|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

***\_\_\_\_\_\_\_\_\_Кластеризация данных LiDAR\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_\_ ИУ5-61Б\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Саркисян С.З.\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ Канев А. И.\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*Москва, 2024 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени**

**Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_ИУ5\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение научно-исследовательской работы**

по теме \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кластеризация данных LiDAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_\_\_\_ИУ5-61Б\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Саркисян Стелла Зограбовна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Исследовательская\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_НИР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения НИР: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

***Техническое задание Исследовать методы кластеризации с помощью машинного обучения для решения задачи сегментации облака точек на отдельные деревья. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление научно-исследовательской работы:***

Расчетно-пояснительная записка на \_18\_ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «13» февраля 2024 г.

**Руководитель НИР**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ **Канев А.И.**\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ Саркисян С.З.**\_\_

(Подпись, дата) (И.О.Фамилия)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_heading=h.gjdgxs)

[1 Постановка задачи 5](#_heading=h.30j0zll)

[2 Описание исходных данных и используемых методов 8](#_heading=h.1fob9te)

[3 Выполнение работы 10](#_heading=h.3znysh7)

[ЗАКЛЮЧЕНИЕ 17](#_heading=h.2et92p0)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_heading=h.tyjcwt)

# **ВВЕДЕНИЕ**

LiDAR (от англ. Light Identification Detection and Ranging – обнаружение света и определение дальности) является одной из самых перспективных технологий сканирования с широкой областью применения. Чаще всего она используется для геодезических и картографических задач, позволяя сканировать и воссоздавать различные рельефы и объёмные модели, но также находит применение в качестве сенсоров для автономных транспортных средств, в климатологии и метеорологии, океанологии и ботанике, а также археологии.

Одно из самых широких распространений технология LiDAR получила в лесной отрасли, особенно в области таксации, где она используется для отслеживания изменений структуры лесных массивов и состояний отдельных деревьев. Лазерная таксация позволяет получать максимально точные и объективные данные, при этом затрачивая гораздо меньше ресурсов и не требуя слишком дорогостоящего оборудования. Сбор данных может производиться как с воздуха, так и с земли. Получившееся в результате сканирования местности облако точек необходимо классифицировать и сегментировать, после чего можно исследоваться отдельно взятые деревья, замерять их параметры, прослеживать состояние и др.

Целями данной научно-исследовательской работы является исследование методов кластеризации на основе технологий машинного обучения для сегментации облака точек на отдельные деревья, технологий машинного обучения для сегментации облака точек на отдельные деревья, а также конвертации облаков точек между собой. Задачами данной работы являются исследование существующих методов машинного обучения для кластеризации и основанная на них сегментация деревьев из облака точек с последующей оценкой качества этих методов и написании класса для конвертации файлов форматов LAS, LAZ, PCD, CSV.

# **1 Постановка задачи**

В результате наземной съёмки деревьев сканнером LiDAR были получены трёхмерные облака точек. Первая задача состоит в сегментации набора точек на отдельные деревья, то есть в выделении классов для каждого индивидуального дерева и присвоения каждой точке подходящего ей класса. Сложность сегментации заключается в том, что деревья расположены близко друг к другу, из-за чего их кроны переплетаются, и становится проблематично определить, к какому именно дереву стоит отнести точки.

Актуальность этой задачи растёт вместе с расширением области применения технологии LiDAR и активной разработкой новых методов обработки результатов сканирования. В отрасли лесного хозяйства задача сегментации деревьев играет огромную роль и позволяет решать множество разнообразных задач: от слежения за состоянием отдельно взятых деревьев до сканирования целого лесного массива для сбора информации о рельефе.

Леса являются ценным природным ресурсом, который поддерживает экосистему на благо человечества, в том числе сохранение биоразнообразия, участие в круговороте углерода и уменьшение парникового эффекта. Однако различные болезни, вредители и экстремальные погодные условия могут вызвать гниение деревьев, засуху и прочие заражения. Поэтому поддержание и мониторинг состояния лесов имеет огромное значение для сохранения целостности лесных экосистем. Традиционные методы лесопользования, такие как полевая съемка, требуют очень много времени и затрат из-за большой площади лесов. С развитием технологий дистанционного зондирования состояние лесов можно оценивать без проведения измерений на месте экономически эффективным способом с высоким качеством (Wong и др. [5]).

С точки зрения лесного хозяйства, способность быстро и надежно обнаруживать деревья в различных ситуациях остается важной задачей для разработки множества приложений. Например, на выбор дерева при прореживании заметно влияют переменные параметры дерева, такие как плотность стволов и крон, высота, диаметр и виды, что делает внешний вид дерева чрезвычайно разнообразным. На данный момент визуальные возможности человека-оператора являются единственными, способными оценить эти параметры и принять обоснованное решение во время лесохозяйственных работ. Однако будущая автоматизация этих работ с помощью методов машинного обучения и компьютерного зрения сможет снизить нагрузку на человека (Grondin и др. [4]).

Понимание роста деревьев является важным фактором для владельцев коммерческих садов. Коммерческие лесоводы полагаются на регулярные инвентаризации своего леса с точки зрения количества деревьев на данной территории, их высоты и других размеров, часто на больших площадях. Благодаря точному знанию местоположения деревьев, их индивидуальной структуры, а также количества и качества содержащейся в них древесины можно более эффективно использовать ресурсы во время лесозаготовительных работ и более оптимально планировать решения в цепочке поставок (Windrim, Bryson [4]).

Автоматическая идентификация и картографирование деревьев является давней целью в области дистанционного зондирования лесов, и в настоящее время существует особый интерес к поиску надежных решений для сегментации многовидовых насаждений со сложной структурой. Благодаря широкому внедрению систем LiDAR стало возможным собирать данные о трехмерной структуре лесных насаждений на площади в сотни квадратных километров за считанные часы, обычно в виде 3D-облаков точек, из которых можно сегментировать отдельные деревья (Williams и др. [5]).

Задачу сегментации трёхмерного облака точек можно решать очень многими способами. Одним из часто встречающихся является метод, основанный на графах, так как он позволяет быстро обрабатывать большие объёмы данных, а также менее чувствителен к шумам и выбросам, возникающим из-за данных низкого качества. Например, в работе [2] был разработан метод, который сначала объединяет точки в воксели, которые и являются вершинами графов, затем соседние воксели соединяются ребрами и строятся пути. После суммирования путей можно оценить, сколько раз каждая вершина попалась в путях, что пропорционально её участию в сети дерева и кратчайшему пути к ней. Далее с полученными результатами можно проводить различные сегментации. Эти методы определения местоположения и сегментации показали хорошие результаты, получив F1 score 0,774 и v-measure 0,915, в то время как классификация ствола показала плохие результаты c F1 score 0,490 на реальных данных, хотя продемонстрировала значительно малое время выполнения.

Другие методы используют относительную однородность формы деревьев в лесной среде для сегментации. В статье [4] описан метод, основанный на геометрии деревьев. На 2D изображении с видом сверху используется детектор объектов Faster-RCNN используется для определения границ деревьев путем вывода ограничивающих рамок в плоскости, которые проецируются в трехмерные кубоиды, и все 3D-точки внутри идентифицируются как принадлежащие отдельному дереву. Оценка метода с помощью метрик выдала следующие результаты в среднем: Precision 1.0, Recall 0,946 и F1 score 0,971, показав результаты в 2 раза лучше, чем метод кластеризации DBSCAN.

Точечные облака данных (Point Cloud Data) стали неотъемлемым элементом в области географической информационной системы, компьютерной графики, робототехники и многих других областях. Они представляют собой наборы точек в пространстве, каждая из которых имеет собственные координаты и, возможно, дополнительные атрибуты, такие как интенсивность цвета, высоту и другие параметры. Форматы хранения таких данных разнообразны, что требует использования специализированных инструментов и библиотек для их обработки.

Второй задачей данного проекта является разработка универсальной функциональности для работы с точечными облачными данными, которая позволит пользователям легко открывать, обрабатывать и сохранять данные в различных форматах без необходимости знать специфику каждого формата. Это включает в себя поддержку популярных форматов для хранения данных об облаках точек, таких как LAS, LAZ, CSV и PCD.

Для хранения и обмена точечными облачными данными применяются следующие основные форматы:

LAS (LiDAR ASCII) — это стандартный двоичный формат, разработанный Американским Институтом Геодезии и Картирования (ASCI) для хранения данных лазерного сканирования (LiDAR). Этот формат широко используется в геодезии, топографии, аэрофотосъемке и других областях, где требуется детальное три-dimensional (3D) представление объектов.

LAZ — это сжатый вариант формата LAS, разработанный для уменьшения размера файлов до 90% без потери информации. Формат LAZ был разработан в 2007 году как программное решение с открытым исходным кодом для сжатия файлов LAS.

Этот формат позволяет экономить место хранения и упрощает передачу данных через сеть. Он поддерживает все функции формата LAS, включая дополнительные атрибуты точек и при этом расширяет возможности формата LAS, добавляя новые поля и улучшенные методы сжатия.

CSV (Comma-Separated Values) — это простой текстовый формат, где данные разделены запятыми. Хотя он не является специально предназначенным для хранения трехмерных точек, может быть использован для хранения атрибутивных данных точек в таблице. Этот формат широко используется в статистике, программировании и других областях для обмена данными между различными системами и приложениями. Его особенностью можно выделить универсальность, так как этот формат легко читаемый и поддерживаемый большинством программных продуктов. Он позволяет легко импортировать и экспортировать данные в различные программы и базы данных.

PCD (Point Cloud Data) — это формат файла, разработанный для хранения точечных облачных данных в системах компьютерной визуализации и робототехники. Он используется в ROS (Robot Operating System) и других системах для обмена данными между различными компонентами роботов и системами. Формат поддерживает структурированные и неструктурированные точки, а также расширяется и адаптируется для новых типов данных и приложений.

Каждый из этих форматов имеет свои преимущества и недостатки, но общей проблемой является необходимость использования специализированных инструментов для их обработки. Это ограничивает доступность и удобство работы с точечными облачными данными для широкого круга пользователей.

Учитывая разнообразие форматов и специфику работы с точечными облачными данными, существует потребность в разработке универсальной функции, которая бы объединяла методы открытия и сохранения данных различных форматов. Такая функция должна обрабатывать поля с координатами, цветовыми данными, временной меткой и идентификатором точки, автоматически определять тип файла по его расширению и использовать соответствующие методы для его обработки, обеспечивая тем самым удобство и гибкость при работе с данными.

Разработка функциональности, которая учитывает эти поля, позволит создавать более богатые и информативные аналитические модели, улучшая тем самым способность к распознаванию объектов, оценке их свойств и визуализации результатов.

Для понимания основ и начала работы с данными использовала лекции из курса машинного обучения [7], а также за основу работы брала код из репозитория научно-исследовательской работы по машинному обучению [1].

# **2 Описание исходных данных и используемых методов**

В первую очередь исходными данными является трёхмерное облако точек, полученное в результате съёмки датчиком LiDAR. Эти данные представляют из себя массив координат по трём осям для 10000 точек. При отрисовке этого облака видно, что оно представляет из себя очертания трех близстоящих деревьев разнообразной формы (Рисунок 2.1). Задача заключается в том, чтобы разделить этот массив точек на 3 отдельных дерева и соответствующим образом их раскрасить в качестве визуализации.

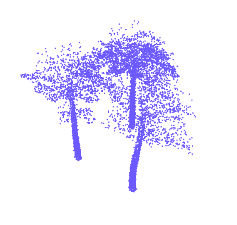


Рисунок 2.1 – Исходное облако точек

В качестве метода для кластеризации был выбран DBSCAN (Density-Based Spatial Clustering of Applications with Noise) – плотностный алгоритм пространственной кластеризации с присутствием шумов. DBSCAN выполняет итерацию по случайным точкам из набора данных, для каждой вычисляя окрестности её. Если эта окрестность содержит более определенного заданного количества точек, то эта точка включается в область. Каждая соседняя точка проходит через один и тот же процесс до тех пор, пока она больше не сможет расширять кластер. Если у рассматриваемой точки недостаточно соседей, она будет помечена как шум, что позволяет алгоритму быть устойчивым к выбросам. На вход алгоритма подается два обязательных параметра: минимальное число соседей точки для её включения в кластер и эпсилон (eps) – максимальное расстояние между двумя точками, то есть радиус поиска соседей вокруг точки.

Аналогичным методом является OPTICS (Ordering Points To Identify the Clustering Structure), который лучше работает на больших объёмах данных. Он динамически подбирает параметр радиуса эпсилон, что позволяет исключить этот параметр из подбора при применении алгоритма. Попытка применения данного алгоритма показала плохие результаты на исходном наборе данных, поэтому в работе он не используется.

Для оценки качества кластеризации будут использоваться две метрики. Silhouette score (оценка силуэта), рассчитываемая по формуле (1), показывает, насколько объект похож на свой кластер по сравнению с другими кластерами. Метрика принимает значение от -1 до 1, где значение близкое к 1 означает хорошую кластеризацию, 0 показывает перекрывающиеся кластеры, а -1 неправильную кластеризацию.

где a(xi,ck) – среднее расстояние от каждой точки xi до других объектов кластера ck (компактность);

b(xi,ck) – среднее расстояние от каждой точки xi до объектов из других кластеров кластера ck (отделимость).

В качестве второй метрики будет использоваться accuracy (аккуратность), выражающаяся как количество правильно определенных кластеров к их общему количеству. Однако автоматизировать данную метрику не получится, так как программа может определить только количество выделенных кластеров, но не их правильность, поэтому эта метрика будет оцениваться визуально.

Последующие данные представляют из себя массив с координатами по трём осям, интенсивностью сигнала, временем измерения, а также цветовыми данными для 1022205 точек. При отрисовке этого облака видно, что оно представляет из себя очертания двух близстоящих деревьев (Рисунок 2.1). Облако точек представлено в нескольких форматах: LAS (.las), LAZ (.laz), PCD (.pcd), CSV (.csv). Задача заключается в написании функции для конвертации файлов разных форматов между собой. Необходимо реализовать открытие и сохранение файлов с автоопределением данных и расширения.



Рисунок 2.1 –Облако точек

Для успешной обработки и анализа точечных облачных данных были выбраны и применены следующие Python библиотеки, каждая из которых предлагает уникальные возможности и инструменты для работы с данными.

NumPy является одной из наиболее популярных библиотек Python, предназначенных для научных вычислений. Она предоставляет поддержку для больших многомерных массивов и матриц, а также включает в себя множество математических функций для выполнения операций над этими массивами. NumPy позволяет эффективно работать с числовыми данными, что критически важно при обработке точечных облачных данных, где часто требуется выполнение сложных математических операций и трансформаций. В контексте работы с точечными облачными данными, NumPy используется для создания и манипулирования массивами координат точек, проведения математического анализа и фильтрации данных, а также для выполнения операций, связанных с геометрическими вычислениями.

Laspy — это специализированная библиотека Python для работы с LAS-файлами, стандартным форматом хранения данных лазерного сканирования (LiDAR)[3]. Laspy предоставляет удобный интерфейс для чтения и записи данных LAS, позволяя пользователям легко загружать, обрабатывать и сохранять точечные облака данных в этом формате. Эта библиотека используется для чтения и записи данных LAS, что позволяет работать с большими объемами точечных облачных данных, хранящихся в этом формате. С помощью Laspy можно извлекать и модифицировать атрибуты точек, такие как координаты, интенсивность сигнала, время съемки и другие метаданные.

Pandas — это еще одна важная библиотека Python, предназначенная для анализа данных. Она предоставляет структуры данных и инструменты для очистки, трансформации и анализа данных, что делает ее идеальным выбором для работы с метаданными и атрибутами точек в точечных облачных данных. В рамках работы с точечными облачными данными, Pandas используется для хранения и анализа метаданных или атрибутов точек.

pypcd — это библиотека Python, предназначенная для чтения и записи данных в формате PCD, который используется в системах компьютерной визуализации и робототехники. pypcd предоставляет функционал для работы с точечными облачными данными в этом формате, позволяя пользователям обмениваться данными между различными приложениями и платформами. С помощью pypcd можно загружать и сохранять точечные облака данных, а также извлекать и модифицировать атрибуты точек [6].

# **3 Выполнение работы**

Работа начинается с выполнения примера по заранее заготовленными функции сегментации на основе метода DBSCAN, функции визуализации с присвоением цветов каждому кластеру и выставленными параметрами eps = 0.5 и min\_samples = 240 (Рисунок 3.1). Значение метрики sillhouette score составило 0.41635. Как видно, было распознано 2 кластера, что неверно, так как два дерева слились в один кластер. Это означает, что заданные параметры не подходят для исходных данных.



Рисунок 3.1 – Результат сегментации при eps = 0,5 и min\_pts = 240

Чтобы найти оптимальное значение параметра eps, был применен метод локтя. Возьмем n =680 количество ближайших соседей. Для каждой точки из набора вычисляется среднее расстояние до ее n ближайших соседей. По оси X откладывается количество рассчитанных точек, а по оси Y посчитанное среднее расстояние. Значение точки сгиба по Y и будет оптимальным значением eps. В нашем случае получилось значение eps = 0.5035.

Для найденного eps необходимо подобрать оптимальное значение min\_pts. В результате перебора была получена таблица, часть которой представлена ниже (Таблица 3.3). При значениях min\_sample <220 получаются чересчур обобщённые кластеры (Рисунок 3.4). При значениях min\_samples> 500 получается сегментировать только стволы, а вся крона определяется в класс шумов (Рисунок 3.6). Самыми оптимальными получаются значения в диапазоне от 270 до 300, ведь они правильно сегментируют бо́льшую часть крон, но имеются погрешности в виде мелких кластеров на краях (Рисунок 3.5).

|  |  |  |  |
| --- | --- | --- | --- |
| Eps | Min\_samples | Количество кластеров | Silhouette score |
| 0.5035 | 100 | 1 | 0.1757 |
| 0.5035 | 120 | 1 | 0.1112 |
| 0.5035 | 150 | 1 | 0.1134 |
| 0.5035 | 180 | 1 | 0.1081 |
| 0.5035 | 200 | 1 | 0.1102 |
| 0.5035 | 220 | 2 | 0.2398 |
| 0.5035 | 250 | 2 | 0.2489 |
| 0.5035 | 270 | 3 | 0.3605 |
| 0.5035 | 280 | 3 | 0.361 |
| 0.5035 | 290 | 3 | 0.3587 |
| 0.5035 | 300 | 3 | 0.357 |
| 0.5035 | 320 | 4 | 0.3493 |
| 0.5035 | 350 | 4 | 0.3336 |
| 0.5035 | 380 | 5 | 0.3043 |
| 0.5035 | 400 | 4 | 0.3132 |
| 0.5035 | 450 | 4 | 0.3866 |
| 0.5035 | 500 | 4 | 0.323 |
| 0.5035 | 540 | 3 | 0.3444 |
| 0.5035 | 550 | 3 | 0.3442 |
| 0.5035 | 600 | 3 | 0.3419 |

Таблица 3.3 – Часть таблицы из подбора значений min\_pts

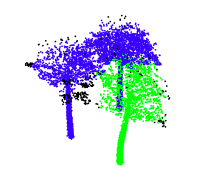


Рисунок 3.4 – Результат сегментации при eps = 0,5035 и min\_pts = 220

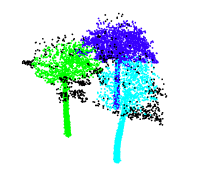


Рисунок 3.5 – Результат сегментации при eps = 0,5035 и min\_pts = 280

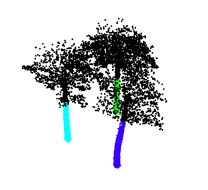


Рисунок 3.6 – Результат сегментации при eps = 0,5035 и min\_pts = 600

Было решено попробовать заменить метод кластеризации DBSCAN на OPTICS и проверить, какой он выдаст результат на исходных данных. При большом значении параметра min\_samples (Рисунок 3.7) метод хорошо справляется с сегментацией стволов, однако кроны попадают в класс шумов, так как в них плотность точек намного меньше. При малом значении min\_samples (Рисунок 3.8) OPTICS выделяет множество маленьких кластеров, что является наихудшим результатом сегментации. Отсюда можно сделать вывод, что метод OPTICS не подходит для решения поставленной задачи.

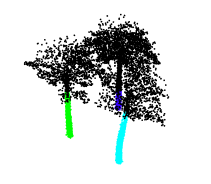


Рисунок 3.7 – Результат сегментации OPTICS при большом min\_samples



Рисунок 3.8 – Результат сегментации OPTICS при малом min\_samples



Рисунок 3.9 – Результат работы дополненного алгоритма сегментации

Самая успешная сегментация получилась в результате работы DBSCAN с найденным оптимальным eps = 0,5035 и min\_samples = 280 (Рисунок 3.5), так как там деревья сегментируются практически верно за исключением класса шумов. Можно избавиться от него. Для каждой точки из класса шумов ищется ближайший к ней кластер и назначается его класс. Введём в функцию параметр remove\_noise = True / False для включения и выключения обработки шумов соответственно. Результат работы получившейся функции в сравнении со старым алгоритмом представлен на рисунке 3.9.

Для решения второй задачи был разработан класс PC, представленный на рисунке 3.10., для эффективной работы с точечными облачными данными, который позволяет создавать экземпляры точечных облаков с возможностью хранения и анализа различных атрибутов точек. Этот класс включает в себя конструктор, который принимает атрибуты точек в качестве параметров, и два метода: open и save.

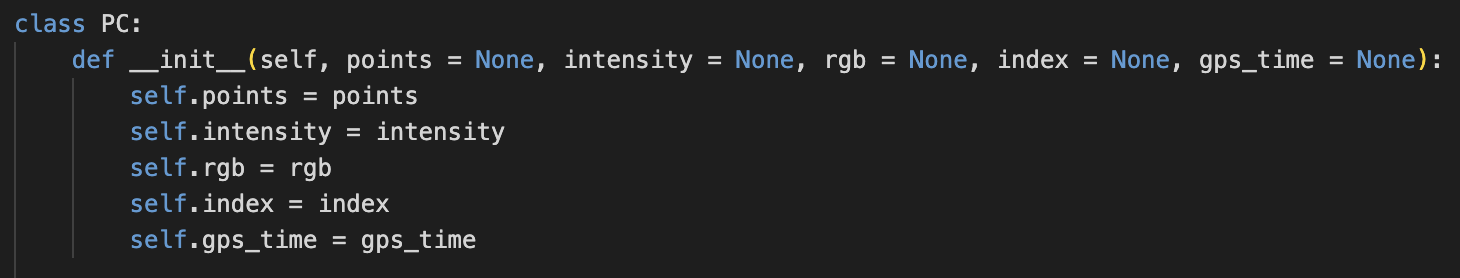


Рисунок 3.9 – Класс PC

Методы open и save предназначены для чтения и записи данных точечного облака. В зависимости от расширения файла каждый метод выбирает соответствующий формат для чтения или записи данных. В настоящее время реализована поддержка форматов .las, .laz, .csv и .pcd. Время, затраченное на открытие файла, измеряется и выводится при активированной опции verbose.

Работа над этой задачей начинается с изучения форматов LAS и LAZ. Это бинарные форматы, которые используются для хранения облаков точек. Двоичный формат LAS – стандарт хранения и обмена данных, получаемых практически любым лазерным сканером.

Файлы LAS содержат информацию о точках облака точек, включая их трёхмерные координаты. Также в формате могут быть записаны атрибуты для каждой точки. Эти данные позволяют визуализировать облака точек в различных представлениях. Наиболее часто используются две версии файлов LAS – 1.2 и 1.4. Файлы в формате LAS организованы на три основных компонента:

Header – заголовок файла, который содержит общую информацию о файле, такую как версия формата, количество точек, размеры пространственных единиц измерения, а также другие метаданные.

Variable Length Records (VLRs) – переменные длины записей, которые могут содержать дополнительную информацию о файле, такую как географические координаты области съемки, информацию о проекции, используемой в данных.

Point Records – записи точек, содержащие саму информацию о точках, включая их координаты (X, Y, Z), интенсивность, RGB-цвета, индекс и временную метку GPS. Количество точек определяется в заголовке файла.

Для работы с файлами LAS была использована библиотека laspy.

Далее был изучен формат PCD. Файлы в этом формате PCD используются для хранения данных точечных облаков, собранных с помощью LiDAR систем. Этот формат был разработан в рамках проекта Point Cloud Library (PCL) и представляет собой текстовый или бинарный файл, содержащий метаданные и данные точек.

Файлы PCD разделены на две основные части:

Заголовок (Header): Содержит метаданные о файле, включая версию формата, количество точек, размеры пространственных единиц измерения (например, метры), а также другие метаданные.

Данные точек (Data Points): Содержат сами данные точек. Количество точек определяется в заголовке файла.

Для работы с файлами PCD была использована библиотека pypcd.

В ходе работы над проектом по конвертации файлов точечных облаков, использование файлов CSV оказалось крайне полезным для отладки и проверки содержимого других файлов, особенно тех, которые были в форматах .las, .laz и .pcd. Файлы CSV благодаря своей простоте и универсальности стали незаменимым инструментом в процессе разработки и тестирования.

Перед тем как приступить к конвертации данных из одного формата в другой, были использованы CSV-файлы для быстрой проверки содержимого исходных файлов точек. Это позволило убедиться, что данные корректно считываются и содержат ожидаемую информацию, прежде чем приступать к более сложным операциям. После проведения конвертации данных из одного формата в другой, результаты сравнивались с исходными данными, сохраненными в CSV-файле. Это помогало быстро выявить любые расхождения или ошибки в процессе конвертации, позволяя своевременно внести необходимые корректировки.

В ходе разработки и тестирования системы конвертации файлов точечных облаков, была реализована функциональность для измерения времени, затраченного на открытие и сохранение данных в различных форматах, включая .las, .laz, .csv .pcd. Была создана таблица, которая отображает время, затраченное на каждую операцию, а также формат файла, с которым проводилась работа. Данные представлены в таблице 1.

|  |  |  |
| --- | --- | --- |
| формат | время на открытие файла | время на сохранение |
| las | 0.083 s | 0.053 s |
| laz | 0.168 s | 0.109 s |
| pcd | 0.003 s | 0.029 s |
| csv | 0.528 s | 4.066 s |

Табл. 1. время, затраченное на каждую операцию

Анализируя предоставленные данные о времени, затраченном на открытие и сохранение файлов в разных форматах, можно сделать несколько наблюдений и выводов относительно производительности работы с этими форматами.

* Производительность записи: Формат PCD демонстрирует наилучшую производительность при открытии и сохранении данных, что делает его оптимальным для операций чтения и записи.
* Общая производительность: Формат CSV, несмотря на свою простоту и широкую поддержку, показывает худшую производительность как при открытии, так и при сохранении файлов, что может быть нежелательно для приложений, требующих частой работы с данными.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения данной научно-исследовательской работы были выполнены следующие задачи:

1. Были изучены методы кластеризации DBSCAN и OPTICS, а также методы подбора оптимальных гиперпараметров для их работы. Вдобавок была использована метрика Silhouette score для оценки их работы.
2. Был разработан алгоритм, дополняющий работу метода DBSCAN, позволяющий указывать число максимальных кластеров и включать обработку класса шумов, для успешной сегментации облака точек на отдельные деревья.
3. Были изучены такие форматы файлов для точечных облаков, как LAS, LAZ, PCD и CSV. Каждый формат был изучен с точки зрения его применения, преимуществ и недостатков, а также способов работы с ними с помощью соответствующих инструментов. Также были изучены библиотеки для работы с точечными облаками, включая pypcd, laspy.
4. Была разработана система для конвертации данных между различными форматами файлов точечных облаков. Это включало в себя разработку алгоритмов для чтения, преобразования и записи данных в выбранных форматах. Также была реализована система мониторинга производительности, позволяющая отслеживать время выполнения операций открытия и сохранения файлов. Это позволило анализировать эффективность работы с различными форматами файлов и оптимизировать процессы обработки данных.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Data-Science. Репозиторий курса НИРС по машинному обучению: [Электронный ресурс]. Дата обновления: 14.03.2023. URL: [https://github.com/iu5git/Data-Science](https://github.com/iu5git/Data-Science/blob/main/lidar_segmentation/tree_segmentation.ipynb) (Дата обращения: 31.03.2024).
2. Jonathan Williamsa, Carola-Bibiane Schonlieb, Tom Swinfield, Juheon Lee, Xiaohao Cai, Lan Qie, David A. Coomes. Three-dimensional Segmentation of Trees Through a Flexible Multi-Class Graph Cut Algorithm (MCGC): [Электронный ресурс]. Дата обновления: 20.03.2019. URL: <https://arxiv.org/abs/1903.08481> (Дата обращения: 31.03.2024).
3. laspy: Python library for lidar LAS/LAZ IO: [Электронный ресурс]. Дата обновления: 31.10.2022. URL: <https://laspy.readthedocs.io/en/latest/index.html> (Дата обращения: 18.03.2024).
4. Lloyd Windrim, Mitch Bryson. Forest Tree Detection and Segmentation using High Resolution Airborne LiDAR: [Электронный ресурс]. Дата обновления: 30.10.2018. URL: <https://arxiv.org/abs/1810.12536> (Дата обращения: 31.03.2024).
5. Vincent Grondin, Jean-Michel Fortin, Francois Pomerleau, Philippe Giguere. Tree Detection and Diameter Estimation Based on Deep Learning: [Электронный ресурс]. Дата обновления: 31.10.2022. URL: https://arxiv.org/abs/2210.17424 (Дата обращения: 31.03.2024).
6. Докуметация по библиотеке pypcd: [Электронный ресурс]. Дата обновления: 01.10.2017. URL: <https://github.com/DanielPollithy/pypcd> (Дата обращения: 31.04.2024).
7. Репозиторий курса "Технологии машинного обучения", бакалавриат, 6 семестр: [Электронный ресурс]. Дата обновления: 26.03.2024. URL: [https://github.com/ugapanyuk](https://github.com/ugapanyuk/courses_current/wiki/COURSE_TMO_SPRING_2024/) (Дата обращения: 31.03.2024).