



Министерство образования и науки 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 (ИУ)

КАФЕДРА: Системы обработки информации и управления (ИУ5)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К МЕТОДОЛОГИИ МАШИННОГО ОБУЧЕНИЯ НА ТЕМУ:

**Создание набора данных облака точек деревьев на
основе LiDAR и исследование методов классификации
деревьев**

Студент ИУ5И-31М
(Группа)

(Подпись, дата)

Чэ Чжан
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

Ю.Е. Гапанюк
(И.О.Фамилия)

Консультант

(Подпись, дата)

(И.О.Фамилия)

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	2
2. ОБЛАКО ТОЧЕК	3
3. ОБРАБОТКА ДАННЫХ ОБЛАКА ТОЧЕК.....	4
3.1 Фильтрация облака точек	4
3.2 Извлечение ключевых точек	5
4. ПОДГОТОВКА НАБОРА ДАННЫХ ОБЛАКОВ ТОЧЕК	7
4.1 Состав набора данных.....	7
4.2 Удаление некачественных данных	9
4.3 Удаление шумов из облаков точек	13
5. КЛАССИФИКАЦИЯ НАШИХ НАБОРОВ ДАННЫХ ОБЛАКОВ ТОЧЕК 15	
5.1 Методы	15
5.2 Результаты.....	18
6. ЗАКЛЮЧЕНИЕ.....	22
СПИСОК ЛИТЕРАТУРЫ.....	23

1. ВВЕДЕНИЕ

Таксация леса – это учёт леса, его всесторонняя материальная оценка и составление технической характеристики (таксационной описания и плана) насаждений, определение их возраста, запаса (количества) древесины, прироста и объема отдельных деревьев и их частей.

Для эффективной таксации леса применяют LiDAR, который позволяет получить подробную информацию о размерах и формах деревьев. В основе технологии LiDAR лежит получение и обработка данных о различных удаленных объектах при помощи оптической системы. Локатор использует свойство света отражаться и рассеиваться в зависимости от степени прозрачности среды.[1] Прибор способен получать информацию о твердых предметах и о жидкости.

Одним из важных целей обработки облаков точек является их классификация.[2] Правильная классификация объектов на основе этих данных дает верное понимание, что находится на отсканированной 3D-картине окружающей среды. Использование методов машинного и глубокого обучения приводит к упрощению чтения таких данных, а также ускоряет их интерпретацию.

Для того чтобы обучить модель машинного обучения, способную классифицировать облака точек, необходимо подготовить качественный набор данных. Наша работа заключается в исследовании методов обработки данных облака точек и подготовке качественного набора данных облака точек.

2. ОБЛАКО ТОЧЕК

Облако точек - это, по сути, огромная коллекция крошечных отдельных точек, нанесенных в трехмерном пространстве. Оно состоит из множества точек, полученных с помощью лазерного 3D-сканера. Например, если вы сканируете здание, каждая виртуальная точка будет представлять собой реальную точку на стене, окне, лестнице, металлоконструкции или любой другой поверхности, с которой соприкасается лазерный луч.

Сканер автоматически комбинирует вертикальные и горизонтальные углы, создаваемые лазерным лучом, чтобы рассчитать положение каждой точки в 3D координатах X , Y , Z и получить набор измерений 3D координат. Затем эти данные могут быть преобразованы в цифровую 3D-модель, которая дает точное детальное изображение объекта.

Чем плотнее точки, тем детальнее изображение, что позволяет более четко и точно определить мелкие особенности и детали текстуры.

Данные облака точек отличаются от обычных изображений тем, что данные не имеют регулярного расположения. Он имеет две важные характеристики: (1) инвариантность перестановки[7] (2) инвариантность вращения[8].

Инвариантность перестановки означает, что порядок, в котором хранятся облака точек, не зависит от общих характеристик облака точек. Нарушение порядка следования каждого ряда не окажет никакого влияния на характеристики облака точек. Учитывая это, можно найти эту особенность, удалив; определенное количество рядов данных облака точек по порядку.

Вращательная инвариантность означает, что характер облака точек не зависит от точки зрения, и что произвольный поворот облака точек эквивалентен просмотру одного и того же облака точек под разными углами и в разных направлениях.

После завершения обработки данных и получения облака точек их можно использовать различными способами, включая создание топографических карт местности или цифровых моделей рельефа для незастроенных земель.[9-10] Однако их также можно непосредственно визуализировать, то есть

преобразовать в полигонные или треугольные сетчатые модели или модели автоматизированного проектирования (CAD), которые можно использовать для классификации объектов или создания трехмерных моделей. [11]

3. ОБРАБОТКА ДАННЫХ ОБЛАКА ТОЧЕК

3.1 Фильтрация облака точек

Фильтрация облака точек - это удаление шума из данных облака точек. Фильтрация облаков точек похожа на фильтрацию при обработке сигналов. Необработанные данные облака точек часто содержат большое количество хэш-точек и изолированных точек. При получении данных облака точек неизбежно возникает некоторый шум, обусловленный влиянием точности оборудования и факторов окружающей среды, изменениями свойств поверхности измеряемого объекта, а также влиянием операций совмещения и выравнивания данных. Как первый шаг в процессе предварительной обработки, фильтрация оказывает значительное влияние на последующий анализ.[14] Только когда шум, выбросы и дыры разумно обрабатываются в процессе предварительной обработки, извлечение признаков может быть выполнено лучше.

Фильтр PassThrough - это простейший тип фильтра, который отфильтровывает точки, чьи значения в данном измерении не попадают в заданную область значений. Прямоточный фильтр реализуется следующим образом: во-первых, задается размерность и область значений под этой размерностью, во-вторых, каждая точка в облаке точек обходится для определения того, находится ли точка в области значений в заданной размерности, и точки, которые не находятся в области значений, удаляются, и, наконец, обход завершается, и оставшиеся точки образуют отфильтрованное облако точек. Прямоточный фильтр прост и эффективен и подходит для таких операций, как удаление фона.

Статистический фильтр[15-17] предназначен для удаления разрозненных шумов. Во время сбора данных облака точек возникает некоторый шум, обусловленный шумом измерений, который неравномерно распределен в пространстве облака точек. Эти шумовые точки могут привести к неправильным

результатам вычислений при оценке локальных особенностей в облаке точек. Основная идея статистического фильтра заключается в предположении, что среднее расстояние между всеми точками в облаке точек и их k ближайшими соседями удовлетворяет гауссовскому распределению, затем на основе среднего значения и дисперсии может быть определен порог расстояния, и если среднее расстояние между точкой и ее k ближайшими точками больше этого порога, точка определяется как выброс и удаляется.

Удаление выбросов по радиусу[18-20] - это фильтрация по количеству точек в радиусе пространственной точки, имя соответствующего класса в библиотеке PCL - RadiusOutlierRemoval. В данных облака точек задано, что каждая точка имеет не менее k окружающих ее точек в определенном радиусе d . Если количество точек в радиусе d точки меньше k , она будет удалена.

Фильтр VoxelGrid[21] использует метод воксельной сетки для реализации понижающей дискретизации, уменьшая количество точек при сохранении характеристик формы облака точек. VoxelGrid создает трехмерную воксельную сетку из входных данных облака точек, а затем использует центр тяжести всех точек внутри каждого вокселя для аппроксимации других точек в этом вокселе. Таким образом, все точки в пределах вокселя в итоге будут представлены одной точкой центра тяжести. Этот метод медленнее, чем аппроксимация центра вокселя, но обеспечивает более точное представление поверхности, соответствующей отобраным точкам.

3.2 Извлечение ключевых точек

Набор стабильных, дифференцированных точек на облаке ключевых точек. Количество ключевых точек намного меньше объема данных в исходном облаке точек или изображении, и часто используется для формирования компактного представления исходных данных без потери репрезентативности и описательности. Ключевые точки могут ускорить обработку последующего распознавания, отслеживания и других операций.

Ключевые точки NARF (Normal Aligned Radial Feature) [22] предлагаются для идентификации объектов на изображениях глубины, и процесс извлечения

ключевых точек NARF имеет следующие требования.

- 1) В процессе извлечения учитывается информация о краях и изменениях на поверхности объекта.
 - 2) Ключевые точки могут быть неоднократно обнаружены в различных ракурсах.
 - 3) расположение ключевых точек имеет достаточную площадь опоры для вычисления дескрипторов и однозначной оценки нормальных векторов
- Соответствующие этапы обнаружения показаны ниже.

- 1) Итерация по каждой точке изображения глубины и выполнение обнаружения краев путем поиска мест с изменением глубины в непосредственной близости.
- 2) итерация по каждой точке изображения глубины, принятие решения о коэффициенте для измерения изменения поверхности на основе изменения поверхности в непосредственной близости и основного направления изменения.
- 3) Вычислить точку интереса на основе главного направления, найденного на этапе(2), характеризуя, насколько это направление отличается от других направлений и насколько стабильна поверхность в этой точке.
- 4) Выполните сглаживающий фильтр для интересующего значения.
- 5) Выполните сжатие без максимума, чтобы найти конечную ключевую точку, которая является ключевой точкой NARF.

Масштабно-инвариантное преобразование признаков (SIFT)[23] - это алгоритм для компьютерного зрения. Он используется для обнаружения и характеристики локальных особенностей изображения путем нахождения крайних точек в пространственном масштабе и извлечения их инвариантов положения, масштаба и поворота. В настоящее время он также используется в области трехмерных облаков точек для обнаружения ключевых точек.

Алгоритм обнаружения ключевых точек Harris[24], также известный как алгоритм обнаружения ключевых точек Plessey, был предложен Крисом Харрисом и Майком Стивенсом в 1988 году и является одним из ранних

классических алгоритмов обнаружения ключевых точек. Уравнение 1 представляет собой матрицу Харриса. Обнаружение ключевых точек Харриса определяет, является ли точка изображения ключевой точкой, путем вычисления матрицы Харриса этой точки и соответствующего собственного значения матрицы. Если оба собственных значения матрицы Харриса велики, то точка является ключевой. Уравнение 2, как правило, можно использовать вместо вычисления собственных значений матрицы Харриса.

$$A = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix}^T = \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \quad (1)$$

$$m_k = \det(A) - k \cdot \text{tr}^2(A) \quad (2)$$

Здесь A обозначает Harris-матрицу точек. I_x и I_y представляют собой градиент точки пикселя в направлении x , y . Точка является ключевой, когда $m_k > 0$. Определение ключевых точек Harris в облаках точек заимствует идею определения ключевых точек Harris для двумерных изображений, хотя ключевые точки Harris для облаков точек используют информацию из векторов нормалей поверхности облака точек, а не градиенты изображения, используемые для определения ключевых точек Harris в двумерных изображениях.

4. ПОДГОТОВКА НАБОРА ДАННЫХ ОБЛАКОВ ТОЧЕК

4.1 Состав набора данных

Наш набор данных содержит облака точек 624 деревьев. Эти облака точек были извлечены из всего облака точек леса с помощью программы CloudCompare. Это извлечение основано на встроенном алгоритме CloudCompare и поэтому является грубыми данными, с выбросами и шумом. Для того чтобы повысить точность последующего анализа, нам необходимо предварительно обработать эти данные.

Для анализа были отобраны только пять деревьев с наибольшим количеством образцов, так как в других категориях было слишком мало образцов, чтобы использовать их для обучения нейронной сети. Пять видов деревьев, которые мы выбрали, включают ель, сосну, березу, рябину и дуб.

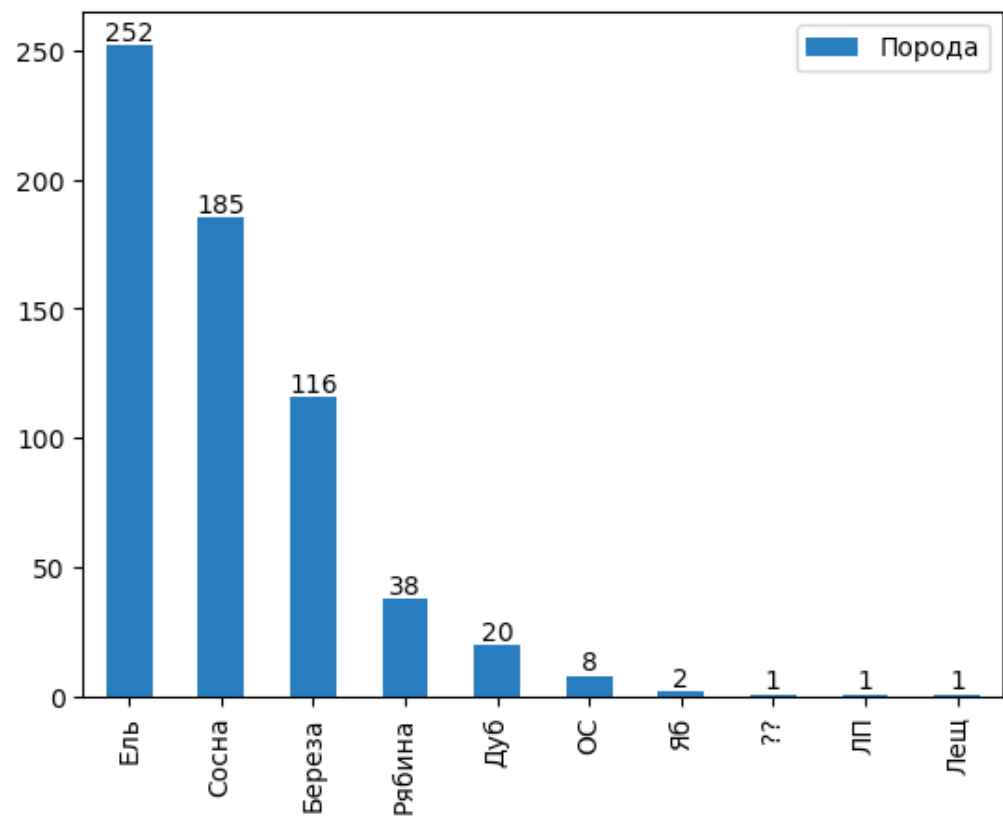


Рисунок 1. Состав пород деревьев в исходных данных

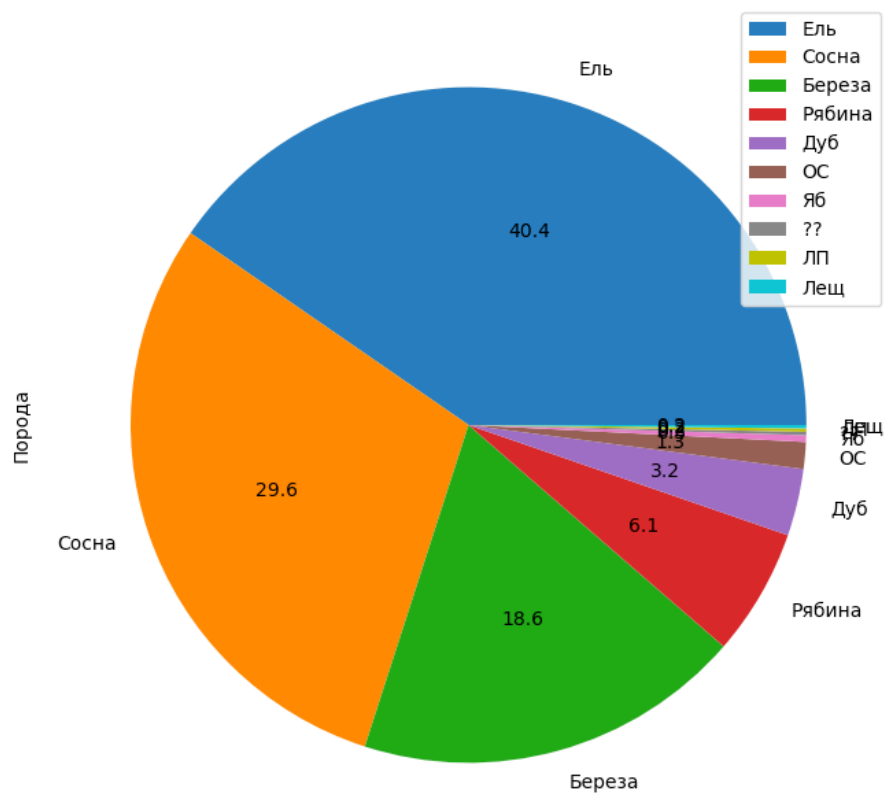


Рисунок 2. Процентное соотношение пород деревьев в исходных данных

Размеры выборок пяти деревьев, которые мы выбрали, были неравномерными, и числа между ними значительно различались, но мы могли уменьшить проблемы, вызванные различиями в размерах выборок, путем увеличения данных и корректировки весов функции потерь.

4.2 Удаление некачественных данных

Сначала мы удалили некоторые непригодные образцы путем ручного отбора. Эти образцы имели неполные формы и разрывы из-за маскировки и ошибок в процессе сканирования.

Неполное облако точек - это облако точек с настолько низким разрешением, что трудно различить видовые характеристики деревьев.

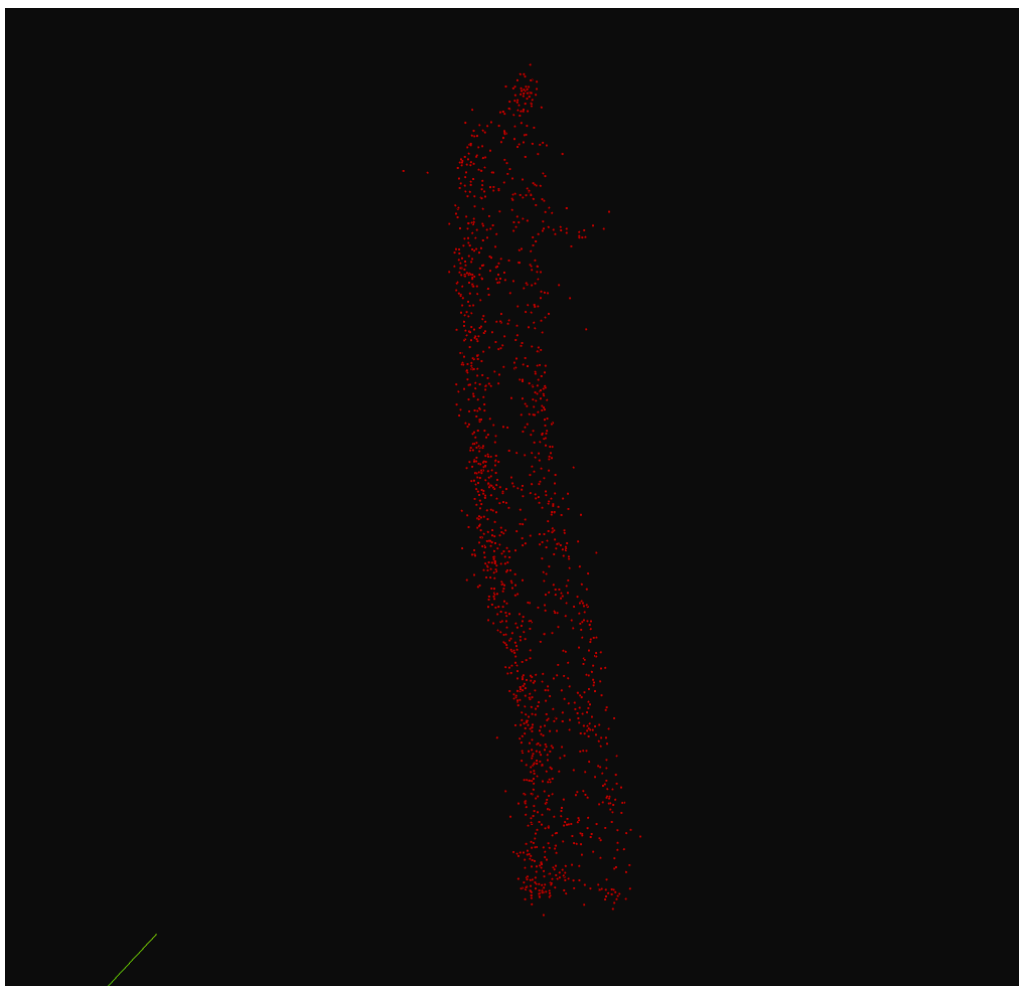


Рисунок 3. Неполное облако точек дерева

Облако точек на рисунке 3 отображает ствол дерева, но часть кроны дерева отсутствует. Поэтому мы отбросили такие облака точек.

Прерывистое облако точек - это облако, в котором одна часть облака точек явно отсутствует. Из-за отсутствия, дерево не может быть охарактеризовано в отсутствующей части, и мы иногда испытываем трудности в определении того, являются ли разделенные части облака точек одним и тем же деревом.

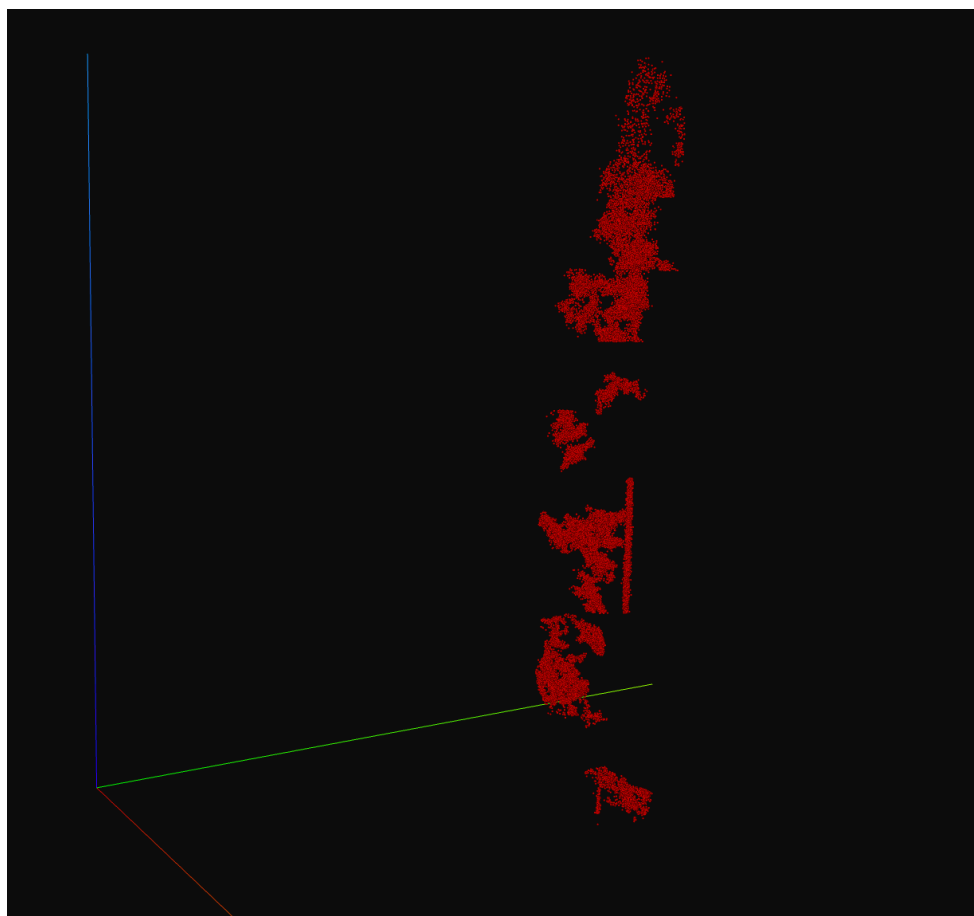


Рисунок 4. Прерывистое облако точек дерева

Облако точек на рис. 4 имеет два явных пропусков у полога, и трудно интерполировать данные в пропущенных местах. Поэтому мы удалили эти прерывистые облака точек.

Облако точек без разделения означает, что образец содержит облака точек более чем одного дерева. Рисунок 5 иллюстрирует эту ситуацию. Поскольку облако точек в этом случае трудно разделить, мы исключили эти данные.



Рисунок 5. Облако точек дерева без разделения

После исключения данных низкого качества состав набора данных изменился. Набор данных после удаления данных имеет общее количество 564 образцов.

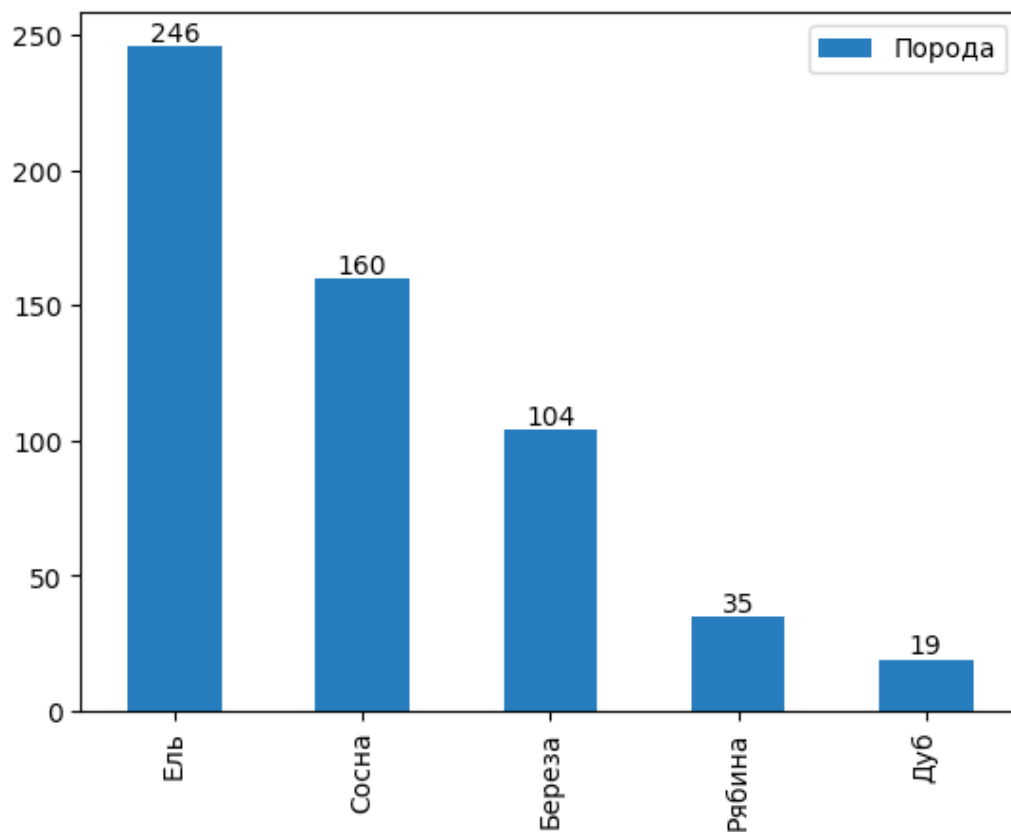


Рисунок 6. Состав пород деревьев в исходных данных после удаления некачественных данных

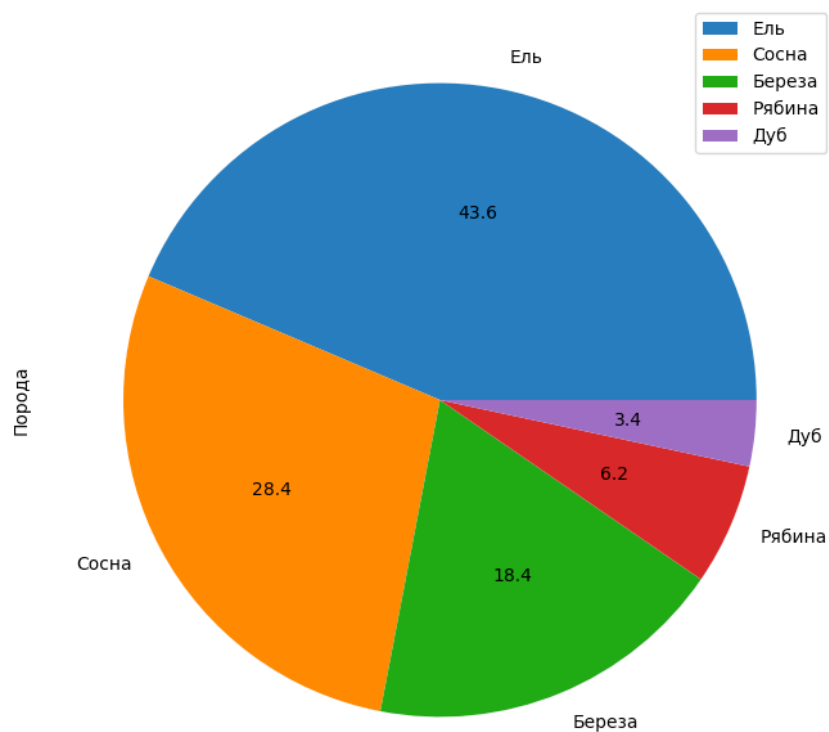


Рисунок 7. Процентное соотношение пород деревьев в исходных данных некачественных данных

4.3 Удаление шумов из облаков точек

Статистические фильтры используются для удаления шума из облака точек. Основная идея статистического фильтра заключается в предположении, что среднее расстояние между всеми точками в облаке точек и их k ближайшими соседями удовлетворяет гауссовскому распределению, затем на основе среднего значения и дисперсии может быть определен порог расстояния, и если среднее расстояние между точкой и ее k ближайшими точками больше этого порога, точка определяется как выброс и удаляется. На рисунке 8 показан результат удаления шума, где красные точки - шум, а синие точки - сохраненное облако точек.

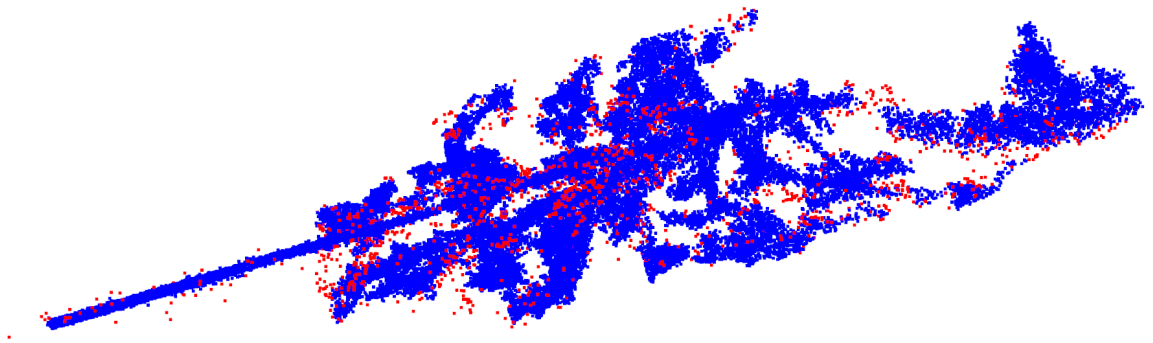


Рисунок 8. Результат после статистической фильтрации

Удаление выбросов по радиусу - это фильтрация по количеству точек в радиусе пространственной точки. Эффект этого метода показан на рисунке 9, где зеленые точки - это сохраненные точки, а фиолетовые - устраненный шум. Этот метод достигает результатов, аналогичных статистическому фильтру.

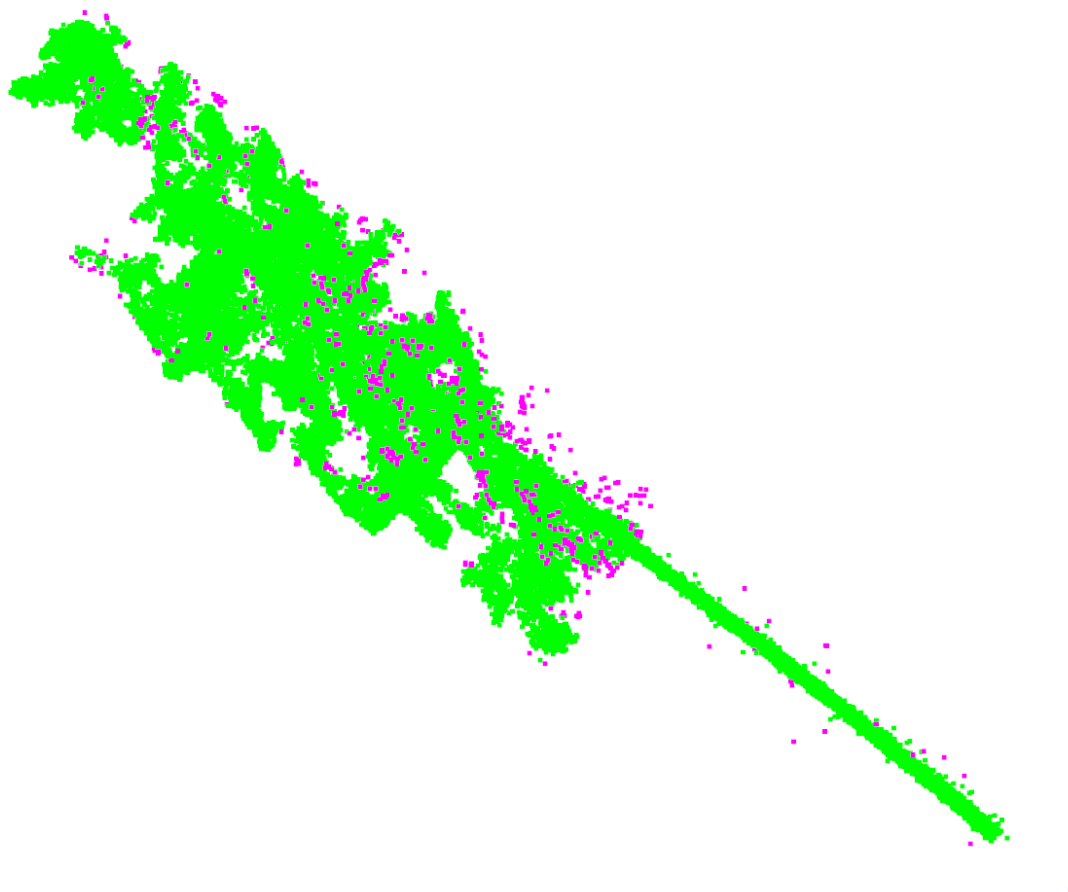


Рисунок 9. Результат после удаления выбросов по радиусу

VoxelGrid также используется для шумоподавления. VoxelGrid создает трехмерную воксельную сетку из входных данных облака точек, а затем использует центр тяжести всех точек внутри каждого вокселя для аппроксимации других точек в этом вокселе.

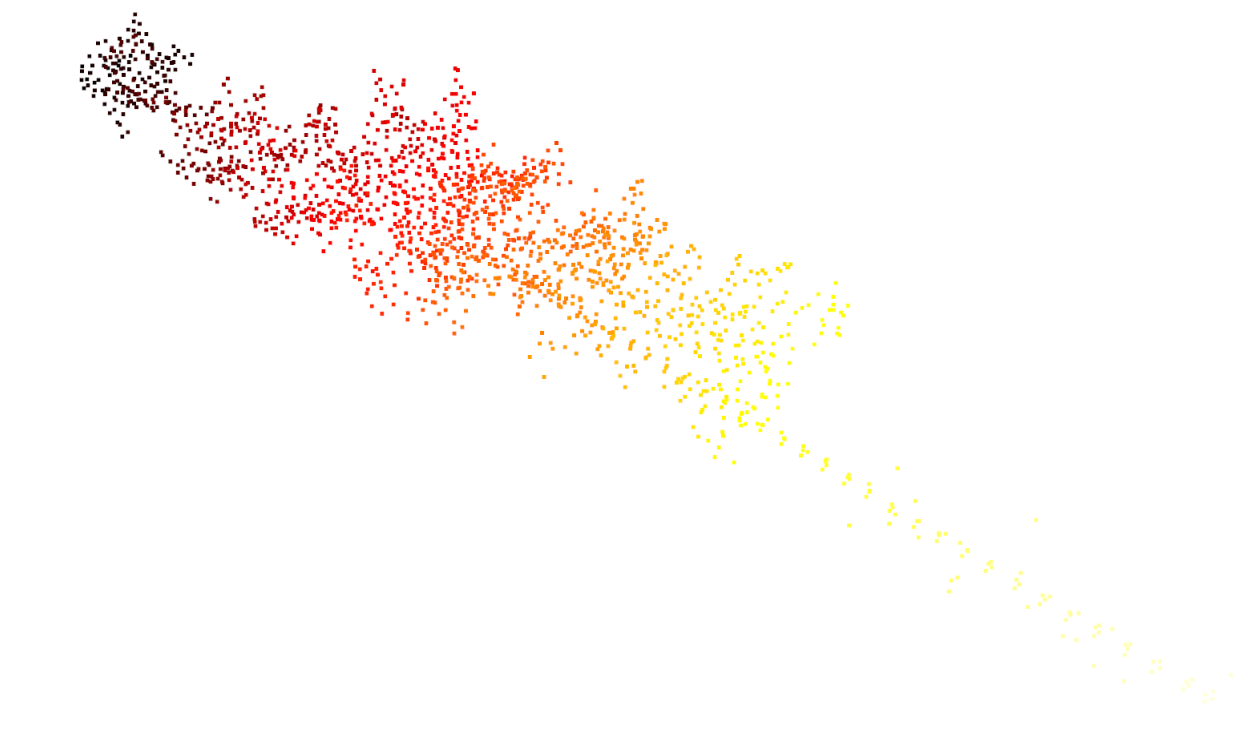


Рисунок 10. Результат после фильтрации VoxelGrid

Результаты удаления шума с помощью VoxelGrid показаны на рисунке 10, где этим методом сохраняется меньшее количество точек.

5. КЛАССИФИКАЦИЯ НАШИХ НАБОРОВ ДАННЫХ ОБЛАКОВ ТОЧЕК

5.1 Методы

В 2017 году была разработана модель PointNet [7,25] - нейронная сеть, которая напрямую обучается на беспорядочных облаках точек. Архитектура представлена на рисунке 11.

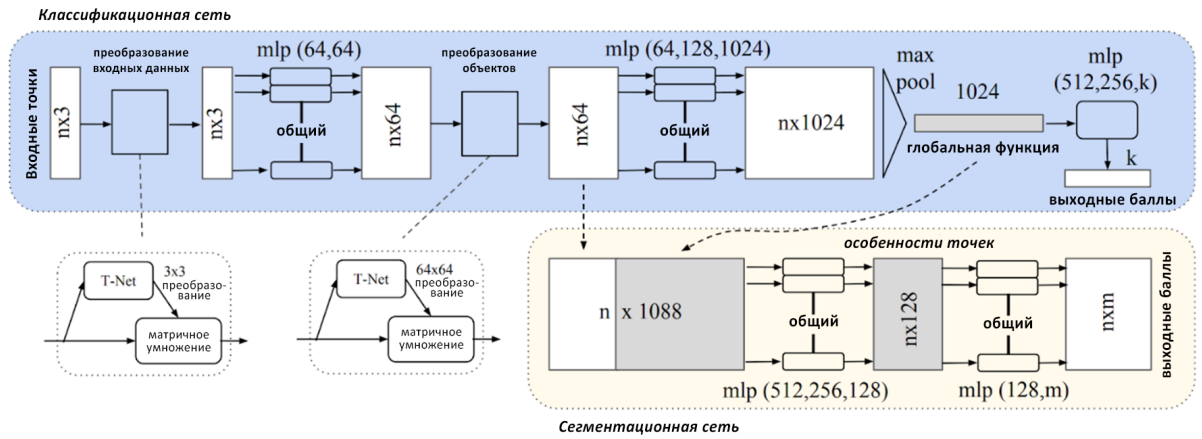


Рисунок 11. Архитектура PointNet

В сети есть 3 ключевых модуля: слой Max Pooling, принимающий n векторов входных данных и выводящий новый вектор, две сети трансформации с многослойным персептроном (MLP) с размерами $(64, 64)$ и $(64, 128, 1024)$ и две прогнозные сети с обученной матрицей преобразования T-Net.

PointNet изучает характеристики индивидуальной точки с помощью MLP и объединяет все их характеристики с помощью симметричной функции для выполнения классификации объектов и их сегментации на части.

Однако для сложных и детализированных данных этого оказалось недостаточно, поэтому позже была представлена иерархическая нейронная сеть PointNet++ [26-29]. Архитектура, изображенная на рисунке 12, состоит из нескольких компонентов, которые собирают локальную информацию и передают ее на следующий шаг.

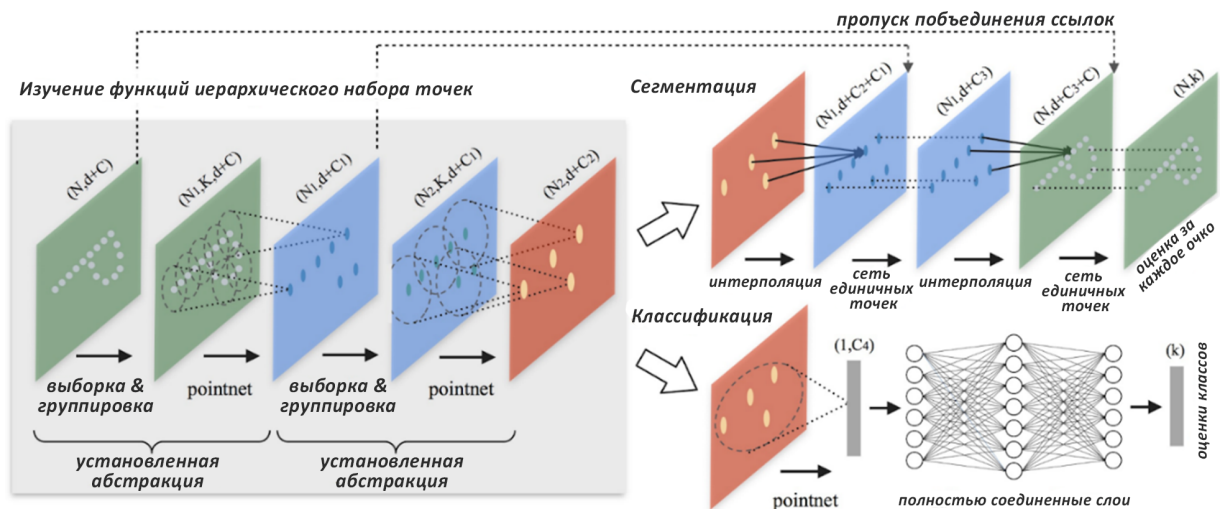


Рисунок 12. Архитектура PointNet++

Изначально точки в облаке точек собирают в определенное количество кластеров, которые в последствии группируются в единую точку, которая несет в дополнение к исходной информации дополнительную. В зависимости от задачи классификации или сегментации обработка меняется на противоположную и восстанавливает исходную структуру.

Но на обучение нейронных сетей нередко тратятся большие ресурсы, поэтому для решения этой проблемы был представлен алгоритм машинного обучения, основанный на стратегии последовательной аппроксимации подпространств, PointHop [30] с итеративным пошаговым процессом обмена информацией. Входное облако точек, которое содержит N точек с 3 трехмерными координатами, подается каскадно в несколько блоков алгоритма, представленного на рисунке 13. Выходные данные агрегируются с помощью M различных схем для получения функций, которые в последствии каскадируются для получения результата классификации.

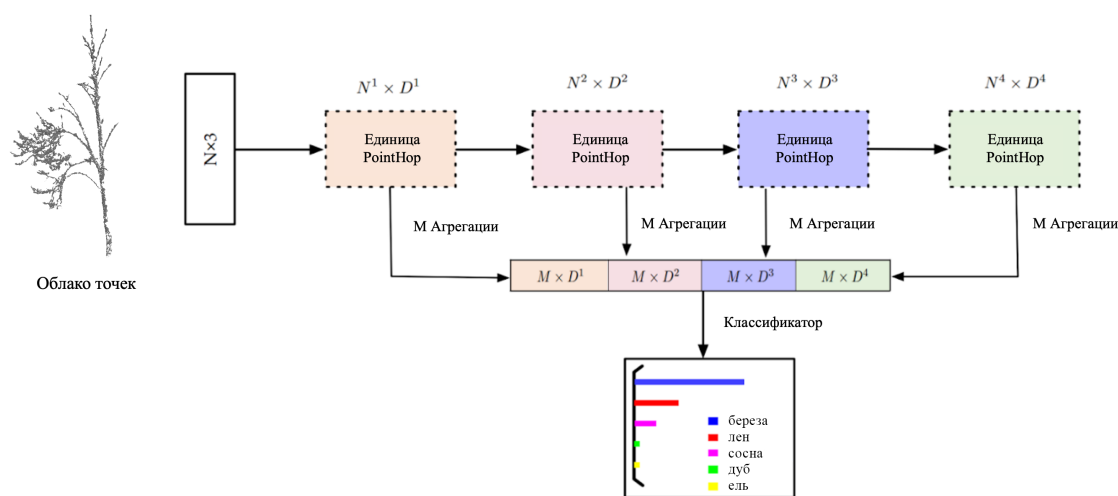
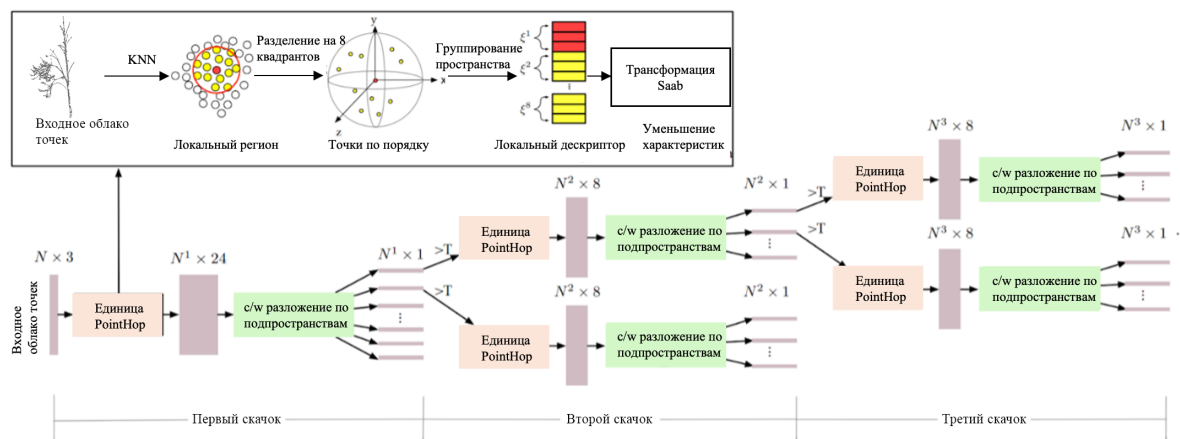


Рисунок 13. Архитектура PointHop

В качестве расширения и дополнительного уменьшения сложности этой модели был предложен PointHop++ [31,32], который автоматически упорядочивает дискриминантные признаки на основе критерия кросс-энтропии для повышения производительности. Процесс работы PointHop++ продемонстрирован на рисунке 14. Данные методы машинного обучения прекрасно работают и наравне с алгоритмами глубокого обучения.



Рисунке 14. Архитектура PointHop++

В этой работе описанные модели будут использоваться для сравнения друг с другом в задаче классификации пород деревьев.

В области машинного обучения и, в частности, в проблеме статистической классификации, матрица смешения, также известная как матрица ошибок[8], представляет собой определенную таблицу, позволяющую визуализировать производительность алгоритма, обычно контролируемого обучения (в неконтролируемом обучении она обычно называется матрицей соответствия). Каждая строка матрицы представляет экземпляры в фактическом классе, а каждый столбец - экземпляры в предсказанном классе. Название матрицы связано с тем, что она позволяет легко определить, путает ли система два класса (т.е. обычно неправильно определяет один класс как другой).

5.2 Результаты

Обучение алгоритмов было реализовано с использованием GPU (NVIDIA K80) на Google Colab с процессом обучения в 50 эпох. Полученная многоклассовая точность представлена в таблице 2.

Таблица 2. Многоклассовая точность и время обучения алгоритмов

Метод	Точность	Время обучения, с
PointNet	0,32	246
PointNet++	0,16	298
PointHop	0,20	129
PointHop++	0,21	139

Кроме того, был проведен анализ точности алгоритмов на классах, число экземпляров которых является наибольшим, а именно береза, сосна, ель и тополь. Для оценки были использованы метрики precision, recall и F1-score, результаты чего изображены в таблице 3.

Таблица 3. Точности моделей для разных классов

	PointNet			PointNet++			PointHop			PointHop++		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Береза	0,06	0,09	0,09	0,11	0,12	0,12	0,03	0,04	0,04	0,05	0,05	0,05
Сосна	0,33	0,22	0,27	0,09	0,17	0,12	0,15	0,17	0,17	0,17	0,2	0,2
Ель	0,5	0,5	0,5	0,08	0,09	0,09	0,2	0,2	0,4	0,4	0,1	0,1
Тополь	0,12	0,2	0,15	0,5	0,2	0,29	0,1	0,2	0,15	0,15	0,4	0,4

Для класса березы наиболее высокие результаты по всем метрикам показала модель PointNet++, для сосны и ели - PointNet, в то время как для тополя по метрикам precision и F1 наилучшие результаты показаны благодаря PointNet++, recall - PointHop.

Также сделано предположение о том, что для небольшой выборки лучше провести бинарную классификацию пород деревьев. Для этой цели были выбраны 3 класса с самым большим числом экземпляров – сосна, ель, береза. Далее были созданы 3 новых датасета, где в отдельный класс был выделен целевая порода дерева, а ко второму классу были отнесены все оставшиеся экземпляры деревьев. Для классификации был выбран алгоритм, показавший наилучшие результаты, PointNet. В результате разбиения на 3 новых датасета было получено распределение данных, которое отображено в таблице 4.

Таблица 4. Распределение экземпляров по новым датасетам

	Датасет с классом "Сосна"		Датасет с классом "Береза"		Датасет с классом "Ель"	
	Сосна	Другое	Береза	Другое	Ель	Другое
Кол-во экземпляров	66	187	35	218	48	205

Для сравнения результатов, полученных во время экспериментов с бинарной классификацией, построена таблица совокупности всех метрик по всем датасетам. Для сравнения используются следующие метрики: accuracy, precision, recall, balanced accuracy, F1-score. Собранные и полученные метрики качества показаны в таблице 5.

Таблица 5. Метрики качества для бинарной классификации всех датасетов

Класс дерева	Accuracy	Recall	Precision	Balanced accuracy	F1-score
Класс "Береза"	0,59	0,37	0,47	0,55	0,41
Класс "Ель"	0,71	0,22	0,2	0,52	0,21
Класс "Сосна"	0,66	0,07	0,4	0,51	0,12

По данной таблице можно сделать вывод, что лучшие результаты данный алгоритм показал на датасете с целевым классом «Ель». Данные результаты могут быть связаны с тем, что для класса «Ель» было собрано больше всего экземпляров. Также данные результаты могут быть связаны с тем, что у экземпляров класса «Ель» больше всего характерных и ярких признаков, которые позволяет идентифицировать образцы этой породы.

Подводя итог по всем алгоритмам и суммируя все полученные данные, получаем таблицу с результатами по accuracy-метрике по каждому из алгоритмов и датасетов, которые были затронуты в этой работе. Таблица результатов анализа алгоритмов представлена в таблице 6.

Таблица 6. Точность и время обучения всех алгоритмов и датасетов

Метод	Assurasy	Время обучения, с
PointNet для мультиклассификации	0,32	246
PointNet ++для мультиклассификации	0,16	298
PointHop для мультиклассификации	0,2	129
PointHop++ для мультиклассификации	0,21	139
PointNet для класса «Береза»	0,59	270
PointNet для класса «Сосна»	0,66	277
PointNet для класса «Ель»	0,71	289

Лучшие результаты по метрике ассурасу показал алгоритм PointNet для бинарной классификации с целевым классом «Береза», но при этом этот алгоритм показал не самые лучшие показатели по времени работы алгоритма, так как алгоритм отработал за 100 эпох и с оптимизатором SDG со скоростью обучения 0,0005. Лучшим по метрике скорости оказался алгоритм PointHop, но при этом по метрике качества ассурасу он показал далеко не самые лучшие результаты. Быструю работу алгоритма можно объяснить тем, что этот алгоритм не является нейросетевым алгоритмом. Также можно предположить, что по этой же причине качество работы этого алгоритма ниже, чем у остальных алгоритмов. Качество обучения алгоритмом PointHop сильно усугубляется недостаточным объемом обучающей и тестовой выборок.

Таким образом после всех экспериментов можно сделать вывод о том, что необходимо собрать более репрезентативную выборку, в которой будет больше обучающих и тестовых экземпляров. Также в итоге лучшим алгоритмом для классификации облаков точек с изображениями деревьев на данной выборке оказался алгоритм PointNet по метрики точности работы алгоритма.

6. ЗАКЛЮЧЕНИЕ

Правильная идентификация видов деревьев является основой для использования и охраны лесных ресурсов, и на данном этапе еще есть много возможностей для развития классификации видов деревьев. В данном исследовании анализируются и сравниваются различные методы и инструменты обработки данных облака точек. Эти методы и инструменты хорошо подходят для предварительной обработки данных облаков точек и создания качественных наборов данных для использования в моделях машинного обучения.

В основном мы выполнили следующие задачи.

1. Мы проанализировали различные методы предварительной обработки данных облака точек.
2. Мы описали наш набор данных облаков точек деревьев и провели предварительную обработку данных облаков точек выше методом. Также проанализирован эффект снижения шума при использовании различных методов.
3. Наконец, мы протестировали распространенные модели классификации облаков точек на нашем наборе данных.

Последующие исследования будут направлены на сравнение влияния различных методов предварительной обработки на эффективность обучения моделей машинного обучения, устранение окклюзии и улучшение алгоритмов выделения отдельных деревьев, чтобы повысить точность классификации и обеспечить более прочную теоретическую основу для применения лазерного картирования при таксации леса.

СПИСОК ЛИТЕРАТУРЫ

1. Anuchin N. P., Lesnaya taksaciya, 3 izd., M., 1971 (in Russian)
2. GISver Integro // LIDAR (Light Detection And Ranging) — Tekhnologiya lazernogo skanirovaniya — URL: <https://gisw.ru/ru/lidar> (accessed 03.11.2022)
3. Aldoma A., Marton Z.C., et al., Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. // IEEE Robot. Autom. Mag. — 2012. — C. 80–91.
4. Saval-Calvo M., Orts-Escolano S., Azorin-Lopez J., et al. A comparative study of downsampling techniques for non-rigid point set registration using color // Bioinspired Computation in Artificial Systems, Springer — 2015. — C. 281–290.
5. Cracknell A. P. Introduction to Remote Sensing (2 ed.). / Cracknell, A. P., Hayes L. // — UK: London. — ISBN 978-0-8493-9255-9.
6. Carter J. Lidar 101: An Introduction to Lidar Technology, Data, and Applications. / Carter J., Schmid K., Waters K., Betzhold L., Hadley B., Mataosky R., Halleran J. // (NOAA) Coastal Services Center. — 2017. — C. 14.
7. Qi C. R. et al. Pointnet: Deep learning on point sets for 3d classification and segmentation // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2017. — C. 652-660.
8. Yang Y. B. PRIN: Pointwise Rotation-Invariant Network // Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). — 2020.
9. Debeunne C., Vivet D. et al. A review of visual-LiDAR fusion based simultaneous localization and mapping // Sensors. — 2020. — C. 2068.
10. Boniardi F., Caselitz T. et al. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) // — 2017. — C. 3318-3324. — ISSN: 1070-9932
11. Gong W., Sun Jia. et al. Investigating the potential of using the spatial and spectral information of multispectral LiDAR for object classification // Sensors. — 2015. — C. 21989-22002.
12. Rusu R. B. 3D is here: Point Cloud Library (PCL) / Rusu R. B., Cousins S. // IEEE International Conference on Robotics and Automation (ICRA). — China: North Charleston, 2011.

13. Zhou Q. Y. Open3D: A Modern Library for 3D Data Processing / Zhou Q. Y., Park J., Koltun V. /— Текст : электронный // arXiv. — 2018. — arXiv:1801.09847.
14. Han X., Jin J. S. et al. A review of algorithms for filtering the 3D point cloud // Signal Processing: Image Communication. — 2017. — С. 103-112.
15. O. Schall, A. Belyaev, H.P. Seidel, Robust filtering of noisy scattered point data, in: Proceedings Eurographics // IEEE VGTC Symposium Point-Based Graphics, Stony Brook, NY, USA, June, 2005, pp. 71–144.
16. Narváez E.A.L., Narváez N.E.L. Point cloud denoising using robust principal component analysis // Proceedings of the First International Conference on Computer Graphics Theory and Applications, Setúbal, – Portugal, 2006. – С. 51–58.
17. Jenke P.M.W., Bokeloh M., Schilling A., Straßer W. Bayesian point cloud reconstruction // Comput. Graph. Forum – 2006. – С. 379–388.
18. Schall O., Belyaev A., Seidel H.P. Adaptive feature-preserving non-local denoising of static and time-varying range data // Comput. Aided Des. – 2008. – С. 701–707.
19. Huhle B., Schairer T., Jenke P., Strasser W. Robust non-local denoising of colored depth data // IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. – Anchorage, 2008. – С. 1–7.
20. Tomasi C., Manduchi R. Bilateral filtering for gray and color images // International Conference on Computer Vision. – Bombay, 1998. – С. 839–846.
21. Xiong B., Jiang W. et al. Voxel Grid-Based Fast Registration of Terrestrial Point Cloud // Remote Sensing. – 2021. – С. 1905.
22. Steder B., Rusu R. B., Konolige K., et al. NARF: 3D range image features for object recognition // Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). – 2010, – С. 2.
23. Ng P. C., Henikoff S. SIFT: Predicting amino acid changes that affect protein function. // Nucleic acids research. – 2003. – С.3812-3814.
24. Loog M, Lauze F. The improbability of Harris interest points. // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2010. – С. 1141-1147.

25. Chen X., Jiang K., Zhu Y., et al. Individual tree crown segmentation directly from UAV-borne LiDAR data using the PointNet of deep learning. // *Forests*. – 2021. – C.131.
26. Qi C. R. et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space // *Advances in neural information processing systems*. – 2017. – Т. 30.
27. Liu B., Chen S., Huang H., et al. Tree Species Classification of Backpack Laser Scanning Data Using the PointNet++ Point Cloud Deep Learning Method. // *Remote Sensing*. – 2022. – C. 3809.
28. Xi Z, Hopkinson C, Rood S B, et al. See the forest and the trees: Effective machine and deep learning algorithms for wood filtering and tree species classification from terrestrial laser scanning. // *ISPRS Journal of Photogrammetry and Remote Sensing*. – 2020, – C. 1-16.
29. Xi Z., Hopkinson C., Rood S. B., et al. See the forest and the trees: Effective machine and deep learning algorithms for wood filtering and tree species classification from terrestrial laser scanning. // *ISPRS Journal of Photogrammetry and Remote Sensing*. – 2020. – C. 1-16.
30. Zhang M. et al. Pointhop: An explainable machine learning method for point cloud classification // *IEEE Transactions on Multimedia*. – 2020. – Т. 22. – №. 7. – С. 1744-1755.
31. Zhang M. et al. Pointhop++: A lightweight learning model on point sets for 3d classification // *2020 IEEE International Conference on Image Processing (ICIP)*. – 2020. – С. 3319-3323.
32. Федорова А. А., Нурлыева Д. Д., Желанкина А. С., et al. СРАВНЕНИЕ МЕТОДОВ КЛАССИФИКАЦИИ ОБЛАКОВ ТОЧЕК, ПОЛУЧЕННЫХ С ПОМОЩЬЮ ТЕХНОЛОГИИ LIDAR // *Искусственный интеллект в автоматизированных системах управления и обработки данных*. – 2022. – С. 262-267.