МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

ИАСТ

Кафедра автоматизированных систем и технологий

09.03.02

Направление подготовки/Специальность Информационные системы и технологии

Дисциплина Технологии компьютерного зрения

# Лабораторная №8.

# Метод мешка слов.

Выполнили студенты Копосов Лев Владимирович

Копосов Владимир Владимирович

Группа 22-ИСбо-1б

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Кострома

Вопросы:

1. **Что такое визуальное слово?**

Визуальное слово - это некий часто повторяющийся фрагмент в изображении или группе изображений. Порядок этих изображений не важен. Визуальное слово должно быть достаточно простым, чтобы встречаться в разных изображениях, и достаточно сложным, чтобы быть узнаваемым.

1. **В методе мешка визуальных слов, чему соответствуют кластеры дескрипторов?**

В методе мешка визуальных слов кластеры дескрипторов соответствуют “визуальному словарю”. Для извлечения фрагментов выделяют локальные особенности или делят изображение по регулярной сетке. Для выбранных особенностей стоится описание для последующего сравнения.

1. **Из каких элементов состоит вектор-описание классифицируемого изображения?**

Вектор-описание классифицируемого изображения состоит из гистограммы частот распределения визуальных слов. Гистограмма показывает, какие визуальные слова присутствуют на изображениях и сколько их.

1. **Какой из указанных параметров оказывает наибольшее влияние на успех кластеризации методом k-средних и почему?**
   1. **Количество точек в кластере**
   2. **Степень "разбросанности" точек в кластере**
   3. **Степень "вытянутости" кластера**

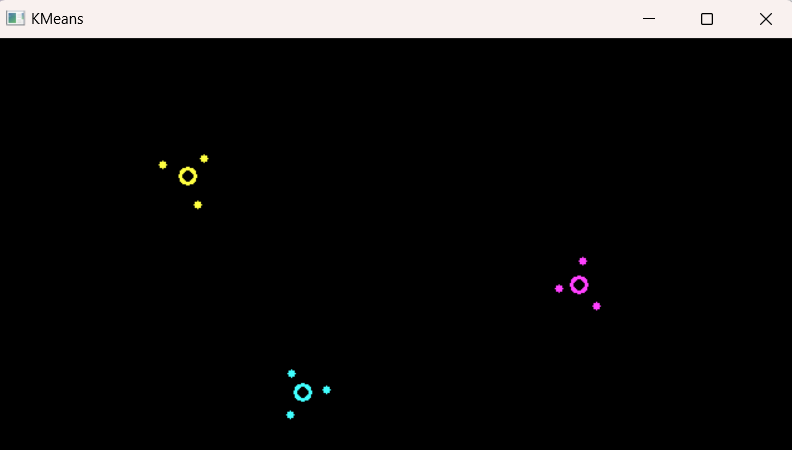
На успех кластеризации методом k-средних оказывает наибольшее влияние параметр степени "разбросанности" точек в кластере. Кластеризация методом k-средних стремится минимизировать сумму квадратов расстояний от каждой точки до центроида этого кластера. Если точки в кластере сильно разбросаны, то алгоритм с большой вероятностью может ошибочно объединить точки в один кластер или разделить их на несколько.

Вывод в консоль.

##### **Задание 1**

Создайте программу, выполняющую следующие действия.

* Программа должна вывести на экран изображение размером 640х480 и регистрировать щелчки мышью по нему, не разделяя их на классы.
* Нажатие на 2-9 должно производить кластеризацию введённых точек алгоритмом k-средних на соответствующее число классов, при условии что отмечено достаточно точек.
* После кластеризации программа должна отметить центры классов, и вывести каждую из записанных точек цветом ближайшего центра. После добавления новой точки эта разметка может быть удалена.

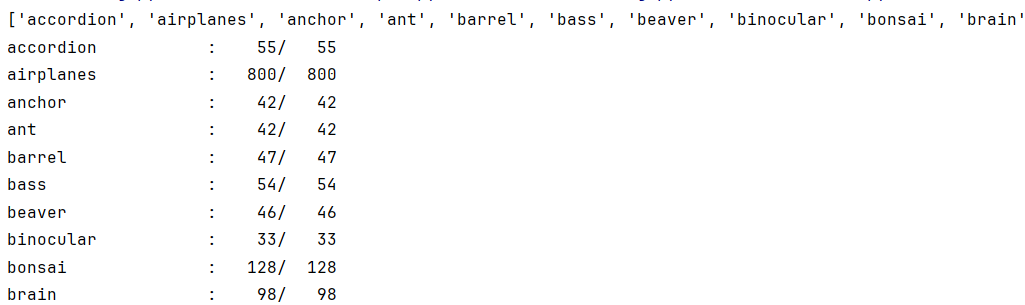


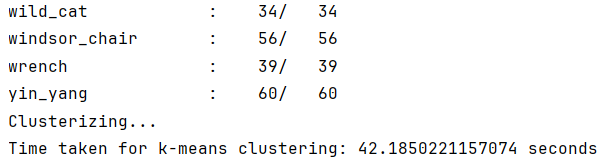
##### **Задание 2**

Создайте программу, формирующую классификатор изображений методом bag of words.

Алгоритм построения классификатора:

1. Сформируйте список классов и изображений в каждом классе.
2. Загрузите каждое изображение из базы и нормализуйте его (оттенки серого, размер 128x128).
3. Найдите на изображении особые точки дескриптором SIFT.
4. Поместите дескрипторы всех точек в один список. При этом для каждого изображения запомните соответствующий диапазон индексов.
5. Кластеризуйте полученный массив индексов методом k-means, разделив его на 128 кластеров - визуальных слов. Замерьте и выведите на экран время, которое это заняло.
6. Для каждого изображения определите метки кластеров, соответствующих дескрипторам этого изображения. Используя стандартный класс Python **collections.Counter**, постройте для изображения гистограмму визуальных слов.
7. Сохраните, используя формат JSON (стандартный модуль **json**), следующие данные:
   1. Количество использованных кластеров-слов.
   2. Позиции центров (SIFT-дескрипторы) использованных кластеров-слов.
   3. Список имён классов изображений.
   4. Номер класса, имя и значения гистограммы слов для каждого изображения.



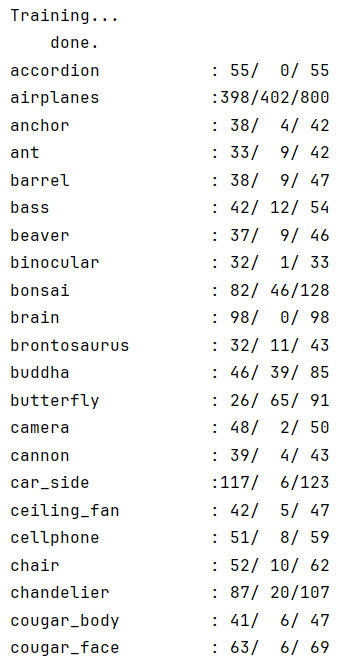


##### **Задание 3**

Создайте программу, загружающую созданный в задании 5 классификатор изображений и оценивающую качество его работы.

Алгоритм работы программы:

1. Сформируйте список классов и изображений в каждом классе.
2. Загрузите данные из сформированного JSON файла. Убедитесь, что позиции центров кластеров-слов, а также элементы гистограмм слов имеют тип данных **numpy.float32**. Убедитесь, что тип данных массива меток (номеров классов) **numpy.int32**.
3. Обучите линейный классификатор типа **SVM\_C\_SVC** с ядром **SVM\_LINEAR**, используя массив гистограмм и меток.
4. Постройте список классов и изображений в базе для проверки (используйте ту же базу, что и при обучении).
5. Для каждого изображения произведите нормализацию и построение набора особенностей как в задании 5. Используя **cv2.BFMatcher** с метрикой **cv2.NORM\_L2**, найдите наиболее похожее кластер-слово для каждой особенности. Используя класс **Counter**, постройте гистограмму для изображения.
6. Классифицируйте полученную гистограмму для изображения и определите номер класса, к которому его отнёс классификатор.
7. Посчитайте отдельно для каждого класса количество успешных и ошибочных классификаций.
8. В конец программа должна вывести в текстовый файл список имён классов и степеней успешности их определения, в порядке убывания успешности.



##### **Дополнительное задание**

Поскольку процесс обучения и проверки займёт несколько минут, выводите в процессе работы имя текущего класса и количество обработанных изображений. При проверке выводите количество корректно и некорректно распознанных изображений класса, а также общее количество изображений в классе. По окончании работы выведите среднюю успешность классификации.

