

# ЛЕКЦИЯ 7. ПЕРЦЕПТИВНОЕ ХЭШИРОВАНИЕ. КАК ПОИСКОВИК ИЩЕТ ПОХОЖИЕ КАРТИНКИ

2

Что значит “похожие”?

# Схожесть цвета

3



# Схожесть формы

4



# Схожесть текстуры

5



6

# Разложение по цветам

# Цветовая модель RGB

7

- RGB – red, green, blue – цветовая модель, определяющая способ кодирования изображения с помощью трёх основных каналов.
- Цвета получаются путём добавления к чёрному.
- Смешение всех основных цветов даёт белый цвет.



# Гистограмма яркости

8

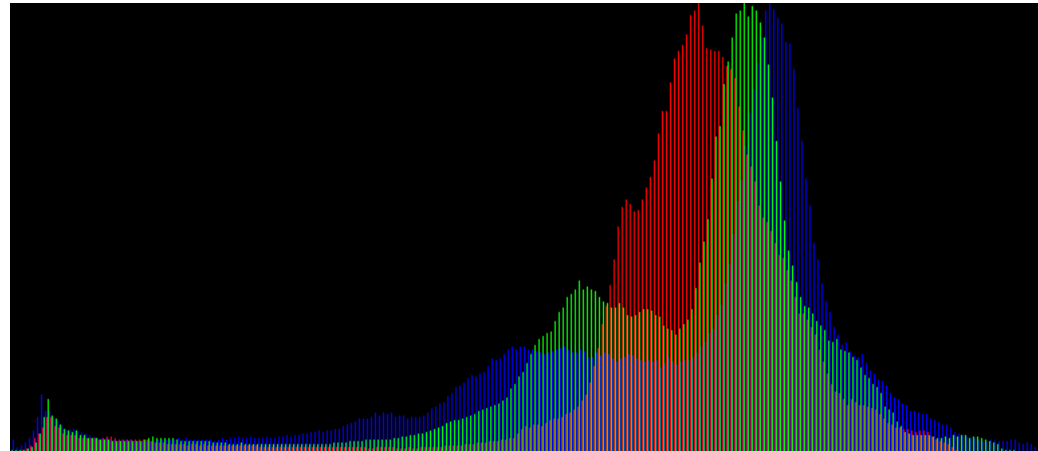
- Всего 256 значений яркости пикселя
- Для каждого значения яркости рассчитывается количество пикселей данной яркости.
- Строится столбчатая диаграмма, где по оси X отложены градации яркости, а по оси Y количество пикселей с такой яркостью.



# Цветовая гистограмма

9

- Совокупность трёх гистограмм по каждому цветовому каналу



# Сравнение гистограмм

10

- Манхеттенское расстояние:

$$d(H_1, H_2) = \sum_i |H_1(i) - H_2(i)|$$

- Корреляция:

$$d(H_1, H_2) = \frac{\sum_i (H_1(i) - \bar{H}_1)(H_2(i) - \bar{H}_2)}{\sqrt{\sum_i (H_1(i) - \bar{H}_1)^2 * (H_2(i) - \bar{H}_2)^2}}$$

- Метрика Хи-квадрат:

$$d(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i)}$$

# Текстурные признаки

# Статистические признаки

12

- Матрица Харалика:
  - ▣ Энергия
  - ▣ Момент инерции
  - ▣ Максимальная вероятность
  - ▣ Локальная однородность
  - ▣ Энтропия
  - ▣ След нормализованной матрицы пространственной смежности
  - ▣ Среднее значение яркости
  - ▣ Корреляция значений яркости изображения

# Сравнение формы

Выделение контуров

Метод зон

# Контурный оператор Прюитт

14

- Маски оператора Прюитт:

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

- Частные производные в матричном виде:

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * A$$

- Частные производные в скалярном виде:

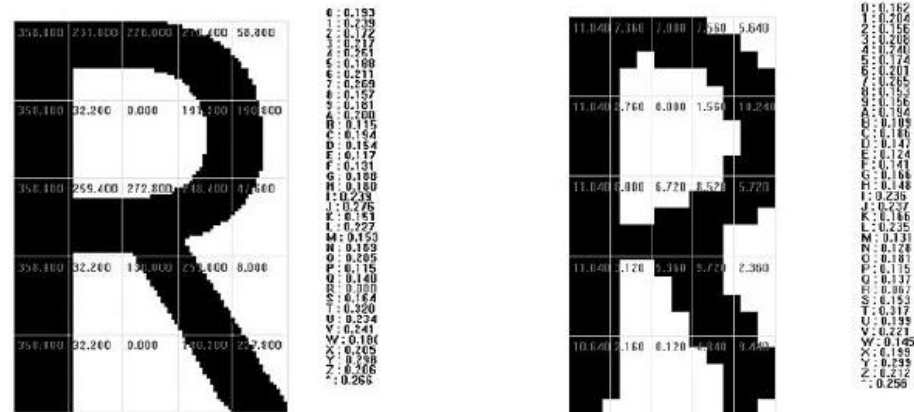
$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

# Метод зон для контурного изображения

15

- Вместо исходного изображения зонируется бинаризованная градиентная матрица
- Далее вычисляется несколько скалярных признаков для каждой зоны



16

# Перцептивное хэширование



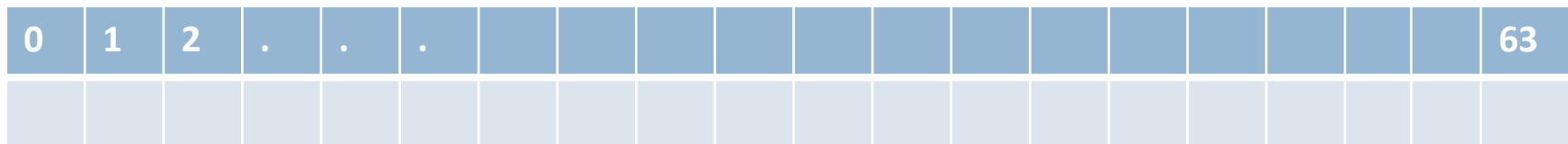
## 17



# Цветовые гистограммы

18

- Рассматриваются три слоя изображения:
  - ▣ Красный
  - ▣ Зеленый
  - ▣ Синий
- 256 оттенков каждого цветового слоя разбиваются на 4 группы по 64 оттенка: 0..63, 64..127, 128..195, 196..255.
  - ▣ Вместо  $256*256*256 = 16.7$  млн цветов получается  $4*4*4=64$  «грубых» цвета.
  - ▣ Рассчитывается число пикселей каждого «грубого» цвета.
  - ▣ Полученные значения нормируются:  $nhist = hist * 256 / S$



# Текстурные признаки

19

□ Рассчитывается матрица Харалика и 10 признаков:

- 1)  $Asm = \text{Энергия} * 65536$
- 2)  $Con = \text{Контрастность}$
- 3)  $Mpr = \text{Максимальная вероятность} * 65536$
- 4)  $Lun = \text{Локальная однородность} * 256$
- 5)  $Ent = \text{Энтропия} * 256$
- 6)  $Tr = \text{След} * 256$
- 7)  $Mi = \text{Математическое ожидание по строкам}$
- 8)  $Mj = \text{Математическое ожидание по столбцам}$
- 9)  $Stdev = \text{Среднеквадратическое отклонение}$
- 10)  $Corr = \text{Корреляция значений яркости} * 256$

64	65	66	67	68	69	70	71	72	73
asm	con	mpr	lun	ent	tr	mi	mj	stdev	cor

- Строится градиентная матрица с помощью оператора Прюитт. Приводится к чёрно-белой.
- Рассчитываются признаки:
  - ▣ Нормированный общий вес контуров
  - ▣ Координаты центра тяжести ( $X_c$ ,  $Y_c$ )
- Градиентная матрица разбивается на  $5 \times 5$  зон.
  - ▣ Рассчитывается нормированный вес каждой зоны.
  - ▣ Всего 25 чисел.

[illegible]

# Мера близости хэшей

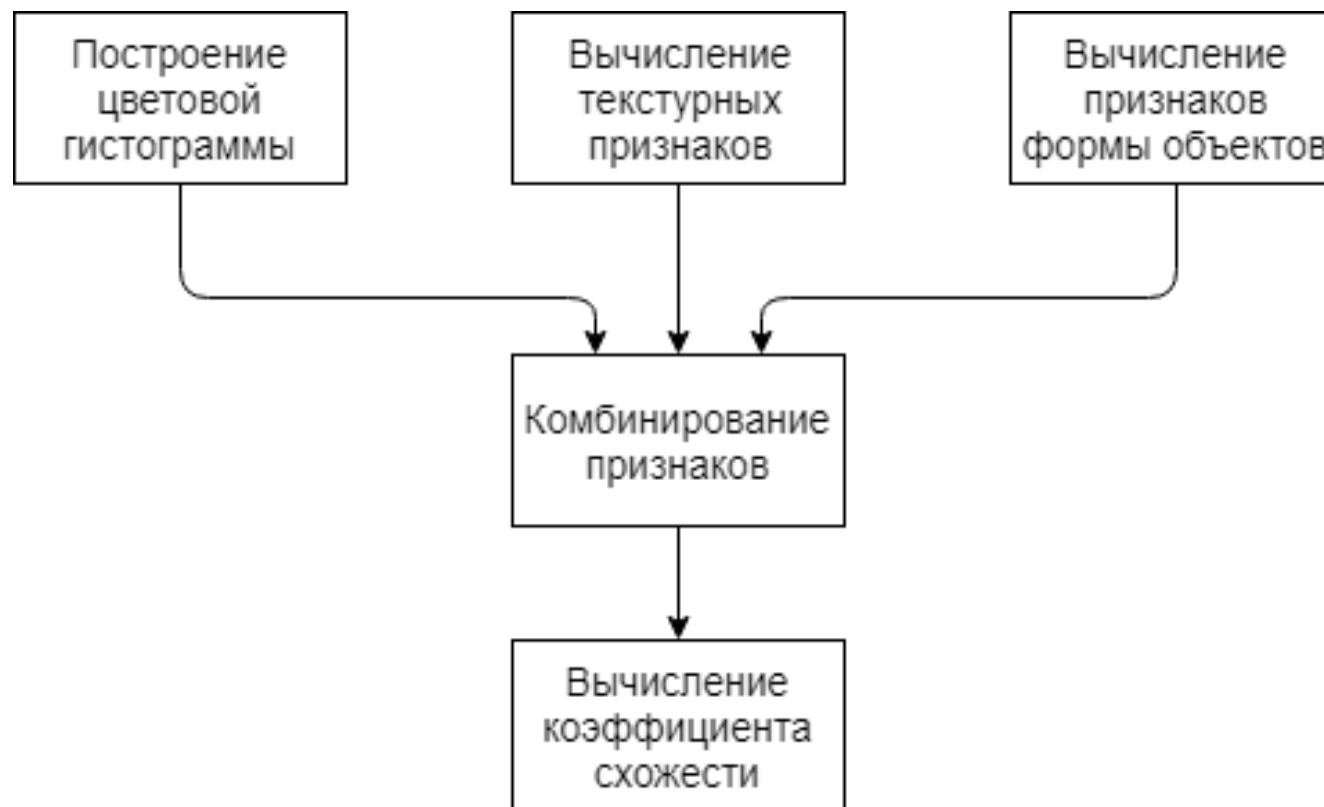
21

- Каждой группе признаков назначается весовой коэффициент.
  - ▣ Чем больше коэффициент, тем важнее считается группа признаков. Например:
    - Для гистограммы: 0.3
    - Для текстурных признаков: 0.4
    - Для формы и контуров: 0.3
- Для каждой группы вычисляется манхеттенское расстояние. Получается три числа.
- Умножаем на весовые коэффициенты и складываем:
  - ▣  $\text{dist} = k1 * \text{histogram\_dist} + k2 * \text{texture\_dist} + k3 * \text{morph\_dist}$
- Это и есть мера близости двух изображений



# Алгоритм поиска

23



# Пример 1

24

□ Искать похожее на



□ Выдача





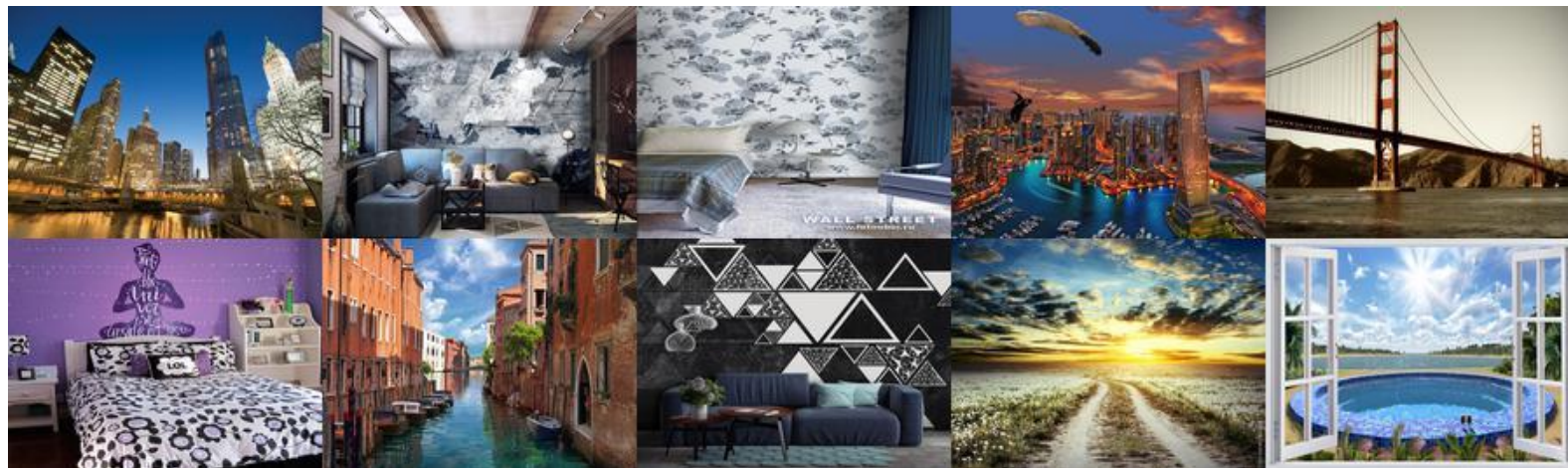
# Пример 2

25

□ Искать похожее на



□ Выдача



# Семантическая индексация

# Обучающая выборка

27

- Ключевые слова: *вода, море, лес, зима, снег, осень, космос, небо, город, огонь*
- Соответствующие им эталоны:



autumn\_forest.jpg



city.jpg



city\_sky.jpg



fire.jpg



forest.jpg



sea\_sky.jpg



snow.jpg



space.jpg



water.jpg



winter\_forest.jpg



# Индексируемое множество

28

- Хотим автоматически разметить на основе эталонов и перцептивных хэшей



01.jpg



02.jpg



03.jpg



04.jpg



05.jpg



06.jpg



07.jpg



08.jpg



09.jpg



10.jpg



11.jpg



12.jpg



13.jpg



14.jpg



15.jpg



16.jpg



17.jpg



18.jpg



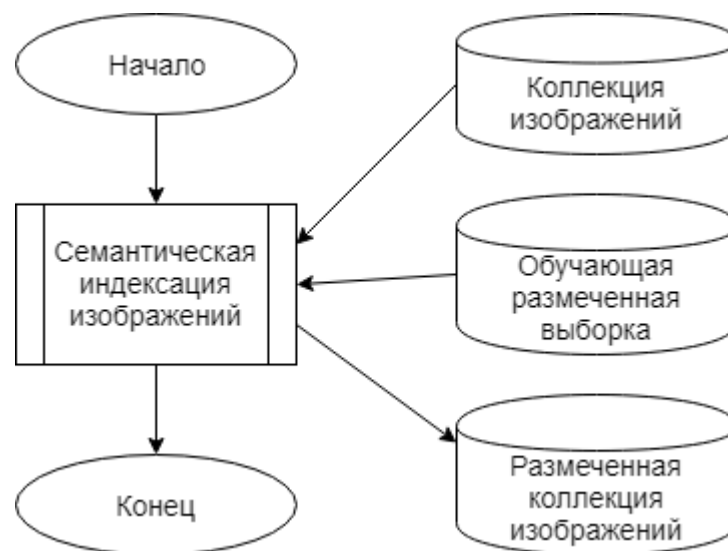
19.jpg



20.jpg

# Схема индексации изображений

29



# Результаты автоматической разметки

30



01.jpg

- 01.jpg
  - море: 0.63,
  - небо: 0.58,
  - вода: 0.39,
  - город: 0.27,
  - зима: 0.02



02.jpg

- 02.jpg
  - город: 0.85,
  - небо: 0.47,
  - море: 0.17,
  - вода: 0.09,
  - лес: 0.02



03.jpg

- 03.jpg
  - осень: 0.95
  - лес: 0.73
  - зима: 0.61
  - снег: 0.32
  - огонь: 0.13



04.jpg

- 04.jpg
  - космос: 0.45,
  - зима: 0.43,
  - лес: 0.4,
  - снег: 0.22,
  - осень: 0.12



05.jpg

- 05.jpg
  - город: 0.8,
  - небо: 0.33,
  - огонь: 0.03,
  - вода: 0.02,
  - осень: 0.02



06.jpg

- 06.jpg,
  - город: 0.66,
  - небо: 0.25,
  - космос: 0.15,
  - море: 0.02,
  - лес: 0.02



# Результаты запроса Море+вода

31



море: 95%, вода: 84%,  
небо: 59%, город: 12%,  
космос: 2%



море: 91%, вода: 83%,  
небо: 56%, город: 13%,  
огонь: 3%



море: 86%, вода: 75%,  
небо: 49%, осень: 14%,  
лес: 10%



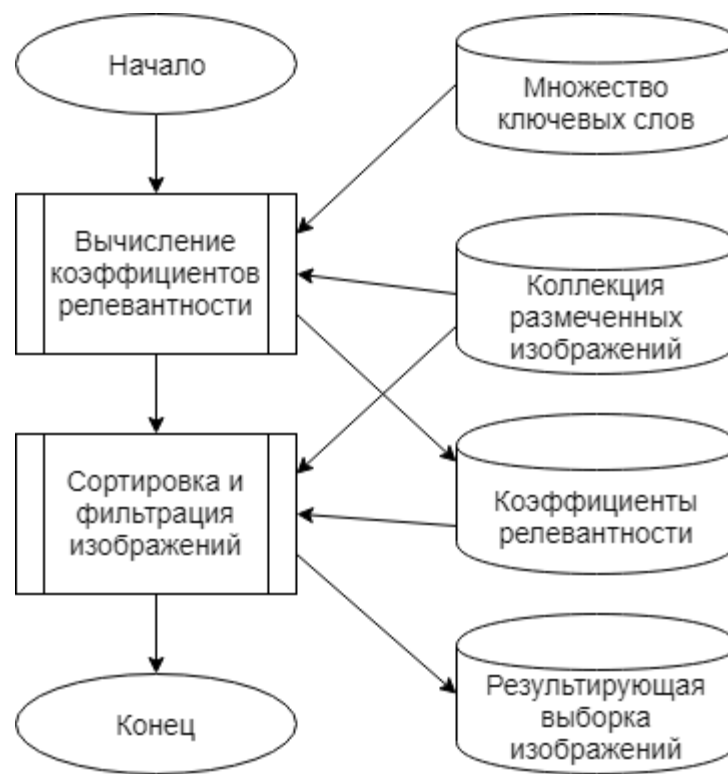
вода: 76%, море: 55%,  
небо: 49%, город: 23%,  
зима: 4%



вода: 76%, море: 53%,  
небо: 47%, город: 21%,  
зима: 2%

# Алгоритм поиска по ключевым словам

32





# Результаты запроса Город+небо

33



город: 85%, небо: 47%,  
море: 17%, вода: 9%,  
лес: 2%



город: 80%, небо: 33%,  
огонь: 3%, вода: 2%,  
осень: 2%



город: 66%, небо: 25%,  
космос: 15%, море:  
2%, лес: 2%



море: 63%, небо: 57%,  
вода: 39%, город: 27%,  
зима: 2%



вода: 76%, море: 55%,  
небо: 49%, город: 23%,  
зима: 4%

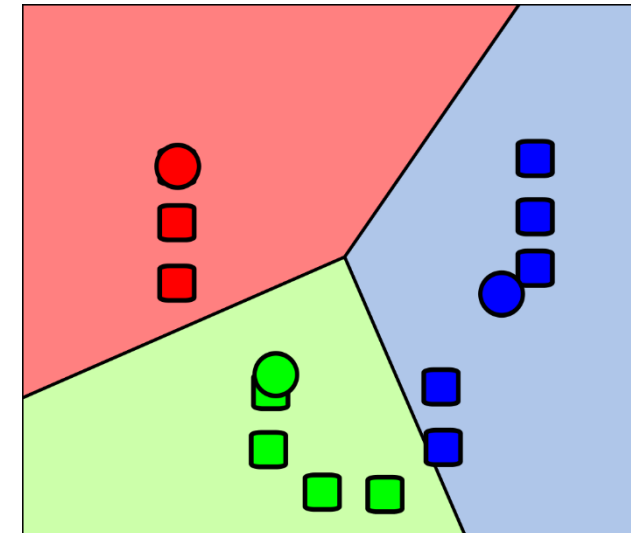
# Кластеризация

Задача автоматически подобрать наборы схожих изображений для облегчения разметки эталонов

# Кластеризация

35

- **Кластеризация** (Clustering) — является логическим продолжением идеи классификации, но сложнее.
- Особенность кластеризации заключается в том, что классы объектов изначально не predetermined.
- Результатом кластеризации является разбиение объектов на группы.



# Алгоритм обучения

36

- Множество изображений разбивается на кластеры
  - ▣ Например, методом k-средних.
- Кластеры с семантически схожими изображениями именуются человеком вручную и становятся классами.
- После разметки обучается классификатор на основе признаков.
  - ▣ Например, алгоритмом C5.0
- Далее по именам классов можно искать схожие изображения.

# Примеры кластеров

37

□ Еда:



□ Вода





# Примеры кластеров

38

## □ Белый



## □ Люди



# Пример работы классификатора

39

- Пример дерева решений классификатора C5.0 на основе перцептивного хэша:
- Правее результаты классификации 15 изображений

The screenshot shows a window titled "Cross-reference for img decision tree". It displays a decision tree on the left and a list of 15 classification results on the right.

**Decision Tree:**

```
graph TD
    Root["h1 > 0.3956:"]
    Root --> L1["h1 <= 0.7769: sky (8)"]
    Root --> R1["h1 > 0.7769: white (4)"]
    L1 --> L2["h1 <= 0.3956:"]
    L2 --> L3["h2 > 0.1689: food (7)"]
    L2 --> R2["h2 <= 0.1689:"]
    R2 --> L4["h0 <= -0.1873:"]
    R2 --> R3["h0 > -0.1873:"]
    L4 --> L5["h15 <= 0.1181: water (7)"]
    L4 --> R4["h15 > 0.1181: grass (2)"]
    R3 --> L6["h0 <= 0.1772: field (6)"]
    R3 --> R5["h0 > 0.1772: people (8)"]
```

**Classification Results:**

Index	Class
0	sky
1	sky
2	water
3	white
4	people
5	white
6	water
7	water
8	water
9	food
10	sky
11	water
12	water
13	field
14	field

Buttons: Save, Reset

# Что почитать

40

- Шитова О.В., Пухляк А.Н., Дроб Е.Н. Анализ методов сегментации текстурных областей изображений в системах обработки изображений // Научные ведомости, 2014 № 8 (179). Выпуск 30/1, с. 182–188
- Фраленко В. П. Методы текстурного анализа изображений, обработка данных дистанционного зондирования // Программные системы: теория и приложения № 4(22), 2014, с. 19–39
- Перевалов Д. Анализ текстур <https://ru.scribd.com/doc/115002510/opencv-%D0%90-%D0%BD-%D0%B0-%D0%BB-%D0%B8-%D0%B7-%D1%82-%D0%B5-%D0%BA-%D1%81-%D1%82-%D1%83-%D1%80>
- <https://cyberleninka.ru/article/n/raspoznavanie-izobrazheniy-na-osnove-teksturnyh-priznakov-haralika-i-iskusstvennyh-neyronnyh-setey>
- <https://www.nature.com/articles/s41598-017-08764-7/tables/5>
- Метод k-средних  
[https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4\\_k-%D1%81%D1%80%D0%B5%D0%B4%D0%BD%D0%B8%D1%85](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_k-%D1%81%D1%80%D0%B5%D0%B4%D0%BD%D0%B8%D1%85)