1 K-Means

Алгоритм K-Means - алгоритм машинного обучения, который используется для группировки объектов на основе их признаков и является одним из наиболее популярных методов кластеризации.

Основная идея метода заключается в следующем:

1. Задается количество кластеров K.
2. Инициализируются K случайных центров каждого кластера.
3. Пока не будет выполнено условие остановки:
   (a) Для каждого объекта данных x:
      i. Находится ближайший центр кластера с помощью заданной функции расстояния.
      ii. Объект x присваивается к ближайшему кластеру.
   (b) Для каждого кластера k:
      i. Вычисляется новый центр кластера, как среднее значение всех объектов, отнесенных к данному кластеру.
4. Возвращаются кластеры.

Полученные кластеры описываются векторами средних значений - центроидами. В процессе разбиения выполняется итеративная минимизация внутриклассовых расстояний J. Соответствующая целевая функция выглядит следующим образом:

$$J = \sum_{j=1}^{K} \sum_{x^{(i)} \in C^{(j)}} ||x^{(i)} - c^{(j)}||^2 \to min, \tag{1}$$

где $x^{(i)}$ – вектор характеристик объекта, K - количество кластеров, $c^{(j)}$ – центроид кластера $C^{(j)}$. Функция расстояния обычно выбирается в зависимости от пространства, в котором расположены объекты. В качестве используемой метрики рассмотрим Евклидово расстояние:

$$\rho(x_i; y_i) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{2}$$

Оно представляет собой расстояние между точками в n-мерном пространстве.

Данный алгоритм может быть применен для сегментации изображений - разбиения исходного изображения на группы пикселей, сегменты, каждый из которых содержит объекты одного типа или имеет одинаковые характеристики. Рассмотрим работу представленного метода на примере растрового изображения. В качестве объектов x будут выступать пиксели изображения, а в качестве характеристик - их цвета в трехмерном пространстве RGB.

1. Мы преобразуем каждое изображение в матрицу признаков. В нашем случае для каждого изображения получаем матрицу размерности $(N \times M \times 3)$, где $N \times M$ - размер изображения, а каждый элемент матрицы содержит тройку значений R, G и B для соответствующего пикселя.

2. Затем мы применяем алгоритм K-Means к полученным данным, указав K кластеров и выбрав их центры случайным образом.

3. После достижения сходимости мы можем использовать полученные центры кластеров для создания сегментированного изображения, где каждый пиксель заменяется на центр соответствующего кластера, что и дает нам K сегментов изображения.

Преимуществом использования алгоритма K-Means для сегментации изображений является его простота и эффективность в работе с большими объемами данных. Однако, как и любой алгоритм кластеризации, K-Means не всегда может давать оптимальный результат, особенно если изображение имеет сложную структуру или содержит неоднородные объекты. На практике может быть использовано несколько модификаций и улучшений, таких как использование случайной инициализации центров кластеров несколько раз для более надежных результатов или применение критериев остановки на основе изменения инерции кластеров.

2 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) - это алгоритм кластеризации, который позволяет искать группы данных на основе плотности их расположения в пространстве. Этот алгоритм может быть использован для кластеризации данных различного типа, включая точки, тексты, звуки, изображения и т.д.

Основной идеей DBSCAN является выделение кластеров путем нахождения областей плотной группировки точек. В этом алгоритме вводится понятие радиуса ϵ и минимального числа точек $minSamples$, которые должны находиться в этом радиусе. Точки, которые находятся на расстоянии меньшем, чем ϵ, объединяются в кластер, если количество точек в этом кластере превышает заданное число $minSamples$. Если же точки не входят ни в один кластер, они считаются выбросами (или шумом).

Основная идея метода заключается в следующем:

1. Задаются параметры: радиус ϵ и минимальное количество точек $minSamples$.

2. Случайным образом выбирается точка, которая еще не была отнесена ни к одному кластеру.

3. Ищутся все точки, которые находятся в пределах радиуса эпсилон от выбранной точки.

   (a) Если количество точек, найденных в радиусе, меньше минимального количества, то эта точка помечается как выброс (или шум).
   (b) Иначе, создать новый кластер и добавить все точки, находящиеся в радиусе, к этому кластеру.
4. Повторять шаги 2-3 для всех точек, которые еще не были отнесены ни к одному кластеру.
5. После завершения поиска кластеров, вернуть результаты, включая множество кластеров и выбросов.

Данный алгоритм также может быть использован для сегментации изображений, позволяя выделять группы пикселей, которые имеют сходные характеристики. Процесс сегментации с помощью DBSCAN алгоритма может быть выполнен следующим образом:

1. Преобразовать изображение в массив точек. В нашем случае для каждого изображения получаем матрицу размерности $(N * M \times 3)$, где $N * M$ - произведение длины и ширины изображения, а каждый элемент матрицы содержит тройку значений R, G и B для соответствующего пикселя.
2. Определить параметры DBSCAN: радиус ϵ и минимальное количество точек $minSamples$ в выбранном радиусе.
3. Применить алгоритм к массиву точек изображения, чтобы разбить их на кластеры на основе плотности.
4. Для каждого кластера, заменить все точки, принадлежащие этому кластеру, на один цвет.
5. Вывести полученное изображение с новыми цветами для каждого кластера.

После описанного выше процесса сегментации пиксели помещаются в разные кластеры и формируют разные сегментированные области изображения.

Сегментация изображения с помощью DBSCAN может быть особенно полезна для выделения объектов на фотографиях, в том числе для определения границ объекта, удаления фона на многобайтовых изображениях, таких как медицинские изображения. Однако важно учитывать, что представленный метод не всегда может справиться со сложными формами объектов, и для этого могут быть необходимы другие методы сегментации.

3 OPTICS

Ordering Points To Identify the Clustering Structure (OPTICS) - это алгоритм кластеризации, который позволяет определить структуру кластеров и выбрать оптимальное их количество. OPTICS является расширением алгоритма DBSCAN. Оба алгоритма основаны на понятии плотности, а именно, на

плотности точек в заданном пространстве. Однако, в отличие от DBSCAN, OPTICS учитывает не только плотность, но и взаимное расположение объектов в пространстве, что позволяет строить более гибкие и устойчивые к шуму кластеры.

Основная идея алгоритма заключается в том, чтобы рассматривать объекты в порядке возрастания их расстояний от других объектов, начиная с того, что находится ближе всего к центру кластера, и строить граф на основе этих расстояний. Затем алгоритм извлекает из графа кластеры, используя минимальное расстояние до соседних объектов.Основным преимуществом алгоритма OPTICS является то, что он не требует предварительной настройки параметров, таких как число кластеров или радиусы кластеров.

Шаги алгоритма OPTICS:

1. Задается начальная точка, обозначаемая как p.
2. Вычисляются расстояния от точки p до всех других точек в наборе данных.
3. Определяется значение *core-distance* для точки p - это расстояние до k-го ближайшего соседа, где k - это минимальное число точек, которое должно быть в радиусе ϵ для того, чтобы точка считалась "основной". Если количество соседей в радиусе ϵ меньше, чем k, то точка считается "шумовой"и не рассматривается в дальнейшем.
4. Если точка p является основной, то определяется ее набор соседей, т.е. точки, которые находятся в радиусе ϵ от нее.
5. Для каждой точки из набора соседей вычисляется *reachability-distance* - это максимальное расстояние до начальной точки через любую из уже посещенных точек. Точки сортируются по возрастанию *reachability-distance*.
6. Сформированный список точек является кластером, причем порядок точек внутри кластера определяется их *reachability-distance* . Если точка не имеет достаточного числа соседей, то она считается выбросом и не включается в кластер.
7. После обработки текущего кластера, процесс повторяется для следующей нерассмотренной точки, пока все точки не будут просмотрены.

В результате выполнения алгоритма OPTICS получается список точек, отсортированных по возрастанию *reachability-distance*. Для построения кластеров необходимо использовать метод DBSCAN, который определяет кластеры на основе параметров *core-distance* и ϵ.

OPTICS также как и DBSCAN использовать параметры радиуса и $minSamples$, но при этом не строит кластеры напрямую. Вместо этого он строит упорядоченный список точек, отсортированных по взаимному расстоянию их соседей. Каждой точке в списке присваивается значение *reachability-distance*, которое определяет минимальное расстояние до соседней точки, необходимое

для достижения области плотности. *Reachability-distance* может использоваться для нахождения кластеров разной плотности и формы, а также для определения выбросов.

Таким образом, OPTICS расширяет возможности DBSCAN, добавляя гибкость и устойчивость к шуму, но при этом требует больше вычислительных ресурсов для построения упорядоченного списка точек.

4 Постановка задачи

Дано произвольное растровое изображение человеческого глаза в формате *.png*. Необходимо исследовать применимость методов кластерного анализа для решения задачи сегментации глаза. В качестве основных алгоритмов были выбраны K-Means, DBSCAN и OPTICS.Объектами исследования являются пиксели изображения, а вектором характеристик - цвета пикселя в трехмерном пространстве RGB. В качестве метрики используется евклидово расстояние (2).

5 Результаты

В качестве исходных данных был взять датасет с набором фотографий человческих глаз в инфракрасном излучении, а также масок зрачков для получения метрик качества (рис. 1).

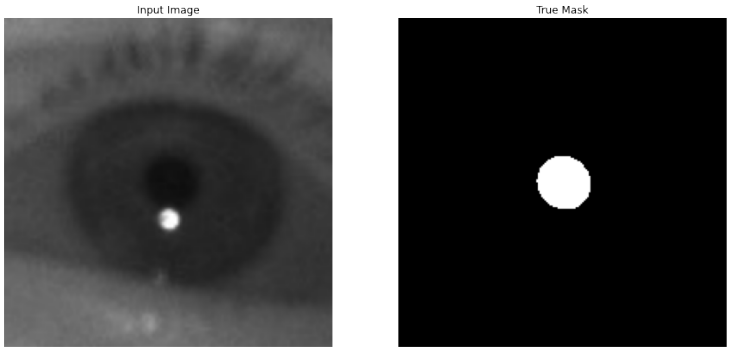

Рис. 1: Пример входного изображения глаза и соответствующей маски зрачка

1. Дано растровое изображение в формате RGB, размером 200x200 пикселей и соответствующая ему маска зрачка (рис. 1).
2. Накладываем фильтр для сглаживание изображения Input Image по краям с использованием библиотеки OpenCV. Применяем метод GaussianBlur, что позволит нам избавиться от острых краев, броских деталей и различных шумов, сводя к минимуму чрезмерное размытие (рис. 2).

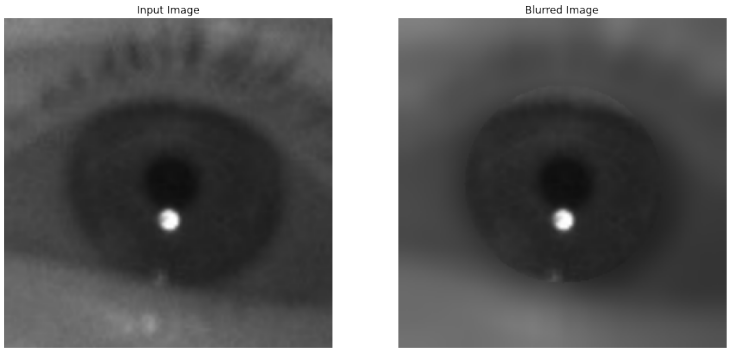

Рис. 2: Размытие краев изображения с применением фильтра GaussianBlur

3. К предобработанному изображению Blurred Image применяем исследуемые алгоритмы: K-Means, DBSCAN и OPTICS (рис. 3).

4. Выделяем только те пиксели, которые были отнесены к кластеру зрачка, тем самым формируя предсказанную маску Predicted Mask (рис. 4 - рис. 6).

5. В качестве метрики качества рассмотрим Intersection over Union (IoU) - это число, которое количественно определяет степень перекрытия между двумя блоками. В случае сегментации объектов IoU оценивает перекрытие областей True Mask и Predicted Mask - отношение площади пересечения к объединенной площади предсказанного объекта и истинного.
$$IoU = \frac{S_{Overlap}}{S_{Union}}, \tag{3}$$
где $S_{Overlap}$ - площадь пересечения True Mask и Predicted Mask; S_{Union} - площадь объединения True Mask и Predicted Mask.

6. Применим описанную процедуру для датасета, содержащего 100 изображений, и рассмотрим средние значения метрик IoU для каждого алгоритма.

Таблица 1: Средние показатели метрики качества IoU для алгоритмов K-Means, DBSCAN и OPTICS.

Название метода	Среднее значение IoU для изображений без предпроцессинга	Среднее значение IoU для изображений с предпроцессингом
K-Means	0.726	0.750
DBSCAN	0.464	0.735
OPTICS	– – –	0.623

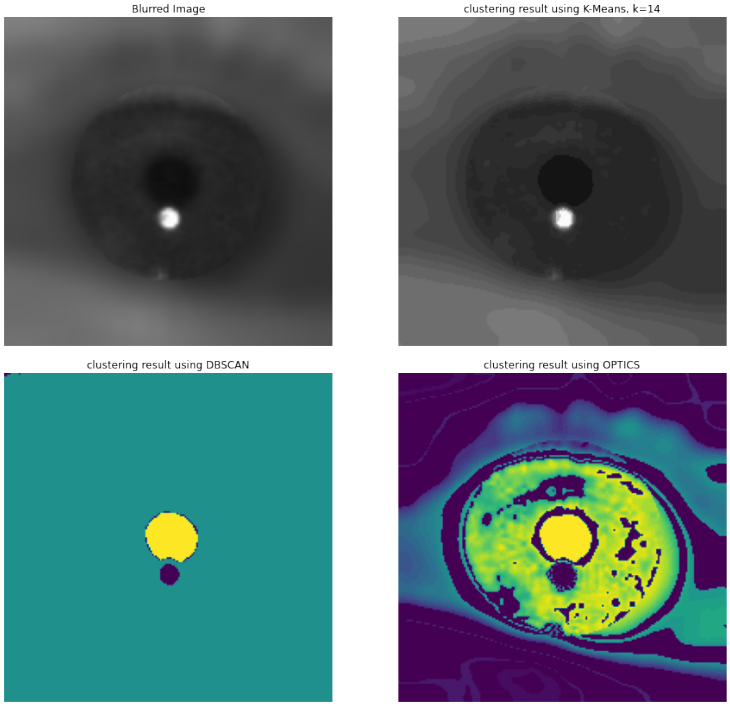

Рис. 3: Результаты кластеризации алгоритмов K-Means, DBSCAN и OPTICS

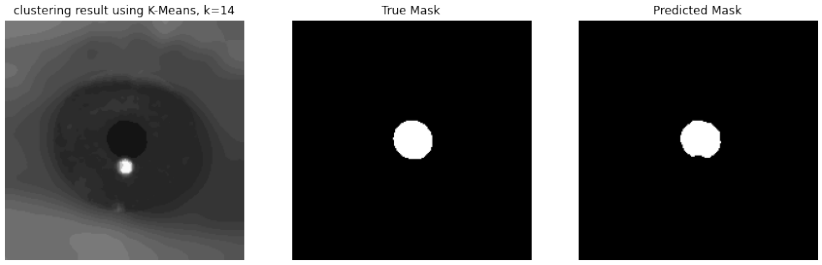

Рис. 4: Предсказанная маска зрачка, полученная в результате работы K-Means, IoU = 0.892

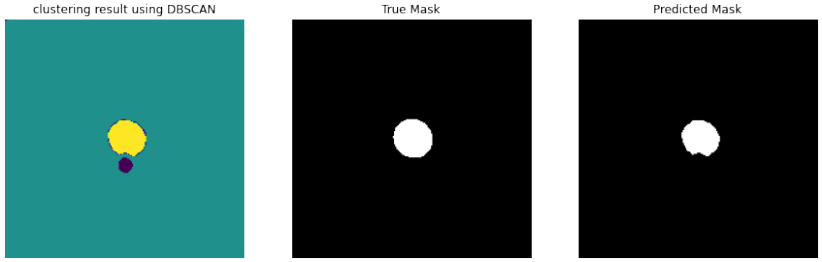

Рис. 5: Предсказанная маска зрачка, полученная в результате работы DBSCAN, IoU = 0.854

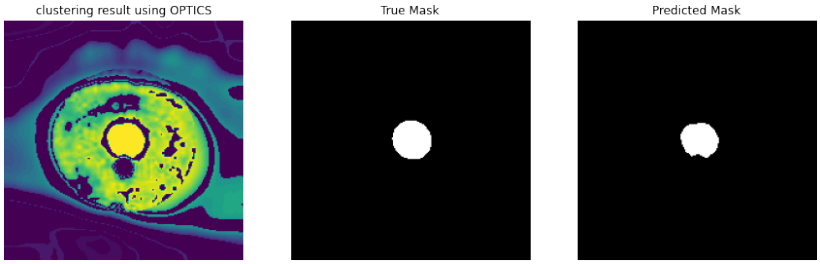

Рис. 6: Предсказанная маска зрачка, полученная в результате работы OPTICS, IoU = 0.835