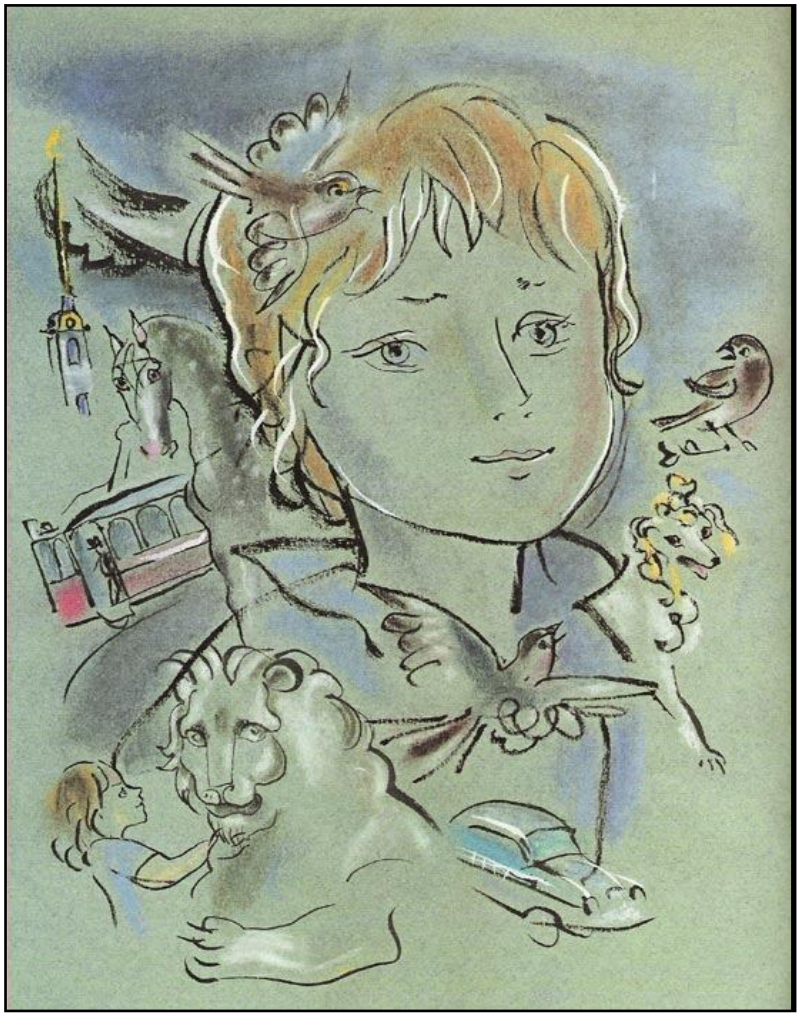
Распознавание изображений 1



Антон Конушин

Slides adapted from Alexei Efros, Fei-Fei Li, Rob Fergus, Antonio Torralba, Jean Ponce and Svetlana Lazebnik

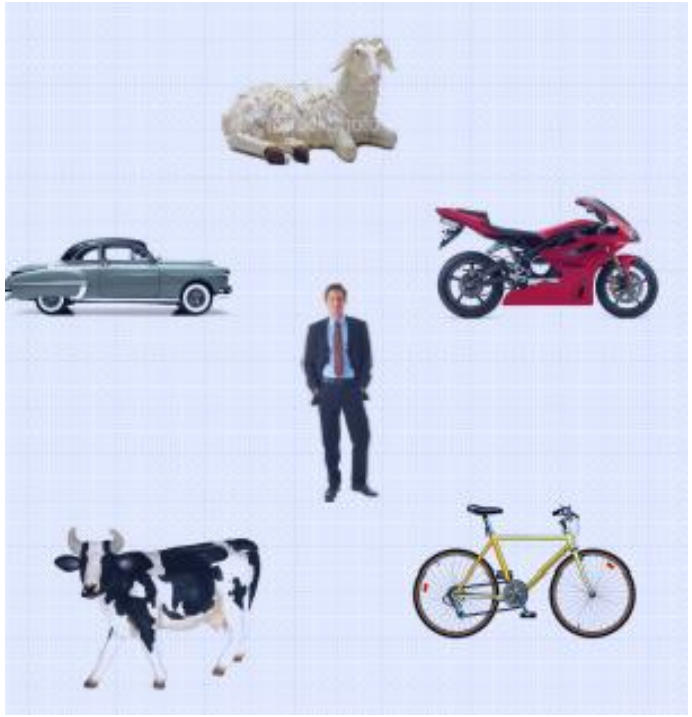
Выделение объектов

Яндекс

Объекты и Материалы

Объекты (Things)

- Имеют определенные размер и формы



Найти «человека»

Материалы (Stuff)

- Однородный или повторяющийся шаблон мелких деталей
- Нет определенного размера и формы

Найти «небо»

Ted Adelson, Forsyth et al. 1996.

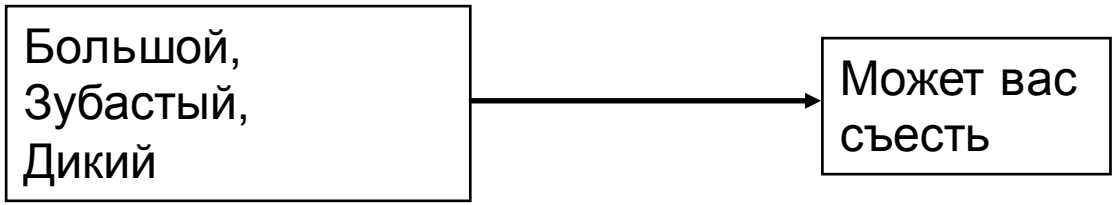
Slide: Geremy Heitz

Объекты и категории

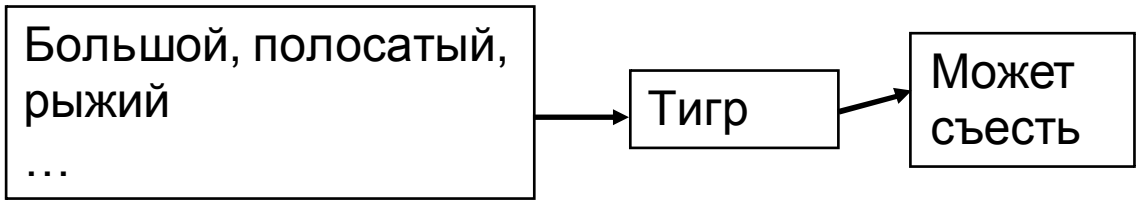
Категории

Яндекс

- Понятие «категории» содержит информацию о том, что мы можем делать с объектом.
- Восприятия напрямую:



- Опосредственное восприятие (Категоризация)



- Для каждого объекта своё слово не придумаешь, и категорий (слов в языке) меньше, чем объектов в мире
- Категории позволяют компактно передать информацию

Субъективность категорий

Яндекс

Функции объекта зависят от наблюдателя.

«Женщины, Огонь и Опасные Вещи» – это отдельная категория в языке Австралийских аборигенов (Lakoff 1987)

«Естественные» категории

- Определяется наилучшими примерами (прототипами)
- Задаем степень соответствия категории
- Нечеткие правила

Протитип или набор примеров?

Модель на прототипе

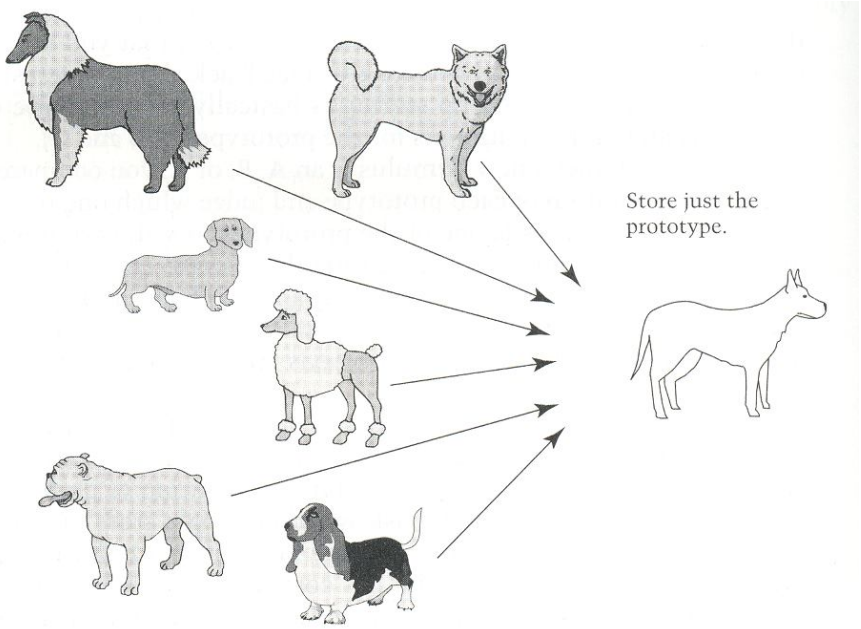


Figure 7.3. *Schematic of the prototype model. Although many exemplars are seen, only the prototype is stored. The prototype is updated continually to incorporate more experience with new exemplars.*

Решение о соответствии категории принимается путем сравнения объекта с прототипом

Модель на примерах

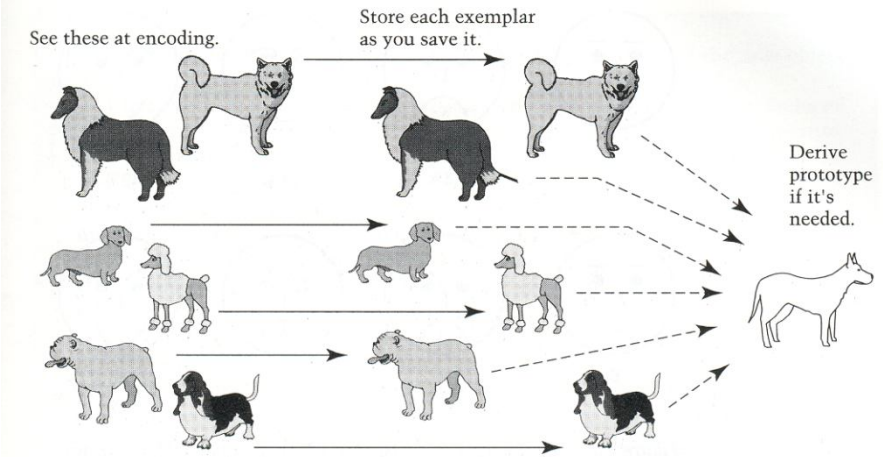
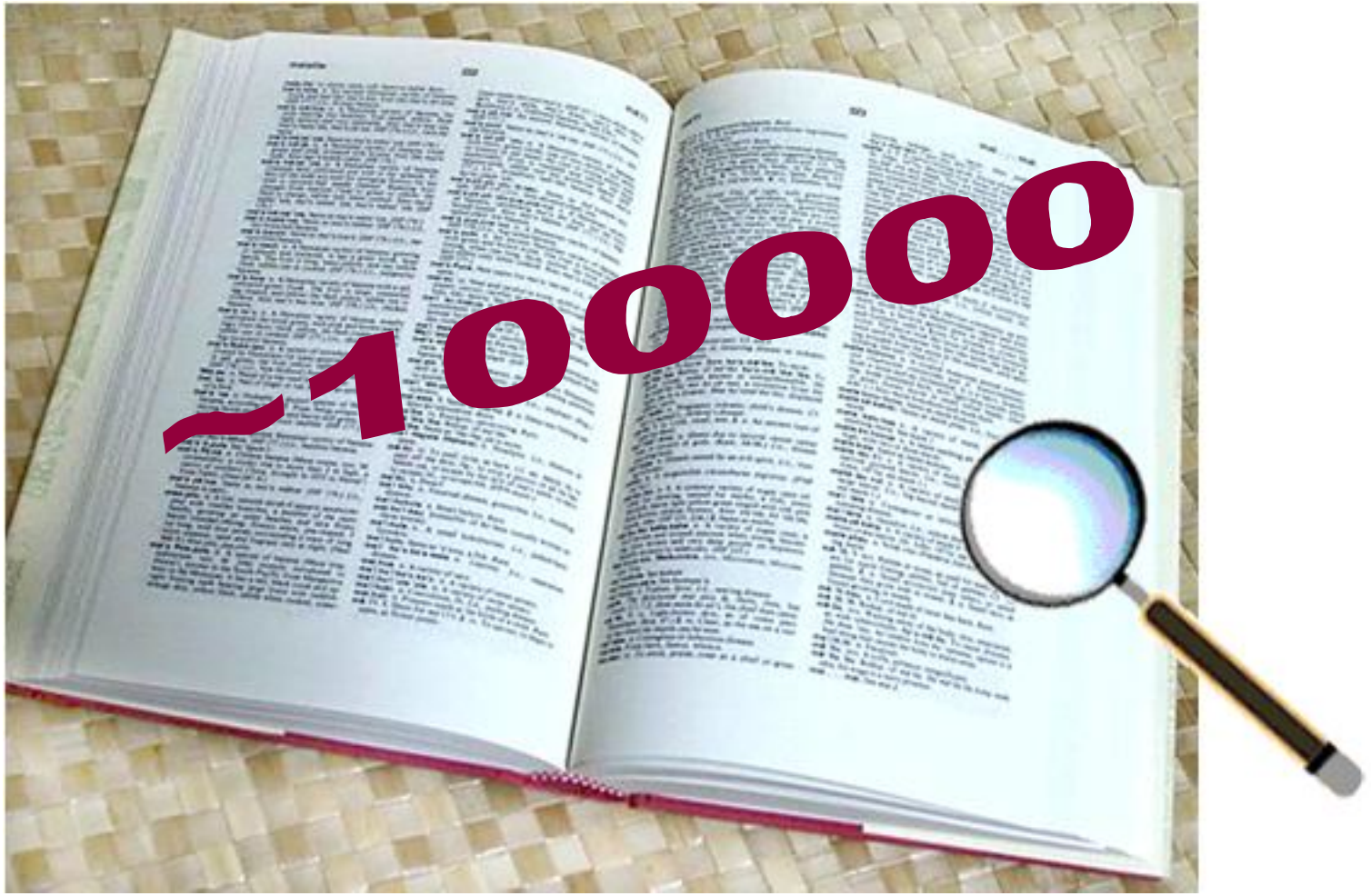


Figure 7.4. *Schematic of the exemplar model. As each exemplar is seen, it is encoded into memory. A prototype is abstracted only when it is needed, for example, when a new exemplar must be categorized.*

Решение о соответствии категории принимается путем сравнения объекта со множеством примеров из категории

Slide by Torralba

Сколько всего категорий?

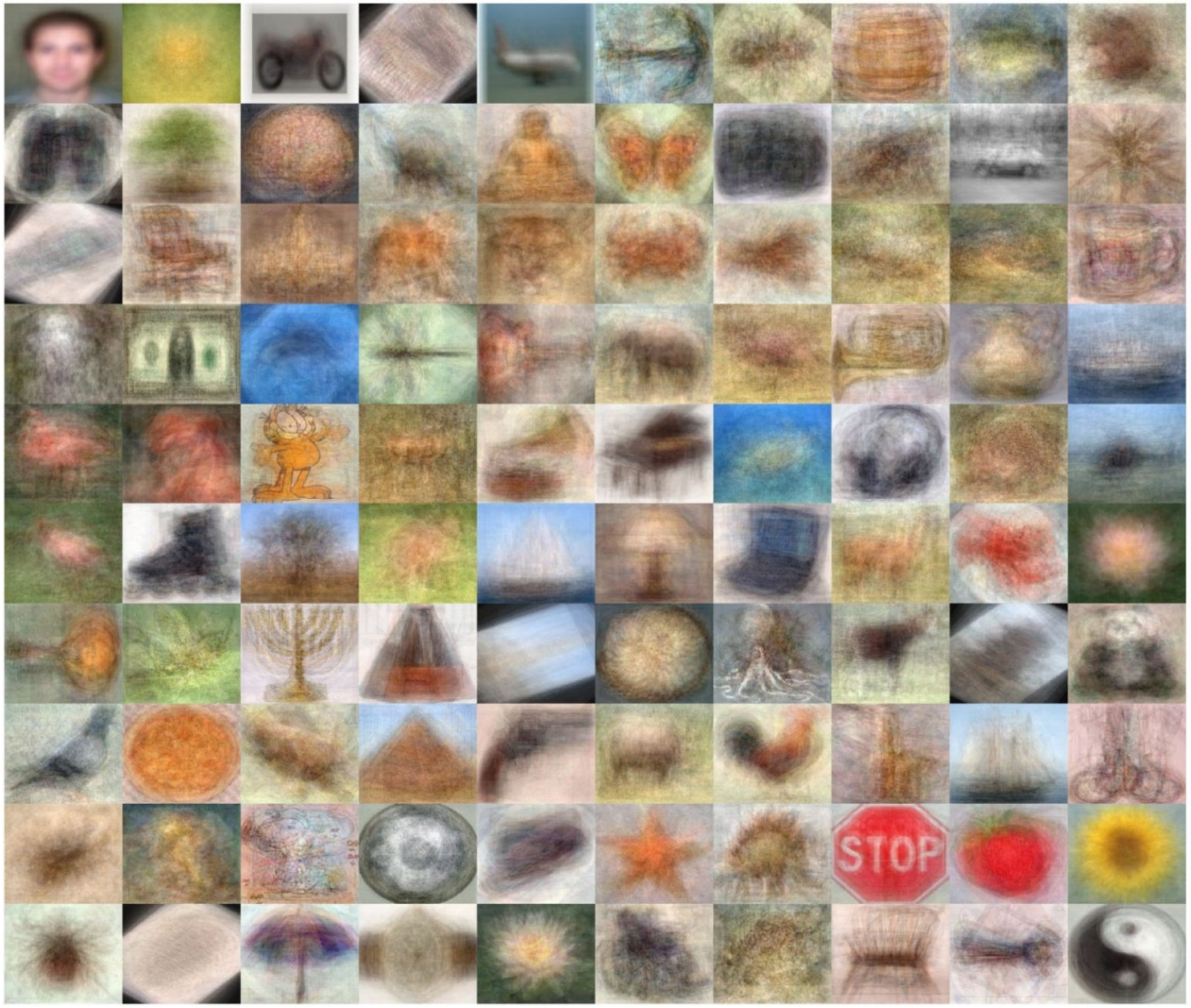


171000+ слов всего, 47000 устаревших слов, половина существительные

Biederman 1987

Caltech-101

Яндекс



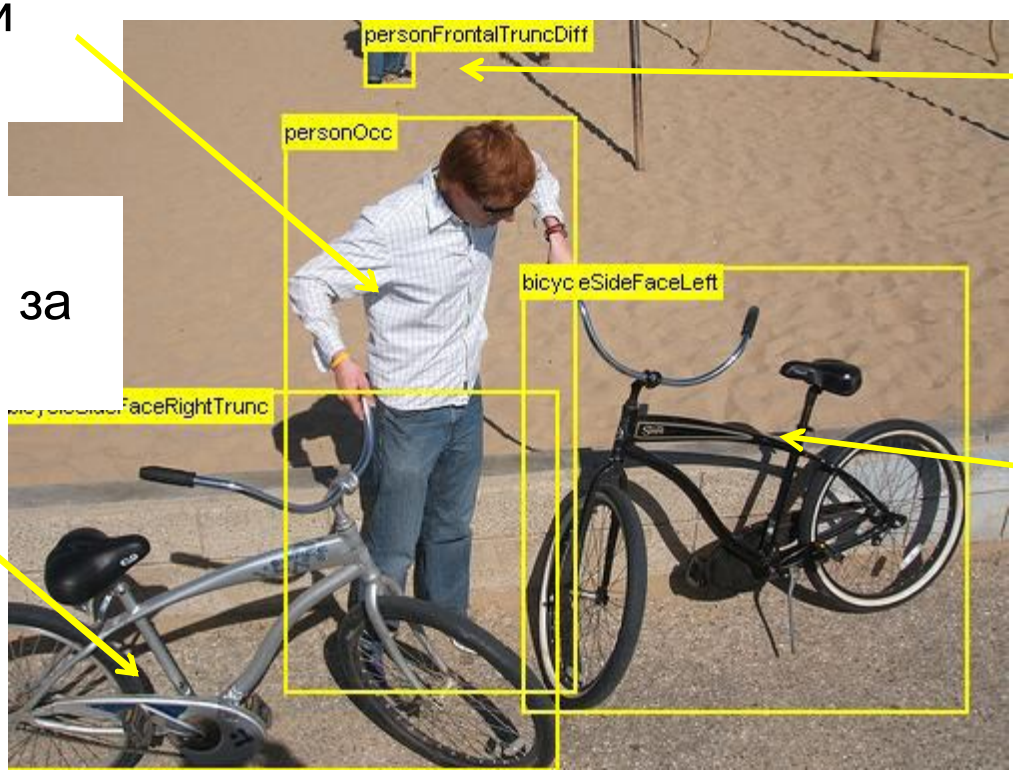
Pascal Visual Object Classes Challendge

Яндекс

Несколько десятков категорий, но реальные изображения из коллекций

Перекрытый
Объект перекрыт другим внутри рамки

Обрезанный – объект выходит за пределы рамки



Сложный
Не участвует в оценке

Положение
Смотрит налево

http://pascallin.ecs.soton.ac.uk/challenges/VOC/

IMAGENET

Яндекс

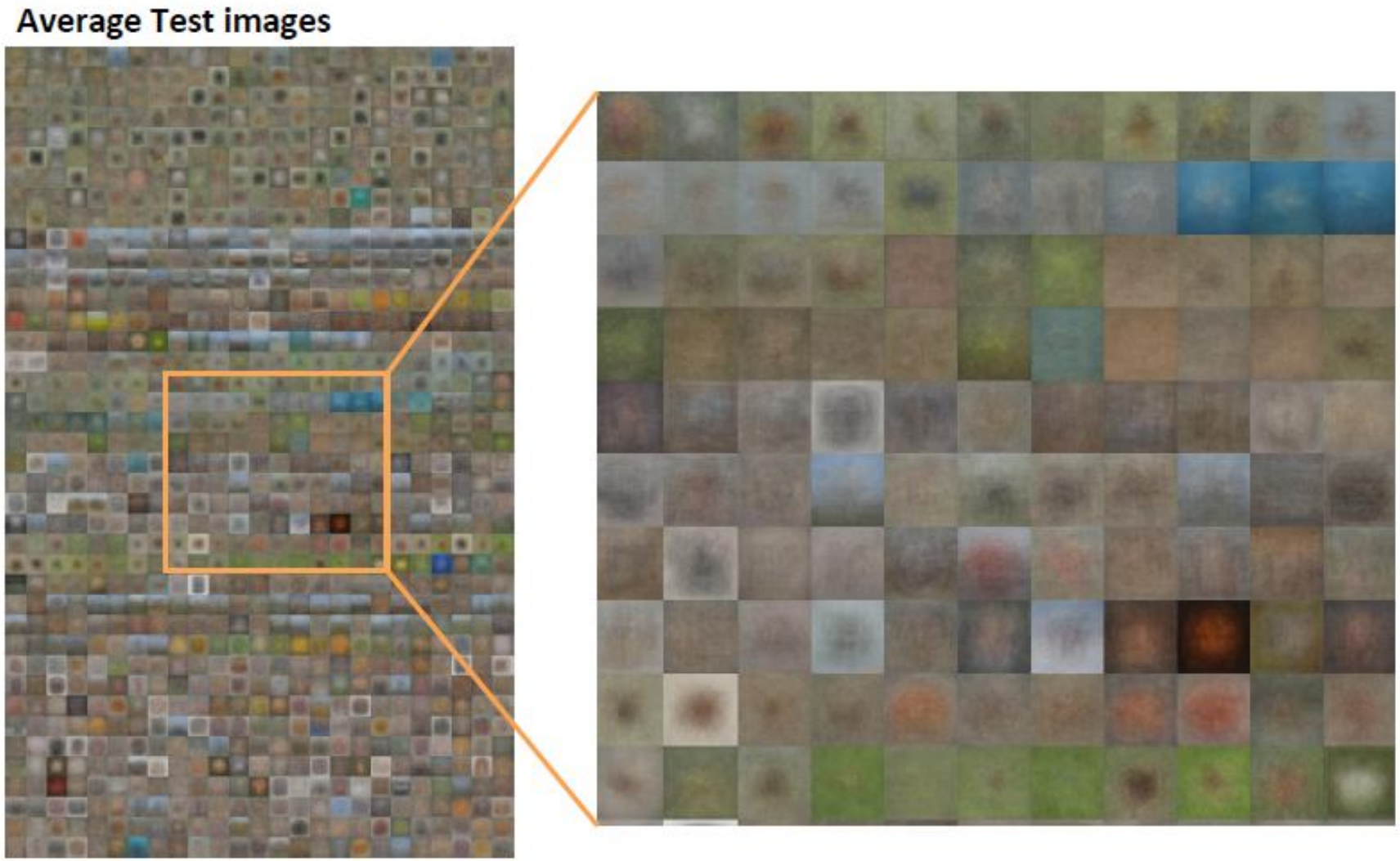
14M изображений

Будет 1000 на каждую категорию

http://www.image-net.org

Изображения в среднем

Яндекс



Fine-graded categorization

Яндекс

- 150 пород собак
- 20000 изображений

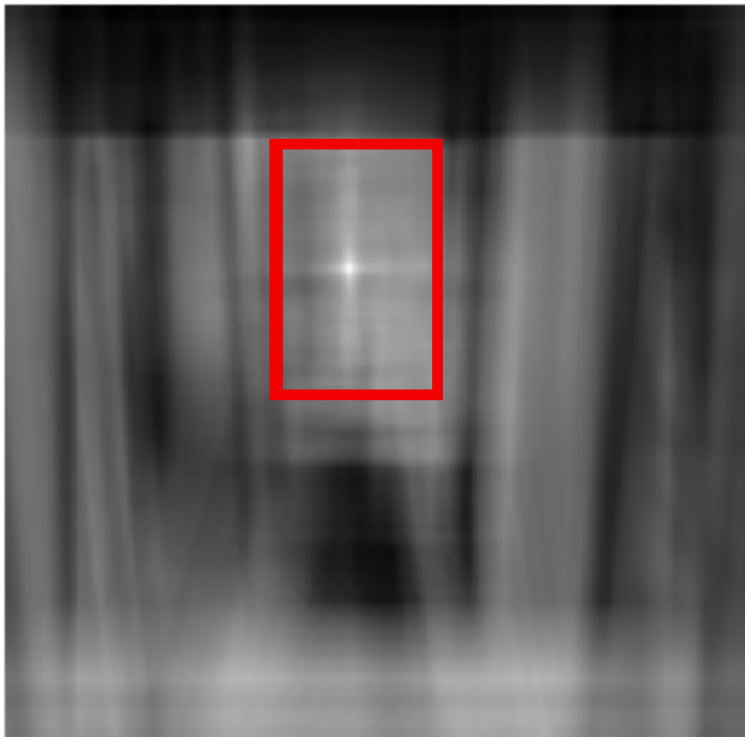
http://vision.stanford.edu/aditya86/ImageNetDogs/

Простейший метод выделение объектов

Найдём стул в изображении

Шаблон

Выход корреляции

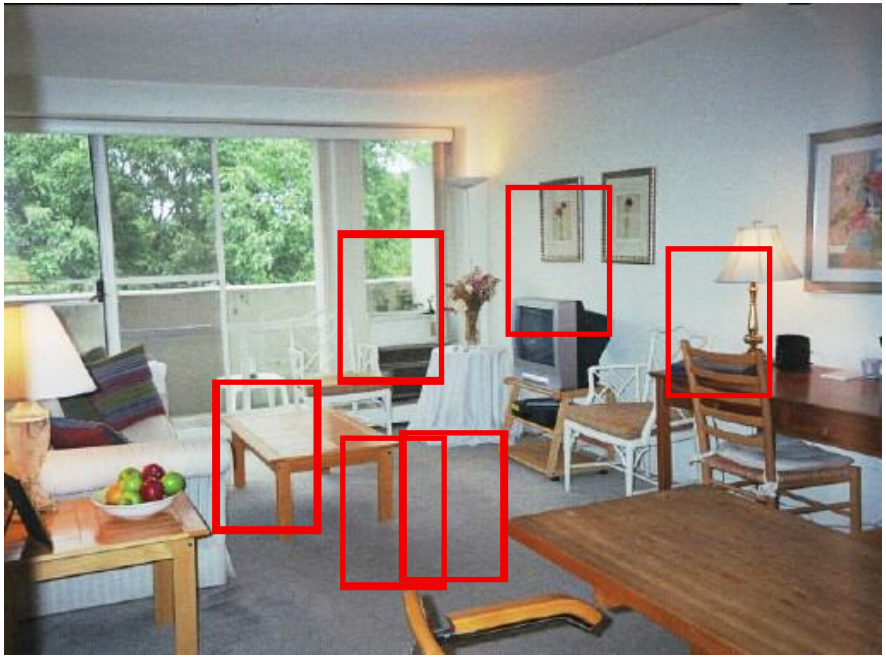


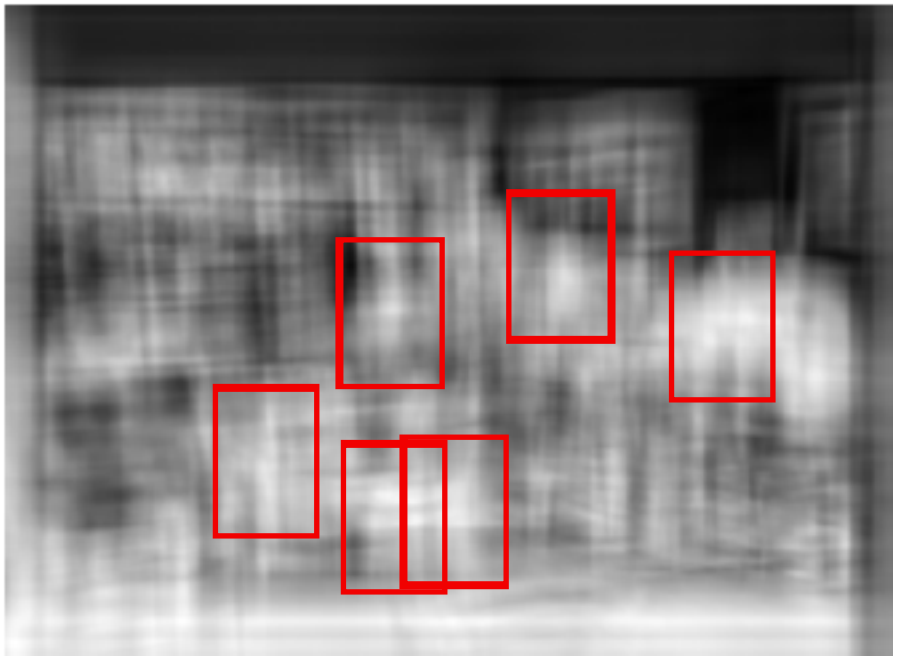
Сопоставление шаблонов - простейший алгоритм для выделения объектов

- Фиксируем изображение объекта («шаблон»)
- Сканируем шаблоном всё целевое изображение, сравнивая фрагменты целевого изображения с шаблоном

Сопоставление изображений

Найдём стул в этом изображени





Мусор

Почему не получилось?

Категоризация изображений

Это стул?

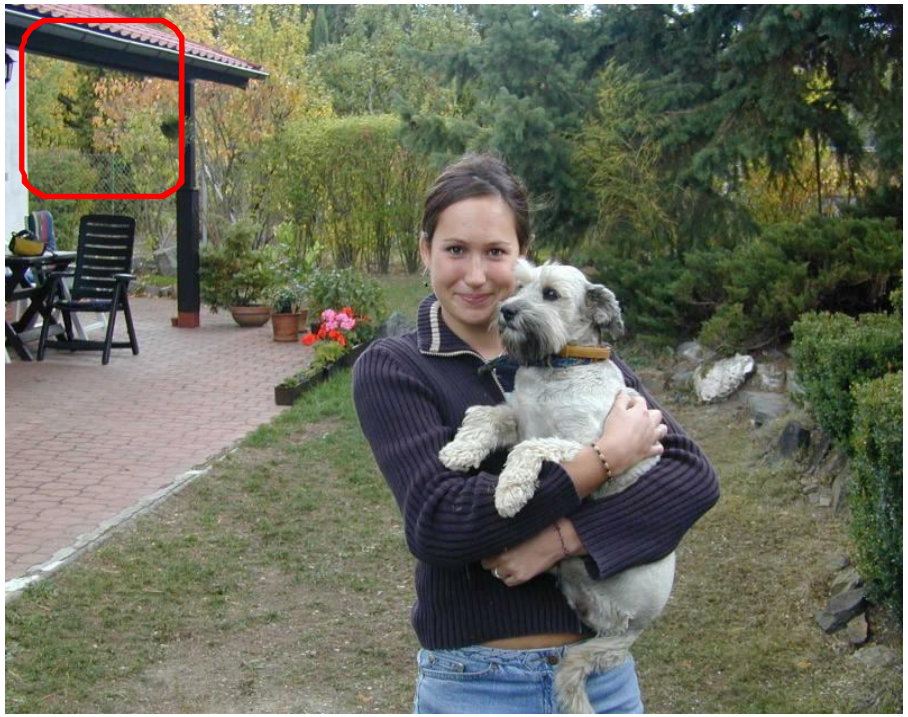
Да

Нет

- Слишком слабый алгоритм классификации
- Детали алгоритма
  - Бинарный
  - Признаки – цвет пикселов
  - Вектор-признак – вектор цветов всех пикселов
  - Функция сравнения – среднеквадратичное расстояние

Скользящее окно

Сам принцип – очень хороший



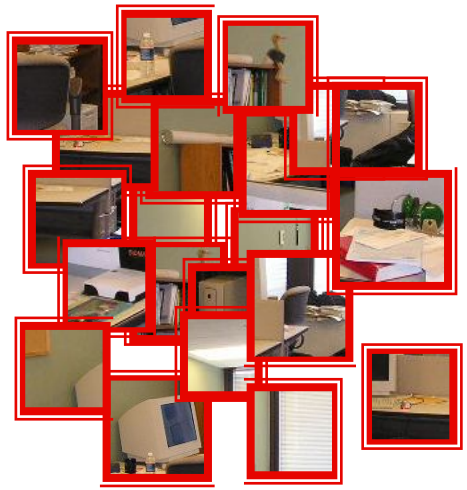
- Рассматриваем только прямоугольную область (окно)
- Если выбрали точно, то в эту область попадёт только лицо
- Просмотрим все области, «сканируя» окном изображение

- Фактически, это простейший вариант сегментации изображения (разбиения изображения на фрагменты)

Классификация окон

- Разделяем изображение на множество перекрывающихся окон
- Сформулируем задачу выделения объектов как задачу классификации окна – объект / не объект
- Для обучения классификатора будем собирать выборки окон с объектами и без объектов

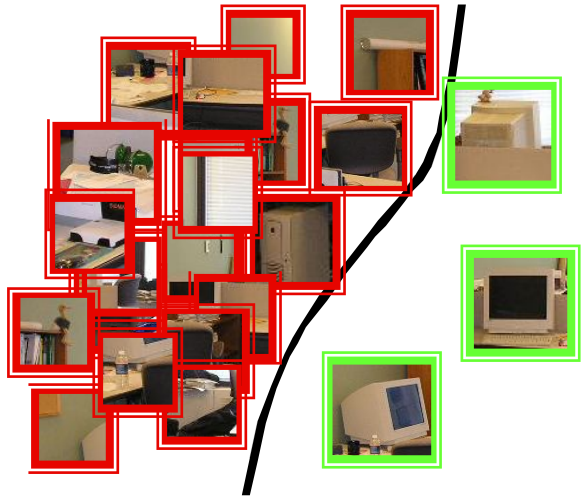
Где мониторы?



Мешок фрагментов

Решающая граница

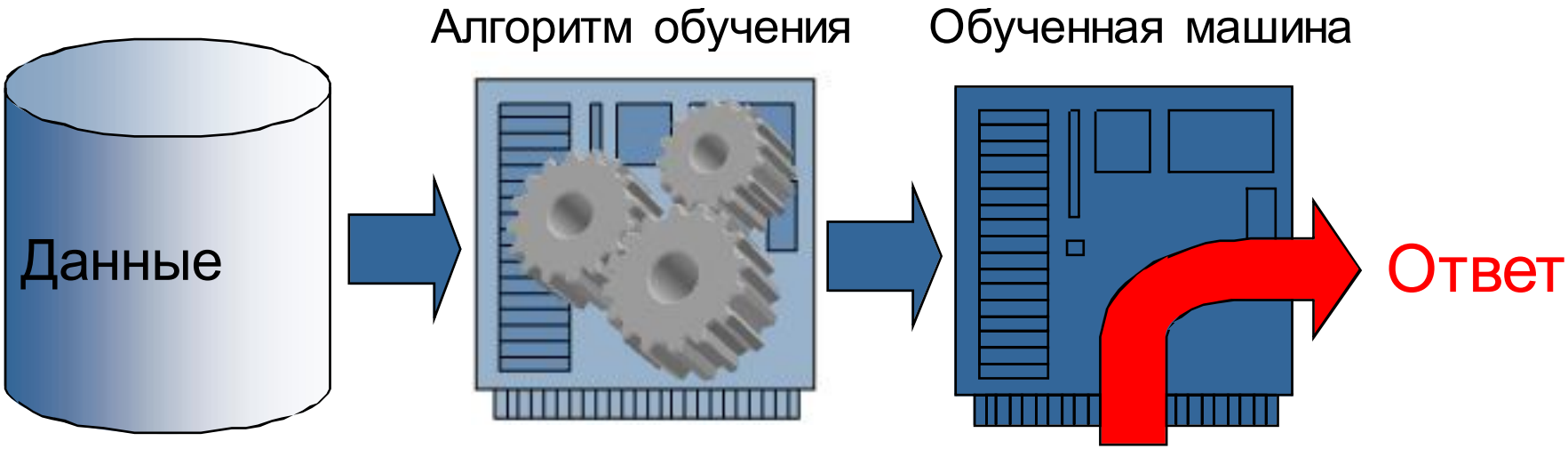
Фон



Мониторы

В пространстве признаков

Машинное обучение

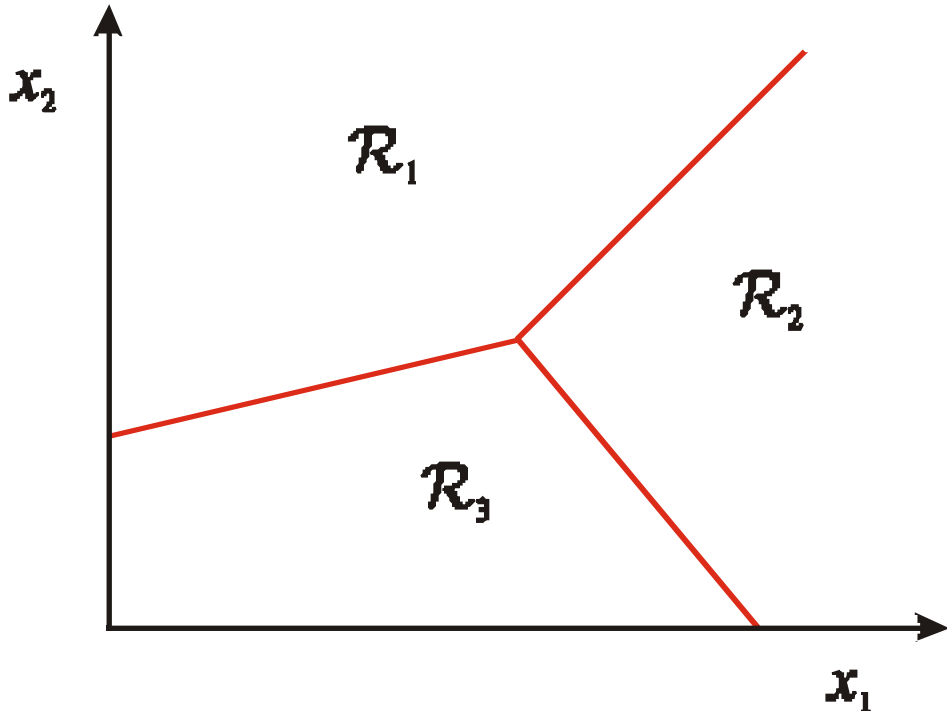


- Обучение ≠ «заучивание наизусть»
  - Заучить наизусть – для машины не проблема
  - Мы хотим научить машину делать выводы!
  - Машина должна корректно работать на новых данных, которые мы ей раньше не давали
  - По конечному набору обучающих данных машина должна научиться делать выводы на новых данных

Задача классификации образов

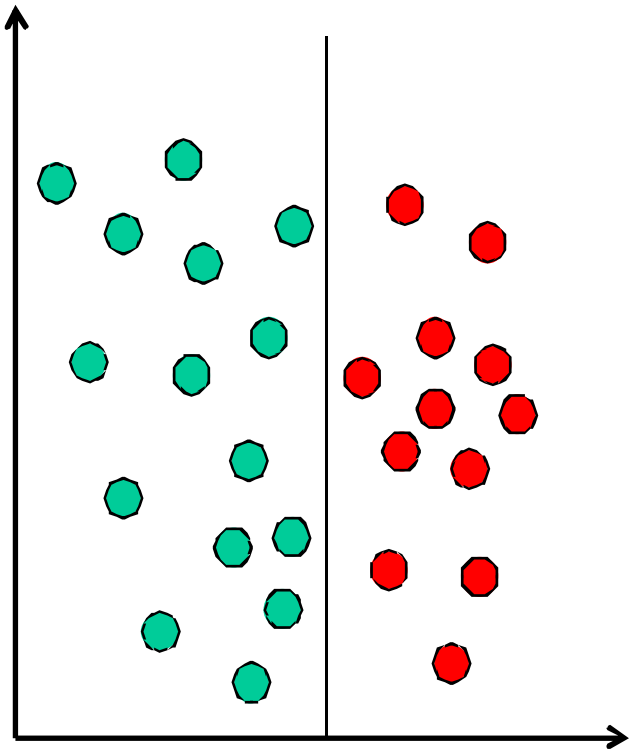
- Есть «обучающая выборка» – набор вектор-признаков **x** с известными метками **y**
- Задача построить функцию **y=f(x)**, которая для каждого вектора-признаков **x** даёт ответ **y**, какому классу принадлежит объект **x**
- Функция **f()** назвается *решающее правило или классификатор*
- Любое *решающее правило* делит пространство признаков на *решающие регионы* разделенные *решающими границами*

Решающее правило

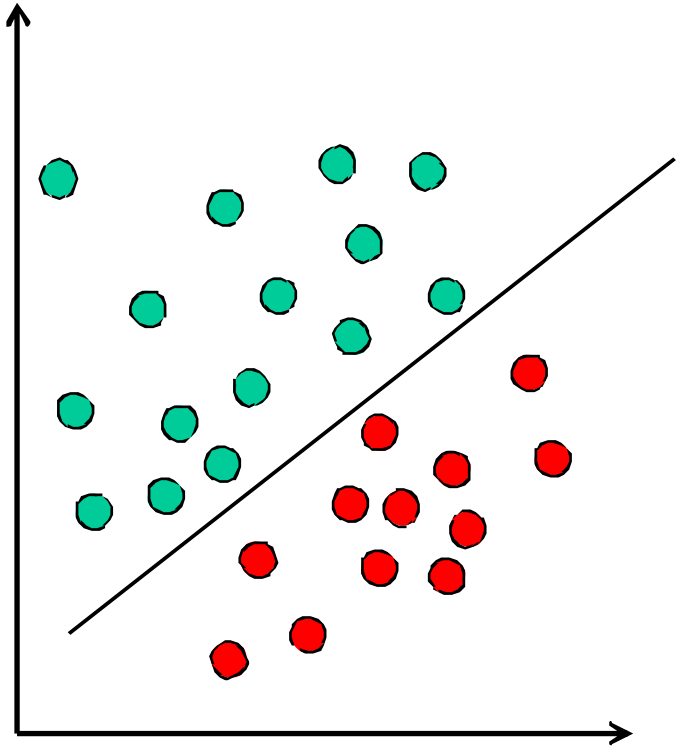


Простые классификаторы

Яндекс



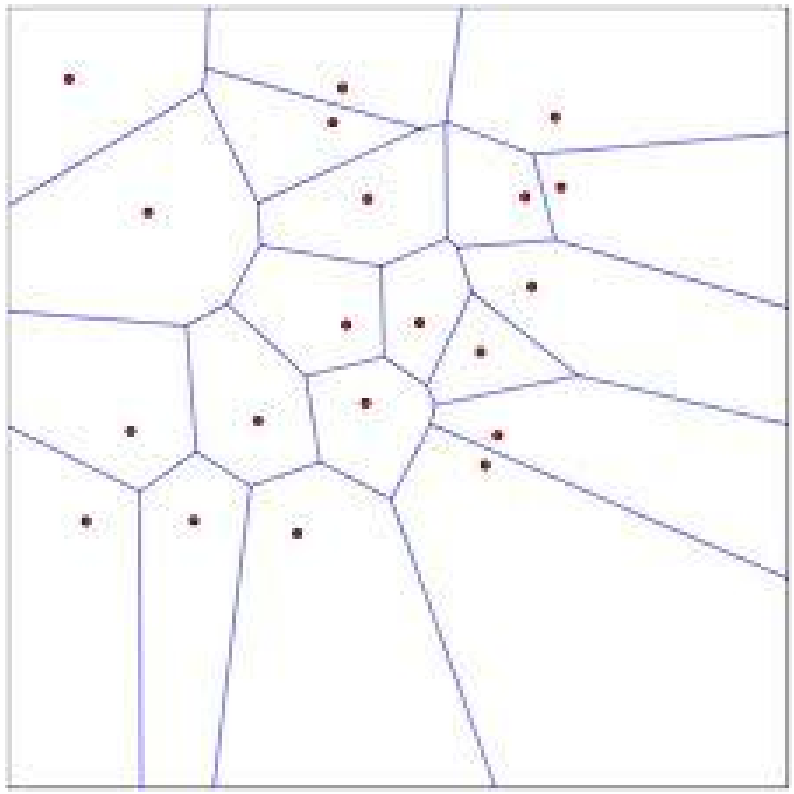
порог



линейный

Ближайший сосед

Яндекс



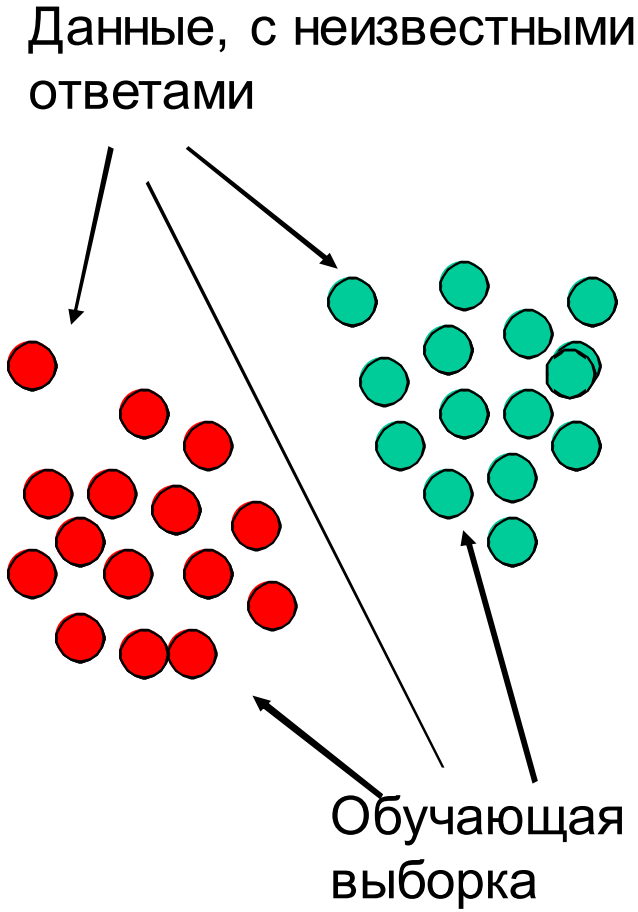
- Nearest Neighbor
- Простейший нелинейный классификатор
- Запоминаем M элементов
- Для нового вектора x ищем ближайший и берём метку его

- Сколько параметров у NN?
  - Зависит от M – количества элементов

Линейная классификация

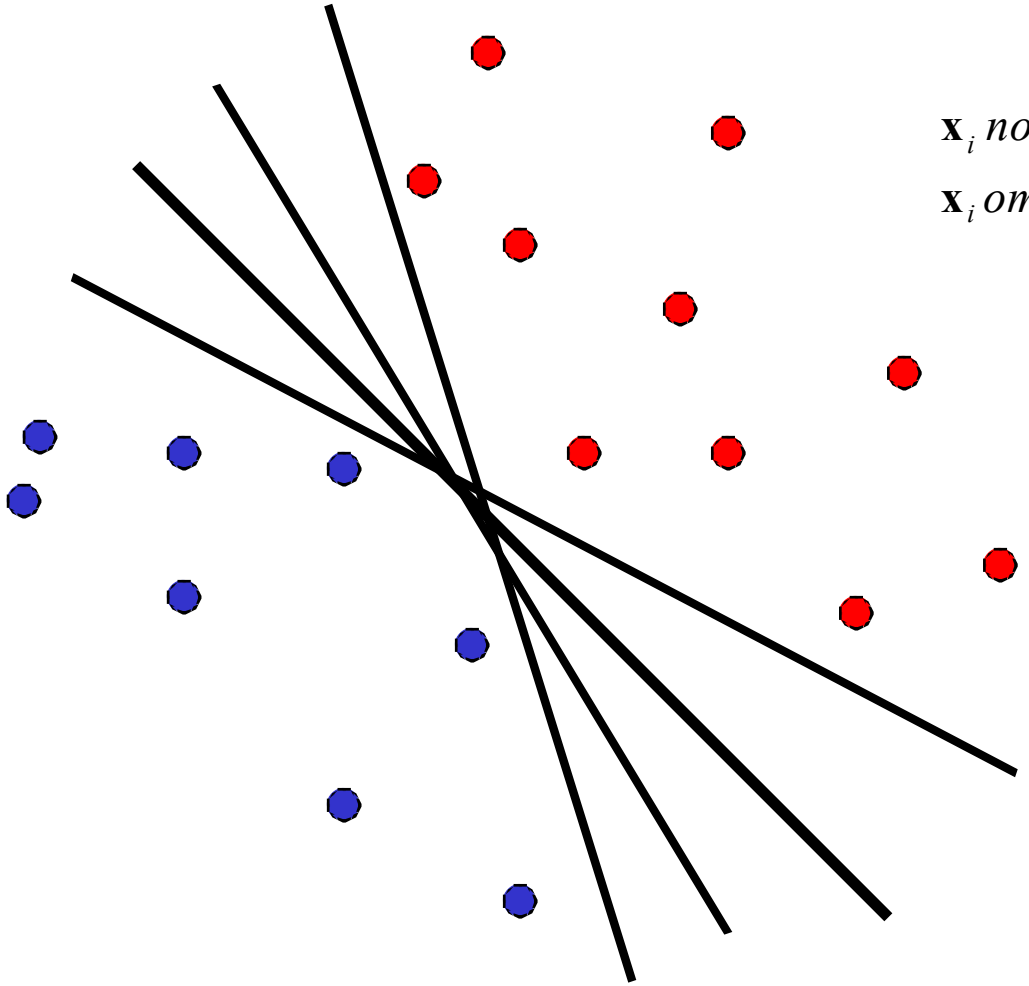
Яндекс

- Рассмотрим случай линейно разделимых данных
- Только для 2х классов
- Т.е. таких, что вектора в пространстве признаков можно отделить друг от друга гиперплоскостью



Линейный классификатор

- Найдем линейную функцию (гиперплоскость) и разделим положительные {y=+1} и отрицательные {y=-1} примеры



$$\mathbf{x}_i \text{ положительные}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

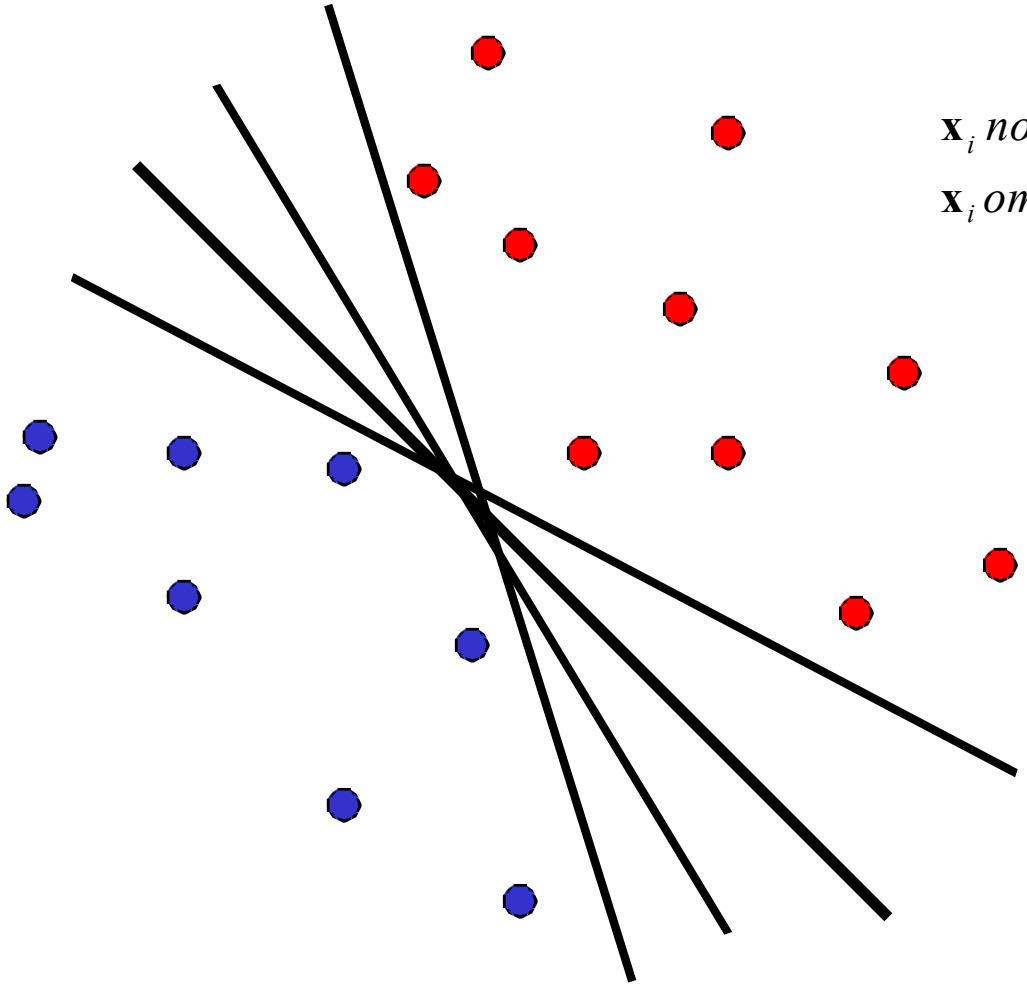
$$\mathbf{x}_i \text{ отрицательные}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Какая гиперплоскость наилучшая?

Линейный классификатор

Яндекс

- Найдем линейную функцию (гиперплоскость) и разделим положительные {y=+1} и отрицательные {y=-1} примеры



$$\mathbf{x}_i \text{ положительные}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

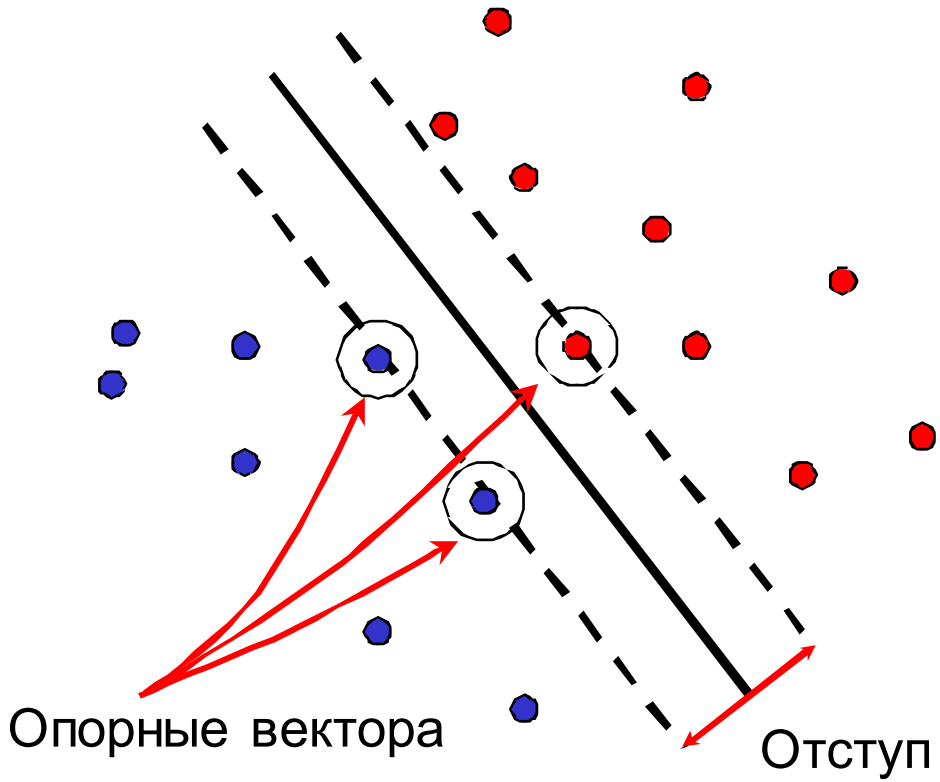
$$\mathbf{x}_i \text{ отрицательные}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Для всех рассмотренных плоскостей эмпирический риск одинаковый

Какая гиперплоскость наилучшая?

Метод опорный векторов (SVM)

- Найдем гиперплоскость, максимизирующую *отступ* между положительными и отрицательными примерами



$\mathbf{x}_i$ *положительные* $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ *отрицательные* $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

Для опорных векторов, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Расстояние от точки до гиперплоскости: $\dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

Поэтому отступ равен $2 / \|\mathbf{w}\|$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Поиск гиперплоскости

Яндекс

1. Максимизируем $2/\|\mathbf{w}\|$
2. Правильно классифицируем все данные:

$$\mathbf{x}_i \text{ положительные } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$

$$\mathbf{x}_i \text{ отрицательные } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

- *Квадратичная оптимизационная задача*:

- Минимизируем $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

  При условии $y_i(\mathbf{w}\cdot\boldsymbol{x}_i+b) \geq 1$

- Решается с помощью методом множителей Лагранжа
- Есть и итеративные методы на основе градиентного спуска

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Поиск гиперплоскости

Яндекс

- Решение: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

Обученные веса

Вектора

- Для большей части векторов вес = 0!
- Все вектора, для которых вес >0 называются опорными
- Определяется только опорными векторами
- Опорные = самые трудные примеры

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Поиск гиперплоскости

Яндекс

- Решение: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

  $b = y_i - \mathbf{w} \cdot \mathbf{x}_i$ для любого опорного вектора

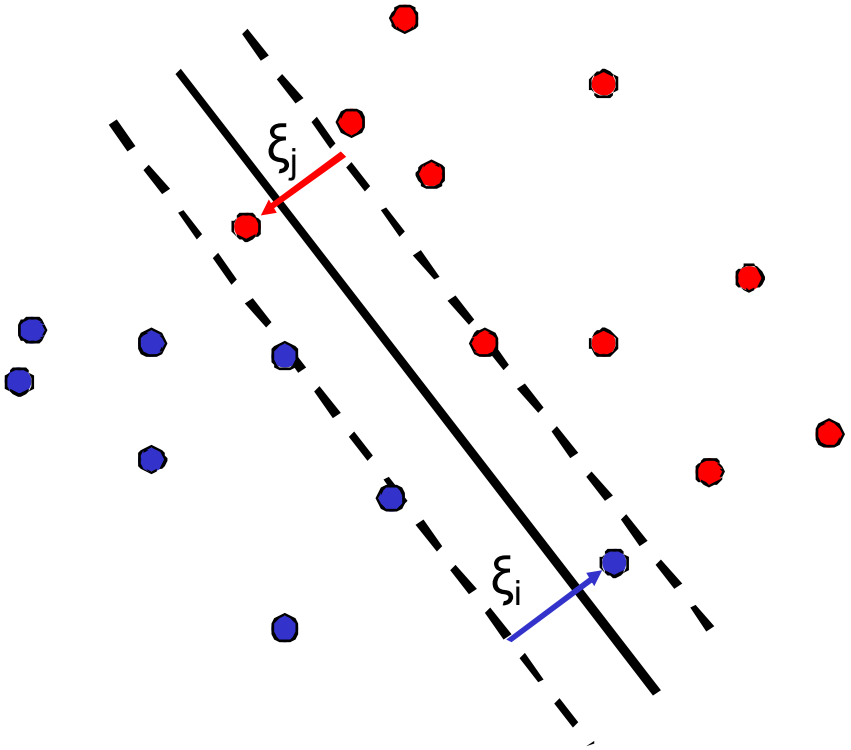
- Решающая функция:

$$\mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

- Решающая функция зависит от скалярных произведений (inner product) от тестового вектора $\boldsymbol{x}$ и опорных векторов $\boldsymbol{x}_i$
- Решение оптимизационной задачи также требует вычисления скалярных произведений $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ между всеми парами векторов из обучающей выборки

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Реальный случай



Вводим дополнительные «slack» переменные: $\xi = (\xi_1, \ldots, \xi_n)^T$

Минимизируем $\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i$

При условии $y_i(wx_i + b) \geq 1 - \xi_i$

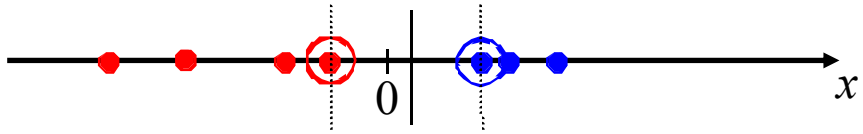
C – параметр регуляризации

В реальном случае идеально разделить данные невозможно (эмпирический риск всегда больше 0)

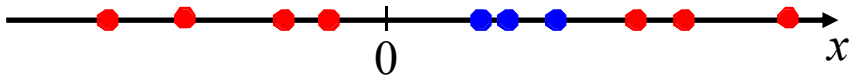
Нелинейные SVM

Яндекс

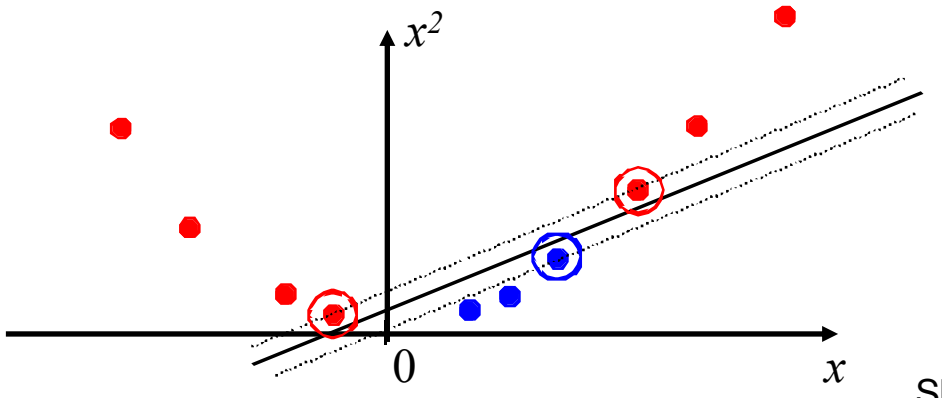
- На линейно разделимых данных МОВ работает отлично:



- Но на более сложных данных не очень:



- Можно отобразить данные на пространство большей размерности и разделить их линейно там:

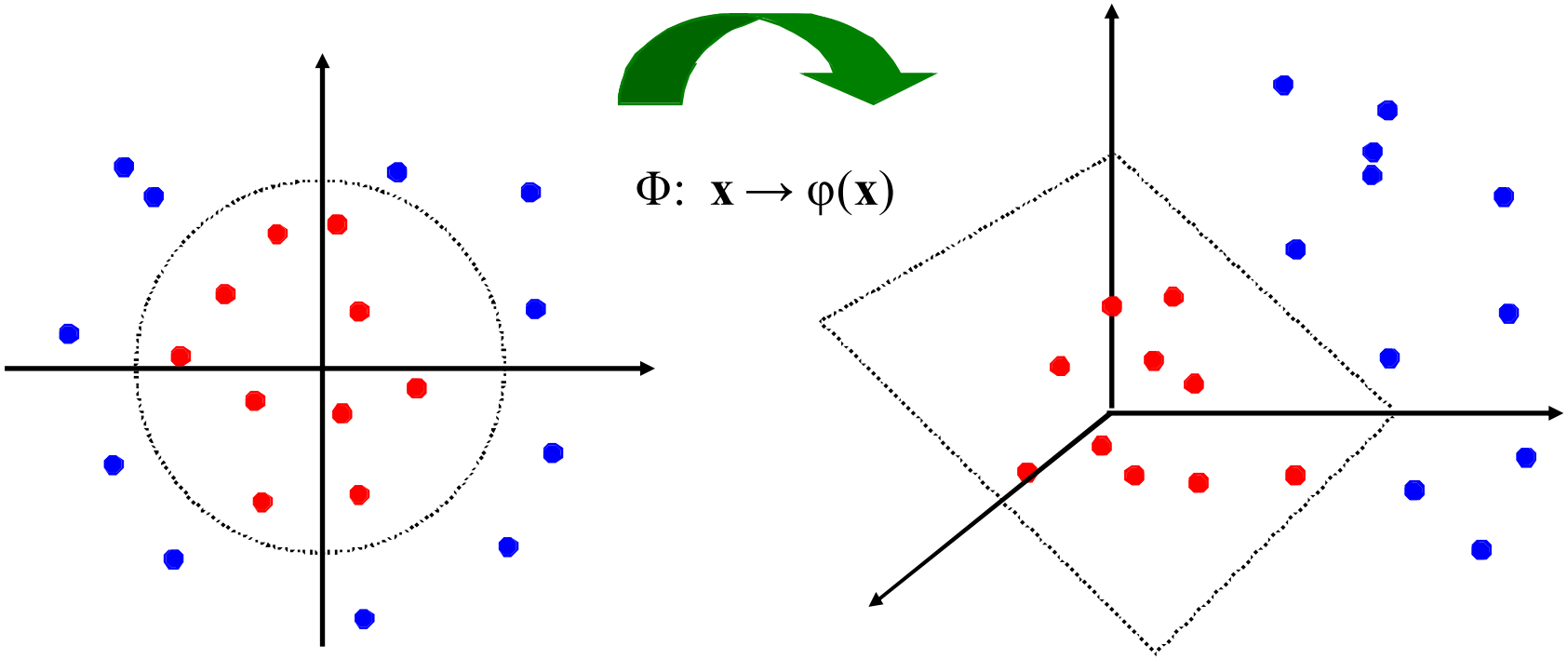


Slide credit: Andrew Moore

Нелинейные SVM

Яндекс

- Идея: отображение исходного пространства параметров на какое-то многомерное пространство признаков (feature space) где обучающая выборка линейно разделима:



Slide credit: Andrew Moore

Нелинейные SVM

Яндекс

- Вычисление скалярных произведений в многомерном пространстве вычислительно сложно
- *The kernel trick:* вместо прямого вычисления преобразования $\varphi(\mathbf{x})$, мы определим ядровую функцию K:

$$K(\mathbf{x}_i, \mathbf{x}_{jj}) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- Чтобы все было корретно, ядро должно удовлетворять условию Мерсера (*Mercer's condition*)
  - Матрица $K(x_i,x_j)$ должна быть неотрицательно определенной
- С помощью ядра мы сможем построить нелинейную решающую функцию в исходном пространстве:

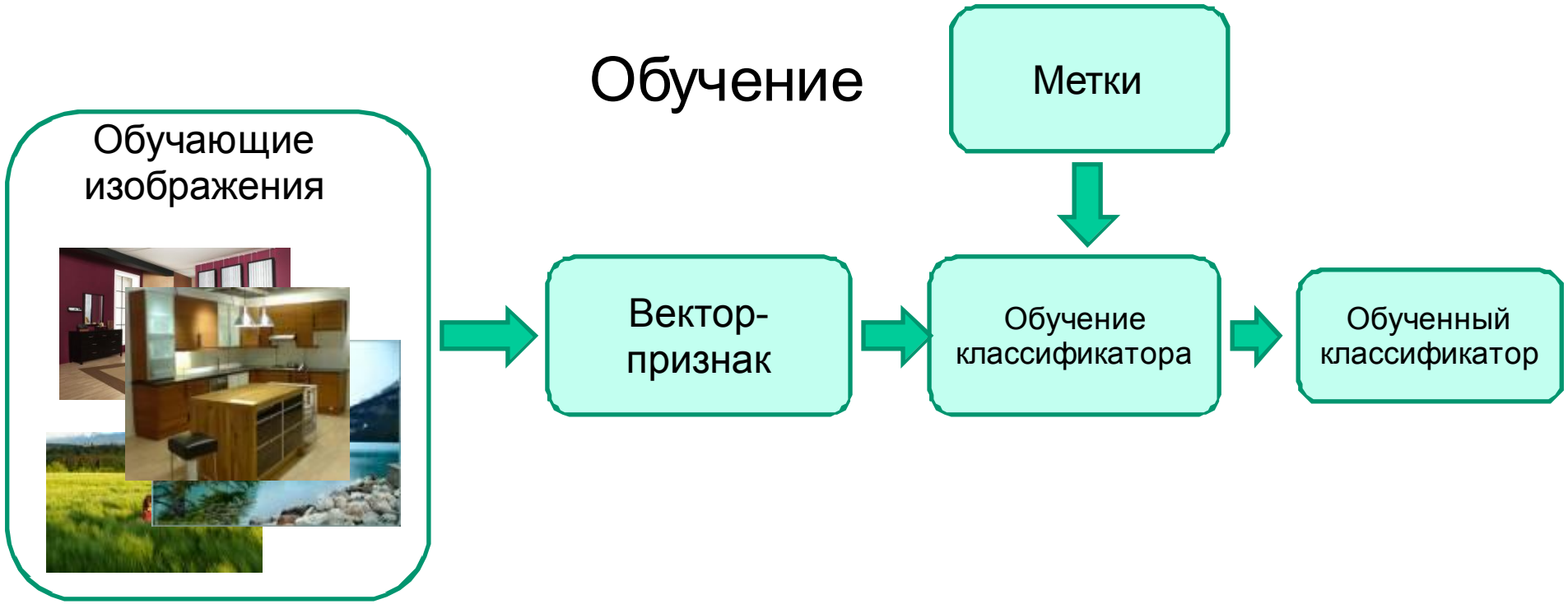
$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Яндекс

Категоризация изображений

Общая схема


Обучение
Метки
Обучающие изображения
Вектор-признак
Обучение классификатора
Обученный классификатор

Категоризация изображений

Яндекс

Обучение

Обучающие изображения

Метки

Вектор-признак

Обучение классификатора

Обученный классификатор

Тестирование

Тестовое изображение

Вектор-признак

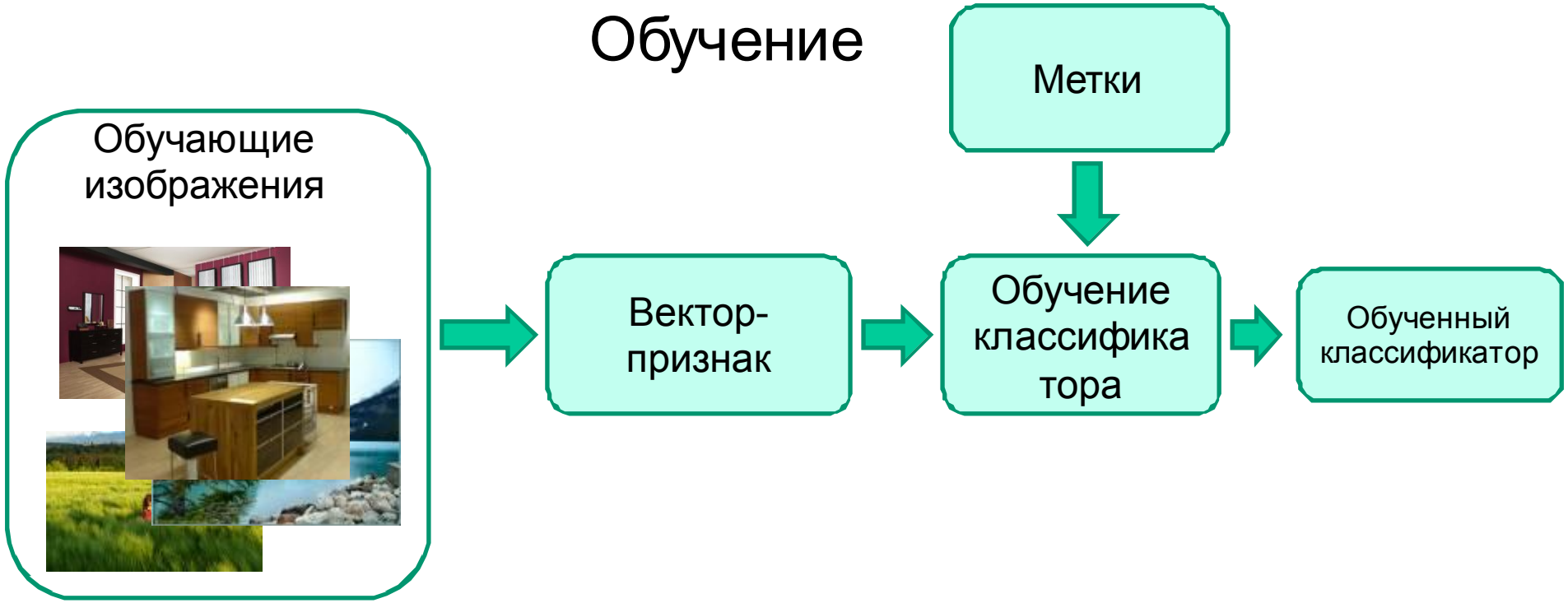
Обученный классификатор

Предсказание

Outdoor

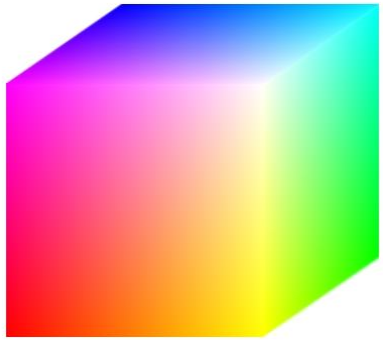
Признаки изображения

Обучение

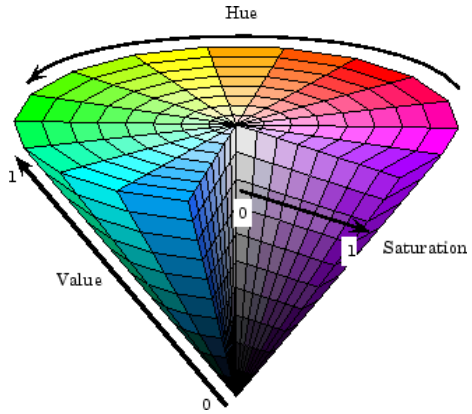


- Мы будем рассматривать в основном признаки изображения
- Но немного коснёмся и методов классификации

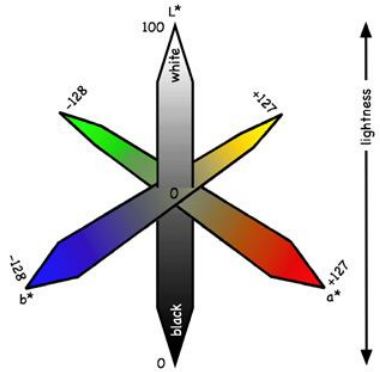
Признаки пикселей



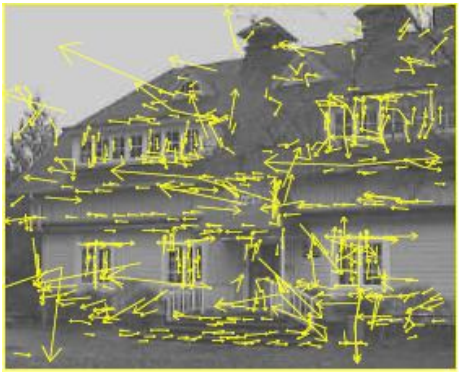
Пространство цветов RGB

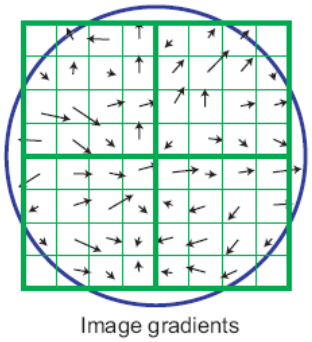


Пространство цветов HSV

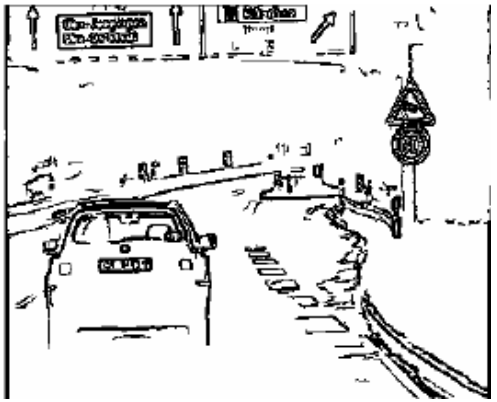


Пространство цветов L*a*b*





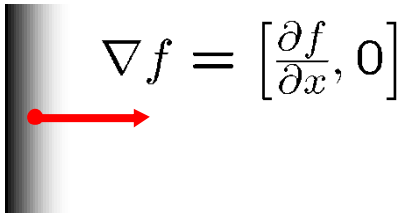
Направление и норма градиента в каждом пикселе

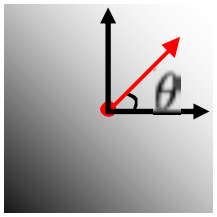


Наличие и ориентация края в каждом пикселе

Градиент изображения

- Градиент изображения: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

- 


$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$


$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

Градиент направлен в сторону наибольшего изменения интенсивности

Направления градиента задается как: $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

- Как направление градиента соответствует направлению края?
- Величина (норма) градиента:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Steve Seitz

Как считать производные?

Яндекс

• Для функции 2х переменных, f(x,y):

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0} \left(\frac{f(x+\varepsilon, y)}{\varepsilon} - \frac{f(x,y)}{\varepsilon} \right)$$

• Разностная производнаяя:

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}$$

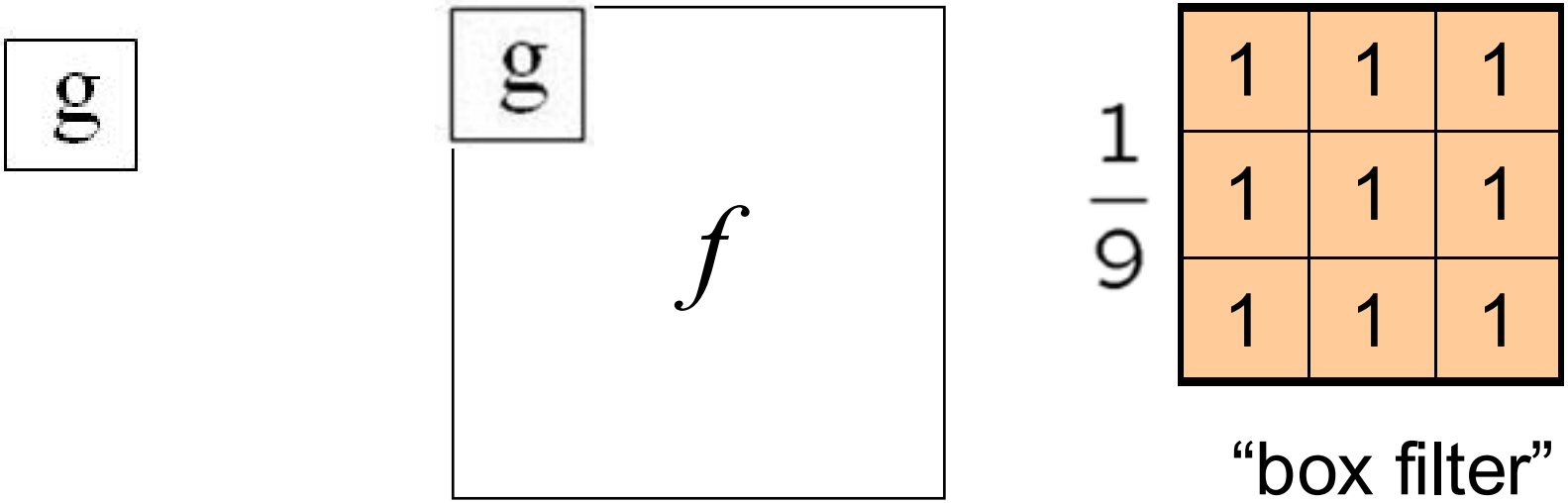
- Можно записать как свёртку (линейную фильтрацию изображения)

Source: D. Forsyth, D. Lowe

Свертка / линейная фильтрация

- «Пространственная фильтрация» - заменим каждый пиксель изображения f взвешенным средним по окрестности
- Веса задаются матрицей g – *ядром фильтра*

$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$

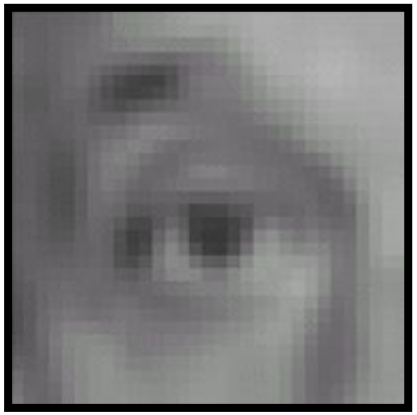


Source: F. Durand

Работа Box-фильтра

Исходное

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$



Результат

Source: D. Lowe

Как считать производные?

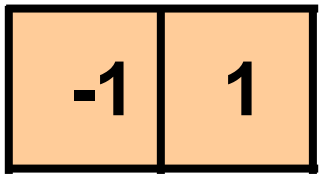
• Для функции 2х переменных, f(x,y):

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0} \left(\frac{f(x+\varepsilon, y)}{\varepsilon} - \frac{f(x,y)}{\varepsilon} \right)$$

• Разностная производнаяя:

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}$$

- Можно записать как свёртку (линейную фильтрацию изображения)

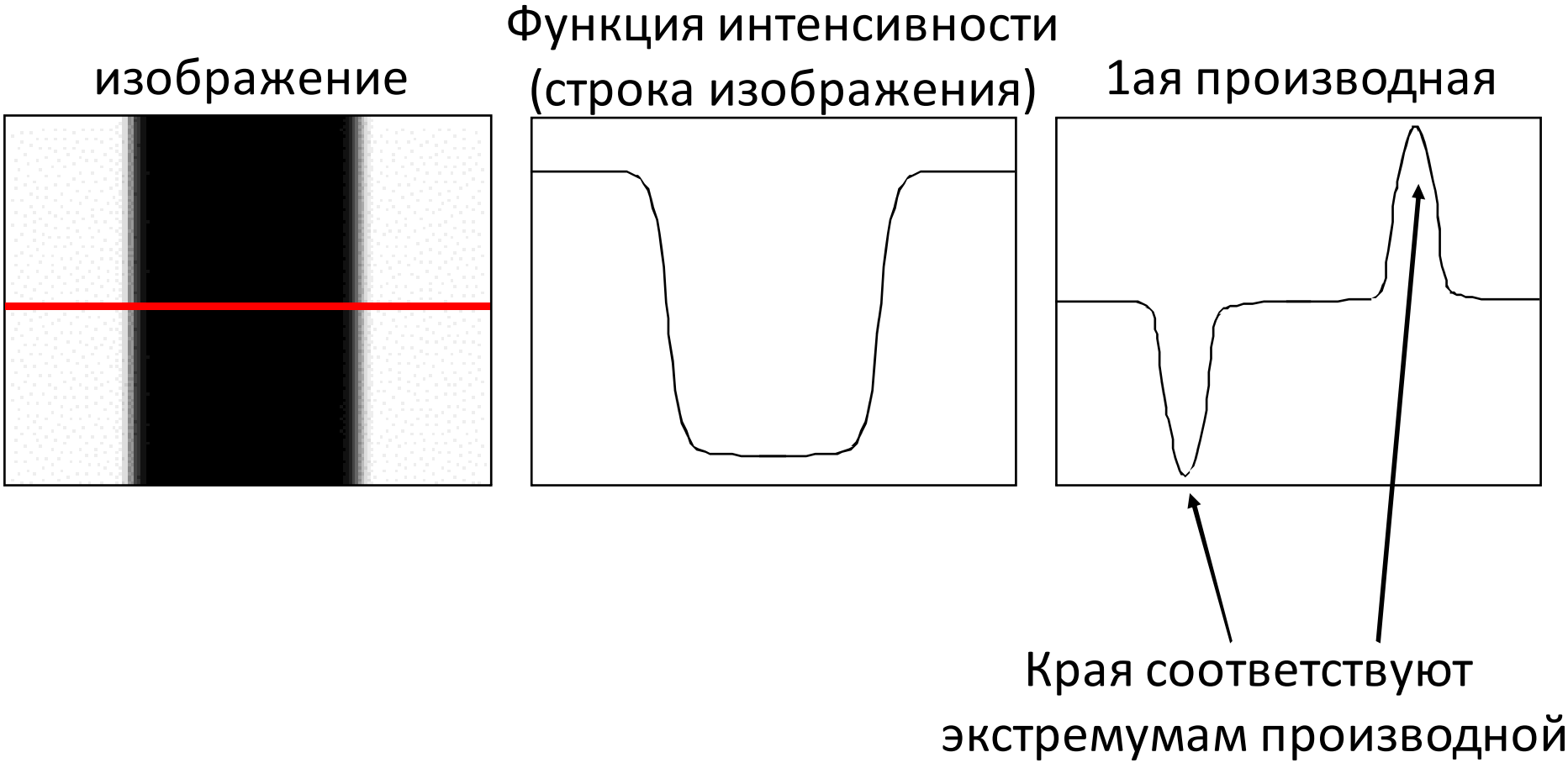


Простейший фильтр

Source: D. Forsyth, D. Lowe

Описание «края»

- Край – это точка резкого изменения значений функции интенсивности изображения

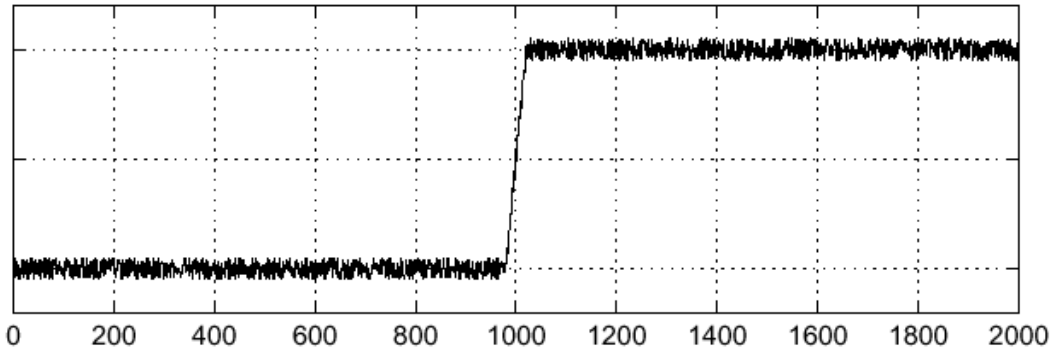


Slide by S. Lazebnik

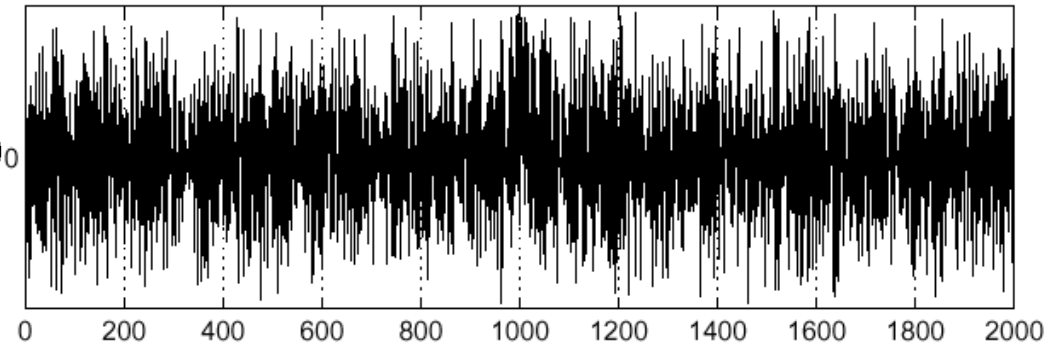
Влияние шума

- Рассмотрим строку или столбец изображения
  - Интенсивность от положения можно рассматривать как сигнал

$f(x)$



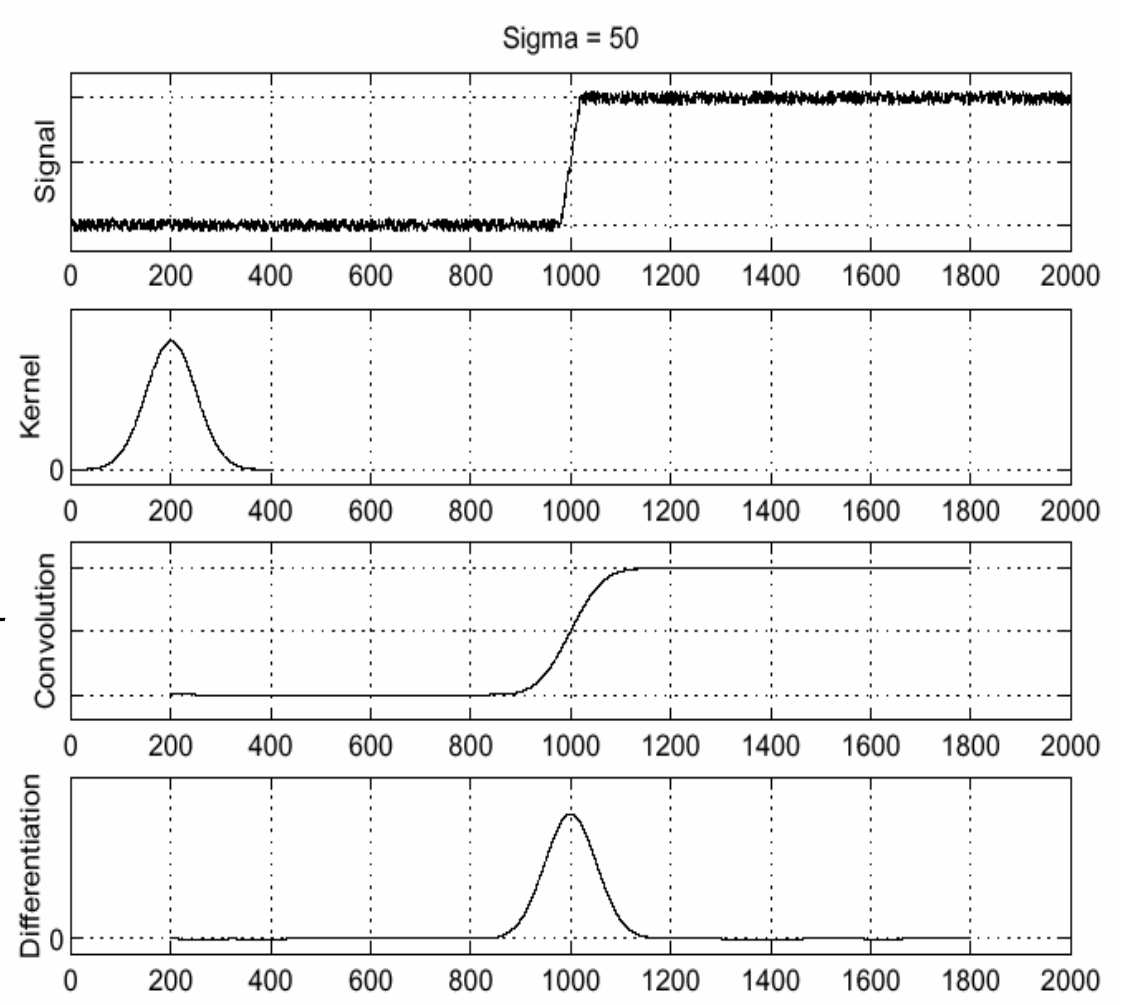
$\frac{d}{dx}f(x)$



Край исчез

Source: S. Seitz

Предобработка (сглаживание)



- Для поиска краев ищем пики в: $\frac{d}{dx}(f * g)$

Source: S. Seitz

Известные фильтры

Яндекс

Несколько фильтров, по разному оценивающие производные по направлению:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Робертса Превитт Собеля

Превитт и Собель чуть-чуть сглаживают шум

Карта силы краев

Яндекс

Примеры:

Робертса

Превитт

Собеля

Если взять локальные максимумы градиента, то получим края в изображении

Пример

- Исходное изображение (Lena)

Slide by S. Lazebnik

Пример

Норма градиента

Slide by S. Lazebnik

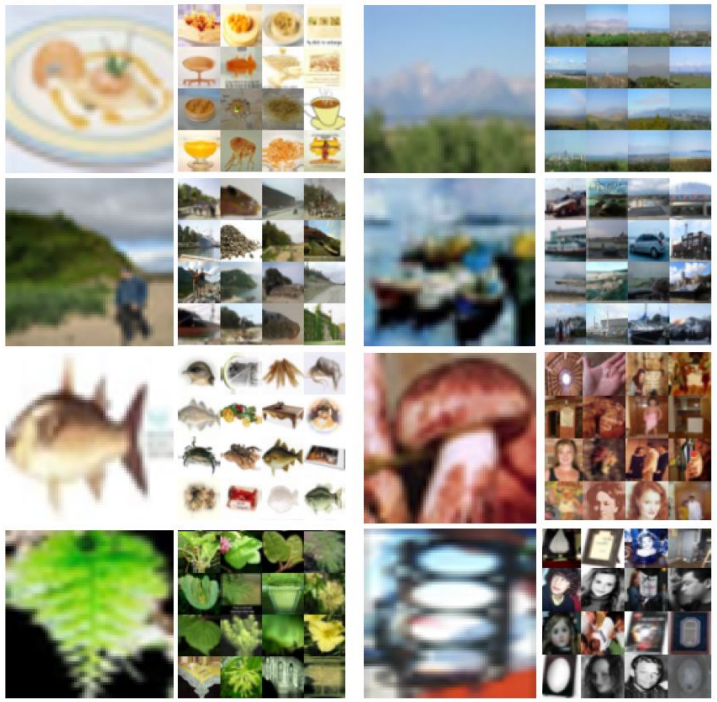
Пример

Утоньшение

(non-maximum suppression)

Slide by S. Lazebnik

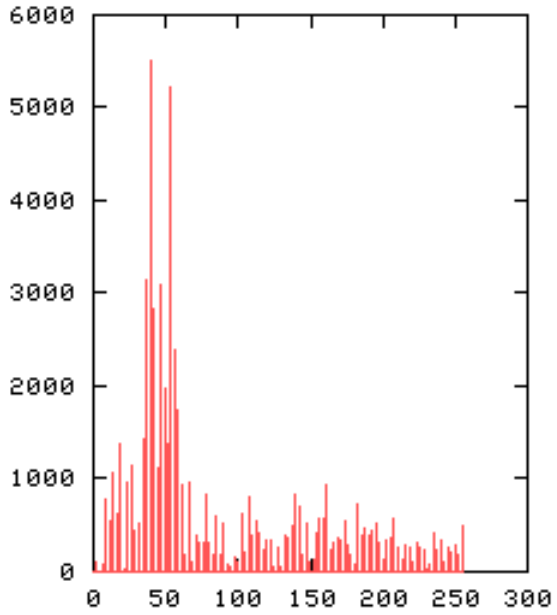
Использование пикселей напрямую

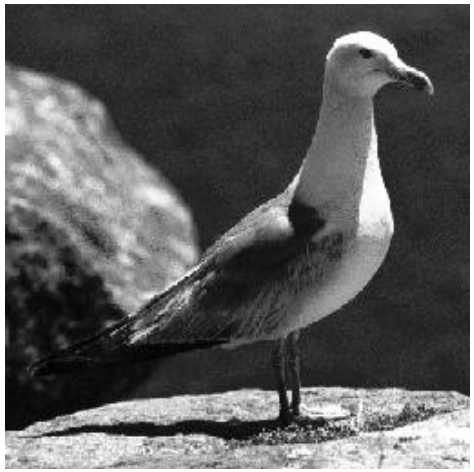


- Можно ли использовать признаки всех пикселей напрямую?
- Можно, если все изображения для классификации будут одинакового размера
- Нормализуем изображения, т.е. приведем их к одному размеру
- Вытянем изображение в вектор, признаки пикселей будут элементами вектора

Для распознавания, изображение должно описываться вектор-признаком фиксированной длины!

Гистограммы





- Гистограмма – это график распределения значений признака на изображении. На горизонтальной оси – значение признака, на вертикальной оси - число пикселей с заданным значеним.
- Мы можем измерять разные признаки (features) изображения
  - Цвет, края, градиенты, текстуру и т.д.
- Гистограммы – стандартный способ непараметрического описания распределения признаков

Images from Dave Kauchak

Квантование признаков

Яндекс

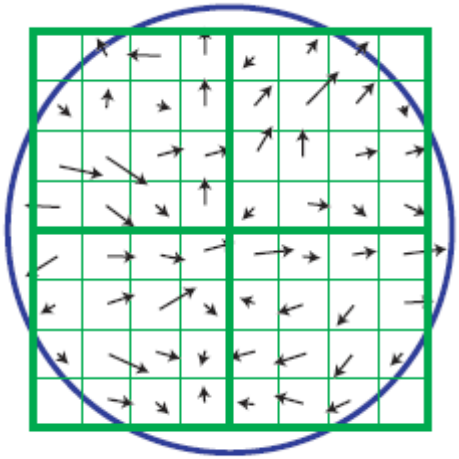
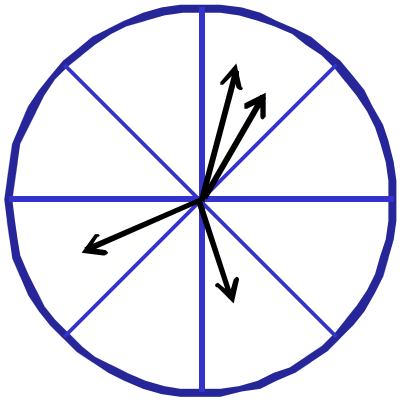
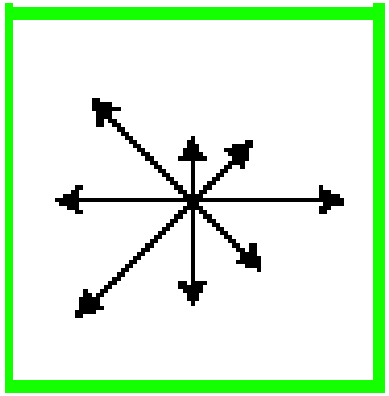


Image gradients



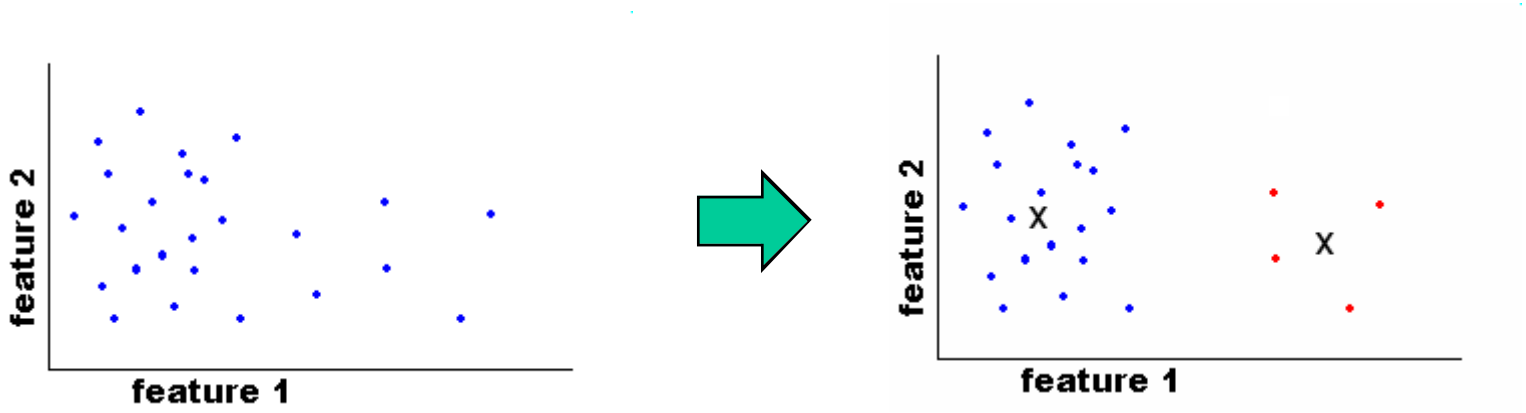


- Как построить гистограмму для признаков, которые изменяются непрерывно?
- Или разрешение слишком большое?
- «Квантование» - дискретизация признаков
  - Пр: все направления градиента разбиваем на 8 вариантов
  - Каждый вариант нумеруем от 1 до 8

Квантование признаков

- Есть фотографии тигров и медведей
- Хотим научиться отличать одни от других по цвету
- Оптимально ли использовать гистограмму цветов на интервале от [0,255]?
- Не все цвета встречаются!

Адаптивное квантование

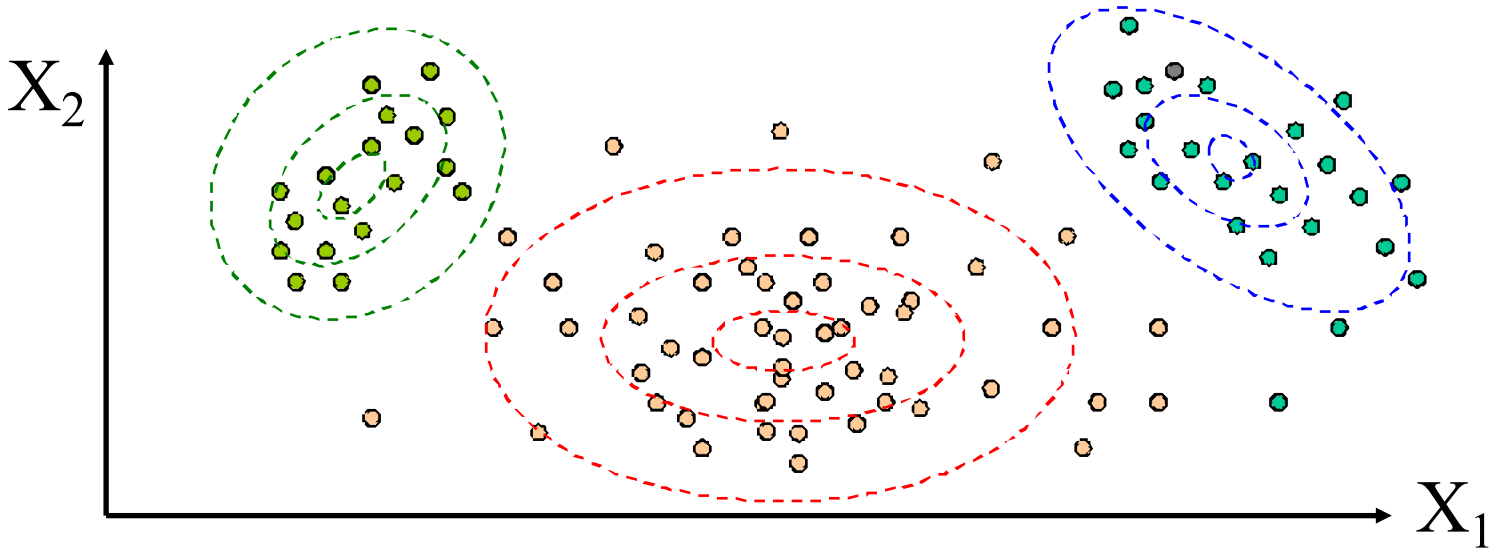


- Хотим разбить элементы (пр. пиксели) на часто встречающиеся группы
- Затем можем каждый элемент сопоставить своей группе (дать ему номер группы)
- Тогда будем строить гистограммы частот принадлежности элементов группам
- Кластеризация!

Кластеризация

Яндекс

- **Кластерный анализ** (Data clustering) — задача разбиения заданной выборки *объектов* (ситуаций) на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались.
- Одна из задач «обучения без учителя»



Кластеризация K-средними

Яндекс

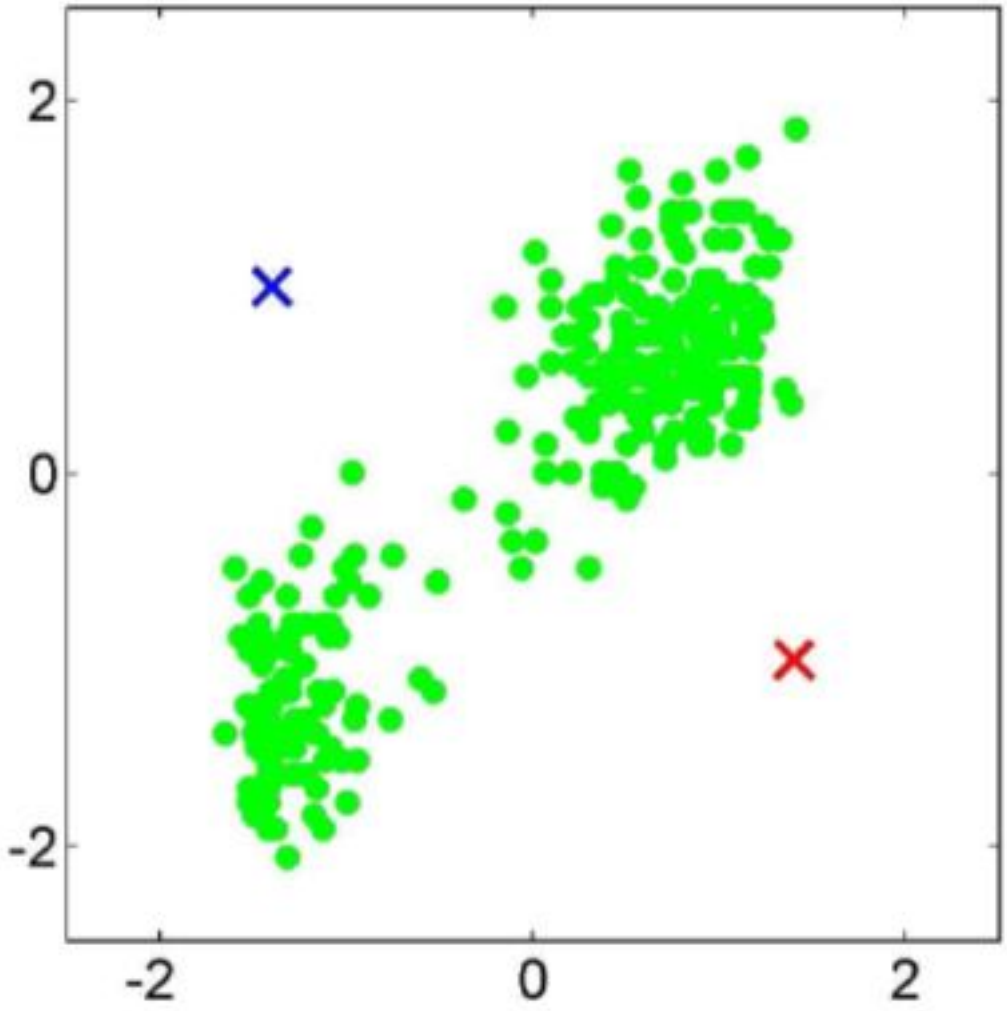
- Минимизируем сумму квадратов Евклидовых расстояний между точками x_i и ближайшими центрами кластеров m_k

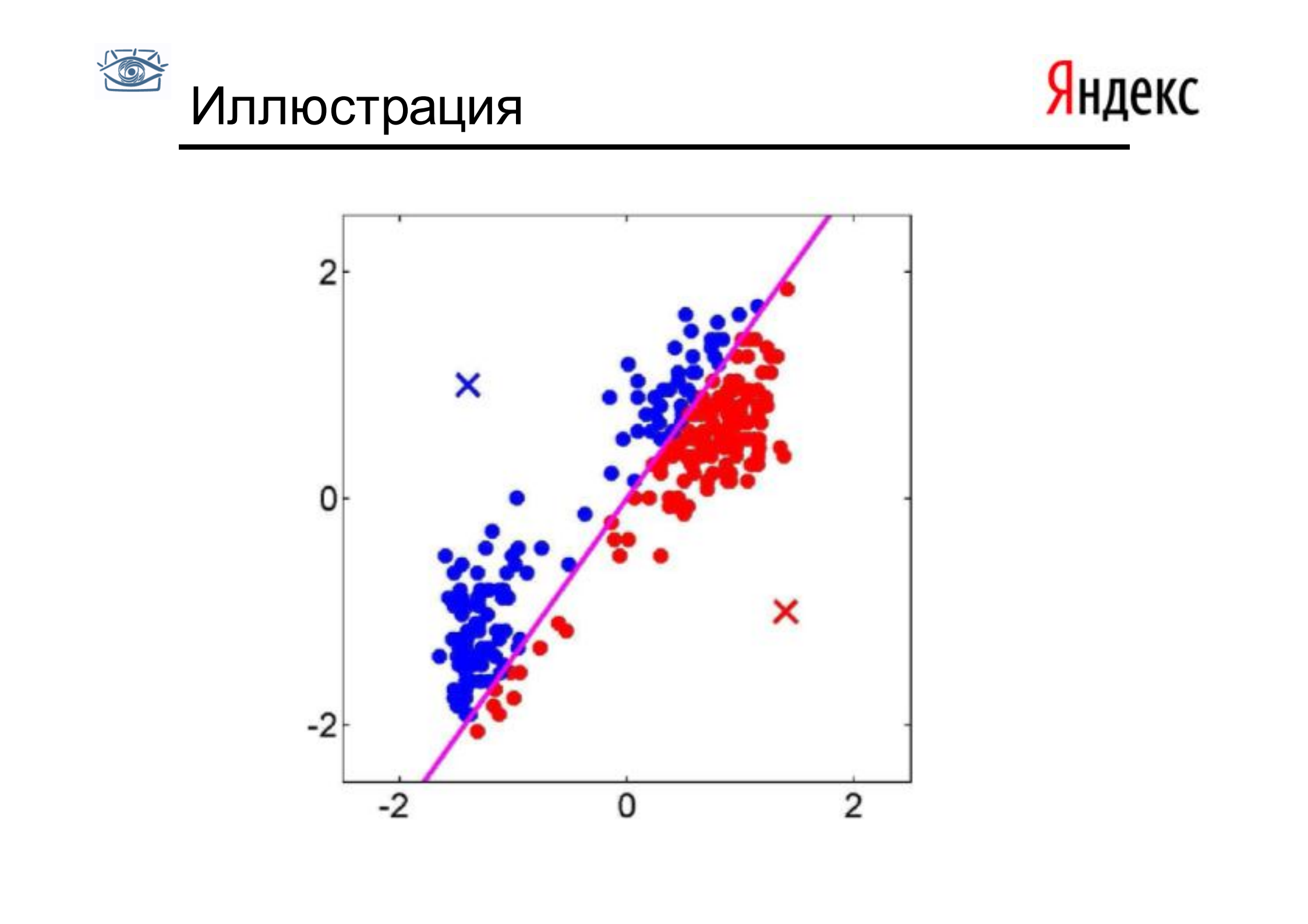
$$D(X,M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (x_i - m_k)^2$$

- Алгоритм:
- Случайно инициализируем K центров кластеров
- Повторяем до сходимости:
  - Назначаем каждую точку ближайшему центру
  - Пересчитываем центр каждого кластера как среднее всех назначенных точек

Иллюстрация

Яндекс





Иллюстрация

Яндекс

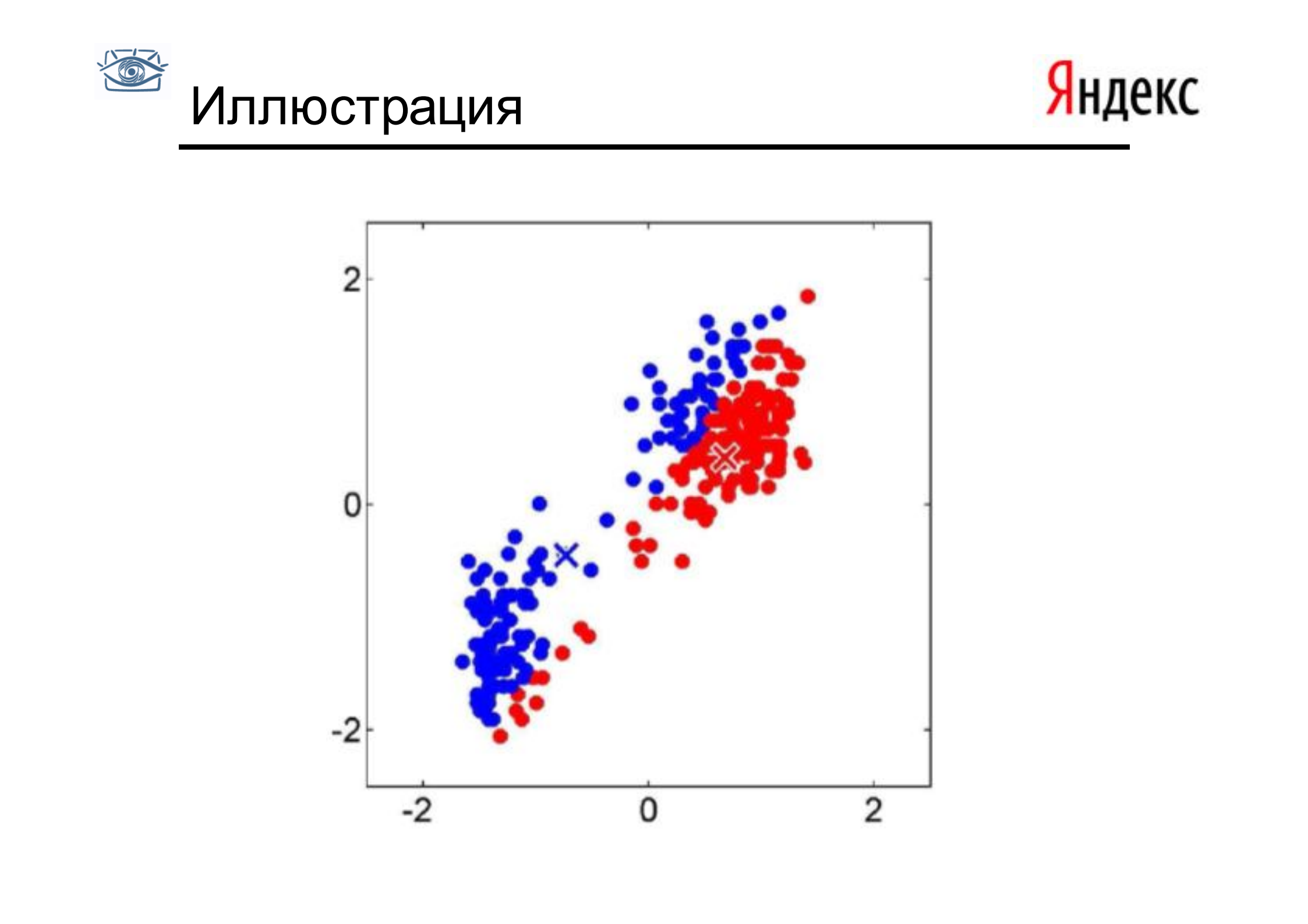
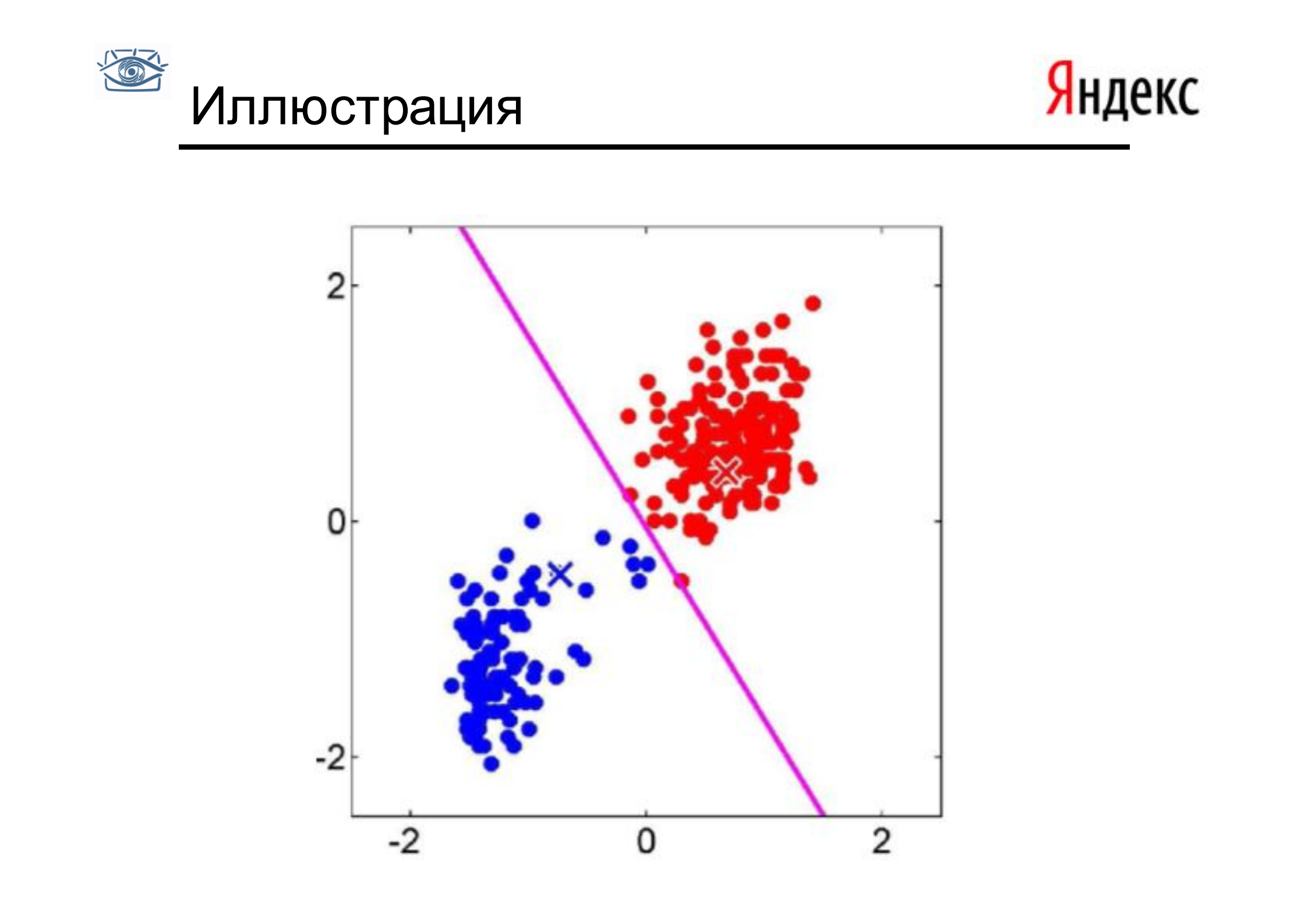
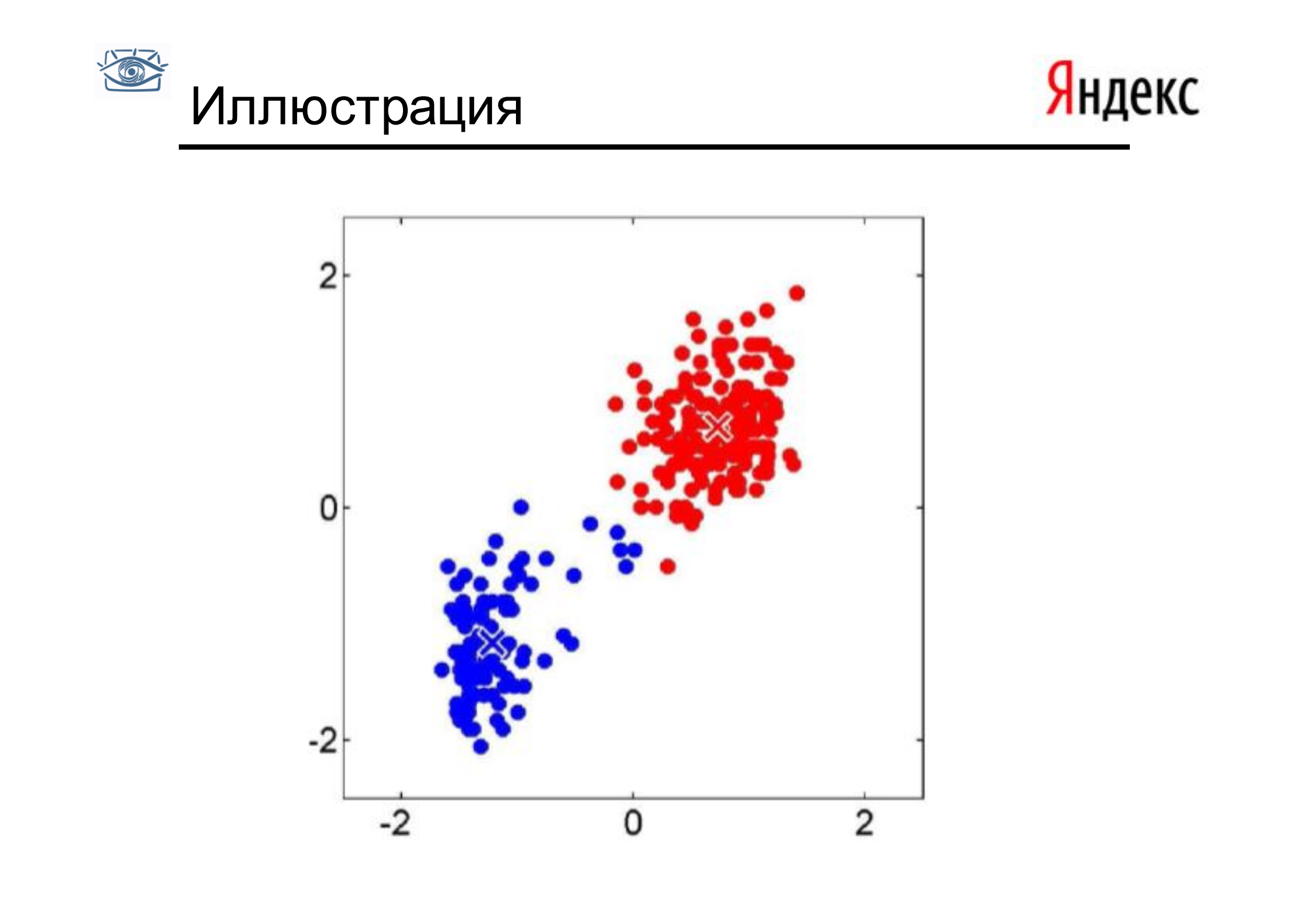


Иллюстрация
Яндекс
2
0
-2
-2
0
2



Иллюстрация

Яндекс



Иллюстрация

Яндекс

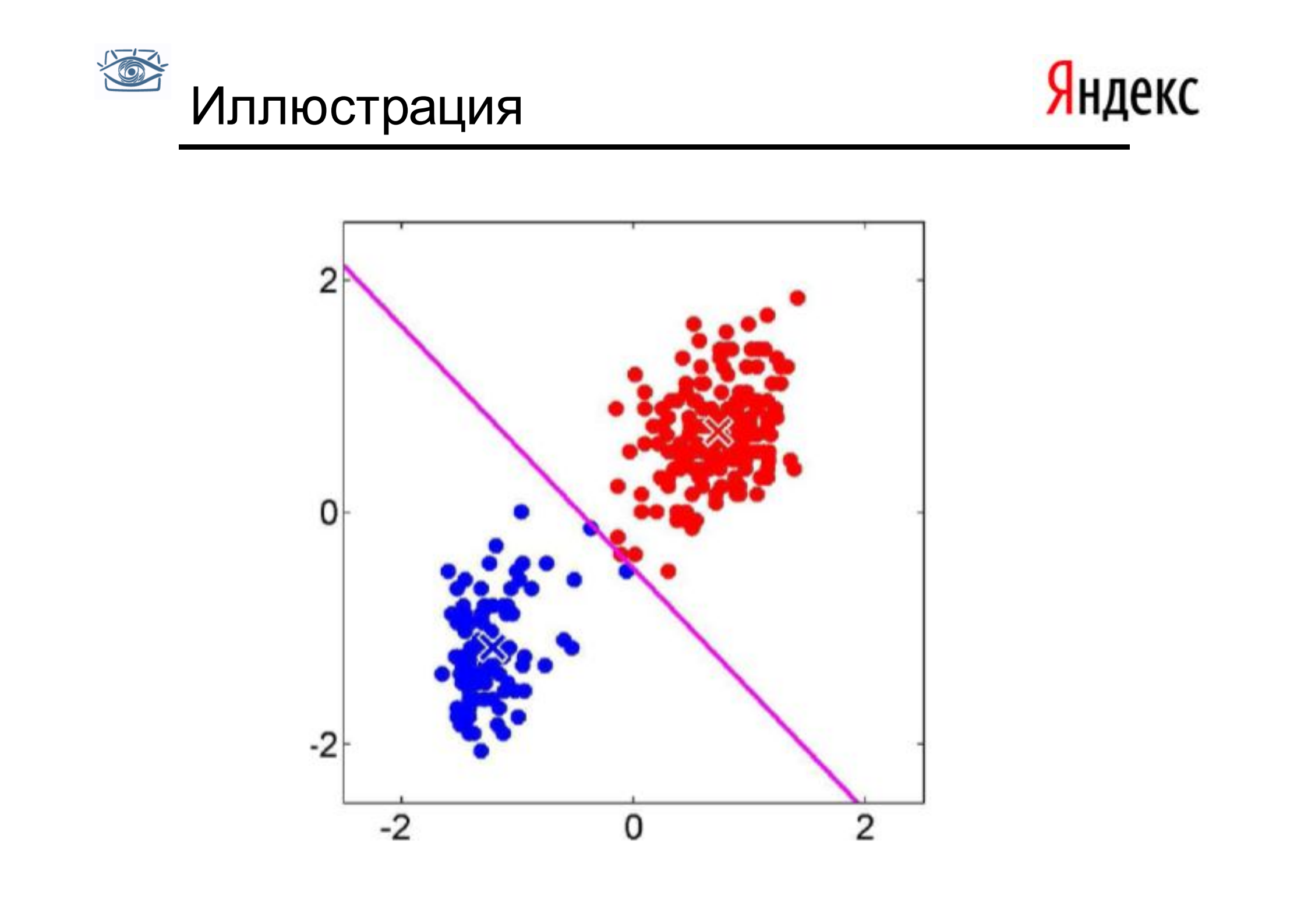
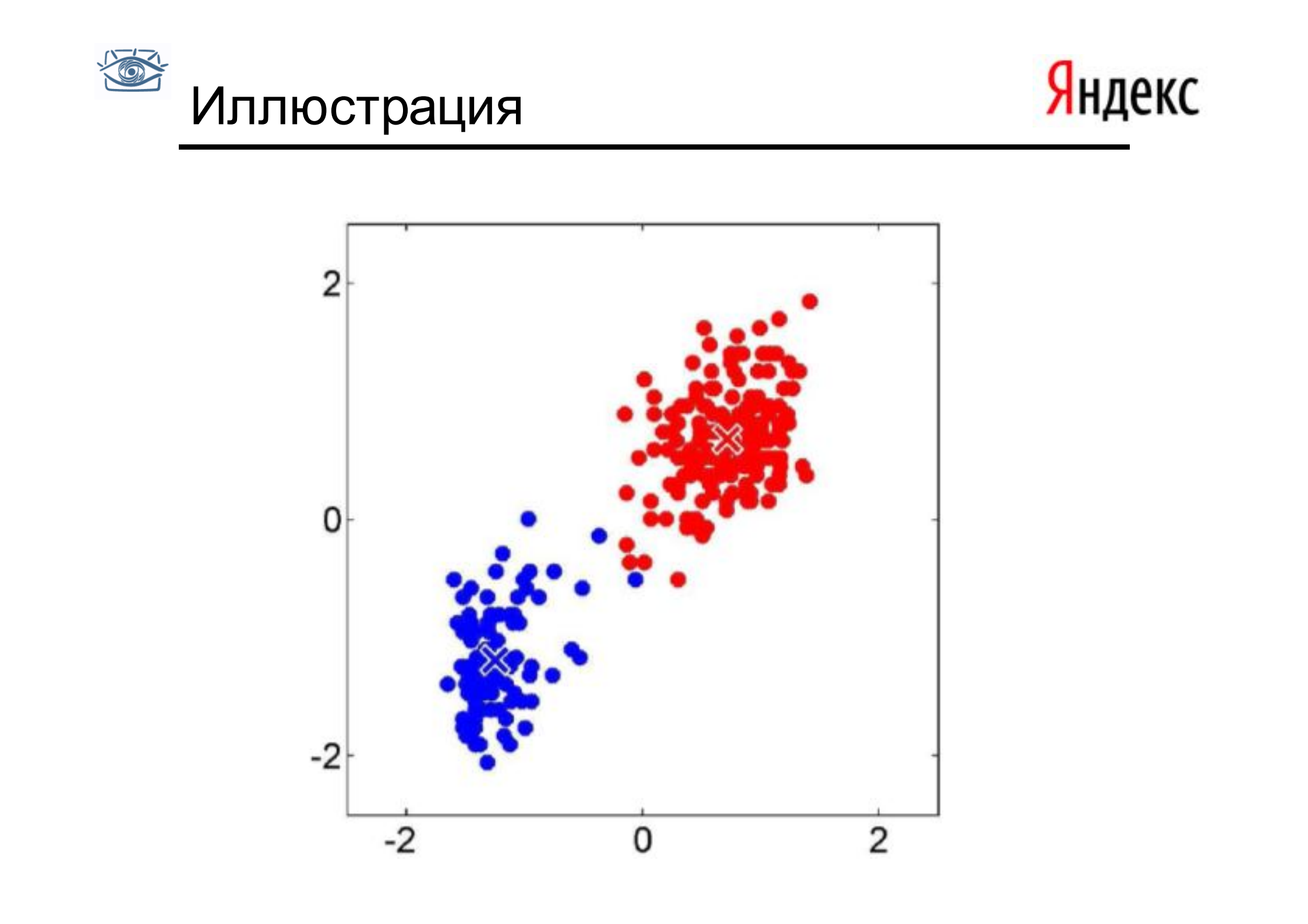
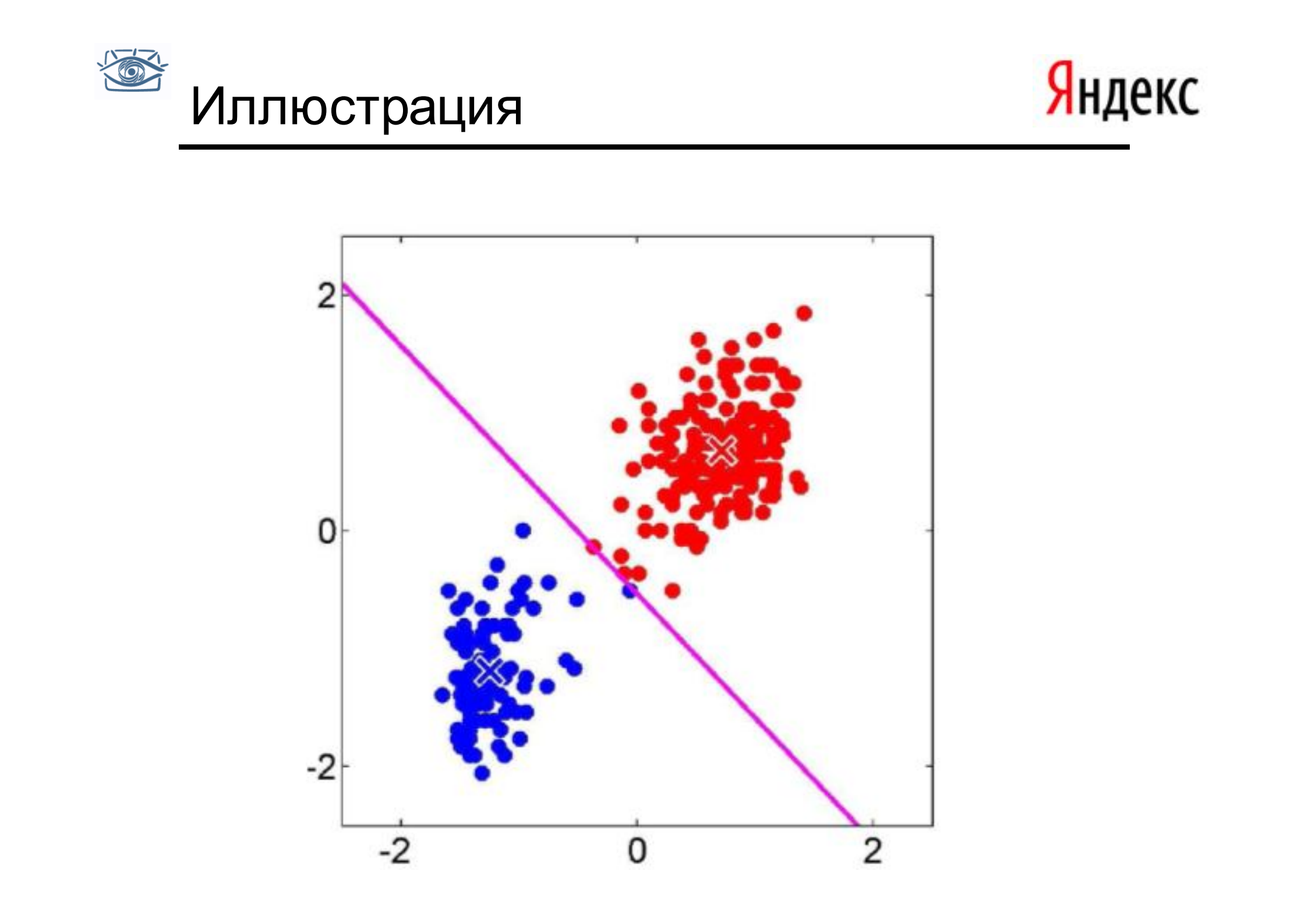


Иллюстрация
Яндекс
2
0
-2
-2
0
2

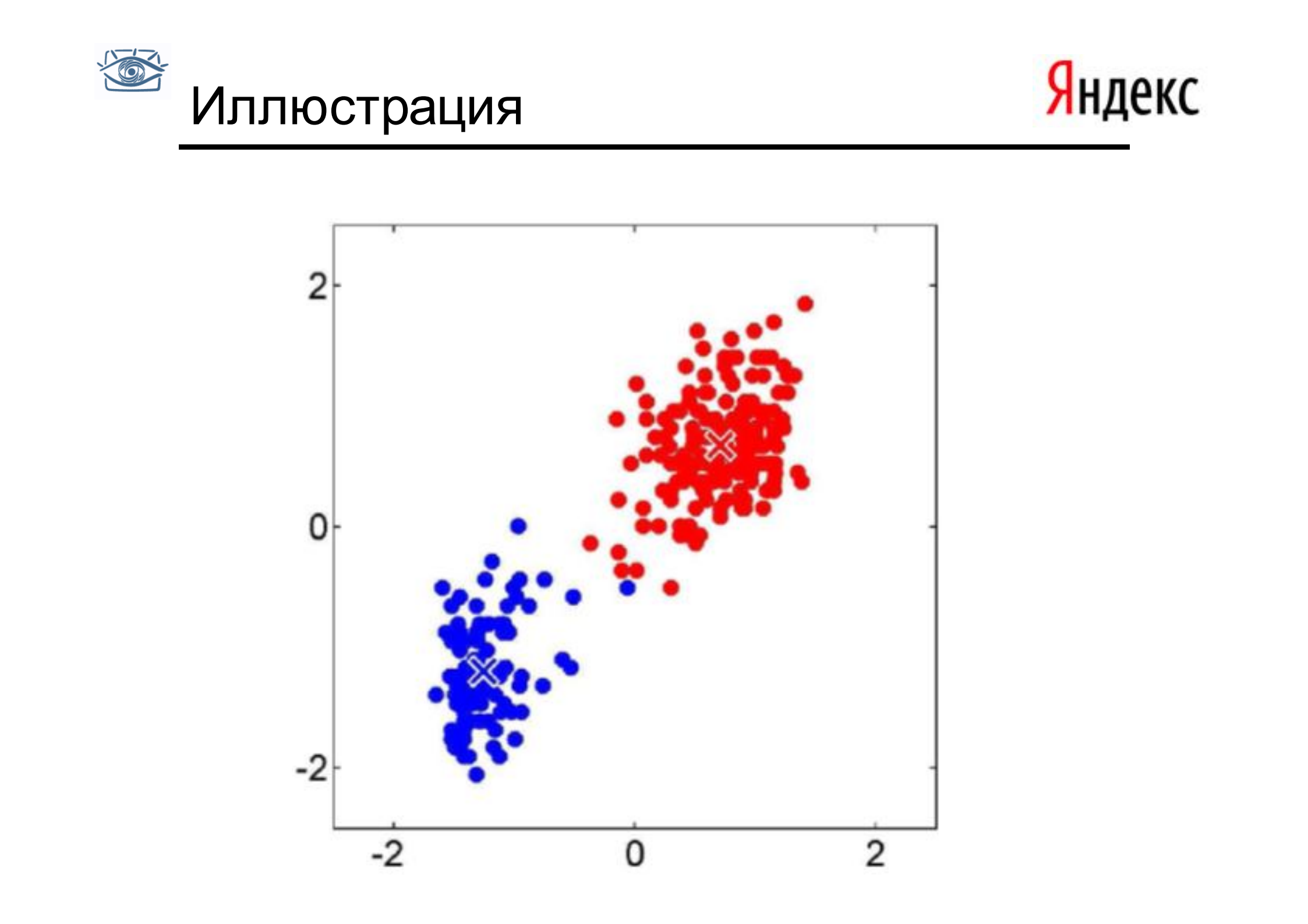


Иллюстрация



Иллюстрация

Яндекс



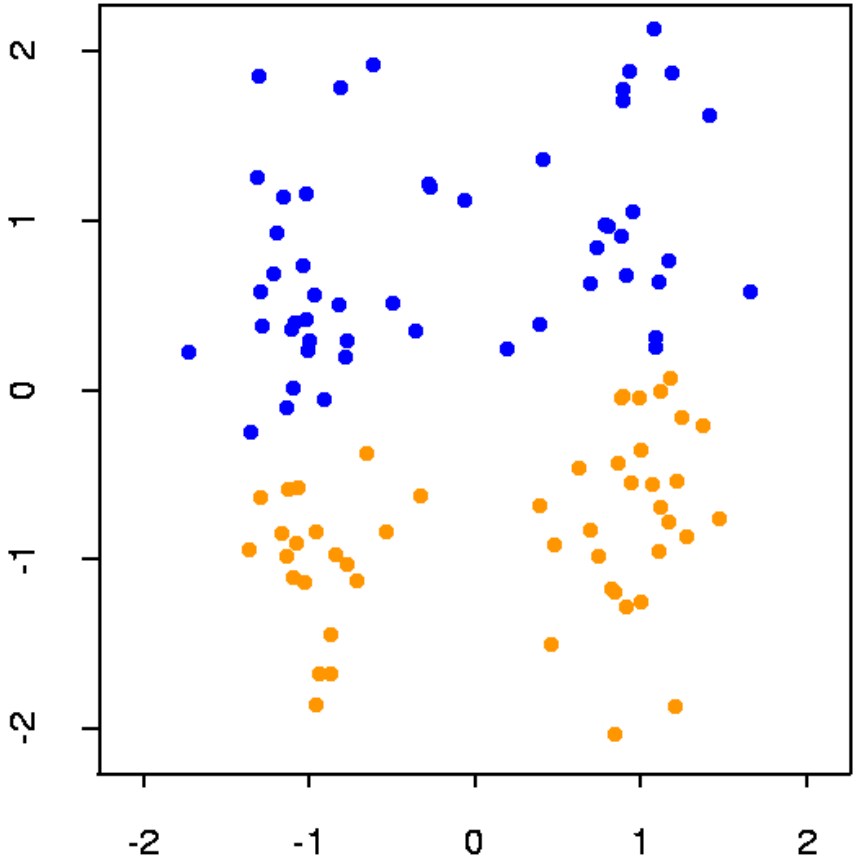
Иллюстрация

Яндекс

Алгоритм К-средних

Яндекс

- Однопараметрический
  - Требует знания только о количестве кластеров
- Рандомизирован
  - Зависит от начального приближения
- Не учитывает строения самих кластеров
- Часто применяется



Для нашего примера

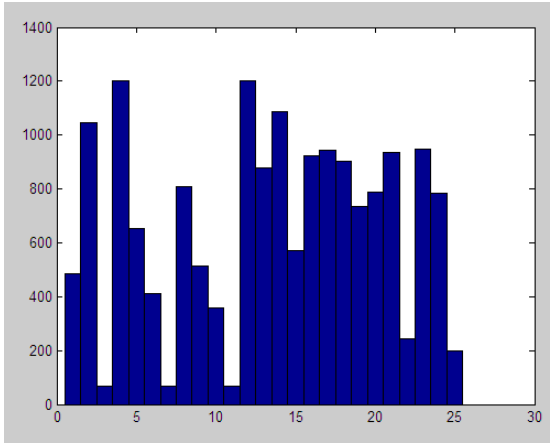
Яндекс

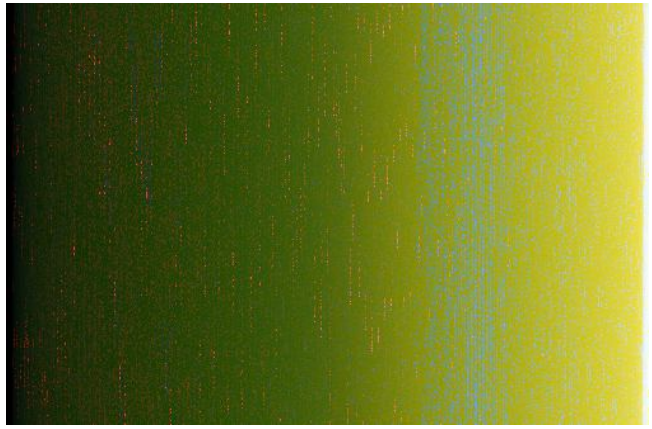
- Зададим количество кластеров (например, 100)
- Кластеризуем
- Будут кластеры в белой, зеленой, черной, оранжевой областях

Для нашего примера

Яндекс

- Зададим количество кластеров (например, 100)
- Кластеризуем
- Будут кластеры в белой, зеленой, черной, оранжевой областях





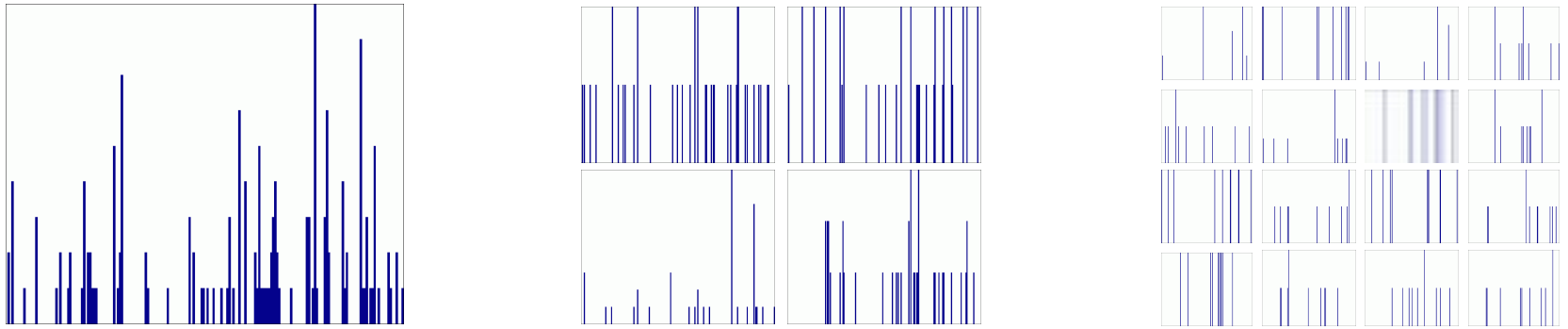
Пространственное распределение

У всех этих трех изображений похожие гистограммы цветов

Пространственная пирамида

Яндекс

Вычислим гистограму в каждом блоке и объединим все гистограммы в один вектор-признак



Уровень 0 Уровень 1 Уровень 2

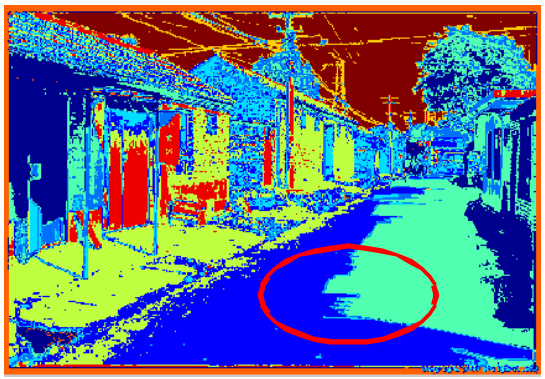
S. Lazebnik et. Al. **Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories** *CVPR 2006*

Общая схема

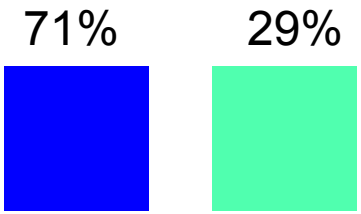
Яндекс

Вычислим признаки по изображению → Квантуем признаки → Определим область для вычисления → Агрегируем признаки

RGB



Квантование на 10 уровней



Посчитанная гистограмма

Features → Coding → Spatial pooling

Мешок слов

Классификация текстов

Яндекс

- Представление документа без порядка: частоты слов из словаря («мешок слов») Salton & McGill (1983)

Классификация текстов

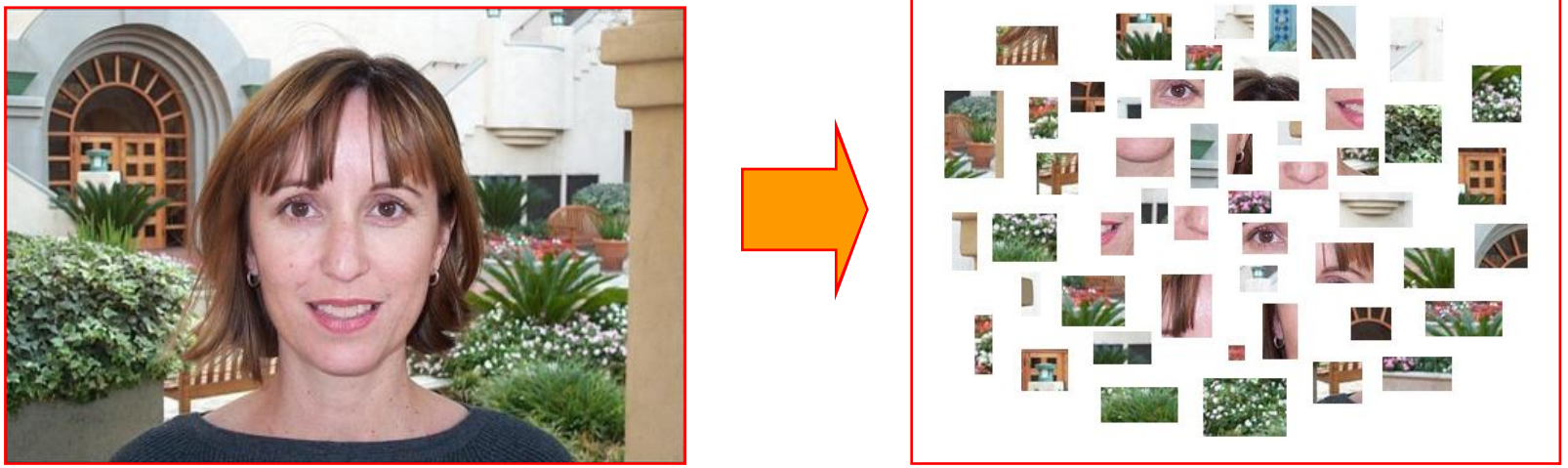
- Представление документа без порядка: частоты слов из словаря («мешок слов») Salton & McGill (1983)

Классификация текстов

Яндекс

- Представление документа без порядка: частоты слов из словаря («мешок слов») Salton & McGill (1983)

«Визуальные слова»



- Что в нашем случае «слово» и «словарь»?
- «Визуальное слово» – часто повторяющийся фрагмент изображения («visual word»)
- В изображении визуальное слово может встречаться только один раз, может ни разу, может много раз

Sivic, J. and Zisserman, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. ICCV 2003

«Визуальный словарь»

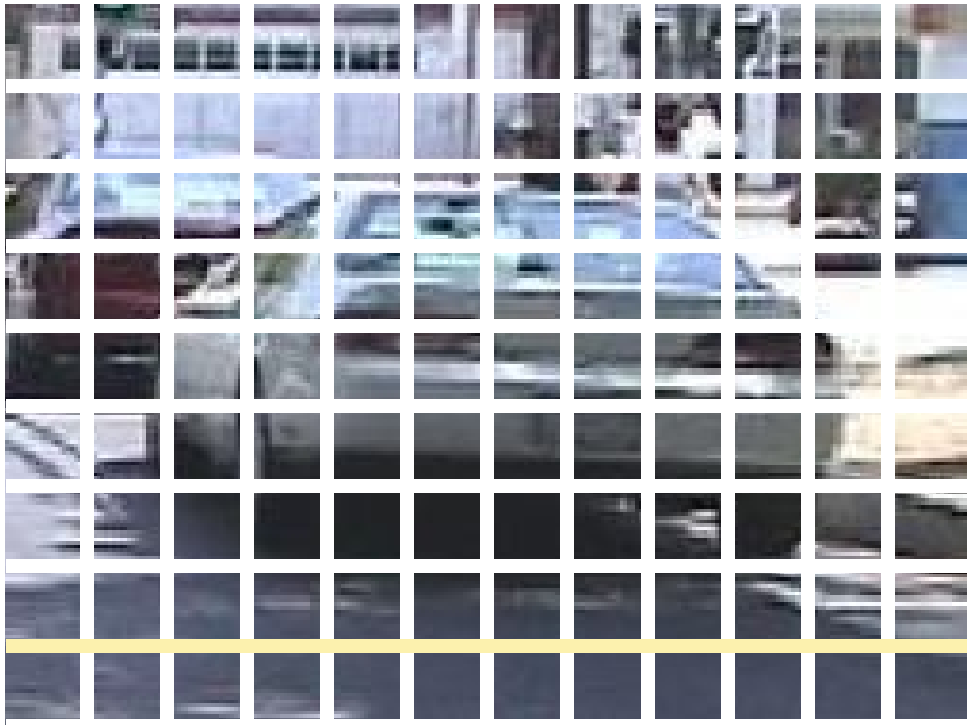
Яндекс

- Словарь – набор фрагментов, часто повторяющихся в коллекции изображений

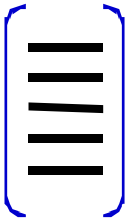
- Как составить словарь?
  - Составить большой список всех фрагментов по всей коллекции
  - Разделить весь список на похожие группы
  - Будем считать все фрагменты в одной группе – «экземплярами» одного и того же слова

1. Извлечение фрагментов

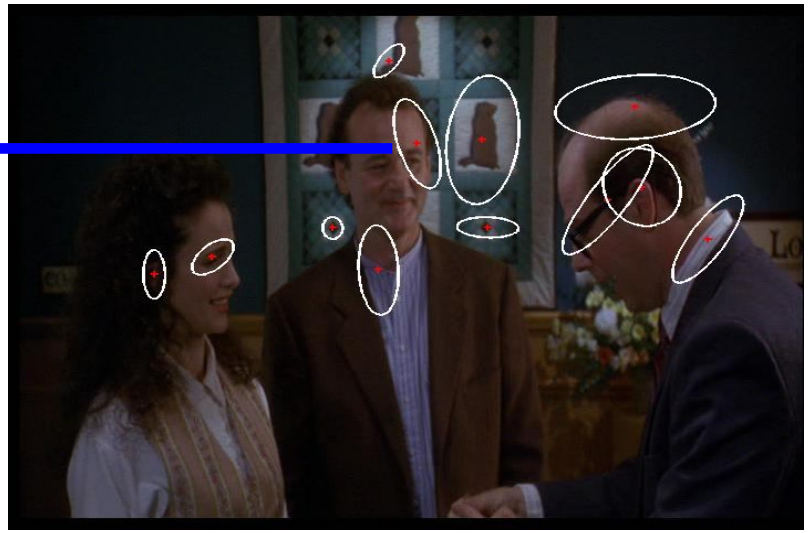
- Регулярная сетка
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005



Вычисление признаков



Вычисление вектор-признака



Выбранные фрагменты

- Для каждого выбранного фрагмента вычисляем вектор-признак
- Чаще всего используется SIFT или цветной SIFT
- Но используют и другие дескрипторы

Slide credit: Josef Sivic

Гистограммы градиентов SIFT

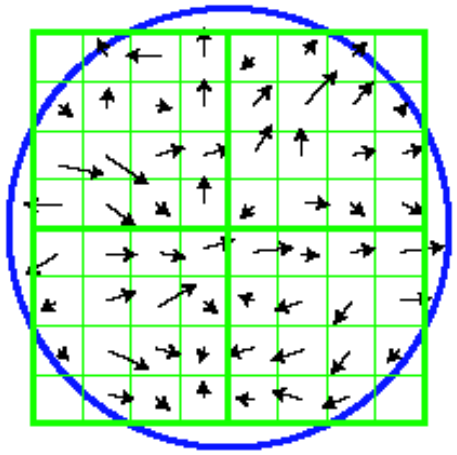
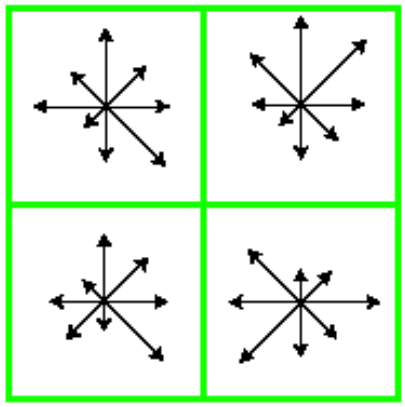
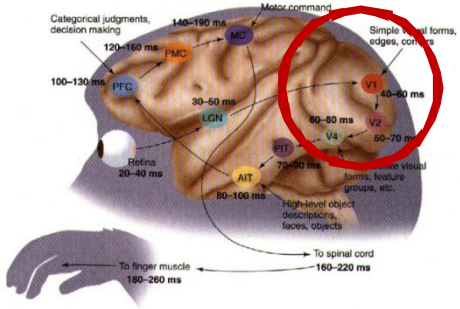
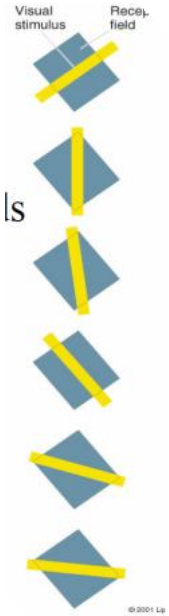


Image gradients



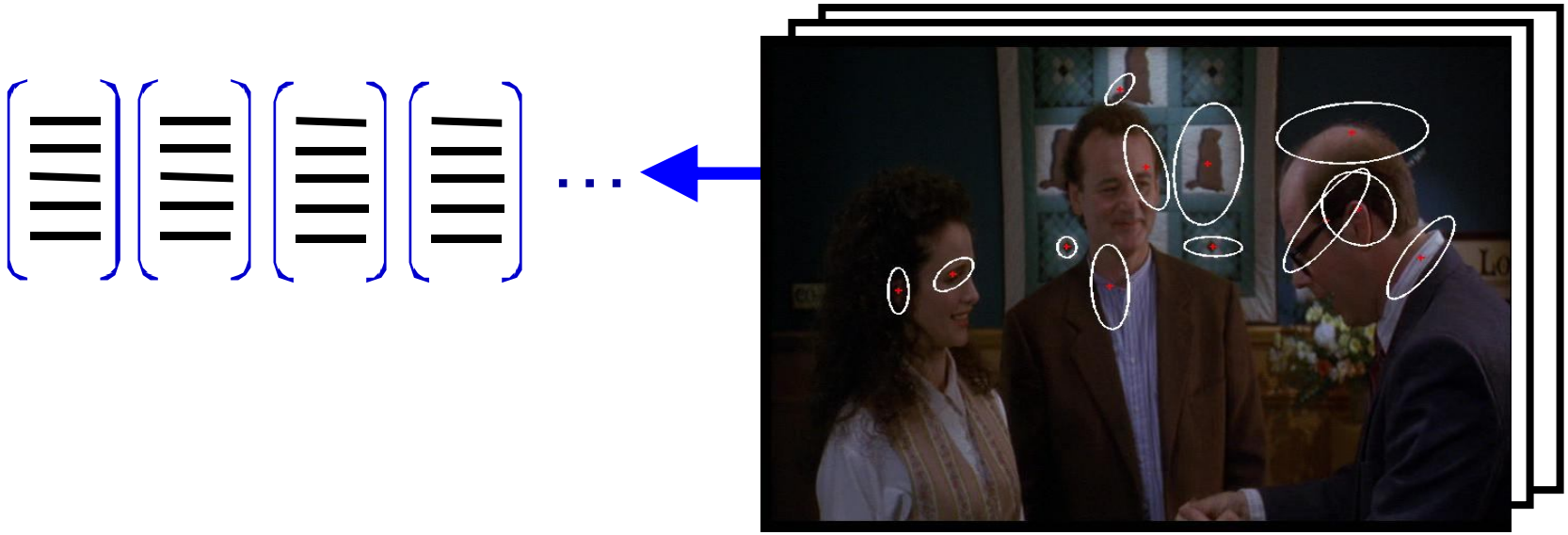
Keypoint descriptor





- Вычисляем градиент в каждой точке
- Строим гистограммы по окрестностям
- Обычно – сетка 4x4, в каждой гистограмма с 8ю ячейками
- Стандартная длина вектора-дескриптора – 128 (4*4*8)
- Устойчив к изменениям освещенности и небольшим сдвигам
- **Считается одним из самых эффективных дескрипторов для поиска изображений**

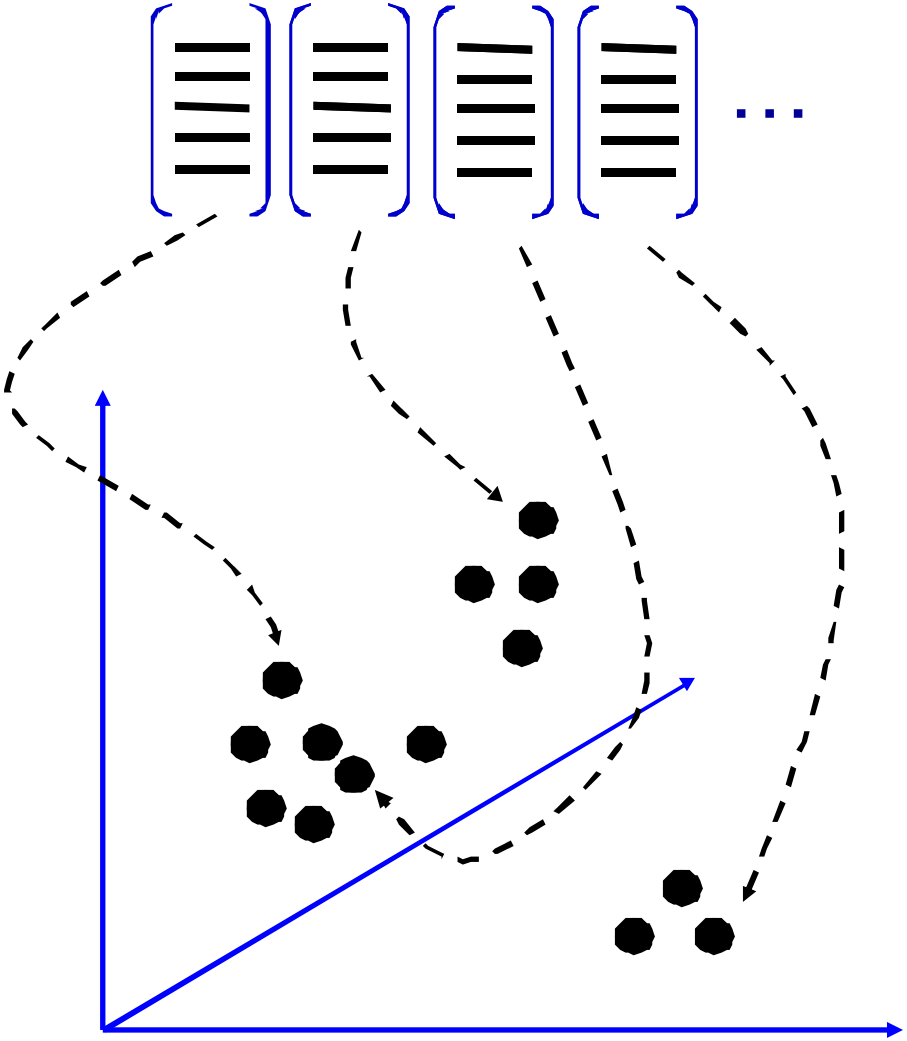
Вычисление признаков

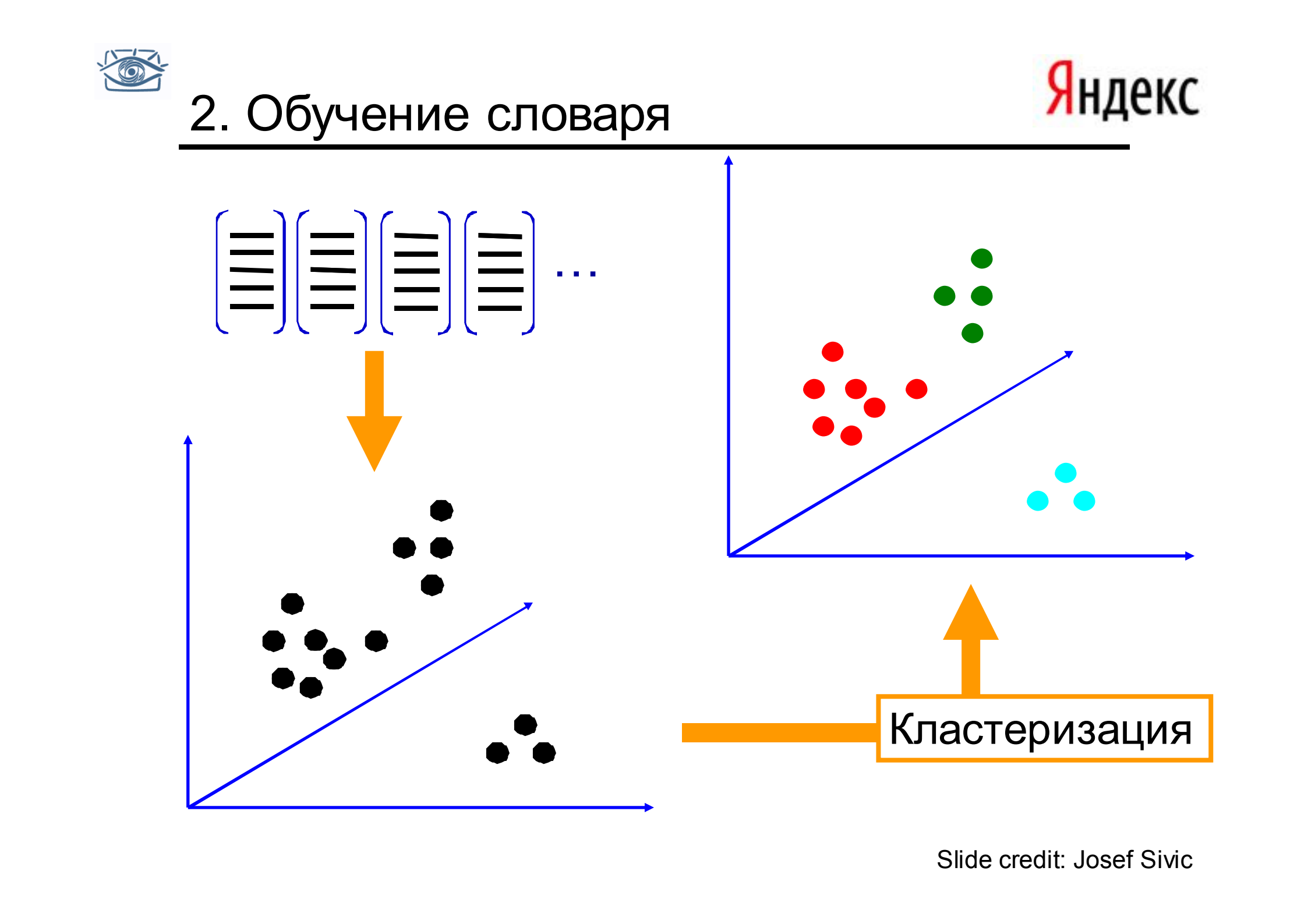


Неупорядоченный список дескрипторов (вектор-признаков)
для всех фрагментов из всех изображений из базы

2. Обучение словаря

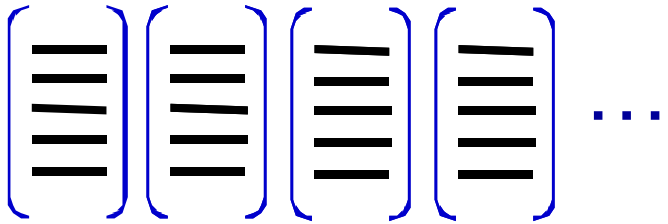
Яндекс

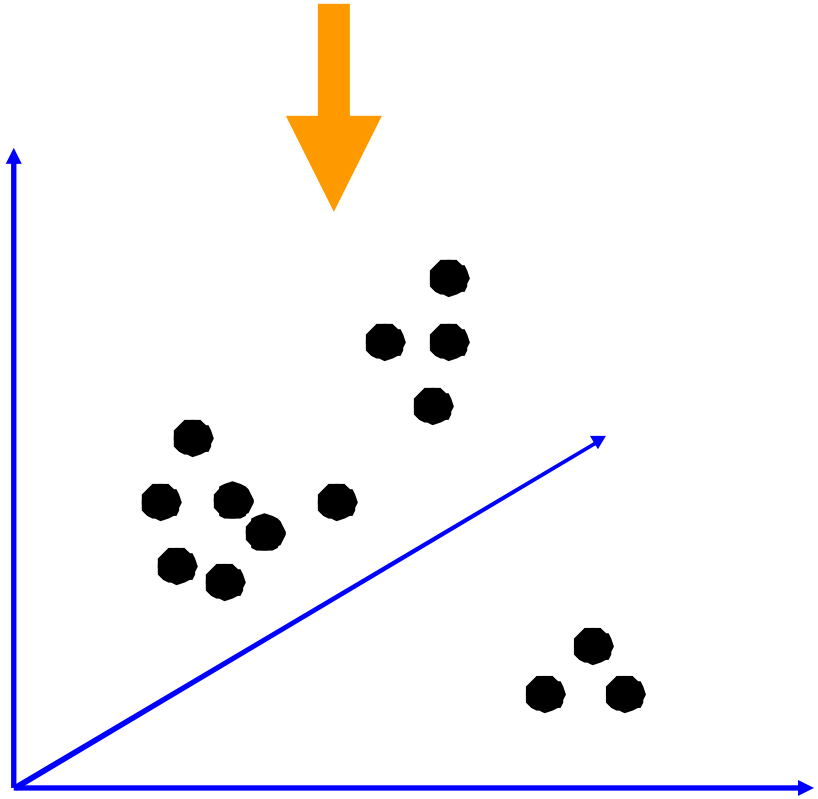


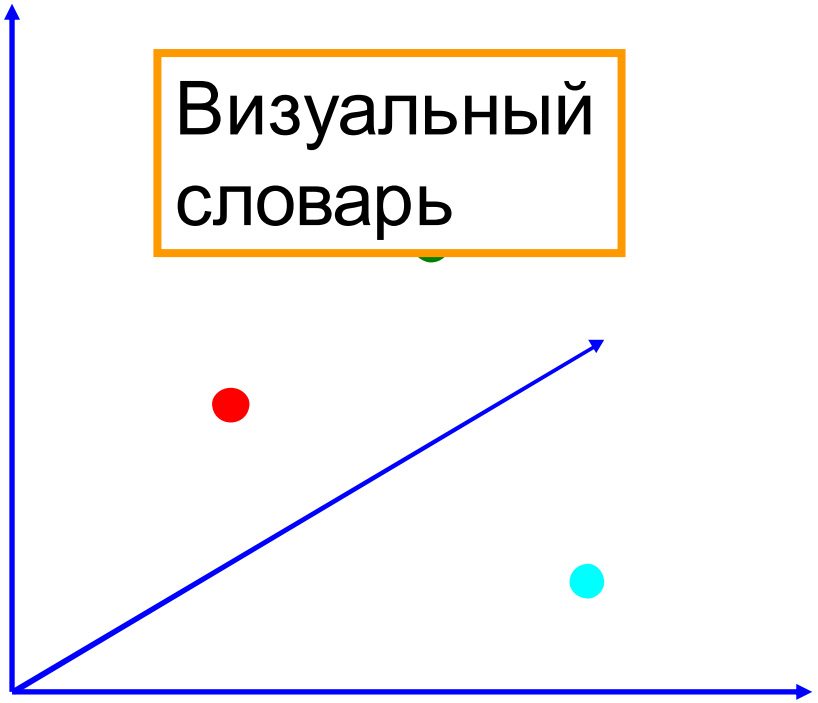

2. Обучение словаря
Яндекс
...
Кластеризация
Slide credit: Josef Sivic

2. Обучение словаря

Яндекс



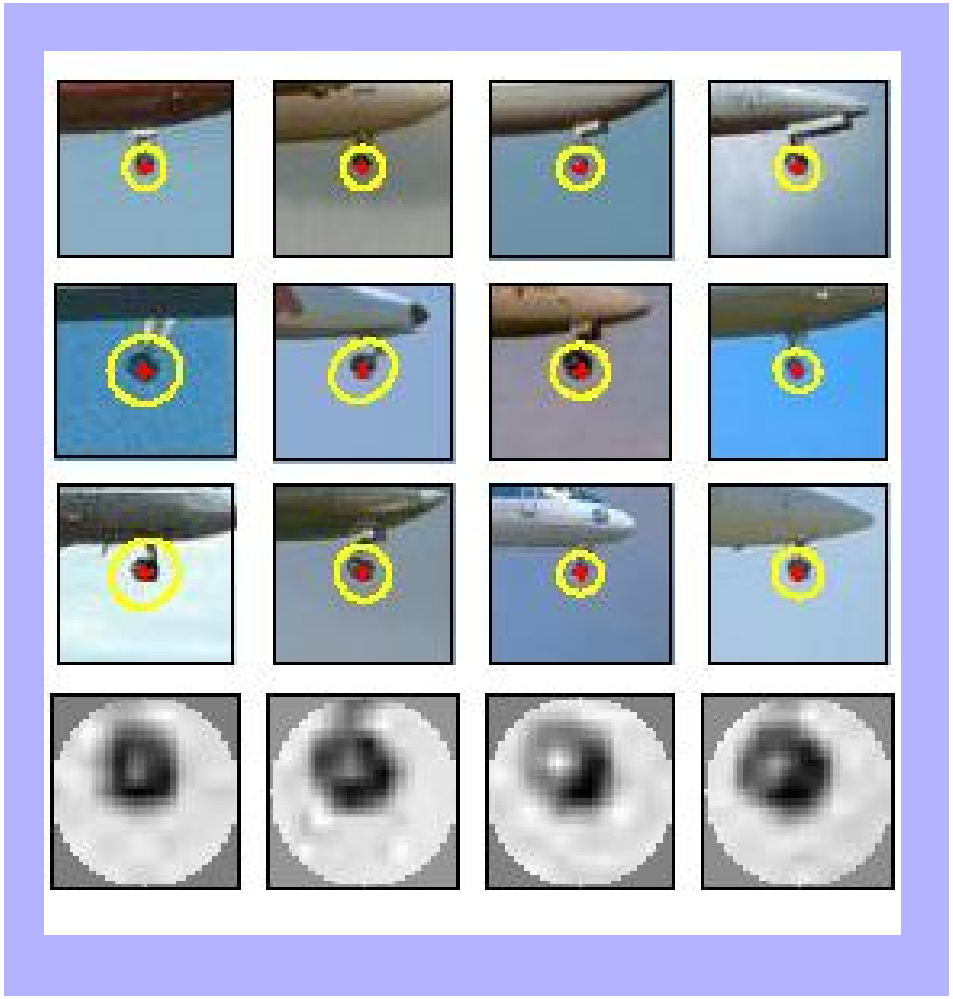


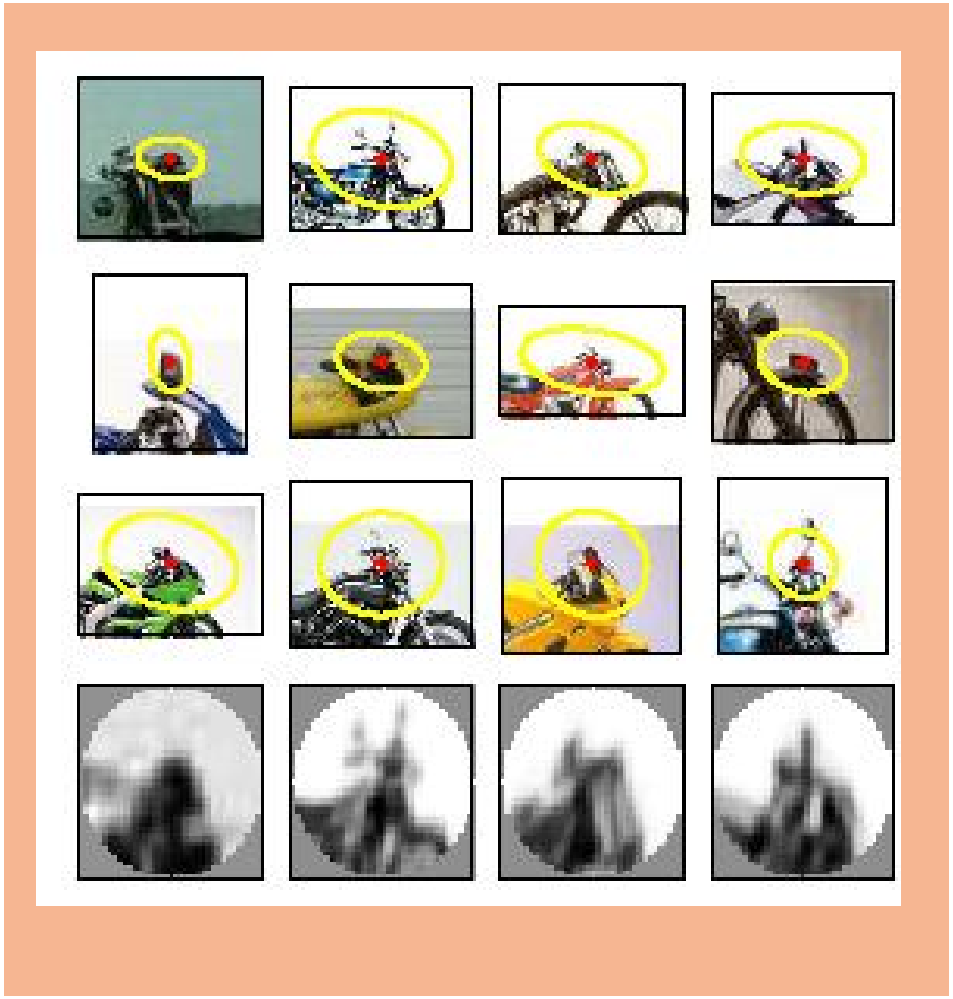


Slide credit: Josef Sivic


Пример словаря
Яндекс
Fei-Fei et al. 2005

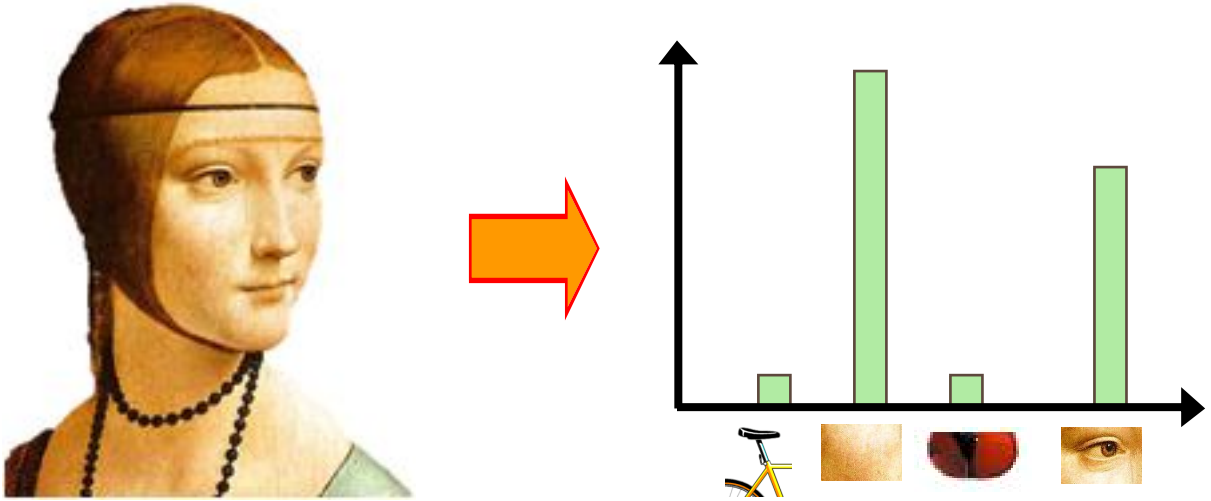
Примеры визуальных слов





Sivic et al. 2005

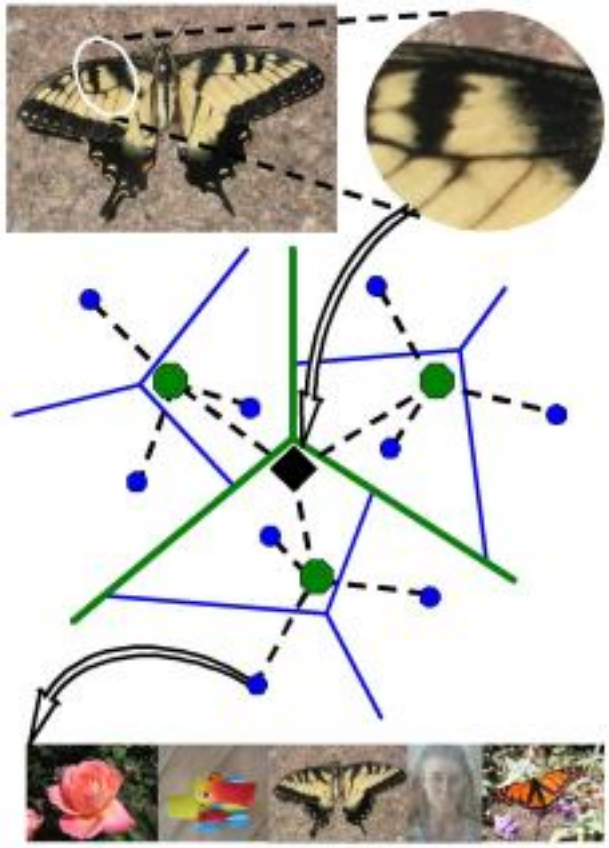
«Мешок визуальных слов»



- Теперь мы можем для каждого фрагмента изображения найти слово из словаря
  - Находим ближайшее по дескриптору слово в словаре
  - «Квантование» - сопоставляем каждому фрагменту изображения число от 1 до N, где N – размер словаря
- Вектор частот визуальных слов – «мешок слов»

Визуальные словари

- Как выбрать размер словаря?
  - Маленький: слова не могут описать все особенности
  - Большой: переобучение
- Вычислительная сложность
  - Деревья словарей (Nister & Stewenius, 2006)
  - Приближенные методы
  - Хеширование



Thank you for evaluating AnyBizSoft PDF Splitter.

A watermark is added at the end of each output PDF file.

To remove the watermark, you need to purchase the software from

http://www.anypdftools.com/buy/buy-pdf-splitter.html