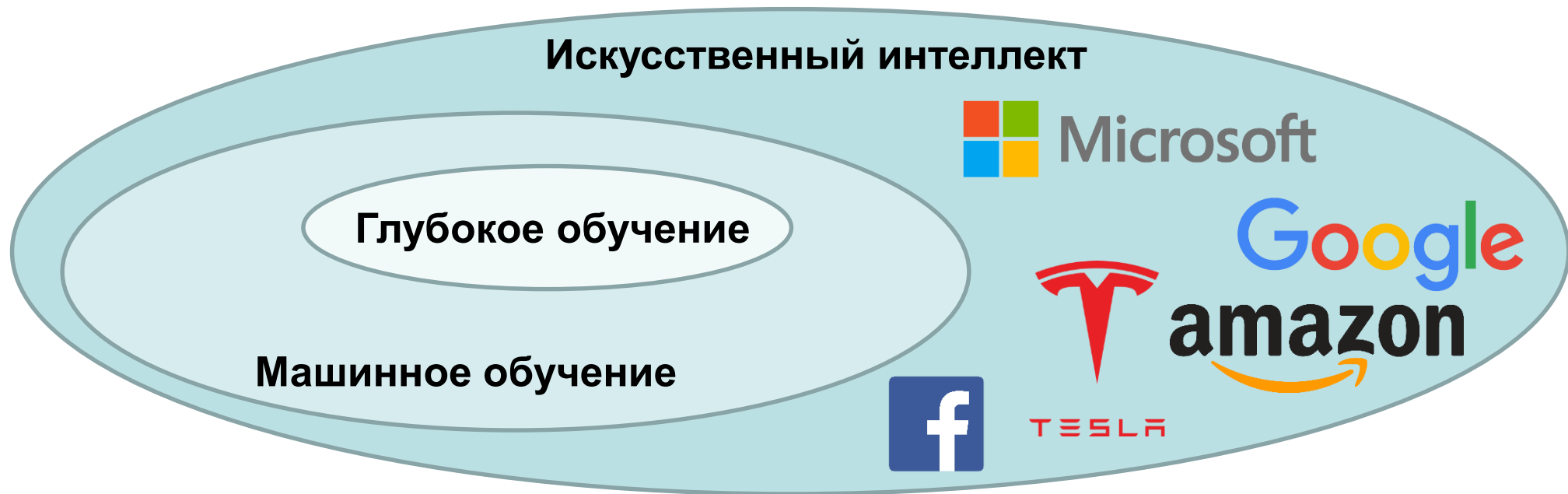


# **РЕЗУЛЬТАТЫ РЕШЕНИЯ ЗАДАЧ КОМПЬЮТЕРНОГО ЗРЕНИЯ С ПОМОЩЬЮ ГЛУБОКОГО ОБУЧЕНИЯ**



# Что такое «глубокое обучение»?

- ❑ **Глубокое обучение** (Deep Learning) – область машинного обучения (Machine Learning), которая рассматривает методы решения задач искусственного интеллекта (Artificial Intelligence) с использованием глубоких нейронных сетей



# План рассмотрения задач

---

- ❑ Постановка задачи
- ❑ Общепринятые показатели качества решения задачи
- ❑ Результаты решения задачи на открытом наборе данных
- ❑ Примеры приложений из реальной жизни

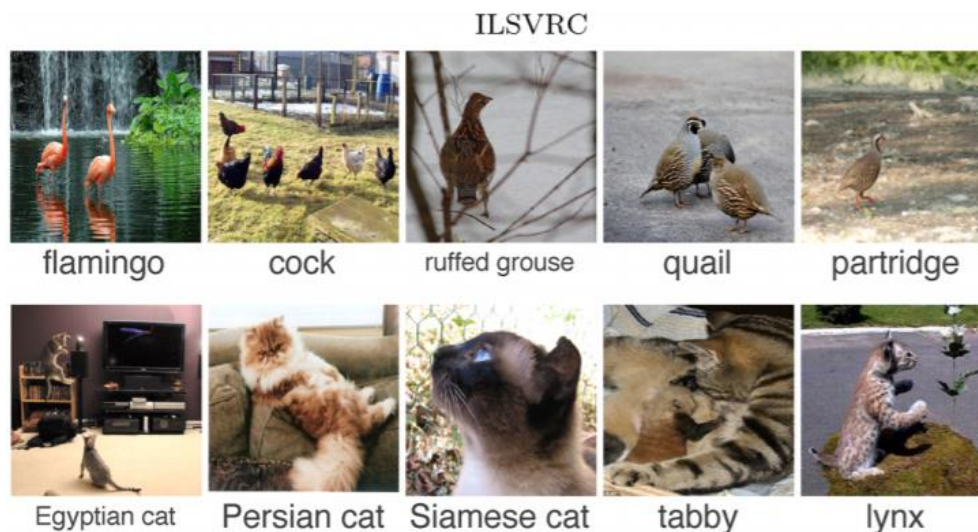




# Классификация изображений.

## Постановка задачи

- ❑ Задача классификации изображений состоит в том, чтобы поставить в соответствие изображению класс объектов, содержащихся на этом изображении
- ❑ Примеры изображений и соответствующих им классов:



\* Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A.C., Fei-Fei L. ImageNet Large Scale Visual Recognition Challenge // International Journal of Computer Vision, 2015.

# Классификация изображений.

## Показатели качества

- Предположим, что  $N$  – количество категорий изображений
- Для каждого изображения  $I_j, j = \overline{1, S}$  в выборке метод строит вектор достоверностей  $p^j = (p_1^j, p_2^j, \dots, p_N^j)$ , где  $p_i^j$  – достоверность того, что изображение  $I_j$  принадлежит классу  $i$
- **Точность top-K** (top-K accuracy) определяется следующим образом:

$$topK = \frac{\sum_{j=1}^S 1_{\{i_1^j, i_2^j, \dots, i_K^j\}}(l_j)}{S},$$

где  $\{i_1^j, i_2^j, \dots, i_K^j\} \subseteq \{1, 2, \dots, N\}$ , а  $p_{i_1^j}^j, p_{i_2^j}^j, \dots, p_{i_K^j}^j$  –  $K$  наибольших достоверностей,  $l_j$  – класс, которому принадлежит изображение  $I_j$  согласно разметке,  $1_{\{i_1^j, i_2^j, \dots, i_K^j\}}(l_j)$  –

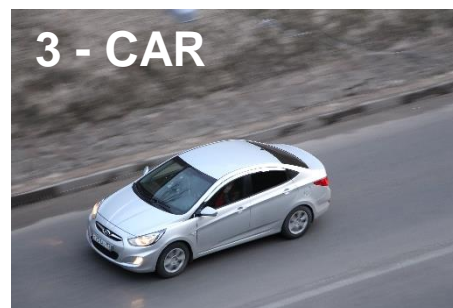
индикаторная функция



# Классификация изображений.

## Пример вычисления точности top-K (1)

- ❑ Допустимые классы изображений – {car, train, bus}
- ❑ 4 входных изображения (сверху указана разметка):



- ❑ Результаты классификации – вектора достоверностей (индекс компонента вектора соответствует номеру класса из допустимого множества):

1 – (0.5, 0.78, 0.8)

2 – (0.7, 0.76, 0.65)

3 – (0.9, 0.2, 0.4)

4 – (0.3, 0.93, 0.91)



# Классификация изображений.

## Пример вычисления точности top-K (2)

### ❑ *Вычисление top-1:*

– Выбор класса с максимальной достоверностью



1 – (0.5, 0.78, 0.8)

CAR → 0.5  
TRAIN → 0.78  
**BUS → 0.8**



2 – (0.7, 0.76, 0.65)

CAR → 0.7  
**TRAIN → 0.76**  
BUS → 0.65



3 – (0.9, 0.2, 0.4)

**CAR → 0.9**  
TRAIN → 0.2  
BUS → 0.4



4 – (0.3, 0.93, 0.91)

CAR → 0.3  
**TRAIN → 0.93**  
BUS → 0.91

# Классификация изображений.

## Пример вычисления точности top-K (2)

### ❑ *Вычисление top-1:*

– Сопоставление с разметкой



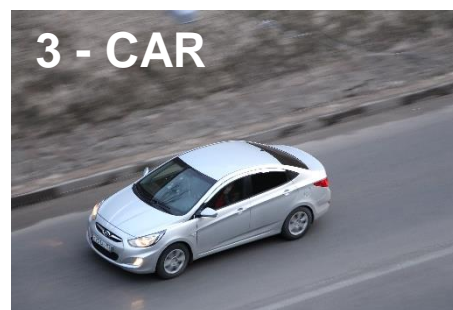
1 – (0.5, 0.78, 0.8)

CAR → 0.5  
TRAIN → 0.78  
**BUS → 0.8**



2 – (0.7, 0.76, 0.65)

CAR → 0.7  
**TRAIN → 0.76**  
BUS → 0.65



3 – (0.9, 0.2, 0.4)

**CAR → 0.9**  
TRAIN → 0.2  
BUS → 0.4



4 – (0.3, 0.93, 0.91)

CAR → 0.3  
**TRAIN → 0.93**  
BUS → 0.91

– Значение точности

$$top1 = \frac{1 + 0 + 1 + 0}{4} = \frac{2}{4} = \frac{1}{2}$$



# Классификация изображений.

## Пример вычисления точности top-K (3)

### □ *Вычисление top-2:*

- Выбор классов, соответствующих двум наибольшим достоверностям



1 – (0.5, 0.78, 0.8)

CAR → 0.5  
TRAIN → **0.78**  
BUS → **0.8**



2 – (0.7, 0.76, 0.65)

**CAR** → **0.7**  
TRAIN → **0.76**  
BUS → 0.65



3 – (0.9, 0.2, 0.4)

**CAR** → **0.9**  
TRAIN → 0.2  
**BUS** → **0.4**



4 – (0.3, 0.93, 0.91)

CAR → 0.3  
**TRAIN** → **0.93**  
**BUS** → **0.91**

# Классификация изображений.

## Пример вычисления точности top-K (4)

### □ *Вычисление top-2:*

- Сопоставление с разметкой – поиск размеченного класса среди выбранной пары классов



1 – (0.5, 0.78, 0.8)

CAR → 0.5

TRAIN → 0.78

BUS → 0.8

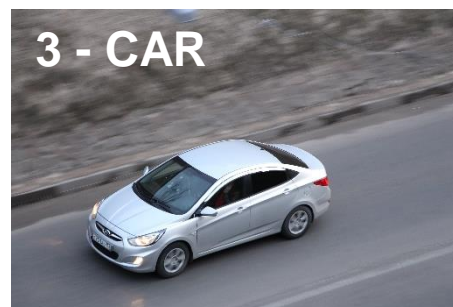


2 – (0.7, 0.76, 0.65)

CAR → 0.7

TRAIN → 0.76

BUS → 0.65



3 – (0.9, 0.2, 0.4)

CAR → 0.9

TRAIN → 0.2

BUS → 0.4



4 – (0.3, 0.93, 0.91)

CAR → 0.3

TRAIN → 0.93

BUS → 0.91

- Значение точности

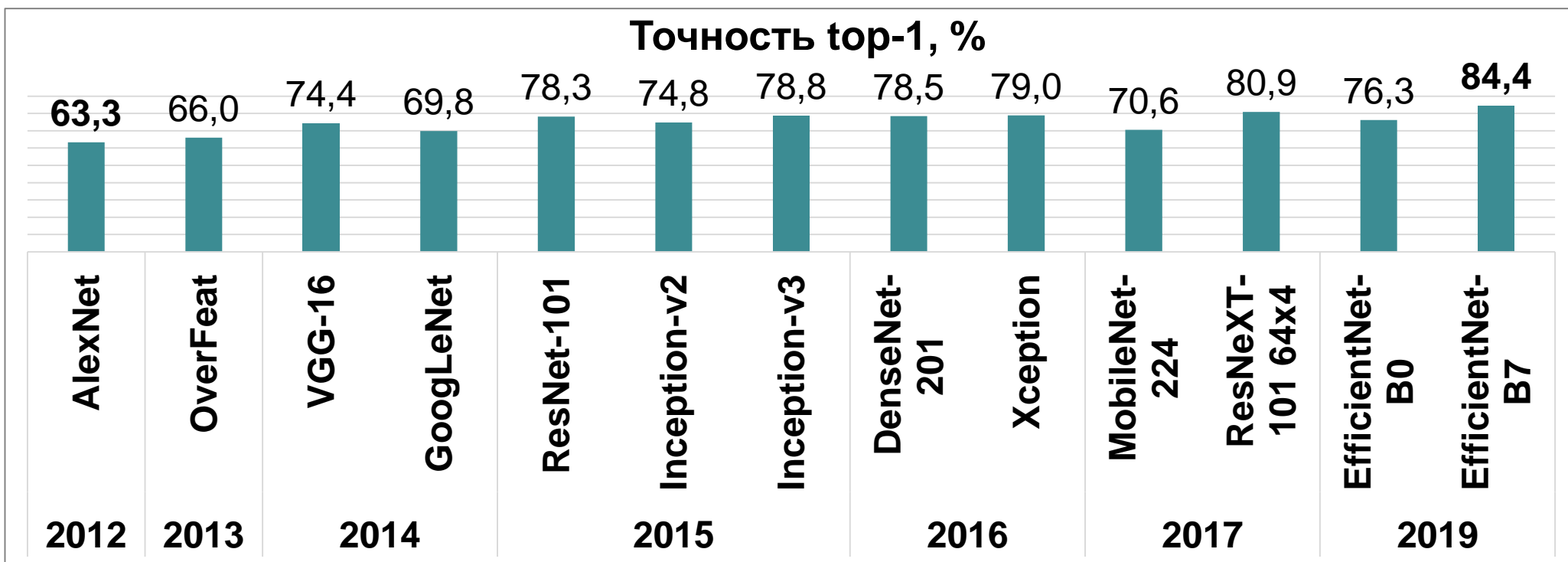
$$top2 = \frac{1 + 1 + 1 + 1}{4} = \frac{4}{4} = 1$$



# Классификация изображений.

## Результаты решения задачи на ImageNet

- Изменение точности top-1 на наборе данных ImageNet для избранных моделей:



- *За 7 лет точность выросла на 20%*





# Классификация изображений.

## Примеры приложений

---

- ❑ Автоматический контроль качества производимой продукции
- ❑ Управление промышленными роботами
- ❑ Обнаружение различных событий (оставленные вещи, подозрительные люди)
- ❑ Задачи взаимодействия человека с компьютером
- ❑ Анализ медицинских изображений (принятие решений о наличии какого-либо заболевания по снимкам КТ, МРТ)
- ❑ Навигация автономных транспортных средств или мобильных роботов
- ❑ Организация информации, например, для индексирования баз данных изображений и последовательностей изображений



# Детектирование объектов на изображениях.

## Постановка задачи

- ❑ Задача детектирования объектов состоит в том, чтобы определить положение прямоугольника, окаймляющего объект заданного класса

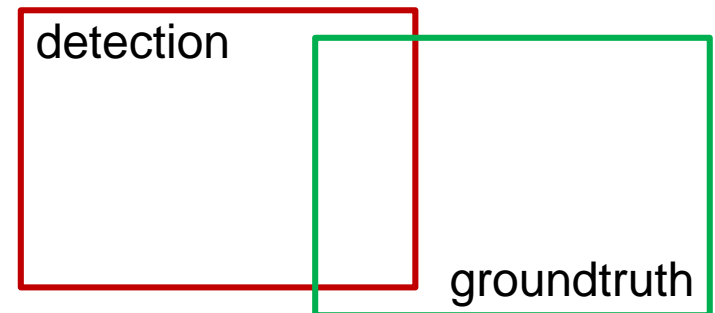


# Детектирование объектов на изображениях.

## Показатели качества (1)

### □ Обозначения:

- $IoU = \frac{S_{d \cap g}}{S_{d \cup g}}$  – доля перекрытия обнаруженного (detection) и размеченного (groundtruth) окаймляющих прямоугольников (Intersection over Union),  $IoU \in [0; 1]$
- $TP$  – количество объектов, для которых доля перекрытия не меньше некоторого порога  $\tau$  (т.е. считается, что объект обнаружен правильно – true positive)
- $FP$  – количество обнаруженных объектов с долей перекрытия, меньшей  $\tau$  (объект найден ошибочно), или объект обнаружен более одного раза (false positives)
- $FN$  – количество необнаруженных объектов (false negatives)



# Детектирование объектов на изображениях.

## Показатели качества (2)

---

- ❑ Пороговое значение  $\tau$ , как правило, выбирается равным 0.5
- ❑ **Точность** (precision) – отношение количества правильно обнаруженных объектов к общему числу срабатываний детектора

$$Precision = p = \frac{TP}{TP + FP}$$

- ❑ **Отклик** (recall) – отношение количества правильно обнаруженных объектов к общему числу объектов

$$Recall = r = \frac{TP}{TP + FN}$$



# Детектирование объектов на изображениях.

## Показатели качества (3.1)

- ❑ **Средняя точность предсказания** (average precision) – математическое ожидание точностей

$$AP = \int_0^1 p(r) dr$$

- ❑ Доопределение функции точности:

Кол-во объектов разметки	Кол-во обнаруж. объектов	TP	FP	FN	Precision	Recall
>0	>0	?	?	?	$\frac{TP}{TP + FP}$	$\frac{TP}{TP + FN}$
0	>0	0	>0	0	0	1
>0	0	0	0	>0	1	0
0	0	0	0	0	1	1



# Детектирование объектов на изображениях.

## Показатели качества (3.2)

- **Средняя точность предсказания** (average precision) – математическое ожидание точностей

$$AP = \int_0^1 p(r) dr$$

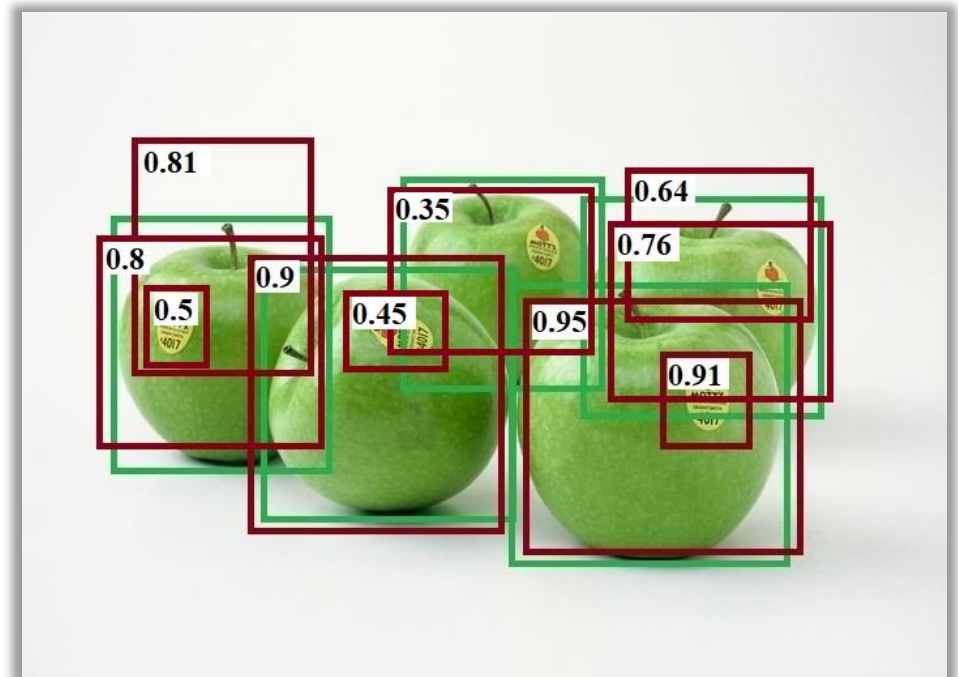
- Схема вычисления:
  - Обнаруженные окаймляющие прямоугольники сортируются в порядке убывания достоверности наличия в них объектов
  - Для каждого обнаруженного прямоугольника выполняется поиск соответствия из разметки согласно условию  $IoU \geq \tau$
  - Выполняется вычисление точности и отклика
  - Строится зависимость точности от отклика
  - Вычисляется площадь под графиком построенной зависимости



# Детектирование объектов на изображениях.

## Показатели качества (4)

- Пример вычисления средней точности предсказания:
  - Исходное изображение – фотография яблок из набора данных ImageNet [<http://www.image-net.org>]
  - Разметка содержит окаймляющие прямоугольники для 5 яблок (зеленые прямоугольники)
  - Алгоритм детектирования обнаруживает 10 яблок (красные прямоугольники)
  - Для определенности предполагается, что достоверности различны
  - Считается, что объект обнаружен правильно при  $IoU \geq 0.5$

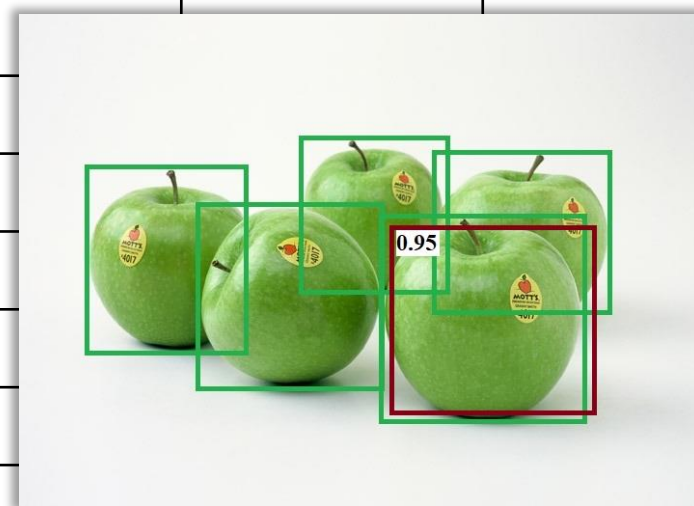


# Детектирование объектов на изображениях.

## Показатели качества (5.1)

- Пример вычисления средней точности предсказания:
  - Сортировка прямоугольников, вычисление точности и отклика

№	Достоверность	Объект?	Точность	Отклик
1	0.95	Да	$1/1 = 1.0$	$1/5 = 0.2$
2	0.91			
3	0.9			
4	0.81			
5	0.8			
6	0.76			
7	0.64			
8	0.5			
9	0.45			
10	0.35			



**Нашлось соответствие  
из разметки и  $IoU \geq 0.5$**

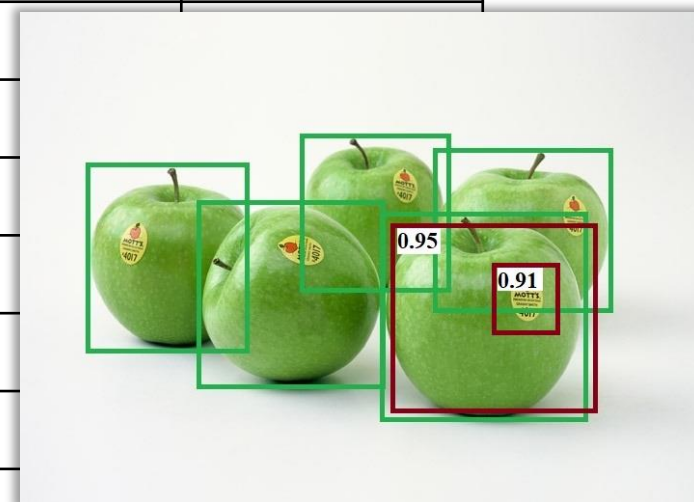


# Детектирование объектов на изображениях.

## Показатели качества (5.2)

- Пример вычисления средней точности предсказания:
  - Сортировка прямоугольников, вычисление точности и отклика

№	Достоверность	Объект?	Точность	Отклик
1	0.95	Да	$1/1 = 1.0$	$1/5 = 0.2$
2	0.91	Нет	$1/2 = 0.5$	$1/5 = 0.2$
3	0.9			
4	0.81			
5	0.8			
6	0.76			
7	0.64			
8	0.5			
9	0.45			
10	0.35			



**Нашлось соответствие  
из разметки, но  $IoU < 0.5$**

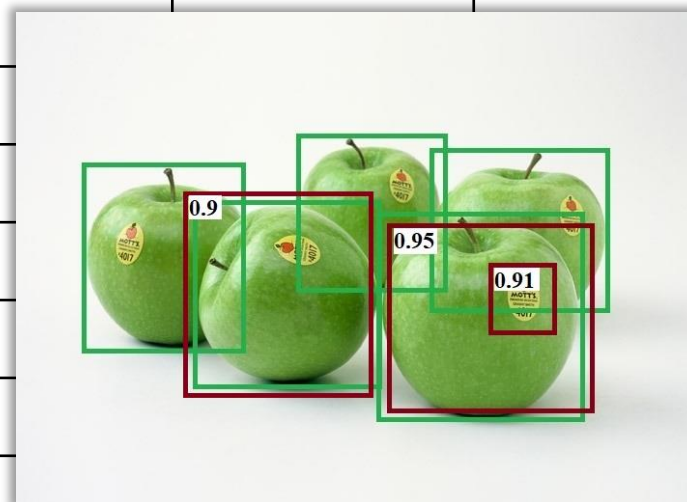
# Детектирование объектов на изображениях.

## Показатели качества (5.3)

- Пример вычисления средней точности предсказания:
  - Сортировка прямоугольников, вычисление точности и отклика

№	Достоверность	Объект?	Точность	Отклик
1	0.95	Да	1/1 =	
2	0.91	Нет	1/2 =	
3	0.9	Да	2/3 ≈ 0.67	2/5 = 0.4
4	0.81			
5	0.8			
6	0.76			
7	0.64			
8	0.5			
9	0.45			
10	0.35			

Нашлось соответствие  
из разметки и  $IoU \geq 0.5$



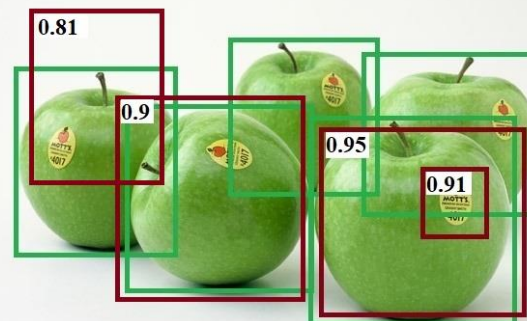
# Детектирование объектов на изображениях.

## Показатели качества (5.4)

- Пример вычисления средней точности предсказания:
  - Сортировка прямоугольников, вычисление точности и отклика

№	Достоверность	Объект?	Точность	Отклик
1	0.95	Да	$1/1 = 1.0$	$1/5 = 0.2$
2	0.91	Нет	$1/2$	
3	0.9	Да	$2/3 \approx$	
4	0.81	Нет	$2/4 = 0.5$	$2/5 = 0.4$
5	0.8			
6	0.76			
7	0.64			
8	0.5			
9	0.45			
10	0.35			

**Нашлось соответствие  
из разметки, но  $IoU < 0.5$**



# Детектирование объектов на изображениях.

## Показатели качества (5.6)

- Пример вычисления средней точности предсказания:
  - Сортировка прямоугольников, вычисление точности и отклика

№	Достоверность	Объект?	Точность	Отклик
1	0.95	Да	$1/1 = 1.0$	$1/5 = 0.2$
2	0.91	Нет	$1/2 = 0.5$	$1/5 = 0.2$
3	0.9	Да	$2/3 \approx 0.67$	$2/5 = 0.4$
4	0.81	Нет	$2/4 = 0.5$	$2/5 = 0.4$
5	0.8	Да	$3/5 = 0.6$	$3/5 = 0.6$
6	0.76	Да	$4/6 \approx 0.67$	$4/5 = 0.8$
7	0.64	Нет	$4/7 \approx 0.57$	$4/5 = 0.8$
8	0.5	Нет	$4/8 = 0.5$	$4/5 = 0.8$
9	0.45	Нет	$4/9 \approx 0.44$	$4/5 = 0.8$
10	0.35	Да	$5/10 = 0.5$	$5/5 = 1.0$

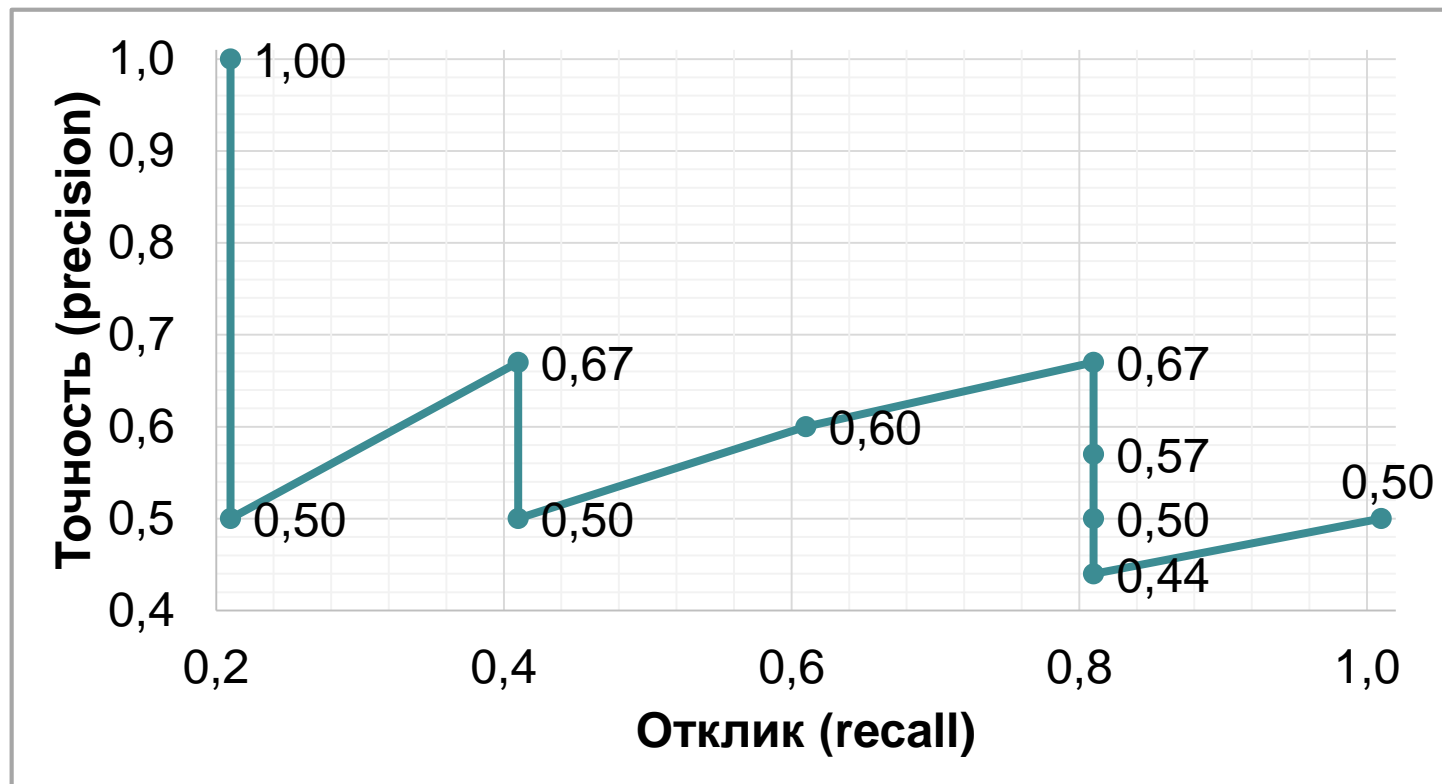
Отклик  
нарастает



# Детектирование объектов на изображениях.

## Показатели качества (6)

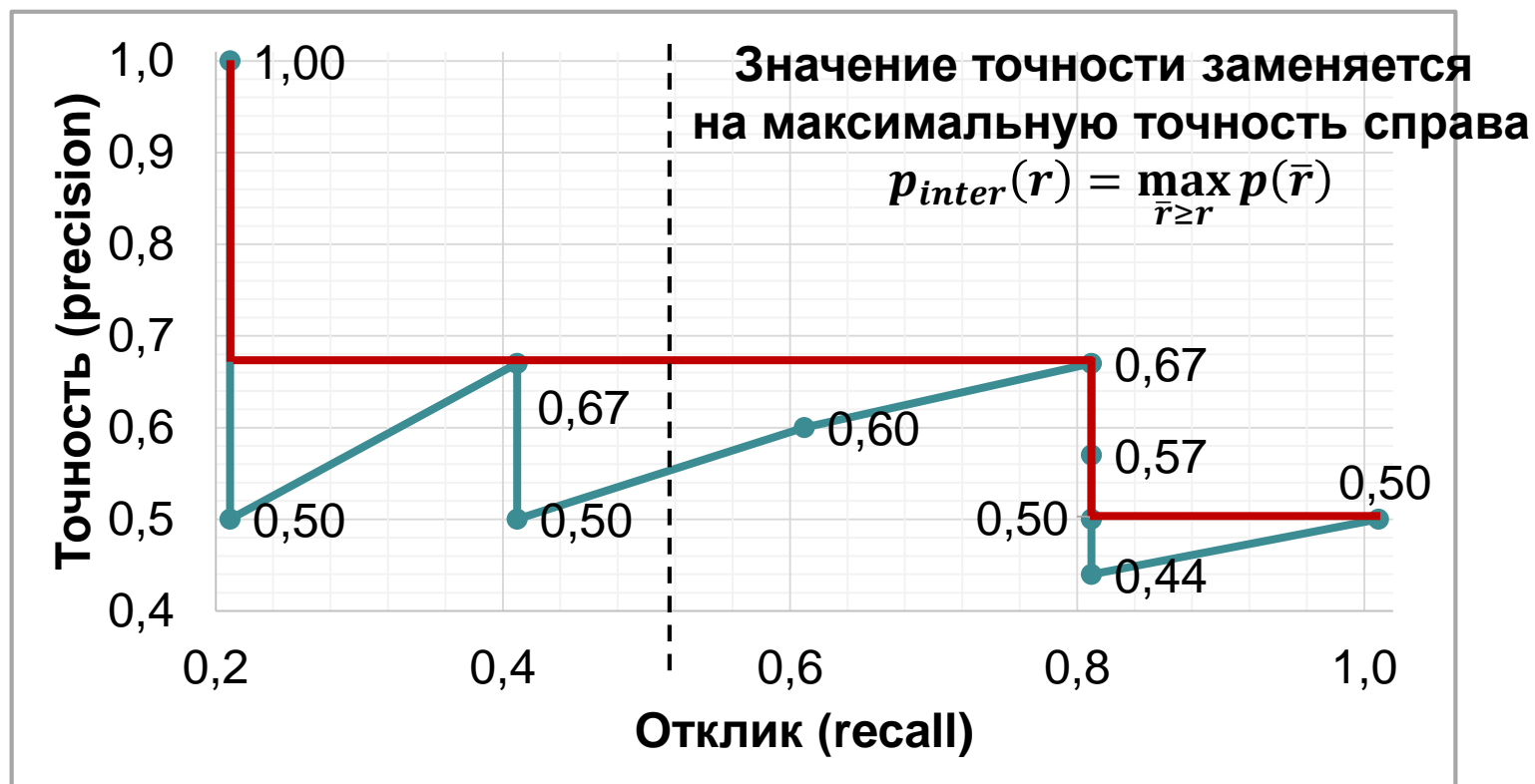
- Пример вычисления средней точности предсказания:
  - Построение зависимости точности от отклика
  - Результат – зигзагообразная кривая



# Детектирование объектов на изображениях.

## Показатели качества (7)

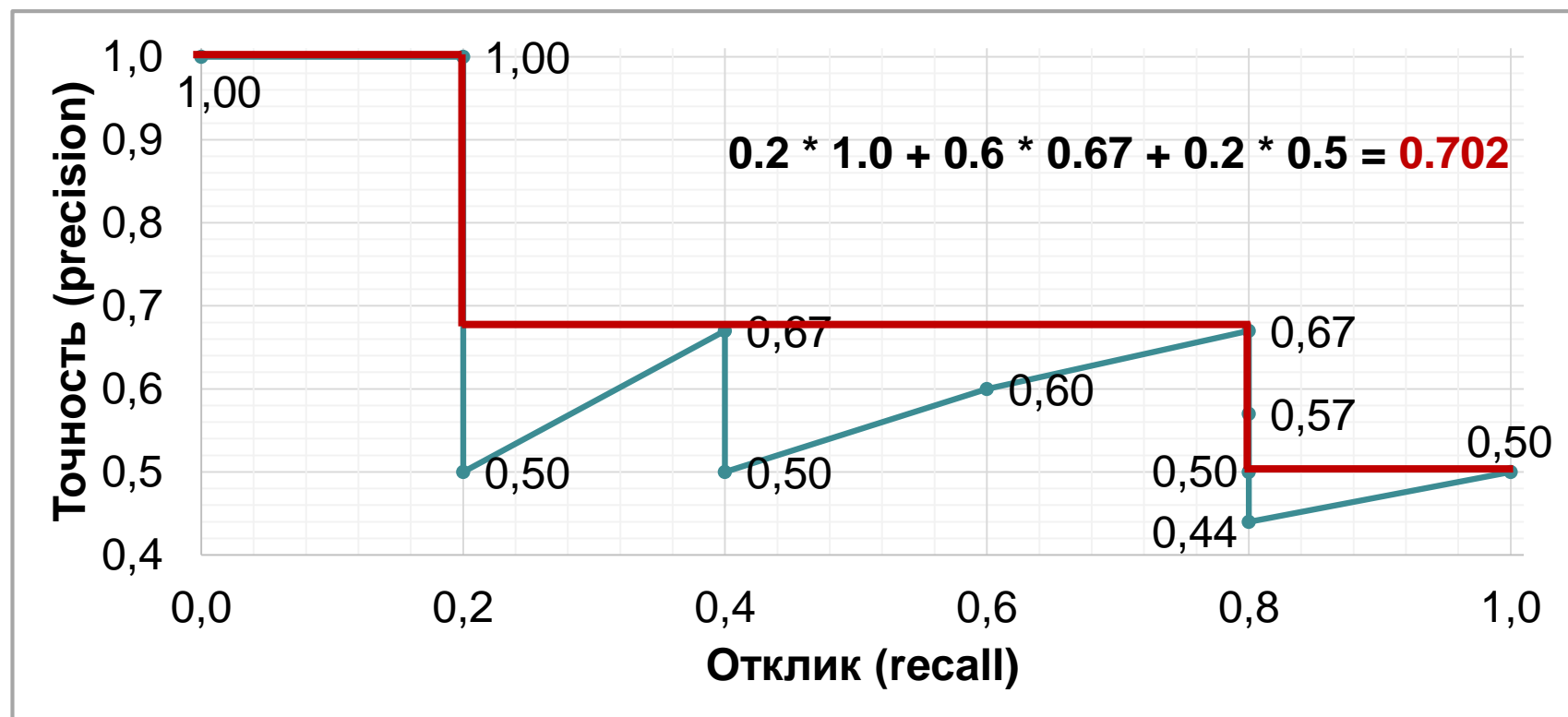
- Пример вычисления средней точности предсказания:
  - Вычисление площади под зигзагообразной кривой – интерполяция и вычисление площади под «ступенькой»



# Детектирование объектов на изображениях.

## Показатели качества (8)

- Пример вычисления средней точности предсказания:
  - Вычисление площади под зигзагообразной кривой – интерполяция и вычисление площади под «ступенькой»



# Детектирование объектов на изображениях.

## Результаты решения задачи на PASCAL VOC 2012

Класс объектов	LSVM <sup>1</sup> – модель деформируемых частей на базе HOG-признаков 2011 г., %	ALTDETv2 <sup>2</sup> – нейронная сеть 2019 г., %	Разница, %
BOTTLE	18.1	89.0	70.9
CAT	24.2	98.0	73.8
DINING TABLE	4.5	82.1	77.6
DOG	17.5	97.4	79.9
HORSE	15.2	97.6	<b>82.4</b>
PERSON	7.9	96.1	<b>88.2</b>
SOFA	7.1	87.0	79.9
TV/MONITOR	25.7	92.5	66.8
<b>Среднее по 20 классам</b>	<b>20.9</b>	<b>92.9</b>	<b>72.0</b>

1. Felzenszwalb P.F., Girshick R.B., McAllester D., Ramanan D. Object Detection with Discriminatively Trained Part Based Models // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2010. – Vol. 32, No. 9.
2. ALTDETv2 [[http://host.robots.ox.ac.uk:8080/leaderboard/displaylb\\_main.php?challengeid=11&compid=4](http://host.robots.ox.ac.uk:8080/leaderboard/displaylb_main.php?challengeid=11&compid=4)], полное описание не опубликовано.





# Детектирование объектов на изображениях.

## Примеры приложений

---

- ❑ Сортировка изображений в домашних цифровых фотоальбомах
- ❑ Поиск товаров по их изображениям в интернет-магазинах
- ❑ Системы контроля передвижения людей и животных
- ❑ Целевой поиск изображений в социальных сетях
- ❑ Анализ складских запасов
- ❑ Обнаружение препятствий
- ❑ Одновременная локализации объектов и построение карты в неизвестном пространстве (Simultaneous localization and mapping, SLAM)
- ❑ Дефектоскопия – обнаружение каких-либо недостатков (изъянов) в различных конструкциях



# Семантическая сегментация изображений.

## Постановка задачи

- Задача семантической сегментации состоит в том, чтобы каждому пикселю изображения поставить в соответствие класс объектов, которому этот пиксель принадлежит (разные цвета соответствуют разным классам)



Оригинал



Разметка



Результат сегментации

\* The PASCAL Visual Object Classes Homepage [<http://host.robots.ox.ac.uk/pascal/VOC>].

# Семантическая сегментация изображений.

## Показатели качества

- ❑ **Метрика  $IoU$**  (Intersection over Union) или индекс Жаккара (Jaccard index)

$$IoU = \frac{TP}{TP + FP + FN}$$

где  $TP$  – количество правильно проклассифицированных пикселей (true positives),

$FP$  – количество пикселей, которые метод проклассифицировал как принадлежащие классу, но они таковыми не являются (false positives),

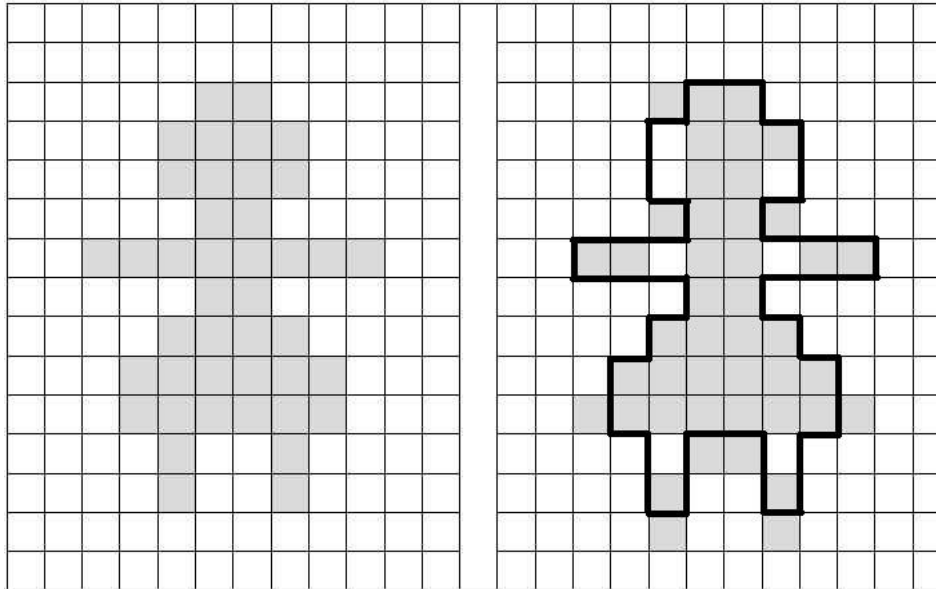
$FN$  – количество пикселей, которые принадлежат классу, но метод проклассифицировал их как не принадлежащие классу (false negatives)

	Предсказание	
	True	False
Разметка	True	FN
	FP	TN

# Семантическая сегментация изображений.

## Пример вычисления показателя качества (1)

- ❑ Целевой класс объектов – «человечек»



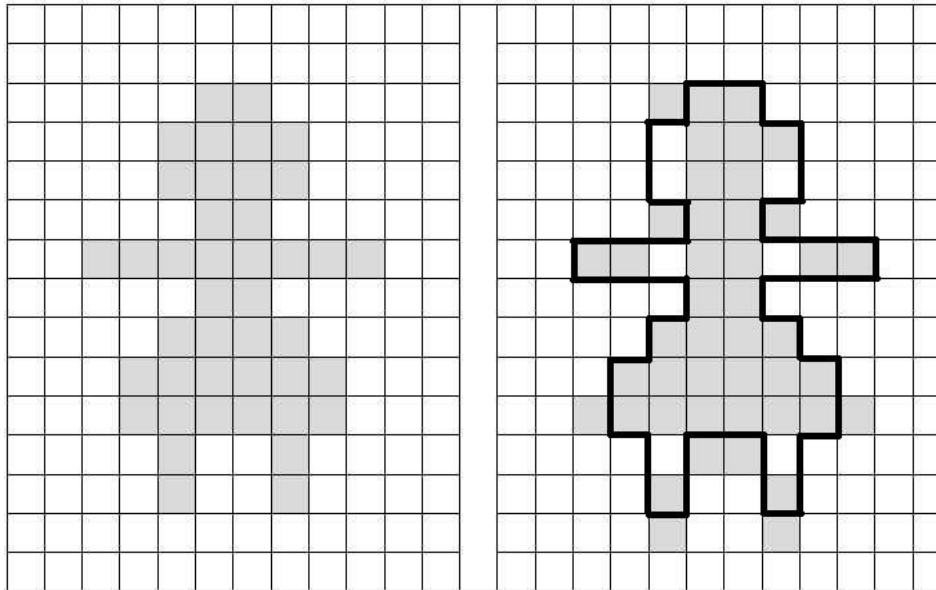
- ❑ Разметка – изображение слева (серым цветом залит размеченный объект)
- ❑ Результат сегментации – изображение справа (серым цветом залит сегментированный объект, черным обведены границы пикселей, соответствующих разметке)



# Семантическая сегментация изображений.

## Пример вычисления показателя качества (2)

- Целевой класс объектов – «человечек»



- $TP = 35$  (серые пиксели внутри черной границы)
- $FP = 7$  (серые пиксели за пределами черной границы)
- $FN = 7$  (белые пиксели внутри черной границы)

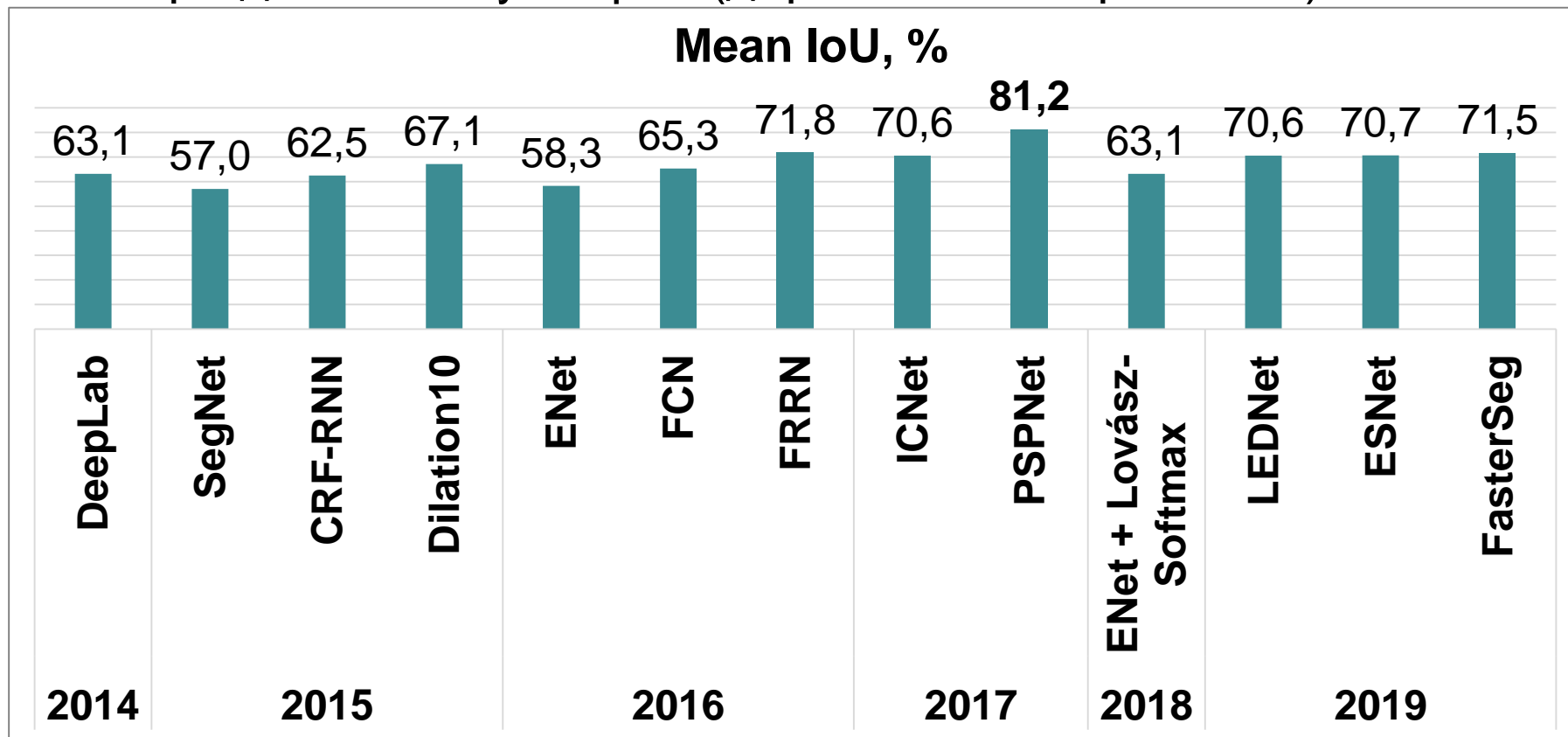
$$IoU = \frac{35}{35 + 7 + 7} = \frac{35}{49} = \frac{5}{7}$$



# Семантическая сегментация изображений.

## Результаты решения задачи на CityScapes

- Изменение среднего значения IoU по набору классов на наборе данных CityScapes (дорожные изображения):



\* Real-Time Semantic Segmentation on Cityscapes test [<https://paperswithcode.com/sota/real-time-semantic-segmentation-on-cityscapes>].

# Семантическая сегментация изображений.

## Примеры приложений

---

- ❑ Сегментация медицинских изображений (обнаружение патологий, диагностика)
- ❑ Системы управления автономными автомобилями (сегментации дорожных полос)
- ❑ Сегментация аэрофотоснимков
- ❑ Определение типов земной поверхности со спутников, например, контуров водоемов для предоставления более точной картографической информации
- ❑ Извлечение из изображений людей элементов их одежды для предложения похожих товаров в интернет-магазинах

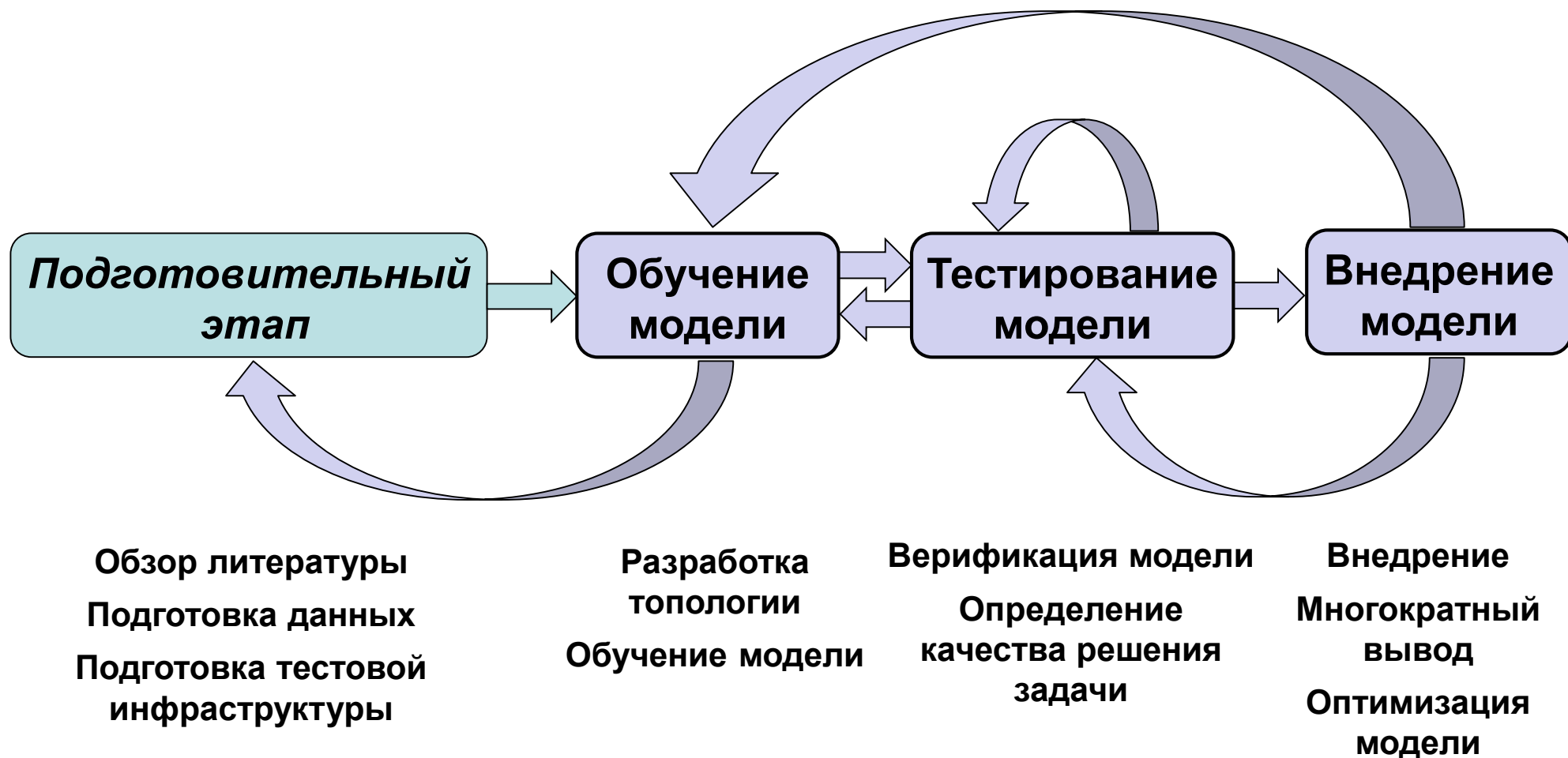


# **ОБЩАЯ СХЕМА РЕШЕНИЯ ЗАДАЧ КОМПЬЮТЕРНОГО ЗРЕНИЯ С ПОМОЩЬЮ ГЛУБОКОГО ОБУЧЕНИЯ**





# Общая схема решения задач с использованием глубокого обучения



# Предварительный этап решения задачи

---

## ❑ ***Обзор литературы***

- Существующие модели (возможность переноса обучения)
- Показатели качества
- Открытые наборы данных

## ❑ ***Подготовка и разметка данных***

- Поиск похожих данных в сети Интернет
- Сбор и разметка собственных данных
- Предварительная обработка данных и подготовка разметки

## ❑ ***Подготовка тестовой инфраструктуры***

- Поиск существующих или разработка собственных инструментов для оценки качества работы моделей



# Обучение и тестирование модели

---

- ❑ **Разработка топологии модели**, разработка собственных слоев и функций ошибки
  - ❑ **Обучение (*training*) модели**
  - ❑ **Тестирование (*verification*) модели** – оценка качества работы модели в соответствии с выбранными показателями
- 
- ❑ **Примечание:** на этапах обучения и тестирования используется тренировочный фреймворк



# Внедрение модели

---

- ❑ Портирование модели на фреймворк, который будет использован для вывода (inference)
- ❑ Анализ сложности и производительности модели на целевом оборудовании
- ❑ **Оптимизация** и сжатие модели
- ❑ Возврат к обучению и тестированию модели
- ❑ **Множественный вывод модели** на целевом оборудовании



# Заключение

---

- ❑ Значительное число задач компьютерного зрения решается с использованием методов глубокого обучения
- ❑ Предварительная обработка изображений (или кадров видео) играет важную роль в ходе подготовки данных
- ❑ Анализ производительности вывода глубоких моделей является важным этапом в процессе их внедрения в реальные системы



# Основная литература

- ❑ Хайкин С. Нейронные сети. Полный курс. – М.: Издательский дом «Вильямс». – 2006. – 1104 с.
- ❑ Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика. – 2002. – 344 с.
- ❑ Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press. – 2016. – [<http://www.deeplearningbook.org>].
- ❑ Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. – Изд-во «Питер». – 2018. – 476 с.
- ❑ Учебно-образовательный курс «Современные методы и технологии глубокого обучения в компьютерном зрении» [[http://hpc-education.unn.ru/ru/обучение/курсы/магистратура/deep\\_learning\\_in\\_computer\\_vision](http://hpc-education.unn.ru/ru/обучение/курсы/магистратура/deep_learning_in_computer_vision)].



# Контакты

---

- ❑ Кустикова Валентина  
к.т.н., доцент каф. МОСТ ИИТММ,  
ННГУ им. Н.И. Лобачевского  
[valentina.kustikova@itmm.unn.ru](mailto:valentina.kustikova@itmm.unn.ru)

