



Лаборатория компьютерной  
графики и мультимедиа  
ВМК МГУ имени М.В. Ломоносова

# Компьютерное зрение

Антон Конушин

Заведующий лабораторией компьютерной графики и  
мультимедиа ВМК МГУ



# Что такое компьютерное зрение?

---



**Задача зрения:** понять, что находится на изображении

**Тест Тьюринга для компьютерного зрения:**

Ответить на любой вопрос про изображении, на который может ответить человек.

Компьютерное зрение – часть области искусственного интеллекта (AI)



# Что и где находится на изображении?

Выделение объектов

Здание

Текст

Лицо

Текст

中华人民共和国万岁



世界人民大团结万岁

Человек

Человек

Лицо

Необходимо определить, есть ли на изображении объекты заданного типа и если да, то определить их положение



# Что за сцена и откуда?

- вне помещения
- город
- Пекин, Китай
- Пл. Тяньаньмэнь

Классификация  
изображений



Общая характеристика изображения

# Какой?



Характеристики (атрибуты) отдельных объектов

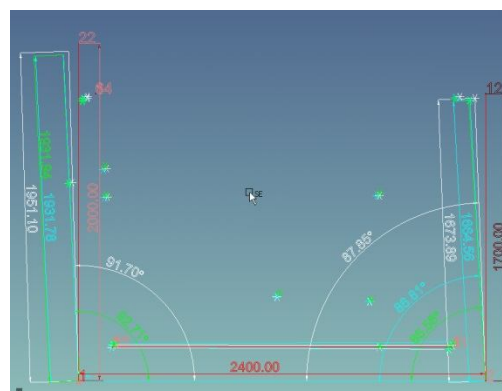


# Какой формы?

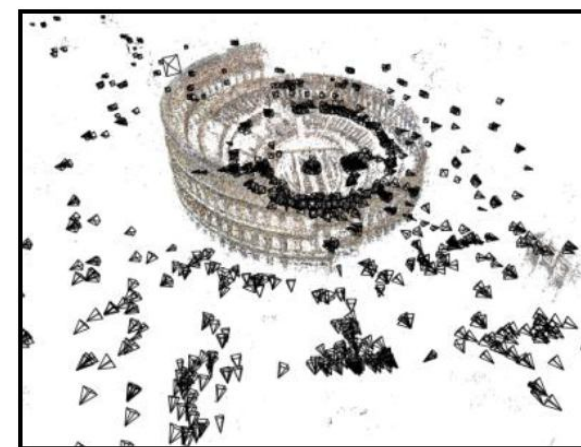
Стереозрение



Измерения  
расстояний по  
снимкам



3D моделирование по  
снимкам



Измерение расстояний и формы объектов

Почему зрение – это сложно?

# Точка обзора

---



Michelangelo 1475-1564





# Освещение

---

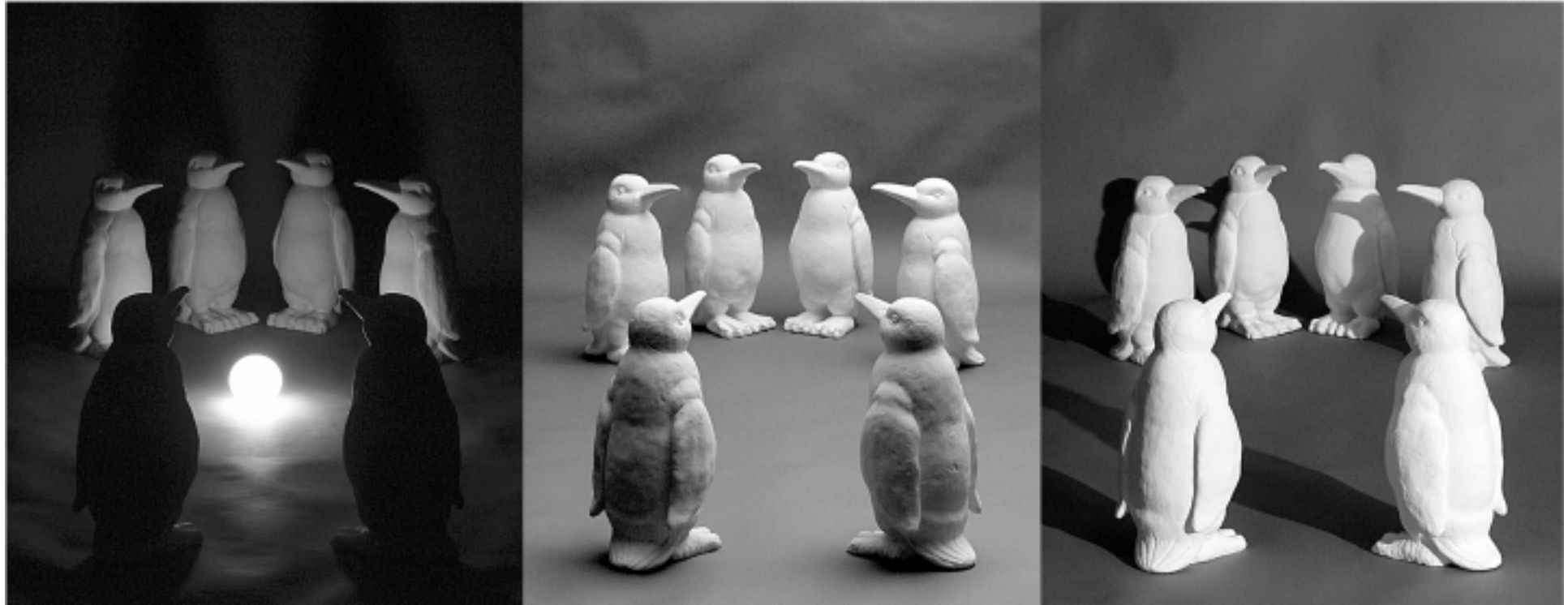


image credit: J. Koenderink



# Изменчивость объектов

---



Slide credit: Fei-Fei, Fergus & Torralba



# Зрение – это сложно

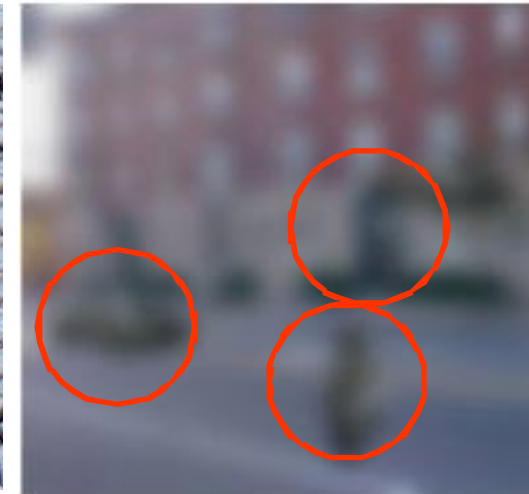
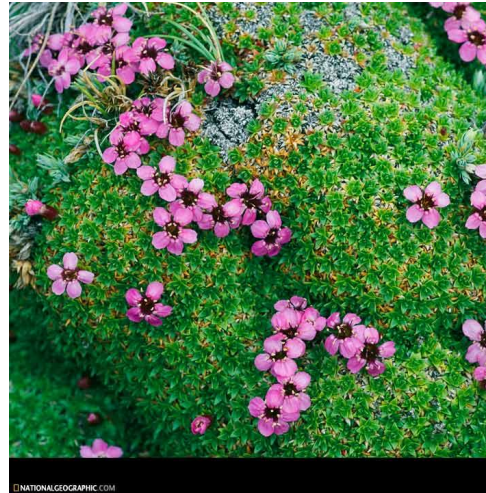
---



- Одно 2D изображение допускает разные 3D интерпретации
- Нужны априорные знания о структуре и свойствах мира



# Зрение человека



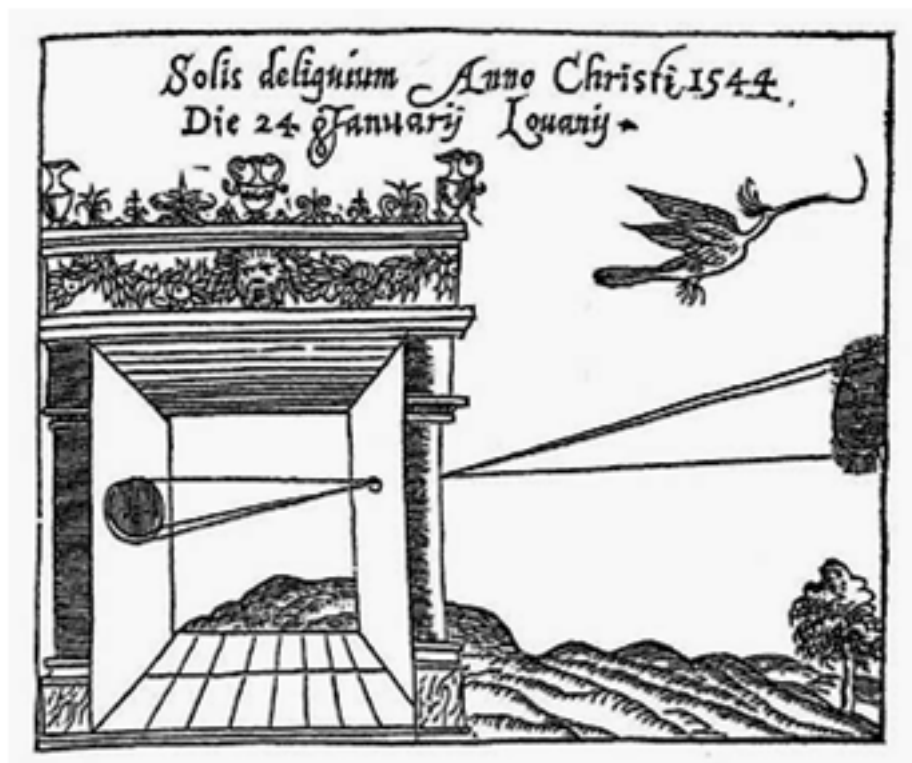
- Интерпретация подсказок, с учётом наших знаний о мире
- ~25% мозга занято ЭТИМ

# История и достижения компьютерного зрения



# Камера-обскура

---



Принцип был известен еще Аристотелю (384-322 до Н.Э.)

Первое устройство для получения изображений



# Живопись Ренессанса



*Яна Ван Эйк «Портрет Четы Арнольфини» 1434г*



# Секреты мастеров

---



BBC David Hockney's «Secret Knowledge»



# Камера-обскура с линзой (1500-1600е)



Возможность проецировать изображение на большой холст. Побочный результат – много левшей на картинах



<http://www.adme.ru/hudozhniki-i-art-proekty/sekretное-znanie-543505/>



## Камера-люцида (1807)

---



Camera Lucida — устройство, которое представляет собой конструкцию с призмой, которая крепится, например, на стойке к планшету.

Художник, глядя на свой рисунок одним глазом, видит реальное изображение, а другим — собственно рисунок и свою руку. Получается оптическая иллюзия, позволяющая точно переносить пропорции реальные на бумагу.



# Первая фотография

---



Самая первая фотография  
1825 год



Figure 5. J. N. Niepce.

Требовала 8 часов проявки

# Видео

---



1878 – первая скоростная съемка, Eadweard Muybridge



1888 – первое кино на плёнке, Louis Le Prince



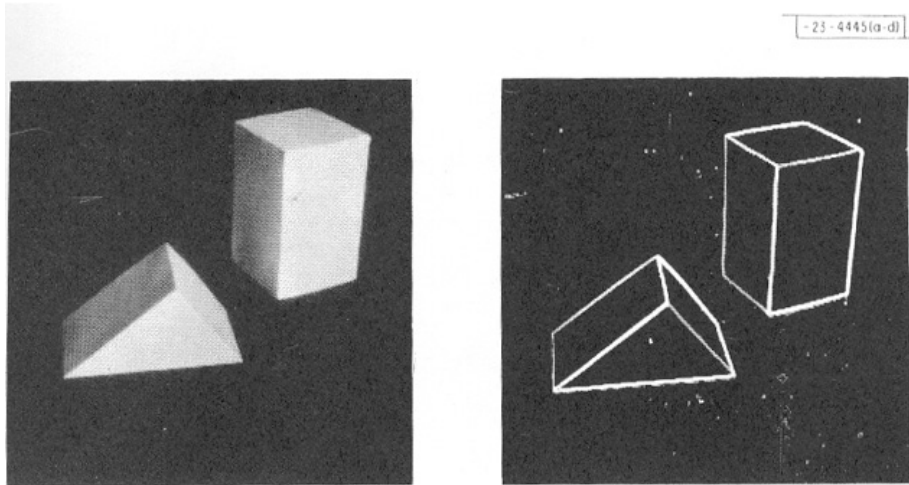
# Whirlwind, MIT (1951)

---



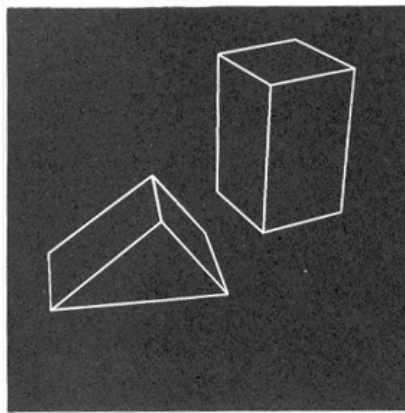
- Первый компьютер, отображающий текст и графику в реальном времени на мониторе
- Точками карту, значком самолёт.
- «Световое перо» для взаимодействия с экраном (запрос информации об объекте)

# Зарождение компьютерного зрения (1960)

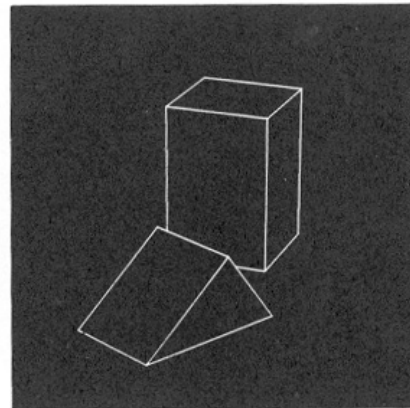


(a) Original picture.

(b) Differentiated picture.



(c) Line drawing.



(d) Rotated view.

L. G. Roberts, *Machine Perception of Three Dimensional Solids*,  
Ph.D. thesis, MIT Department of  
Electrical Engineering, 1960



# Детектор лиц Viola-Jones (2001)

---



Алгоритм Viola-Jones – первый быстрый и надежный алгоритм поиска лиц. Демонстрация силы машинного обучения.



# Распознавание по лицу



- Одна из самых популярных компьютерных технологий в кино («умные машины»)
- Системы автоматического паспортного контроля
- Распознавание лиц на фотографиях пользователей в соц.сетях

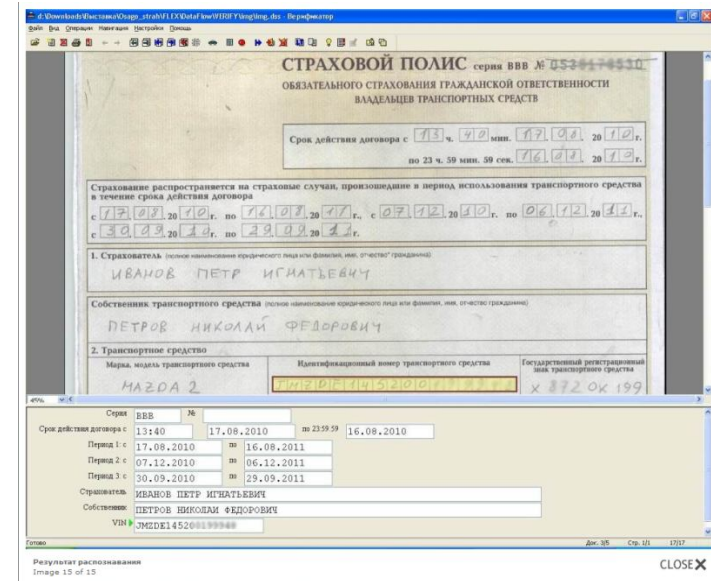




# Распознавание текста



FineReader, ABBYY, Россия  
<http://www.abbyy.ru/finereader/>



Cognitive Technologies, Россия  
<http://cognitiveforms.ru/products/cognitive-forms/>

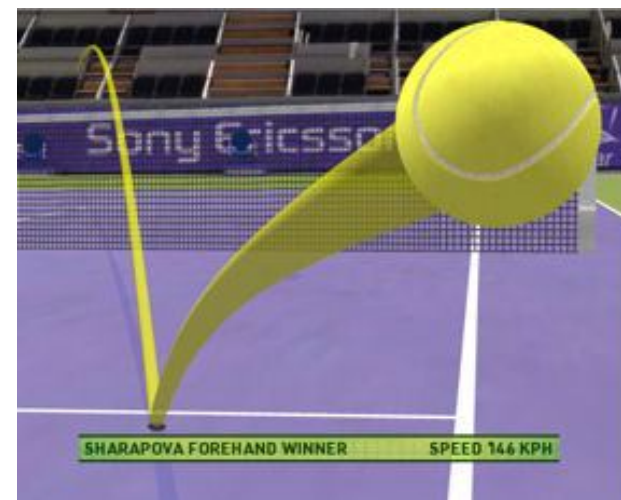


Распознавание текста в произвольных изображениях пока работает не так хорошо!



# Спортивные соревнования

---



<http://www.hawkeyeinnovations.co.uk>

Отслеживание скоростного  
теннисного мячика для  
определения нарушения правил



# 3D модели и захват движения



*L.A. Noire*, Team Bondi/Rockstar, 2011

Построение трёхмерных моделей  
людей и захват движений для кино  
и видеоигр





# Microsoft Kinect (2010)



Первая потребительская система взаимодействия с компьютером с помощью жестов. При создании использовались суперкомпьютеры



# Поиск изображений по содержанию


Яндекс **картинки**


Поиск Почта Карты Маркет Новости Словари Блоги Видео Картинки ещё a.konushin Помощь Выход


Загруженная картинка x Найти Умеренный поиск


Размер  
Большой  
Средний  
Маленький  
Свой размер  
Обои

Файл  
JPEG  
PNG  
GIF

 Нашлось 4 таких же картинки

 Гамбург 2008 \* Форум Винского  
Гамбург 2008... 800×488  
JPG, 278 КБ  
[http://forum.awd.ru/gallery/image\\_page.php?album\\_id=17164&im...](http://forum.awd.ru/gallery/image_page.php?album_id=17164&im...)

 Гамбург 2008 \* Форум Винского  
Гамбург 2008... 800×488  
JPG, 135 КБ  
[http://forum.awd.ru/gallery/image\\_page.php?album\\_id=17164&im...](http://forum.awd.ru/gallery/image_page.php?album_id=17164&im...)

 Гамбург / Путешествия / Германия / Pinme.ru  
Войдите, для того, чтобы комментировать Пин... 600×400  
JPG, 61 КБ  
<http://pinme.ru/pin/50c31047c00470a8740000bb/>

Поиск картинок, содержащих похожие сцены и объекты



# Мобильные приложения

## kooba

MOBILE IMAGE RECOGNITION?  
TRY IT OUT NOW!!!



[Show another poster](#)

Movie data provided by:



**1. POINT**  
YOUR MOBILE  
PHONE CAMERA TO  
THE MOVIE  
POSTER.

**2. SNAP** A  
PICTURE AND SEND  
IT:

IN SWITZERLAND:  
MMS TO 5555 (OR  
079 394 57 00  
FOR ORANGE  
CUSTOMERS)

IN GERMANY:  
MMS TO 84000

EVERYWHERE:  
EMAIL TO  
M@KOOABA.COM

**3. FIND** ALL  
RELEVANT INFOR-  
MATION ABOUT THE  
MOVIE ON YOUR  
MOBILE PHONE

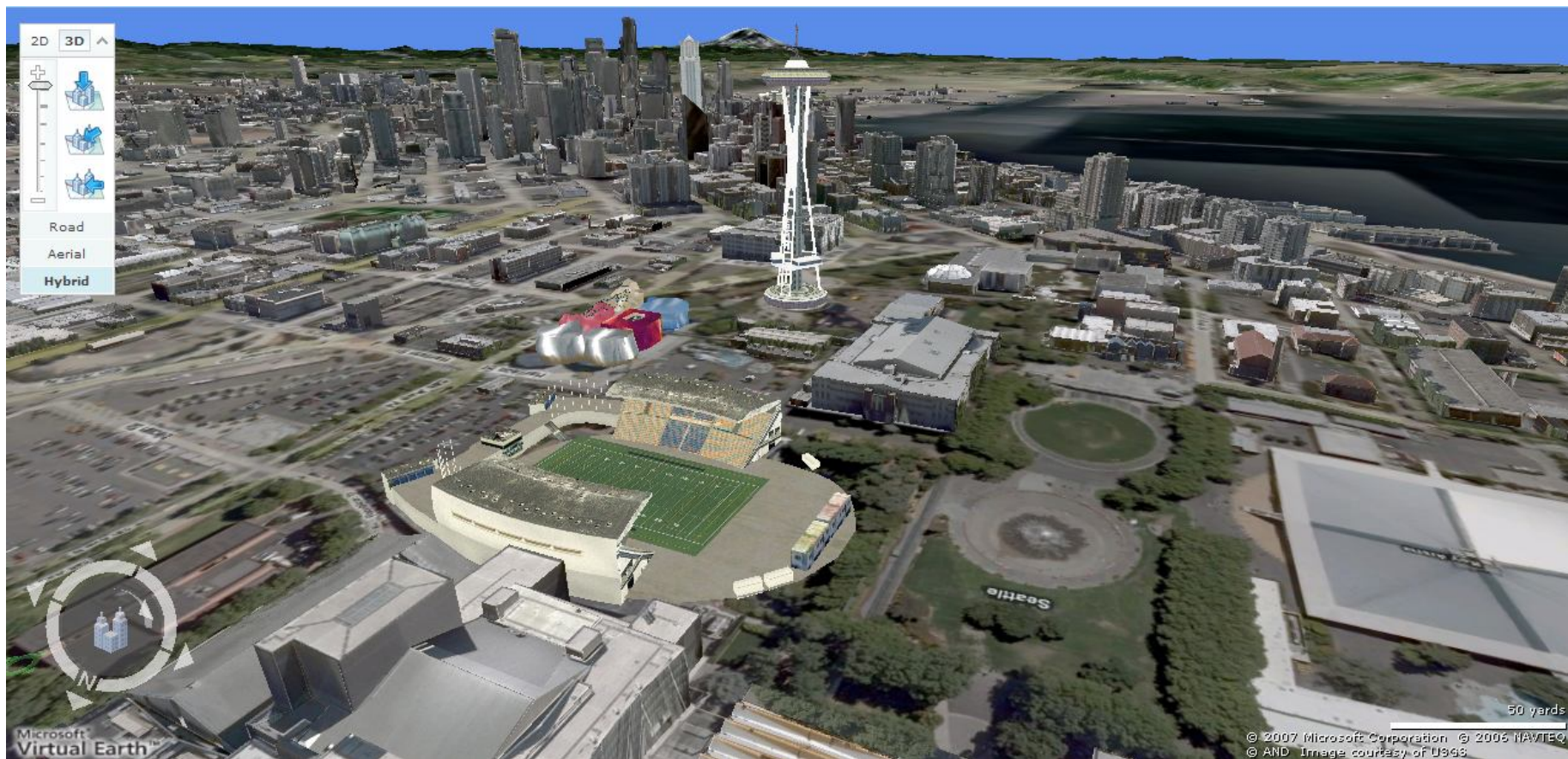


## Дополненная реальность

[www.kooba.com](http://www.kooba.com)



# Карты



Изображение из Microsoft's [Virtual Earth](#) (аналогичные [Google Earth](#))

- Построение карт по аэрофотосъёмке
- Определение положения на местности



## Dubrovnik, Croatia.

# images downloaded: 57,845

Size of largest connected component: 4,619

# points: ~2M

# observations: ~11.3M

Time: 18 hours







# Пример модели по фотографиям

---





# Определение места съёмки



# История по фотографиям

---



## Scene Chronology

Kevin Matzen and Noah Snavely  
Cornell University



ECCV 2014



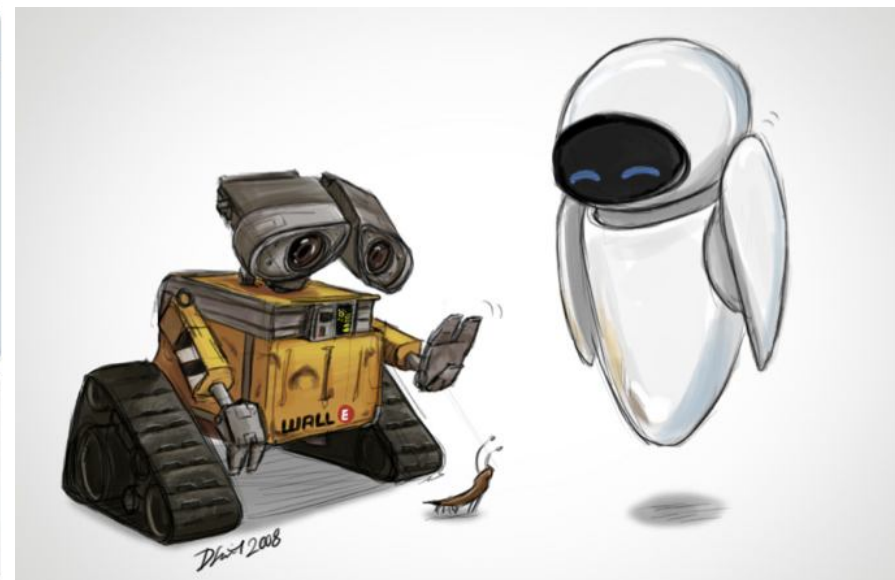
# Робототехника

▶ manufacturer products    consumer products ◀

## Our Vision. Your Safety.

rear looking camera    forward looking camera  
side looking camera

- ▶ **EyeQ** Vision on a Chip  [read more](#)
- ▶ **Vision Applications** Road, Vehicle, Pedestrian Protection and more  [read more](#)
- ▶ **AWS** Advance Warning System  [read more](#)



Зрение автомобилей от  
[Mobileye](#)

Перспектива!

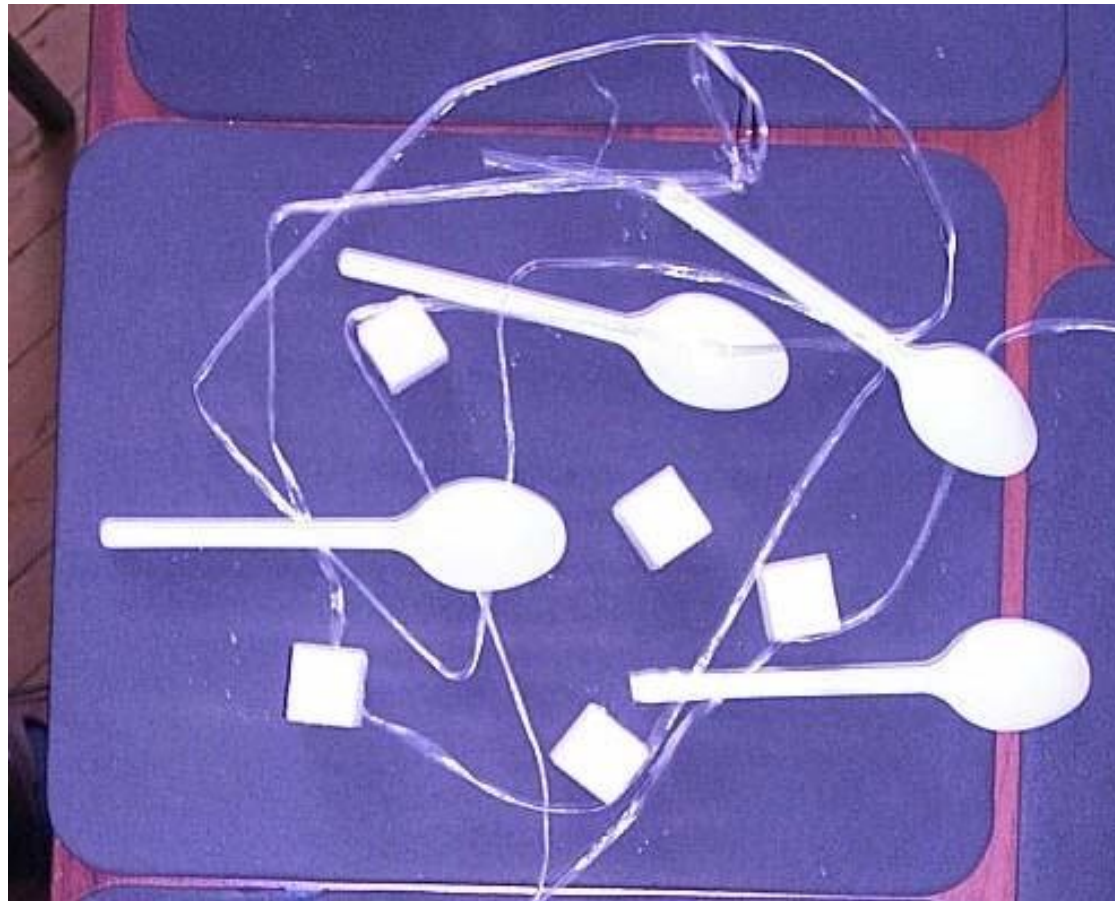


Как реализовать зрение на  
компьютере?



# Простая задача

---

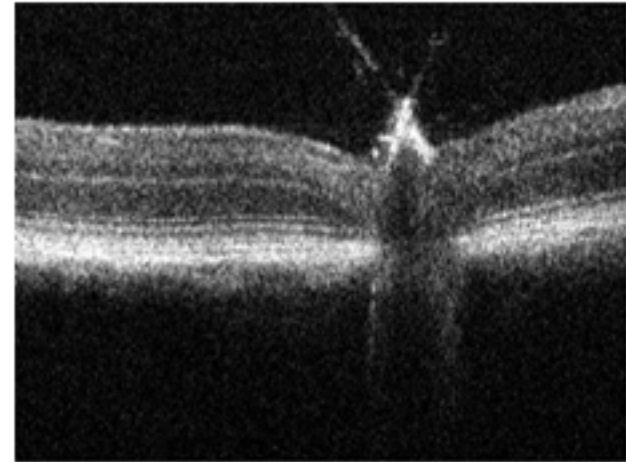


Ложки и сахар



# С чем мы работаем?

---



**Изображение оптическое** – картина, получаемая в результате прохождения через оптическую систему лучей, распространяющихся от объекта, и воспроизводящая его контуры и детали.

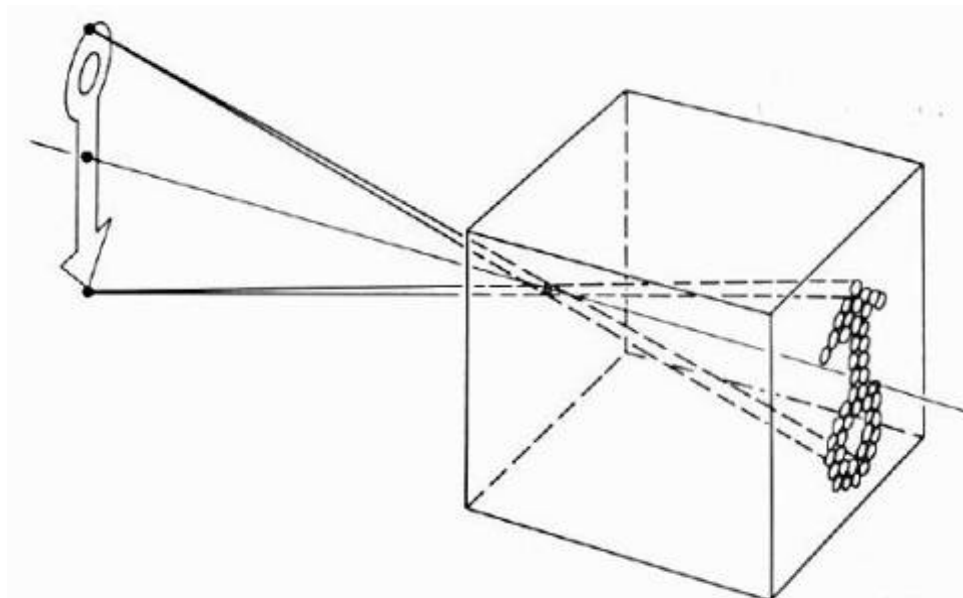
*(Физический энциклопедический словарь.)*

Оптические системы могут быть очень разными!



# Камера-обскура

---

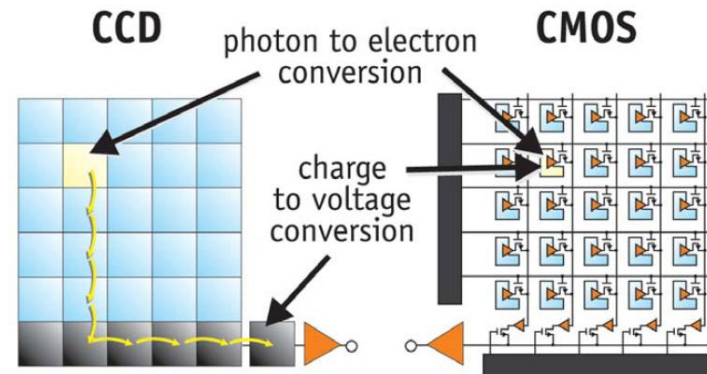


- Основная модель оптической системы получения изображений
- Математическая модель - перспективная проекция
- Пучок лучей проходит через одну точку («центр проекции»)

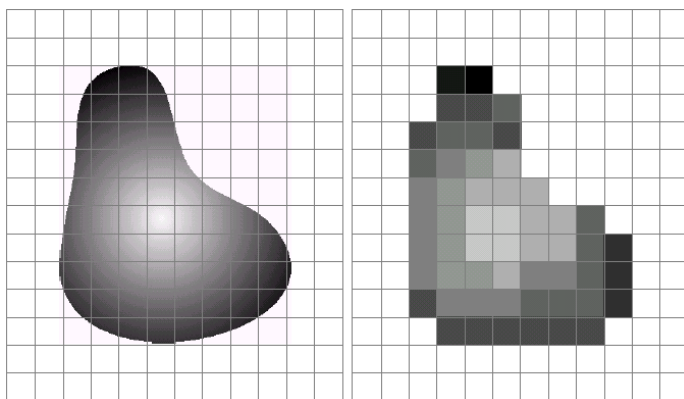




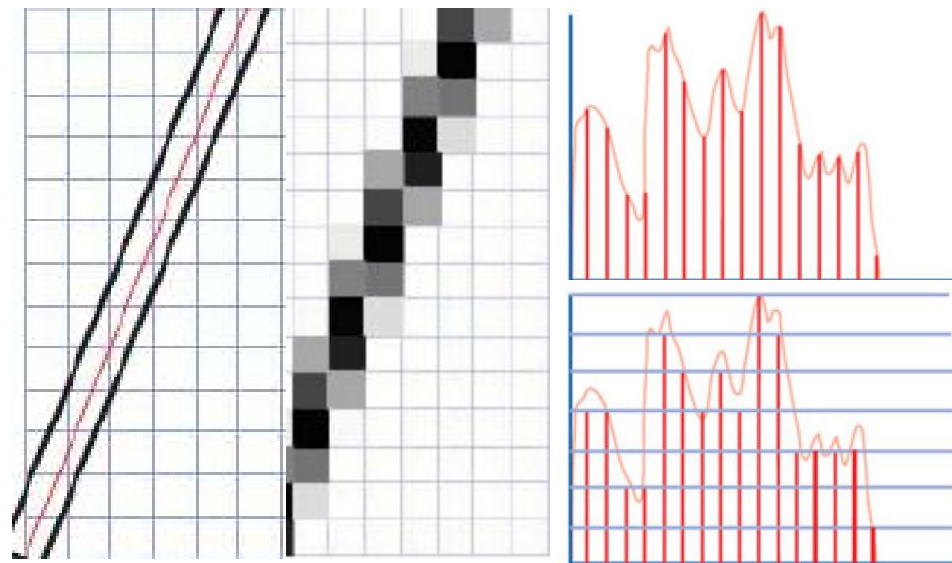
# Цифровая камера - дискретизация



CCDs move photogenerated charge from pixel to pixel and convert it to voltage at an output node. CMOS imagers convert charge to voltage inside each pixel.



**FIGURE 2.17** (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.



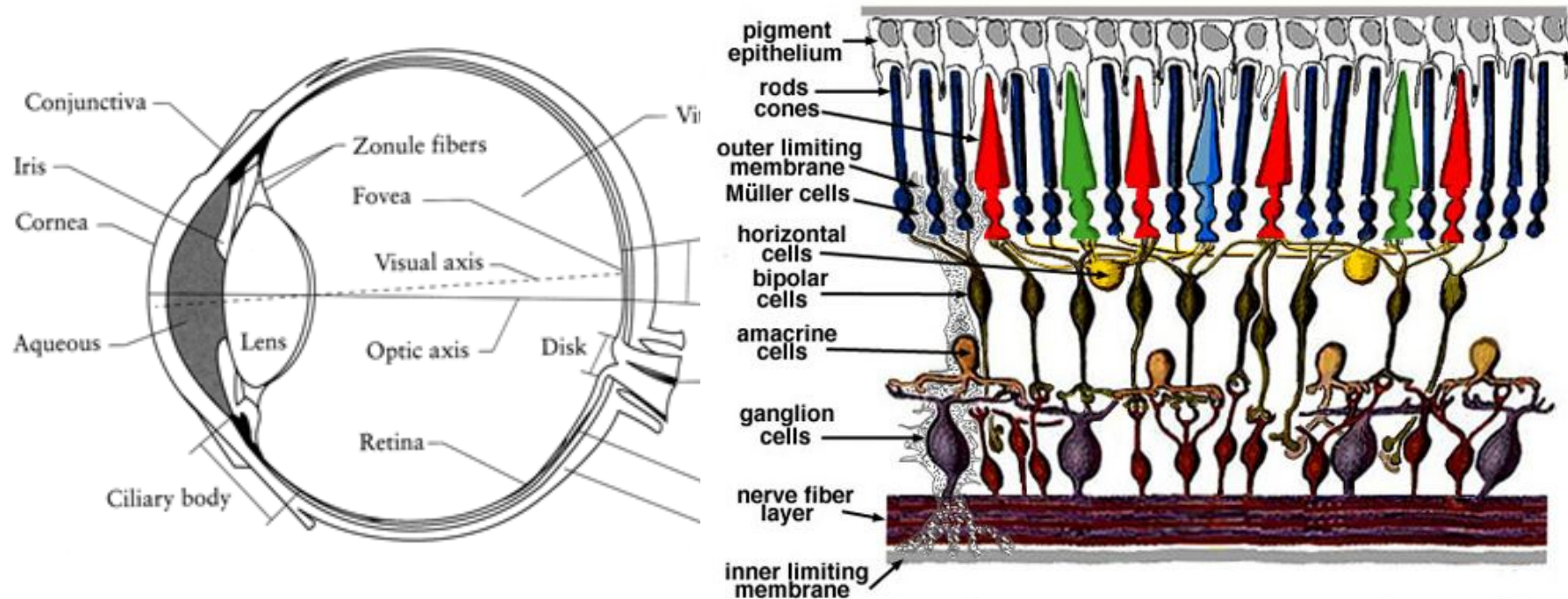
По пространству

Цифровое изображение – функция яркости, заданная на прямоугольной сетке

По яркости



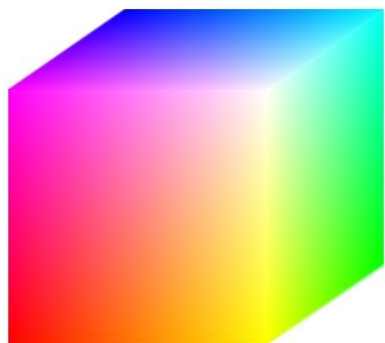
# Человеческий глаз как камера



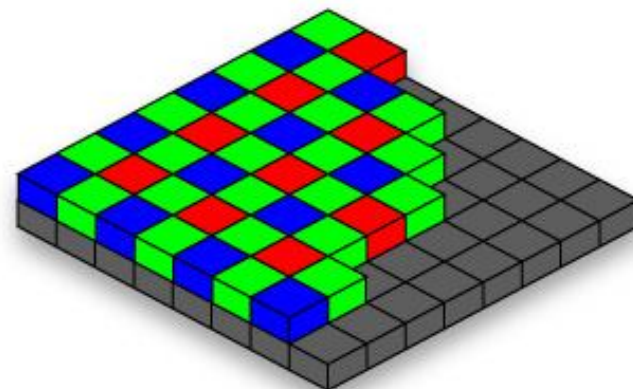
- **Хрусталик** – «линза», меняющая форму под действием мышц
- **Зрачок** - отверстие (апертура), диаметр управляется радужкой
- **Сетчатка** – «матрица» фоторецепторов двух типов – палочки и колбочки (красные, зелёные, синие)



# Цветное цифровое изображение



- $p_1 = 645.2 \text{ nm}$
- $p_2 = 525.3 \text{ nm}$
- $p_3 = 444.4 \text{ nm}$

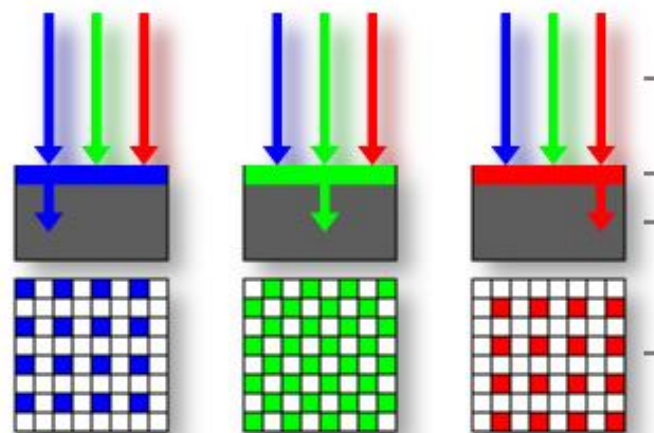


Модель RGB  
(Красный, Зелёный, Синий)



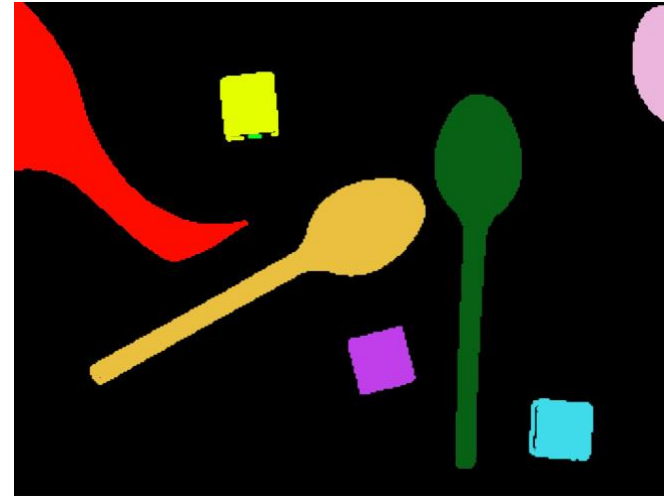
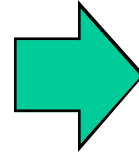
Демозаикинг (оценка пропущенных значений цвета)

Байеровский шаблон





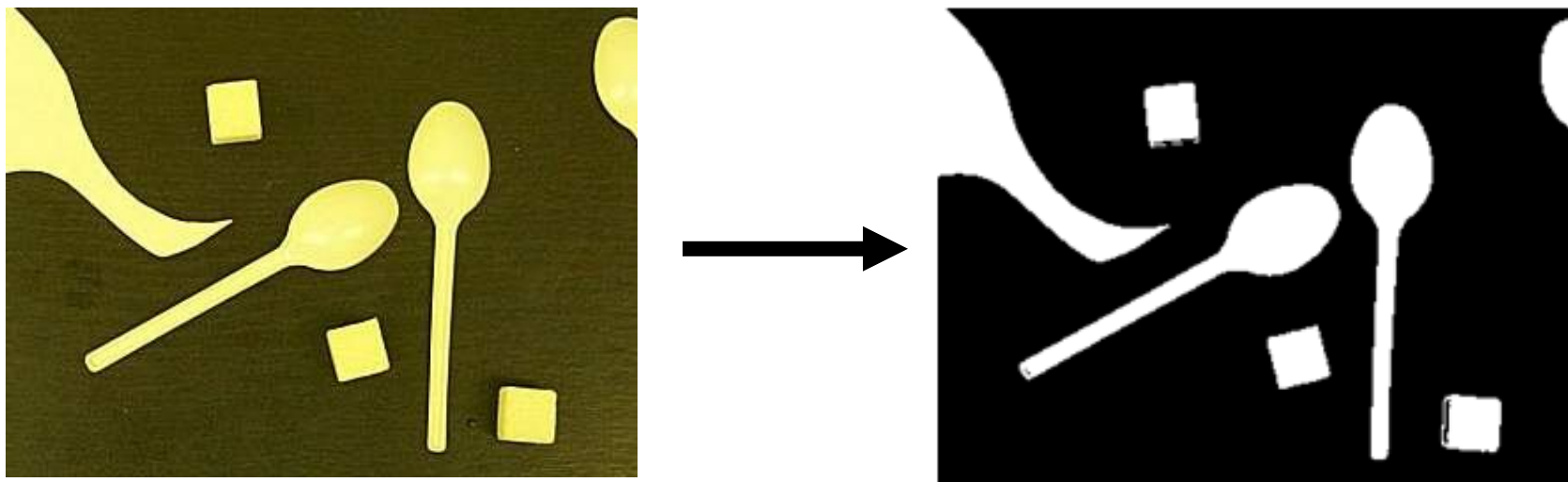
# Наш пример



- Сегментация - это способ разделения сцены на «куски», с которыми проще работать
- Как мы будем записывать результат сегментации?
- Сделаем карту разметки – изображение, в каждом пикселе которого номер сегмента, которому принадлежит этот пиксель
- Визуализировать удобно каждый сегмент своим цветом



# Пороговая фильтрация



Будем делать так:

- Пиксели, которых выше/ниже некоторого порога, заданного «извне», помечаются 1
- Ниже порога помечаются 0
- Бинарное изображение – пиксели которого могут принимать только значения 0 и 1
- Бинаризация - построение бинарного изображения по полутоновому / цветному

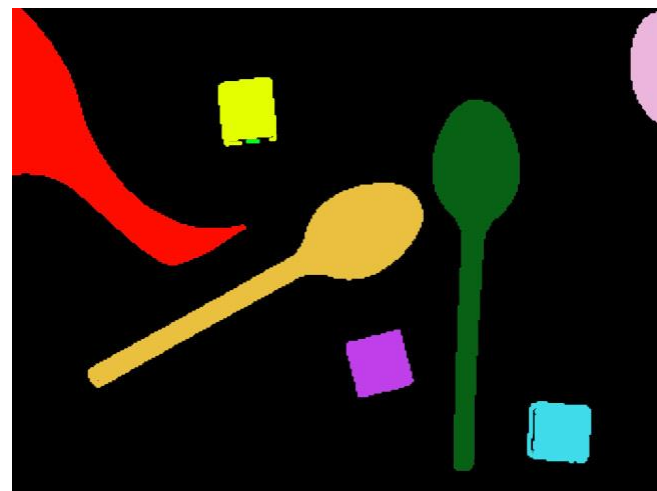


# Что дальше?

---



Получили бинарное изображение



Нужна карта разметки



# Выделение связанных областей

Определение связанной области:

Множество пикселей, у каждого пикселя которого есть хотя бы один сосед, принадлежащий данному множеству.



Соседи пикселей:

	1	
2	*	3
	4	

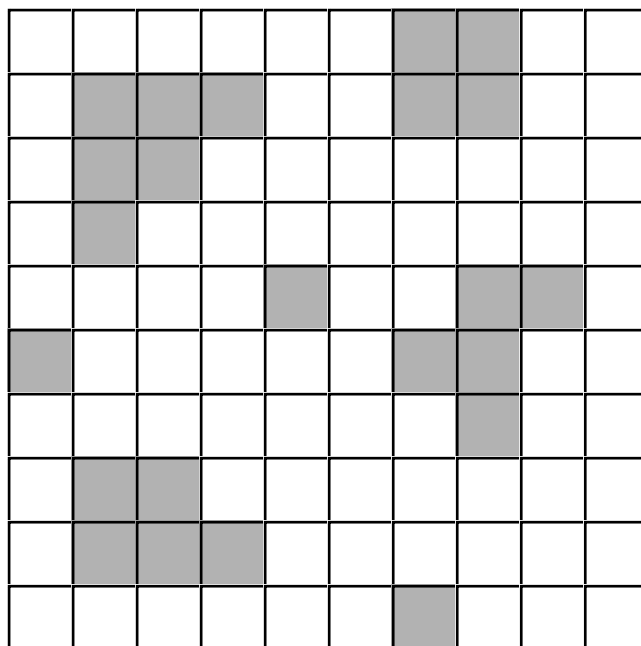
4-связность

1	2	3
4	*	5
6	7	8

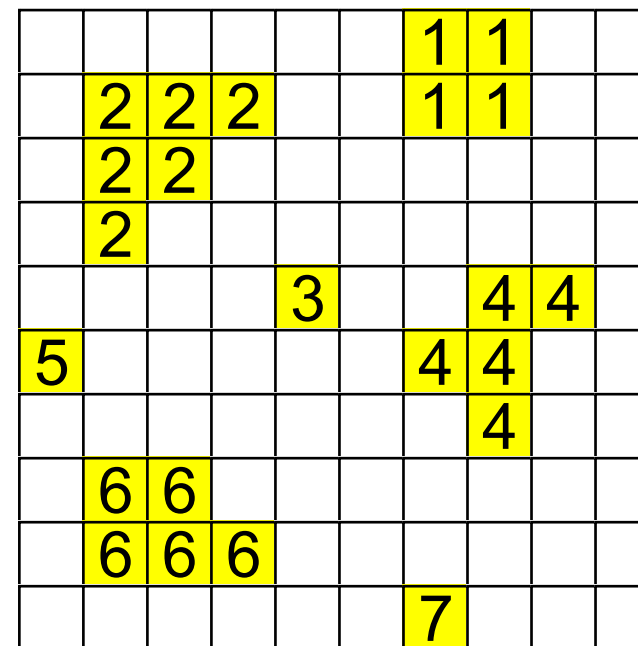
8-связность



# Разметка связанных областей



Бинарное изображение



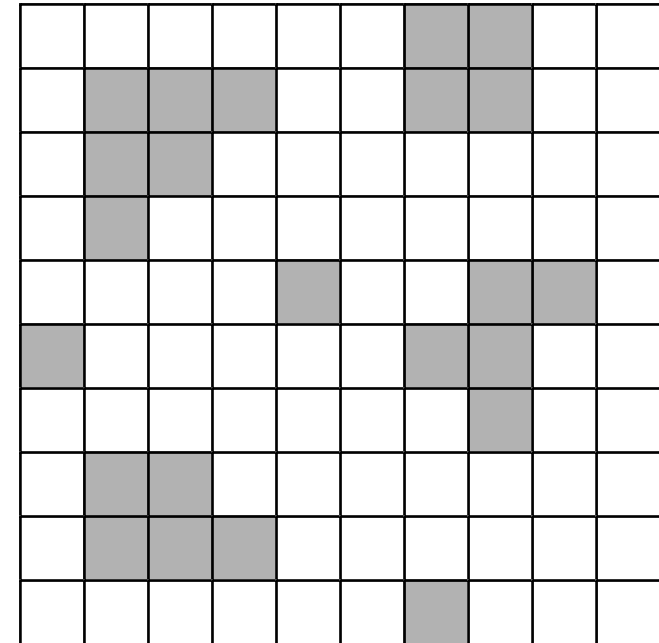
Размеченное изображение





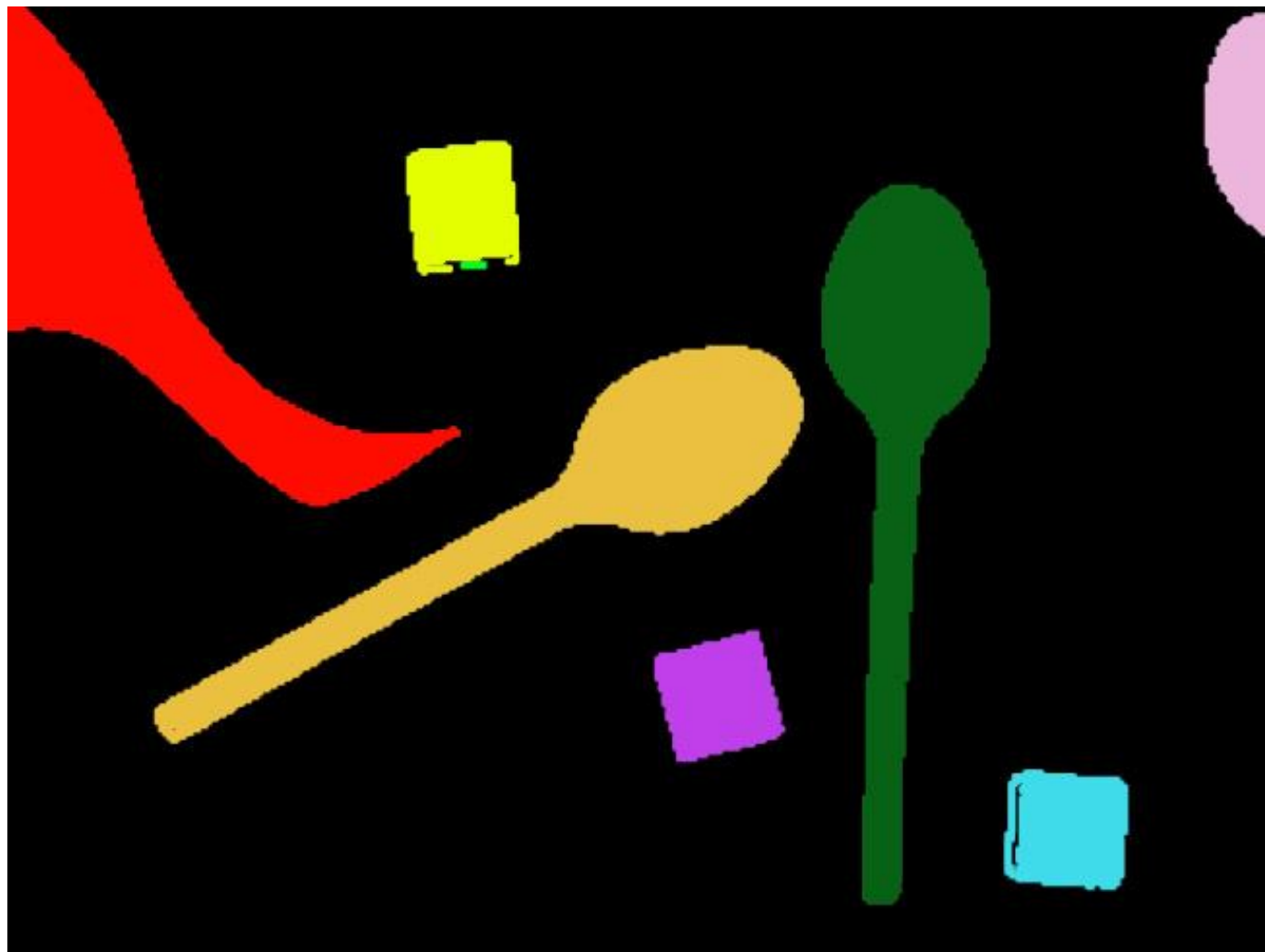
# Рекурсивный алгоритм

```
Labeling(img[], labels[])
{
    // labels должна быть обнужена
    L = 1;
    for(y = 0; y < H; y++)
        for(x = 0; x < W; x++)
        {
            Fill(img, labels, x, y, L++);
        }
}
Fill(img[], labels[], x, y, L)
{
    if( (labels[x][y] == 0) && (img[x][y] == 1) )
    {
        labels[x][y] = L;
        if( x > 0 )
            Fill(img, labels, x - 1, y, L);
        if( x < W - 1 )
            Fill(img, labels, x + 1, y, L);
        if( y > 0 )
            Fill(img, labels, x, y - 1, L);
        if( y < H - 1 )
            Fill(img, labels, x, y + 1, L);
    }
}
```



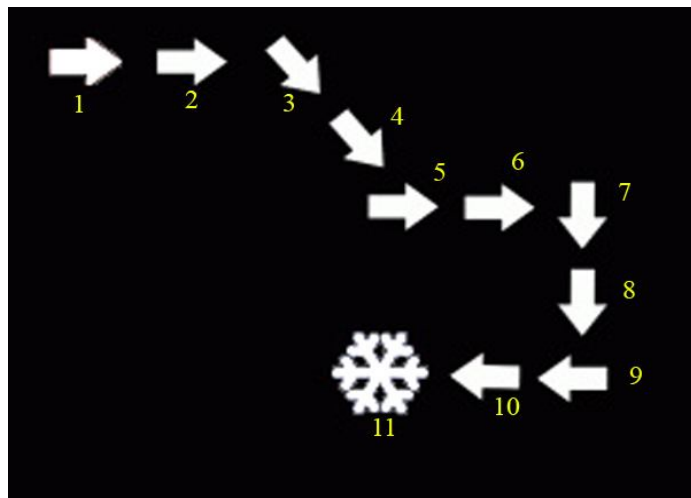


# Выделенные связанные компоненты





# Анализ выделенных областей



Для анализа требуется вычислить некоторые числовые характеристики (признаки) областей:

- геометрические признаки (формы)
- фотометрические признаки (цвет)



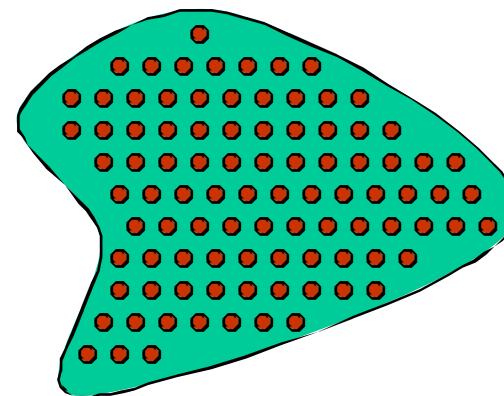
На основе этих характеристик можно классифицировать получаемые области



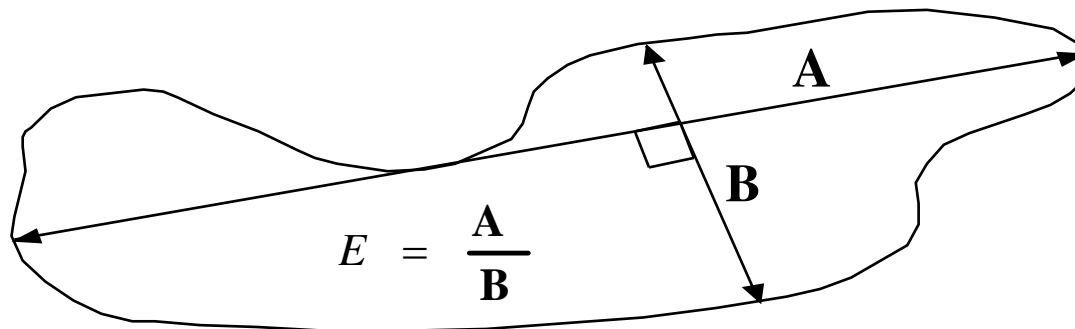
# Мы воспользуемся

- Площадь – количество пикселей в области;

$$A = \sum_{x=0}^m \sum_{y=0}^n I(x, y)$$



- Удлиненность (эксцентриситет) – отношение длины объекта к его ширине

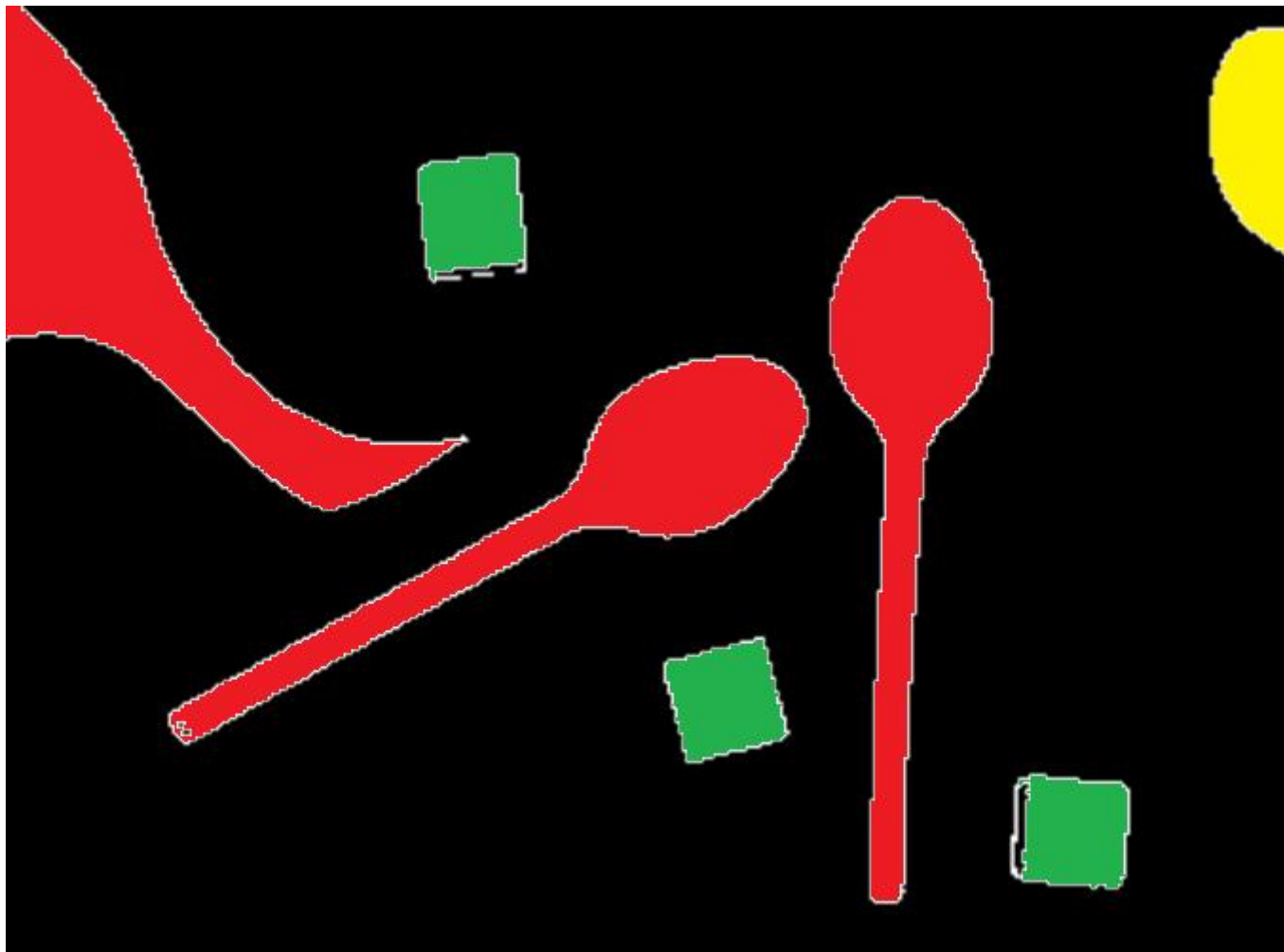






# Задача решена!

---





# Гораздо более сложная задача!



Распознавание человека по лицу



# Классификация

---

Задача классификации – определить для объекта  $X$  его «класс»  $Y$  из заданного конечного набора классов



Объект  $X$

Что на картинке?

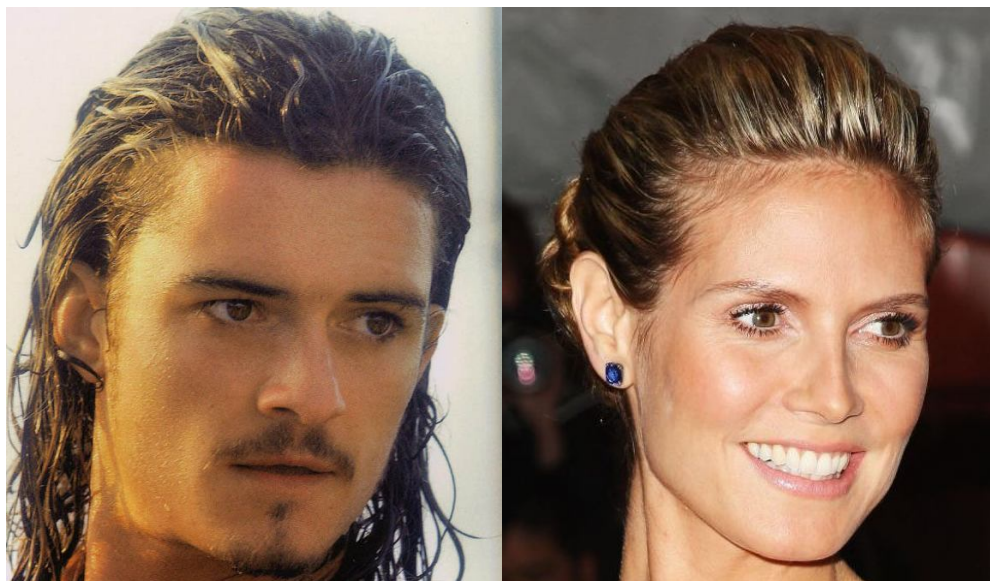
- Человек ( $y=1$ )
- Животное ( $y=2$ )
- Яблоко ( $y=3$ )
- Ягоды ( $y=4$ )
- Собака ( $y=5$ )
- ....

Модель (или классификатор)  $f(x) = y$  – функция, предсказывающая по  $x$  соответствующий класс

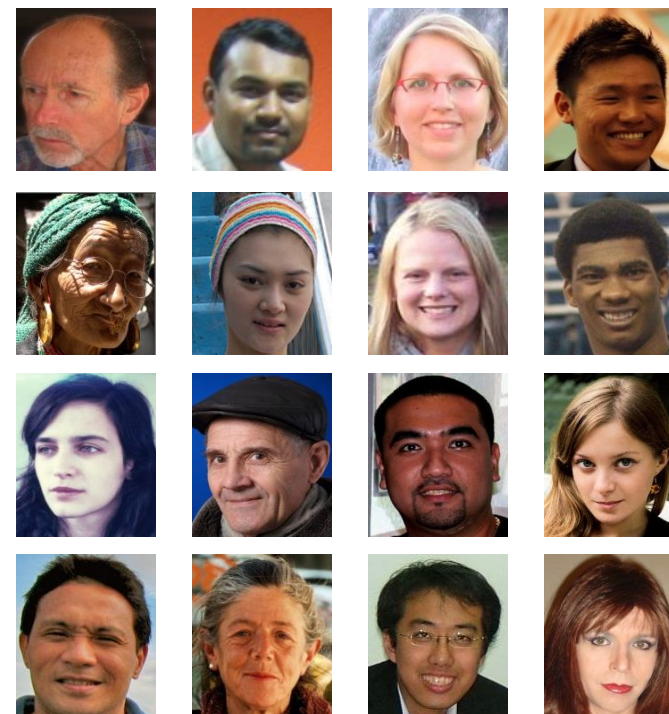




# Примеры для лиц



Один и тот же человек ( $y=1$ ),  
или разные ( $y=0$ )?  
(бинарная классификация)  
**«Верификация»**



Кто из списка?  
**«Идентификация»**



# Выделение объектов на изображении

Задача поиска объектов сводится к классификации



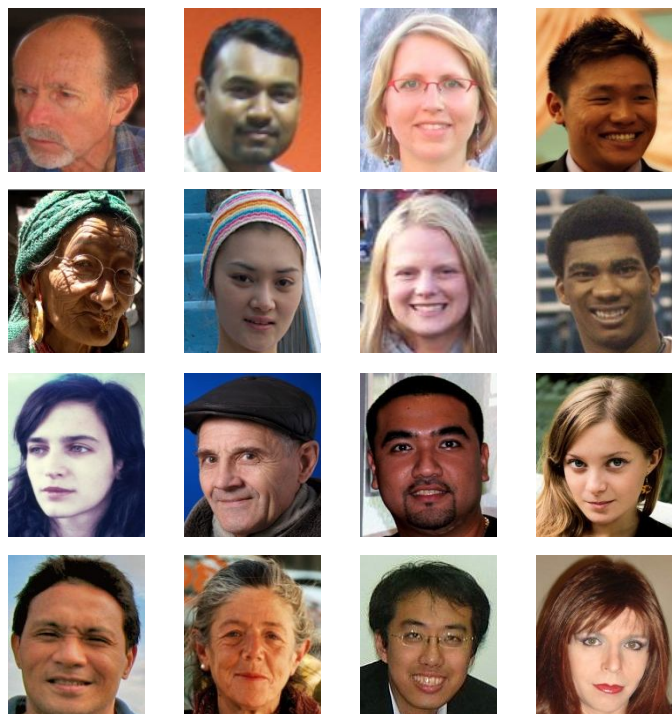
Разделили изображение на фрагменты, которые классифицируем независимо друг от друга

- «Скользящее окно» – сканирование окном изображения
- Каждый фрагмент отдельно от других классифицируем «объект» / «не объект»



# Что такое «лицо»?

- Соберём много примеров «лиц» и «не лиц»



Лица

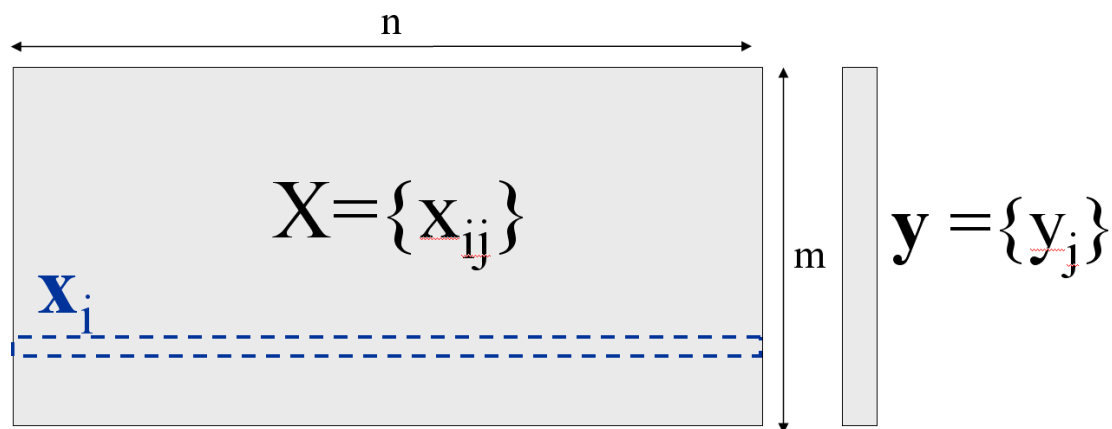


Не лица



# Построение классификатора

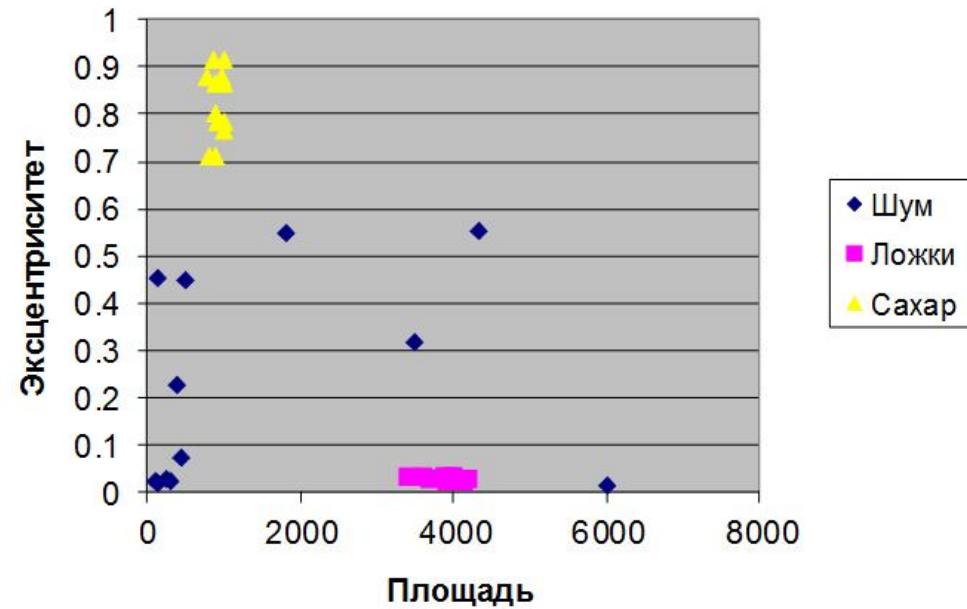
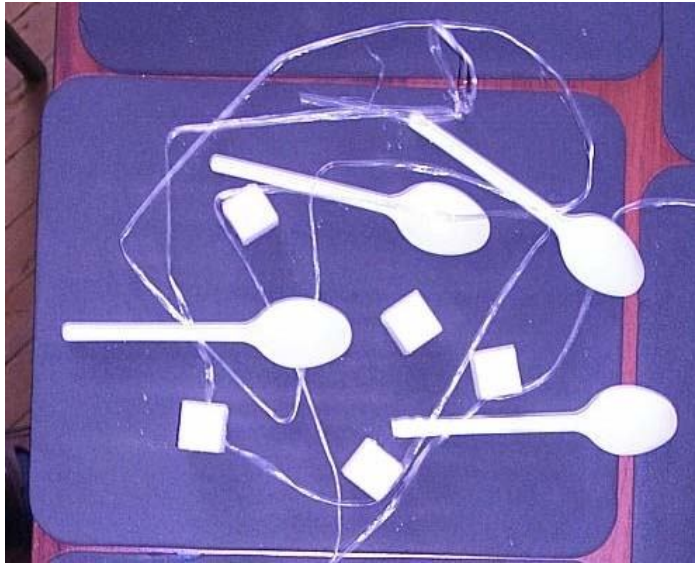
- Есть выборка данных  $X$  для которых известны  $Y$
- Каждый объект из  $X$  с номером  $j$  можно описать вектором признаков  $x_j$  (набором характеристик)
- Всё множество известных наблюдений (конечное) можно записать в следующем виде (обучающая выборка):



Нужно построить функцию  $f(x) = y$



# Эвристический подход

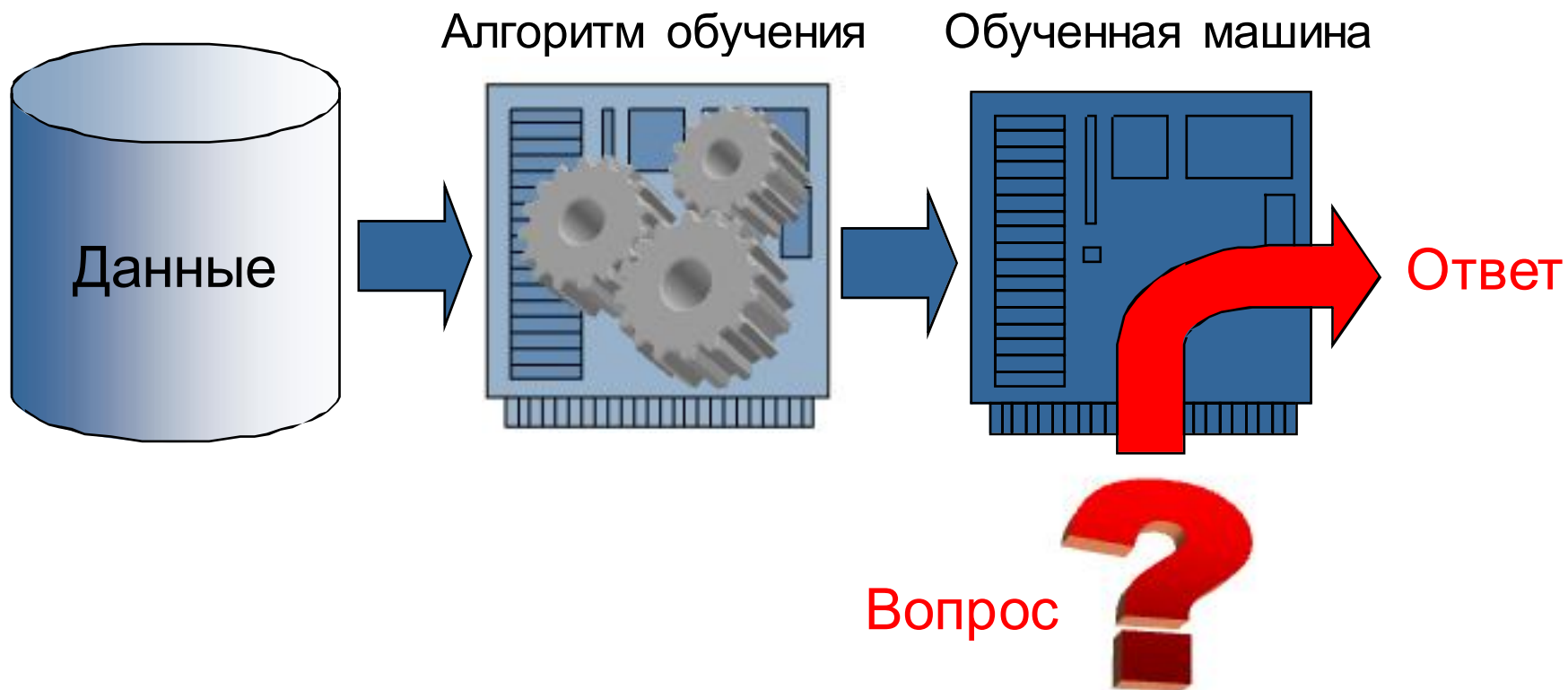


Обычно слишком много признаков, которые по отдельности невозможно интерпретировать



# Как должно быть в идеале

---



Машинное обучение



# Машинное обучение

---

Развитие методов компьютерного зрения неразрывно связано с развитием методов машинного обучения

- Нейронные сети - Перспептрон (Розенблатт, 1958), когнитрон (Фукушима, 1975)
- Нейронные сети - обратное распространение ошибки (1980е)
- Метод опорных векторов (1990е)
- Бустинг (конец 1990х)
- Рандомизированный решающий лес (начало 2000х)
- Нейронные сети - «глубинное обучение» (2006 и далее)



# Прежде чем решать задачу...

---

- Нужно определиться с критериями решения
- Как определить, насколько хорошо мы научились решать задачу?





# Labeled Faces in the Wild

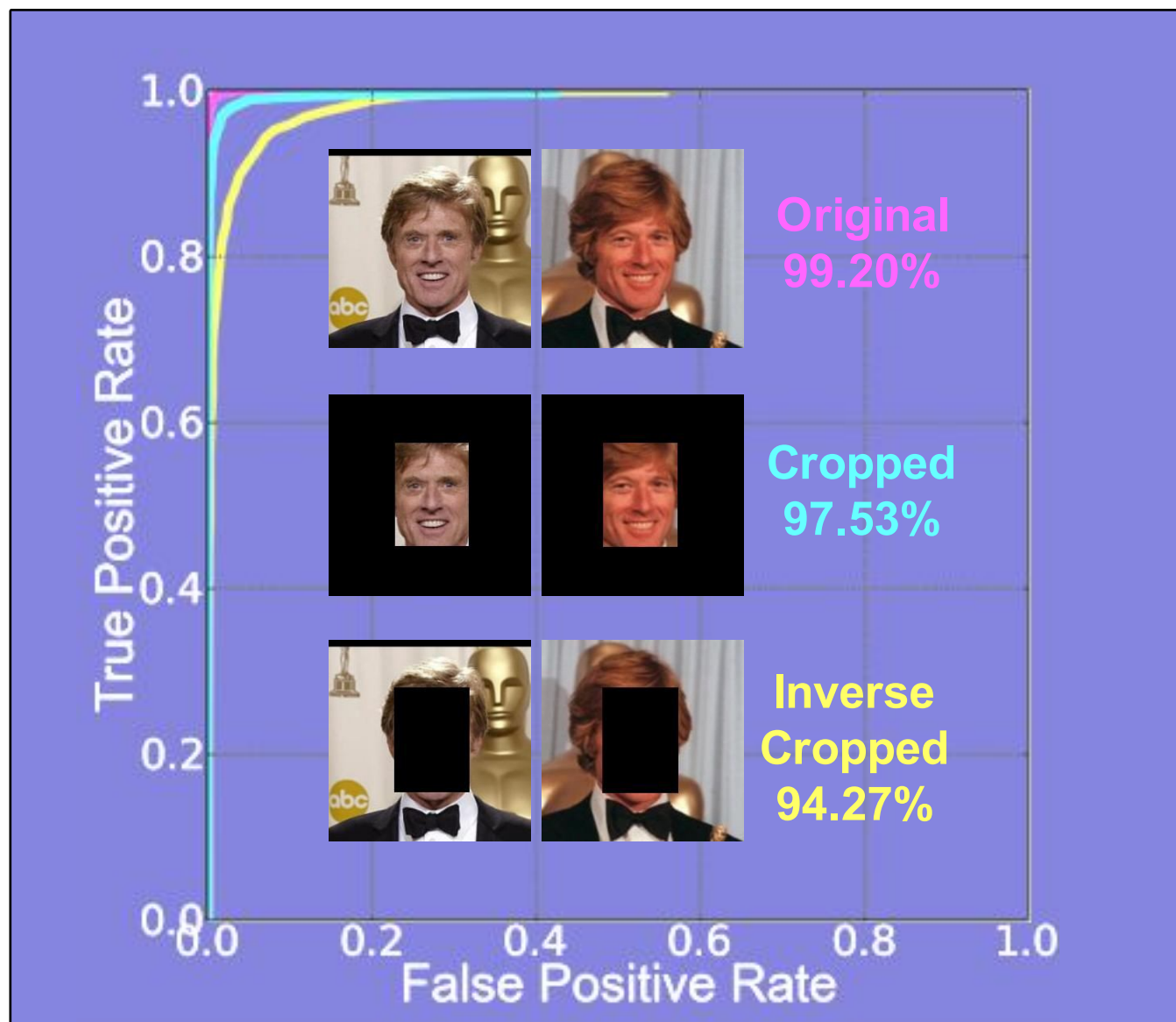
- Соберём большую коллекцию изображений из интернета, разметив её вручную
- 5749 человек, 12К изображений, 1680 человек по 2 и более фотографии, остальные – по одной
- Фотографии разрешения 250\*250, JPEG
- Неконтролируемые условия, очень разные фоны, позы, разное время съёмки
- Выберем некоторое количество пар одинаковых и разных людей. Доля правильно определённых будет показателем качества
- На этой коллекции до текущего времени проводились сравнения алгоритмов распознавания лиц



Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller.  
**Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.** *University of Massachusetts, Amherst, Technical Report 07-49, 2007.*

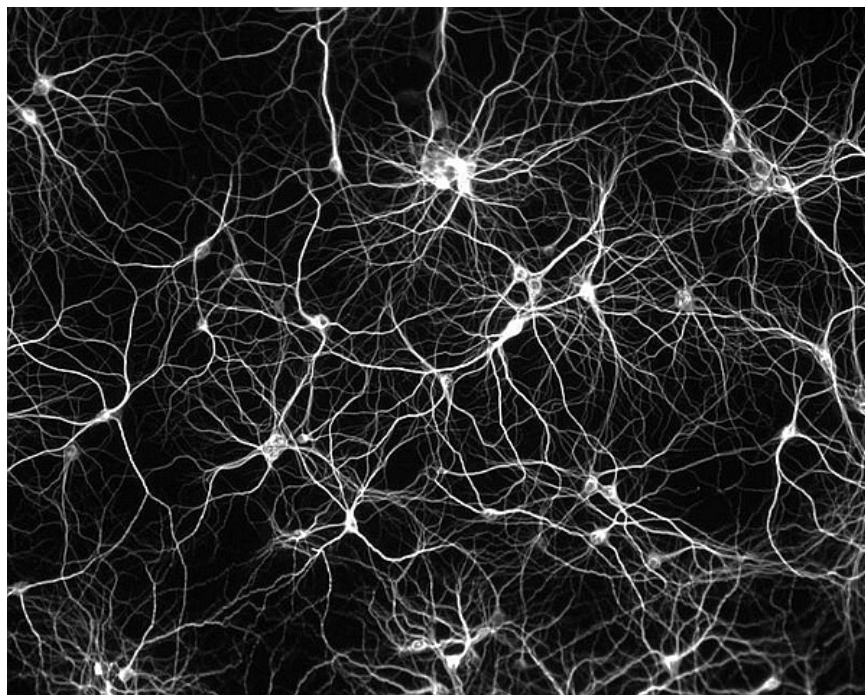


# Верификация человеком на LFW



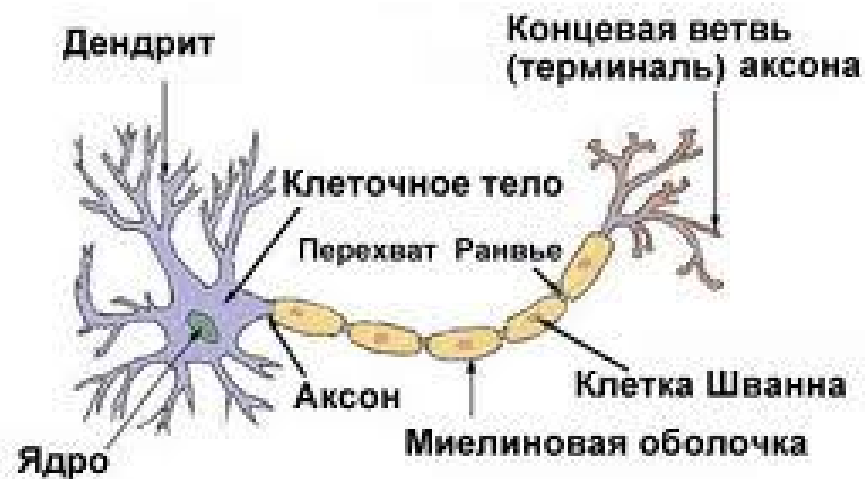


# Структура мозга человека



Нейросеть

## Типичная структура нейрона



Отдельный нейрон

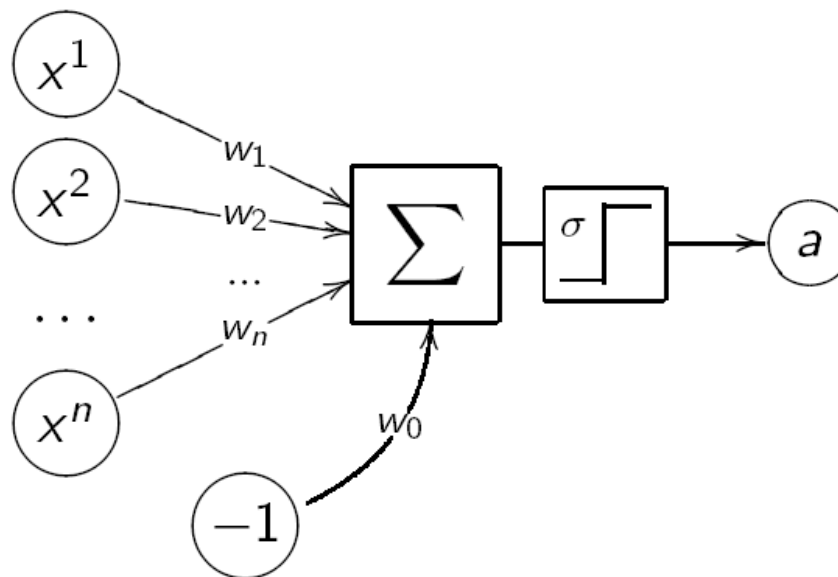


# Математическая модель нейрона

Линейная модель нейрона МакКаллока-Питтса [1943]:

$$a(x, w) = \sigma(\langle w, x \rangle) = \sigma\left(\sum_{j=1}^n w_j f_j(x) - w_0\right),$$

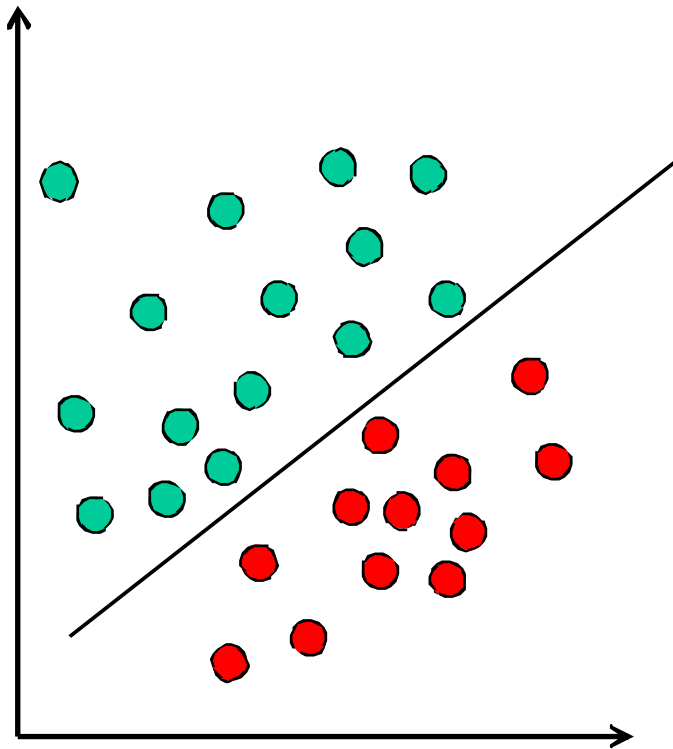
где  $\sigma(s)$  — функция активации (в частности, sign).





# Нейрон = линейный классификатор

---



$$a(x, w) = \text{sign} \left( \sum_{j=1}^n w_j f_j(x) - w_0 \right)$$

$$x_i \text{ положительные} : \quad x_i \cdot w - w_0 \geq 0$$

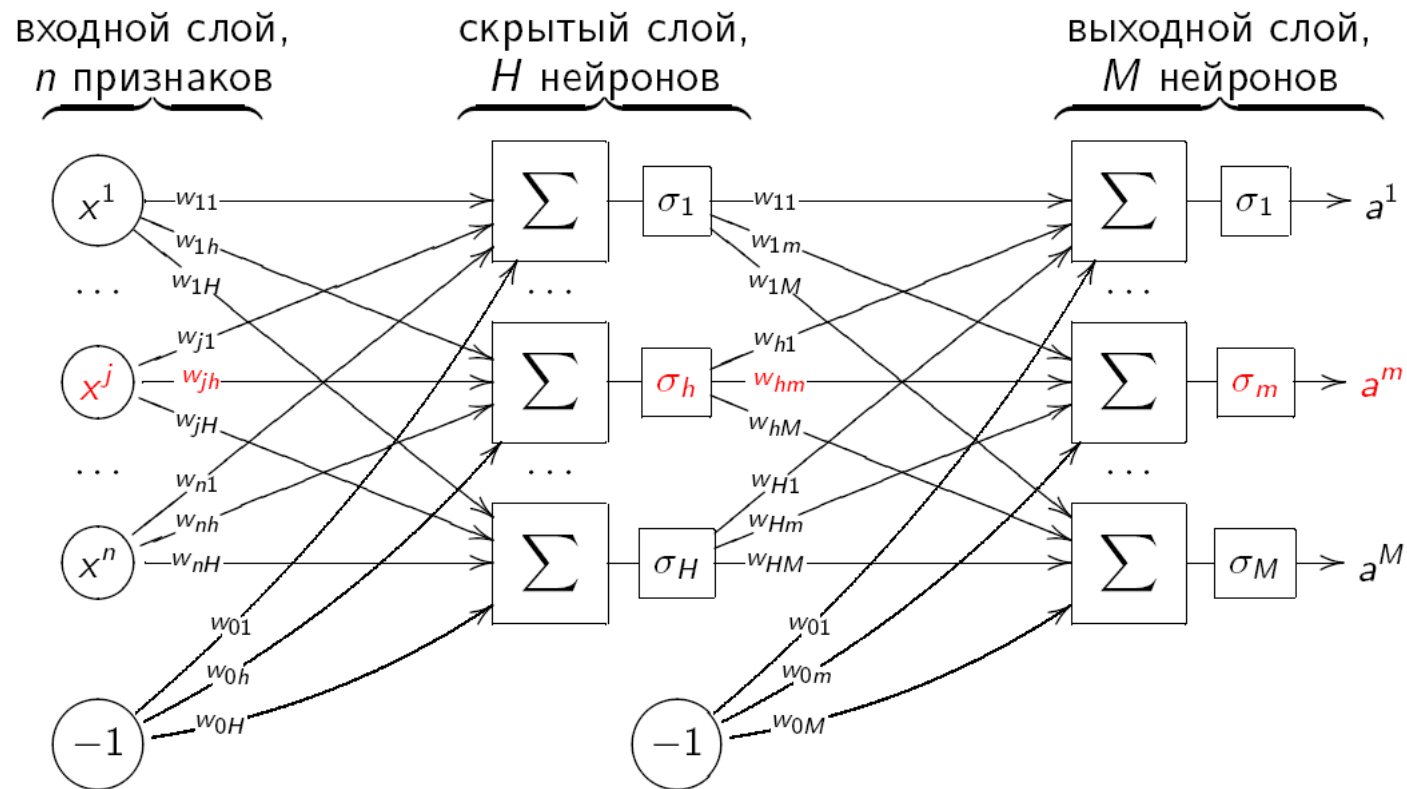
$$x_i \text{ отрицательные} : \quad x_i \cdot w - w_0 < 0$$

- Нейрон задает линейную поверхность (гиперплоскость) разбивающую пространство признаков на две области
- «Обучение» нейрона = настройка весов
- Есть несколько методов



# Многослойная нейросеть

Пусть для общности  $Y = \mathbb{R}^M$ , для простоты слоёв только два.

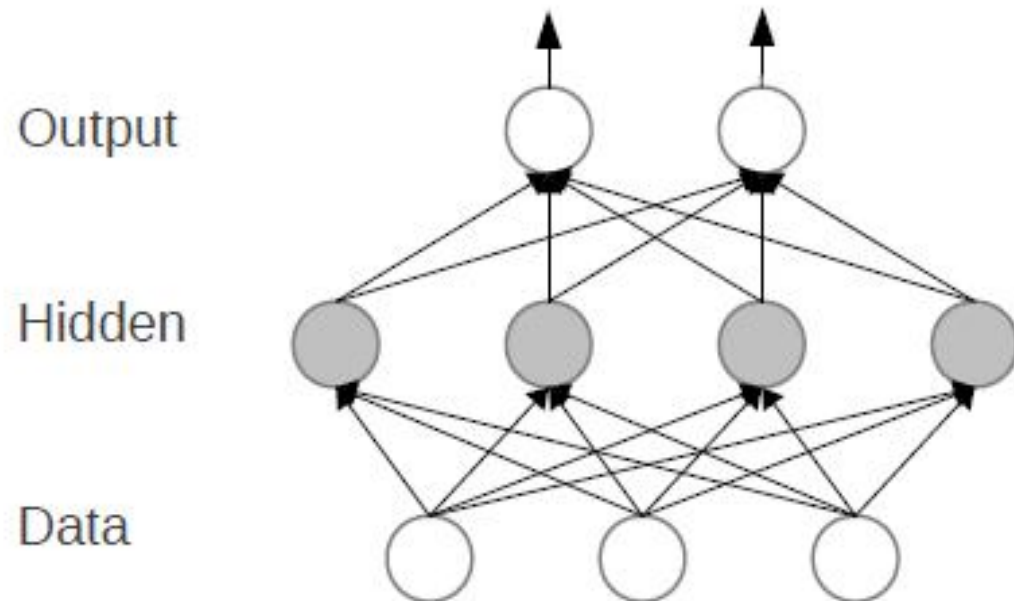


- С помощью многослойных сетей можно реализовать любую непрерывную функцию.
- $Y = f_n(f_{n-1} \dots (f_1(x)) \dots)$



# Обратное распространение ошибки

---



Нейросеть  
вычисляет  
дифференцируемую  
функцию от своих  
ВХОДОВ

- «Метод стохастического градиентного спуска»
- Подадим на вход пример, посчитаем его, и вычислим ошибку
- Рассчитаем, как нужно изменить веса каждого нейрона («градиент») чтобы ошибку уменьшить
- Изменим (уточним) веса
- Подадим новый пример....



# История

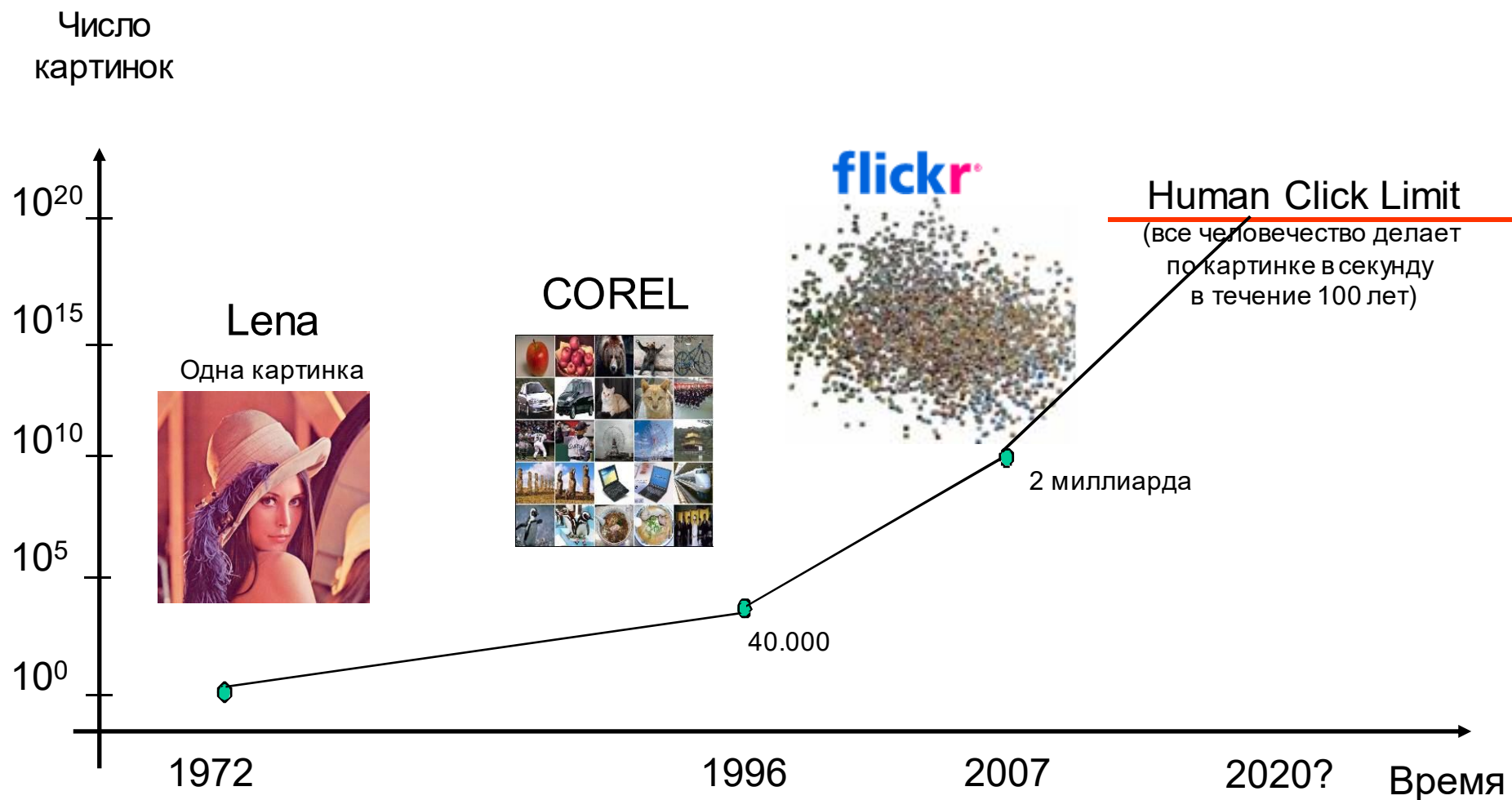
---

- Метод обратного распространения ошибки оказался очень эффективным, но неустойчивым
- Требовалось изрядное умение, чтобы заставить нейросеть обучаться
- В 2000е поэтому нейросети стали реже использовать в пользу других методов машинного обучения





# «Интернет-бум» + «Закон Мура»





14М изображений  
Будет 1000 на  
каждую категорию

Конкурс Large-  
scale visual  
recognition (LSVR)  
– 1000 классов,  
1200 изображений  
на класс

## Vegetable, veggie, veg

Edible seeds or roots or stems or leaves or bulbs or tubers or nonsweet fruits of any of numerous herbaceous plant

1369 pictures

73.58% Popularity Percentage



Numbers in brackets are the number of synonyms in the subtree.

- ImageNet 2011 Winter Release (17)
  - animal, animate being, beast, br...
  - sport, athletics (165)
  - fabric, cloth, material, textile (2)
  - instrumentality, instrumentation
  - appliance (50)
  - structure, construction (1238)
  - fruit (308)
  - flower (461)
  - fungus (302)
  - tree (992)
  - vegetable, veggies, veg (175)**
    - fennel, Florence fennel, froot
    - cucumber, cuke (1)
    - squash (16)
    - cruciferous vegetable (18)
    - peppert, rhubarb (0)
    - root vegetable (25)
    - solanaceous vegetable (25)
    - greens, green, leafy vegetabl
    - potherb (0)
    - legume (37)
    - raw vegetable, rabbit food (0)
    - artichoke, globe artichoke (0)
    - artichoke heart (0)
    - asparagus (0)
    - plantain (0)
    - truffle, earthnut (0)
    - pumpkin (0)
    - mushroom (0)

TreeMap Visualization | Images of the Synset | Downloads

ImageNet 2011 Winter Release - Vegetable, veggie, veg

Legume	Greens	Root	Cruciferous
Squash	Artichoke	Julienne	Carrots
Potterb	Leek		
Solanaceous	Plantain	Raw	Artichoke
	Gumbo	Celery	
Truffle	Fennel	Pumpkin	Cucumber
	Mushroom		
Bamboo	Phacelia	Asparagus	Onion

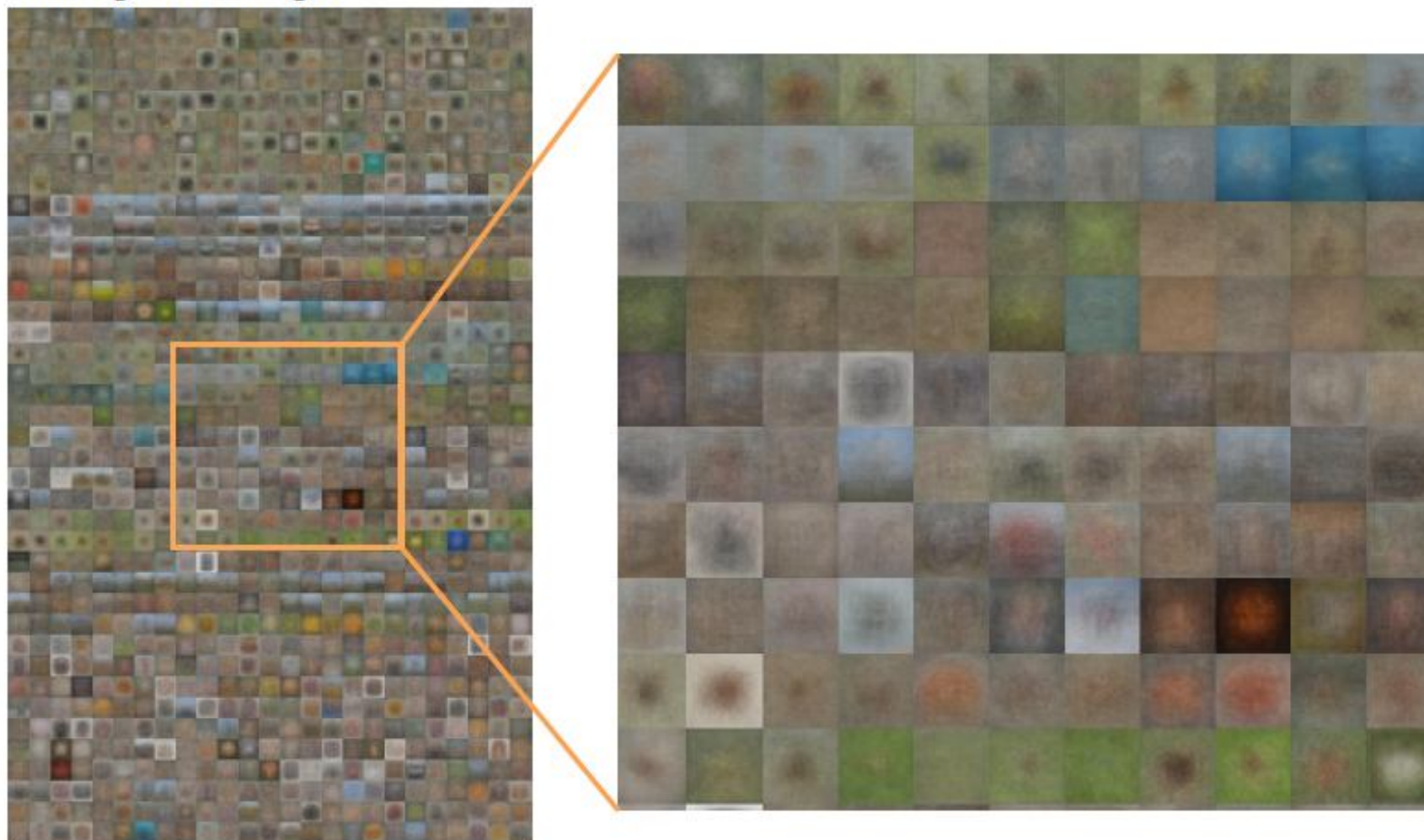
<http://www.image-net.org>



# Изображения в среднем

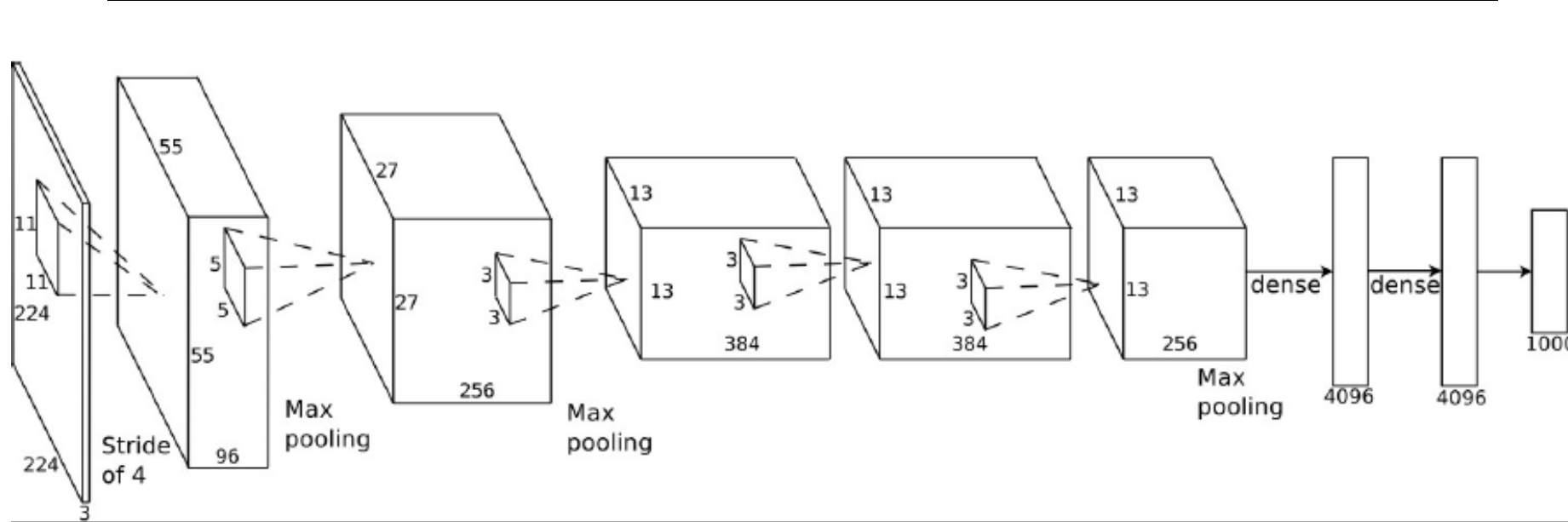
---

Average Test images





# Победитель – нейросеть «SuperVision»



- 650,000 нейронов
- 60,000,000 параметров
- 630,000,000 связей
- 1 компьютер, 2 GPU по 2Gb, 5GB Ram, 27Gb HDD, 1 неделя на обучение

Krizhevsky A., Sutskever I., Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks // NIPS 2012: Neural Information Processing Systems. Lake Tahoe, Nevada.



# Примеры работы



**mite**

**container ship**

**motor scooter**

**leopard**

	mite
	black widow
	cockroach
	tick
	starfish

	container ship
	lifeboat
	amphibian
	fireboat
	drilling platform

	motor scooter
	go-kart
	moped
	bumper car
	golfcart

	leopard
	jaguar
	cheetah
	snow leopard
	Egyptian cat



**grille**



**mushroom**



**cherry**



**Madagascar cat**

	convertible
	grille
	pickup
	beach wagon
	fire engine

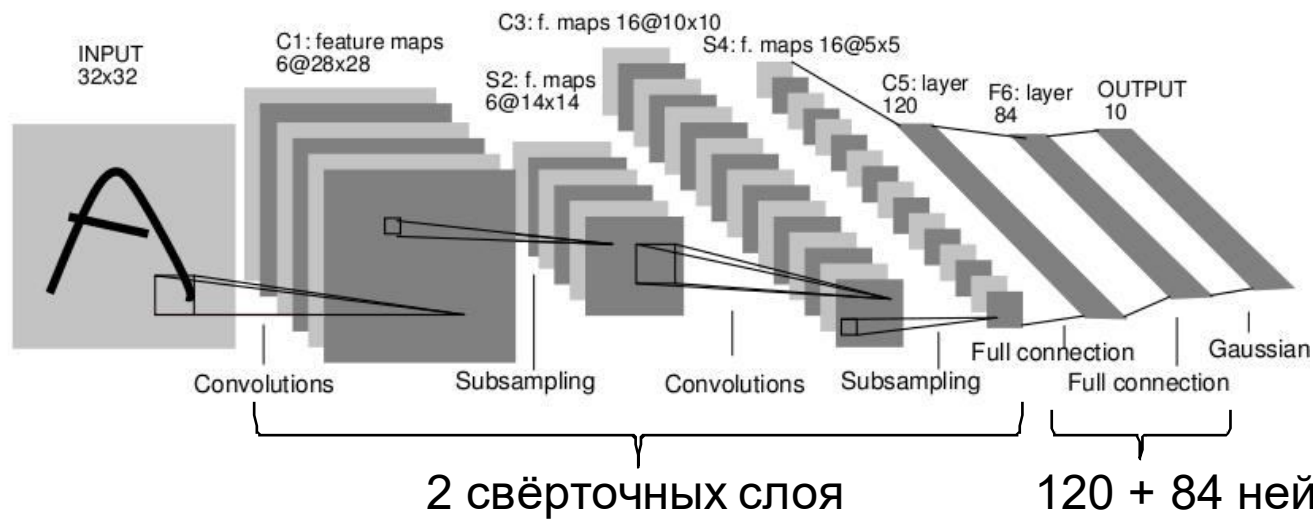
	agaric
	mushroom
	jelly fungus
	gill fungus
	dead-man's-fingers

	dalmatian
	grape
	elderberry
	ffordshire bullterrier
	currant

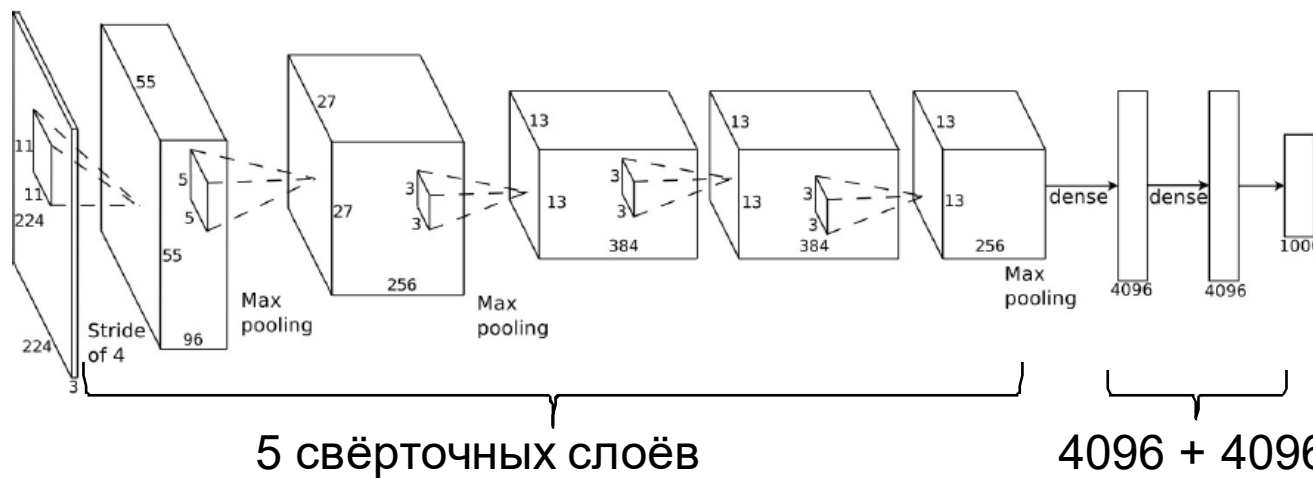
	squirrel monkey
	spider monkey
	titi
	indri
	howler monkey



# 1998 - 2012



1998



2012



## Что ещё поменялось?

---

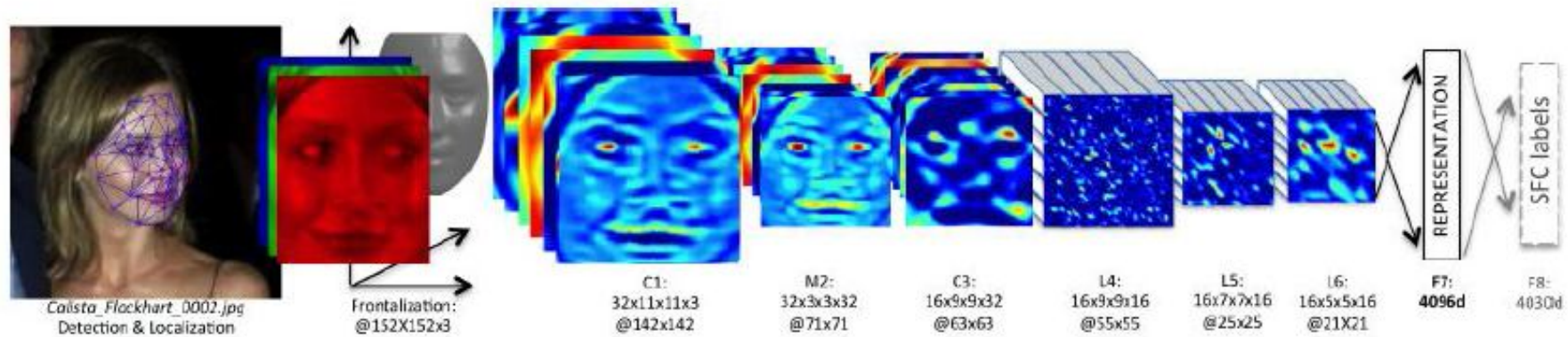
- Ряд небольших улучшений алгоритмов, позволивших учить быстрее и бороться с переобучением
- Например, размножение данных (data augmentation):



- Из  $256 \times 256$  случайно выбираем фрагменты  $224 \times 224$  и их отражения



# DeepFace (Facebook)

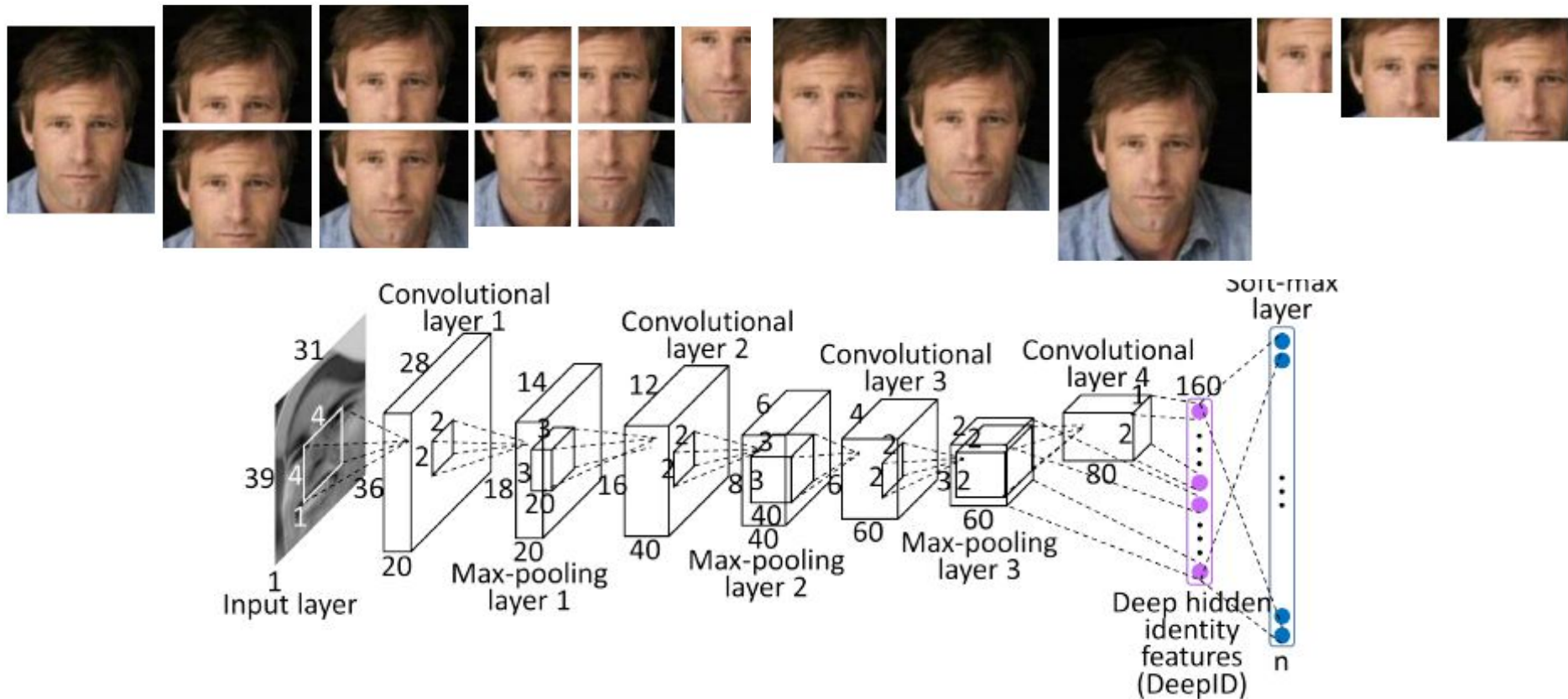


- Обучаем классификатор для 4000 человек – многослойную нейросеть
- 4М изображений – 4000 человек по 1000 изображений
- Последний слой используется как признаки для описания человека
- Точность 97.35% (У человека на обрезанных – 97.53%)

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf. **DeepFace: Closing the Gap to Human-Level Performance in Face Verification.** CVPR 2014.



# DeepID (Chinese University of Hong-Kong)



- Точность комитета сетей 99.15% (У человека на обрезанных – 97.53%, на полных 99.2%)

Yi Sun, Xiaogang Wang, and Xiaoou Tang. **Deep Learning Face Representation from Predicting 10,000 Classes**. *CVPR*, 2014.



# FaceNet от Google (2015)



1.22



1.33

1.04



1.33

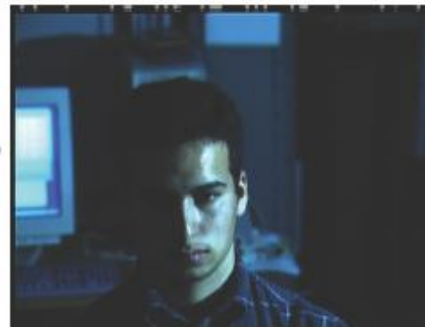


1.26

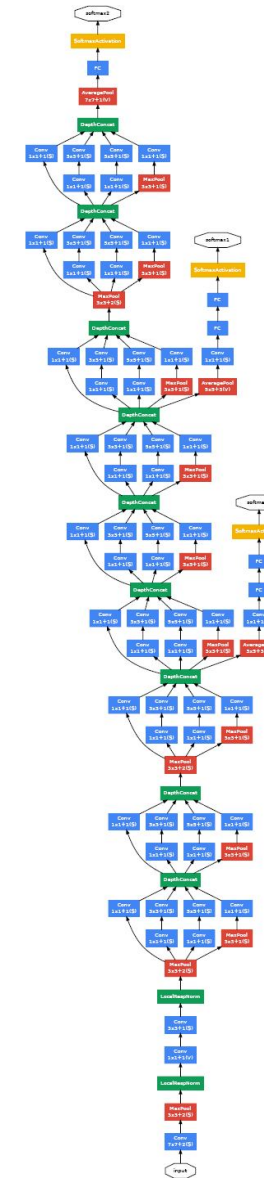
0.78



0.99



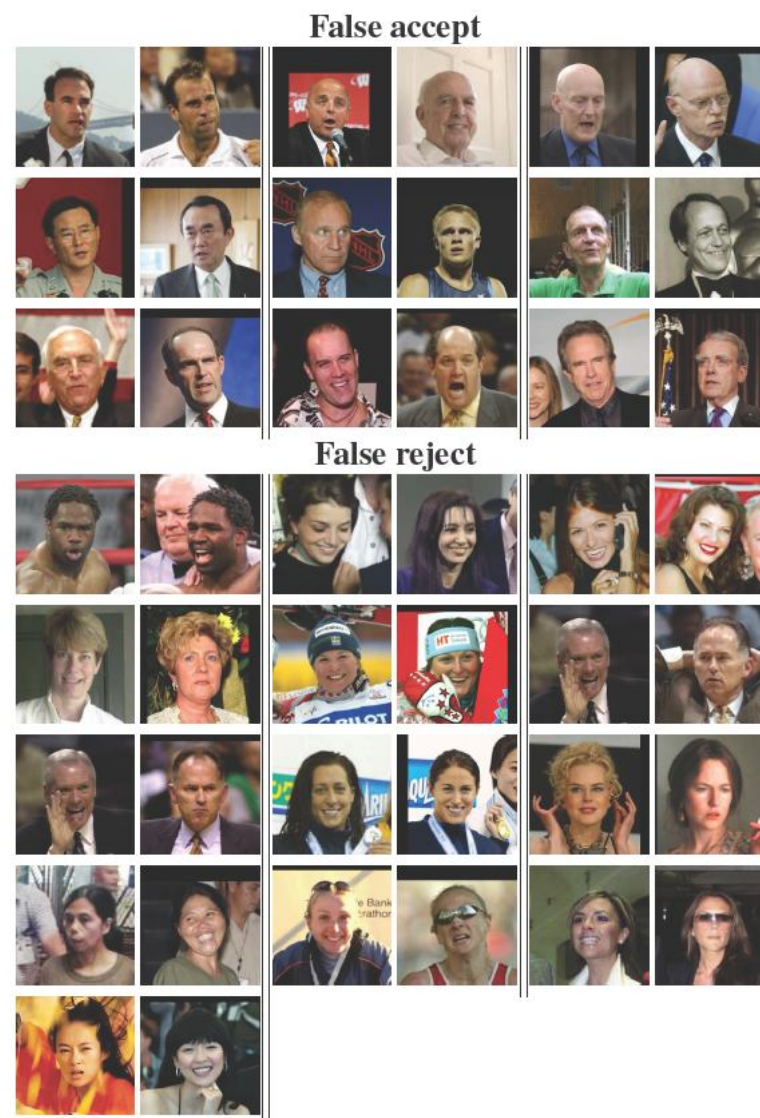
0.99





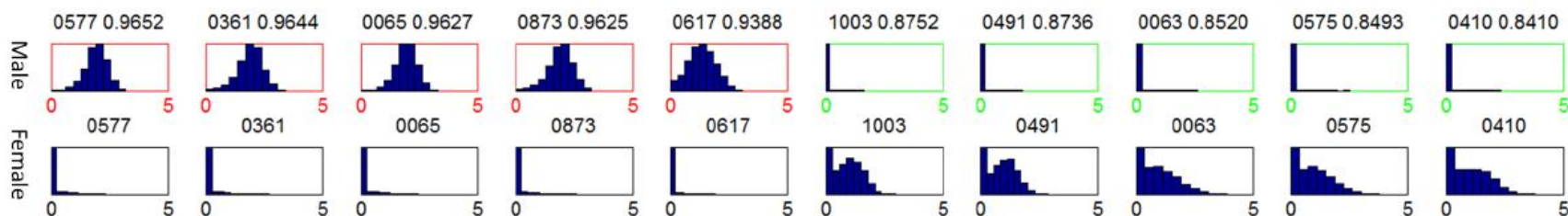
# FaceNet от Google (2015)

- 220 млн. изображений 8 млн. разных людей
- От 6 до 140 млн. параметров в модели
- 99.63% точности

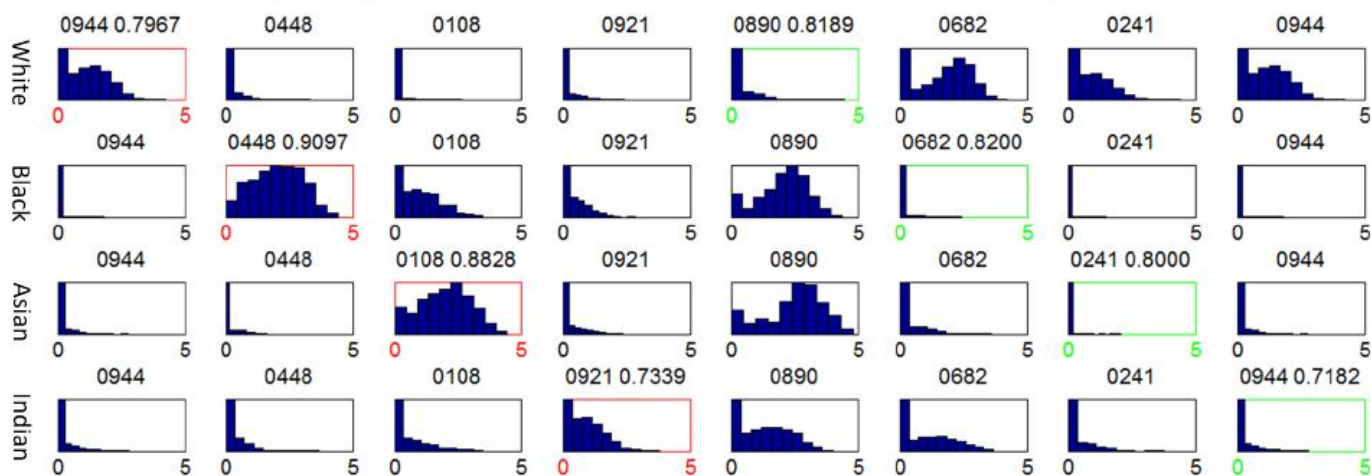




# «Бабушкины нейроны»



(a) Histogram of neural activations over sex-related attributes (Male and Female).

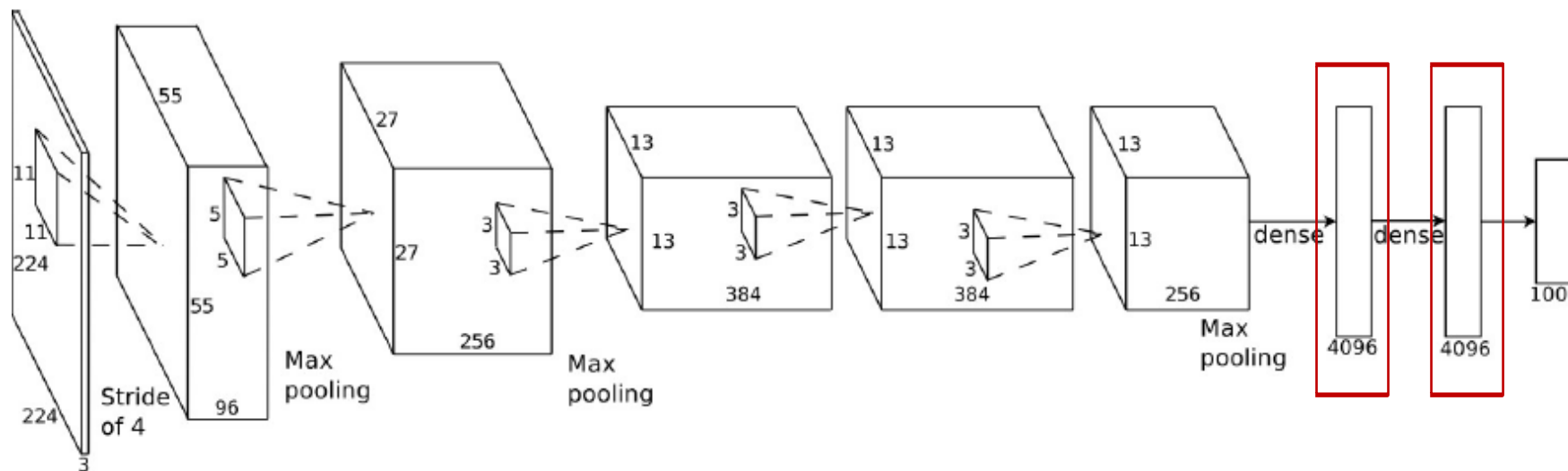


(b) Histogram of neural activations over race-related attributes, i.e., White, Black, Asian, and Indian.

Можно выделить нейроны, которые обычно срабатывают на изображениях одной категории, и не срабатывают на других (т.е. «признаки»)



# Высокоуровневые признаки

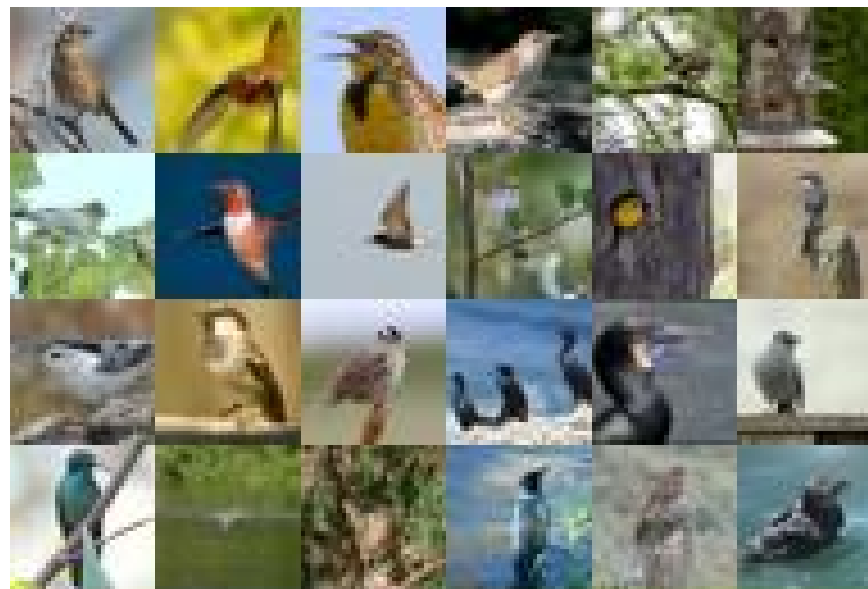


- Получается, что каждый слой нейронной сети извлекает какую-то полезную информацию из изображения
- Выход одного слоя можно использовать как признаки изображения и обучать на них классификатор
- Можно обучить сеть на одних данных (ImageNet) и применять на других для вычисления признаков

Donahue et. al. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, 2013



# Классификация близких объектов



Method	Part info	mean Accuracy
Sift+Color+SVM[45]	✗	17.3
Pose pooling kernel[49]	✓	28.2
RF[47]	✓	19.2
DPD[50]	✓	51.0
Poof[5]	✓	56.8
CNN-SVM	✗	53.3
CNNaug-SVM	✗	61.8
DPD+CNN(DeCaf)+LogReg[10]	✓	65.0

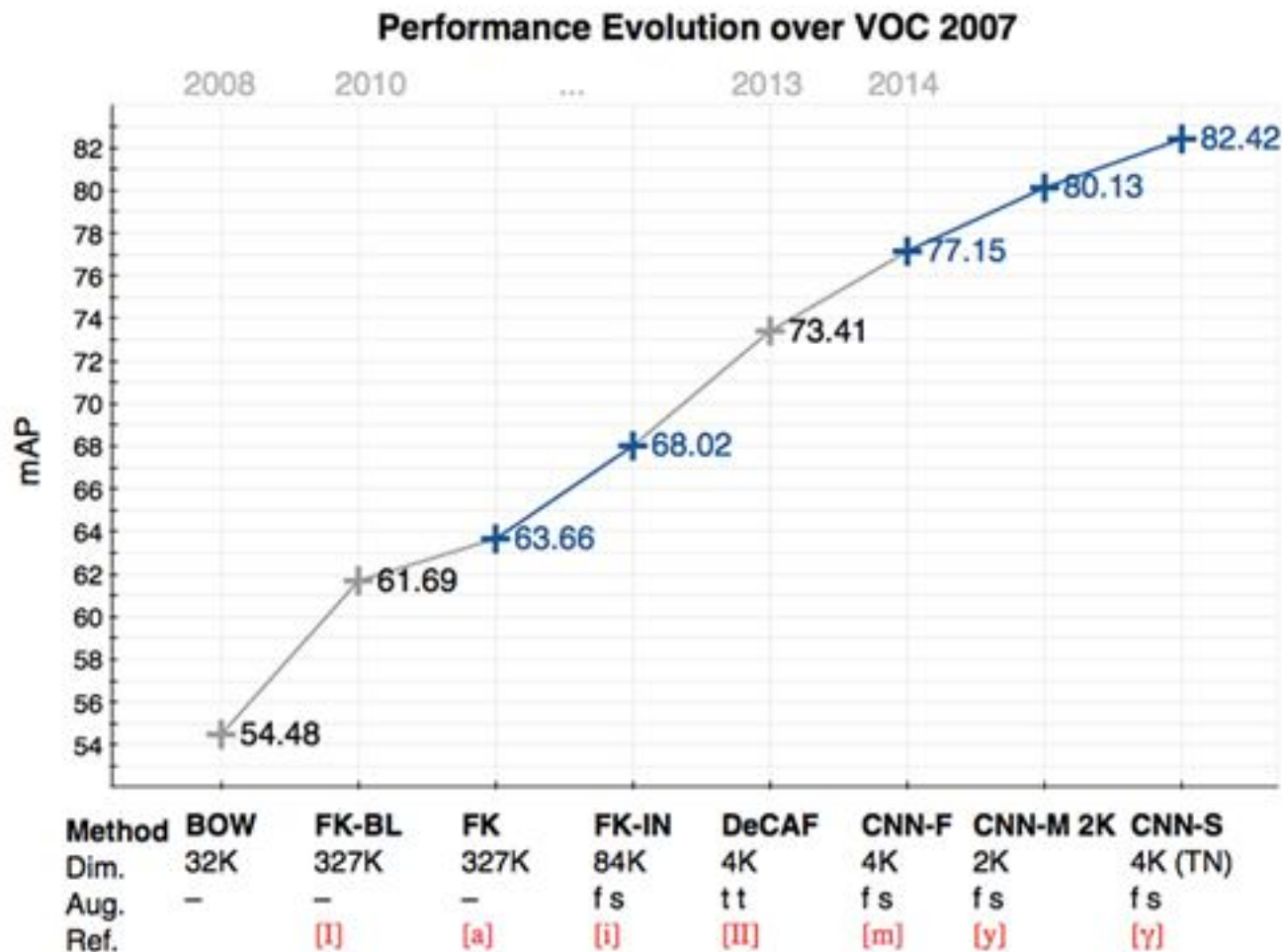
Table 3: Results on CUB 200-2011 Bird dataset. The table distinguishes between methods which use part annotations for training and sometimes for evaluation as well and those that do not. [10] generates a pose-normalized CNN representation using DPD [50] detectors which significantly boosts the results to 64.96.

- Определение видов птиц
- Возьмём нейросеть, обученную для классификации ImageNet
- Применим её для получения вектор-признаков изображений
- Обучаем классификатор поверх этих признаков
- Profit!

Ali Sharif Razavian Hossein Azizpour Josephine Sullivan Stefan Carlsson  
CNN Features off-the-shelf: an Astounding Baseline for Recognition. 2014



# Рост качества по методам



K. Chatfield et. Al. Return of the Devil in the Details: Delving Deep into Convolutional Nets, BMVC 2014



# Решена ли задача?

---

- Коллекция LFW устарела, на ней показывают почти совершенный результат.
- При применении на других данных результаты хорошие, но пока далеки от идеальных
- Создаются новые коллекции, на которых будут оценивать качество работы методов