МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ярославский государственный университет имени П. Г. Демидова»

Кафедра компьютерных сетей

Сдано на кафедру

«_____» _____2021г.

Заведующий кафедры,
профессор, доктор физ.-мат. наук.

_____С. Д. Глызин

Решение задачи «Сегментация изображений» методом кластеризации в пространстве признаков

01.03.21 Прикладная математика и информатика

Научный руководитель
доцент, к. физ.-мат. наук.

_____М.В. Краснов

«_____» _____2021г.

Студент группы ИВТ-42 БО

_____А.В. Ющенко

«_____» _____2021г.

Ярославль, 2021

Реферат

Объем 24 стр., 3 гл., 35 рисунков, 5 источников. **Сегментация изображений методом кластеризации в пространстве признаков**.

Объектом исследований является задача сегментации изображений при помощи методов кластеризации и генетического алгоритма для решения промежуточной задачи поиска коэффициентов.

Цель работы – разработать алгоритм сегментации изображений с использованием методов кластеризации.

В результате работы был создан и реализован модифицированный алгоритм сегментации изображений проведен анализ полученных результатов и сделан вывод о влиянии различных параметров на работу алгоритма.

Содержание

Постановка задачи

Разработать параметризованный алгоритм для сегментации полноцветных изображений на основе методов кластеризации в пространстве признаков и генетического алгоритма.

Глава 1. Сегментация полноцветных изображений

Введение

Сегментация изображения — это разбиение изображения на множество областей, схожих по неким признакам, например, цвету или текстуре. Результатом является набор сегментов изображения, то есть отображение множества точек изображения в конечный и небольшой набор значений 1. . .n, где n есть число сегментов. Такое разбиение может служить как предварительным шагом для последующего морфологического или семантического анализа и интерпретации наблюдаемой сцены, так и конечным результатом, например, в задачах обнаружения объектов. Сегментация изображений используется во многих областях, например, для обработки медицинских снимков, для создания карт местности по снимкам спутников, а также в производстве для нахождения дефектов деталей.

Пример сегментации изображения

Существуют, как минимум, два разных подхода к решению задачи сегментации:

1) путем выделения границ областей
2) путем наращивания точек области

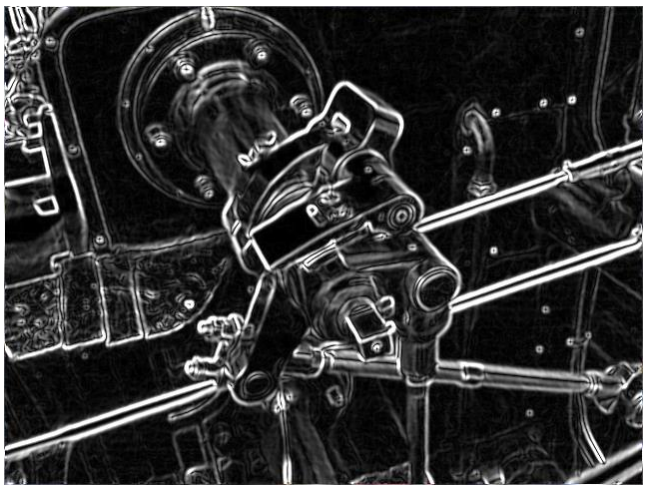
Первый подход основан на идее “разрывности” свойств точек изображения при переходе от одной области к другой. При этом выделение границ областей позволяет идентифицировать и сами области.

Второй подход реализует стремление выделить точки изображения, однородные по своим локальным свойствам, и объединить их в область, которой позже будет присвоено имя или смысловая метка. Этот подход можно подразделить на следующие

1) морфологический подход
2) разбиение по признаку однородности
3) кластеризация в пространстве признаков

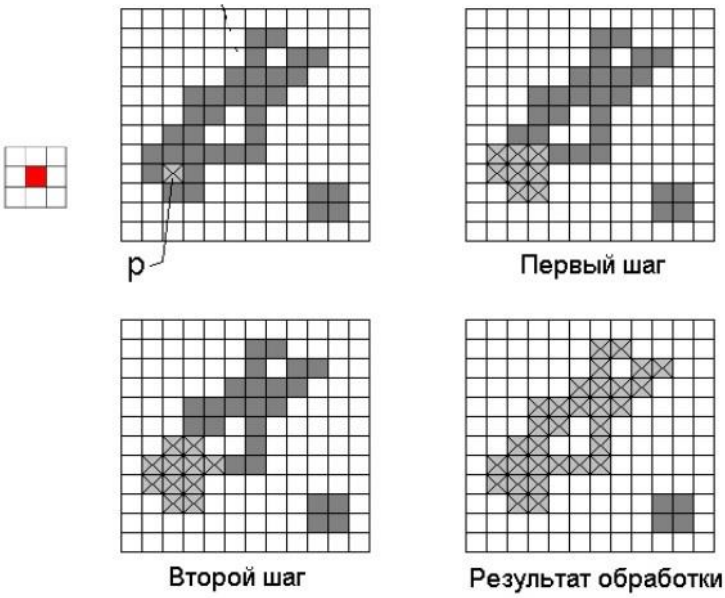
Краткая характеристика методов:

1. Разбиение изображения проведением границ. Подход заключается в обнаружении границ контурными операторами, в их прослеживании, связывании и составлении из них замкнутых границ областей. Большинство алгоритмов обнаружения контурных границ основано на фильтрации. Наиболее известные используют вычисление первой производной (операторы Робертса, Собела) или второй производной (оператор Лапласа). Далее применяются алгоритмы отслеживания контуров такие как алгоритм жука, или Алгоритм сканирования всего изображения секущей полосой. Существуют также и другие методы, и алгоритмы. Сегментация на основе проведения границ используется редко, в частности для некоторых специфических классов анализируемых изображений.



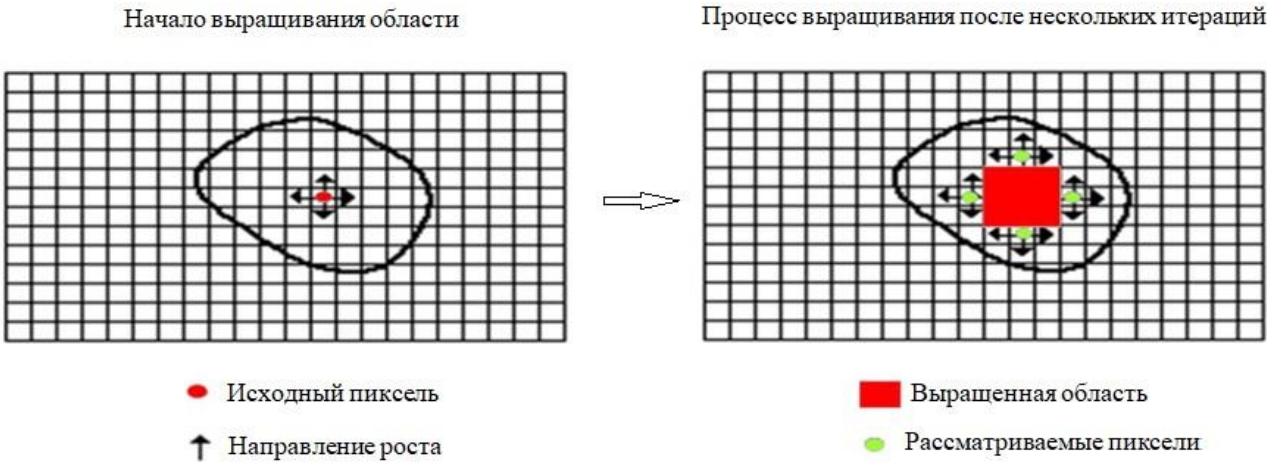
Результат выделения краев и пример работы алгоритма отслеживания контуров

2. Морфологический подход. Основными операциями в этом методе являются дилатация и эрозия. Подход достаточно эффективен для монохромных изображений, но слабо применим к цветным и текстурным.



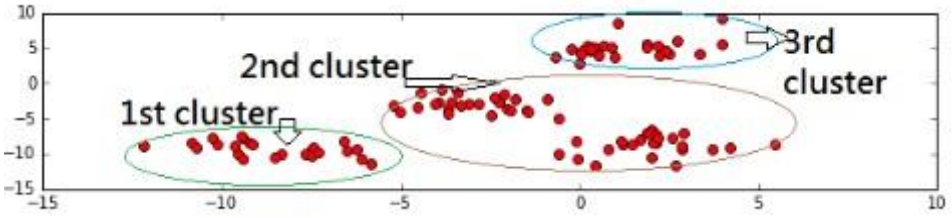
Пример использования морфологической дилатации для сегментации изображения

3. Разбиение изображения на однородные по значению области. Здесь в первую очередь необходимо упомянуть пороговые методы, использующие глобальные и адаптивные пороги. Такие методы просты в реализации, но подобрать удовлетворительные пороги не всегда возможно. Известны методы выращивания областей, заключающиеся в группировке элементов и мелких областей изображения в более крупные, начиная из так называемых «центров кристаллизации». Проблемы данного метода — выбор подходящих мер близости элементов и критериев остановки процесса выращивания областей. Альтернативой выращиванию служат методы, основанные на первичном разбиении изображения на множество малых областей и последующем их слиянии или разделении.



Еще один подобный метод называется - сегментация по водоразделам, который состоит в интерпретации гладких областей как локальных бассейнов, а контуров между ними как водоразделов.

4. Кластеризация в пространстве признаков. Состоит в выборе отображения набора входных данных в некоторое пространство признаков и разбиении выбранного пространства на кластеры в зависимости от плотности распределения. Существуют методы, опирающиеся на то что количество классов объектов в изображении заранее известно, у них есть преимущество – они очень быстро работают в сравнении с методами где изначально количество кластеров не известно.



Задача сегментации полноцветных изображений

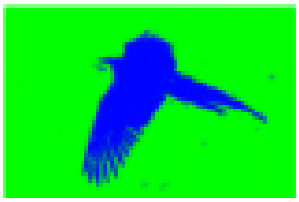
На вход алгоритм получает цветное изображение размера n x m которое можно представить в виде набора точек, имеющих пространственные и цветовые координаты, далее их надо объединить в связные области, отличающиеся от соседствующих областей цветовыми характеристиками. Так же следует учесть некоторые особенности сегментации именно изображений:

1) Области имеют не предсказуемую, за частую сложную форму
2) Число объектов обычно не известно
3) Области могут быть достаточно больших размеров, и цветовые характеристики на их протяжении могут изменяться.
4) На разных изображениях размеры потенциальных сегментов могут изменяться

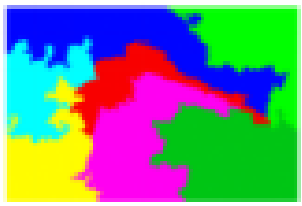
Таким образом алгоритм должен опираться не только на близость выбранных признаков, но и на пространственное расположение элементов на изображении. Ключевым здесь является выбор оценки близости соседних элементов, что в применении к рассматриваемой задаче означает выбор пространства признаков и способа оценки близости точек в этом пространстве.

В ходе работы были изучены выше перечисленные методы сегментации, лучшими для неопределенных задач оказались методы кластеризации в пространстве признаков. При написании программы были реализованы два алгоритма k-Means и Mean Shift. Из них, последний оказался более результативным и удобным в модификации, однако, так же и более трудоемким. Главным недостатком k-Means является то, что количество кластеров должно быть указанно явно.

а) оригинальное изображение



б) MeanShift



в) K-Means

Метод кластеризации Mean Shift

Данный алгоритм включает в себя два этапа – операции сдвига и непосредственно этап создания кластеров. Результирующее число кластеров определятся на последнем этапе по исходным данным, что хорошо подходит для решения задачи общего вида.

На первом этапе Mean Shift группирует элементы со схожими признаками путем сдвига их в направлении среднего значения в их области. Область задается радиусом, а элементы попадают в итоговую сумму если они находятся на расстоянии равном или меньшем радиуса области. Их влияние определяется обычной Гауссианой, формула:

$$\text{influance} = e^{\frac{-distance^2}{2*radius^2}}$$

то есть, чем дальше элемент в пространстве признаков, тем меньше влияния он оказывает на сдвиг выбранного элемента. Распределение точек в пространстве признаков выглядит примерно так:

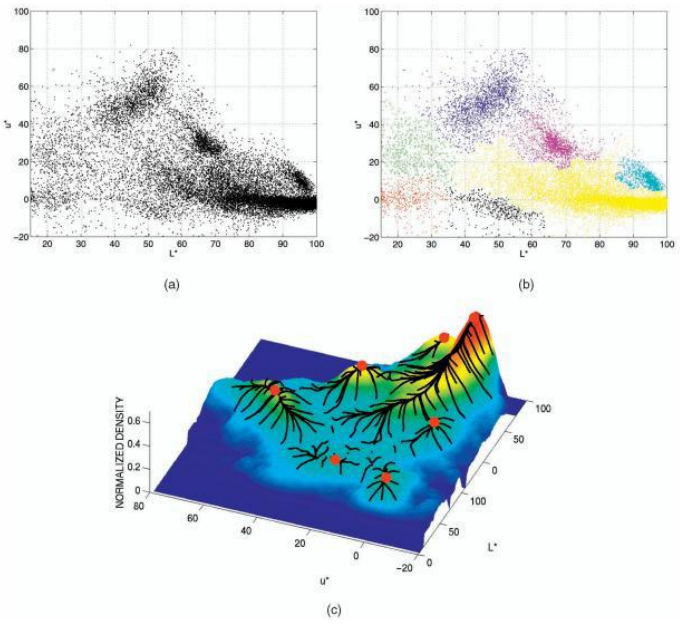


Иллюстрация точек в пространстве

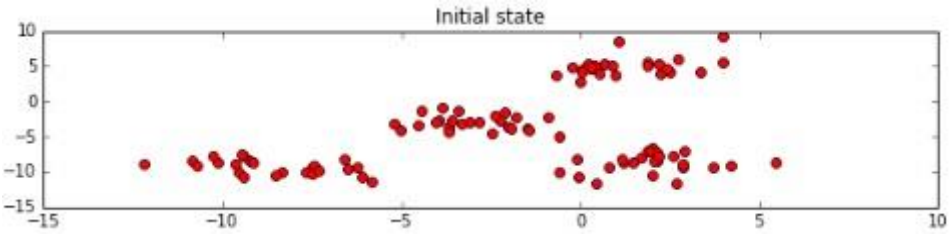
Здесь можно заметить сгущения точек в определенных областях. После работы алгоритма Mean Shift все точки сойдутся к своим локальным максимумам, они и будут считаться сегментом.

Результат работы Mean Shift

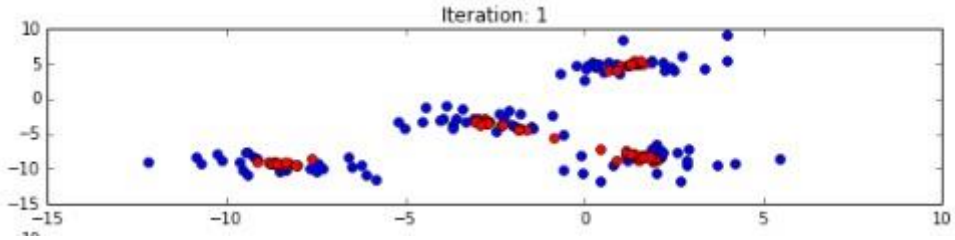
Как видно на изображении выше после сдвигов все элементы находятся в некоторых точках. На втором шаге такое объединение точек будет добавлено в один кластер.

Рассмотрим одну итерацию алгоритма. Синие — это изначальные точки, а красные — их центроиды на каждой итерации.

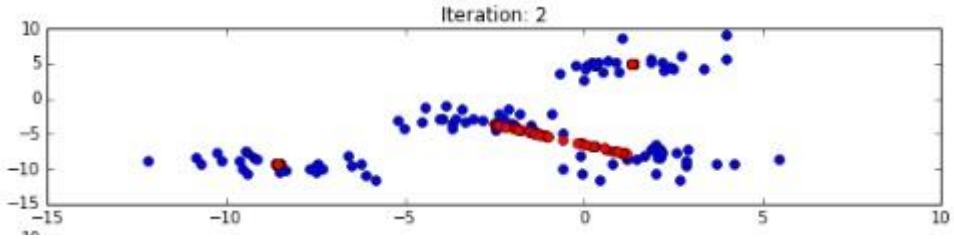
До начала работы алгоритма: Синие и красные точки находятся в одном положении. То есть количество кластеров равняется количеству точек.



Первая итерация. Центроиды двигаются в сторону средних значений.

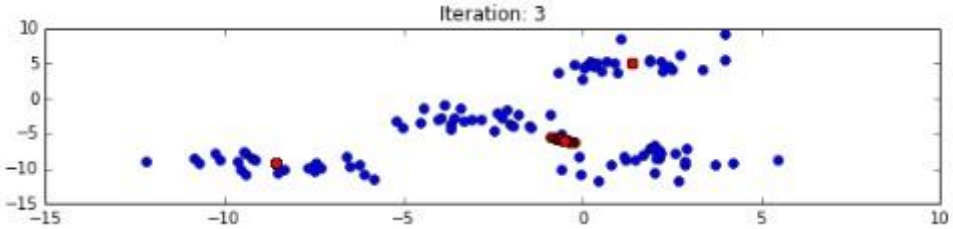


Вторая итерация. Два боковых кластера, уже сошлись. Два центральных

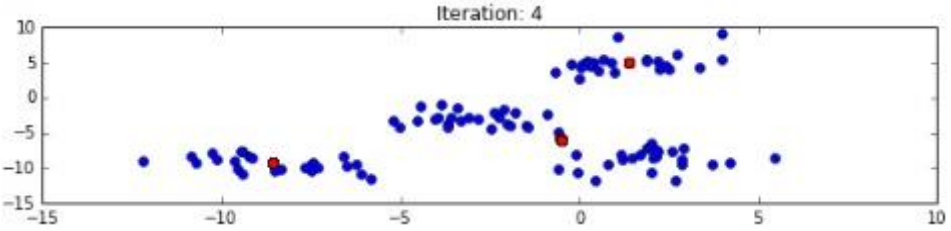


кластера двигаются по направлению друг к другу, скорее всего они сольются в один.

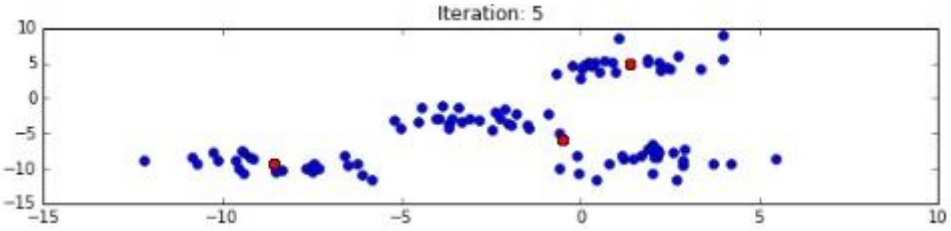
Третья итерация. Без особых изменений в левом и верхнем правом кластерах, положение их центроидов немного корректируется. Центроиды центральных кластеров уже совсем близко друг от друга. Чем ближе точки друг к другу тем сильнее они влияют на смещение центройд. Таким образом кластеры сливаются в один.



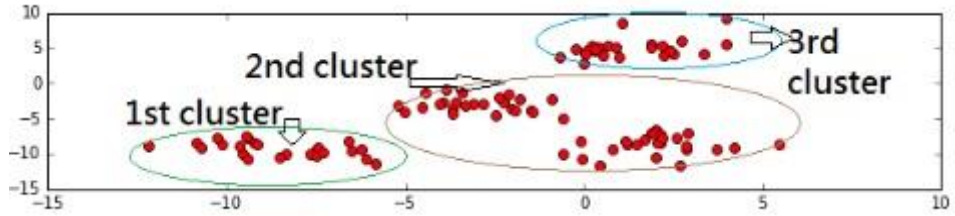
Четвертая итерация. Кластеры определились.



Пятая итерация. Конец алгоритма, так как центроиды перестали двигаться.



В результате Mean Shift определил 3 кластера.



Выбор пространства признаков и оценка расстояния

Достаточно выбрать такое пространство признаков, которое бы удовлетворяло требованиям сегментации полноцветных изображений. Для этого было принято решение объединить несколько простых пространств: цветового (RGB), пространственного (x, y), и яркостного. Для описания цветового пространства достаточно много вариантов для выбора, однако стандартное представления, то есть (RGB) показывает себя вполне не плохо. Таким образом получаем шестимерное пространство (x, y, r, g, b, brightness). Для оценки расстояния была выбрана евклидова метрика

$$\mathrm{D}(x_1, x_2) = \sqrt{\sum_{k=0}^{n} p_k * (x_{1_k} - x_{2_k})^2}$$

p_k – веса для каждого подпространства в метрике, описывают силу влияния расстояния в к-ом подпространстве на результирующее значение метрики. Таким образом алгоритм с разными параметрами может быть эффективен с различным типом изображений. Основная идея алгоритма заключается именно в этом, однако появляется новая проблема – поиск подходящих параметров. Таким образом получается 6 неизвестных (плюс радиус для Mean Shift) которые нужно найти, для этого необходимо ввести оценку работы алгоритма сегментации с указанными параметрами и использовать алгоритм для быстрого поиска параметров, при которых значение оценки максимально.

Глава 2. Генетический алгоритм

Принципы работы генетического алгоритма

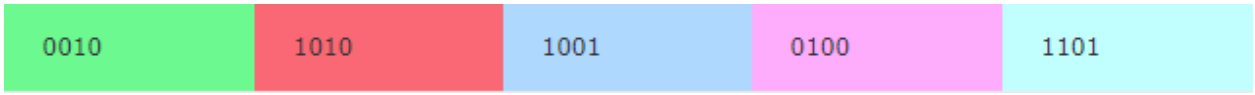
Генетические алгоритмы используются для нахождения параметров и решения задач оптимизации. Используется в NP сложных задачах или трудоемких алгоритмах. Примером подобных задач может служить

обучение нейронных сетей, иными словами подбор параметров (значений весов) таких, при которых достигается максимальный результат или минимальная ошибка. Так же генетического алгоритм основан на методе случайного поиска. Недостатком таких алгоритмов является то, что они не постоянны и их решения могут деградировать, то есть результат может ухудшаться в последующих итерациях, также нам неизвестно сколько понадобится времени для решения задачи. Для избежание таких недостатков, применяются хитрые эвристики, например, факты, проявившиеся в биологии. Например, могут использоваться методы, широко известные в сферах изучающих происхождения видов и эволюцию. Как известно, в процессе эволюции выживают наиболее приспособленные особи. Благодаря этому, потомки текущей популяции становятся более приспособленными, благодаря этому принципу происходит адаптация к окружающим условиям.

Представление объектов

В биологии любой организм может быть представлен своим фенотипом, который определяет, чем объект является в мире, и генотипом, который содержит всю информацию об объекте на уровне хромосомного набора.

Генотип – это совокупность всех цепочек генов. Его можно представить, как совокупность битовых последовательностей(цепочек), которые кодируют некоторую информацию. Обычно цепочки генов имеет фиксированную длину. В генотипе описаны признаки, которые будут закодированы в виде генов, которые представляют собой набор бит, достаточный для описания этого признака. Например, нам нужно закодировать 5 признаков, где значение каждого лежат в приделах от 0 до 16. Достаточно взять ген длиной 4 бита. И получим цепочку длиной 20 бит, который выглядит так:



Хромосома – это набор цепочек генов, количество цепочек в разных хромосомах может отличаться.

При этом каждый ген, то есть элемент информации генотипа, имеет свое отражение в фенотипе. Таким образом, для решения задач нам необходимо представить каждый признак объекта в форме, подходящей для использования в генетическом алгоритме. Все дальнейшее функционирование механизмов генетического алгоритма производится на уровне генотипа, позволяя обойтись без информации о внутренней

структуре объекта, что и обуславливает его широкое применение в самых разных задачах.

Основные генетические операторы

Одним из важнейших принципов эволюции является то, как передавать признаки от родителей к потомкам. В генетических алгоритмах существует операция, которая отвечает за передачу признаков родителей потомкам, которая называется скрещивание (его также называют кроссовер или кроссинговер). Этот оператор отвечает за передачу признаков от родителей к потомкам. Описание работы одноточечного кроссовера:

1. Выбираются две особи из популяции, которые будут родителями.
2. определяются (обычно случайным образом) одна или несколько точек разрыва.
3. потомок определяется как конкатенация частей первого и второго родителя.

Рассмотрим работу кроссовера:

Хромосома_1:	0000000000
Хромосома_2:	1111111111

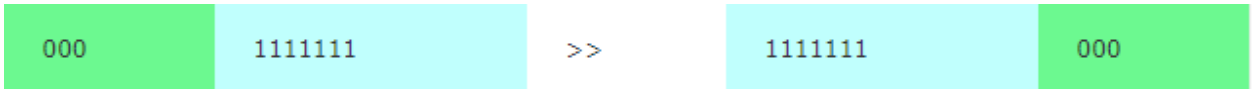
В данном случаи разрыв происходит после 3-го бита хромосомы

Хромосома_1:	0000000000	>>	000	1111111	Результирующая_хромосома_1
Хромосома_2:	1111111111	>>	111	0000000	Результирующая_хромосома_2

После этого равновероятно выбирается одна хромосома из двух результирующих на место потомка.

Следующая операция генетического алгоритма отвечает за поддержку разнообразия особей с популяции. Она называется мутацией. При использовании данной операции с определенной вероятностью инвертируются каждый бит в хромосоме.

Иногда, может использоваться операция под названием инверсии, она делит хромосому на две части, и меняет их местами. Это можно изобразить следующим образом:



В принципе для функционирования генетического алгоритма достаточно первых двух генетических операторов, но на практике применяют еще, и некоторые дополнительные операторы или модификации этих двух операторов. Например, кроссовер может быть не одноточечный (как было описано выше), а многоточечный, когда формируется несколько точек разрыва (чаще всего две). Кроме того, в некоторых реализациях алгоритма оператор мутации представляет собой инверсию только одного случайно выбранного бита хромосомы.

Схема функционирования генетического алгоритма

Ниже приведено базовое описанию функционирования генетического алгоритма. Рассмотрим схему функционирования генетического алгоритма в его классическом варианте.

1. Инициировать начальный момент времени t=0. Случайным образом сформировать начальную популяцию, состоящую из k особей. B_0={A_1, A_2,…, A_k}
2. Вычислить приспособленность каждой особи F_{Ai}=fit(A_i) i=1…k_i=1…k и популяции в целом F_t=fit(B_t) (также иногда называемую термином фитнес). Значение этой функции определяет насколько хорошо подходит особь, описанная данной хромосомой, для решения задачи.
3. Выбрать особь A_c из популяции A_c=Get(B_t)
4. С определенной вероятностью (вероятностью кроссовера P_c) выбрать вторую особь из популяции A_{c1}=Get(B_t) и произвести оператор кроссовера A_c = Crossing(A_c, A_{c1}).
5. С определенной вероятностью (вероятностью мутации P_m) выполнить оператор мутации A_c = mutation(A_c).
6. Поместить полученную хромосому в новую популяцию insert(B_t+1, A_c).
7. Выполнить операции, начиная с пункта 3, k раз.

8. Увеличить номер текущей эпохи t=t+1.
9. Если выполнилось условие останова, то завершить работу, иначе переход на шаг 2.

Реализация, использованная в работе

В данной работе рассматривалась особь, которая состоит из одной хромосомы которая в свою очередь состоит из одной цепочек генов фиксированной длины. Цепочка состоит из переменных, которые являются входными параметрами алгоритма Mean Shift.

$$[\ radius \quad xy \quad rgb \quad brightness\ ]$$

Цепочки генов

Где *radius* – радиус в котором выбираются соседи для сдвига точки к их среднему, xy – коэффициент влияния расстояния в пространстве расположения пикселей на изображении, rgb – коэффициент влияния близости для каждого канала в отдельности и brightness – яркости соответственно. Таким образом получается 6 параметров для полного перебора которых понадобится 100^4 т.к. в для каждого параметра примерно 100 возможных значения.

Выбор родителей

Выбор родителей осуществляется по принципу (стохастический отбор с остатком):

1. Вычислим отношение для каждой особи из популяции $\frac{\text{значение фитнес функции особи}}{\text{среднее по всей популяции}}$
2. Целая часть означает сколько раз данная особь войдет в состав родителей.
3. Дроби будут показывать вероятность попадания, если промежуточная популяция родителей не заполнилось до конца. Просто в цикле будем сравнивать вероятность со случайным значением, пока вероятность не окажется больше, тогда эта особь становится родителем.
4. Так же для того что бы новое поколение не деградировало лучшая особь сразу переходит в следующее поколение без мутаций. Плюс

такого подхода заключается в том, что количество шагов в алгоритме заметно уменьшается, за счет того, что алгоритм быстрее сходиться к решению.

Скрещивание

Так как возможные родители уже выведены в отдельную группу, можно просто случайным образом выбирать двух разных родителей и производить скрещивание. Для получения нового потомка был использован одноточечный кроссовер. Скудность разнообразие при одноточечном кроссовере была частично компенсирована большой вероятностью мутации.

Мутация

Следующий генетический оператор предназначен для того, чтобы поддерживать разнообразие особей с популяции. Он называется оператором мутации. При использовании данного оператора каждый бит в хромосоме с определенной вероятностью инвертируется. Я использовал 40% вероятность инвертировать каждый бит. Таким образом если два родителя одинаковые, вероятность получить кардинально отличающийся результат значительно выше.

Целевая (фитнес) функция

Для оценки работы алгоритма не получилось обойтись без помощи человека. Это связано с тем что объективно оценить вывод алгоритма сегментации нельзя, нежен учитель, который скажет правильно ли была проделана работа. В данной работе было принято решение использовать такую оценку: Необходимо выбрать пиксель, который должен сходить в предполагаемый класс, и обозначить область в которой он должен находиться, кругом.

Красная точка – точка которая входит в класс, красный круг – область.

Целевая функция выглядит таким образом:

$$f = \sum g(x_i), \quad \text{где} \quad g(x_i) = \begin{cases} +1, & \text{в облати} \\ -5, & \text{вне области} \end{cases}$$

x_i – Пиксель в кластере

```
int Metric(const Pixel& pixel, ImVec2 center, float radius)
{
    float distance = sqrt(pow(pixel.x - center.x, 2) + pow(pixel.y - center.y, 2));
    return distance <= radius ? 1 : -5;
}

inline int CalculateTargetValue(const std::vector<Cluster<Pixel>>& clusters,
                                ImVec2 class_point,
                                ImVec2 center,
                                float radius)
{
    int cluster_id = GetClusterByPixel(clusters, class_point);

    int result = 0;
    for (const Pixel& pixel : clusters[cluster_id])
    {
        result += Metric(pixel, center, radius);
    }
    return result;
}
```

код оценки работы алгоритма

За каждый пиксель, который вошел в круг значение увеличивается на один, за выход из области значение уменьшается на пять, задача алгоритма найти максимально значение целевой функции. Получается, генетический алгоритм пытается подобрать параметры алгоритма Mean Shift таким образом, чтобы указанный кластер занимал большую часть указанной области. В данной задаче идея нейронных сетей выглядит очень интересно, в том плане что можно хранить результаты выполнения в каком-то виде и использовать их для решения подобных проблем, тем самым полностью отказаться от участия человека в работе алгоритма. Таким образом можно добавить в алгоритм дополнительные настраиваемы веса, и проводить их настройку в процессе обучения.

Глава 3. Реализация

Использованные инструменты

Все алгоритмы были реализованы с нуля, для их написания использовались – C++ для написания логики и некоторых вычислений, OpenCL для использования видеокарты в вычислениях. Так же для работы с изображениями использовался OpenGL, а для создания визуальной части приложения библиотека SDL2 и ImGui. Для удобной работы с изображениями и вычислениями был написан небольшой фреймворк абстрагирующий работу с памятью, вычислениями на процессоре и видеокарте.

Время работы алгоритмов

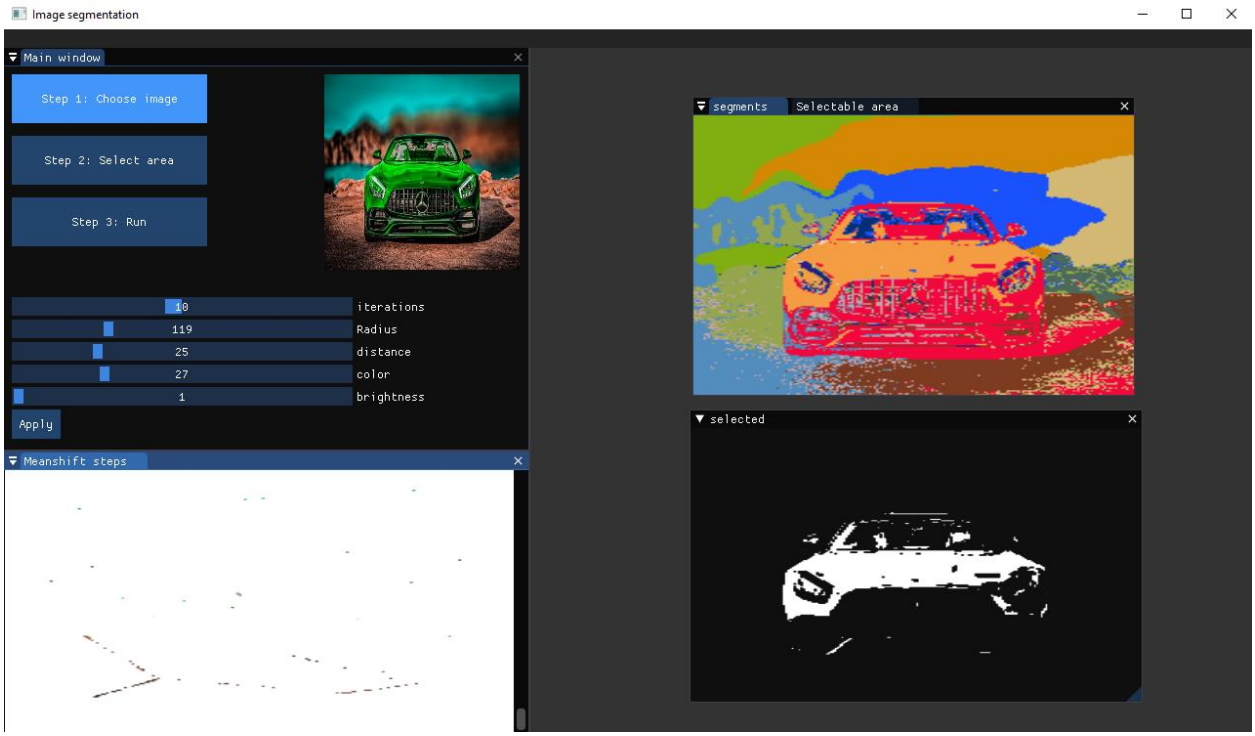
Для начала рассмотрим сложность алгоритма Mean Shift. Она оценивается как квадрат от количества пикселей изображения умноженный на работу ядра или $(\text{высота} * \text{ширина})^2 * (\text{время работы ядра})$. Ядро – это термин OpenCL обозначающий код, который, должен быть выполнен для каждого элемента. Генетический алгоритм работает в среднем 10 итераций. Для каждой итерации нужно вычислить целевое значение для каждой особи, их в популяции 5, таким образом получаем 50 вызовов алгоритма Mean Shift.

Оптимизация

Уже для HD изображений запуск алгоритма Mean Shift это огромное число операций, на выполнение которых уйдут минуты. Поэтому первое же решение по оптимизации заключается в том, чтобы уменьшить изображение. Было использовано обычное линейное масштабирование до фиксированного размера 200x200 пикселей. Алгоритм Mean Shift обладает свойством массового параллелизма (каждую его часть можно считать независимо от других), таким образом он разбивается на большое число маленьких подзадач. Это идеально подходит для вычисления на видеокарте. Для реализации Mean Shift была использована библиотека OpenCL, это GPGPU (General-purpose computing on graphics processing units) технология, позволяющая запускать код на видеокартах. Разница с 8-ми ядерным процессором составила около 40-50 раз.

Приложение

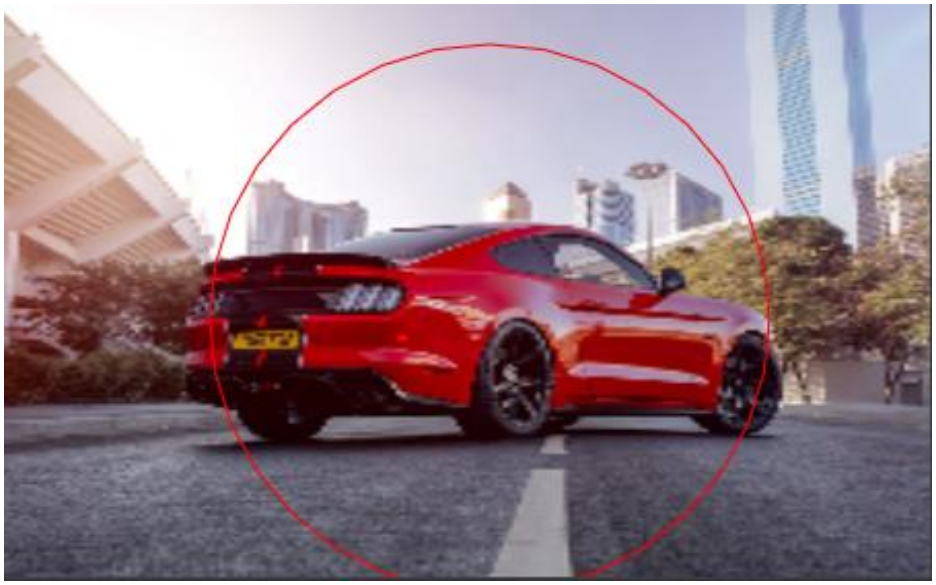
В ходе работы было разработано приложение с графическим интерфейсом позволяющее автоматизировать работу запуска алгоритма и наглядно проиллюстрировать все шаги работы.



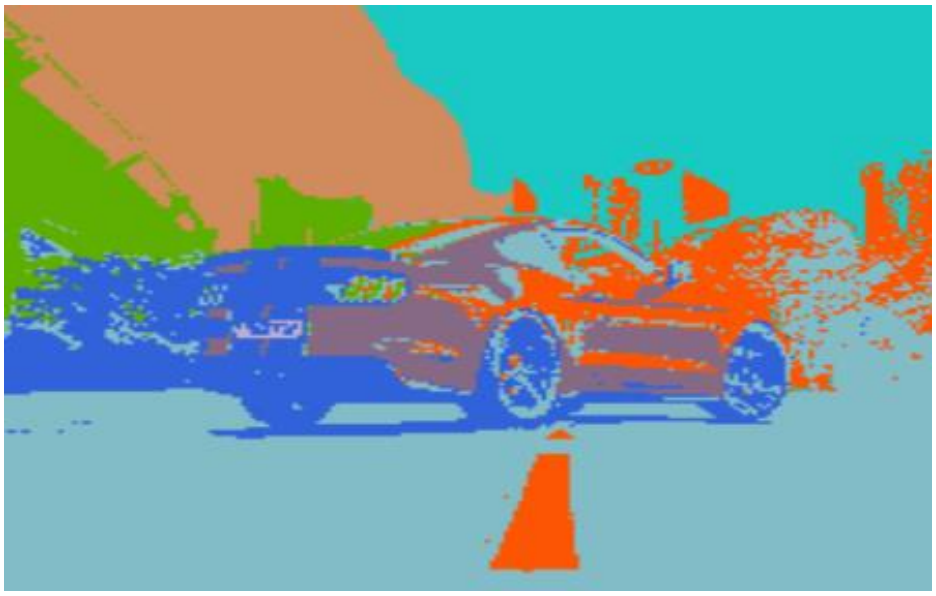
Приложение представлено одним окном, в котором можно выбрать изображение, запустить алгоритм и посмотреть результаты. Так же можно вручную изменять параметры, для того чтобы наглядно демонстрировать изменение каждого из них.

Результаты работы

Далее представлены результаты работы алгоритма в виде трех изображений, первое – оригинальное изображение с выделенной областью и классом, который должен занимать указанную область (красная точка и красный круг), второе – сегменты, выделенные алгоритмом, области покрашены в случайные цвета для наглядности и третье - выделенный класс (сегмент).

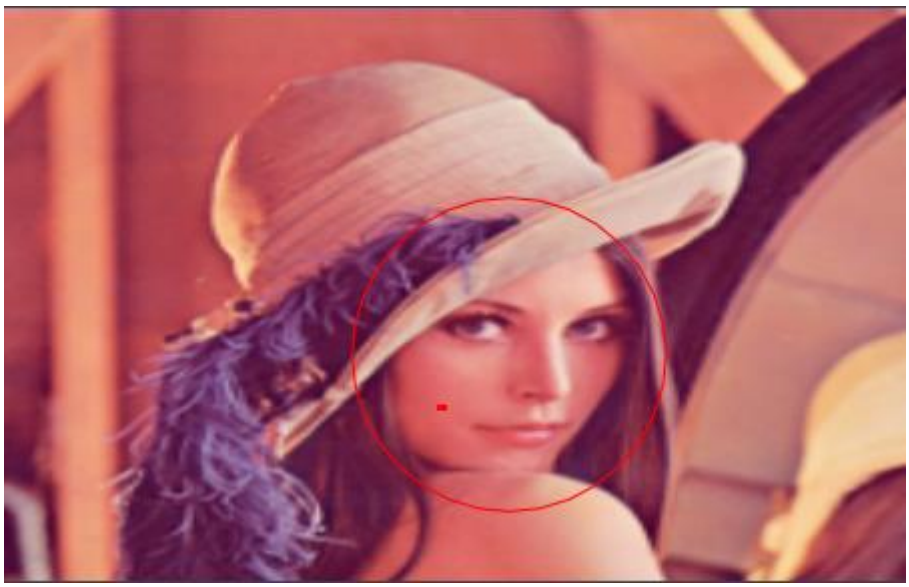


Исходное изображение



Изображение, разбитое на сегменты

Выделенный сегмент



Исходное изображение

Изображение, разбитое на сегменты

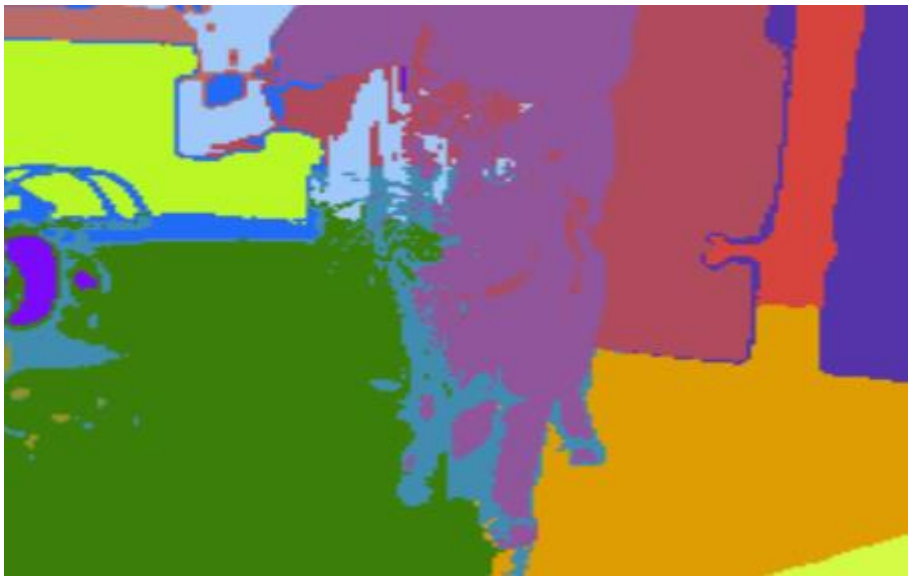
Выделенный сегмент

Исходное изображение

Изображение, разбитое на сегменты

Выделенный сегмент

Исходное изображение



Изображение, разбитое на сегменты

Выделенный сегмент

Список литературы

1. П. А. Чочиа. СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АНАЛИЗА РАССТОЯНИЙ В ПРОСТРАНСТВЕ ПРИЗНАКОВ дата обращения: 10. 04.2021. URL:https://www.sibran.ru/upload/iblock/6c5/6c577369ee7b71132c5a75b17718a6ff.pdf

2. И. Г. Ханыков. КЛАССИФИКАЦИЯ АЛГОРИТМОВ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ: дата обращения: 10. 04.2021. URL:http://pribor.ifmo.ru/file/article/18273.pdf

3. Т.В. Панченко. Генетические алгоритмы. Издательский дом «Астраханский университет» 2007.

4. D. Comaniciu, P. Meer Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002.

5. Р. Гонсалес, Р. Вудс Цифровая обработка изображений, Москва: Техносфера, 2005.