

Решение задачи Dstl Satellite Imagery Feature Detection

(мастер-класс)

Евгений Некрасов

Введение



Задача: Сегментация спутниковых изображений

Платформа: kaggle.com

Организатор: Defence Science and Technology Laboratory (UK)

Провайдер спутниковых изображений: DigitalGlobe, Inc

Результат: 7 место из 419 команд



Данные

- 25 спутниковых изображений в Train
- 425 спутниковых изображений в Test
- Векторная разметка изображений Train в формате WKT или GeoJSON
- 10 классов объектов
- Формат сабмишена WKT
- Все изображения отсняты в одном регионе
- Изображения представляют участки поверхности земли 1x1 км
- Для каждого участка 1x1 км даны 4 файла tiff с разных приборов

Изображения



Название	Кол-во каналов	Пространственное разрешение	Динамический диапазон
RGB	3	0.31 м	11-bits per pixel
Panchromatic	1	0.31 м	11-bits per pixel
Multispectral	8	1.24 м	11-bits per pixel
SWIR	8	7.5 м	14-bits per pixel

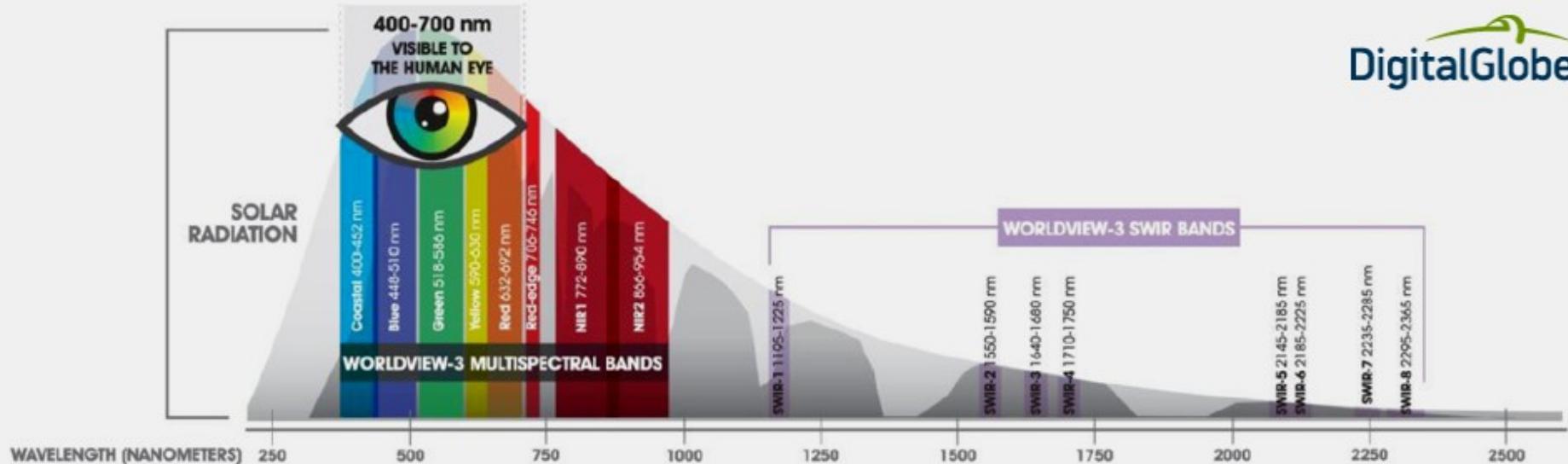


WorldView 3



Multispectral & SWIR

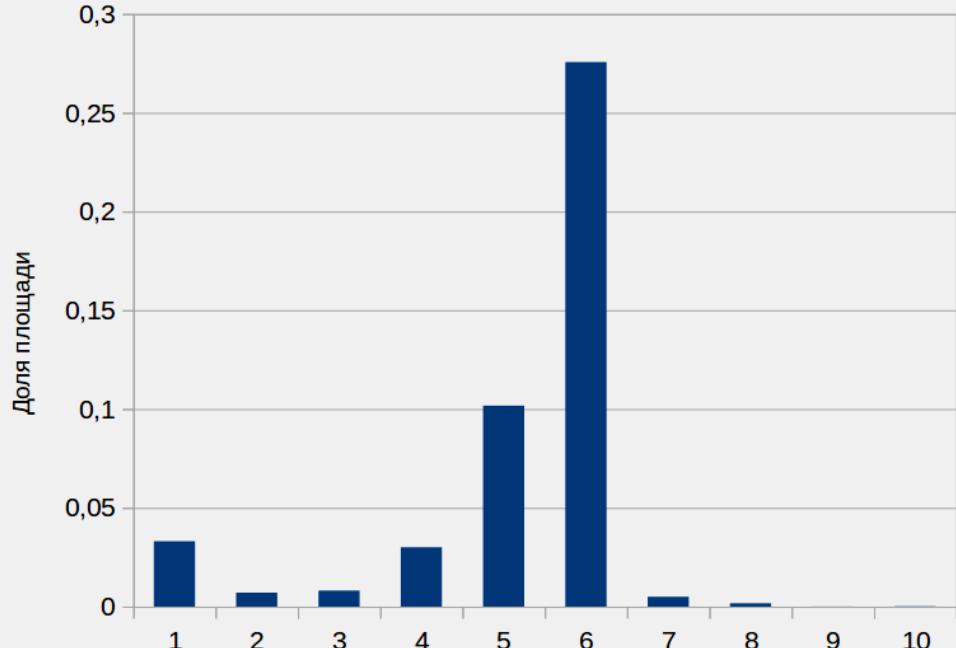
DigitalGlobe™



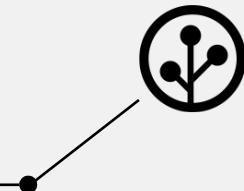
Разметка



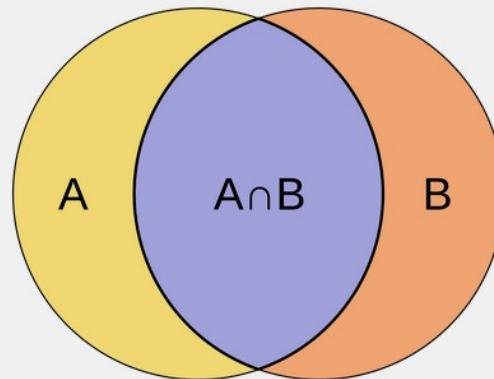
1. Buildings
2. Misc. Manmade structures
3. Road
4. Track
5. Trees
6. Crops
7. Waterway
8. Standing water
9. Vehicle Large
10. Vehicle Small



Метрика качества

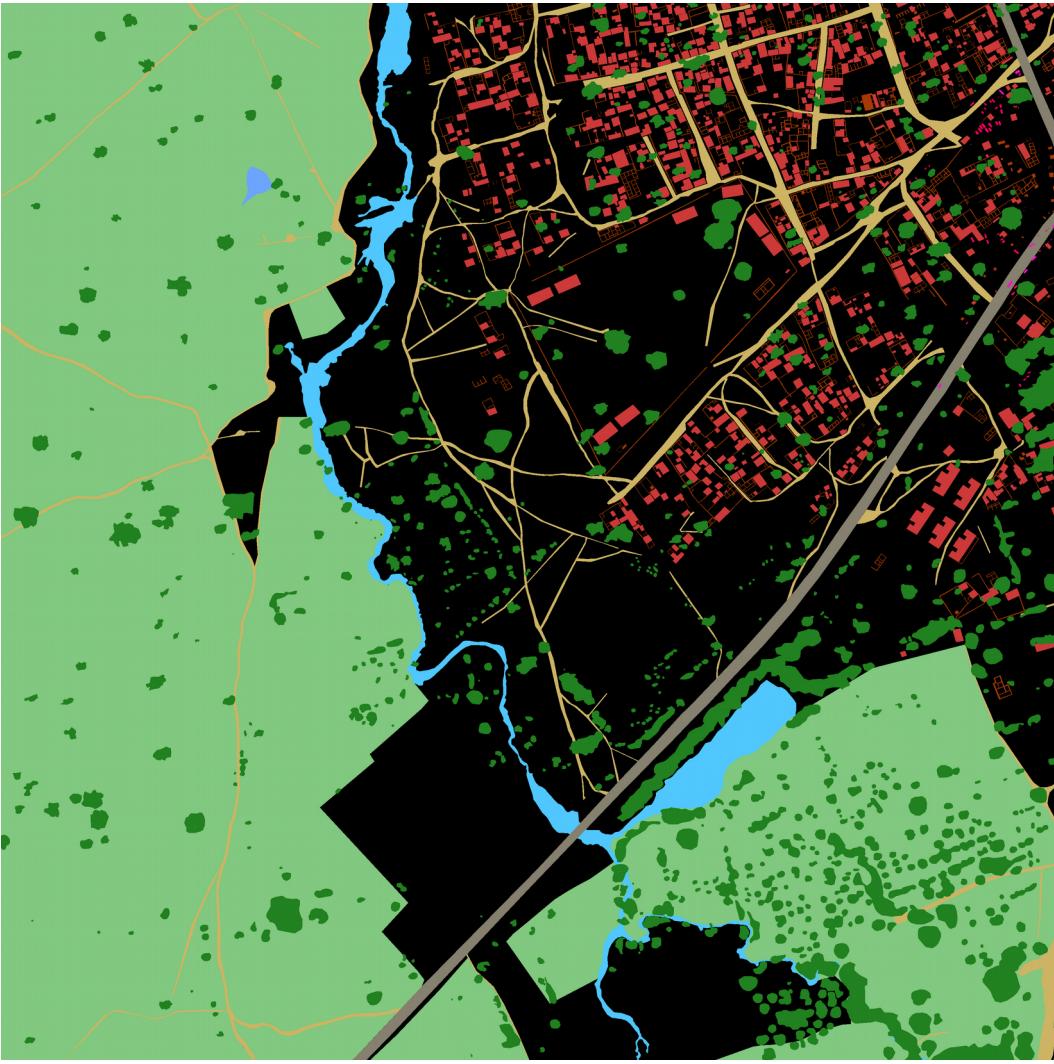


$$Jaccard = \frac{TP}{TP + FP + FN} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$



Jaccard считали для каждого из 10 классов, после чего усредняли





Решение



Препроцессинг

3-Channel RGB

1-Channel Panchromatic

8-Channel Multispectral

8-Channel SWIR

→ 20-Channel Image

1. Нормировал на максимум динамического диапазона
2. Масштабировал к Panchromatic, Multispectral и SWIR к размеру RGB
3. Объединял в одно 20-канальное изображение

WKT vector masks → 10-Channel binary images

1. Перевел векторную разметку в растровые 10-канальные бинарные маски размера изображений RGB

Моделирование



Input: 160x160

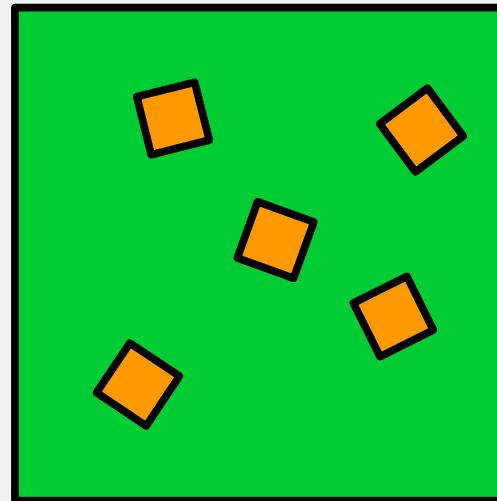
Augmentation: Rotational crops

Architecture: U-Net-like

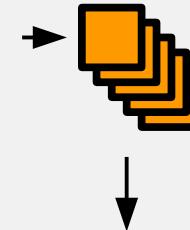
Objective: binary_crossentropy

Optimizer: Adam

~3300x3300px



160x160px

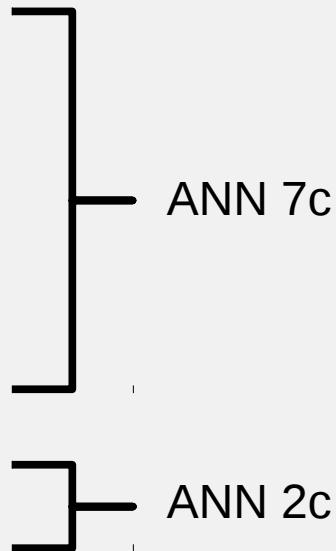


U-Net-like
ANNs

Нейронные сети 7с и 2с



- 1. Buildings
- 2. Misc. Manmade structures
- 3. Road
- 4. Track
- 5. Trees
- 6. Crops
- 7. Waterway —→ 7. Water
- 8. Standing water
- 9. Vehicle Large
- 10. Vehicle Small



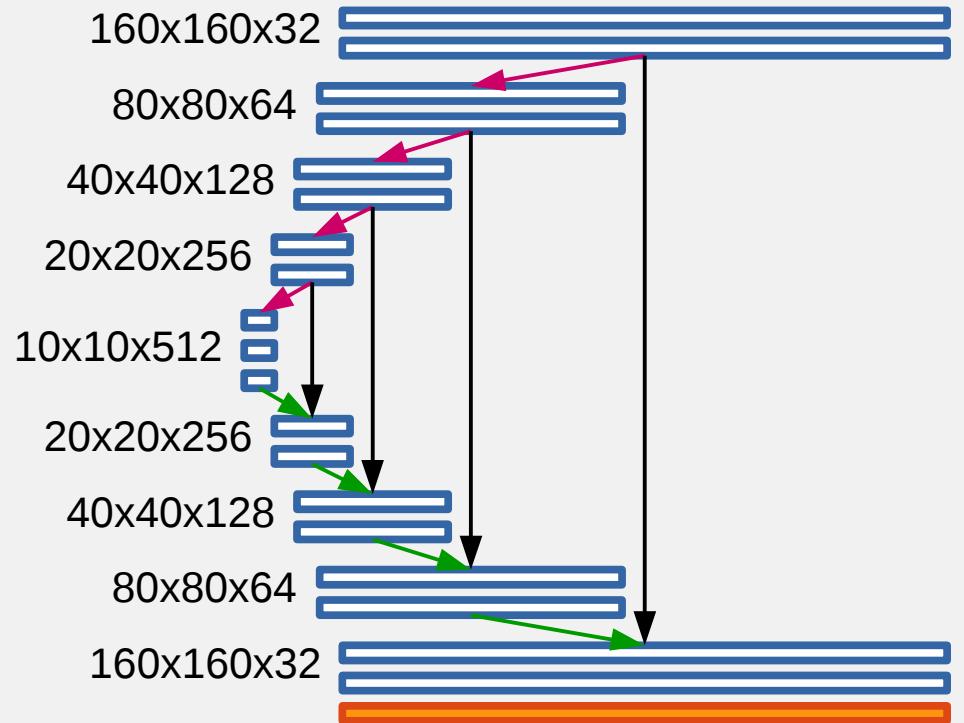
Архитектура нейросети



- [Blue Box] Conv2D 3x3
BN
ReLU
- [Orange Box] Conv2D 1x1
Sigmoid
- [Pink Arrow] MaxPool2D 2x2
- [Black Arrow] Concat
- [Green Arrow] UpSample2D 2x2

Input: 160x160x20

Output: 160x160x7 or 160x160x2



Тренировка нейросетей



7с

1. Был сконструирован автоэнкодер, совместимый с 7с
2. Автоэнкодер был обучен на 600к кропов с Train + Test (с 450 картинок)
3. Веса энкодера были перенесены в 7с и зафиксированы
4. Обучение на 400к вращательных кропов
5. Веса энкодера были отпущены
6. Обучение на 600к вращательных кропов

2с

1. Обучение на 200к вращательных кропов с сэмплированием кропов, в которых объекты присутствуют с вероятностью ~50%
2. Обучение на 700к вращательных кропов

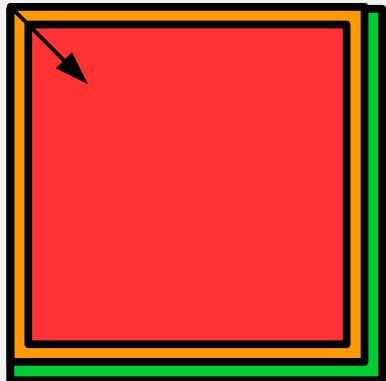
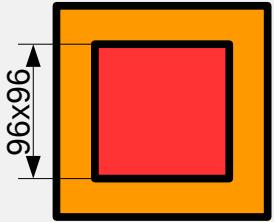
Предсказание

По одному проходу от каждого угла

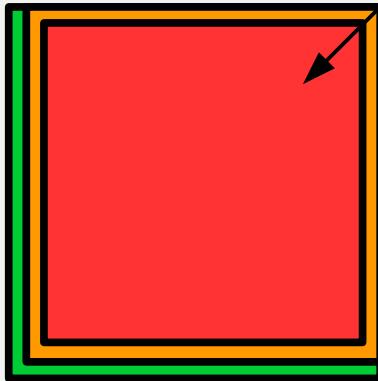
Использование преимущественно центральной части кропа



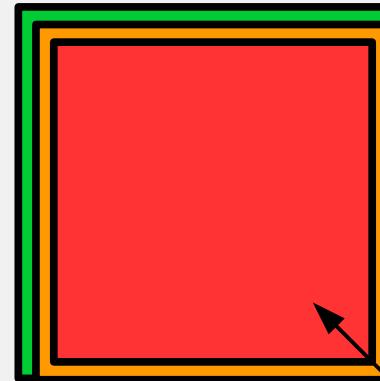
160x160



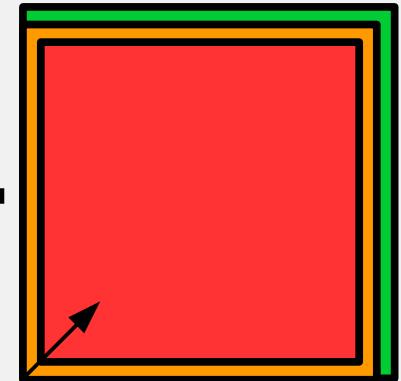
+



+



+





Пост-процессинг

Построение бинарных масок:

1. Предсказывал нейросетями на Train
2. Максимизировал Jaccard относительно порога дискретизации

Класс	Порог бинаризации
1. Buildings	0.44
2. Misc. Manmade structures	0.14
3. Road	0.58
4. Track	0.34
5. Trees	0.38
6. Crops	0.5
7+8. Water	0.42
9. Vehicle Large	0.37
10. Vehicle Small	0.3

Waterway CCCI model



by Vladimir Osin

Canopy Chlorophyl Content Index (CCCI)

$$\text{CCCI} = (\text{MIR-RE})/(\text{MIR+RE}) * (\text{MIR-R})/(\text{MIR+R})$$

RG



Binary



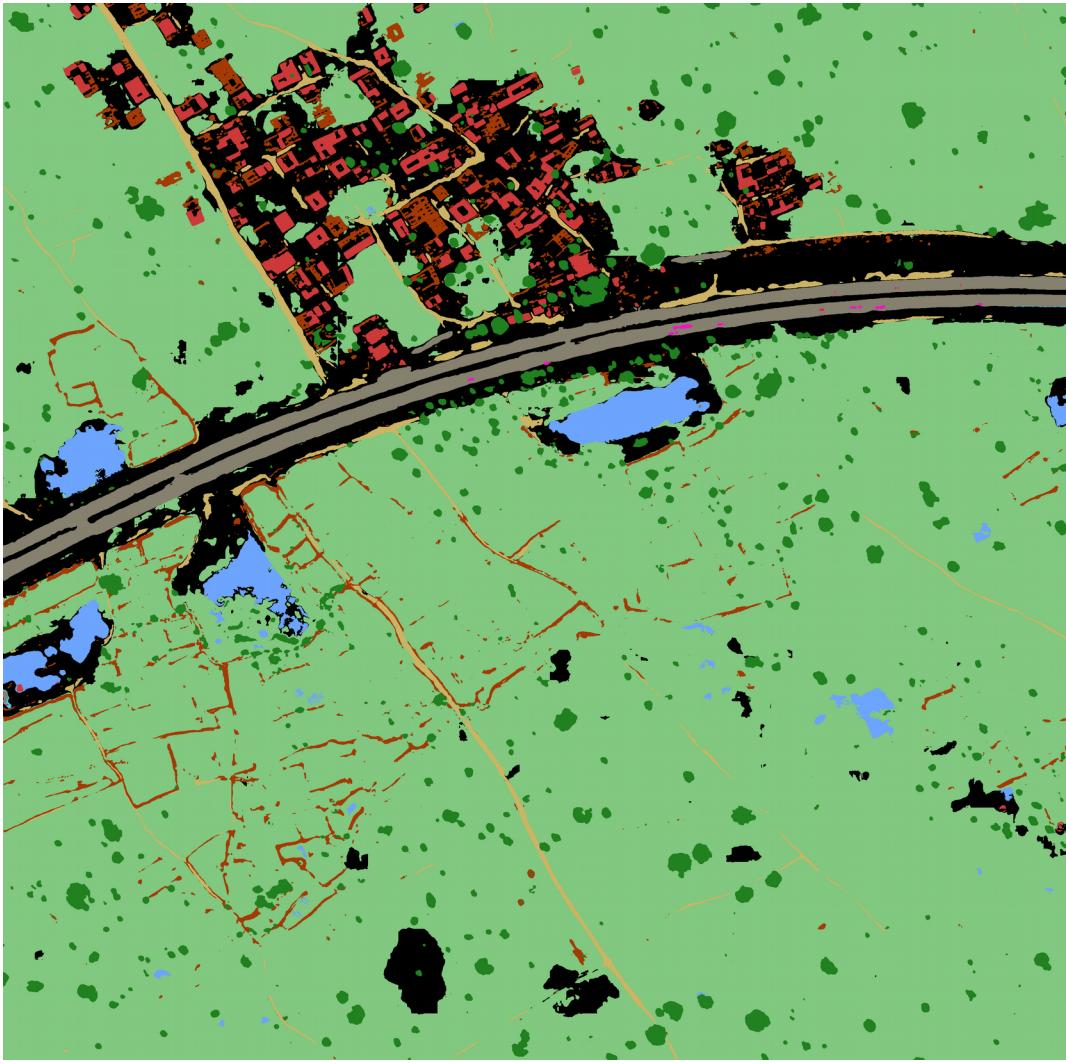
Пост-процессинг



Разделение Water на Waterway и Standing water

1. для каждой области воды вычислял параметры:
 - площадь
 - эллиптичность
 - касание краев
2. Считал линейную комбинацию этих параметров
3. Классифицировал области по порогу





Заключение



Результаты

Public Score: 0.51725

Private Score: 0.43897

Rank: 7

Ключевые идеи

Архитектура U-Net

Аугментация поворотом на произвольный угол

Использование тестовых изображений для моделирования

Применение классических методов CV для классов воды

Техническая часть



Железо:

GeForce GTX 1080
Intel Core i7
16 Gb RAM

Софт:

Python 3
Keras with Tensorflow backend

Послесловие



Недавние соревнования, в которых команды, занявшие первое место, использовали U-net подобные архитектнурьи:

- **Understanding Clouds from Satellite Images**, Nov 2019
- **Severstal: Steel Defect Detection**, Oct 2019
- **SIIM-ACR Pneumothorax Segmentation**, Sep 2019

Оригинальная статья:

Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation; Medical Image Computing and Computer-Assisted Intervention 2015.

https://doi.org/10.1007/978-3-319-24574-4_28

Спасибо за
внимание!