|  |  |
| --- | --- |
| Руководство разработчика | |
| Название ПО | ClustSystem |
| Тип ПО | Исследовательский |
| Версия | 1.0 |
| Автор/-ы руководства | Минеев С. А. [mineeff20@yandex.ru] |



2023–2024 г. Москва

Содержание

[Описание 3](#_Toc155442253)

[1 Установка 3](#_Toc155442254)

[2 Требования к ПО 5](#_Toc155442255)

[3 Архитектура ПО 6](#_Toc155442256)

[4 Примеры работы ПО 10](#_Toc155442257)

[4.1 Генерация данных на примере с распределениями 10](#_Toc155442258)

[4.2 Генерация данных с помощью make\_\* функций 20](#_Toc155442259)

[4.3 Загрузка изображений 23](#_Toc155442260)

[4.4 Кластеризация данных 24](#_Toc155442261)

[4.5 Настройки программы 27](#_Toc155442262)

[4.6 Анализ данных 30](#_Toc155442263)

[Заключение 31](#_Toc155442264)

[Приложение 1: Кода 31](#_Toc155442265)

# Описание

ClustSystem – это ПО предназначенное для генерации, кластеризации и анализа данных.

Версия v1.0 позволяет кластеризовать изображения формата ‘png’, ‘jpg’, сгенерированные данные.

В версии v1.0 поддерживаются следующие виды генерации данных:

1. Генерация N-мерных векторов размерности K.
2. Генерация различных фигур и их комбинаций с использованием следующих функций: make\_blobs, make\_circle, make\_moons и make\_dna.

Разработчики ПО:

Студенты группы КММ0-01-23

* Мешкова О. В.
* Плохотина Ю. С.
* Николаев М. А.
* Корезин В. А.
* Залетин Н. А.
* Грибань М. С.
* Минеев С. А.

# Установка

Рекомендуется использовать python 3.11.7 . В версии python 3.12 были внесены значительные изменения, которые затрудняют установку окружения.

Установка начинается с обновления pip

python -m pip install --upgrade pip

Установите окружение и активируйте его.

…

Установите пакеты

Установите следующие зависимости:

#mainwindow.py

python.exe -m pip install Pillow

python.exe -m pip install PySide6

python.exe -m pip install matplotlib

# ClustringAlgorithms.py:

python.exe -m pip install opencv-python

python.exe -m pip install numpy

python.exe -m pip install pyclustering

python.exe -m pip install --force-reinstall -v “scikit-learn==1.3.2”

! Не рекомендуется брать 1.4rc1.

# Leader.py + SettingApp.py:

python.exe -m pip install qstylizer

В качестве IDE вы можете использовать PyCharm или любой другую IDE, поддерживающую программирование на python

# Требования к ПО

В следующей таблицы представлены основные требования, которым удовлетворяет разработанное ПО.

Таблица 1 – Требования к ПО

|  |  |  |
| --- | --- | --- |
| **№** | **Требование** | **Примечание** |
|  | | |
| 1 | Запуск сеанса работы | Реализовано |
| 1.1 | Установка параметров кластеризации | Реализовано |
| 1.1.1.v1 | Загрузка набора данных для кластеризации | Только изображения форматов png/jpg |
| 1.1.1.v1.1 | Кластеризации изображений | Реализовано |
| 1.1.1.v2 | Выбор параметров генерации набора данных кластеризации | Реализовано |
| 1.1.1.v2.1 | Выбор размерности фитчей (количество фитчей кластеризации) | Реализовано |
| 1.1.1.v2.2 | Выбор задания стохастического распределения данных как для всего набора фитчей, так и для каждой из фитч | Имеется следующие функции для генерации данных по распределениям:  • Нормальное,  • Показательное,  • Стандартное,  • Биноминальное. |
| 1.1.1.v2.3 | Задание распределений с параметрами для фитч и генерация данных | Реализовано |
| 1.1.2 | Выбор алгоритма(-мов) кластеризации | Имеется следующие способы кластеризации:  Birch из sklearn,  Birch, Cure, Rock из pyclustering. |
| 1.1.3 | Задание параметров кластеризации (количество кластеров, и т. д. …) | Реализовано |
| 1.2 | Запуск кластеризации | Реализовано |

Продолжение таблицы 1 – Требования к ПО

|  |  |  |
| --- | --- | --- |
| 1.3 | Формирование результатов кластеризации | Реализовано |
|  | | |
| 1.4 | Анализ результатов кластеризации | - |
| 1.4.1 | Анализ результатов на основе алгоритмов оценки качества кластеризации | Реализовано |
|  | | |
| 1.5 | Выгрузка результатов кластеризации | Реализовано |

# Архитектура ПО

На Рис.1-2 продемонстрирована архитектура программы, отражающая основные связи между компонентами, их обозначения и типы в следующей форме:

name\_var[value]: type,

где name\_var – имя переменной/класса/компонента в коде,  
 value – значение, type – тип переменной.



Рис. 1 – Диаграмма компонентов cnv\_wg и spl.



Рис. 2 – Диаграмма компонентов [основная]

2.2 Таблицы параметров по умолчанию

Таблица 2 – Параметры по умолчанию, принимаемые функциями make\_\*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index | type | default\_value [Example] | name\_column | make\_blobs | make\_circles | make\_moons | make\_dna | make\_spheres |
| 1 | int32 | 100 | n\_samples | + | + | + | + | + |
| 2 | int32 | 2 | n\_features | + |  |  |  |  |
| 3 | int32 | None | centers | + |  |  |  |  |
| 4 | float | 1.0 | cluster\_std | + |  |  |  |  |
| 5 | tuple[float, float] | [-0.1,0.1] | center\_box | + |  |  | + |  |
| 6 | int | None | random\_state | + | + | + |  | + |
| 7 | float | 0.8 | factor |  | + |  |  | + |
| 8 | float | None | noise |  | + | + |  | + |
| 9 | float | 1.0 | norm | + | + | + |  |  |
| 10 | float | 0.0 | y | + | + | + |  |  |
| 11 | float | 0.0 | z | + | + | + |  |  |
| 12 | bool | False | shuffle | + | + | + |  | + |
| 13 | bool | False | return\_centers | + |  |  |  |  |

Подробное описание параметров:

1. n\_samples – задает количество точек генерации.
2. n\_features – задает размерность фитч.
3. centers – количество центров кластеров.
4. cluster\_std – стандартное отклонение кластеров.
5. center\_box – ограничивающая рамка для каждого центра кластера, когда центры генерируются случайным образом.
6. random\_state – определяет генерацию случайных чисел для создания набора данных.

Источник: <https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html>

1. factor – масштабный коэффициент между внутренним и внешним кругом в диапазоне [0, 1).
2. noise – стандартное отклонение гауссовского шума, добавленного к данным.

Источник: <https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_circles.html>

1. norm – коэффициент нормировки. Используется для нормировки данных. Например, когда сгенерированные make\_blobs нужно расположить внутри круга можно задать norm=15.0 и будет произведено покоординатное деление точек на этот коэффициент и тем самым make\_blobs окажутся внутри make\_circles.
2. y – позволяет сместить сгенерированные данные относительно оси Y. Используется только для одномерных данных.
3. z – позволяет сместить сгенерированные данные относительно оси Z. Используется для двумерных данных.
4. shuffle – Следует ли перетасовывать точки.
5. return\_centers – возвращать ли точки.

Таблица 3 – Параметры по умолчанию, принимаемые методами кластеризации

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Параметр | Значение по умолчанию | Тип | BIRCH\_SKLEARN | BIRCH\_PYCLUSTERING | CURE | ROCK |
| 0 | used | False | bool | + | + | + | + |  |
| 1 | **n\_clusters** | 3 | uint | + | + | + | + |
| 3 | **branching\_factor** | 50 | uint | + | + |  |  |
| 2 | **threshold** | 0.5 | float | + |  |  | + |
| 4 | compute\_labels | True | bool | + |  |  |  |
| 5 | copy | True | bool | + |  |  |  |
| 6 | max\_node\_entries | 200 | uint |  | + |  |  |
| 7 | diameter | 0,5 | double |  | + |  |  |
| 9 | entry\_size\_limit | 500 | uint |  | + |  |  |
| 10 | diameter\_multiplier | 1.5 | double |  | + |  |  |
| 8 | type\_measurement | EUCLIDEAN | struct type\_measument |  | + |  |  |
| 11 | **ccore** | True | bool |  | + | + | + |
| 12 | number\_represent\_points | 5 | uint |  |  | + |  |
| 13 | compression | 0.5 | double |  |  | + |  |
|  |  |  |  |  |  |  |  |
| 14 | eps | 2.0 | double |  |  |  | + |

Подробное описание параметров:

1. used – установите флаг в колонке метода кластеризации, если хотите, чтобы выбранный метод произвел кластеризацию. По умолчанию у всех методов кластеризации данный параметр имеет состояние False, что не позволяет кластеризовать данные.
2. n\_clusters - количество кластеров после заключительного этапа кластеризации, на котором подкластеры из листьев рассматриваются как новые samples(точки).
3. branching\_factor - максимальное количество подкластеров CF в каждом узле.
4. threshold - Радиус подкластера, полученного путем объединения новой выборки и ближайшего подкластера, должен быть меньше порогового значения.
5. compute\_labels - cледует ли вычислять метки для каждого соответствия. В случае отсутствия данного флага метки получаются из метода sklearn.birch.predict().
6. copy - cледует ли создавать копию данных.

Источник: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html>

1. max\_node\_entries - Максимальное количество записей, которые могут содержаться в каждом конечном узле в CF-дереве.
2. diameter - CF-входной диаметр, используемый для построения CF-дерева, он может увеличиться, если превышен 'entry\_size\_limit'.
3. entry\_size\_limit - Максимальное количество записей, которое может храниться в CF-дереве, если оно превышено при создании, то "диаметр" увеличивается и CF-дерево перестраивается.
4. diameter\_multiplier - множитель, который используется для увеличения диаметра при превышении 'entry\_size\_limit'.
5. type\_measurement - тип измерения, используемый для вычисления показателей расстояния.
6. ccore - Если True, то для обработки используется часть библиотеки C++.

Источник: <https://pyclustering.github.io/docs/0.10.1/html/d6/d00/classpyclustering_1_1cluster_1_1birch_1_1birch.html>

1. number\_represent\_points - количество репрезентативных точек для каждого кластера.
2. compression - коэффициент определяет уровень сжатия точек представления по отношению к среднему значению для нового созданного кластера после объединения на каждом шаге. Обычно он определяется от 0 до 1.

Источник: <https://pyclustering.github.io/docs/0.10.0/html/dc/d6d/classpyclustering_1_1cluster_1_1cure_1_1cure.html>

1. eps - радиус связности (порог подобия), точки являются соседними, если расстояние между ними меньше радиуса связности.

Источник:

<https://pyclustering.github.io/docs/0.8.2/html/d8/dde/classpyclustering_1_1cluster_1_1rock_1_1rock.html>

# Примеры работы ПО

## Генерация данных на примере с распределениями

Запустите программное обеспечение, поставляемое вместе с данным руководством. При первом запуске программы программа может в зависимости от операционной системы по стилю немного отличаться, но расположение компонентов и по функционированию совпадать [Рис. 1].

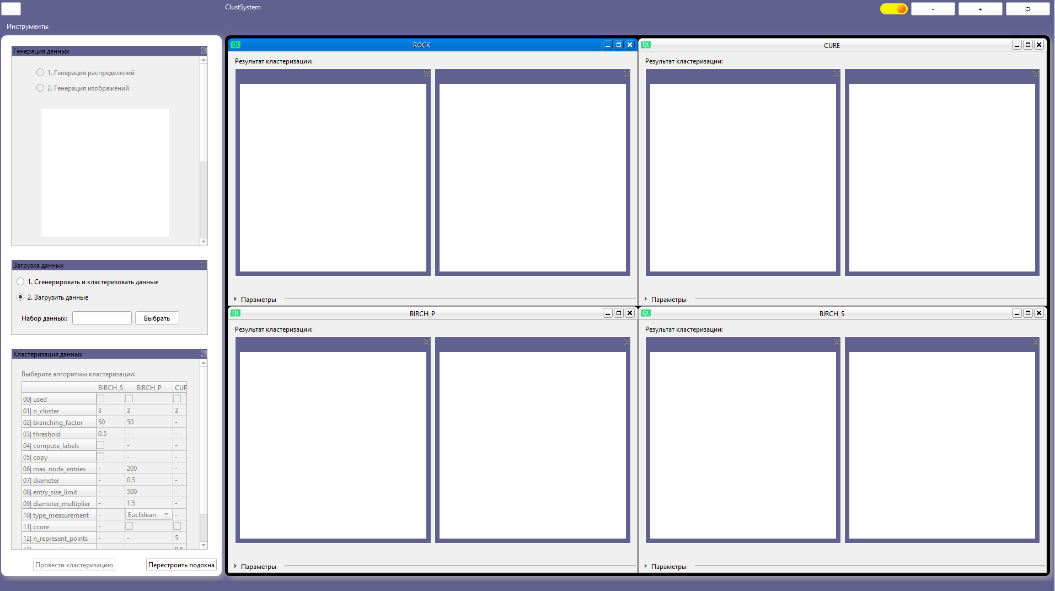


Рис. 3 – Демонстрация основного окна программы при первичном запуске

Для генерации данных по распределениям в панель “Загрузка данных” выберите пункт “1. Сгенерировать и кластеризовать данные”. При выборе этого пункта у вас должна стать активной панель “Генерация данных” [Рис. 3].

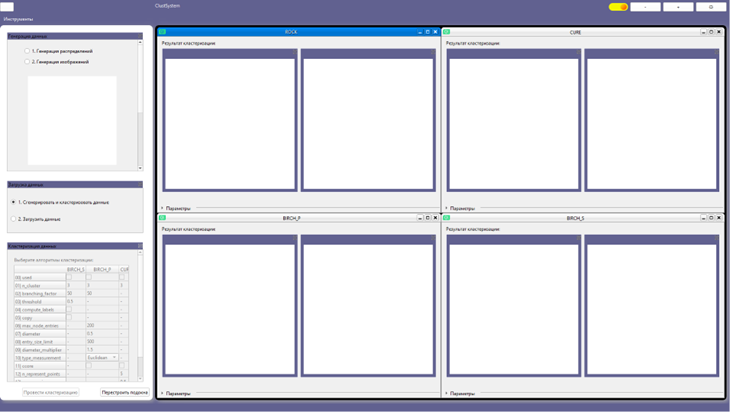


Рис. 4 – Демонстрация расположения панели “Генерация данных”

Выберите на панели “Генерация данных” пункт “1. Генерация распределений” [Рис. 5]

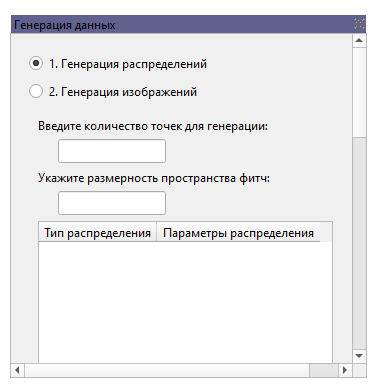


Рис. 5 – Выбор пункта “1. Генерация распределений”

При необходимости вы можете отделить данную панель от основного окна, нажав на заголовок “Генерация данных” и растянуть как вам будет удобнее:

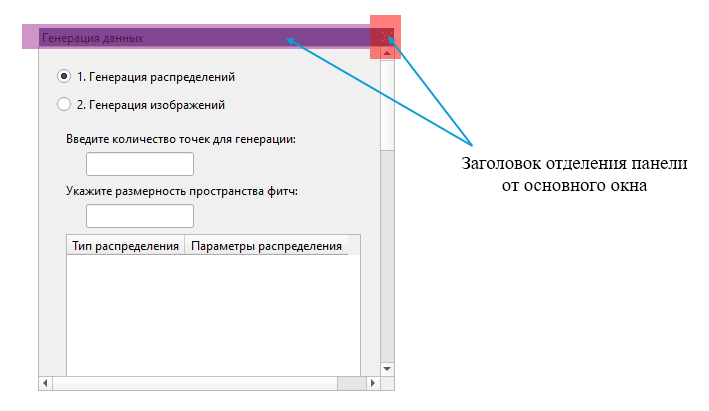


Рис. 4 – Демонстрация областей захвата для отделения подокон от основного окна.

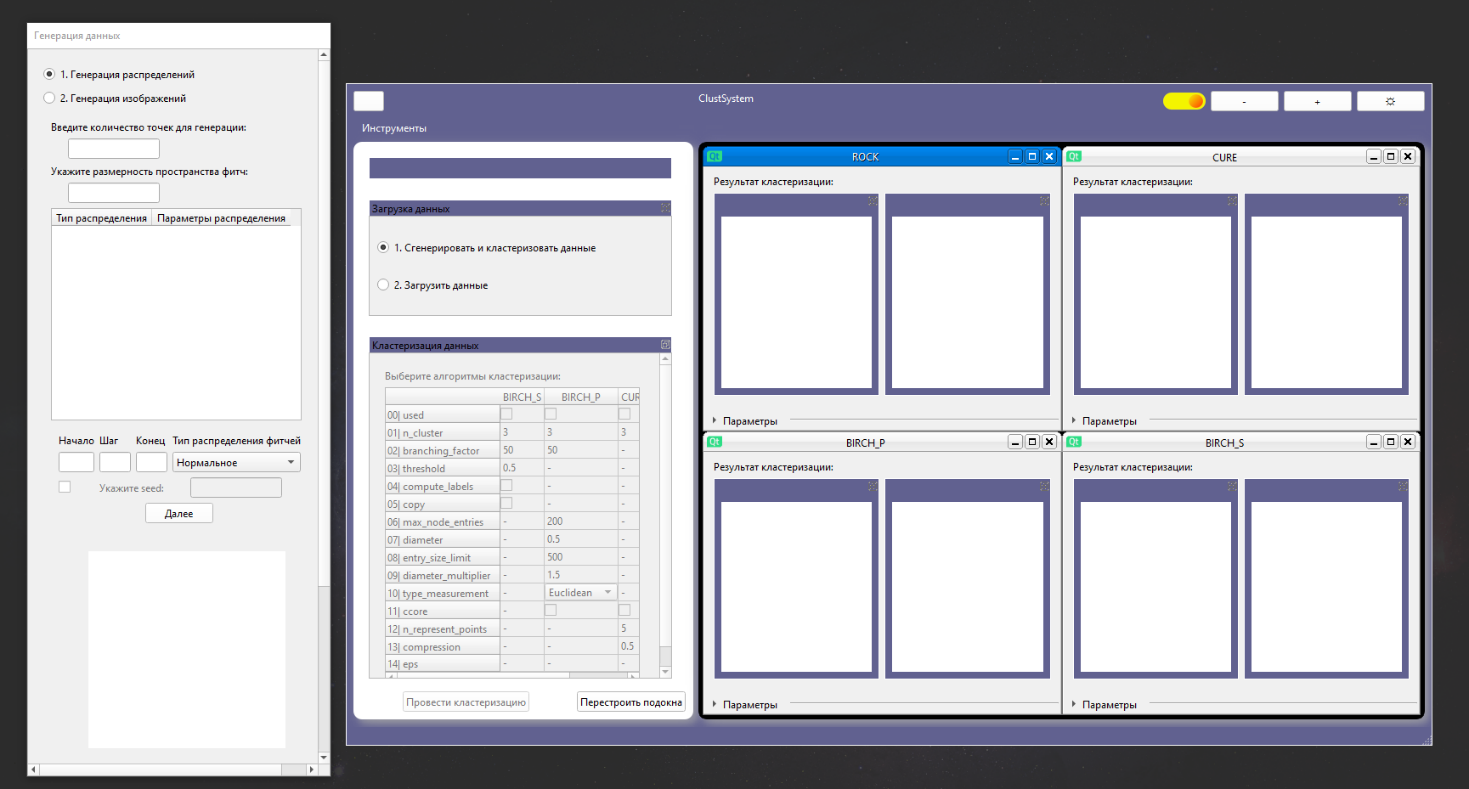


Рис. 5 – Демонстрация от соединенной панели ”Генерация данных”

Далее укажите количество точек, для которых в совокупности будет генерироваться распределение.

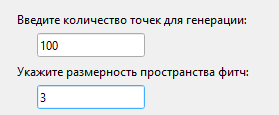


Рис. 6 – Демонстрация задания начальных параметров

Затем, необходимо определить тип распределения для каждой из фитч (На Рис. 6 их 3) тип распределения и параметры распределения.

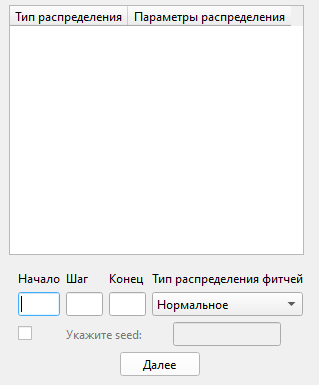


Рис. 7 – Демонстрация задания параметров распределения

Укажите диапазон фитч, для которых требуется установить определенный тип распределения. Параметр “Начало”, указывает на то, от какой фитчи надо задать выбранный далее тип распределения. Параметр “Шаг” задаёт сдвиг от начальной фитчи до следующей, для которой будет определенно выбранное распределение, а параметр “Конец” определяет номер фитчи до которой включительно будет определенно выбранное распределение, при условии, что шаг распределения не перекрывает данный номер.

Стоит отметить, что нумерация начинается с единицы и пока для всех фитч не будут заданы распределения не появится кнопка “Сгенерировать распределения”.

В версии v1.0 поддерживаются следующие типы распределений:

Номальное, Показательное и Биноминальное.

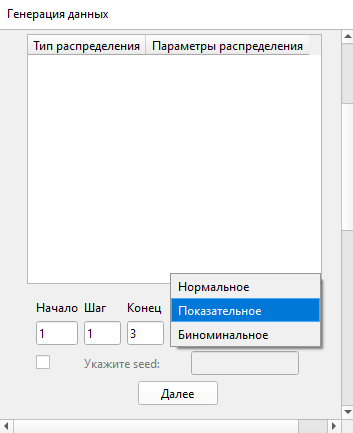


Рис. 8 – Демонстрация вариантов выбора типов распределения.

Выберите нужный вам тип распределения, после казания тех фитч, для которых оно должно быть определено и нажмите кнопку “Далее”.

В зависимости от выбранного типа распределения у вас будут появляться те или иные параметры, которые необходимо задать сразу после нажатия кнопки “Далее”.

|  |  |
| --- | --- |
|  |  |
|  |  |

Рис. 9-12 – Демонстрация вариантов задания распределений и их параметров

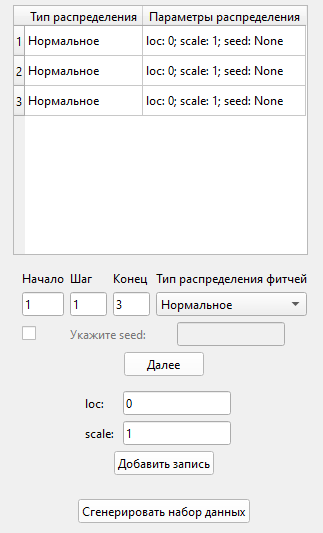


Рис. 13 – Демонстрация заданных параметров

После каждого выбора распределения и задания соответствующих параметров, нажмите на кнопку “Добавить запись”, чтобы в таблицы появились сведения для какой фитчи, какие параметры были заданы.

Стоит отметить, что в данной таблице, представленной на Рис. 13 номер строки совпадает с номером фитчи и запись в строке определенных параметров будет интепретироваться программай как применение этих параметров для генерации данных по конкретно этой фитче.

После того, как будут заполнены все данные появится кнопка “Сгенерировать набор данных”.

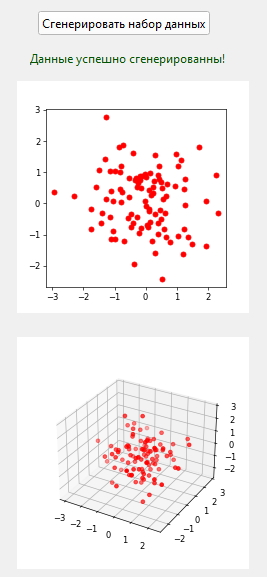


Рис. 14 – Демонстрация сгенерированных данных по распределениям фитч

Стоит отметить, что данные, сгенерированные по распределениям и изображениям, автоматически сохраняются в файлы соответствующие файлы рядом с программой в файлы “dataPoints.csv”, “ dataLabels.csv”, image2D и image3D.

## Генерация данных с помощью make\_\* функций

На панели “Загрузка данных” выберите пункт “Сгенерировать и кластеризовать данные”. Затем на панели “Генерация данных” выберите пункт “2. Генерация изображений” и нажмите на кнопку “Добавить запись”:

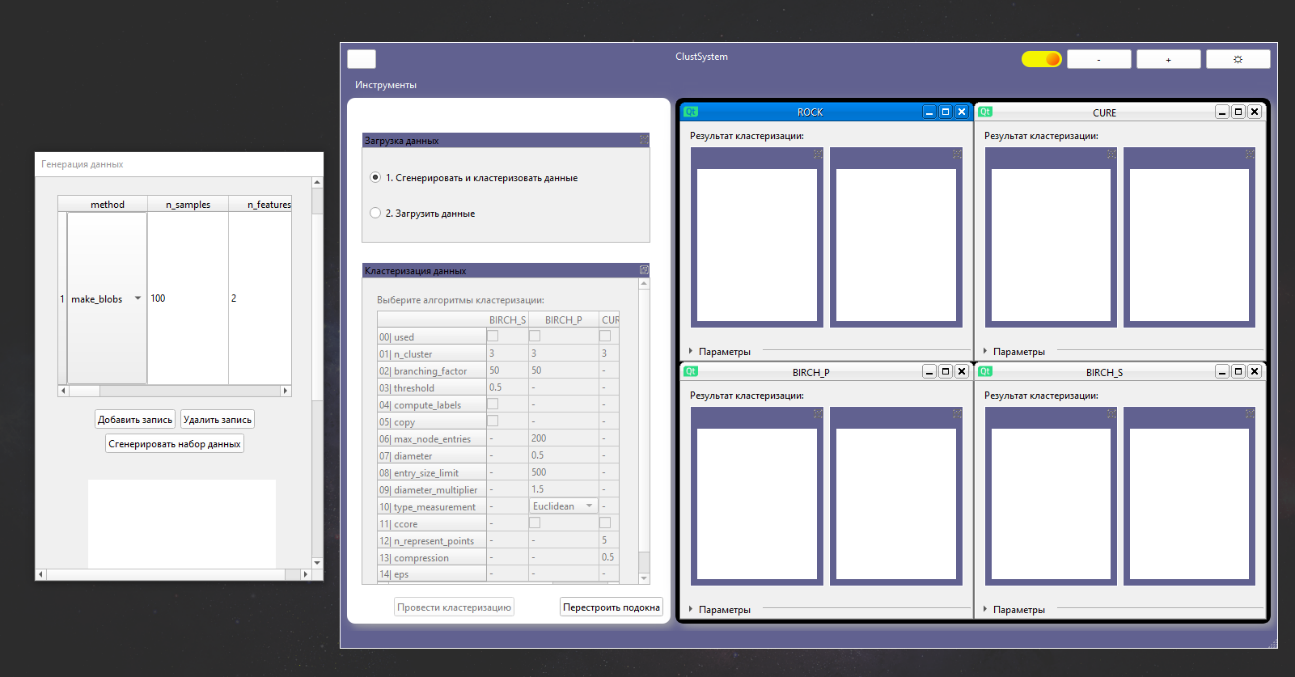


Рис. 15 – Демонстрация панели “Генерация данных” для случая “2. Генерация изображений”

Выберите нужные методы генерации для единого(одного) изображения и по необходимости задайте параметры не по умолчанию. Стоит отметить, что при наведении на компонент мышкой, у вас будет всплывает подсказка.

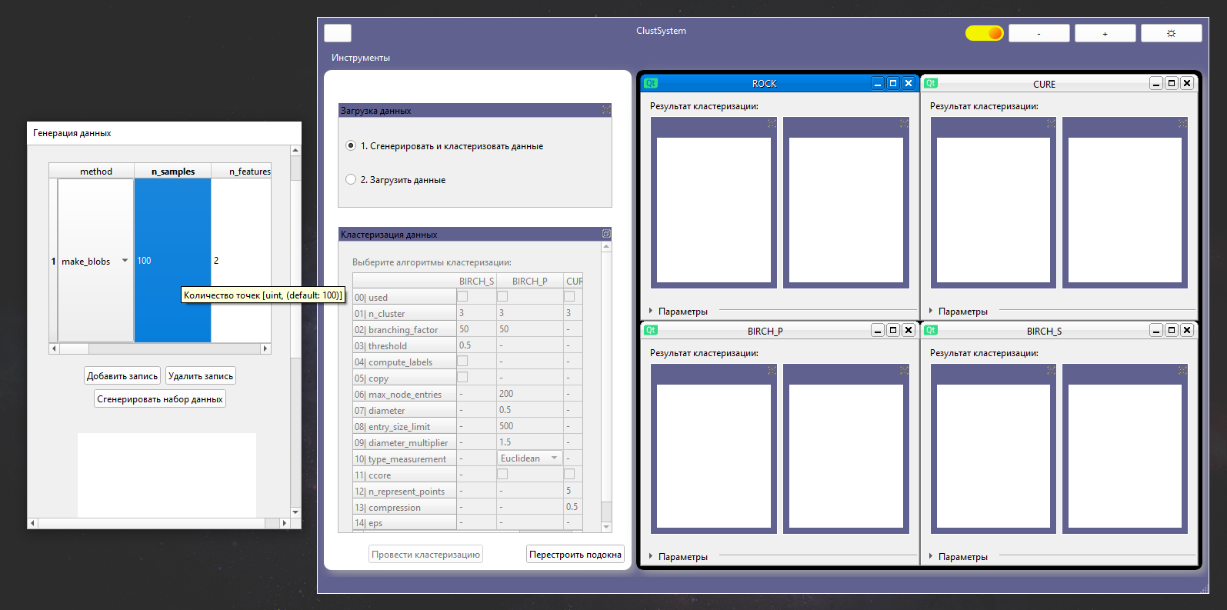


Рис. 16 – Демонстрация активации подсказки

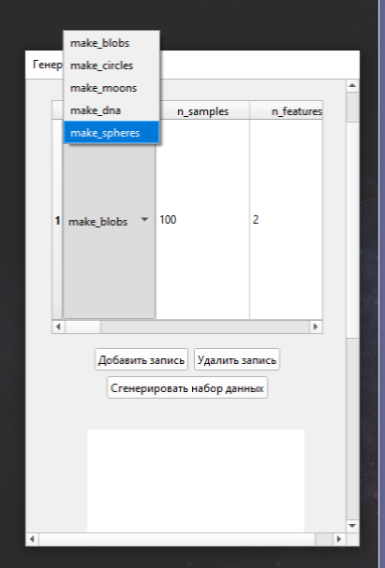


Рис. 17 – Демонстрация вариантов функций для генерации изображений

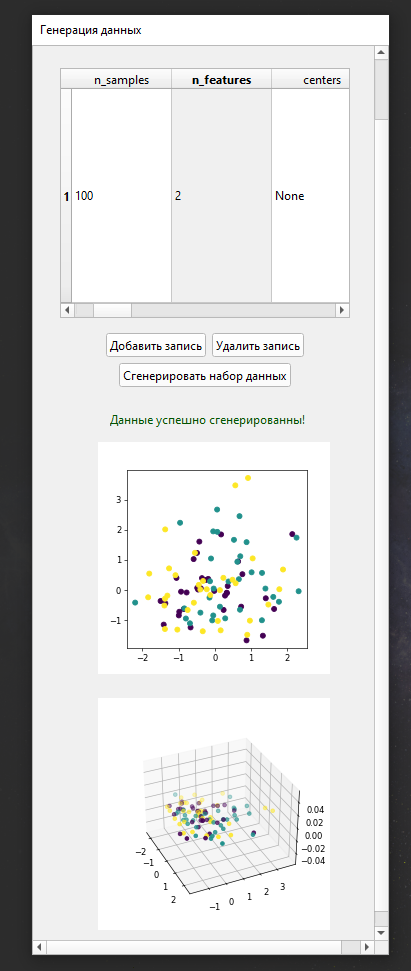


Рис. 18 – Результат генерации данных с использованием функции make\_blobs

## Загрузка изображений

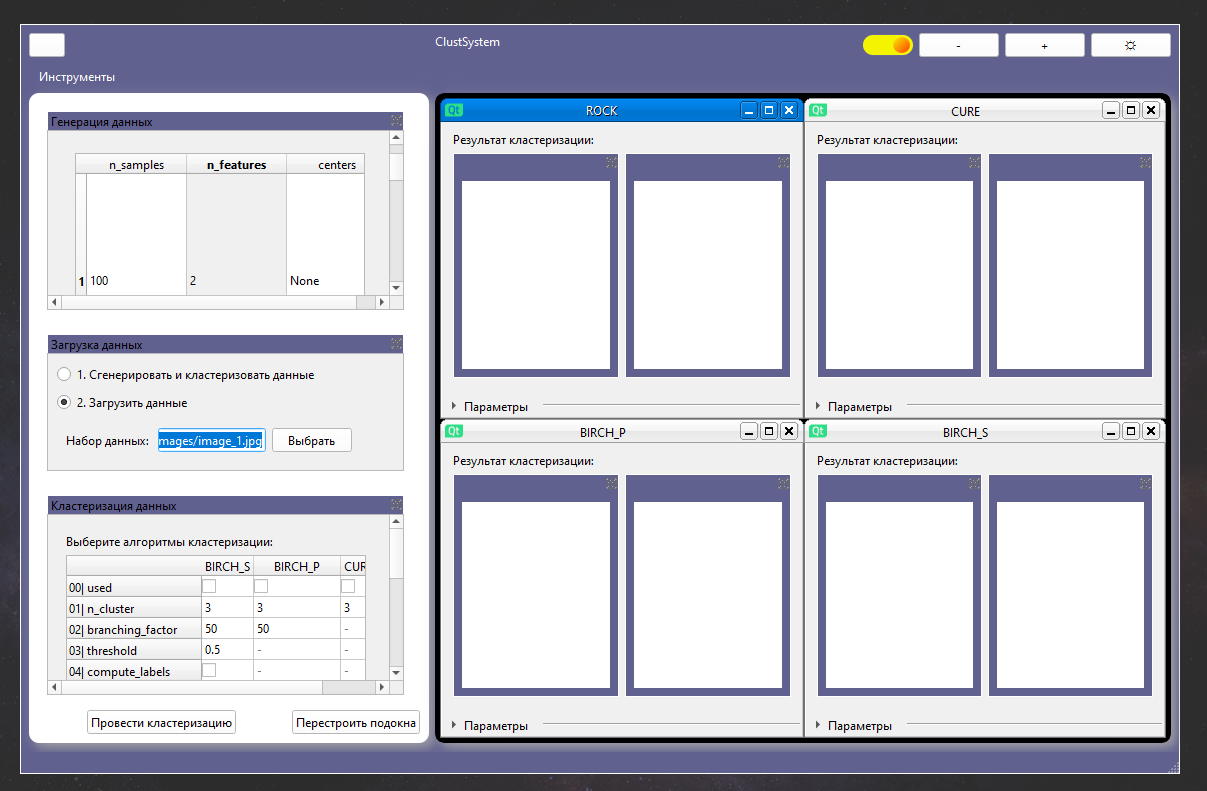


Рис. 19 – Демонстрация загрузки данных

## Кластеризация данных

### 4.4.1 Кластеризация точечных данных

После успешной генерации данных автоматически станет активной панель “Кластеризация данных”.

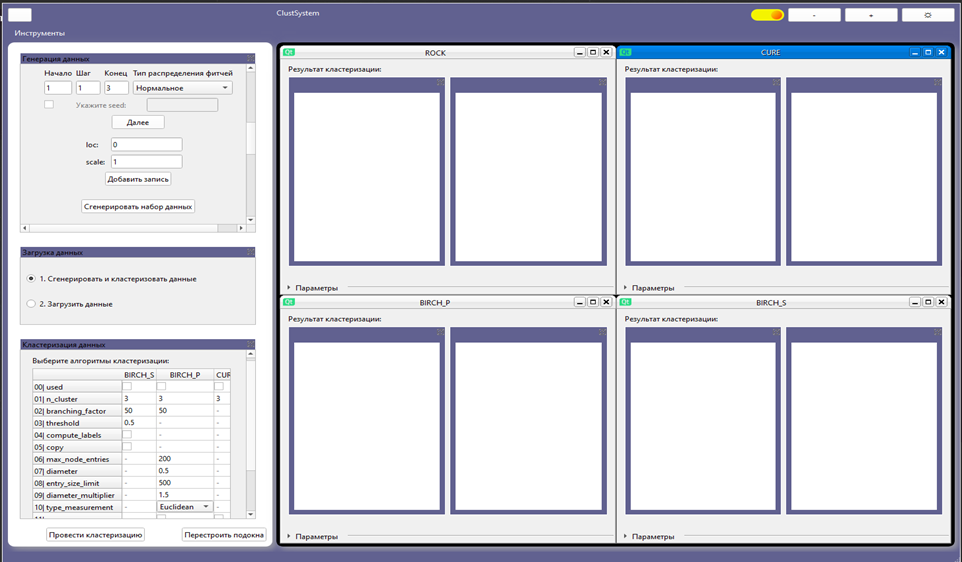


Рис. 20 – Демонстрация панели “Кластеризация данных”

Выберите алгоритмы, которые вы хотите использовать для кластеризации, установив соответствующую галочку в поля “used” таблицы ниже

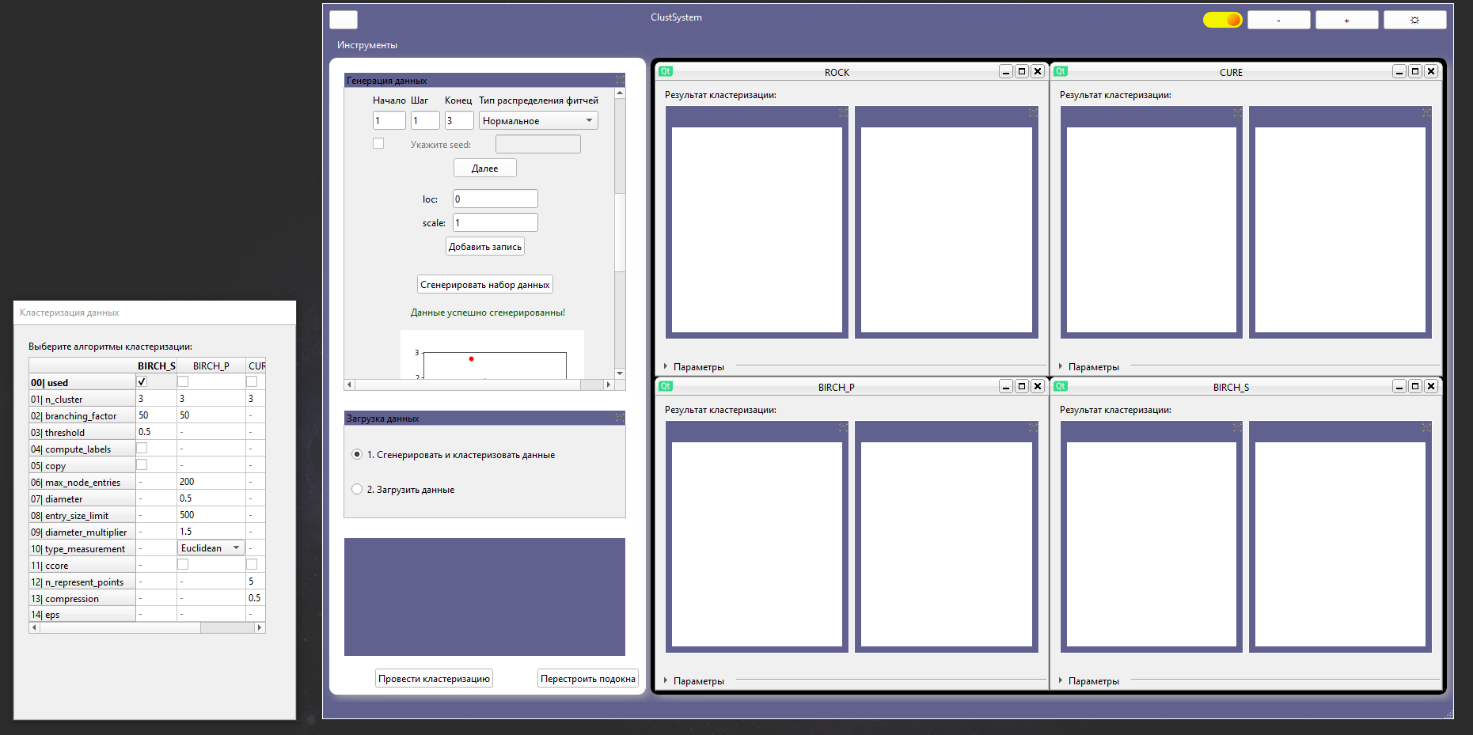


Рис. 21 – Демонстрация выбора алгоритмов кластеризации с проставлением отметки в таблице алгоритмов

Затем нажмите на кнопку “Провести кластеризацию”. После успешной кластеризации в левом нижнем углу окна вы получите статус об успешном этапе проведения кластеризации либо сообщение с указанием для какого метода при данных параметрах программы не удалось кластеризовать данные. После успешной генерации данных вы также можете их сохранить, нажав “Сохранить снимки” и “Сохранить таблицу”.

Результат кластеризации:



Рис. 22 - Демонстрация результата кластеризации

### Кластеризация пиксельных данных

Программа позволяет также кластеризовывать изображения

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис. 23 – Демонстрация задания параметров кластеризации изображений.

В данной реализации имеется 4 типа предварительной конвертации изображения в следующие форматы HSV, HLS, YUV. По умолчанию не происходит конвертация изображения в какой-либо формат, а используется метод rashape() для сегментирования изображения.

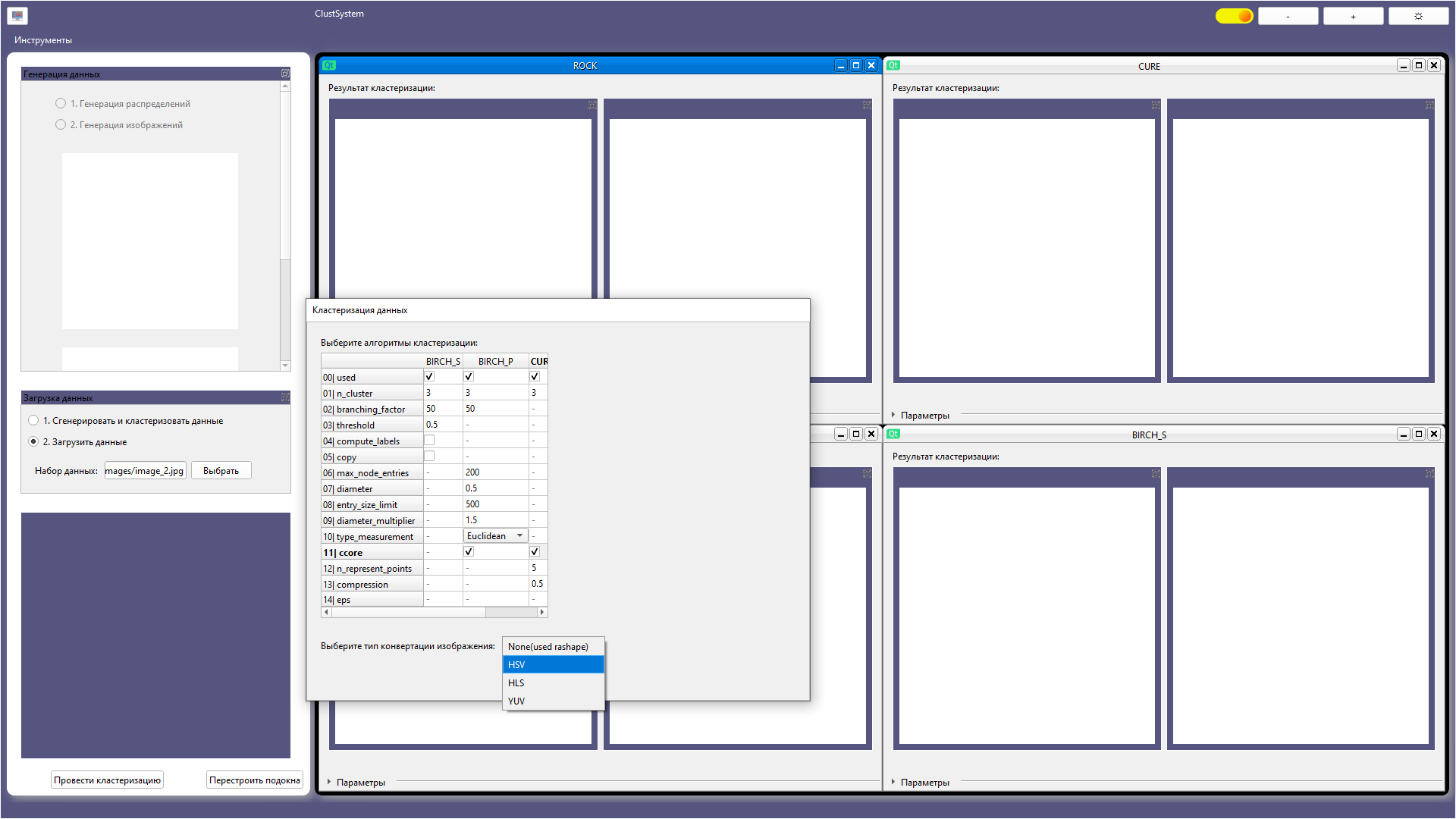


Рис. 24 – демонстрация типов конвертации формата пикселей перед кластеризацией.

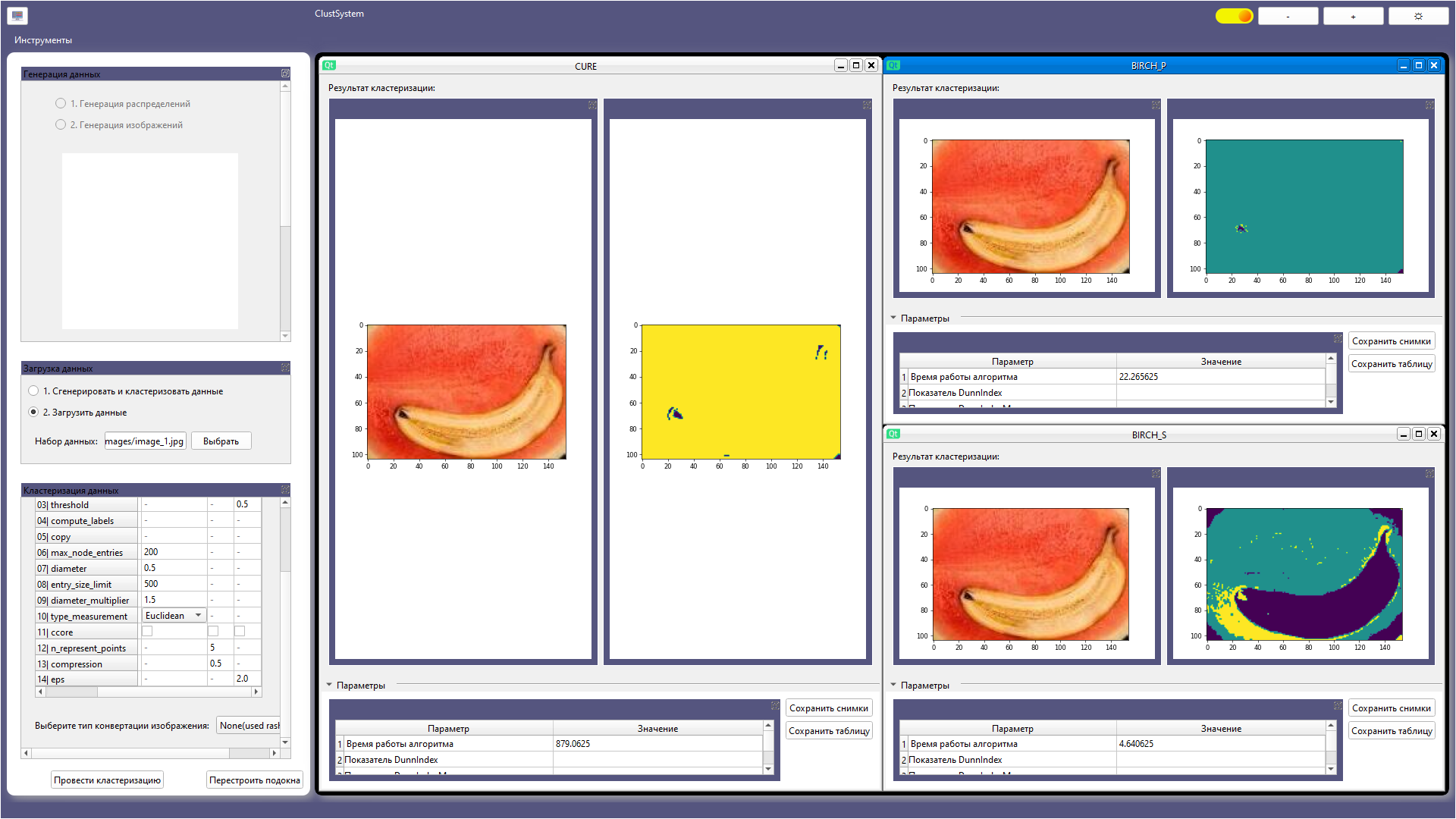


Рис. 25 – Демонстрация результатов кластеризации изображения

Стоит отметить, что кластеризация изображений довольно долго может происходить и напрямую зависит как от качества, размеров изображения, так и от характеристик вашего компьютера.

Метод **ROCK** если и кластеризует изображения, то очень небольшой размерности.

## 4.5 Настройки программы

|  |  |
| --- | --- |
| Первая тема | Вторая тема |
|  |  |

Рис. 26 – 27 – Демонстрация первой(светлой) и второй(темной) тем программы

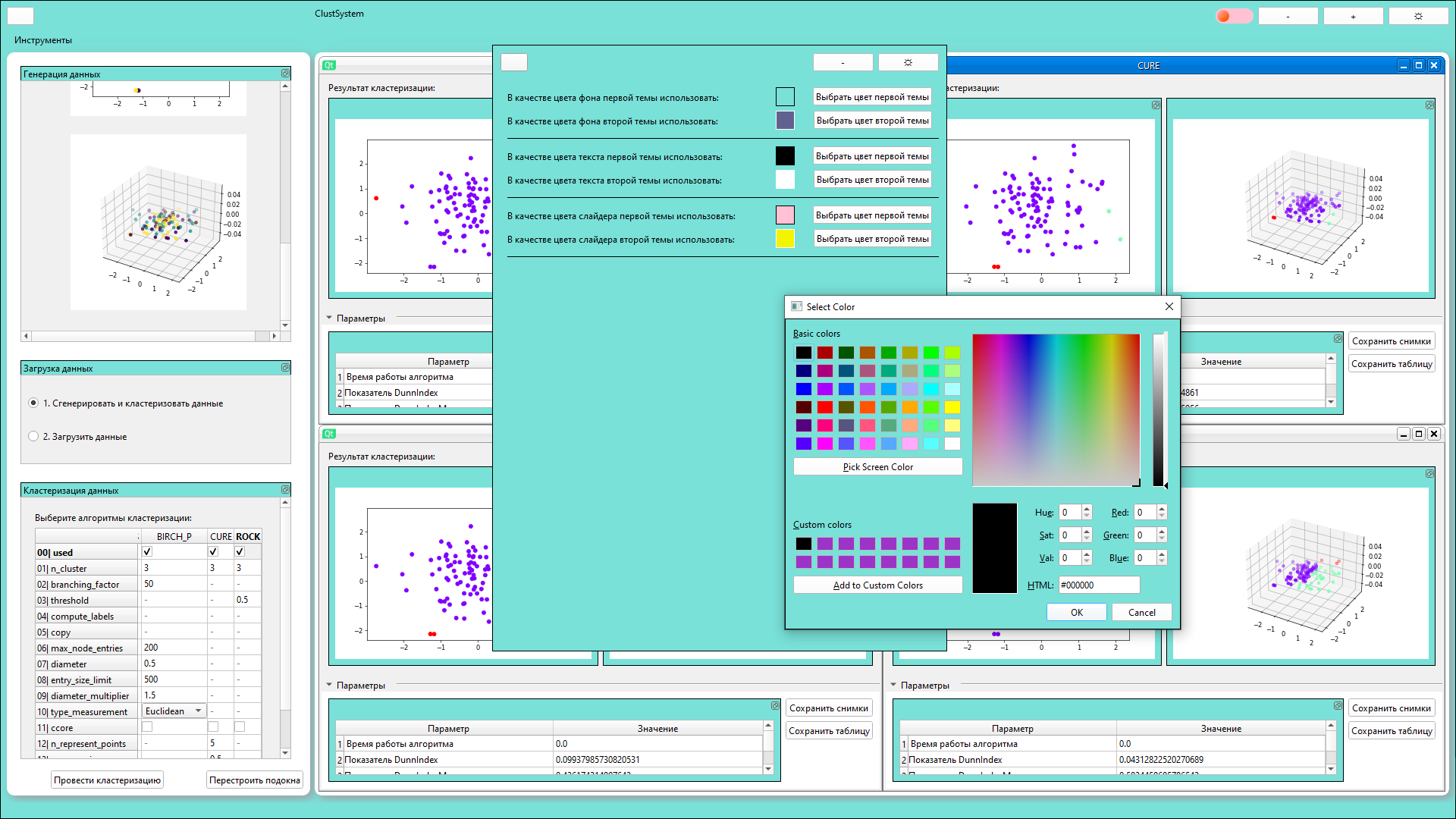


Рис.28 – Демонстрация окон задания параметров цвета для различных графических элементов

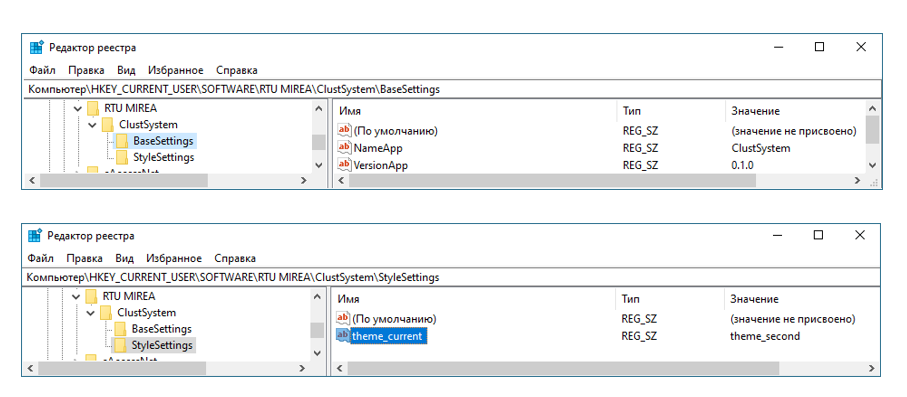


Рис.29 – Демонстрация хранения настроек приложения в реестре программы

## Анализ данных

После успешной кластеризации данных в таблицах для каждого алгоритма кластеризации формируются следующие показатели:

1. Время выполнения программы
2. DunneIndex – минимальное расстояние между кластерами
3. DunneIndexMean – минимальное среднее расстояние между кластерами

# Заключение

Разработанное ПО полностью удовлетворяет основным заявленным требованиям и может в дальнейшем использоваться в последующих разработках по тематике, связанной с кластеризацией.

# Приложение 1: Кода

Код поставляется в архиве и в виде ссылки на открытый репозиторий   
к данному ПО.

**Ссылка на GitHub:**

<https://github.com/MineevS/DMM_ClusteringSystem.git>