**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Научный руководитель  доцент департамента  программной инженерии  факультета компьютерных наук,  канд. техн. наук  Родригес Залепинос Р.А.  **«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  2018 г. | УТВЕРЖДЕНО  Академический руководитель  образовательной программы  «Программная инженерия»  профессор департамента программной инженерии, канд. техн. наук  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** В.В. Шилов  **«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  2018 г. |

|  |  |
| --- | --- |
| **Подп. и дата** |  |
| **Инв. № дубл.** |  |
| **Взам. инв. №** |  |
| **Подп. и дата** |  |
| **Инв. № подл** |  |

**ПРОГРАММА ОБНАРУЖЕНИЯ ИЗМЕНЕНИЙ ЗЕМЛЕПОЛЬЗОВАНИЯ ПО МУЛЬТИСЕНОСОРНЫМ СПУТНИКОВЫМ ДАННЫМ**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.16 81 01-1-ЛУ**

|  |  |
| --- | --- |
|  | Исполнитель:  студент группы БПИ152  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** /А.А. Лукин/  **«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  2018 г. |

УТВЕРЖДЕН

RU.17701729.04.14 81 01-1-ЛУ

**ПРОГРАММА ОБНАРУЖЕНИЯ ИЗМЕНЕНИЙ ЗЕМЛЕПОЛЬЗОВАНИЯ ПО МУЛЬТИСЕНСОРНЫМ СПУТНИКОВЫМ ДАННЫМ**

|  |  |
| --- | --- |
| **Подп. и дата** |  |
| **Инв. № дубл.** |  |
| **Взам. инв. №** |  |
| **Подп. и дата** |  |
| **Инв. № подл** |  |

**Пояснительная записка**

**RU.17701729.04.16 81 01-1**

**Листов 33**

# **АННОТАЦИЯ**

В данном документе приведена пояснительная записка для «Программы обнаружения изменений землепользования по мультисенсорным спутниковым данным». Данная программа предназначена для обнаружения изменений землепользования по спутниковым снимкам.

Оформление программного документа «Пояснительная записка» произведено по требованиям ГОСТ 19.404-79 «Пояснительная записка. Требования к содержанию и оформлению» [1].

**Содержание**

[Аннотация 2](#_Toc482558782)

[1. Введение 4](#_Toc482558783)

[1.1. Наименование разработки 4](#_Toc482558784)

[1.2. Документы, на основании которых ведется разработка 4](#_Toc482558785)

[2. Назначение и область применения 5](#_Toc482558786)

[2.1. Назначение программного модуля 5](#_Toc482558787)

[2.2. Область применения программы 5](#_Toc482558788)

[2.3. Место подсистемы как компоненты сервиса **Ошибка! Закладка не определена.**](#_Toc482558789)

[3. Технически характеристики 6](#_Toc482558790)

[3.1. Постановка задачи для разработки программы 6](#_Toc482558791)

[3.2. Описание алгоритмов и функционирования подсистемы 6](#_Toc482558792)

[3.2.1. Генерация умных контрактов 6](#_Toc482558793)

[3.2.2. Компиляция сгенерированного умного контракта и его публикация в сеть Ethereum 8](#_Toc482558794)

[3.3. Описание и обоснование выбора метода организации входных и выходных данных 17](#_Toc482558795)

[3.3.1. Описание метода организации входных и выходных данных 17](#_Toc482558796)

[3.4. Описание и обоснование выбора состава технических и программных средств 18](#_Toc482558797)

[3.4.1. Состав технических и программных средств 18](#_Toc482558798)

[3.4.2. Обоснование выбора технических и программных средств 19](#_Toc482558799)

[4. Ожидаемые технико-экономические показатели 20](#_Toc482558800)

[4.1. Предполагаемая потребность 20](#_Toc482558801)

[4.2. Ориентировочная экономическая эффективность 20](#_Toc482558802)

[5. Список использованной литературы 21](#_Toc482558803)

[Приложение 1 Терминология 23](#_Toc482558804)

[Приложение 2 Описание и функциональное назначение файлов 24](#_Toc482558805)

[Приложение 2 Описание и функциональное назначение переменных и методов 25](#_Toc482558806)

# **Введение**

## **1.1. Наименование разработки**

Наименование программного продукта – «Программа обнаружения изменений землепользования по мультисенсорным спутниковым данным».

Наименование программного продукта на английском языке – «Program for Land-Use Change Detection using Multisensor Satellite Data».

**1.2. Документы, на основании которых ведется разработка**

Разработка программы ведется на основании приказа Национального исследовательского университета «Высшая Школа Экономики» № 2.3-02/1903-01 «Об изменении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук» от 19.03.2017.

# **Назначение и область применения**

## **2.1. Назначение программного модуля**

Программа предназначена для детектирования изменений землепользования на основе мультисенсорных спутниковых данных, предоставляемых спутниками программы Sentinel 2 Европейского космического агентства.

Виды землепользования должны быть представлены следующими классами:

1. Водные ресурсы.
2. Пахотные земли и пастбища.
3. Застроенные области.
4. Лесные массивы.

Программа должна визуализировать изменения землепользования по всем классам в совокупности с указанием видом изменения землепользования, а также по изменению каждого класса индивидуально.

Программа предоставляет возможность сохранения результатов обработки данных в виде гео-векторных фалов.

## **2.2. Область применения программы**

Программный продукт может применяться учеными в сферах экологии и урбанистики, а также в области учета землепользования государственными органами, занимающимися контролем пользования земельными участками.

# **Технические характеристики**

## **3.1. Постановка задачи для разработки программы**

Задачей разработки программы является создание программы для определения изменений землепользования на основе мультисенсорных спутниковых данных с использованием машины опорных векторов (SVM). Виды землепользования должны быть представлены следующими классами:

1. Водные ресурсы.
2. Пахотные земли и пастбища.
3. Застроенные области.
4. Лесные массивы.

Программа должна предоставлять доступ к загрузке мультисенсорных данных Sentinel 2 Европейского и предоставлять возможность преобразования загруженных данных до уровня Level 2 [2]. Программа должна предоставлять возможность пользователю переобучить или обучить на дополнительных данных классификатор SVM для определения изменений землепользования на основе сцены Sentinel 2 и векторных файлов Open Street Map. Программа должна визуализировать результаты обнаружения изменений по мультисенсорным спутниковым данных и предоставлять возможность сохранять результаты в векторном файле ERSI Shapefile.

## **3.2. Описание алгоритмов и функционирования программы**

### **3.2.1. Поиск и загрузка данных Sentinel 2**

Поиск и загрузка мультисенсорных данных Sentinel 2 осуществляется с помощью Copernicus Open Access Hub API [3], предоставляющий поиск и загрузку данных c помощью технологии OpenSearch [4], предоставляющей результаты поиска c помощью протокола Atom [5]. Для работы с технологией OpenSearch в программе используется библиотека Apache Abdera [6].

#### **3.2.1.1. Ввод и обработка параметров поиска данных Sentinel 2**

Модуль программы для поиска и загрузки данных Sentinel 2 принимает от пользователя в качестве параметров поиска данных некоторые или все значения, указанные в списке ниже:

* Дата начала периода времени, в течении которого были зафиксированы и созданы данные спутниками ESA Sentinel 2.
* Дата завершения периода времени, в течении которого были зафиксированы и созданы данные спутниками ESA Sentinel 2.
* Область земной поверхности, на которой были зафиксированы мультисенсорные данные спутников ESA Sentinel 2.
* Максимальный процент покрытия спутниковых данных облаками.

Даты начала и завершения периода фиксирования данных, а также процент максимального покрытия мультисенсорного снимка облаками указываются пользователем в графическом интерфейсе программы с помощью предназначенных для этого контролеров. Указанные пользователем даты преобразовываются в формат ISO8601 [7].

Область земной поверхности, на котором производиться поиск данных указывается в специальном контроллере, содержащий интерактивную карту Земли, реализованной на языке JavaScript с использованием библиотеки Leaflet [8]. При начале процесса поиска данных, при указании области интереса пользователя в интерактивной карте пользователем, данных об полигоне области передаются в программу в формате WKT [9]в системе координат WGS84 [10].

#### **3.2.1.2. Создание запроса поиска данных**

После ввода необходимых пользователю параметров поиска данных Sentinel 2, программа создает URL для запроса методом GET с указанием в параметрах запроса необходимые для поиска параметры, согласно спецификации ESA Copernicus Open Access API [3]. Ниже представленные примеры создаваемых адресов для запроса данных спутников Sentic периодом фиксирования поверхности земли с 1 июля 2017 года по 31 августа 2017 года, с максимальным процентом покрытия сцены облаками 10 % и пересекающейся с областью, ограниченной областью Средиземного моря:

[*https://scihub.copernicus.eu/dhus/search?q=platformname:Sentinel-2*](https://scihub.copernicus.eu/dhus/search?q=platformname:Sentinel-2) *AND beginposition:[ 2017-07-01T00:00:00.000Z TO 2017-08-21T00:00:00.000Z] AND footprint:"Intersects(POLYGON((-4.53 29.85, 26.75 29.85, 26.75 46.80,-4.53 46.80,-4.53 29.85)))" AND cloudcoverpercentage:[0 TO 10]*

Рис.1. Пример запроса данных

Запрос поиска данных производиться с помощью OpenSearch клиента, созданного с помощью библиотеки Apache Abdera [6].

#### **3.2.1.3. Получение, обработка результата запроса данных и загрузка данных Sentinel 2**

При успешном запросе поиска данных спутников ESA Sentinel 2 в качестве ответа с серверов ESA приход данные в формате Atom. Данные ответа сервера обрабатываются клиентом OpenSearch и преобразовываются в коллекцию элементов описания данных, из которых извлекается следующая информация:

1. Название спутниковых данных
2. Информация о дате фиксирования данных
3. Размер загружаемого файла
4. Ссылка на quick view найденных данных
5. Ссылка загрузки данных

Программа загружается изображение quick view найденных спутниковых снимков для наглядной демонстрации пользователю полученных данных, а также демонстрирует остальную информацию пользователю.

Загрузка данных ESA Sentinel 2 из списка результатов поиска происходить при переходе по ссылке загрузки данных с помощью программы загрузки файлов из Интернета, установленного на ПК пользователя.

#### **3.2.1.4. Диаграмма модуля программы для загрузки и поиска данных Sentinel 2**

На Рис.2. представлена диаграмма работы программы, предназначенного для поиска и загрузки мультисенсорных данных спутников Sentinel-2 Европейского космического агентства.

.

### **3.2.2. Преобразование данных Sentinel 2 Level 1C к Sentinel 2 Level 2A**

Согласно спецификации данных Sentinel 2 «Sentinel 2 Product Specification» [2] мультиспектральные спутниковые снимки, создаваемые спутниками Sentinel 2, имеют несколько типов данных, характеристики которых представлены в таблице № 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип** | **Уровень обработки** | **Описание** | **Целевые пользователи** |
| S2HKTM | N/A | Телеметрия космического спутника Sentinel 2 в формате Transfer Frame. | Полетные операции Sentinel 2. |
| S2MSI0 | 0 | Сжатые необработанные мультиспектральные (MSI) данные в формате ISP. | Эксперты MSI и внутренние пользователи ESA. |
| S2MSI1A | 1A | Несжатые необработанные мультиспектральные (MSI) данные с файлами спектральных полос. | Не распространяется среди внешних пользователей. |
| S2MSI1B | 1B | Радиометрически скорректированные (откалиброванные) данные изображения MSI с файлами спектральными полосами. | Конечные пользователи-эксперты. |
| S2MSI1C | 1C | Ортофототрансформированная и UTM-геокодированная отражательная способность верхней атмосферы с субпиксельной мультиспектральной и многорежимной регистрацией. | Общие конечные пользователи. |
| S2MSI2A | 2A | Ортофототрансформированная и UTM-геокодированная мультиспектральная отражательная способность нижней части атмосферы. Дополнительные выходы - карта аэрозольной оптической толщины (AOT), карта водяного пара (WV), карта классификации сцены вместе с данными показателей качества. |

Таблица №1. Уровни данных спутников ESA Sentinel-2

В целях определения изменения землепользования программа нуждается в данных отражения поверхности Земле, то есть отражения нижних областей атмосферы планеты (Bottom Of Atmosphere – BOA) [11] с соответствующей атмосферной коррекцией, поэтому программа должна работать согласно таблице №1 с данными S2MSI2A или Level 2. Однако данные уровня Level 2A не регулярно публикуются и доступны с помощью Copernicus Open Access Hub. [3] связи с чем возникает необходимость предоставлять пользователю возможность преобразовывать данные Level 1C, полученные с помощью Copernicus Open Access Hub API, до уровня Level 2A.

ESA рекомендует [12] преобразовывать данные Sentinel Level 1C к данным Level 2A c помощью утилиты Sen2Cor [12] для программной среды Anaconda Python 2.7 [13] , которая имеет интерфейс командой строки. Разработанная программа преобразовывает данные c уровня Level 1С к уровню Level 2A при помощи запуска утилиты Sen2Cor с указанием выбранных пользователем параметров преобразования, с отображением состояния преобразования, а также возможностью прервать процесс преображения данных. Модель предоставляет пользователю возможностью выбора следующих параметров преобразования данных:

1. Преобразование спутниковых данных с резолюцией 10 х 10 метров.
2. Преобразование спутниковых данных с резолюцией 20 х 20 метров.
3. Преобразование спутниковых данных с резолюцией 60 х 60 метров.
4. Преобразование спутниковых данных с резолюциями 10 х 10 метров, 20 х 20 метров и 60 х 60 метров.

Таким образом, модуль преобразования мультиспектральных спутниковых данных Sentinel 2 Level 1C к данным c представлением отражения света Sentinel 2 Level 2A представляет собой обертку графического интерфейса Java для управления утилитой Sen2Cor, выполняющаяся в среде Anaconda.

### **3.2.3. Работа с мультисенсорными сценами Sentinel-2 Level 2A**

#### **3.2.3.1. Мультисенсорные данные Sentinel-2 Level-2A**

Согласно спецификации данных Sentinel-2 [11] Загруженная сцена представляет собой одну или несколько гранул в разрешениях 60 x 60 метров, 20 х 20 метров, 10 х 10 метров, содержащие в себе файлы формате JP2000 с геопривязкой, каждый из которых содержит один канал со значениями отражения света определенного диапазона света земной поверхностью. На Рис.2 представлена диаграмма соответствия названия канала с фиксированным диапазоном данных и максимальную резолюцию данного файла.

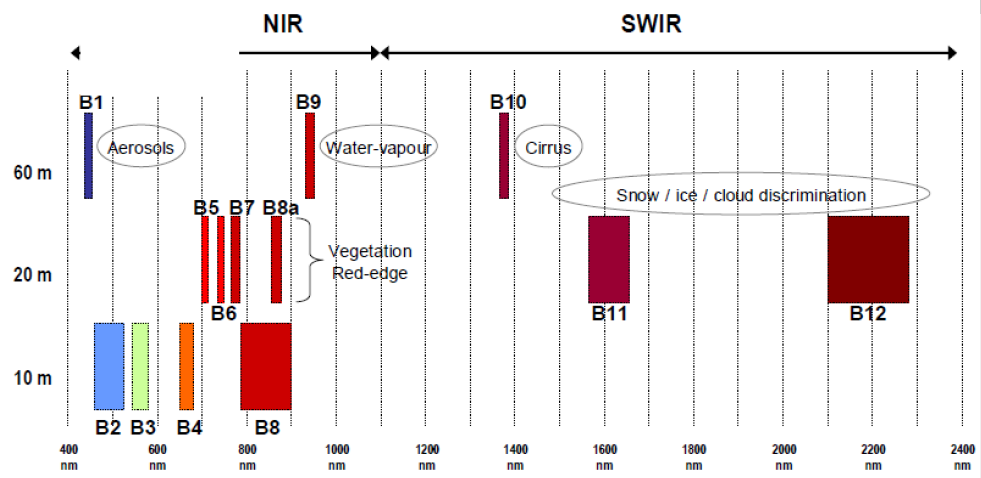


Рис. 2 Диаграмма соответствия названия канала с фиксированным диапазоном данных и максимальную резолюцию данного файла

Также вместе с данными каналами предоставляются следующие маски:

* Маска вероятности облаков
* Маска вероятности облаков
* Маска первичной классификации земной поверхности
* Маска верхнего слоя атмосферы
* Маска плохих пикселей
* Маска первичной классификации водной поверхности

Перечисленные выше маски предоставляются исключительно с величиной резолюции 60х60 и 20х20 метров.

#### **3.2.3.2. Извлечение данных гео-привязки и чтения файлов в формате JP2K**

Данный растровых файлов каналов считываются с помощью библиотеки GeoTools [14], которая требует при чтении фалов в формате .JP2K фалов файла системы координат гео-привязки приложения в формате .prj и world-файла, в котором указываются граничные координаты в указанной проекции считываемого растрового изображения.

Информация о системе координат и world-файла извлекаются из фала растрового изображения с помощью библиотеки абстракции гео-пространственных данных GDAL (Geospatial Data Abstraction Library) [15] и записываются в файлы, имеющие название, соответствующее названию файла растрового изображения, и расширения .prj для системы координат растрового изображения и .j2w для world-файла, после чего данные успешно считываются с помощью библиотеки GeoTools.

### **3.2.3. Модуль обучения классификатора SVM**

#### **3.2.3.1. Классификация мультисенсорных спутниковых данных Sentinel-2**

Согласно базовой статье для разработки программного продукта [16], данные мультиспектральных снимков Sentinel-2 за определенную дату фиксирования земной поверхности классифицируются с помощью классификатора машины опорных векторов (Support Vector Machine) [17], классифицирующий мультиспектральные данные на следующие классы:

1. Водные ресурсы
2. Агрикультура
3. Застроенные области
4. Лесные массивы

Классификатор SVM разделяет гиперпространство значений мультисенсорного снимка, принимая в качестве входных данный вектор, составленный из значений пикселей, соответствующих значениям пикселей с одинаковым географическим положением из считанных данных и величиной пикселя, соответствующей интенсивности отражения света земной поверхности в определенном диапазоне, соответствующей файлу канала данных.

В качестве классификатора SVM использовалась библиотека Statistical Machine Intelligence and Learning Engine [18], реализованной на языке Java.

В связи с тем, что миссия ESA Sentinel-2 [19] включает в себя 2 спутника: ESA Sentinel-2 Instrument A и ESA Sentinel-2 Instrument B, для классификации мультиспектральных данных для данных, зафиксированных каждым спутников, создается 2 классификатора SVM.

#### **3.2.3.2. Выбор набора каналов для классификации вида землепользования**

Для определения состава входного для классификатора SVM вектора признаков был произведено обучение классификатора на множестве всех комбинаций каналов данных Sentinel-2 Level 2A. Задачей данного эксперимента было выявление размера и состава вектора признаков для получение максимальной точности классификации вида землепользования.

В качестве элементов входного вектора признаков не использовались данные каналов AOT и TCI гранул Sentinel-2 Level 2A в связи с тем, что значения пикселей файла AOT (Aerosol Optical Thickness) соответствуют оптической толщине аэрозоля в атмосфере, что не связано с данными пикселей отражения света нижнем слоем атмосферы, а данные TCI (True Color Image) дублируют данные B02, B03, B04 в виде трехканального RGB-изображения в формате JP2K.

По результатам проведения измерений значений точностей классификаторов, все наборы векторов признаков длины менее 12 каналов получили точность классификации менее 80%. На рис.3 представлены значения точности классификации для наборов длины от 12 файлов.

Рис.3. Точность обучения в зависимости от множества входных каналов

Согласно рис.3 максимальная точность классификации данных Sentinel-2 Level 2A достигается на векторе входных признаков, состоящих из всех каналов данных.

#### **3.2.3.3. Обучение классификаторов SVM для данных Sentinel-2**

Классификатор SVM требует обучения для классификации данных. В разработанном программной продукте представляется возможность обучения классификатора для каждого из спутников миссии ESA Sentinel-2.

Обучение классификатора SVM производиться с помощью мультиспектральных снимков Sentinel-2 Level 2A и векторных данных Open Street Map за период фиксации данных Sentinel-2 в формате ERSI Shapefile. Пользователь может использовать в качестве векторных данных собственные векторные файлы, однако для успешного обучения они должны использовать типы тегов, используемые Open Street Map в качестве атрибутов, для описания класса землепользования, которые демонстрируют включенные в них векторные данные.

Процесс обучения включает в себя следующие этапы:

1. Чтение, объединение и растеризация векторных данных Open Street Map
2. Чтение мультиспектральных данных Sentinel-2 Level 2A и информации об спутнике, зафиксировавшего эти данные
3. Чтение данных о покрытии облаками и снегом, а также испорченных пикселей данных Sentinel-2 Level 2A и исключение данных пикселей из мультиспектральных данных для обучения модели SVM
4. Обучение модели SVM по предоставленным данным с использование алгоритма greed search для поиска гиперпараметров классификатора SVM и кросс-валидацией

В следующих пунктах подробно описаны этапы обучения классификатора SVM.

#### **3.2.3.3.1. Выбор областей для обучения и векторизация данных Open Street Map**

Классификатор SVM требует обучения для классификации данных. В разработанном программной продукте представляется возможность обучения классификатора для каждого из спутников миссии ESA Sentinel-2. Для выбора областей землепользования и соответствующего ему класса землепользования используются векторные данные в формате ERSI Shapefile. В таблице №2 представлены значения тегов данных для классов землепользования:

|  |  |
| --- | --- |
| **Класс землепользования** | **Тег OpenStreetMap** |
| Водные ресурсы | * “water” |
| Агрикультура | * "farmland", * "meadow", * "orchard", * "plant\_nursery", * "vineyard", * "farm", * "allotments", * "farmyard", * "greenhouse\_horticulture" |
| Застроенные области | * "commercial", * "garages", * "industrial", * "religious", * "residential", * "retail", * "school", * "brownfield", * "construction", * "landfill", * "quarry", * "salt\_pond" |
| Лесные массивы | * "forest", * "wood" |

Табл. 2 Соответствие тегов OpenStreetMap классам землепользования

После выборки полигонов землепользования, данная коллекция геометрий преобразовывается в растровое изображение с разрешением с использованием библиотеки Java Advanced Imaging. В качестве значения пикселя полученного растрового изображения используется двоичная маска, где i-й блок соответствует номеру классу землепользования. Например, значение 2 в двоичной записи 0010 представляет пиксель класса агрикультуры.

#### **3.2.3.3.2. Извлечение из спутниковых данных пикселей, покрытых облаками или снегом и испорченных пикселей**

Данные Sentinel-2 Level 2A включают в себя растровые файлы в формате JP2K с масками вероятностей облаков, снега и испорченных пикселей. Пиксели масок снега и облаков со значением вероятности более 90% извлекаются каждого файла полос мультисенсорных данных с помощью библиотеки Java Advanced Imaging с установкой значений пикселей в файлах на значение -1, для того, чтобы не обучать классификатор землепользования SVM на пикселях со значениями интенсивности отражения света облаками в различных диапазонах.

#### **3.2.3.3.3. Подбор параметров по сетке и кросс-валидация при обучении классификаторов SVM**

Для преобразования

#### **3.2.3.3. Импорт и экспорт классификаторов SVM**

После успешного обучения объект классификатора SVM сериализуется в файл с расширением .svm, соответствующий виду спутнику, зафиксировавшего данные. В связи с тем, что значение пикселей каналов данных для различных классов могут различаться в связи с различными периодами года, климатическими и культурными особенностями в различных регионах планеты, пользователю предоставляется возможность запуска обучения классификаторов, а также их экспорт и импорт в программу определения изменений землепользования на основе мультисенсорных спутниковых данных.

### **3.2.4. Определение изменений землепользования по мультисенсорным спутниковым данным Sentinel-2 с использованием классификатора SVM**

Определение изменение землепользования производиться на основе двух гранул данных Sentinel-2 Level 2A с различной датой создания снимков спутниками. Рекомендуются использоваться данные, соответствующие одинаковым сезоном создания снимков спутниками Sentinel-2, и классификатором SVM, обученном на территории с не сильно различающимися климатическими особенностями и датой фиксации, соответствующей данным, на основе которых производиться определение изменений землепользования.

Процесс детектирования изменений землепользования на основе мультисенсорных спутниковых данных включает в себя следующие этапы:

1. Считывание и обрезка данных Sentinel-2 Level 2A
2. Обрезка считанных данных на основе области интереса пользователя или пересечении покрытия территорий выбранными спутниковыми данными
3. Извлечение из считанных данных пикселей облаков, снега и отсутствующих значений
4. Классификация считанных данных классификатором SVM, соответствующего спутника, создавшего снимок
5. Проверка на пиксели классифицированных данных, окруженных пикселями другого единственного класса
6. Векторизация классифицированных растровых изображений
7. Определение территорий с изменившимся видом землепользования
8. Расчет площадей изменений землепользования по классам
9. Демонстрация в виде векторных данных и данных в формате WKT

#### **3.2.4.1. Выбор области интереса для классификации снимков**

Пользователь может выбрать область интереса для классификации данных, в таком случае входные изображения Sentinel-2 Level-2A обрезаются по указанной в векторных файлах геометриях с помощью библиотеки Java Advanced Imaging.

#### **3.2.4.2. Проверка пикселей**

После успешной классификации входных данных, программа проверяет растровое изображение на пиксели одного класса, окруженного пикселями одного другого класса. В таком случае, окруженному пикселю присваивается значение класса соседних пикселей.

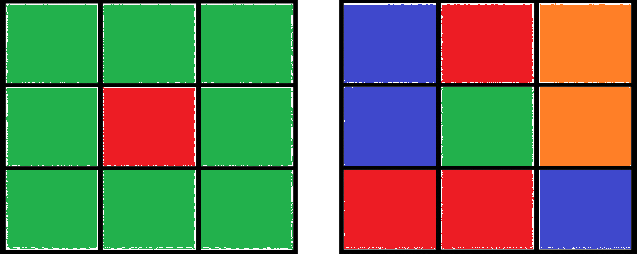


Рис 4. Примеры пикселя для замены и валидного пикселя

#### **3.2.4.2. Векторизация классифицированных данных**

После успешной классификации и проверки пикселей, данные растровые изображения переводятся в векторный формат, в котором объединяются в полигоны пиксели, имеющие одинаковые классы. Векторизация данных производиться с помощью библиотеки GeoTools.

#### **3.2.4.3. Определение территорий с изменением вида землепользования**

Векторизованные данные проверяются на изменение вида землепользования путем перебора геометрий векторов файла «до» с определение пересечений с геометриями второго векторной коллекции. На основе этих пересечений создается новая коллекция векторных данных, где в качестве геометрии указывается область пересечения, и качестве атрибутов класс землепользования «до» и «после».

#### **3.2.4.4. Расчет площади изменений**

Итоговая векторная коллекция используется для определения площади изменений землепользования. Данный геометрии векторов и классов землепользования «до» и «после» отправляются в таблицу базы данных PostgreSQL c установленным расширением PostGIS. Данные агитируются по классам «до» и «после», после чего для агрегированных данных возвращаются значения площадей для измененный и нет классов землепользования. После расчета, данные удаляются из базы данных PostGIS.

#### **3.2.4.4. Демонстрация в виде векторных данных и данных в формате WKT**

После расчета площадей изменений, результат детектирования отображается в интерфейсе пользователя в виде полигонов на карте LeafletJS, в виде записи геометрий в формате WKT, а также отображаются значения площадей классов землепользования.

#### **3.2.4.4. Сохранение и повторное открытие файла результата определения изменения землепользования**

Пользователь может сохранить и открыть позже результат определения изменения землепользования. При сохранении результата, на компьютере пользователя сериализуется файла класса определения землепользования, после чего пользователь может открыть данный файл в программе и посмотреть результат изменения землепользования.

#### **3.2.4.4. Сохранение результата в формате ERSI Shapefile**

Пользователь может сохранить файл результата классификации в формате ERSI Shapefile, где в качестве атрибутов указываются классы землепользования «до» и «после», для последующей работы с экспортированными данными в GIS-средствах. Создание файлов ERSI Shapefile происходить с помощью библиотеки GeoTools.

## **3.3. Описание и обоснование выбора метода организации входных и выходных данных**

### **3.3.1. Описание метода организации входных и выходных данных**

#### **3.3.1.1. Модуль загрузки данных Sentinel-2**

Входные данные вводятся в соответствующие элементы ввода формы:

1. Логин и пароль.
2. Выбор дат фиксирования данных вводятся с помощью контролера календаря.
3. Выбор максимальной облачности вводиться в виде числа в соответствующий контроллер.
4. Выбор области производится путем рисования области на карте LeafletJS

Программа демонстрирует пользователю информацию о найденных данных и начинает загрузку по приходу по ссылке загрузки данных.

#### **3.3.1.2. Модуль преобразования данных**

Для преобразования данных с уровня Level 1C к уровню Level 2A пользователь выбирает необходимый файл загруженных данных и величину резолюции итоговых данных.

Преобразованные данные создаются в той-же директории, где находятся данные Level 1C.

#### **3.3.1.3. Модуль обучения классификаторов SVM**

Для обучения модели SVM пользователю нужно выбрать следующие файлы для обучения:

1. Папку с данными Sentinel-2 Level 2A, по которым будет происходить обучения
2. Выбрать необходимую гранулу и резолюцию данных
3. Папку с векторными файлами OpenStreetMap в формате ERSI Shapefile

Послу успешного обучения модели SVM, соответствующего спутнику, создавшего данные сериализуется в советующий файл и доступны для экспорта и импорта.

#### **3.3.1.4. Модуль определение изменения землепользования**

Для определения изменения землепользования по мультисенсорным спутниковым данным пользователь должен указать путь до 2-х данных Sentinel-2 Level 2A, необходимые гранулы и резолюцию. Опционально пользователь может указать путь на файлы в формате ERSI Shapefile для того, чтобы не производить классификацию данных и определение изменения землепользования за пределами геометрий, указанных в данном файле.

Модуль предоставляет пользователю сохранить результат определения изменения землепользования в виде сериализованного объекта для последующего чтения результата в программном продукте.

Также в программе предусмотрена возможность сохранения результата в формате ERSI Shapefile для последующей работы с ним в GIS-средствах, т.е. формат ERSI Shapefile является одним из самых популярных форматов хранения векторных гео-данных.

## **3.4. Описание и обоснование выбора состава технических и программных средств**

### **3.4.1. Состав технических и программных средств**

Программой предъявляются следующие требования к составу и параметрам технических средств:

1. Процессор с тактовой частотой не менее 2.3ГГц
2. Оперативная память не менее 4Гб
3. Жесткий диск со свободным объемом не менее 20Гб
4. Монитор
5. Клавиатура
6. Мышь
7. Стабильное подключение к сети Интернет

Программой предъявляются следующие требования к информационной и программной совместимости персональных компьютеров пользователей:

1. Операционная система Windows 7, Windows 8, Windows 8.1, Windows 10 или более поздней версии.
2. Программное обеспечение Java 8 JRE или выше.
3. Дистрибутив Anaconda c версией Python 2.7.
4. Установленная база данных PostgreSQL с расширением PostGIS
5. Установления библиотека Geospatial Data Abstraction Library (GDAL/OGR) версии 2.0.1

## **3.4.2.** **Обоснование выбора технических и программных средств**

Растровые данные Sentinel-2 занимают в среднем 600Мб дискового пространства одной гранулой, в связи с чем компьютер пользователя должен обладать большим количество дискового пространства и оперативной памяти.

В качестве языка программирования был выбран язык Java, т.к. для данного языка существует довольно развитая библиотека GeoTools для работы с гео-данными и Java Advanced Imaging для работы с растровой арифметикой.

Программа требует установленного дистрибутива Anaconda, т.к. утилита Sen2Cor, требуемая для преобразования данных Sentinel-2 Level 1C к Sentinel-2 Level 2A исполняется в данной среде, а также библиотека GDAL требует Python 2.7.

Программа использует библиотеку GDAL для чтения информации о загруженные растровых данных и извлечения их системы координат и геопривязки, т.к. данная библиотека является самой широко применяемой для данных целей.

# **Ожидаемые технико-экономические показатели**

## **4.1. Предполагаемая потребность**

Программный продукт может применяться учеными в сферах экологии и урбанистики, а также в области учета землепользования государственными органами, занимающимися контролем пользования земельными участками.

## **4.2. Ориентировочная экономическая эффективность**

Аналогов данного продукта выявлено не было. Преимущество решения заключается в детектировании изменений землепользования по мультисенсорным спутниковым данным ESA Sentinel 2, которые находятся в открытом доступе для всех пользователей, а так в возможности загрузить данные .непосредственно из программы.

# **Список использованной литературы**

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ Р 7.02-2006 Консервация документов на компакт-дисках. Общие требования. – М.: ИПК Издательство стандартов, 2006
11. ГОСТ 18300-87 Спирт этиловый ректификованный технический. Технические условия. – М.:ИПК Издательство стандартов, 1997
12. ГОСТ 9805-84. Спирт изопропиловый. Технические условия. – М.: ИПК Издательство стандартов, 1984.
13. ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001
14. Google Chrome Системные требования // Google URL: https://support.google.com/chrome/answer/95346?co=GENIE.Platform%3DDesktop&hl=ru (дата обращения: 20.04.2017).
15. Ethereum JavaScript API // Ethereum URL: https://github.com/ethereum/wiki/wiki/JavaScript-API (дата обращения: 9.05.2017)
16. Solidity - Solidity.0.4.12 documentation // Solidity URL: http://solidity.readthedocs.io (дата обращения: 9.05.2017).
17. Docs Node.js // Node.js URL: https://nodejs.org/en/docs/ (дата обращения: 9.05.2017).
18. White Paper Ethereum // Ethereum URL: https://github.com/ethereum/wiki/wiki/White-Paper (дата обращения: 9.04.2017).
19. ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER // Yellow Paper URL: yellowpaper.io (дата обращения: 14.04.2017).
20. The Keccak sponge function family // The Keccak URL: http://keccak.noekeon.org/ (дата обращения: 9.05.2017).
21. Descriptions of SHA-256, SHA-384, and SHA-512 // NIST URL: http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf (дата обращения: 9.05.2017).
22. Алгоритм RIPEMD-160 // Solutionmes URL: http://solutionmes.wikidot.com/crypto-ripemd (дата обращения: 9.05.2017).

# **Приложение 1 Терминология**

**Блокчейн (Цепочка блоков транзакций) -**  выстроенная по определённым правилам цепочка из формируемых блоков транзакций.

**Блок транзакций -** специальная структура для записи группы транзакций в системе Биткойн и аналогичных ей.

**Автор проекта** – лицо, отправляющее проект в сервис организации экспертной деятельной для рецензирования экспертами.

**Dapp –** распределенное приложение сети Ethereum, представляющее собой систему умных контрактов и пользовательского интерфейса для работы с этими контрактами.

**Ether –** крипто-валюта блокчейн сети Etherеum.

**Wei –** минимальная кратная часть Ether, 1 Ether = 1018 Wei.

**Gas –** «топливо» для выполнения транзакций в сети Ethereum, покупается пользователями при совершении транзакций.

**Gas Limit –** максимальное количество gas, которое можно купить для выполнения транзакций.

**Эксперт** - лицо, производящее рецензирование проектов.

**Площадка –** место, в рамках которой оцениваются проекты, например, университет или фонд развития инновационных проектов.

**Администратор площадки** – работник информационного отдела площадки, владеющий доступом к информационным ресурсам площадки.

# **Приложение 2 Описание и функциональное назначение файлов**

Таблица 1.1

Описание и функциональное назначение файлов

|  |  |
| --- | --- |
| **Класс** | **Функциональное назначение** |
| contractGenerator.js | Данный файл представляет собой модуль Node.js сервера, отвечающего за генерацию умных контрактов |
| contractCompiler.js | Данный файл представляет собой модуль Node.js сервера, отвечающего за компиляцию сгенерированных умных контрактов |
| frontend.js | Файл содержит обработчики событий сайта сервиса, в том числе сбор параметров и встраивание умного контракта в сеть Ethereum |

# **Приложение 2 Описание и функциональное назначение переменных и методов**

Таблица 2.1

Описание переменных и методов файла contractGenerator.js

|  |  |
| --- | --- |
| **Поля** | |
| **Название** | **Назначение** |
| contractTemplate | Скелет умного контракта |
| contractOwnerDeclaration | Шаблон декларации владельца умного контракта |
| payableFunctionsTemplate | Шаблон функций оплачиваемого контракта |
| contractDesignerTemplate | Шаблон конструктора умного контракта |
| limitedAuthorsMapping | Шаблон декларации отображения авторов |
| checkAuthorFunctionTemplate | Шаблон метода проверки автора |
| checkProjectFunctionTemplate | Шаблон метода проверки проекта |
| addAuthorFunctionTemplate | Шаблон метода добавления автора |
| removeAuthorFunctionTemplate | Шаблон метода удаления автора |
| reviewersMappingTemplate | Шаблон декларации отображения экспертов |
| selectedReviewingMapping | Шаблон декларации отображение в проекты при выборочной экспертизе |
| checkReviewerFunctionTemplate | Шаблон метода проверки эксперта |
| selectedReviewingCheckReviewerArgs | Шаблоны параметров при проверке проектов при оценке при выборочном тестировании |
| selectedReviewingCheckingCondition | Шаблон проверки эксперта при выборочном тестировании |
| notSelectedReviewingCheckingCondition | Шаблон проверки эксперта при не выборочном тестировании |
| addReviewerFunctionTemplate | Шаблон метода добавление эксперта |
| removeReviewerFunctionTemplate | Шаблон метода удаление эксперта |
| addReviewFunctionTemplate | Шаблон метода добавление отзыва |
| checkReviewerContractPart | Шаблон метода проверки эксперта |
| sendRewardTemplate | Шаблон отправки вознаграждения эксперту |
| setProjectToReviewingFunctionTemplate | Шаблон метода назначение проекта для проверки эксперту |
| viewersMappingTemplate | Шаблон декларации отображения наблюдателей |
| checkViewerFunctionTemplate | Шаблон метода проверки наблюдателя |
| addViewerFunctionTemplate | Шаблон метода добавления наблюдателя |
| removeViewerFunctionTemplate | Шаблон метода удаления наблюдателя |
| showProjectsFunctionTemplate | Шаблон метода отзывов проектов |
| showReviewsFunctionsTemplates | Шаблон методов демонстрации отзывов |
| **Методы** | |
| **Название** | **Назначение** |
| app.post | Метод, обрабатывающий POST запросы |
| generateSmartContractCode | Метод, генерирующий умный контракт |
| getShowReviewsFunctions | Метод, генерирующий функции демонстрации отзывов |
| getAddReviewFunction | Метод генерирующий функцию добавления отзыва |
| getCheckReviewerFunction | Метод генерирующий функцию добавления эксперта |
| getReviewFields | Метод установки параметров оценивая |
| getContractDesigner | Метод генерирующий конструктор умного контракта |
| readFile | Метод чтения файлов шаблонов функций |

Таблица 2.2

Описание переменных и методов файла contractCompiler.js

|  |  |
| --- | --- |
| **Методы** | |
| **Название** | **Назначение** |
| app.post | Метод, обрабатывающий POST запросы |
| compileSolidity | Метод компиляции умного контракта |

Таблица 2.3

Описание переменных и методов файла frontend.js

В данной таблице указаны метод, необходимые для работы подсистемы создания умных контрактов.

|  |  |
| --- | --- |
| **Методы** | |
| **Название** | **Назначение** |
| checkAddresses | Проверка введенных адресов учетных записей Ethereum |
| generateSmartcontract | Отправка POST запроса для генерации умного контракта согласно введенным требованиям |
| deploySmartcontract | Компиляция и встраивания умного контракта в блокчейн сеть Ethereum |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ** | | | | | | | | | |
| Изм. | Номера листов (страниц) | | | | Всего  листов  (страниц)  в докум. | № докум. | Входящий № сопроводитель-ного документа и дата | Подпись | Дата |
| изме-ненных | заме-  ненных | новых | анну-  лиро-  ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |