

Реферат

ДИСТАНЦИОННОЕ ЗОНДИРОВАНИЕ ЗЕМЛИ, СПУТНИКОВАЯ СЦЕНА, ПРОСТРАНСТВЕННЫЕ ДЕРЕВЬЯ РЕШЕНИЙ, КЛАССИФИКАЦИЯ

Объект исследования – классификация данных дистанционного зондирования Земли.

Предмет исследования - пространственные деревья решений для классификации данных дистанционного зондирования Земли.

Цель работы – разработка улучшенных версий пространственных деревьев решений для классификации данных дистанционного зондирования Земли.

В результате работы были реализованы и сравнены существующие алгоритмы деревьев решений, которые во время обучения опираются на пространственные характеристики входных данных, а также были реализованы их улучшенные версии. Рассмотрен и протестирован новый подход к обработке входных данных для пространственных деревьев решений, основанный на алгоритме уменьшения изображений. Результаты в отчете представлены в виде псевдокодов и значений метрик качества классификации на спутниковых снимках сельскохозяйственных культур.

Содержание

Введение	4
1 Данные дистанционного зондирования Земли	6
1.1 Принципы дистанционного зондирования Земли	6
1.2 Данные космического аппарата Landsat	7
1.3 Выбранная область исследований	7
1.4 Выводы по разделу	8
2 Пространственные деревья решений	10
2.1 Традиционные и пространственные деревья решений	10
2.2 Пространственное дерево решений	10
2.3 Пространственное дерево на основе фокальных тестов	11
2.4 Выводы по разделу	16
3 Модификации пространственных деревьев решений	17
3.1 Пространственное дерево решений	17
3.2 Пространственное дерево с круглой областью соседствующих пикселей	17
3.3 Пространственное дерево на основе фокальных тестов	19
3.4 Алгоритм быстрого уменьшения изображений	19
3.5 Детали реализации	21
4 Экспериментальное сравнение пространственных деревьев решений	23
4.1 Используемые метрики	23
4.2 Результаты сравнений на необработанных данных	23
4.3 Результаты сравнений на уменьшенных данных	24
4.4 Исследование пространственного дерева на обычных и уменьшенных данных	24
4.5 Сравнение скорости работы и оценок алгоритма пространственного дерева решений при различных параметрах	25
4.6 Выводы по разделу	29
Заключение	31
Список использованных источников	32
А Иллюстрации	33

Термины и определения

В настоящем отчете о НИР применяются следующие термины с соответствующими определениями:

Пиксель — наименьшая единица растрового изображения, получаемого с помощью графических систем вывода информации, составляющая цифровое изображение. Представляет собой число или вектор, описывающий интенсивность излучения в определенных спектрах.

Цифровое (растровое) изображение — прямоугольный двумерный массив чисел, при этом каждое число соответствует пикселю.

Пространственное разрешение космического снимка — величина, характеризующая размер одного пикселя, т. е. наименьшей точки, которую способен различить спутниковый сенсор.

Спутниковая сцена — часть принимаемого со спутника потока данных. Состоит из одного или нескольких цифровых изображений (спектральных каналов), представляющих ту же пространственную область и имеющих одинаковое пространственное разрешение.

Дерево решений — алгоритм машинного обучения, позволяющий решать задачи регрессии и классификации.

Пространственное дерево решений — дерево решений, используемое в задачах классификации пикселей изображения и использующее информацию об их пространственном взаимоотношении.

Данные дистанционного зондирования Земли — обработанные и представленные в виде растров сцены земной поверхности, полученные со спутников.

Шум «соли и перца» — вид шума в цифровых изображениях, представляющий собой пиксели с чрезмерно высоким или низким значением в случайных местах.

Введение

В настоящее время спутниковые данные предоставляют множество возможностей для анализа и классификации различных географических объектов и использоваться для:

- Сельского хозяйства (классификация типов культур, оценка урожайности, мониторинг состояния посевов)
- Изучение природных ресурсов Земли
- Мониторинга (лесного покрова, океана, атмосферных слоев, городов)
- Решения задач метеорологии, геодезии, картографии

Однако, для того чтобы использовать эти данные эффективно, требуется разработка методов и алгоритмов, которые позволят классифицировать их на основе специфических характеристик. В этом контексте методы классификации спутниковых данных с помощью пространственных деревьев решений представляют собой удобный и эффективный инструмент для обработки геоданных.

В отличие от популярных сегодня алгоритмов глубинного обучения, деревья позволяют строить интерпретируемые модели, абсолютно понятные для человека, так как состоят из решающих правил. Также, за счет своей простоты, они быстрее обрабатывают массивы входных данных и генерируют предсказания. Деревья до сих пор остаются актуальными за счет этих свойств.

В отличие от обычных деревьев решений, пространственные деревья решений используют данные о пространственном взаимоотношении пикселей входных данных друг с другом, что обеспечивает им весомое преимущество по сравнению с обычными деревьями решений, а именно выбор более близких к реальности решающих правил. Так как в реальном мире многие географические объекты на спутниковых снимках порождают области пикселей, тесно связанные друг с другом, то при использовании этого свойства в качестве дополнительной информации можно получить более точный и правдоподобный результат классификации.

Помимо широкого применения пространственных деревьев решений в геоинформационных системах, они также могут быть применены в целом для обработки любой многомерной информации, начиная от обычных изображений и, например, рентгеновских снимков в области медицины, заканчивая многомерными спутниковыми сценами.

С другой стороны, данный метод имеет и свои недостатки. Во-первых, усложняется алгоритм построения дерева, что влечет за собой более существенные временные издержки при его использовании. Во-вторых, имеют место недостатки входных данных, такие как шум вида "соль и перец", который возлагает на исследователей задачу по выбору и созданию подходящих для практического применения метрик и оптимизаций. Частичное решение данных проблем представлено в данном отчете.

1 Данные дистанционного зондирования Земли

В данном разделе раскрываются некоторые ключевые понятия и принципы о данных дистанционного зондирования Земли, а также приводятся характеристики выбранной области исследований.

1.1 Принципы дистанционного зондирования Земли

Дистанционное зондирование Земли - это наука и совокупность технологий, которые получают, обрабатывают и интерпретируют информацию о различных объектах земной поверхности без прямого физического контакта с этой поверхностью, - посредством фиксирования отраженной солнечной энергии и излученной тепловой энергии (электромагнитного излучения) от земной поверхности, реализующегося сенсорами, расположенными на орбитальных спутниках планеты.

Дистанционное зондирование Земли основывается на использовании спутниковых датчиков для получения данных об объектах на Земле с помощью считывания отраженного и/или излученного от поверхности планеты электромагнитного спектра.

Спектр электромагнитных волн включает в себя широкий диапазон частот – от радиоволн до гамма-излучения. Каждый тип волны имеет свои особенности и используется для получения определенных данных о поверхности Земли с помощью спутниковых датчиков. Эти приборы регистрируют отраженное от земной поверхности солнечное излучение в нескольких спектральных диапазонах (видимых и инфракрасных), и снимки, таким образом, несут огромное количество качественной информации о структуре и состоянии растительного покрова, выраженной в количественной, цифровой форме.

Для получения подходящей информации используются спектральные характеристики, которые определяют, какие частоты электромагнитного спектра наиболее важны для получения данных о заданном объекте. Например, когда желаемыми данными являются сведения о растительности, то наиболее важной частотой является ближняя инфракрасная область спектра.

1.2 Данные космического аппарата Landsat

Космический аппарат Landsat оснащен несколькими каналами, которые позволяют собирать несколько типов данных. В настоящее время имеются две активные подсистемы Landsat: Landsat 8 и Landsat 9.

Landsat 8 оснащен 11 каналами для получения спутниковых сцен в разных спектрах. Для сбора данных о водной поверхности используется канал 1 – чувствительный к синему и зеленому цветам, "видящий" электромагнитные волны с длинами 0.433 – 0.453 мкм. Для минимизации влияния атмосферных эффектов используются 9-й и 10-й каналы. Каналы 4, 5 и 6 особенно полезны для анализа растительности. Пространственное разрешение каналов – 30 метров на пиксель, кроме 8 канала (здесь – 15 метров на пиксель) и каналов 10, 11 (разрешение – 100м на пиксель).

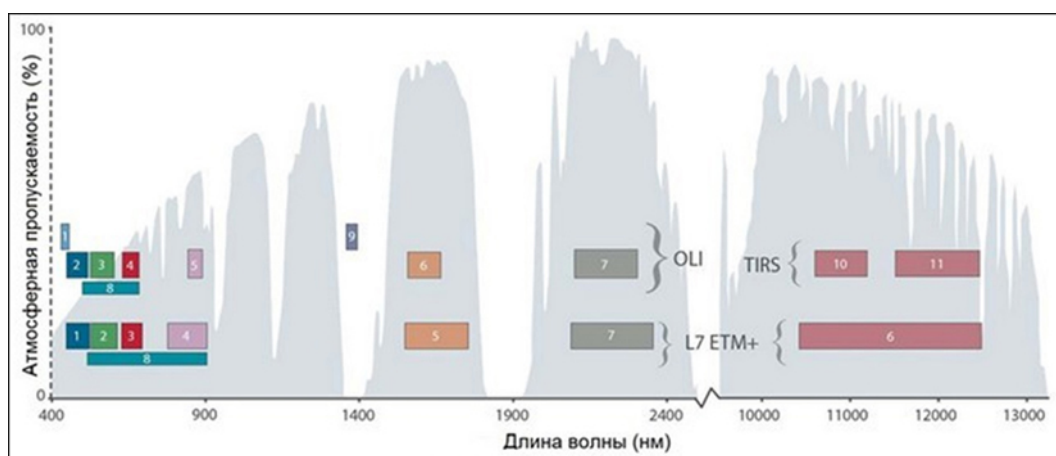


Рисунок 1.1 — Соответствие длин электромагнитных волн каналам спутников Landsat 8 и 9, способных их детектировать

1.3 Выбранная область исследований

В качестве входных данных рассматриваются спутниковые сцены сельхоз. поля с различным спектральным разрешением и пространственным разрешением 30м на пиксель. Во входных данных также используется массив пикселей NDVI (normalized difference vegetation index) — это индекс, используемый в дистанционном зондировании Земли, который позволяет оценить плотность и здоровье растительности на поверхности Земли. Он основывается на измерении отношения света, поглощенного

растительностью, к свету, отраженному ею. Классификация происходит по наличию на участке поля, соответствующему данному пикселю, растений люцерны.



Рисунок 1.2 — Пример спутникового снимка, использованного в качестве входных данных для алгоритмов

1.4 Выводы по разделу

В данном разделе рассмотрены ключевые понятия и принципы дистанционного зондирования Земли, а также приведены характеристики используемых в этом процессе сенсоров и спектральных диапазонов. Описана работа космического аппарата Landsat и указаны наиболее полезные для анализа растительности каналы. В тексте также упоминается индекс NDVI и его использование для оценки здоровья и плотности рас-

тительности на поверхности Земли. Классификация происходит по наличию на участке поля, соответствующему данному пикселю, растений люцерны. Весь текст посвящен тому, каким образом можно получить качественную информацию о структуре и состоянии растительного покрова Земли, используя спутниковые датчики и методы дистанционного зондирования.

Таким образом, современные технологические возможности позволяют получать большое количество разнообразных геоинформационных данных, что делает исследования в данной области с помощью методов машинного обучения актуальным.

2 Пространственные деревья решений

В данном разделе рассмотрены конкретные отличия пространственных деревьев решений от традиционных, а также приведены псевдокоды алгоритмов.

2.1 Традиционные и пространственные деревья решений

Общим для обычных деревьев и рассмотренных в отчете пространственных является то, что оба алгоритма действуют по жадной стратегии, рассматривая каждый признак входных данных и находя такое значение этого признака, на котором целевая метрика является максимальной. Конкретно в данном отчете рассматриваются такие деревья, которые являются бинарными и строят решающие правила так, что на них можно ответить true/false. Решающие правила имеют вид $x \leq t$, где x представляет собой множество входных данных, а t является некоторым числом, которое также называют порог (threshold). Данные деревья решают задачу бинарной классификации.

Пространственные деревья отличаются от обычных использованием определенных эвристик, учитывающих пространственное отношение определенных пикселей друг относительно друга на изображении. Это позволяет алгоритмам машинного обучения учитывать не только характер форм классифицируемых объектов, но и распознавать различные неполноценности во входных данных, такие как шум, в частности - шум вида "соль и перец", на что обычные деревья не способны.

2.2 Пространственное дерево решений

Для нижеприведенных алгоритмов используется следующая вспомогательная метрика, учитывающая этот шум: NSAR (Neighborhood Split Autocorrelation Ratio), вычисляющаяся для конкретного пикселя i по формуле $NSAR_i = \frac{c'(i)}{c(i)}$, где $c(i)$ - количество соседей данного i -того пикселя того же класса до разбиения решающим правилом, а $c'(i)$ - количество соседей того же класса для данного пикселя после разбиения.

Вместо обычного прироста информации, используемого в традиционных деревьях, здесь лучшее разбиение находится по метрике пространственного прироста информации (Spatial Information Gain), который находят по формуле: $(1 - \alpha)IG + \alpha \overline{NSAR}$, где IG - прирост информации (рассчитывающийся как разность между энтропиями до и после разбиения. Энтропия рассчитывается по формуле Шеннона), \overline{NSAR} - среднее значение NSAR по всем пикселям, α - вещественный коэффициент, являющийся гиперпараметром.

Выбор способа определения соседних пикселей к данному может быть задан любым отношением. Ниже (1) представлен псевдокод алгоритма Spatial Decision Tree из книги [1]. В ходе исследовательской работы на языке Python был реализован рассматриваемый алгоритм, где в первом случае соседями являются те пиксели, которые расположены на одну позицию вверх, вниз, вправо и влево (образуя "крест" в квадрате 3 на 3 пикселя), а во втором соседями для пикселя считаются те пиксели, которые попадают в окружность заданного заранее радиуса с центром в данном пикселе. В общем случае, определение соседства также является гиперпараметром.

2.3 Пространственное дерево на основе фокальных тестов

Вторым алгоритмом пространственного дерева решений является дерево под названием Focal Test Based Spatial Decision Tree (FTSDTree), представленный в книге [1] и в виде псевдокода (2). Для поиска лучшего правила данный алгоритм использует обычный Information Gain, в отличие от SDTree. Использование пространственных метрик здесь заключено в изменении алгоритма разбиения набора входных данных на поднаборы. Для этого на каждом разбиении используется т.н. метрика Гамма-индекс, рассчитываемая для каждого пикселя по формуле:

$$\Gamma_i = \frac{\sum_j W_{i,j} I_i I_j}{\sum_j W_{i,j}},$$

где I_i (или I_j) равно 1, если значение признака пикселя i (или j) меньше либо равно значению threshold на данном разбиении и -1 иначе; $W_{i,j}$ равно 1, если пиксель i является соседом для пикселя j и 0 иначе.

Algorithm 1 SDTree (Spatial Decision Tree)

Require: G - a graph (matrix of zeros and ones), representing the input data samples included in the current node of the tree

F - set of 2D training data matrices

C - a two-dimensional matrix of zeros and ones representing the true pixel classes of the input data

α - the significance of the NSAR metric regarding information gain

m - sample count threshold for the node

Ensure: Root of the spatial decision tree

- 1: **if** $|G| < m$ or G consists of one class according to C **then**
 - 2: make a leaf and return it
 - 3: **end if**
 - 4: **for** each feature matrix f in F **do**
 - 5: get array V of sorted values from f
 - 6: **for** each unique value v from V **do**
 - 7: get graphs $G1$ and $G2$ by splitting graph G with threshold v
 - 8: find value of SIG
 - 9: **end for**
 - 10: **end for**
 - 11: find best threshold according to the SIG values
 - 12: get graphs $G1$ and $G2$ by splitting G with the best threshold
 - 13: left child of the current node = SDTree($G1, F, C, \alpha, m$)
 - 14: right child of the current node = SDTree($G2, F, C, \alpha, m$)
 - 15: return current node
-

Таким образом, эта метрика представляет собой меру себеподобных по данному разбиению среди соседей. Отрицательное значение указывает на большое количество соседей, не попавших в то же разбиение, что и рассматриваемый пиксель. На этом основан алгоритм FTSDTree, который формирует матрицу (далее - гамма-матрица) из значений true и false, где true для данного пикселя означает неотрицательное значение гамма индекса и false иначе. Другими словами, значения false данной метрики указывают на потенциальные шумовые пиксели.

На каждом разбиении, получив матрицу из булевых значений $x \leq threshold$ (где x - пиксели входных данных), она подвергается поэлементному XOR'у с гамма-матрицей. Это делается для того, чтобы "вернуть" потенциально шумовые пиксели в своё разбиение. За счет этого достигается устранение шума "соль и перец".

В реализации данного алгоритма в качестве соседствующих пикселей считаются такие, которые находятся в квадратной области (сторона квадрата равна $2s + 1$) такой, что рассматриваемый пиксель находится в ее центре.

В отличие от вышеприведенных алгоритмов, функция предсказания которых идентична той, которая используется в обычных деревьях, в FTSDTree для предсказаний используется рекурсивная функция с вызовом Node-Split, псевдокод которых также представлен ниже (3, 4).

Пояснения к функции Node-Split. neighType - параметр, отвечающий за "тип" подсчета количеств соседствующих пикселей при расчете гамма-матрицы. При значении neighType = 0, вышеописанный метод (по формуле гамма-метрики) не меняется. При значении neighType = 1, гамма-индекс данного пикселя равен 1, если компонента связности данного пикселя касается или выходит за рамки соответствующей квадратной области, и -1 иначе. Пиксели лежат в одной компоненте связности тогда и только тогда, когда имеют одинаковое значение true или false в зависимости от сравнения с threshold, и при этом от одного пикселя к другому можно "пройти" по пикселям с таким же значением относительно threshold такими шагами, что шаг - это переход от пикселя вверх, вниз, влево или вправо на один пиксель.

Algorithm 2 FTSDTree (Focal-Test-Based Spatial Decision Tree)

Require: G - a graph (matrix of zeros and ones), representing the input data samples included in the current node of the tree

F - set of 2D training data matrices

S_{max} - maximum neighbourhood size

$neighType$ - neighbourhood type (0 for fixed, 1 for adapted)

C - a two-dimensional matrix of zeros and ones representing the true pixel classes of the input data

m - sample count threshold for the node

Ensure: Root of the FTSD tree

- 1: **if** $|G| < m$ or G consists of one class according to C **then**
 - 2: make a leaf and return it
 - 3: **end if**
 - 4: **for** each feature matrix f in F **do**
 - 5: get array V of sorted values from f
 - 6: **for** each s from 0 to S_{max} **do**
 - 7: **for** each unique value v from V **do**
 - 8: $G1, G2 = \text{Node-Split}(G, f, v, s, neighType)$
 - 9: find value of IG
 - 10: **end for**
 - 11: **end for**
 - 12: **end for**
 - 13: find best threshold according to the IG values
 - 14: get graphs $G1$ and $G2$ by splitting G using Node-Split
 - 15: left child of the current node = FTSDTree($G1, F, C, \alpha, m$)
 - 16: right child of the current node = FTSDTree($G2, F, C, \alpha, m$)
 - 17: return current node
-

Algorithm 3 Node-Split

Require: G - a graph (matrix of zeros and ones), representing the input data samples included in the current node of the tree

f - data matrix

v - threshold

s - neighbourhood size

$neighType$ - neighbourhood type (0 for fixed, 1 for adapted)

Ensure: Split of the G graph (Graphs $G1$ and $G2$)

1: $M = \text{Gamma-Matrix}(f, v, s, neighType)$

2: $I = f \leq v$

3: $I = \text{element-wise-XOR}(I, M)$

4: return $I == \text{true}$ as $G1$, $I == \text{false}$ as $G2$

Algorithm 4 FTSDTree-Predict

Require: G - a graph (matrix of zeros and ones), representing the input data samples included in the current node of the tree

R - root of the FTSD tree

Ensure: Matrix with predictions of pixel values from G pixels

1: **if** $R.type == \text{Leaf}$ **then**

2: assign G with R class

3: return G

4: **end if**

5: get best parameters of the node

6: $G1, G2 = \text{Node-Split}(params)$

7: $C1 = \text{FTSDTree-Predict}(G1, R.left)$

8: $C2 = \text{FTSDTree-Predict}(G2, R.right)$

9: return $C1 + C2$

2.4 Выводы по разделу

За ключевыми различиями пространственных деревьев от обычных стоят эвристики, интуитивно соответствующие пониманию того, как пиксели объектов на изображении должны быть расположены друг относительно друга в реальном мире. Таким образом, они являются примером того, как понимание природы полученных данных может быть успешно применено для новых подходов в создании алгоритмов машинного обучения.

В свою очередь, это ставит перед исследователями такие нетривиальные задачи, как поиск и изобретение подходящих для определенных данных эвристик и задачи по оптимизации уже существующих алгоритмов пространственных деревьев.

3 Модификации пространственных деревьев решений

В данном разделе описана реализация трех деревьев на основе вышеприведенных алгоритмов с дополнительными оптимизациями и упрощениями, а также представлен модифицированный алгоритм уменьшения размерности пространственных данных на основе алгоритма уменьшений изображений с сохранением качества [4].

3.1 Пространственное дерево решений

Псевдокод обучения данного дерева (Spatial Decision Tree, SDTree) представлен в алгоритме 5. Главное отличие от оригинального алгоритма - более примитивный рандомизированный метод просмотра значений, основанный на случайном выборе матриц входных данных для рассмотрения и случайно выбранных уникальных значений в данной матрице (их количество задается заранее), что позволяет многократно повысить скорость выполнения при небольшой потере качества обучения. Данный метод широко применяется для многих алгоритмов деревьев решений с целью добиться более быстрой их работы за счет незначительного ухудшения в качестве. Помимо минимального размера узла для создания листа, также введено ограничение на максимальную глубину дерева. В данной версии алгоритма соседствующими пикселями являются верхний, нижний, левый и правый пиксели в квадрате 3x3 (в форме "креста").

3.2 Пространственное дерево с круглой областью соседствующих пикселей

В рамках данной работы был реализован и протестирован алгоритм пространственного дерева Circle-Based Spatial Decision Tree (CSDTree), использующий метод выделения соседства в круге заранее заданного радиуса, что можно обосновать интуитивно природой входных данных, а именно наличием в них большого количества предметов со скругленными краями, в частности целиком круглой формы (круглые поля классифицируемых культур). В остальном, алгоритм повторяет вы-

Algorithm 5 SDTree (Spatial Decision Tree)

Require: G - a graph (matrix of zeros and ones), representing the input data samples included in the current node of the tree

F - set of 2D training data matrices

C - a two-dimensional matrix of zeros and ones representing the true pixel classes of the input data

α - the significance of the NSAR metric regarding information gain

m - sample count threshold for the node

$uniqueSize$ - the number of considered unique values from the feature matrix selected randomly

$fsize$ - the number of considered matrices from the input data, chosen randomly

$maxDepth$ - max depth of the tree

$currentDepth$ - depth of the current node

Ensure: Root of the spatial decision tree

- 1: **if** $|G| < m$ or G consists of one class according to C or $currentDepth == maxDepth$ **then**
 - 2: make a leaf and return it
 - 3: **end if**
 - 4: select $fsize$ random matrices from F randomly
 - 5: **for** each feature matrix f in selected matrices **do**
 - 6: get array V of sorted values from f
 - 7: select $uniqueSize$ values from V randomly
 - 8: **for** each unique value v from selected values **do**
 - 9: get graphs $G1$ and $G2$ by splitting graph G with threshold v
 - 10: find value of SIG
 - 11: **end for**
 - 12: **end for**
 - 13: find best threshold according to the SIG values
 - 14: get graphs $G1$ and $G2$ by splitting G with the best threshold
 - 15: left child of the current node = SDTree($G1$, F , C , α , m)
 - 16: right child of the current node = SDTree($G2$, F , C , α , m)
 - 17: return current node
-

шеприведенный, однако на вход также требует параметр ϕ - радиус круга для учета пикселей как соседних.

3.3 Пространственное дерево на основе фокальных тестов

В ходе работы был реализован и протестирован алгоритм Focal-Test-Based Decision Tree (FTSDTree) со следующими упрощениями: размер s всегда фиксирован, `neighType` всегда равен 0. Также каждый узел дерева, как и в вышеприведенных реализациях, оперирует рандомизированными выборками. Данные упрощения позволили значительно ускорить работу алгоритма без значительного снижения качества и протестировать его для обработанных и необработанных данных с более быстрым получением результатов исследования.

3.4 Алгоритм быстрого уменьшения изображений

Алгоритм быстрого уменьшения изображений с сохранением деталей Rapid, Detail-Preserving Image Downscaling (RDPID) из [4] был модифицирован и обобщен для многомерных матриц таким образом, чтобы с его помощью можно было обрабатывать не только rgb изображения, но и любые трехмерные данные, включая спутниковые сцены. Псевдокод полученного алгоритма представлен в алгоритме 6.

Пояснения к алгоритму Rapid downscaling.

Строки 1-4: Масштабируем каждую матрицу, поэлементно разделив каждое число в конкретной матрице на максимум среди чисел данной матрицы, запомнив эти максимумы (чтобы результат привести в исходный вид).

Строки 7-10: Для каждой предварительно отмасштабированной матрицы применяется линейный матричный фильтр (функция Convolution-Filter), осуществляя линейную свертку каждой матрицы. Свертка - это операция преобразования матрицы с помощью ядра (матрицы kernel) так, что ядро прикладывается к пикселям изображения с шагом d , поэлементно умножает покрытые элементы, складывает полу-

ченные числа и результирующее число проецирует в новое изображение. Таким образом, каждое "окно" с размером $d \times d$ пикселей исходной матрицы проецируется в новую матрицу *downscaled matrix*. С оставшимися пикселями, "не влезшими" в окно, можно поступать по-разному. В данной работе был реализован алгоритм, не учитывающий в результате остаточные пиксели.

Строки 13-16: Аналогично предыдущим, но теперь с шагом 1. Обработать угловые и краевые пиксели можно адаптируя размер ядра *kernel* и нормирующий коэффициент, на который делится каждый элемент матрицы данного ядра. В результате получим "размытую" версию уменьшенной матрицы в переменной *smoothed matrix*, которая добавляется к массиву \tilde{I} .

Строка 17: Расчет пикселей выходного изображения по нижеприведенной формуле. При вычислении следует транспонировать набор матриц \tilde{I} таким образом, чтобы получить матрицу $w \times h$ с пикселями размерности n .

Строки 18-19: Результат транспонируется обратно в набор матриц, умножая каждую матрицу при этом на ранее сохраненные максимумы.

Вычисление пикселей выходного изображения осуществляется по следующей формуле:

$$O(p) = \frac{1}{k_p} \sum_{q \in \Omega_I(p)} I(q) (\|I(q) - \tilde{I}(p)\|_2 / \sqrt{n})^\lambda, \text{ где}$$

p - пиксель выходного изображения, в который проецируются пиксели исходного изображения q ;

$\Omega_I(p)$ - область пикселей q , соответствующая пикселю выходного изображения p ;

$I(q)$ - значение пикселя q (вектор из n чисел) исходного изображения;

$\tilde{I}(p)$ - значение соответствующего пикселю q из исходного изображения пикселя p из матрицы \tilde{I} , полученной в алгоритме в строках 13-16;

$$k_p = \sum_{q \in \Omega_I(p)} (\|I(q) - \tilde{I}(p)\|_2 / \sqrt{n})^\lambda \text{ в случае, если isSeparated=false,}$$

иначе формула вырождается в:

$$O[c](p) = \frac{1}{k_p} \sum_{q \in \Omega_I(p)} I(q) (|I(q) - \tilde{I}(p)|)^\lambda \text{ с соответствующим нормализующим коэффициентом } k_p.$$

isSeparated отвечает за режим применения уменьшения: если он равен true, то каждое изображение входных данных уменьшается отдельно, иначе вся матрица воспринимается как изображение с пикселями размерности n и уменьшается подобно оригинальному алгоритму - с учетом значений пикселя во всех n измерениях.

Масштабирование признаков в начале алгоритма необходимо для облегчения работы с формулой: в отличие от оригинальной формулы, здесь нормирующий коэффициент V_{max} всегда равен \sqrt{n} , т.к. это максимально возможное значение евклидового расстояния между вектором из нулей и единиц, каждый с размерностью n .

3.5 Детали реализации

Алгоритмы были написаны в среде разработки PyCharm с использованием интерактивного блокнота Jupyter Notebook, который был использован для тестирования алгоритмов. Язык реализации - python версии 3.10.6, библиотека для выполнения большинства операций подсчета - numru. Также были использованы такие библиотеки, как PIL (для получения изображений из массивов NumPy), sklearn.metrics, matplotlib, pyrgis (для преобразования файлов спутниковых данных формата .tif в numru массивы).

Algorithm 6 Rapid downscaling

Require: *dataArray* - input multidimensional image as an array of images (matrices) with the shape of n, w, h

d - an integer corresponding to the ratio of the size of the sides of the original image to the reduced one

isSeparated - True if we want to reduce the layers individually, otherwise False

λ - lambda hyperparameter, affecting the sharpness of image details

Ensure: array of images reduced by d times with the shape of $n, w//d, h//d$

```
1: for matrix in dataArray do
2:     save maximum value of matrix                                ▷ needed for unscaling
3:     assign matrix with matrix/max                               ▷ scaling data
4: end for
5: kernel =  $d$  by  $d$  matrix of 1s divided by  $d * d$ 
6: assign downscaled an empty array
7: for matrix in dataArray do
8:     apply Convolution-Filter to matrix with kernel and stride  $d$ 
9:     append downscaled matrix to the downscaled array
10: end for
11: kernel =  $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ 
12: assign  $\tilde{I}$  an empty array
13: for matrix in downscaled do
14:     apply Convolution-Filter to matrix with kernel and stride 1
15:     append smoothed matrix to the  $\tilde{I}$  array
16: end for
17: calculate pixel values  $O(p)$  for each pixel  $p$  by formula
18: transpose  $O$  to the shape  $n, w_o, h_o$ 
19: multiply each matrix from  $O$  by corresponding maximum saved
    previously
20: return image  $O$ 
```

4 Экспериментальное сравнение пространственных деревьев решений

В данном разделе приведены результаты некоторых экспериментов с реализованными алгоритмами деревьев решений.

4.1 Используемые метрики

Для оценки качества классификаций используются такие метрики, как recall, precision и accuracy, а также рассмотрена метрика на основе гамма статистик (далее - gamma) для сравнения уровня "за шумленности" полученных предсказаний. В качестве neighbourhood данной метрики для пикселя будет использоваться квадратная область 3x3 целиком, не включая рассматриваемый пиксель.

Метрика gamma вычисляется как $\Gamma = \frac{\sum \Gamma_2(i)}{\sum \Gamma_1(i)}$, где числитель - сумма количеств соседей для пикселей из реальных данных, а в знаменателе - для предсказания. Чем меньше эта метрика, тем больше в среднем у каждого пикселя предсказания одинаковых соседей, и тем меньше потенциально шумовых пикселей.

4.2 Результаты сравнений на необработанных данных

Для тестирования были выбраны 2 участка из исходных данных каждый размером 500x500 пикселей (для тренировки и тестирования). Деревья имели следующие параметры:

Алгоритм	alpha	maxDepth	minNodeSize	uniqueSize	fsize
SDTree	0.25	5	800	10	3
CSDTree	0.25	5	800	10	3
FTSDTree	—	05	800	10	3

Для CSDTree значение ϕ (радиус) было равным 3, а для FTSDTree значение параметра neighbourhoodSize было равным 4. Были получены оценки метрик, указанные в 4.1

Таблица 4.1 — Значения метрик относительно алгоритмов

Алгоритм	Recall	Precision	Accuracy	Gamma
SDTree	0.7153	0.2233	0.8648	0.9849
CSDTree	0.8941	0.4049	0.8995	0.9909
FTSDTree	0.7184	0.3998	0.8817	0.9864

4.3 Результаты сравнений на уменьшенных данных

Для данного тестирования в качестве тренировочных данных была выбрана область 1000x1000 пикселей из исходных данных и уменьшена в 4 раза с параметром λ равным 0.3. В качестве данных для тестирования была выбрана та же область 500x500 пикселей, что и в предыдущем измерении. Сравнения приведены в аналогичной таблице 4.2.

Таблица 4.2 — Значения метрик относительно алгоритмов (уменьшенные данные)

Алгоритм	Recall	Precision	Accuracy	Gamma
SDTree	0.8793	0.4347	0.9024	0.9923
CSDTree	0.8546	0.4343	0.9001	0.9951
FTSDTree	0.8237	0.4640	0.9007	0.9920

4.4 Исследование пространственного дерева на обычных и уменьшенных данных

Условия тестирования следующие:

Случайным образом выбирается 10 областей 500x1000 пикселей, разделяется напополам для формирования тренировочной и тестовой выборок и оценивается среднее значение метрик. Затем повторяется то же самое, но для уменьшенных в 2 раза данных.

Выбор областей таков, что в область 500x1000 пикселей попадает от 16 до 50 процентов пикселей со значением класса true из истинных данных. Результаты приведены в таблице 4.3, а также результаты классификаций в виде изображений приведены в приложении.

Таблица 4.3 — Значения метрик относительно обработки данных

Вид данных	Recall	Precision	Accuracy	Gamma
Исходные	0.28903	0.17707	0.74915	1.0087
Уменьшенные	0.26960	0.18707	0.73814	1.0118

4.5 Сравнение скорости работы и оценок алгоритма пространственного дерева решений при различных параметрах

В данном подразделе сравнивается скорость обработки реализованным алгоритмом SDTree массивов данных при различных размерах этих массивов и при различных значениях параметров алгоритма во время обучения, а также приведены оценки метрик.

В случае тренировки и тестирования алгоритма на областях размером 100x100 пикселей получаем следующие результаты:

Аргументы:

Алгоритм	alpha	maxDepth	minNodeSize	uniqueSize	fsize
SDTree	0.25	6	1	10	3

Значения полученных метрик:

- Recall = 0.98744
- Precision = 0.2861
- Accuracy = 0.8423
- Gamma = 0.945

Выберем теперь такие аргументы, которые позволят обработать все значения исходных (9) матриц:

Алгоритм	alpha	maxDepth	minNodeSize	uniqueSize	fsize
SDTree	0.25	6	1	max	9

Получим следующие оценки:

- Recall = 0.8370
- Precision = 0.3762
- Accuracy = 0.8468
- Gamma = 0.9678

При этом, в первом случае алгоритм затратил на обучение менее секунды, а второй - около 7 минут.

Видно, что во втором случае, проходя по каждому значению и каждой матрице входных признаков, увеличилась гамма метрика, что указывает на большее количество потенциально шумовых пикселей, ухудшился recall - дерево не смогло выявить положительные классы там, где могло в предыдущем случае, но улучшился precision. При этом accuracy практически не изменилась. Это может быть связано с тем, что чем больше значений признаков рассматривается на каждом узле, тем более этот узел "жадный" а в случае со случайным выбором подмножеств значений признаков можно сделать каждый конкретный узел менее "жадным" но в среднем дерево способность к обобщению информации не потеряет. Из этого можно сделать вывод, что рандомизированный выбор ограниченного числа признаков не критично влияет на основные метрики, но очень влияет на время обработки.

Далее приведен график (4.1) зависимости времени обработки от размера матрицы входных данных. Параметры дерева следующие:

Алгоритм	alpha	maxDepth	minNodeSize	uniqueSize	fsize
SDTree	0.25	2	1	10	1

Параметры подобраны так, что рассчитывается время обучения только одного узла.

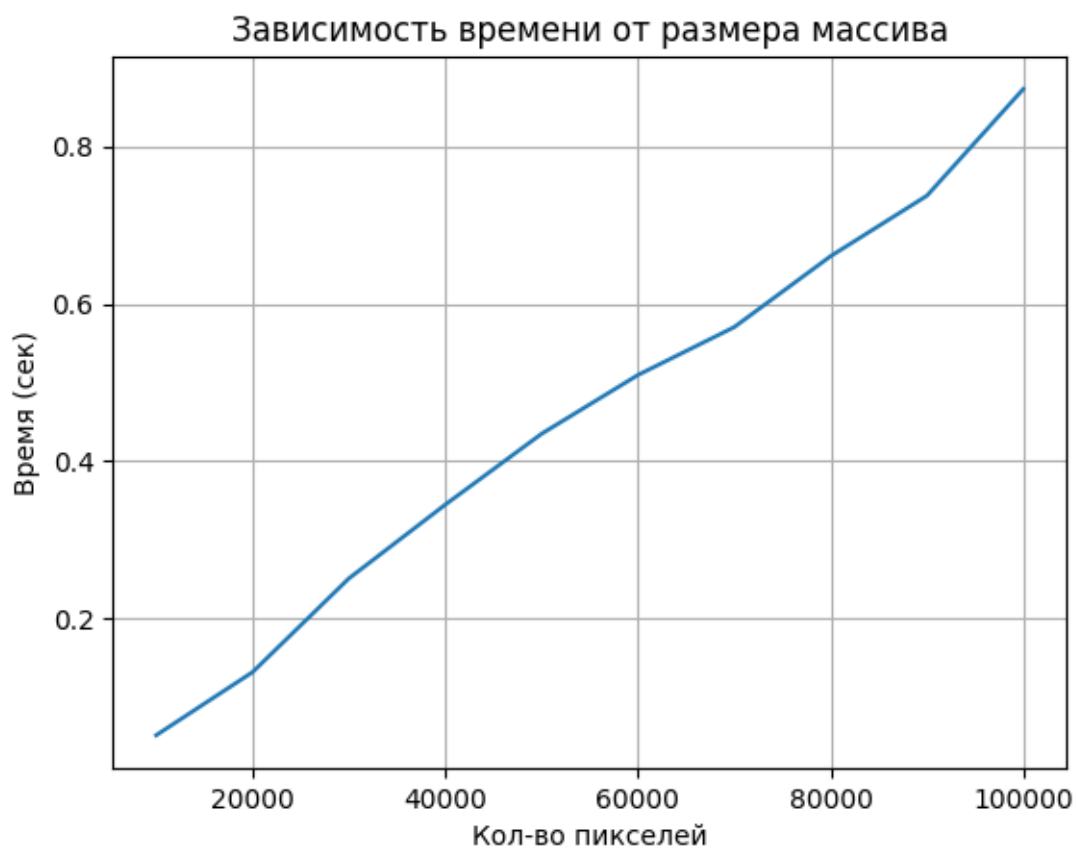


Рисунок 4.1 — График

Присутствует четкая линейная зависимость времени от размера входных данных. Значит, увеличив `fsize` от 1 до 2, время выполнения будет в 2 раза больше, что подтверждается результатами (4.2).

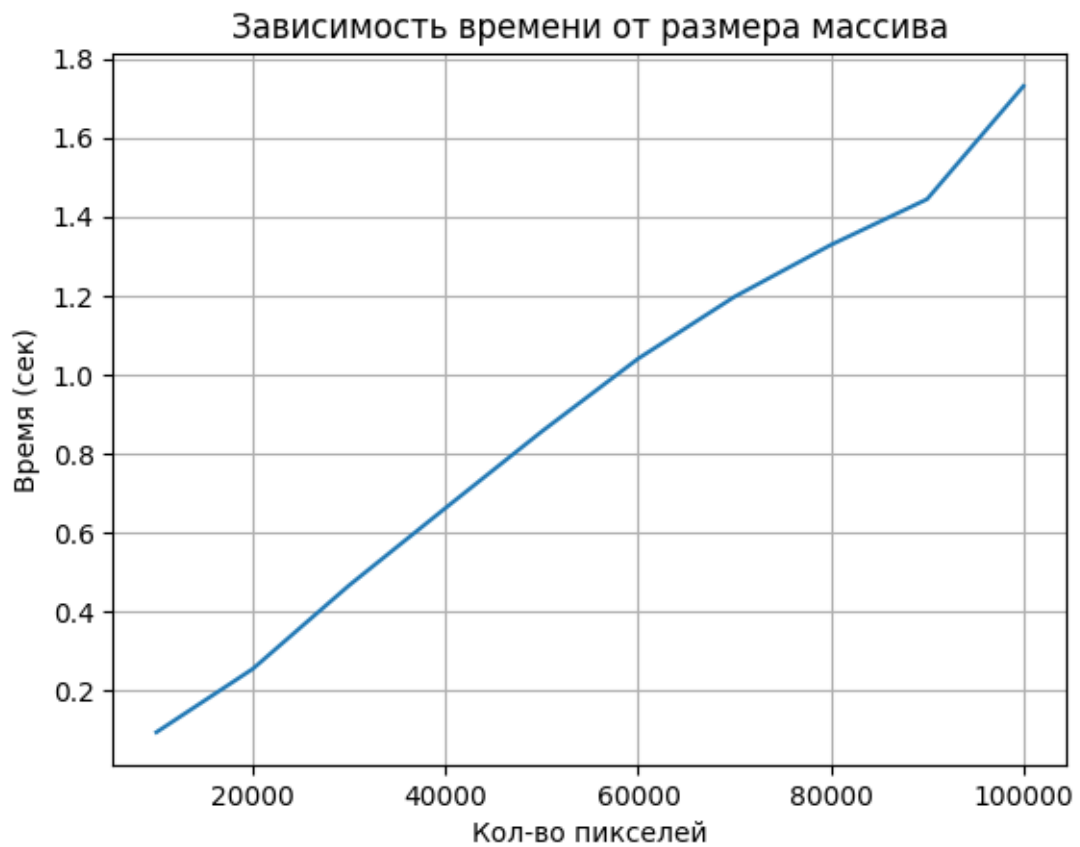


Рисунок 4.2 — График

Рассмотрим такое же дерево, но с параметрами `uniqueSize=20`, `fsize=1`. Получим следующую картину (4.3):

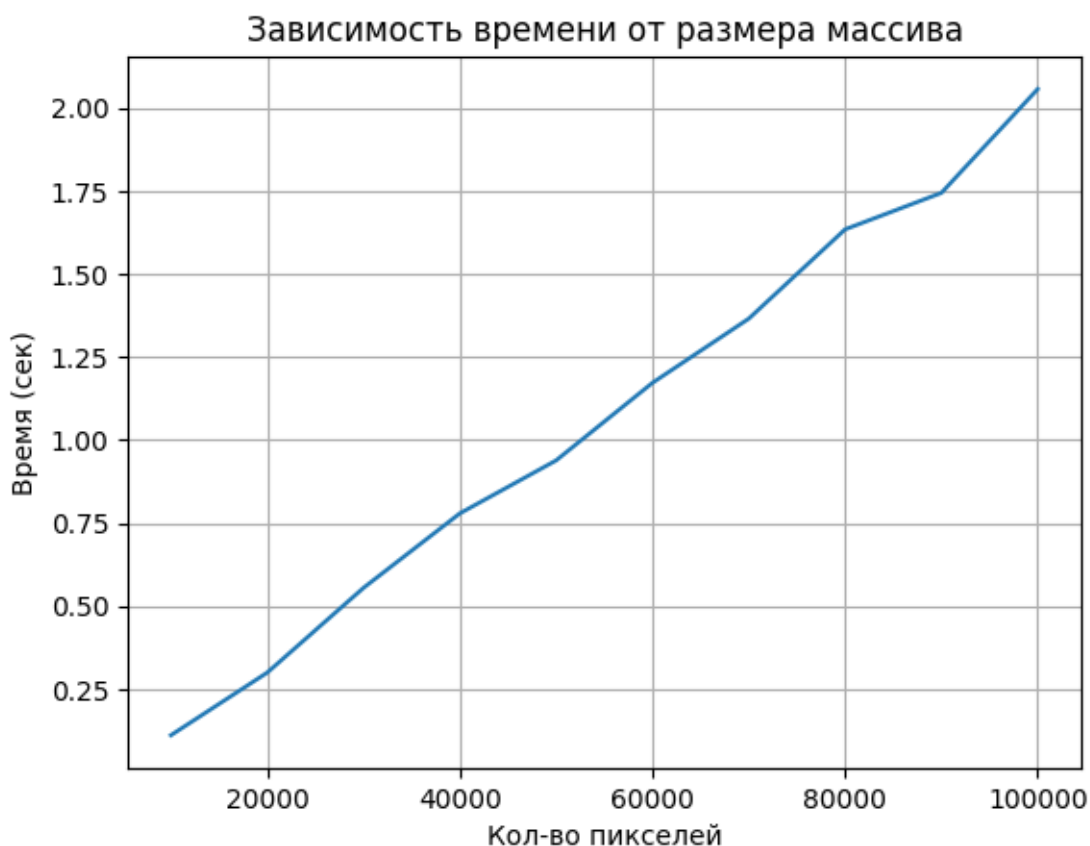


Рисунок 4.3 — График

В итоге подтвердилось, что время обучения узла линейно зависит от количества рассматриваемых пикселей, количества рассматриваемых значений и количества рассматриваемых признаков, что соответствует ожиданиям.

4.6 Выводы по разделу

В ходе экспериментов были получены результаты, демонстрирующие влияние уменьшения данных на обучение алгоритмов. В некоторых случаях обучение алгоритмов на уменьшенных данных позволяет уместить пространственную информацию в более компактное представление, давая алгоритмам возможность обработать большее число пространственных признаков при одном и том же количестве обрабатываемых пикселей, что дает небольшое улучшение основных метрик. Также выявлялись и случаи, когда значения данных метрик (например, ассигасу) немного падали.

Однако, учитывая многократное ускорением тренировки алгоритмов, можно предположить, что это снижение критичным не является (но и это зависит от специфики поставленной задачи, в частности - от целевой метрики). В среднем, по метрике Gamma, при тренировке на уменьшенных изображениях она повышалась, что свидетельствует о возникших в результатах предсказания "отчужденных" пикселей, потенциально являющимися шумом. Вследствие этого, в дальнейшем алгоритмы требуется лучше адаптировать к алгоритму предобработки спутниковых сцен для минимизации побочного эффекта со стороны последнего.

Заключение

В результате проделанной работы были получены рабочие алгоритмы пространственных деревьев решений, которые были протестированы на спутниковых сценах в задаче бинарной классификации, и обобщенный алгоритм уменьшения многомерных изображений, предназначенный для ускорения уже существующих алгоритмов.

Были оценены предсказания алгоритмов с точки зрения основных метрик и пространственной метрики для определения подверженности алгоритмов к шуму в изображениях. Полученные результаты позволяют предполагать, что обобщенный алгоритм уменьшения изображений может быть успешно применен для обучения пространственных деревьев решений и давать результаты не хуже тех алгоритмов, которые были обучены на необработанных данных.

Для достижения успеха в дальнейших исследованиях необходимо сделать следующее:

- Реализовать алгоритмы частично/целиком на других языках программирования (С/С++/Java) с применением многопоточности для ускорения алгоритмов и более широкого их возможного применения при тестировании.

- Протестировать алгоритмы, предварительно подобрав гиперпараметры деревьев под алгоритм уменьшения изображений с помощью кросс-валидации для получения более полного представления о влиянии алгоритма уменьшения изображений на существующие пространственные деревья решений

- Реализовать алгоритмы пространственных деревьев так, чтобы было допустимо дообучать их; применить для дообучения пирамиду изображений, полученную с помощью алгоритма уменьшения изображений и оценить полученные результаты.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Jiang, Zhe*. Spatial Big Data Science / Zhe Jiang, Shashi Shekhar. — Springer, 2017.
2. *Szeliski, Richard*. Computer Vision: Algorithms and Applications / Richard Szeliski. — Springer, 2022.
3. *Richards, John A.* Remote Sensing Digital Image Analysis / John A. Richards. — Springer, 2013.
4. Rapid, Detail-Preserving Image Downscaling / Nicolas Weber, Michael Waechter, Sandra C. Amend et al. // *ACM Transactions on Graphics*. — 2016. — Vol. 35, no. 6.
5. *Breiman, Leo*. Classification and Regression Trees / Leo Breiman. — 1984.
6. *Gao, Jay*. Digital Analysis of Remotely Sensed Imagery, 1st Edition / Jay Gao. — The McGraw-Hill Companies, Inc., 2009.
7. *Criminisi, Antonio*. Decision Forests for Computer Vision and Medical Image Analysis / Antonio Criminisi, Jamie Shotton. — 2013.

Приложение А Иллюстрации

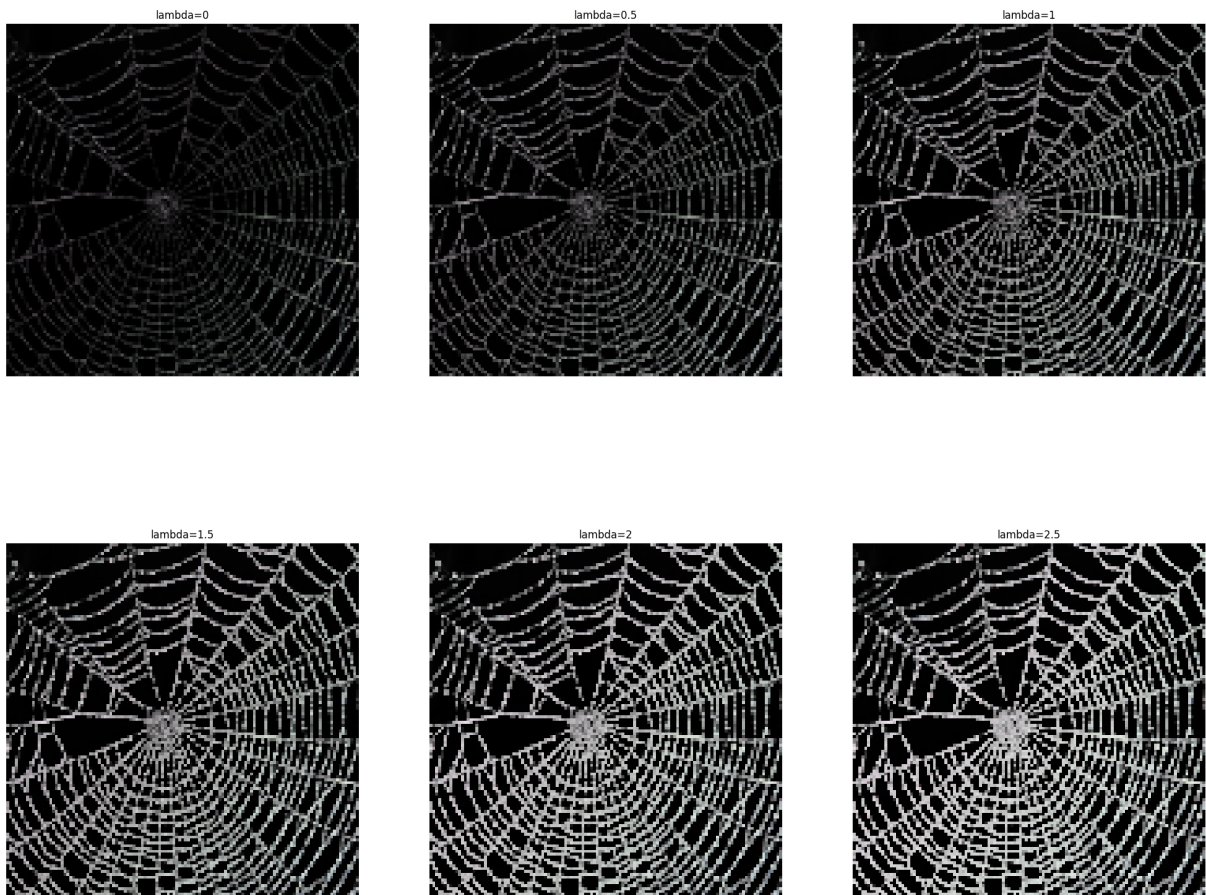


Рисунок А.1 — Пример уменьшения изображения с разными значениями λ

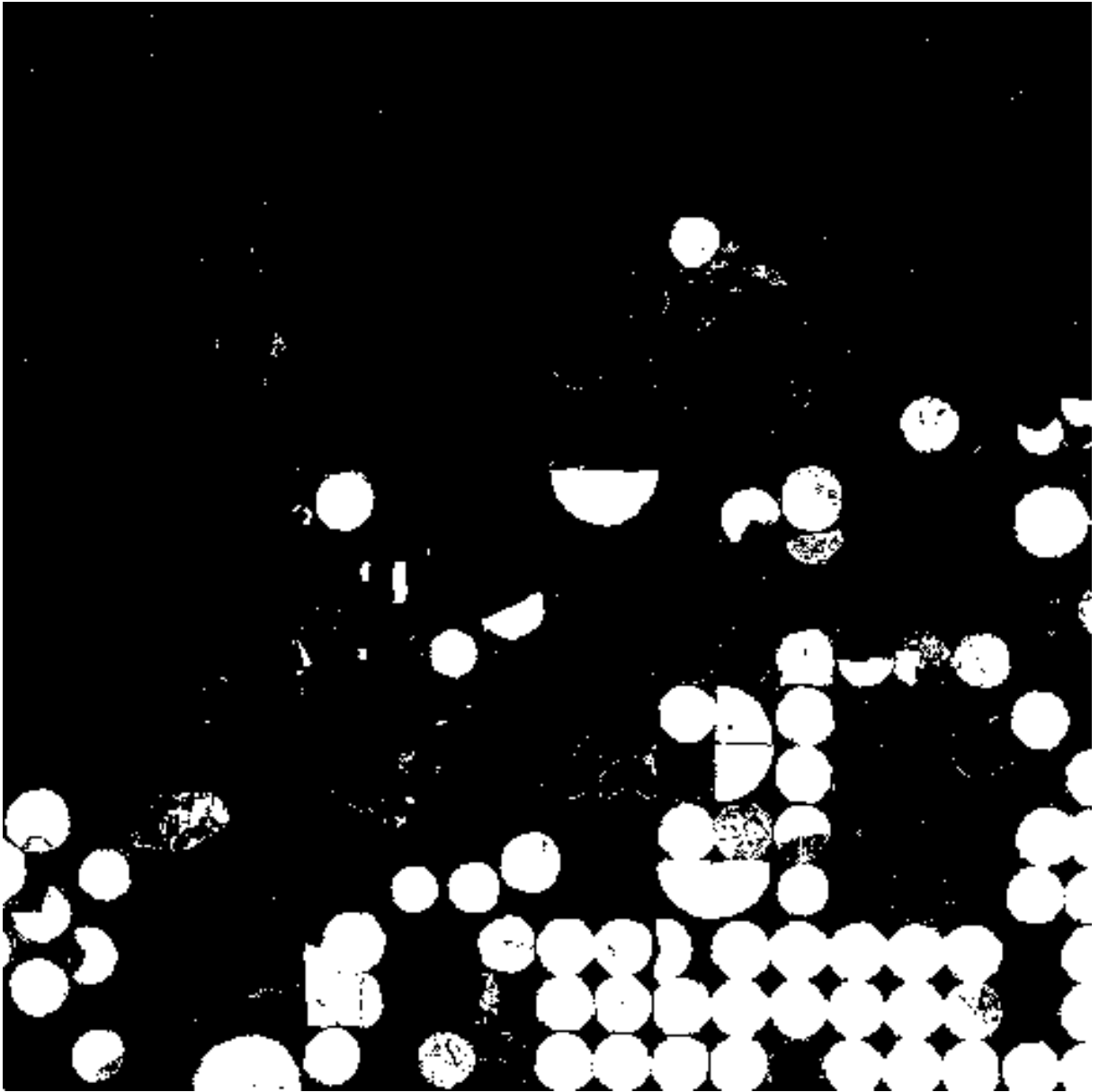


Рисунок А.2 — Истинные значения спутниковой сцены

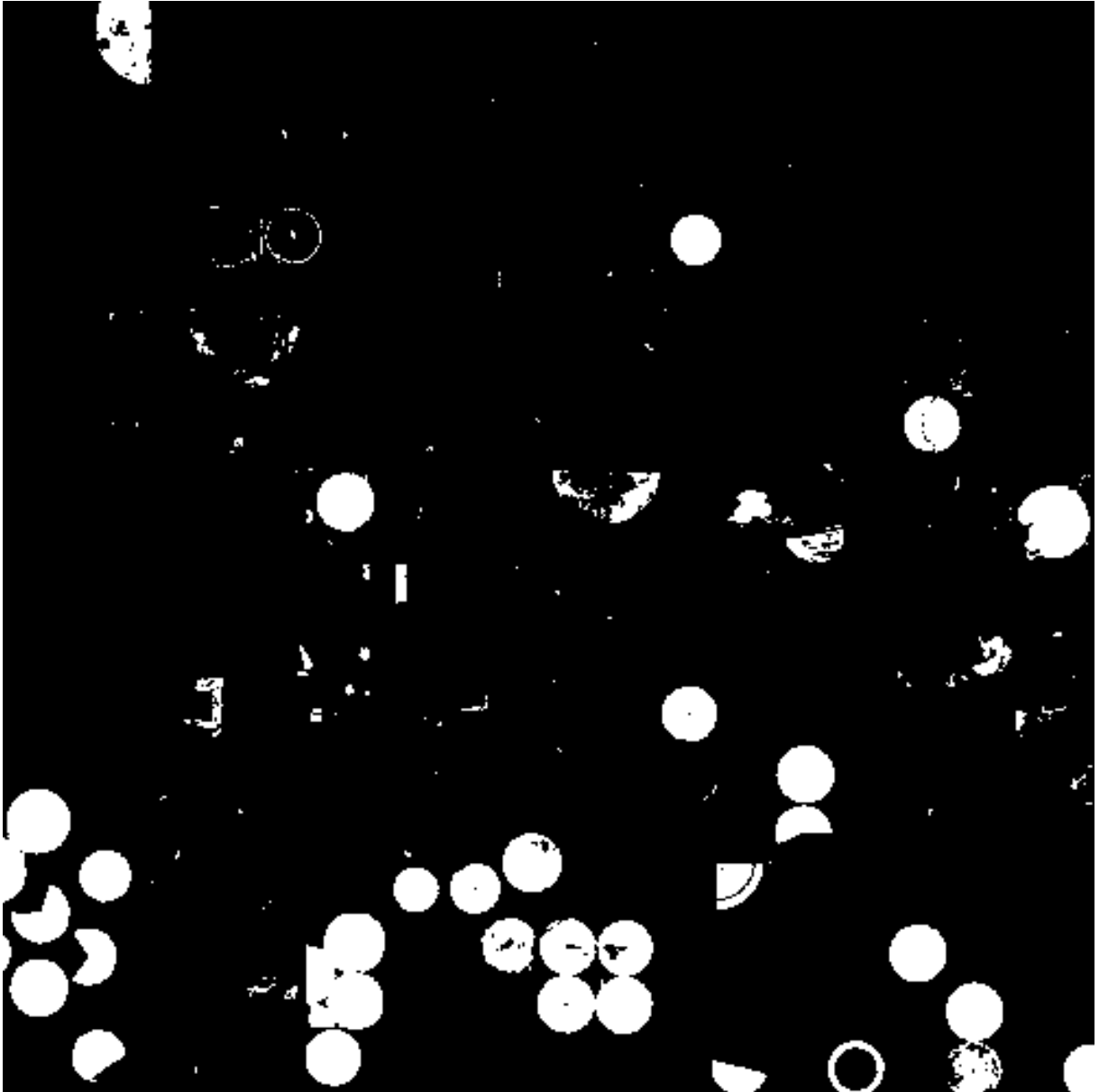


Рисунок А.3 — Предсказание алгоритма CSDTree, обученном на необработанных данных

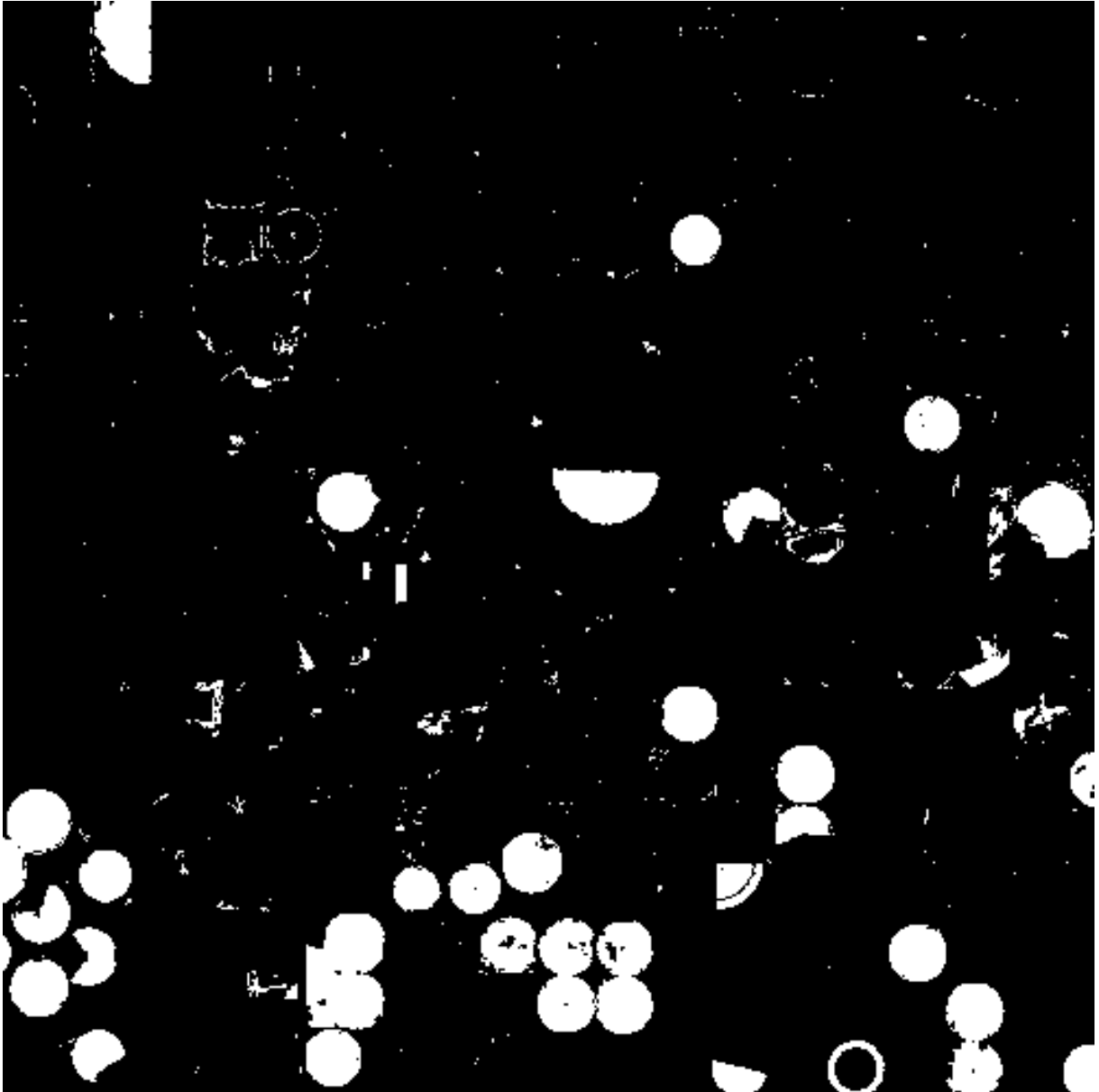


Рисунок А.4 — Предсказание алгоритма CSDTree, обученном на уменьшенных данных