

Borschhevik Detector



СОДЕРЖАНИЕ

1 - Описание проблемы

2 - Техническое решение

2.1 - Загрузка данных из источника

2.2 - Подготовка сырых данных для разметки

2.3 - Создание 2x версий датасета

2.4 - Особенности датасета в torchgeo

2.5 - Выбор модели, лосс-функции и метрик

2.6 - Результаты первых экспериментов

3 - Дальнейшие шаги

Описание проблемы

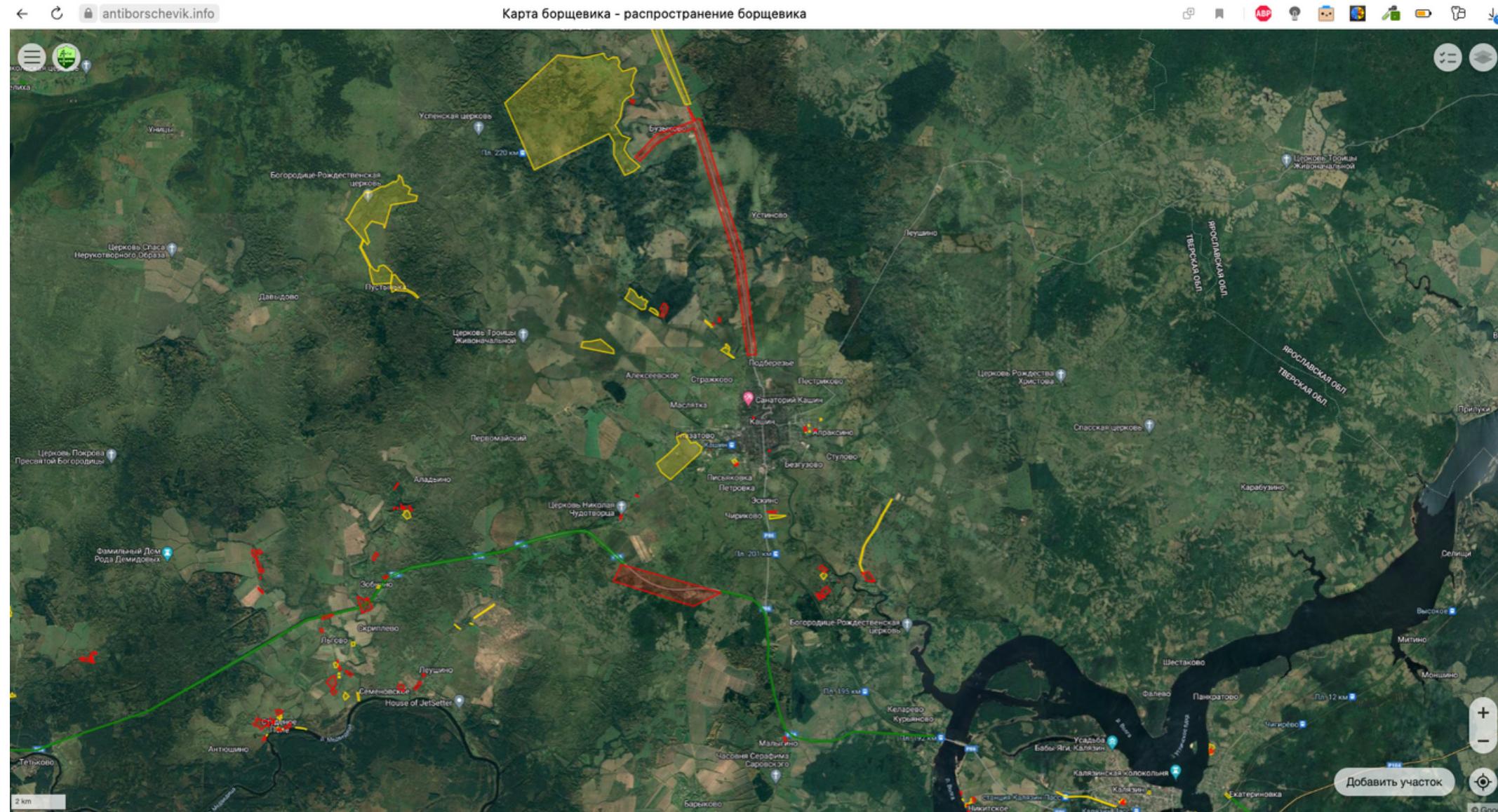


Борщевик — это огромное растение (высотой до 2,5–3 м) с толстыми стеблями и крупными зонтиками из белых цветков. Обычно оно растёт по обочинам дорог, оград дачных участков, на лесных полянах и опушках. Цветёт борщевик примерно в июне — июле, а семена даёт в сентябре. Именно в конце лета происходит наибольшее количество ожогов от этого растения.

Главная опасность борщевика Сосновского в том, что он может спровоцировать сильнейший ожог на коже. Это происходит из-за наличия в соке борщевика особых веществ: кумаринов и фурокумаринов. Именно они повышают чувствительность организма к восприятию солнечного света. Сок под влиянием ультрафиолета усиливает её пигментацию, вызывает сильнейшие дерматиты, которые часто называют специфическими ожогами. Такие ожоги крайне болезненны и могут долго не заживать

Описание проблемы

Что используется в данный момент



Основным драйвером для проверок и выписывания штрафов
являются обращения граждан.

Описание проблемы

Что используется в данный момент

ИСПОЛЬЗОВАНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ

РИСК-ОРИЕНТИРОВАННЫЙ ПОДХОД

На основе анализа 12 критериев определяют земельные участки для проведения осмотров

Земельный фонд Московской области

$$\text{Общий вес земельного участка} = (\sum V_n \cdot v_n) \cdot K_1$$

Составление рейтинга земельных участков от большего веса к меньшему в РГИС

ИСТРЕБЛЕНИЕ БОРЩЕВИКА СОСНОВСКОГО – одна из приоритетных целей.

Критерию присвоен большой вес => начиная с 2020 года каждый участок, на котором выявлено произрастание борщевика, подлежит осмотрам.

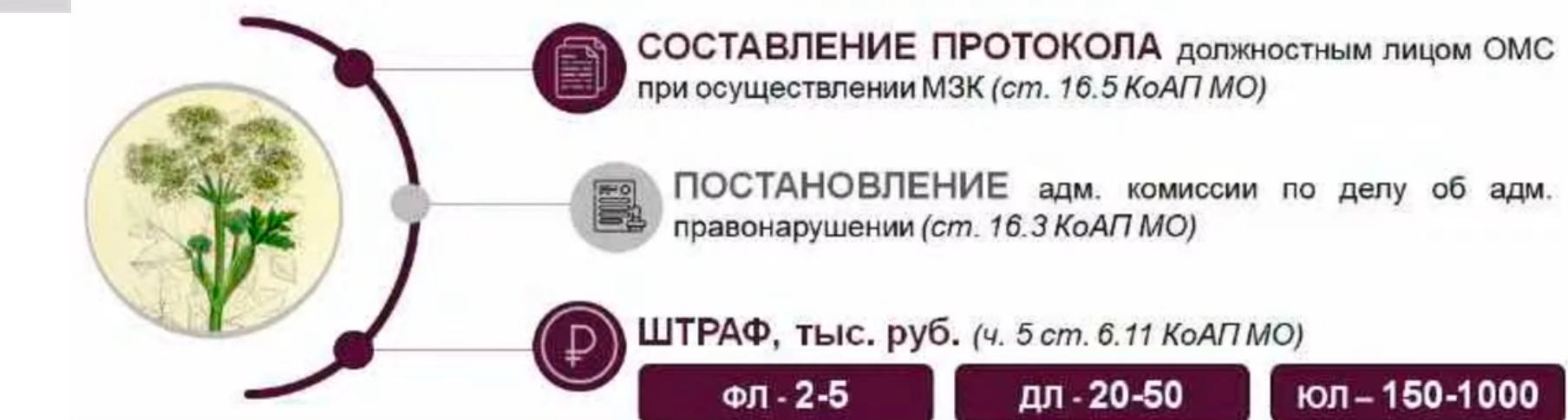
Отображение результатов в РГИС



ПОРЯДОК ПРИВЛЕЧЕНИЯ К АДМИНИСТРАТИВНОЙ ОТВЕТСТВЕННОСТИ

Правообладатели земельных участков обязаны проводить мероприятия по удалению борщевика Сосновского с земельных участков, находящихся в их собственности, владении или пользовании.

(ч. 14 ст. 56 Закона Московской области от 30.12.2014 № 191/2014-ОЗ «О регулировании дополнительных вопросов в сфере благоустройства в МО»)



Техническое решение

Общая схема сервиса

Borschvik Detector - это сервис, который использует машинное обучение для обнаружения борщевика на спутниковых фотографиях. Сервис поможет идентифицировать районы, где присутствует борщевик. Сервис может быть использован государственными учреждениями, аграрными организациями и частными лицами, заинтересованными в контроле и предотвращении распространения борщевика.

Вставить картинку схему

Техническое решение

Функциональность

Выделение области для распознавания

Пользователи смогут выделить район интереса для обнаружения наличия борщевика

Генерация слоя

После обработки район интереса, сервис сгенерирует новый слой для отображения на карте, содержащий информацию о том, где были обнаружены скопления борщевика

Обнаружение борщевика

Сервис использует алгоритм машинного обучения для определения наличия борщевика на загруженных изображениях

Карта распространения *

Сервис может использоваться для создания карт, показывающих распространение борщевика, что облегчает отслеживание распространения растения с течением времени

*Возможно будет реализовано в будущем

Техническое решение

Загрузка данных из источника



Придется создать его самим

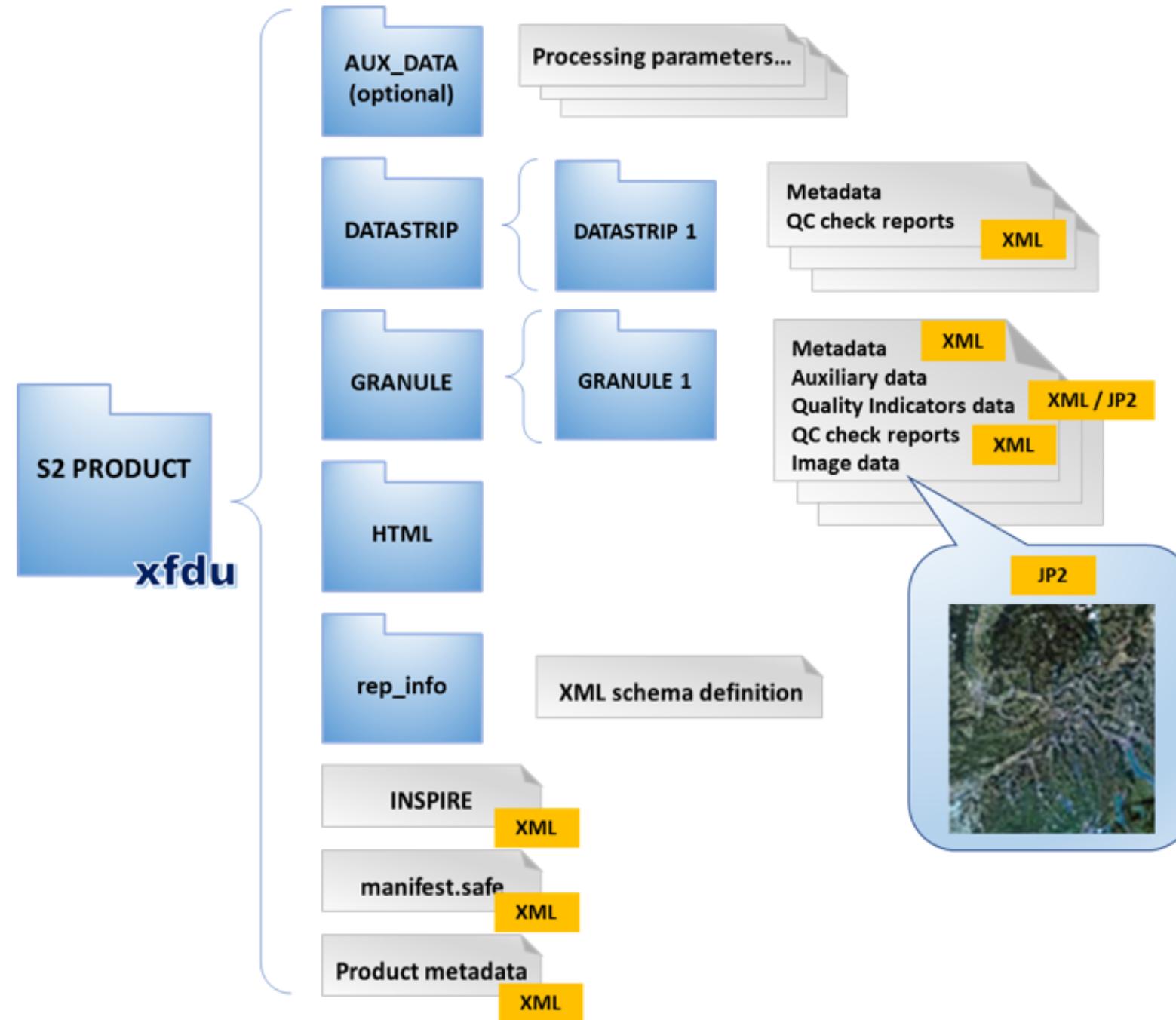
Где можно взять бесплатные спутниковые снимки:

- 1 - Copernicus Open Access Hub
- 2 - Earth Engine Data Catalog from Google
- 3 - Planetary Computer (Microsoft)
- 4 - Договориться с Роскосмосом



Техническое решение

Структура сырых данных



Имя	Размер
T37VCD_20220703T084611_AOT_10m.jp2	565 КБ
T37VCD_20220703T084611_B02_10m.jp2	98 154 КБ
T37VCD_20220703T084611_B03_10m.jp2	109 163 КБ
T37VCD_20220703T084611_B04_10m.jp2	99 029 КБ
T37VCD_20220703T084611_B08_10m.jp2	132 038 КБ
T37VCD_20220703T084611_TCI_10m.jp2	132 364 КБ
T37VCD_20220703T084611_WVP_10m.jp2	91 820 КБ

Каждый band - картинка

10980 x 10980

Band04 - Red Channel

Band03 - Green Channel

Band02 - Blue Channel

Band08 - NIR Channel

Техническое решение

Обработка сырых данных

Шаги данного этапа:

1 - Обработать и собрать из сырых данных:

- "привычное" изображение **RGB** (size 10980x10980)
- изображение **RGNB** (size 10980x10980)
- шаблон маски для разметки (size 10980x10980)

2 - Разбить **RGB & template Mask** на мелкие кусочки размером 512x512

3 - Разметить данные

4 - "Нарисовать" правильные кусочки масок, согласно разметки

5 - Собрать из кусочков маски одну большую **Mask**

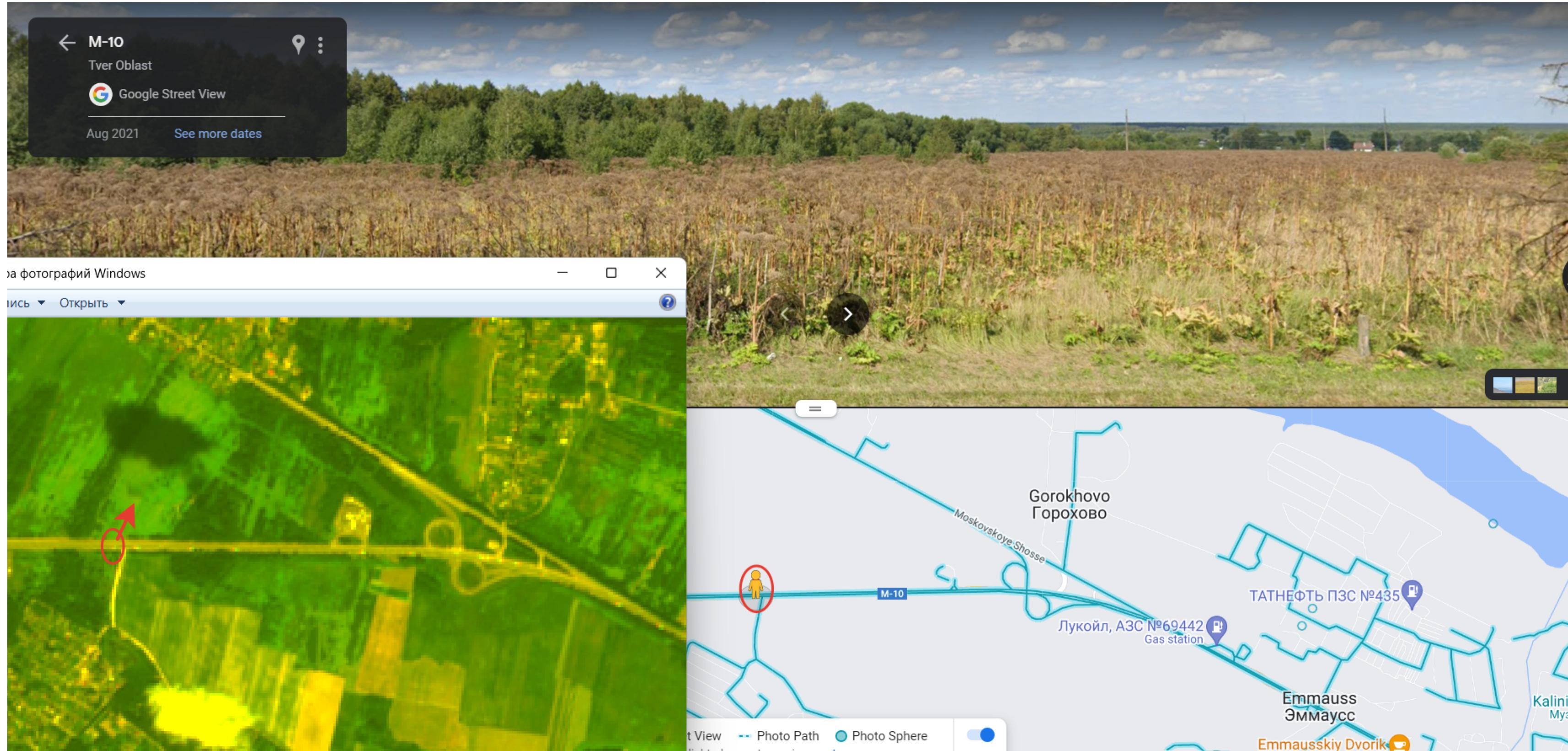
6 - Создать **2 версии** датасета:

- для классического пути
- для torchgeo

Примечание: сложность всех манипуляций осложняется наличием координатной сетки, формат всех файлов **Geotiff**

Техническое решение

Разметка и Наш Fail



Техническое решение

Разметка и Наш Fail



```
for i, band in zip(range(1, len(list_of_bands) + 1), list_of_bands):  
    img.write(band, i)  
  
img.close()
```

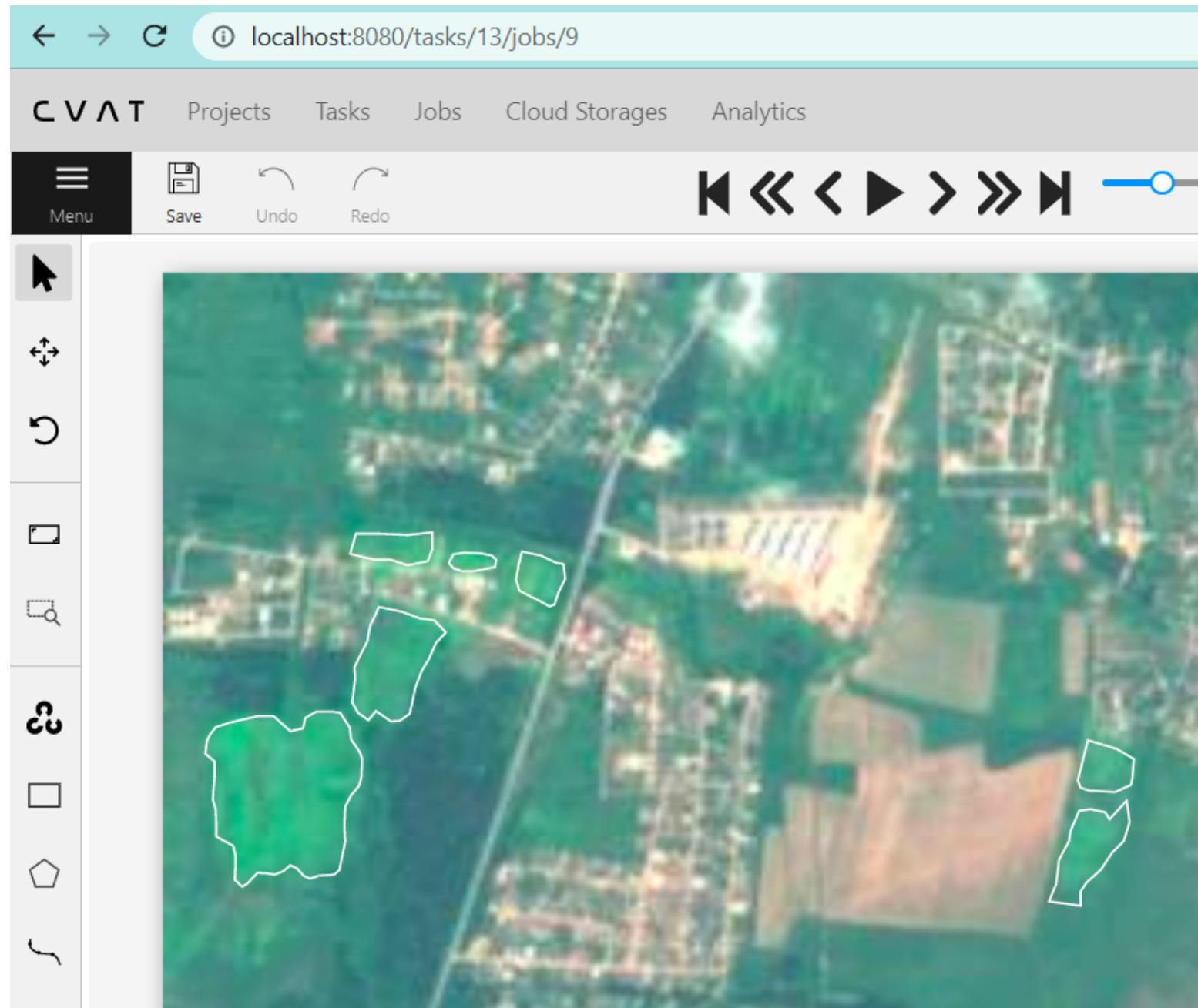


Все делают ошибки, а
большинство повторяют одни
и те же.

(Сара Линден)

Техническое решение

Разметка и Наш Fail



Техническое решение

Особенности датасета в torchgeo - создание Rasterdataset

```
#res - resolution, like the scale of the image, 10 means 10 m in pixel, as it is in Sentinel-2
train_imgs = RasterDataset(root=(root/'img_f/train').as_posix(), crs='epsg:32637', res=10, transforms=scale)
train_msks = RasterDataset(root=(root/'mask_f/train').as_posix(), crs='epsg:32637', res=10)

valid_imgs = RasterDataset(root=(root/'img_f/val').as_posix(), crs='epsg:32637', res=10, transforms=scale)
valid_msks = RasterDataset(root=(root/'mask_f/val').as_posix(), crs='epsg:32637', res=10)

# IMPORTANT
train_msks.is_image = False
valid_msks.is_image = False

train_dset = train_imgs & train_msks
valid_dset = valid_imgs & valid_msks

train_sampler = RandomGeoSampler(train_imgs, size=256, length=600, units=Units.PIXELS)
valid_sampler = RandomGeoSampler(valid_imgs, size=256, length=300, units=Units.PIXELS)

train_dataloader = DataLoader(train_dset, sampler=train_sampler, batch_size=4, collate_fn=stack_samples)
valid_dataloader = DataLoader(valid_dset, sampler=valid_sampler, batch_size=4, collate_fn=stack_samples)

train_batch = next(iter(train_dataloader))
valid_batch = next(iter(valid_dataloader))
```

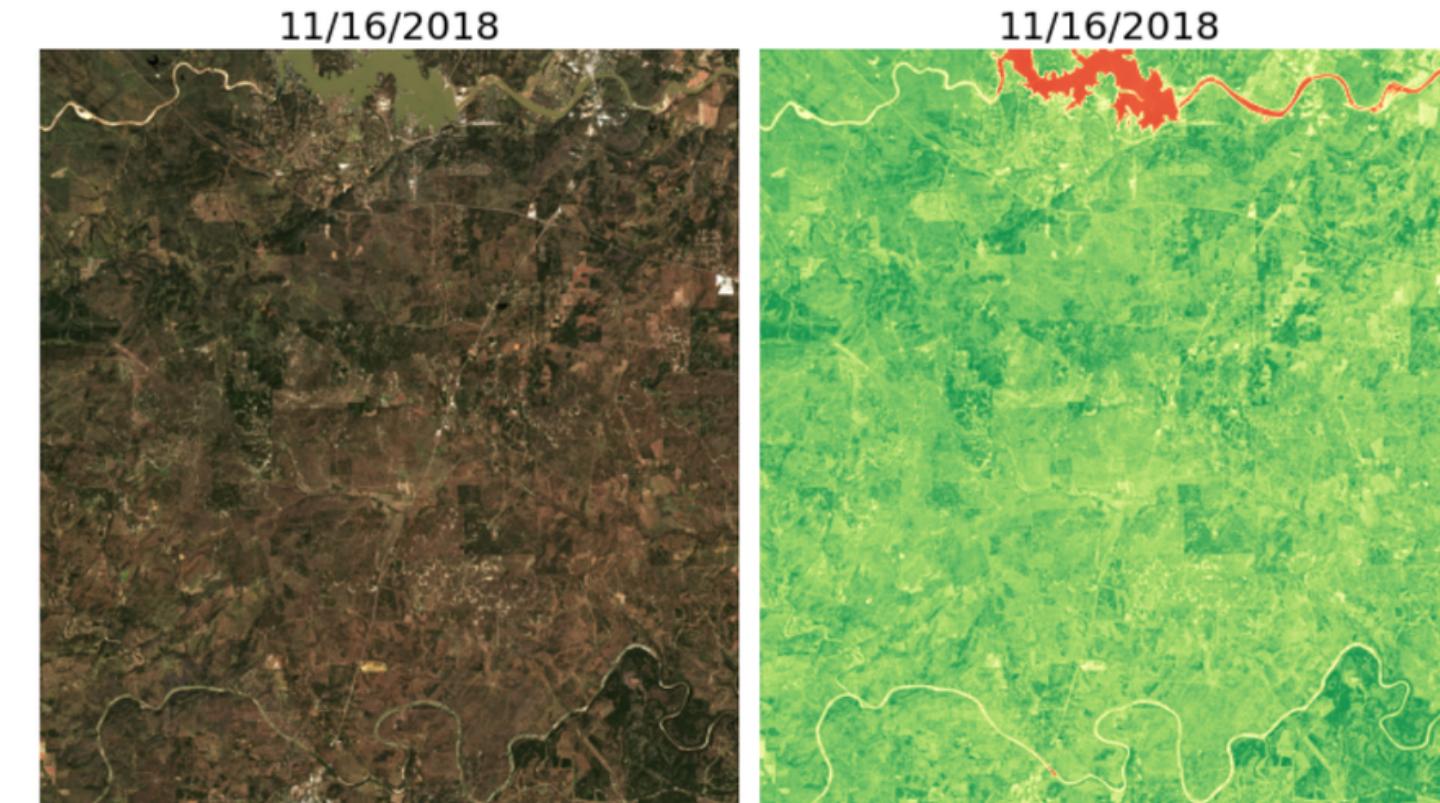
Техническое решение

Особенности датасета в torchgeo - добавление индексов

Normalized Difference Vegetation Index (NDVI)

- нормализованный относительный индекс растительности - простой количественный показатель количества фотосинтетически активной биомассы (обычно называемый вегетационным индексом). Один из самых распространенных и используемых индексов для решения задач, использующих количественные оценки растительного покрова.

$$\text{NDVI} = \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + \text{Red})}$$



```
tfms = torch.nn.Sequential(
    indices.AppendNDVI(index_nir=3, index_red=0),
    normalize
)
```

Техническое решение

Выбор модели, лосс функции и метрики

Модель - DeepLabV3

Loss - на данный момент

Cross Entropy, заменим на Dice

Метрика - IoU

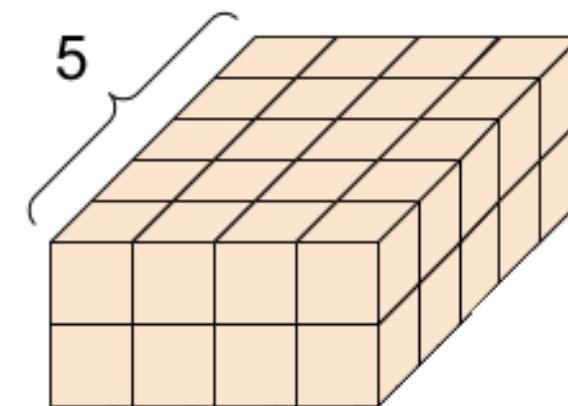
Особенность torchgeo:

В нашем случае подаем

тензор размерностью

[batch_size, 5, img_h, img_w]

- Red ch
- Green ch
- Blue ch
- NIR ch
- NDVI ch



Поэтому меняем 1 слой сети

```
backbone = model.get_submodule('backbone')

conv = torch.nn.modules.conv.Conv2d(
    in_channels=5,
    out_channels=64,
    kernel_size=(7, 7),
    stride=(2, 2),
    padding=(3, 3),
    bias=False
)
backbone.register_module('conv1', conv)
```

```
pred = model(torch.randn(4, 5, 256, 256))
pred['out'].shape
```

torch.Size([4, 2, 256, 256])

Вектор на выходе

Техническое решение

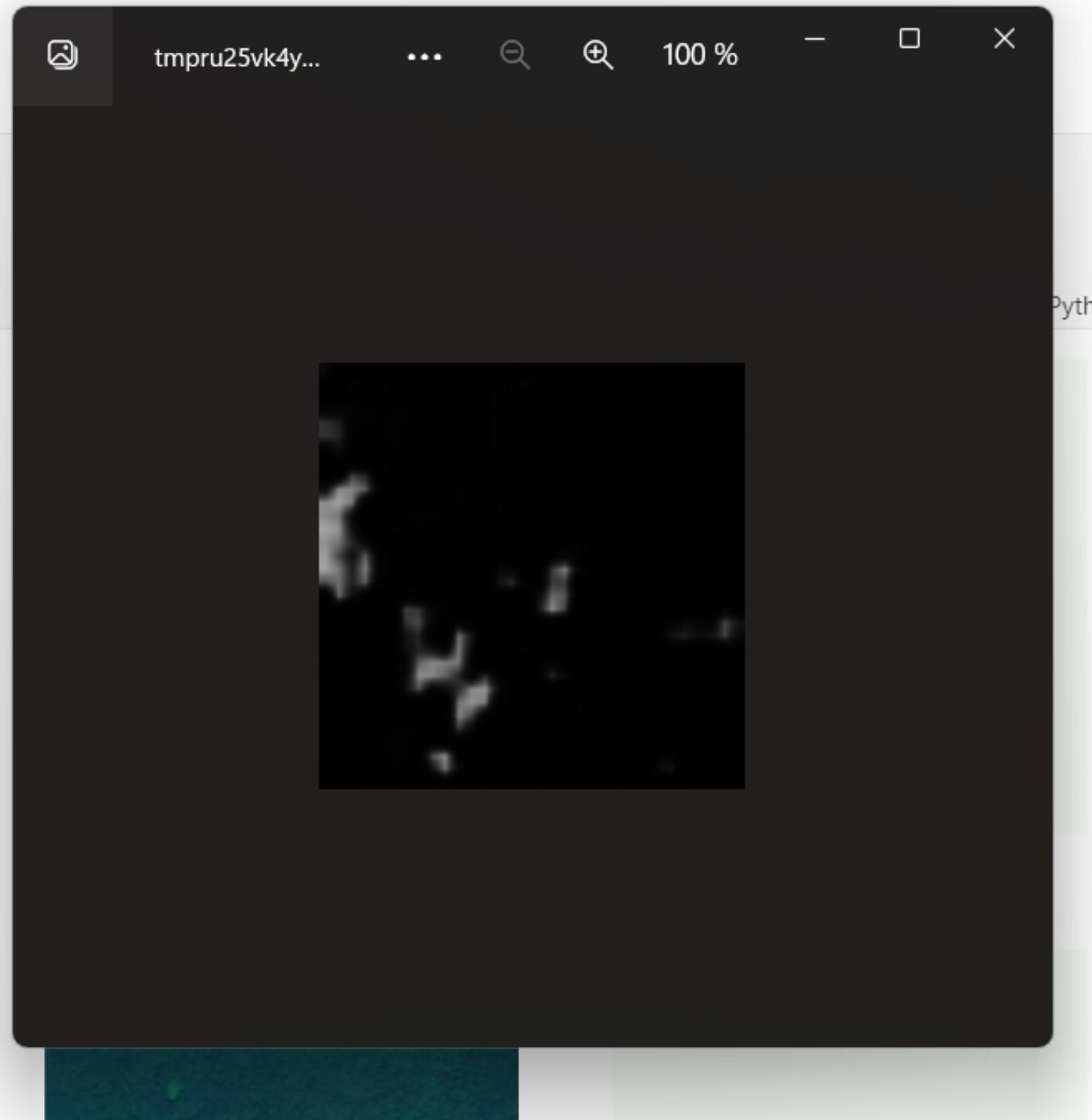
Результаты первых экспериментов

Сравнение результатов

```
valid_batch = next(iter(valid_dataloader))
try_batch = tfms(valid_batch['image'])
plot_batch(valid_batch)
```

[94]

✓ 1.0s



Дальнейшие шаги

1. Сделать интеграцию с ClearML
2. Провести ряд экспериментов (смена лосс функции, аугментация, подбор гиперпараметров и т.д.)
3. Реализовать web-сервис (базовый функционал)
4. Добавление новых функций сервису



Спасибо!



