**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Старший преподаватель департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. В. Патнюхин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЧЕСКОГО ДЕТЕКТИРОВАНИЯ**  **ШАХМАТНЫХ ФИГУР ДЛЯ ЗАПИСИ ХОДОВ**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.01-01** **ТЗ-1-ЛУ** | | |
|  |  | |
| Исполнитель  студент группы БПИ175  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И. О. Балбин/  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | |
|  | | |
| **Москва 2019** | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| УТВЕРЖДЕН  RU. 17701729.04.01-01 81 01-1-ЛУ |  | |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЧЕСКОГО ДЕТЕКТИРОВАНИЯ**  **ШАХМАТНЫХ ФИГУР ДЛЯ ЗАПИСИ ХОДОВ**  **Пояснительная записка**  **RU. 17701729.04.01-01 81 01-1**  **Листов 15** | | | |
|  | |
| **Москва 2019** | |
|

Оглавление

[1. ВВЕДЕНИЕ 4](#_Toc54344087)

[1.1. Наименование программы 4](#_Toc54344088)

[1.2 Краткая характеристика области применения программы 4](#_Toc54344089)

[2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ 5](#_Toc54344090)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕТРИСТИКИ 6](#_Toc54344091)

[3.1. Постановка задача на разработку программы 6](#_Toc54344092)

[3.2. Описание алгоритмов и функционирование программы 6](#_Toc54344093)

[3.2.1 Загрузка спутниковых сцен 7](#_Toc54344094)

[3.2.2 Просмотр спутниковых сцен 8](#_Toc54344095)

[3.2.3. Просмотр покрытия облаков на данной спутниковой сцене 9](#_Toc54344096)

[3.2.4 Визуализация масок для выбранной спутниковой сцены 10](#_Toc54344097)

[3.2.5 Предсказания типов паттернов облаков (flower, sugar, fish, gravel), на данной спутниковой сцене. 11](#_Toc54344098)

[3.2.4 Развертывание кода на сервере. 13](#_Toc54344099)

[3.3. Описание и обоснование метода входных данных. 13](#_Toc54344100)

[3.3.1 Описание метода организации входных и выходных данных 13](#_Toc54344101)

[3.3.2 Обоснование метода организации входных и выходных данных. 13](#_Toc54344102)

[3.4. Описание и обоснование метода выбора технических и программных средств 13](#_Toc54344103)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 14](#_Toc54344104)

[4.1 Предполагаемая потребность 14](#_Toc54344105)

[Данный продукт предназначен для поиска облаков на спутниковых сценах. Фотографии с размеченными облаками можно использовать для дальнейшего анализа облаков, использовать их для моделей машинного обучения и поиска фотографий без облаков 14](#_Toc54344106)

[4.2 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 14](#_Toc54344107)

[Приложение 1 15](#_Toc54344108)

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы: «программа для обнаружения облаков на спутниковых снимках» («Program for Cloud Detection on Satellite Images»).

## Краткая характеристика области применения программы

Программа предназначена для поиска облаков на спутниковых сценах. Фотографии с размеченными облаками можно использовать для дальнейшего анализа облаков, использовать их для моделей машинного обучения и поиска фотографий без облаков (Пример: для карт необходимы фотографий без облаков).

Так же, в дальнейшем, можно классифицировать найденные облака на типы (кучевые, перистые, дождевые и тд).

# ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Программа выполнена в рамках темы курсовой работы — «Приложение для автоматического детектирования шахматных фигур для записи ходов», в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия».

Основанием для разработки является приказ от 10.04.2020 г. № 2.3-02/1004-01 декана факультета компьютерных наук НИУ ВШЭ.

# ТЕХНИЧЕСКИЕ ХАРАКТЕТРИСТИКИ

## 3.1. Постановка задача на разработку программы

Программа должна обеспечить возможность выполнение следующих функций:

1. Загрузку спутниковых сцен
2. Промотр спутниковых сцен
3. Просмотр покрытия облаков, на данной спутниковой сцене
4. Визуализация масок для выбранной спутниковой сцены
5. Предсказания типов паттернов облаков (flower, sugar, fish, gravel), на данной спутниковой сцене
6. Развертывание кода на сервере

## 3.2. Описание алгоритмов и функционирование программы

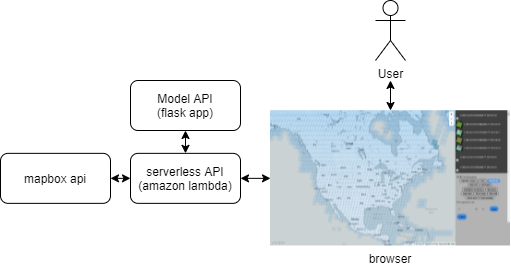


Рис. 1 Архитектура приложения

Приложение состоит из трех частей (рис. 1):

* Serverless приложение, предоставляющее API
* Mapbox api – сервис, предоставляющий API для визуализации карты
* Model API – сервер с Flask приложением, предоставляющий API Для работы с моделью

### 3.2.1 Загрузка спутниковых сцен

Существует множество датасетов, предоставляющих спутниковые сцены. Тем не менее, открытых, бесплатных и охватывающих большую часть земного шара. Поэтому в качестве данных используется спутниковые сцены с миссии landsat-8 за 2017-2018 ый год т.к. они бесплатный и покрывают большую часть земного шара.

Landsat-8 – это спутник дистанционного зондирования Земли, запущенный Nasa.

Для загрузки растровых сцен на карту, были реализованы два метода в serverless сервере, совмещенные с mapbox API, а именно:

* [GET] /metadata/<scene>
  + Возвращает метаинформацию о сцене <csene>
    - Bounds – границы сцены
    - rgbMinMax – минимальное/максимальное значение в каждом канале
    - sceneId – Id сцены
* [GET] /tiles/<scene>/<int:z>/<int:x>/<int:y>.<ext>
  + Возвращает растровое изображение в формате <ext> (поддерживается jpg и png), содержащий квадрат под номером <int:x> <int:y> (проекция меркатора) с увеличением <int:z>

Так же реализованы два метода для отрисовки найденных иконок по найденной границе на карте (см рис.1):

* [GET] /search?<query>
  + Возвращает список найденных сцен содержащих:
    - scenceId – Id сцены
    - date – дата сцены
* [GET] /bounds/<scene>
  + Возвращает границы сцены <sceneId>

Т.к. количество бесплатных запросов в aws Amazon ограничено, а поиск и загрзука изображений выдает большое количество запросов (>100) на одно нажатие, было принято решение использовать бесплатное публичное sat\_api, предоставленное developmentseed.org. К сожалению, в нем содержатся далеко не все изображения, предоставленные для наших сцен.

### 3.2.2 Просмотр спутниковых сцен

Для просмотра спутниковых сцен реализовано web приложение на Node.js.

Для отрисовки карты используется mapbox api



Рис. 2 начальный экран web-приложения

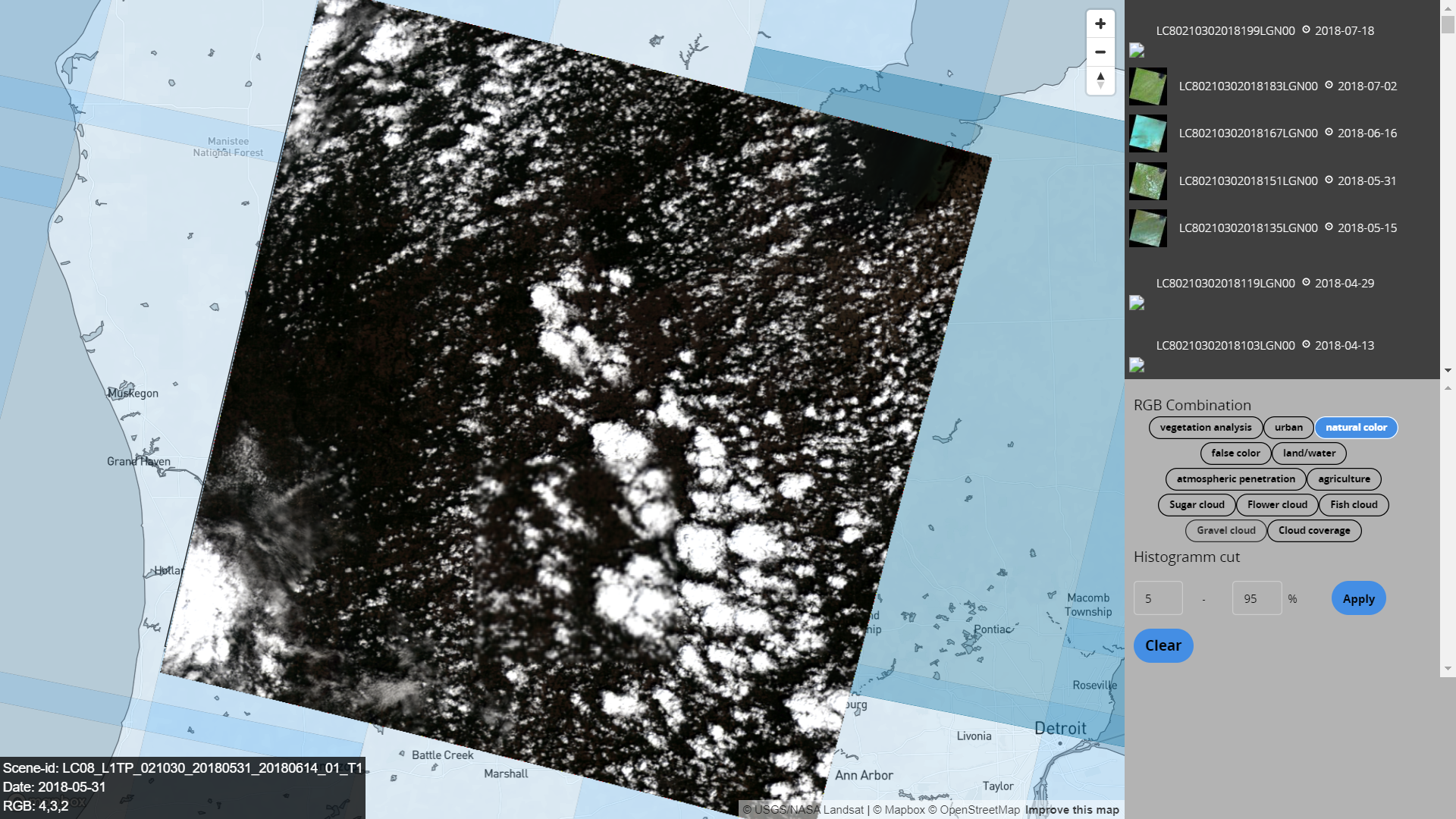
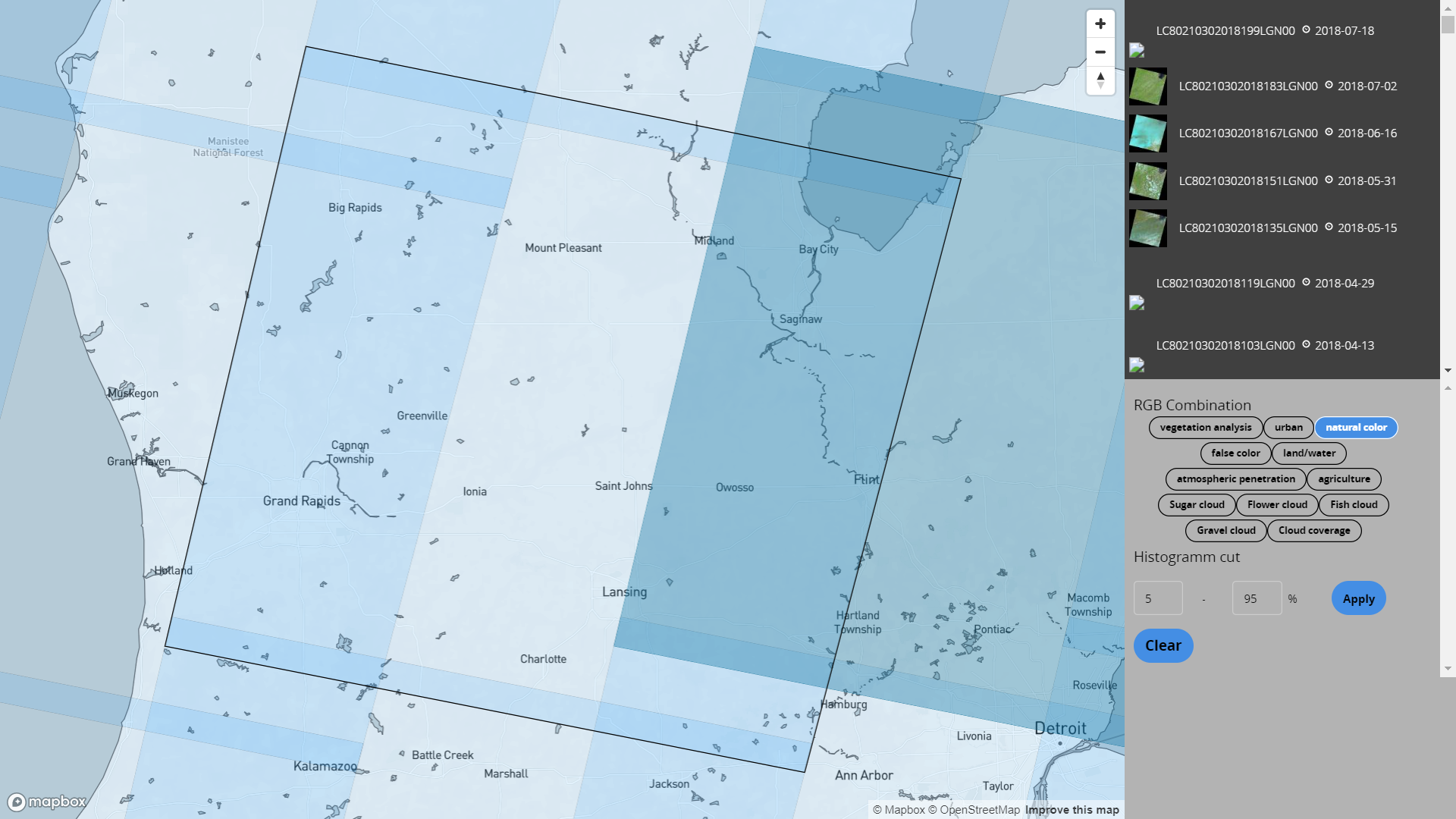
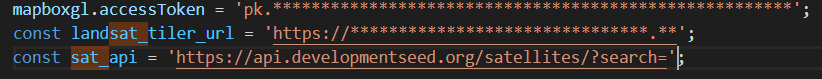


Рис. 3 web-приложения при выборе сцены

Для работы Web-приложения необходимо указать в файле app.js следующие ключи:  


* Mapboxgl.accessToken - публичный ключ доступа к mapbox
* landsat\_tiler\_url – url с endpoint serverless приложения
* sat\_api – url c endpoint для поиска сцен по области на карте

В приложении так же реализован алгоритм для улучшения контрастности изображения. Для этого в интерфейсе есть форма Histogram (Рис. 3). В нее вбивается значения, по которым нужно обрезан спектр изображения, после чего каналы будут обрезаны и растянуты до вписанных границ.

В самом же приложении (app.js) основными функциями являются:

* const UpdateRasterTile () – метод, обновляющий растровый слой на карте
* const InitScene() – метод для инициализации сцены

### 3.2.3. Просмотр покрытия облаков на данной спутниковой сцене

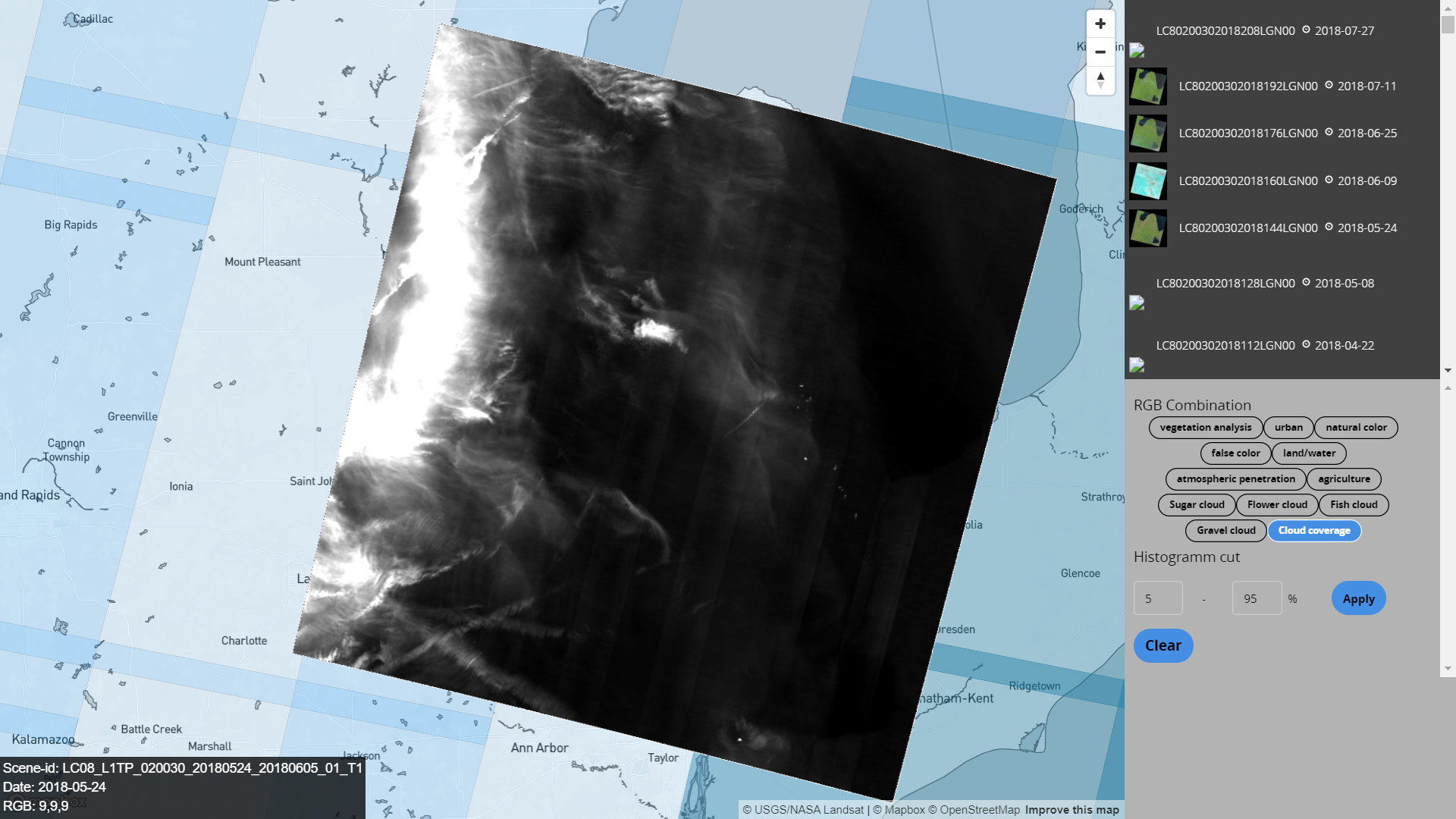


Рис. 4 Cloud coverage

Для просмотра облаков необходимо нажать на кнопку “Cloud coverage”. Эти данные так же предоставлены landsat-8 в виде маски. Каждый пиксель – вероятность [0..1] облака в этой точке.

### 3.2.4 Визуализация масок для выбранной спутниковой сцены

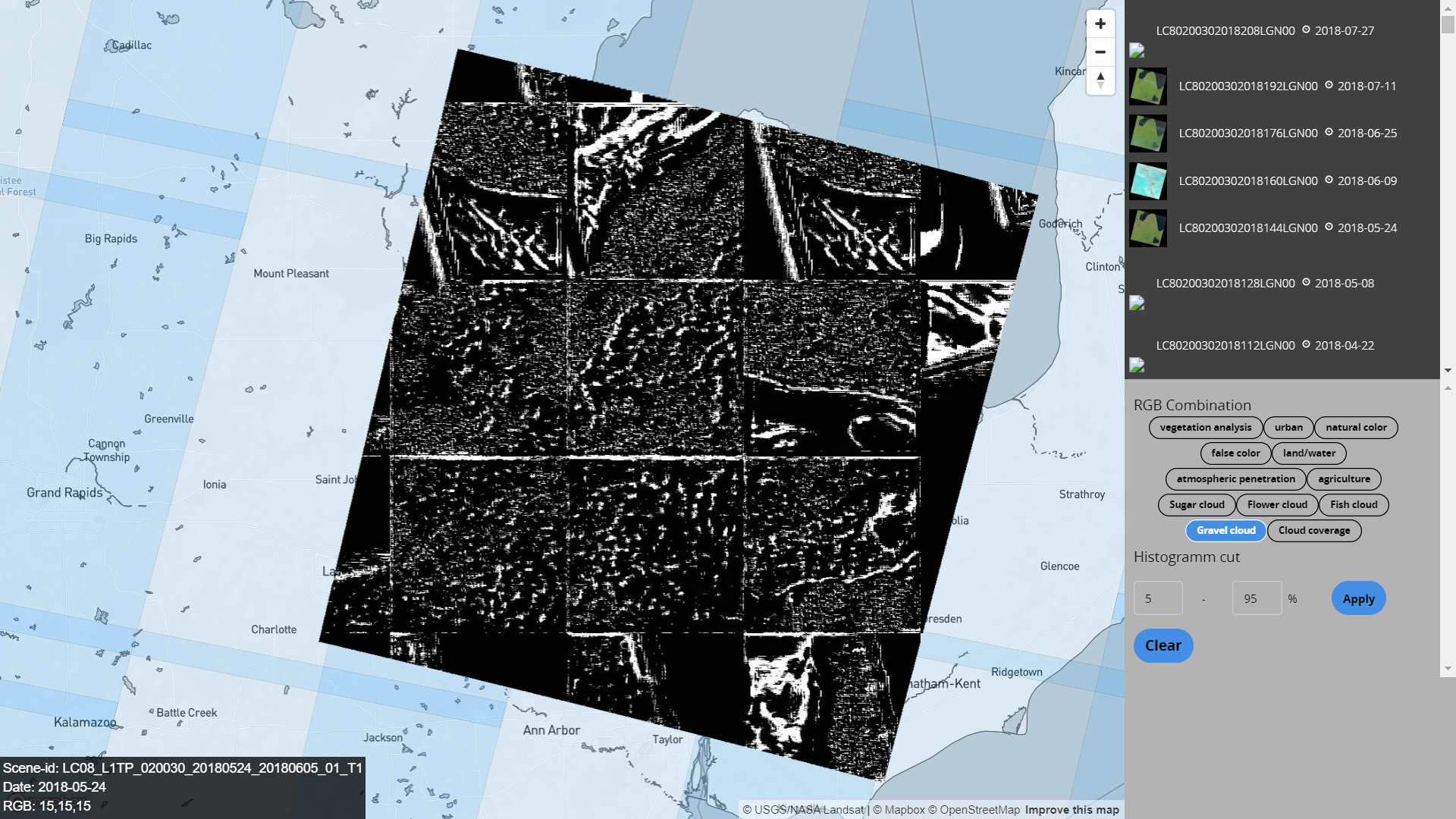


Рис. 5 Grabel cloud

Для предсказания типов паттернов облаков необходимо нажать соответствующею кнопку (Sugar cloud, Flower cloud, Fish cloud, Gravel cloud).

После появиться соответствующая бинарная маска.

### 3.2.5 Предсказания типов паттернов облаков (flower, sugar, fish, gravel), на данной спутниковой сцене.

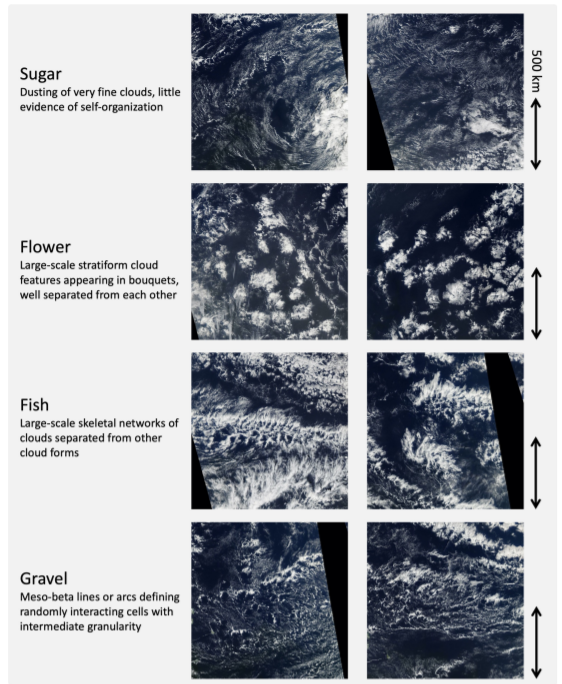


Рис. 6 Описание паттернов облаков с примерами

Для предсказания типов паттернов облаков была обучена нейросеть на соответсвующим датасете.

Для построения масок были выбраны несколько архитектур нейросетей, а именно:

* Unet
* DenseNet
* GoogleNet

Далее для каждой из сетей была собрана сетка параметров для перебора:

* Encoder
* Decoder
* Количество нейронов на внутренних слоях
* Функция активации
* Аугментации

В ходе экспериментов была выявлена лучшая архитектура, а именно Unet с начальной инициализацией весов ImageNet, с Resnset-50 в качестве Encoder - а.

Для реализация использовался pyrhon-фреймворк Catalyst – обертка на pytorch.

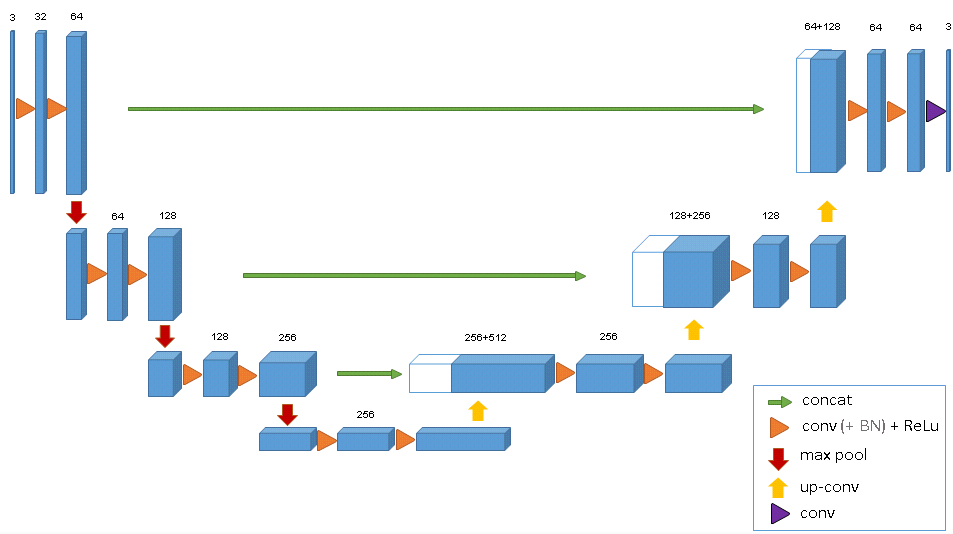


Рис. 7 Архитектура Unet

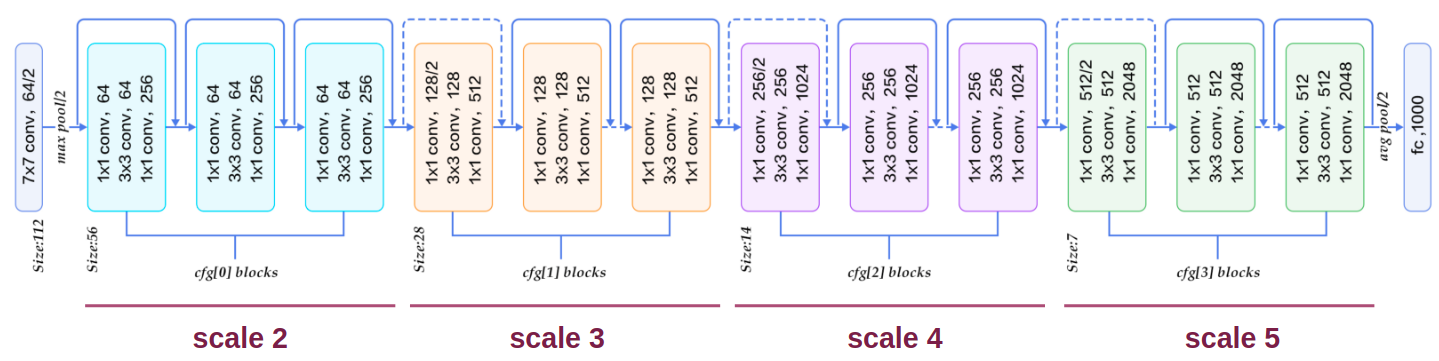


Рис. 8 Архитектура Resnet-50

Для развертывания модели было написано flask приложение, и развернуто на личном рабочем сервере.

Flask сервер предоставляет следующее api:

* [POST] /getmask/
  + Request json:
    - img – бинарная строка с изображением
    - label\_id – id маски:
      * 0 – Sugar
      * 1 – Flower
      * 2 – Fish
      * 3 – Gravel

Сервер развернут с помощью ngrok.

### 3.2.4 Развертывание кода на сервере.

Для доставки и развертывания кода на сервере используется docker + serverless.

Serverless – обертка над AWS lambda, позволяющая разворачивать приложение целиком через командную строку. У AWS lambda есть ограничение на 5к запросов в месяц для бесплатного использования и 60мб на Docker контейнер.

Из за ограничения в 60мб мы не можем в серверлесс развернуть модель (только веса весят 243 мб) и использовать тяжеловесные фреймворки и библиотеки, по типу opencv, tensorflow, pytorch и тд.

Для того чтобы доставить код, был написан:

* Dockerfile – содержащий настройки для контейнера
* Makefile – содержащий сценарий для созданий и тестирования Docker – контейнера и пакета для Serverless

Чтобы развернуть приложения необходимо иметь предустановленные nodejs (npm) и dokcer и выполнить следующие шаги.

* Перейти в консоль в папку landsat\_viewer
* Выполнить: npm install (установиться serverless и необходимые библиотеки)
* Выполнить: sls login и выполнить инструкции, уканые в браузере
* Выполнить: make all – создать пакет для развертывания, может занять несколько минут
* Выполнить: sls deploy – задеплоить приложение, может занять несколько минуть
* Сохранить endpoint, который отобразится в консоли

## 3.3. Описание и обоснование метода входных данных.

### 3.3.1 Описание метода организации входных и выходных данных

Входными данными является клавиатура, нажатие копок мыши. Для серверной части входными данными является различные API.

### 3.3.2 Обоснование метода организации входных и выходных данных.

Для взаимодействия и управление программой пользователю необходимы клавиатура и мышь.

## 3.4. Описание и обоснование метода выбора технических и программных средств

Для использования web – приложения необходим только доступ к браузеру, на любом устройстве, с техническим средствам позволяющем ему работать без перебоев с запущенным браузером.

Для сервера с моделью рекомендуемые требования:

* ОС: Ubuntu server
* Видеокарта: nvidia geforce gtx 1060 6gb
* Процессор: intel Core i5-6400
* Оперативная память: 8 Гб

# 4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## 4.1 Предполагаемая потребность

## Данный продукт предназначен для поиска облаков на спутниковых сценах. Фотографии с размеченными облаками можно использовать для дальнейшего анализа облаков, использовать их для моделей машинного обучения и поиска фотографий без облаков

## 4.2 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

В рамках данной работы не найдены аналоги на рынке.

# Приложение 1

**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 19.201-78 Техническое задание. 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
14. <https://docs.mapbox.com/>
15. <https://www.serverless.com/framework/docs/>
16. <https://flask.palletsprojects.com/en/1.1.x/>
17. <https://docs.aws.amazon.com/lambda/index.html>
18. <https://arxiv.org/abs/1505.04597>
19. <https://arxiv.org/abs/1409.4842>
20. <https://arxiv.org/abs/1608.06993>
21. https://arxiv.org/abs/1512.03385

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |