Отчет летний практики 2024

# 1.Введение

Космические снимки являются драгоценным ресурсом, и защита авторских прав на них является достаточно острой проблемой. В то же время не следует вносить значительные искажения в важные области изображений, так как это может негативно сказаться на результате обработки спутниковых изображений. Проект Международной климатологии облачности оценивает глобальное среднегодовое облачное покрытие до 66%, что подтверждает, что облака занимают значительную часть в большинстве космических изображений. Более того, эти облака не содержат важной информации при цифровой обработке. Исходя из вышеизложенных причин, мы можем обеспечить защиту авторских прав на космические снимки, внедряя в них цифровые водяные знаки (ЦВЗ) в облаках, где искажения от внедрения ЦВЗ все равно не повлияют на содержащуюся в изображении важную информацию.

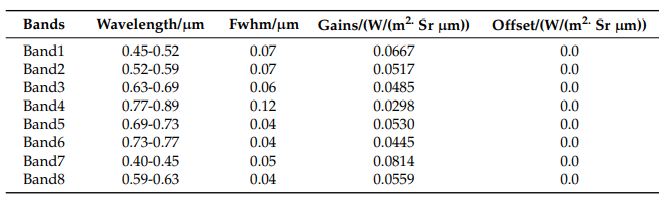
# 2. Методы сегментации на космические снимки

**В последние годы было предложено множество методов обнаружения облаков и их тени.** Многие классические методы были применены к таким спутникам, как MODIS, Landsat и Sentinel-2, что привело к появлению различных продуктов, связанных с облаками и их тенью. **Методы обнаружения облаков и их тени можно разделить на три группы в зависимости от количества используемых изображений: методы на основе одного изображения, методы на основе нескольких изображений и методы машинного обучения.**

## **2.1. Метод на основе одного изображения (Одноразовый метод)**

Одноразовые методы обычно устанавливают пороги на основе физических характеристик облаков. Например, маска облаков MODIS использует множественные обнаружения облаков для оценки вероятности ясного неба, используя серию порогов для проведения различных спектральных тестов на пикселях, что приводит к классификации нескольких пикселей. Геологическая служба США (USGS) применяет автоматическую оценку облачного покрытия ACCA и функцию маски Fmask в качестве методов обнаружения облаков и теней от облаков для Landsat. ACCA оценивает процент облачного покрытия в каждом изображении Landsat с использованием нескольких спектральных фильтров, с особенным акцентом на тепловой инфракрасный диапазон. С другой стороны, Fmask интегрирует различные диапазоны, включая циррусные диапазоны, цифровую модель рельефа (DEM), глобальную маску воды, обнаружение снега и т. д. Fmask демонстрирует точные возможности обнаружения облаков и теней от облаков на различных сложных сценах в изображениях Landsat и Sentinel-2. Индекс перемещения облаков CDI дополняет теорию обнаружения облаков. В дополнение к учету спектральных характеристик, CDI также интегрирует систематическую структуру наблюдения Sentinel-2 для точного различения ярких объектов от облаков на основе параллакса полосы. Успешное применение CDI в обнаружении облаков в крупных городских районах интегрировано в рамки Fmask и Force. Sen2Cor был разработан Европейским космическим агентством (ESA) для атмосферной коррекции спутников Sentinel-2. Результирующая карта классификации сцен (SCL), с идентификаторами такими как облака, тень от облаков, циррус и вероятность снега, также может использоваться для обнаружения облаков и теней от облаков.

Рассмотрим один пример этого метода из литературы [?]: An Improved Fmask Method for Cloud Detection in GF-6 WFV Based on Spectral-Contextual Information

Данные GF-6 WFV содержат восемь диапазонов. На основе четырех обычных диапазонов синего, зеленого, красного и ближнего инфракрасного (NIR) добавлены позиции красного края 1 и 2, ультрафиолетовый диапазон и желтый диапазон. Конкретная информация о диапазонах представлена в Таблице 1. Несмотря на то, что данные включают восемь диапазонов, для обнаружения облаков может быть использовано не так много диапазонов из-за узкого спектрального диапазона, который они охватывают.

Процессы построения маски облаков состоит из 2 главных этапа: тесты на идентификация потенциальных облачных пикселей (PCPs) и Расчет вероятности облачных пикселей

Тесты на PCPs:

1. Базовый тест:

где

2. HOT test:

3. Rock and Bare Soil Test

4. Build test:

(Band1, 25) и (Band3, 75) представляют собой 25-й и 75-й проценты, упорядоченные по каждому ряду синего и красного каналов.

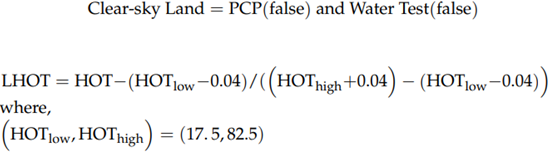
5. Stratus Test

Расчет вероятности облачных пикселей:

1. Вероятность яркости для воды:

2. Вероятность облачности для суши:

1) LHOT



Где (17.5, 82.5) – проценты от коэффициента HOT

2) Вероятность изменчивости для суши:

LCloud-Prob считаются по LHOT и Variability-Prob:

Potential Cloud Layer является true, если:

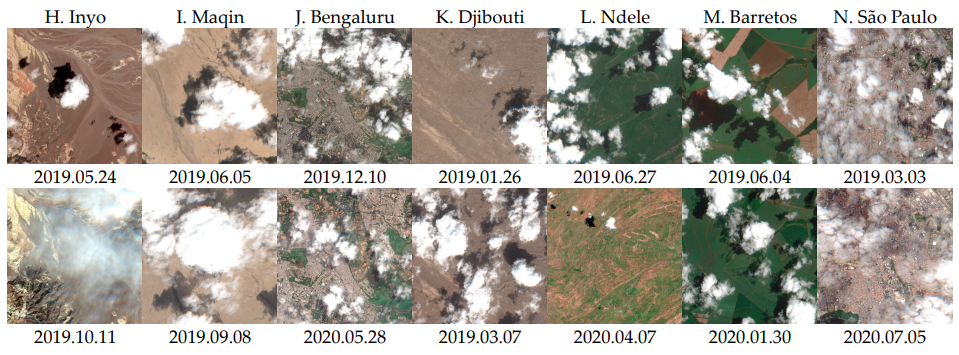
* (PCP(true) и Water Test(true) и wCloud\_Prob > 0.5) или
* (PCP(true) и Water Test(false) и LCloud\_Prob > Land \_threshold) или
* (LCloud\_Prob > 0.99 and Water Test(false))

## **2.2. Метод на основе нескольких изображений**

Многовременной метод включает в себя характеристики временного измерения, предполагая, что наземные объекты обычно меняются медленно в течение сезонов или периодов. Распознавание облаков и теней от облаков достигается путем обнаружения изменений в временном ряду изображений. По сравнению с одноразовым методом, метод временного ряда выигрывает от временной информации и дополнительно усиливает возможности обнаружения облаков и теней от облаков. Многовременное обнаружение облаков (MTCD) сочетает соседние временные пиксели, чтобы зафиксировать резкое увеличение отражательной способности синего диапазона для обнаружения облаков. Этот метод дает хорошие результаты для толстых облаков и демонстрирует определенный уровень эффективности для обнаружения тонких облаков.

Многовременная маска, Tmask, расширяет принципы Fmask, включая зеленый диапазон, ближний инфракрасный диапазон и коротковолновый инфракрасный диапазон. Синус-косинусная временная модель каждого пикселя строится с использованием итеративного метода взвешенных наименьших квадратов, что значительно усиливает способность обнаруживать тени от тонких облаков. Лин предложил многовременной метод обнаружения облаков, основанный на неизменных пикселях. Неизменные пиксели извлекаются с использованием взвешенного метода главных компонент.

Можно рассматривать этот метод как расширение одноразового метода, при этом маска облаков Fmask не только зависеть от расположения пиксели в изображении а еще от времени.

Рис ?. Пример набор данных для метода **на основе нескольких изображений: снимки каждой области на 2 различных дня.**

## 2.3. Методы основаны на машинное обучение и глубокое обучение

Машинное обучение (ML) значительно продвинулось в создании масок облаков, обеспечивая эффективную и точную идентификацию сложных облачных структур в больших наборах данных, таких как спутниковые изображения и климатические данны. В отличие от традиционного программирования, модели ML автономно учатся на основе входных данных и соответствующих выходных данных, выявляя скрытые корреляции. Этот процесс обучения улучшает точность и эффективность создания масок облаков, снижая необходимость в обширном программировании и глубоких экспертных знаниях.

Ключевым фактором успеха ML в создании масок облаков является использование размеченных данных, необходимых для обучения моделей различать разные типы облаков и области с ясным небом. Размеченные данные позволяют проводить обучение и валидацию данных, передавая моделям способности к выводу знаний в прикладную область. Более того, это позволяет обеспечивать точные прогнозы для новых, ранее не виденных данных, повышая надежность масок облаков в различных условиях.

Методы ML обычно применяются к обнаружению облаков как к задаче классификации одного пикселя с учителем, идентифицируя такие классы, как облака, отсутствие облаков, тень от облаков, земля или определенные типы облаков. Распространенные алгоритмы включают искусственные нейронные сети (ANN), машины опорных векторов (SVM) и методы случайного леса (RF), каждый из которых захватывает сложные взаимосвязи между входными признаками и выходными метками.

В традиционных алгоритмах машинного обучения перед анализом спутниковые данные могут проходить предварительную обработку для извлечения соответствующих спектральных диапазонов и индексов, которые затем используются в качестве входных данных для этих моделей. Финальные маски состоят из вероятностей пикселей, относящихся к предопределенным классам или меткам, соответствующим категориям.

Подготовка обширных размеченных данных для обучения алгоритмов может быть ресурсоемкой и требовать много времени, что ограничивает их применимость в определенных сценариях. Кроме того, создание адекватных наборов данных для обучения является значительной проблемой. Это требует знаний о спектральных свойствах облаков, а также методов анализа изображений. Более того, трудно учитывать все возможные комбинации облаков заранее при создании классификатора, особенно в редких или локализованных условиях. Спутниковые изображения предоставляют различную спектральную информацию, разные пространственные разрешения и большое разнообразие сложных типов земной поверхности, что следует учитывать для адаптации модели. Кроме того, производительность сенсора может ограничивать эффективность модели. Например, отсутствие ключевых длин волн может затруднить нахождение признаков, которые могут снизить частоту неверной дискриминации облаков. Наконец, отсутствие пространственной информации, в виде пространственной корреляции между соседними пикселями, что характерно для методов с одним пикселем, может значительно снизить точность.

Эти критические аспекты традиционных моделей машинного обучения улучшаются или преодолеваются применением методов глубокого обучения. Алгоритмы глубокого обучения добились выдающихся результатов в классификации изображений благодаря их превосходному представлению признаков, что повышает окончательную точность.

Глубокое обучение позволяет разработке вычислительных моделей, структурированных с несколькими уровнями обработки. Эти модели способны изучать и идентифицировать представления данных через различные уровни абстракции. Архитектуры глубокого обучения, такие как сверточные нейронные сети (CNN) и полностью сверточные нейронные сети (FCN), значительно улучшили результаты обнаружения облаков и их маскировки. В частности, их способность эффективно захватывать пространственные узоры и структуры изображений, в дополнение к спектральным характеристикам, привела к более точным алгоритмам классификации и сегментации изображений

Двухэтапная структура кодировщик-декодировщик широко используется, где пространственная информация восстанавливается после уменьшения размера карты признаков путем постепенного увеличения пространственных признаков на этапе апсемплинга. Классические примеры - хорошо известные архитектуры U-Net и SegNet, разработанные для сегментации биомедицинских изображений и природных изображений соответственно, а также более новые SFANet и CANet. Модель U-Net породила несколько вариаций в различных областях, таких как UNet++, вложенная архитектура U-Net, предложенная в для медицинских изображений.

В этом исследовании мы будем реализовать сегментацию облаков с применением данного метода. Одним из важных требований данного метода является достаточно большой датасет, который мы будем смотрить на следующий пункт.

# 3. Используемый датасет

Ссылка к датасету: <https://www.kaggle.com/datasets/sorour/38cloud-cloud-segmentation-in-satellite-images>

Название датасета: “38-Cloud: A Cloud Segmentation Dataset”

Этот набор данных содержит 38 снимков сцен Landsat 8 и их вручную извлеченные на уровне пикселей истинные значения для обнаружения облаков. Весь изображения этих сцен были обрезаны на несколько участков размером 384x384, чтобы они подходили для алгоритмов семантической сегментации на основе глубокого обучения. Существует 8400 участков для обучения и 9201 участков для тестирования. Каждый участок имеет 4 соответствующих спектральных канала, которые включают Красный (канал 4), Зеленый (канал 3), Синий (канал 2) и Ближний инфракрасный (канал 5). В отличие от других изображений компьютерного зрения, эти каналы не объединены вместе. Вместо этого они находятся в своих соответствующих директориях.

Каждое изображение имеет расширение ".TIF" и использует 2 байта для представления яркости каждого пикселя. Максимальное значения яркости при этом равно

Дерево каталогов этого набора данных выглядит следующим образом:

├──38-Cloud\_training

│------------├──train\_red

│------------├──train\_green

│------------├──train\_blue

│------------├──train\_nir

│------------├──train\_gt

│------------├──Natural\_False\_Color

│------------├──Entire\_scene\_gts

│------------├──training\_patches\_38-Cloud.csv

│------------├──training\_sceneids\_38-Cloud.csv

├──38-Cloud\_test

│------------├──test\_red

│------------├──test\_green

│------------├──test\_blue

│------------├──test\_nir

│------------├──Natural\_False\_Color

│------------├──Entire\_scene\_gts

│------------├──test\_patches\_38-Cloud.csv

│------------├──test\_sceneids\_38-Cloud.csv

Основываясь на этом наборе данных, мы разработали модель, которая принимает на вход нормализованные изображения размером (4, 384, 384) или (3, 384, 384), при этом значения яркости пиксели изображения находятся в диапазоне [0, 1]. На выходе модели (метка) получается бинарная маска размером (1, 384, 384), где значения элементов маски либо 0 (отсутствие облаков), либо 1 (наличие облака).

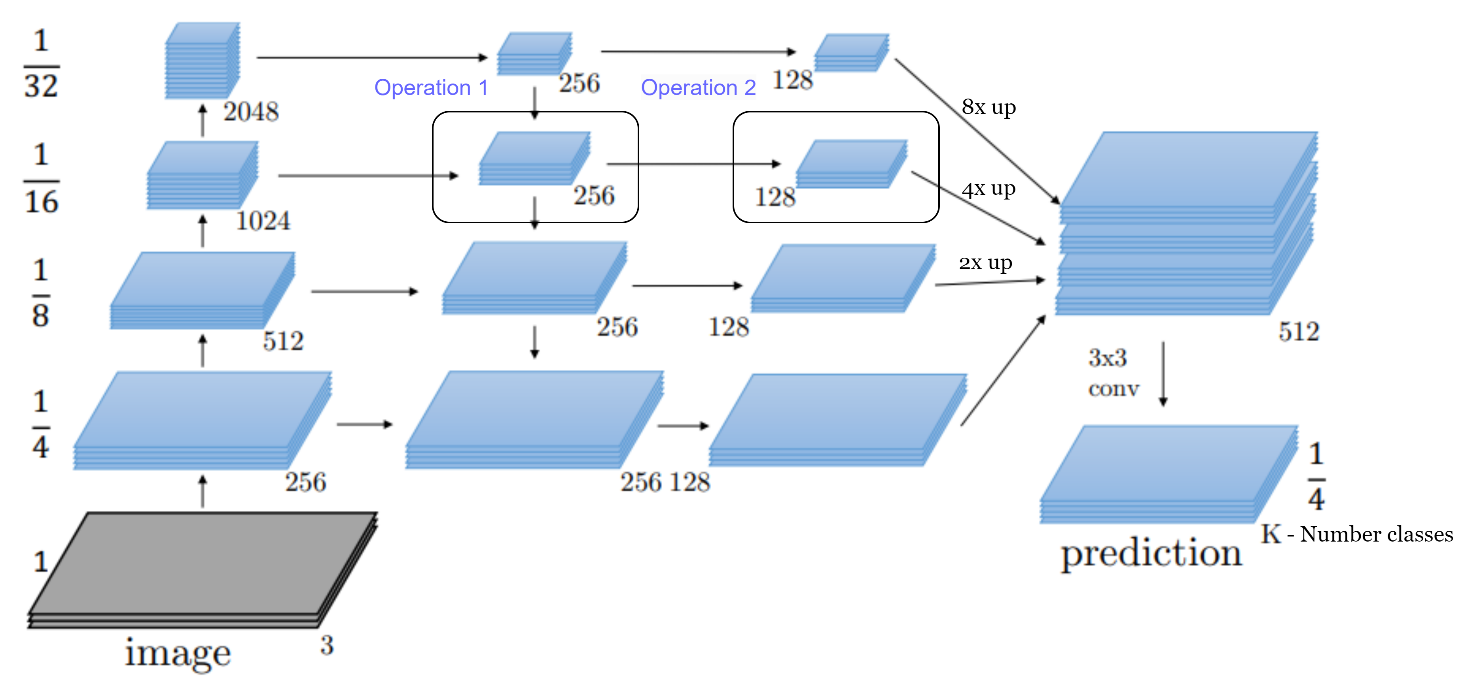
# 4. Выбор архитектуры для модели глубокого обучения в ислледовании

## 4.1. Готовый модель 1 на FPN

Ссылка: <https://www.kaggle.com/code/timyapew/fpn-cloud-detection>

Структура:

Feature Pyramid Net (FPN) – свёрточная нейронная сеть, построенная в виде пирамиды, служащая для объединения достоинств карт признаков нижних и верхних уровней сети. С начало, FPN построен в задаче обнаружения объекта, но мы можем модифицировать его последние слоя для использования его в задаче сегментации объекта



Где:

|  |  |
| --- | --- |
| Операция 1:  C:\Users\phamn\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\47F2939F.tmp | Операция 2:  C:\Users\phamn\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\4011FE05.tmp |

Рис ?. Архитектура FPN в задаче сегментации

FPN использует пирамидальную иерархию глубоких сверточных сетей для построения пирамид признаков с незначительными дополнительными затратами. Она состоит из восходящих и нисходящих путей. Для построения карт признаков высокого уровня на всех масштабах разработан нисходящий путь с боковыми соединениями. Эта архитектура демонстрирует значительное улучшение в качестве универсального извлекателя признаков в нескольких приложениях, таких как обнаружение объектов и сегментация отдельных объектов.

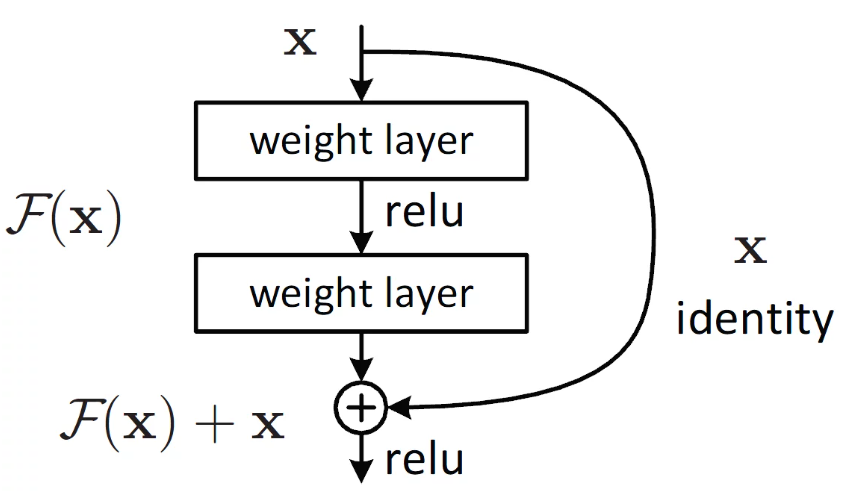
На входе модели принимает 3 белых-черных изображения с соответствием 3 каналам Red, Green, Blue, размер входа при этом будет [Batch, 3, 384, 384].

Процесс уменьшения размера c увеличением глубинка входа происходит:

Процесс повышения размера c фиксированием глубинка входа происходит:

На выходе принимаем функцию softmax по оси 1 и выбираем индекс элемента с максимальным значением. Поскольку по оси 1 имеются всего 2 элемента, значение индекса возможно только 0 или 1. 0 означает что на маски нет облаков, а 1 означает что в данном пикселе есть облако.

В качестве кодера/декодера используем ResNet, который имеет преобучаемый вес, позволяющий увеличение качество глубокой модели



Рис? Реализация соединения быстрого доступа в ResNet

Когда более глубокая сеть начинает сворачиваться, возникает проблема: с увеличением глубины сети точность сначала увеличивается, а затем быстро ухудшается. Снижение точности обучения показывает, что не все сети легко оптимизировать.

Чтобы преодолеть эту проблему, Microsoft ввела глубокую «остаточную» структуру обучения. Вместо того, чтобы надеяться на то, что каждые несколько stacked layers непосредственно соответствуют желаемому основному представлению, они явно позволяют этим слоям соответствовать «остаточному». Формулировка F(x) + x может быть реализована с помощью нейронных сетей **с соединениями для быстрого доступа.**

Соединения быстрого доступа (shortcut connections) пропускают один или несколько слоев и выполняют сопоставление идентификаторов. Их выходы добавляются к выходам stacked layers. Используя ResNet, можно решить множество проблем, таких как:

* ResNet относительно легко оптимизировать: «простые» сети (которые просто складывают слои) показывают большую ошибку обучения, когда глубина увеличивается.
* ResNet позволяет относительно легко увеличить точность благодаря увеличению глубины, чего с другими сетями добиться сложнее.

## 4.2. Готовый модель 2 на U-Net

Ссылка: https://www.kaggle.com/code/simple11/cloud-segmentation

**U-Net** считается одной из стандартных архитектур CNN для задач сегментации изображений, когда нужно не только определить класс изображения целиком, но и**сегментировать его области по классу**, т. е. создать маску, которая будет разделять изображение на несколько классов. Архитектура состоит из стягивающего пути для захвата контекста и симметричного расширяющегося пути, который позволяет осуществить точную локализацию.

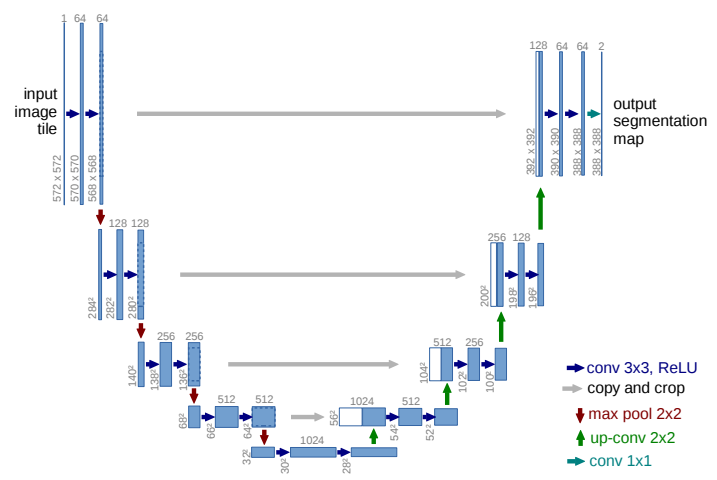


Рис ? Структура U-Net

Архитектура сети проиллюстрирована на Рисунке ?. Она состоит из сжимающего пути (левая сторона) и расширяющего пути (правая сторона). Сжимающий путь следует типичной архитектуре сверточной сети. Он включает повторное применение двух сверточных операций 3x3 (свертки без добавления нулей), каждая из которых сопровождается выпрямляющим линейным элементом (ReLU) и операцией максимального объединения 2x2 с шагом 2 для понижающей выборки. На каждом шаге понижающей выборки мы удваиваем количество каналов признаков. Каждый шаг в расширяющем пути включает увеличение масштаба карты признаков, за которым следует свертка 2x2 ("обратная свертка"), которая уменьшает количество каналов признаков вдвое, объединение с соответствующим обрезанным картой признаков из сжимающего пути и две свертки 3x3, каждая из которых сопровождается ReLU. Обрезка необходима из-за потери граничных пикселей на каждой свертке. На финальном слое используется свертка 1x1 для отображения каждого вектора признаков, состоящего из 64 компонентов, в нужное количество классов.

В общей сложности сеть имеет 23 сверточных слоя. Чтобы обеспечить бесшовную укладку карты сегментации на выходе, важно выбрать размер входного блока таким образом, чтобы все операции максимального объединения 2x2 применялись к слою с четными размерами по осям x и y.

В качестве кодера/декодера данный модель тоже использует ResNet с преобучаемым весом датасета ImageNet.

## 4.3. Готовый модель 3 на U-Net без кодера

Ссылка: <https://www.kaggle.com/code/lukasgren/cloud-segmentation>

Данный модель тоже имеет архитектуру как предыдущий модель, но без преобучаемый вес кодера

## 4.4. Обучить модель с помощью пакета pip segmentation-models-pytorch

segmentation-models-pytorch — это библиотека для сегментации изображений на базе PyTorch, которая предоставляет готовые к использованию модели сегментации с предварительно обученными спинками (backbones).

### Основные особенности:

1. **Модели сегментации**: Включает популярные архитектуры моделей сегментации, такие как U-Net, FPN, LinkNet и другие.
2. **Предварительно обученные спинки (backbones)**: Использует предварительно обученные модели из библиотек, таких как ResNet, VGG, EfficientNet и других, для улучшения качества сегментации и ускорения обучения.
3. **Легкость использования**: Обеспечивает простой интерфейс для загрузки и использования моделей, что позволяет легко интегрировать их в ваши проекты.



Ссылка к папке pip: <https://pypi.org/project/segmentation-models-pytorch/>

В данной исследовании мы будем обучить 2 модели на основе архитектуры UNet и FPN. В качестве кодера, 2 модели имеют готовый вес на основе ResNet34

Некоторые гиперпараметры:

1. Batch size: 16
2. Epochs: 5
3. Optimizer: Adam
4. Learning rate: 0.0001

Для оптимзации потери в задаче сегментации, будем использовать функцию потери DiceLoss.

Коэффициент Дайса, или коэффициент Дайса-Сёренсена, является общепринятой метрикой для сегментации пикселей, которую также можно модифицировать, чтобы она действовала как функция потерь:

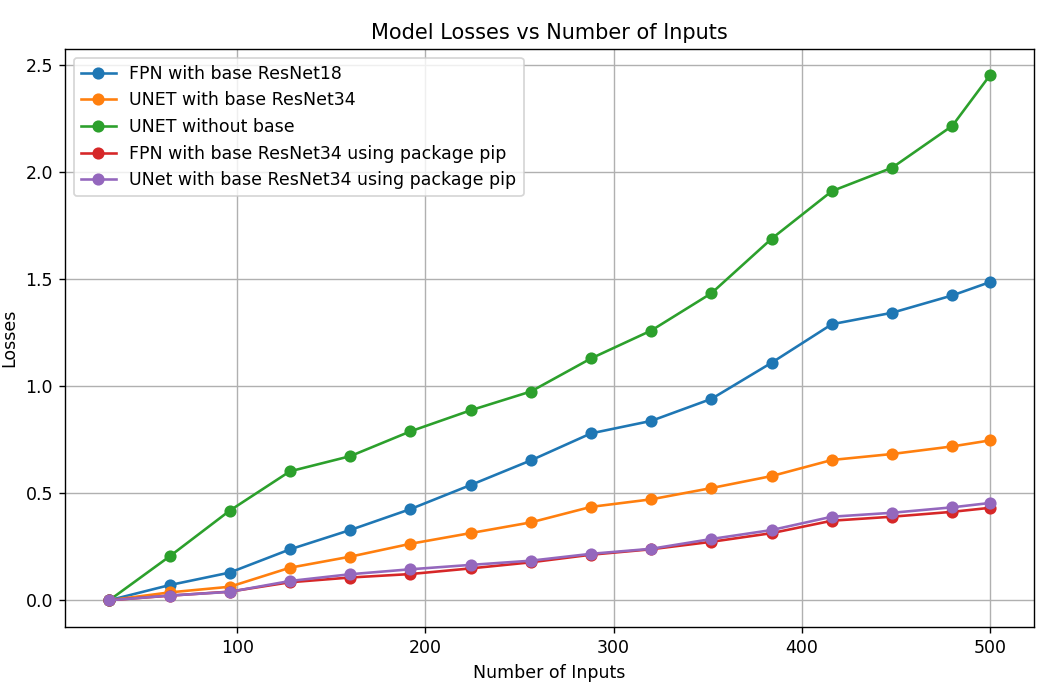
характерирует степень совпадения между 2 области, равен 2 раза отношению пересечения на сумме площади 2 области. Значение находится на интервале [0, 1] и наша стратегия - максимизировать этот коэффициент к 1. Поэтому построим фукцию потери , который уменьшается к 0 при увеличении к 1.

## 4.5. Сравнение работы моделей

Рассмотрим работы 5 моделей с одним входом:

|  |  |  |  |
| --- | --- | --- | --- |
| Исходное изображение | C:\Users\phamn\PycharmProjects\PracticeSummer5\1_true_dataset\truecolor_patch_192_10_by_12_LC08_L1TP_002053_20160520_20170324_01_T1.jpg | Верная маска |  |
| Выход модели 1 |  | Выход модели 2 |  |
| Выход модели 3 |  | Выход модели 4 (FPN) |  |
| Выход модели 5  (UNet) |  |  |  |

Графики ошибки в зависимости от число входных изображений:



Заключение:

* Архитектура FPN или UNet не оказывает существенного влияния на производительность модели на данном наборе данных.
* Использование кодера/декодера с обучаемыми весами приводит к улучшению производительности модели.
* Будем выбирать модель 4 (UNet) для реализации задачи ЦВЗ на облако

# 5. Применение модели для встраивания и извлечения ЦВЗ на облако

https://app.diagrams.net/?src=about#G1bJB4Ij7z4ETrDSk-CFvIKdpviOKgvgp5#%7B%22pageId%22%3A%22C5RBs43oDa-KdzZeNtuy%22%7D

Схема процедуры встраивание:

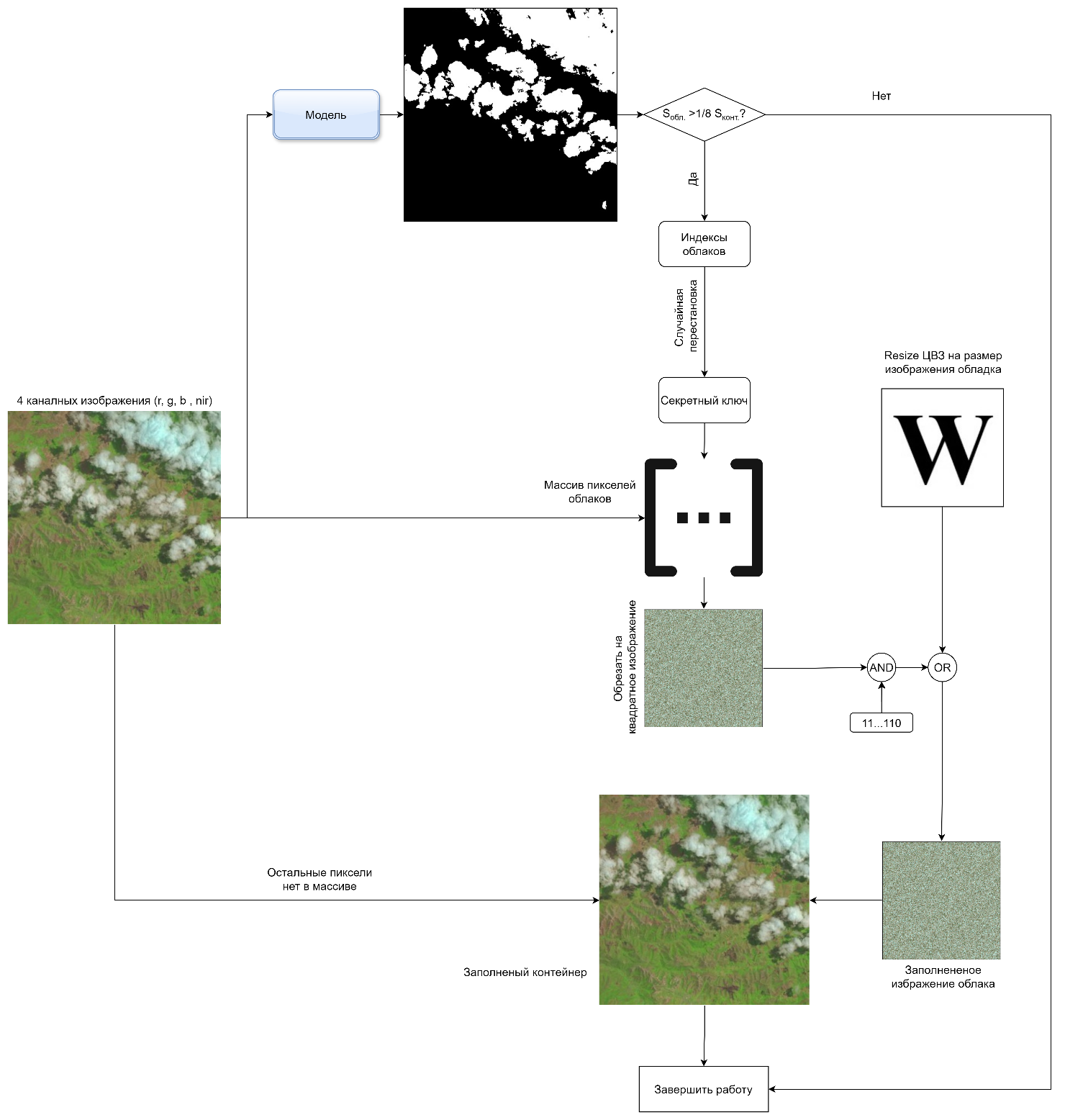


Рис ?. Схема процедуры встраивание

1. Мы будем выбирать тот модель сегментации облаков, который работает с наименьшимошибком.
2. После того, как получить маски, мы считаем отношение площади облака/ площади изображений . Процесс продолжается тогда и только тогда, когда . В противоположном случае, завершить работу.
3. Делаем случайную перестановку полученных индексов пикселей облаков и сохраним результат в качестве секретный ключ.
4. С помощью секретного ключа, получим смешенной массив пикселей облака из контейнера.
5. Обрезать массив до наибольшей длины квадратного числа, получим квадратное изображение смешенного облака.
6. Изменить размер стеганографического изображения на размер квадратного изображения смешенного облака.
7. Применим алгоритм ЦВЗ LSB - Метод замены наименее значимого бита для встраивания стеганографическое изображения в квадратного изображения.

Схема процедуры извлечения:

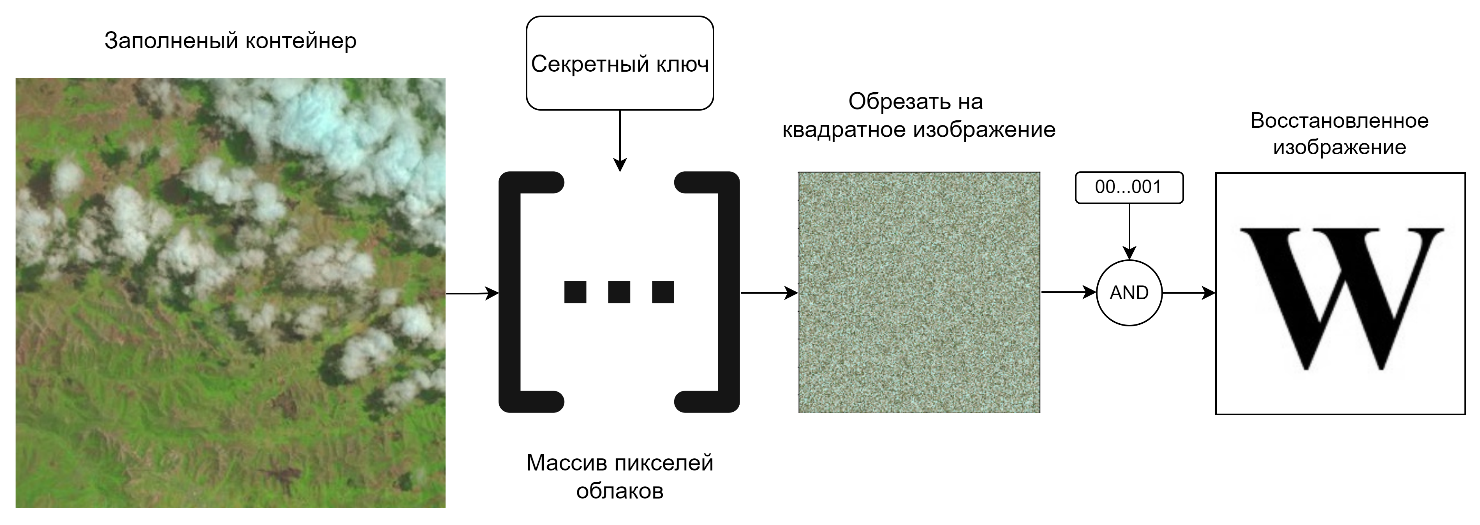


Рис ?. Схема процедуры извлечения

1. С применением секретного ключа определим массив пиксели облаков.
2. Обрезать массив до наибольшей длины квадратного числа, получим квадратное изображение смешенного облака.
3. Применим алгоритм ЦВЗ LSB для извлечения стеганографического изображения

Модель не используется в процессе извлечения, так как после встраивания яркостные значения облаков изменены, что может повлиять на предсказания модели, маска до встраивания и маска до извлечения могут различаться, что может привести к ошибкам при извлечении цифрового водяного знака.

Рассмотрим пример 2 маски до встраивания/извлечения с помощью предсказания модели на контейнер до встраивания/извлечения.

|  |  |  |
| --- | --- | --- |
| Маска до встраивания | Маска до извлечения | Разница |
|  |  | Число не совпадающих  пикселей = 193 |
| Контейнер | Стеганографическое изображение | Восстановленное изображение |
|  |  |  |

Иллюстрация работы правильного метода:

|  |  |  |
| --- | --- | --- |
| Контейнер | Стеганографическое изображение | Предсказанная маска |
|  |  |  |
| Заполненный контейнер | Восстановленное изображение | Разница между контейнерами до и после заполнения |
|  |  |  |