



Сегментация объектов на спутниковых снимках (Kaggle Dstl Satellite Imagery Feature Detection)

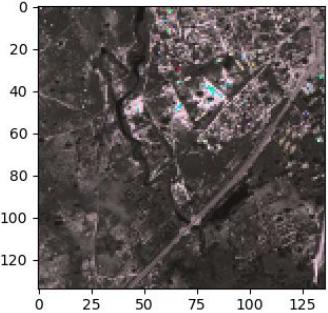
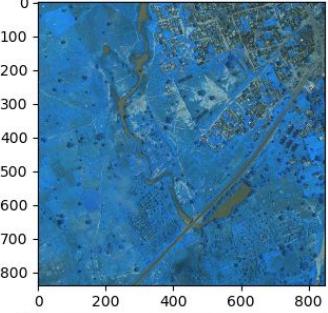
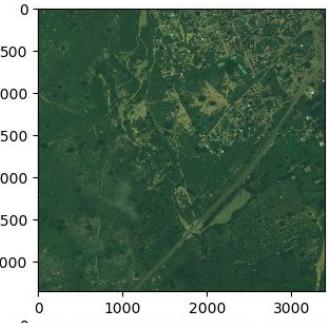


Артур Кузин
kaggle.com/drn01z3
ODS: @n01z3



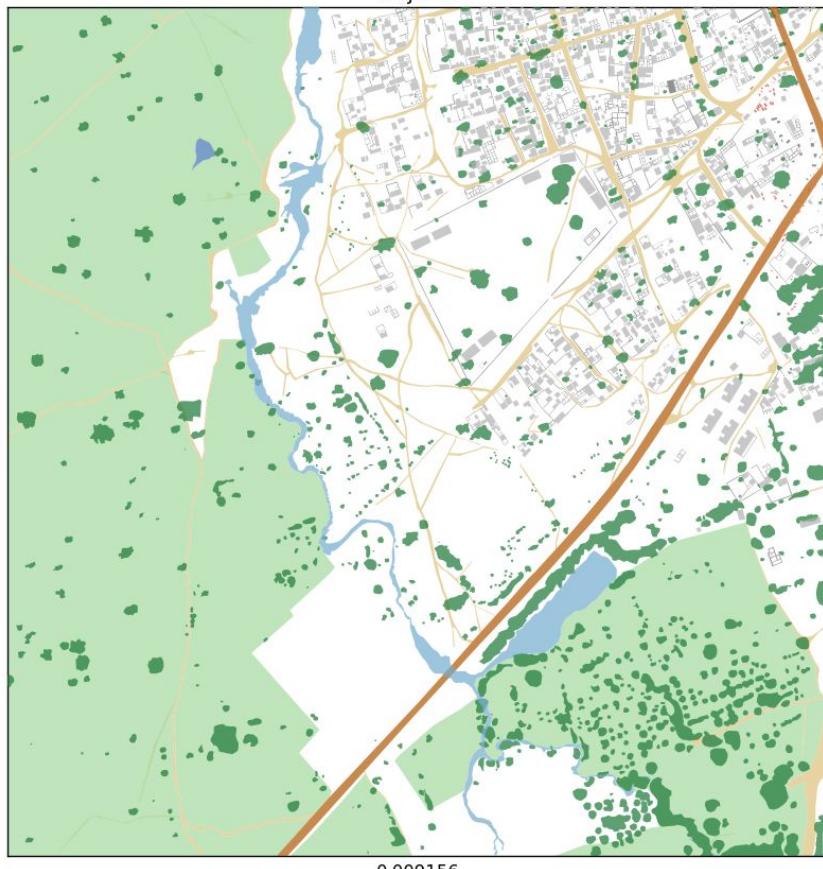
Роман Соловьев,
Ph.D.
kaggle.com/zfturbo
ODS: @zfturbo

Постановка задачи



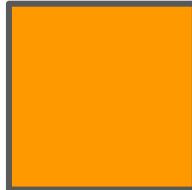
Objects-6100_2_2	
Bldg (633)	Road (1)
Struct (454)	Track (27)
	Trees (878)
	Crops (11)
	Fast H20 (4)
	Slow H20 (1)
	Truck (13)
	Car (101)

Objects

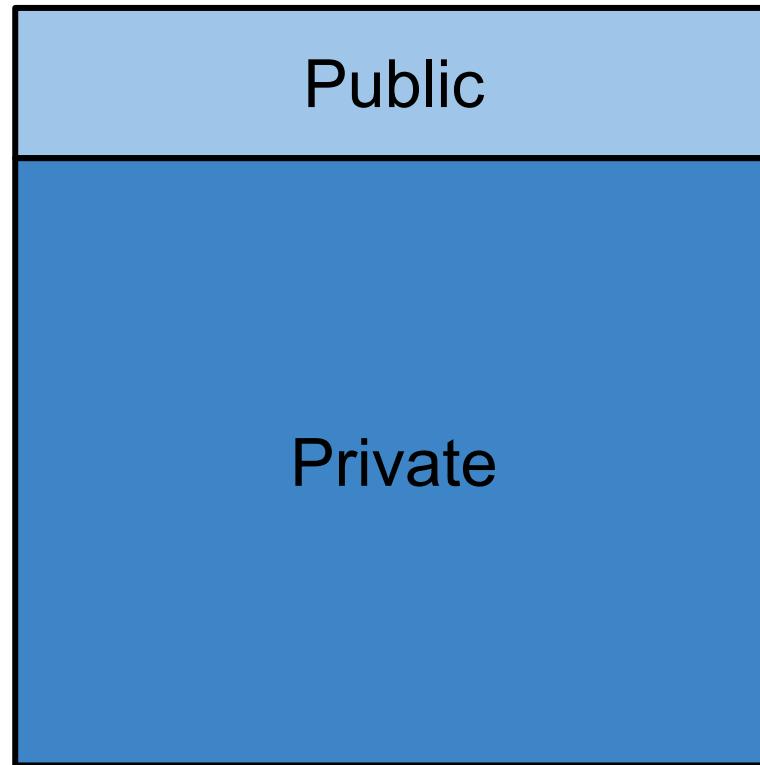


Размер данных

Train
25 samples



Fake Test: 425 samples

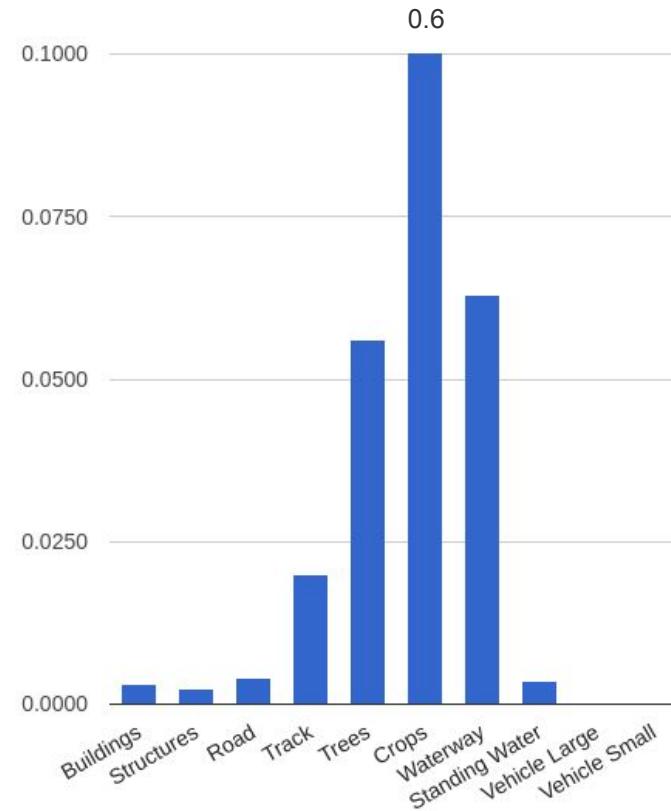
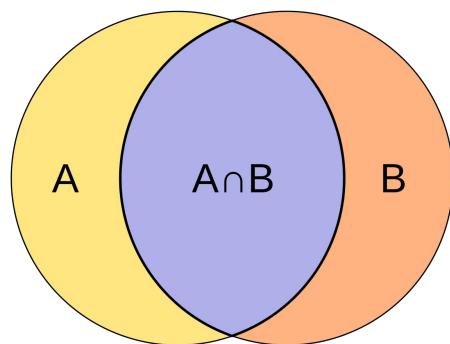


Real Test
32 samples



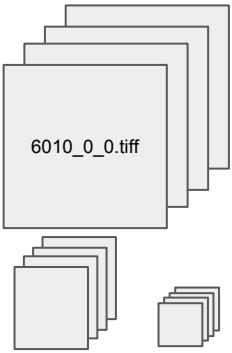
Метрика: Average Jaccard Similarity

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

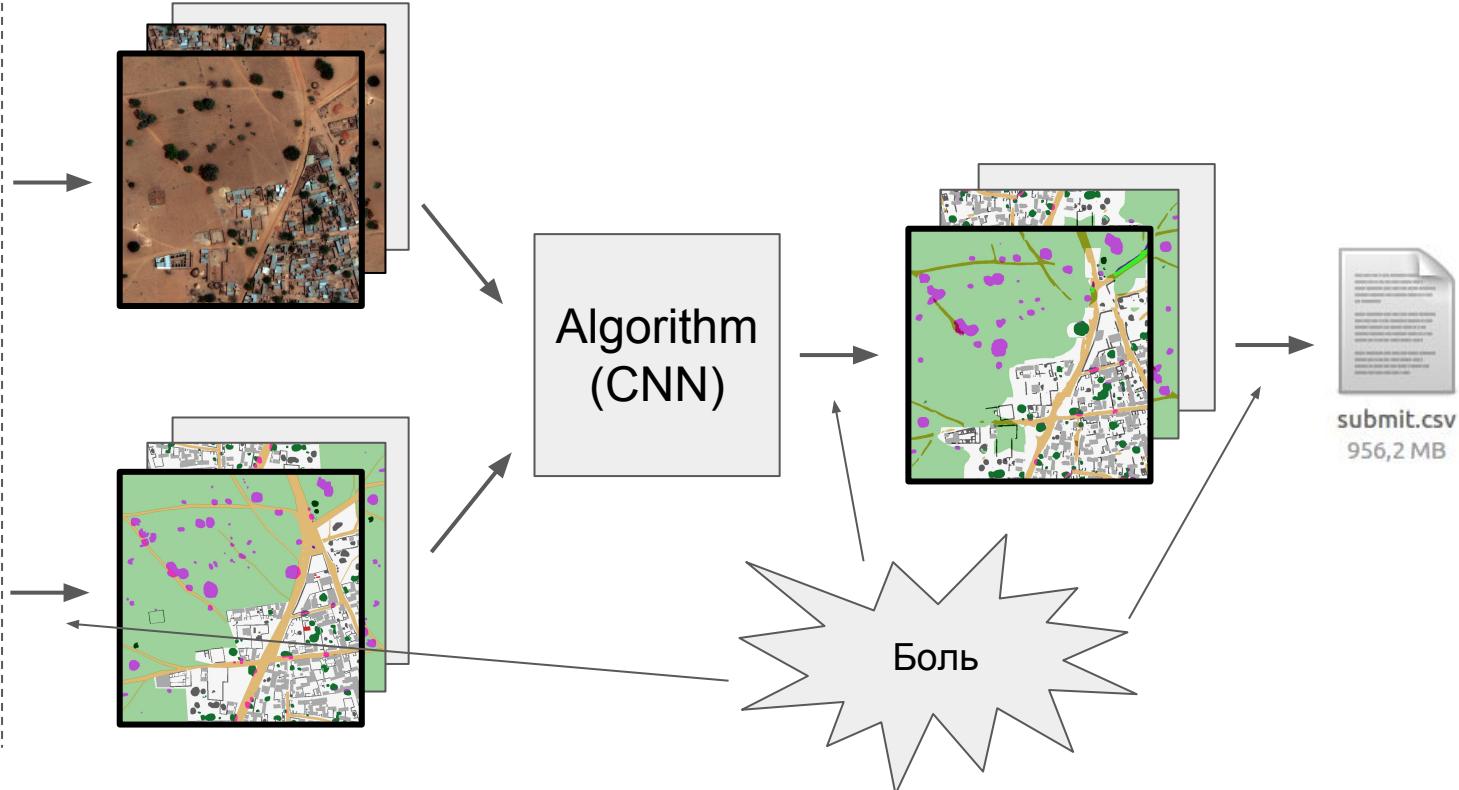


Особенности соревнования

Дано:

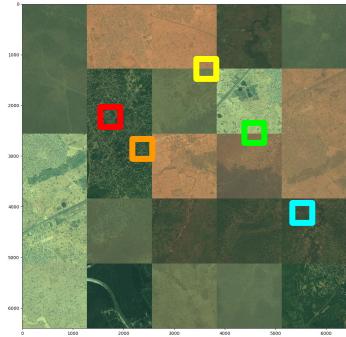


Что нужно сделать:

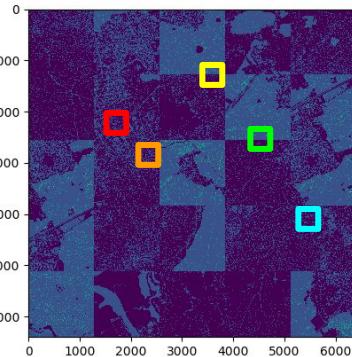
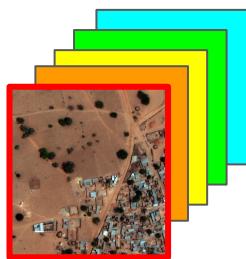


Baseline

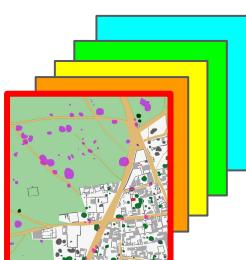
Train



X



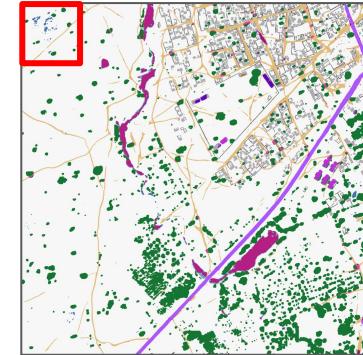
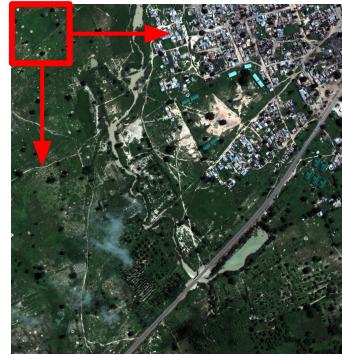
Y



U-net

Predict

Sliding window



submit.csv
956,2 MB

Validation: 



Full pipeline demo: poly -> pixels -> ML -> poly
run 2 months ago by Konstantin Lopuhin

Expectation vs Reality

25 new cepera.ang

Your Best Entry ↑

Your submission scored 0.27684, which is not an improvement of your best score. Keep trying!

VS

Scoring should take about 0 seconds

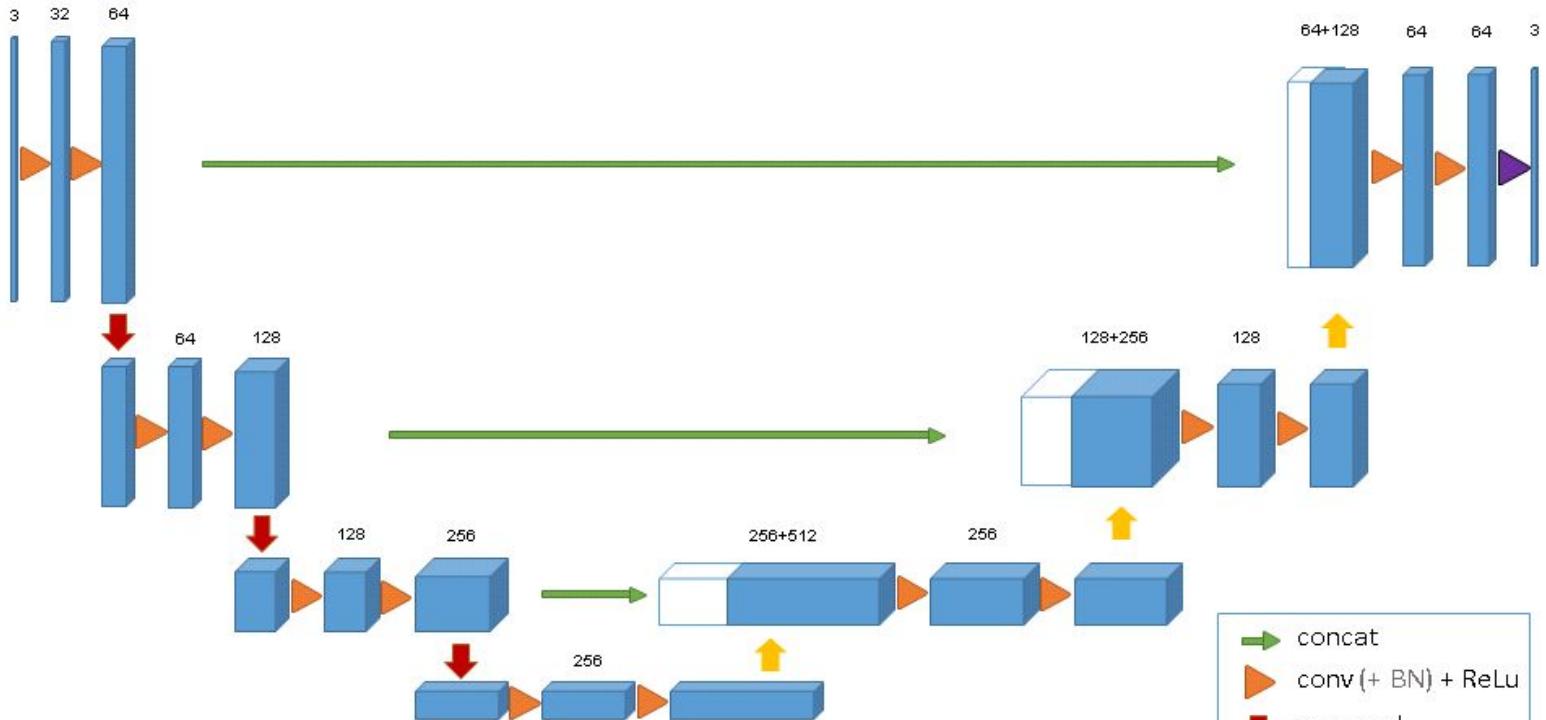
Evaluation Exception: For image 6060_0_1, class 1, there's an exception in geometry

NetTopologySuite.Geometries.TopologyException: side location conflict [(0.00189164444677645, -0.00054066908470115, NaN)] at

Kaggle.Metrics.Custom.JaccardDSTL.compareMultipolygon(IGeometry predMultipoly, IGeometry truthMultipoly) at Kaggle.Metrics.Custom.JaccardDSTL.calculateIoUVector(Dictionary`2 submission, Dictionary`2 solution)..

[Try again or go to submissions page](#)

U-net

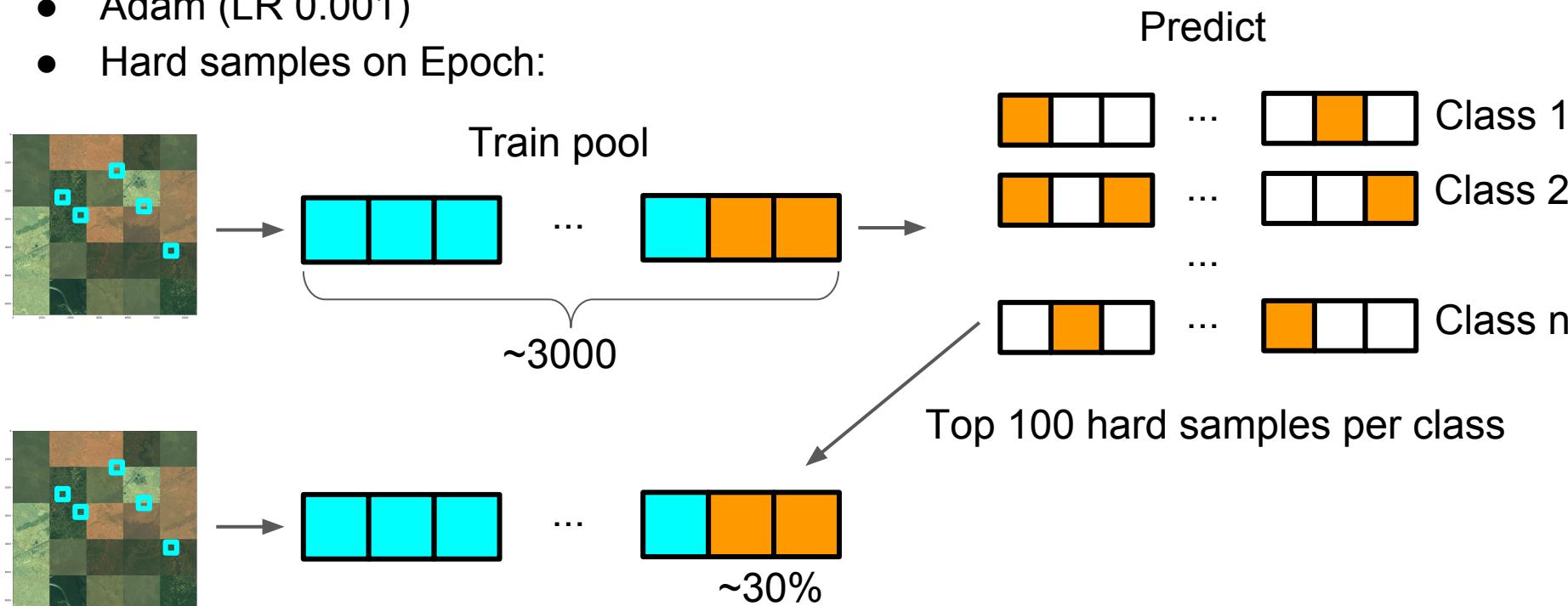


- concat
- conv (+ BN) + ReLu
- max pool
- up-conv
- conv

U-Net: Convolutional Networks for Biomedical Image Segmentation
arXiv:1505.04597 [cs.CV]
github.com/jocicmarko/ultrasound-nerve-segmentation

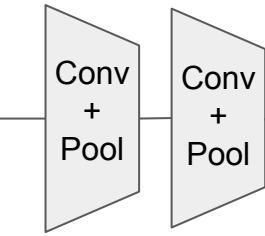
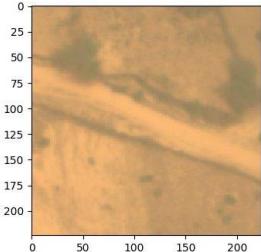
Улучшения базового решения

- Conv + ReLu → Conv + BN + ReLu
- Augs: Rotation, Mirror, Flip
- Adam (LR 0.001)
- Hard samples on Epoch:

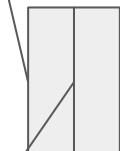


Multi input

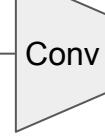
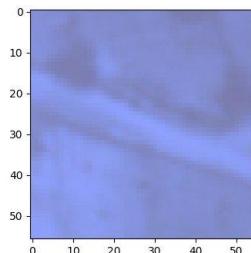
RGB+P, 224 x 224



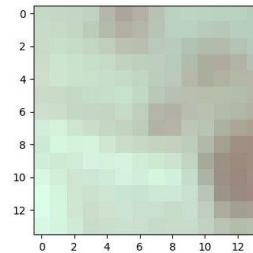
Concat



M band
56 x 56



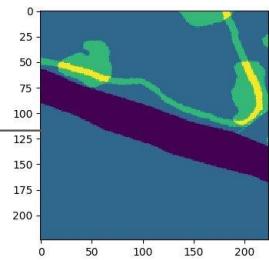
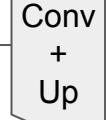
A band
14 x 14



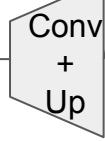
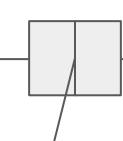
Multi input

masks, 224x224

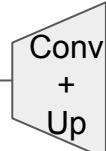
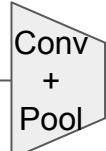
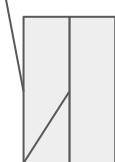
Concat



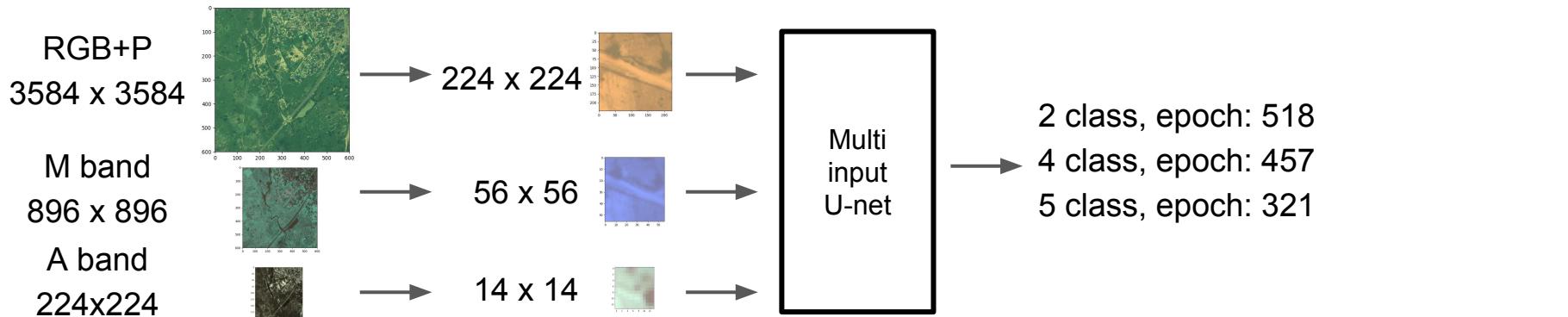
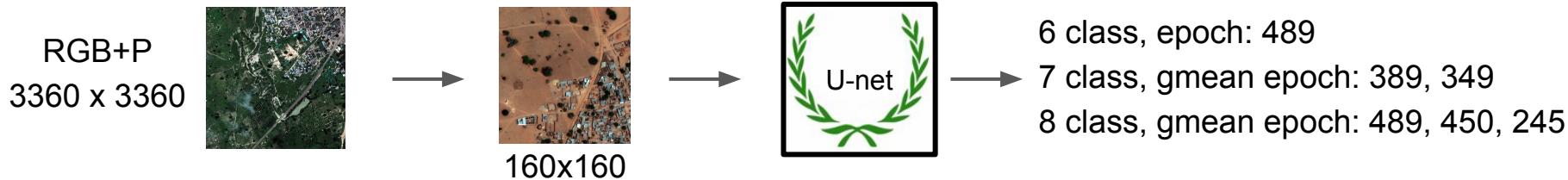
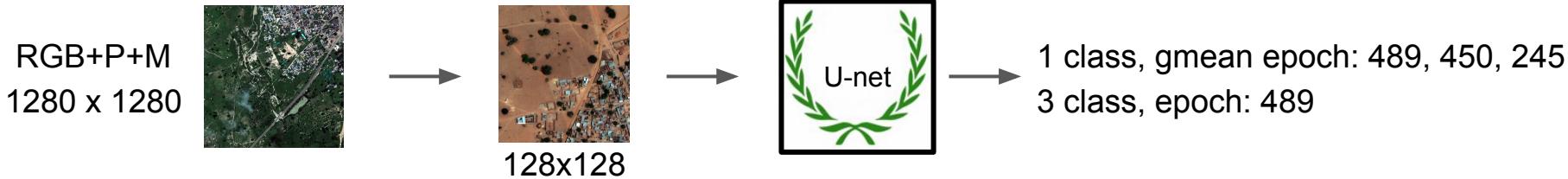
Concat



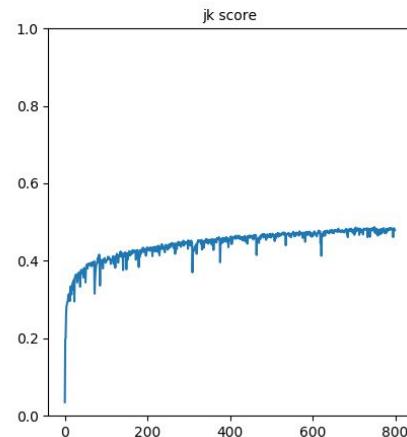
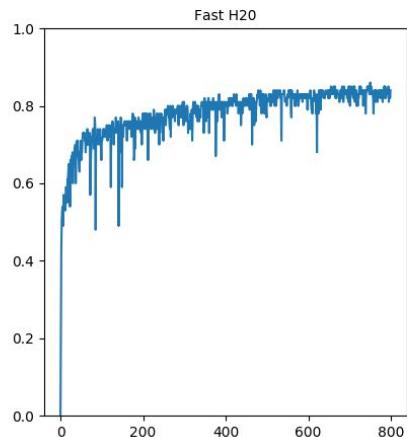
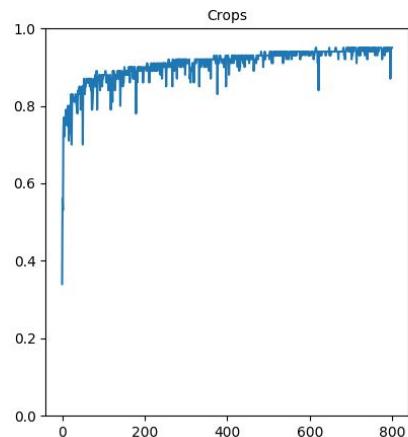
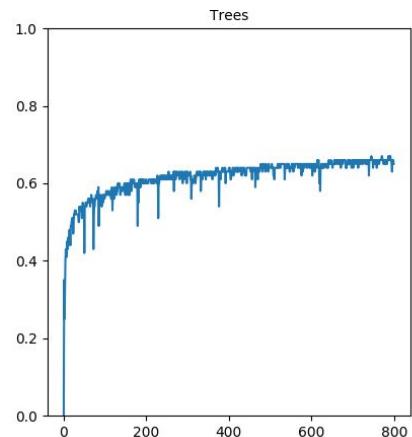
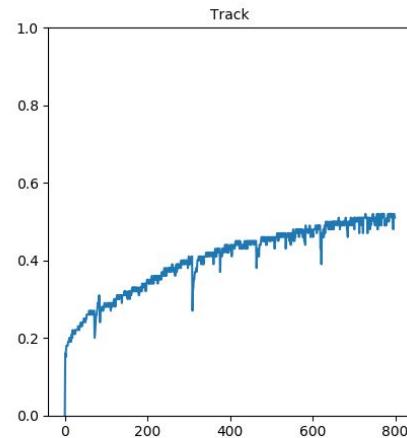
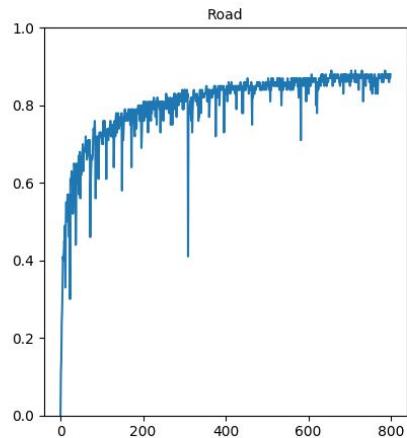
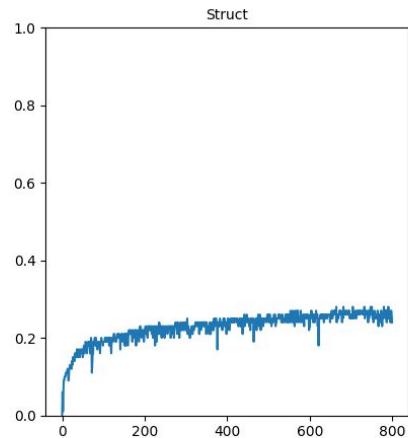
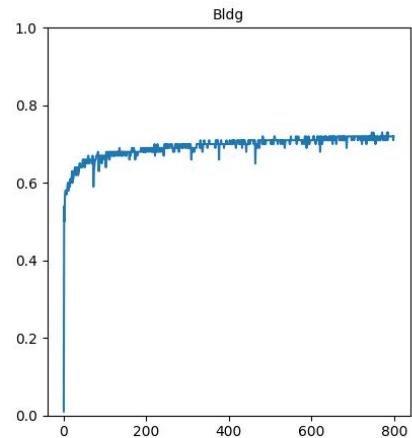
Concat



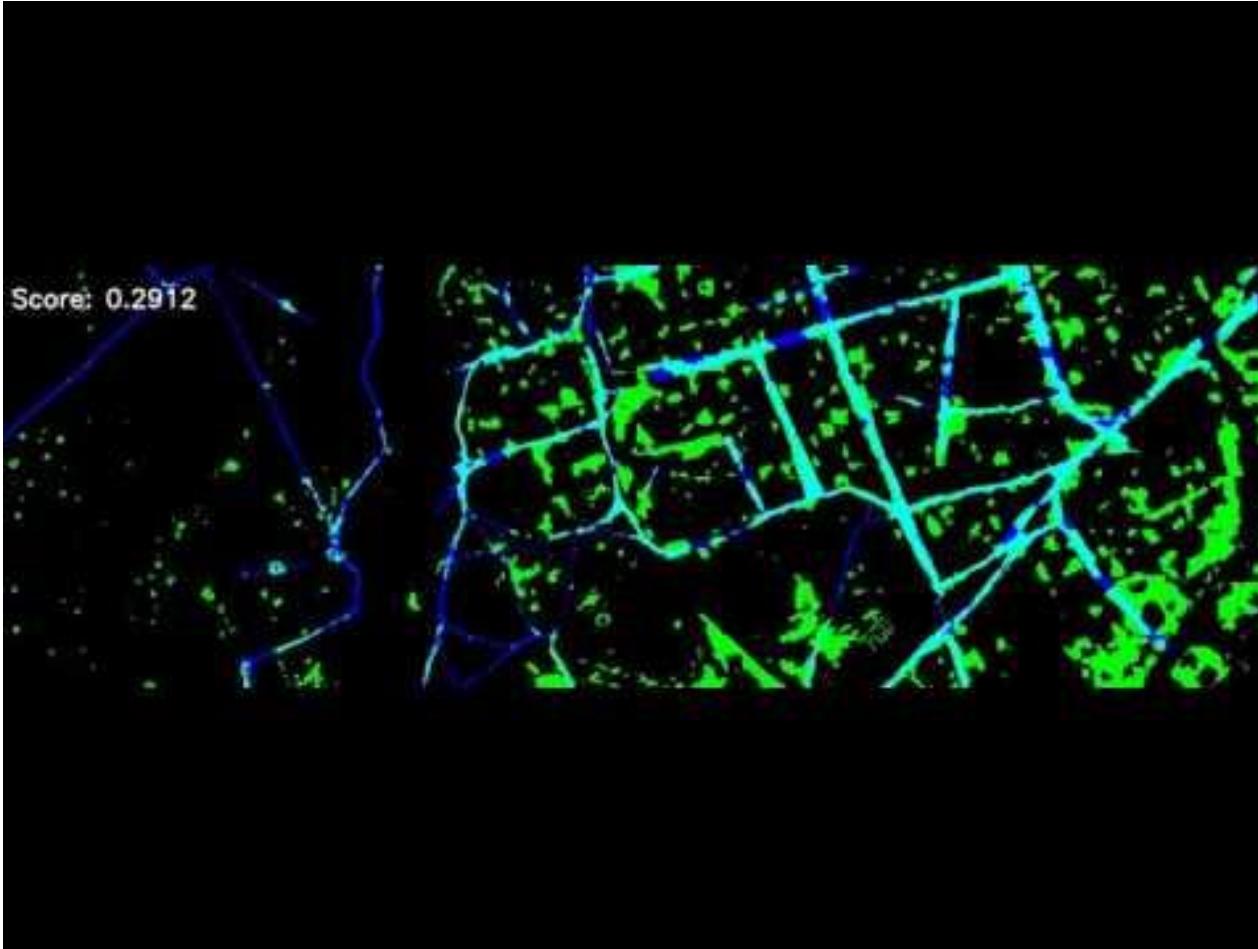
Full pipeline



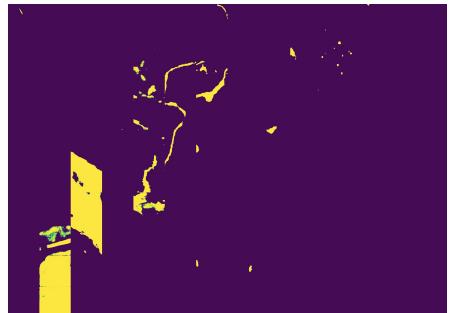
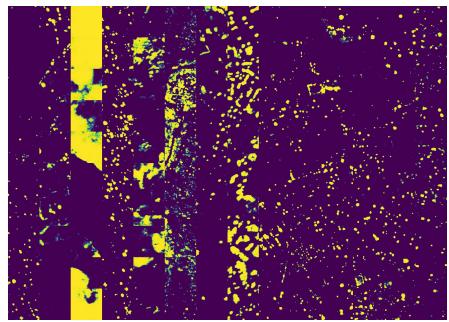
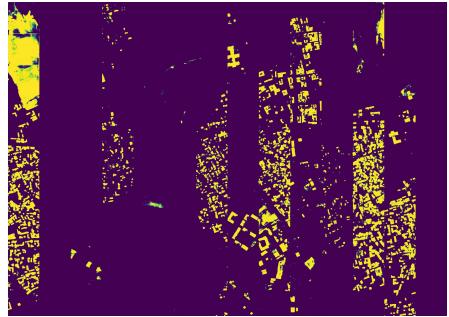
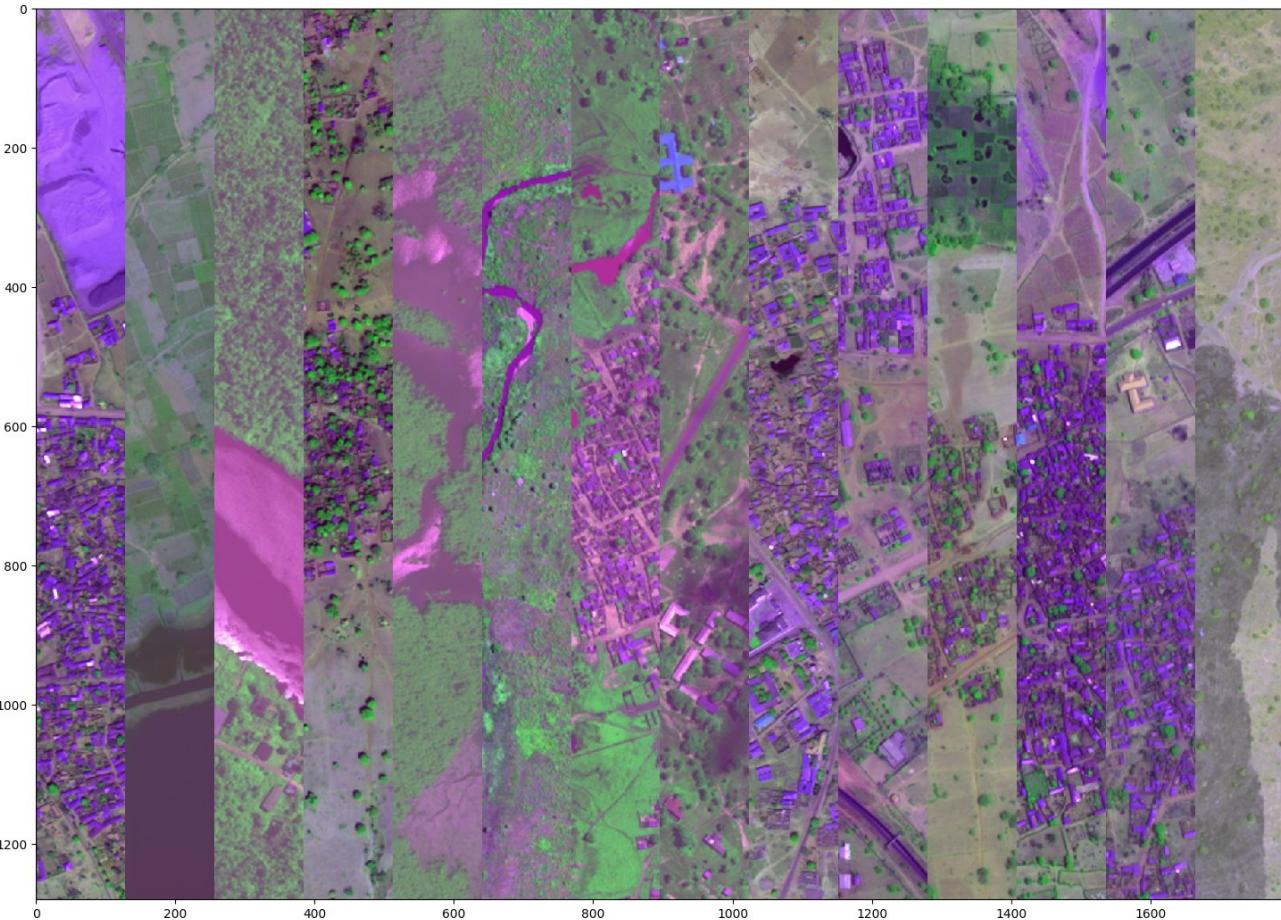
Logs



Logs

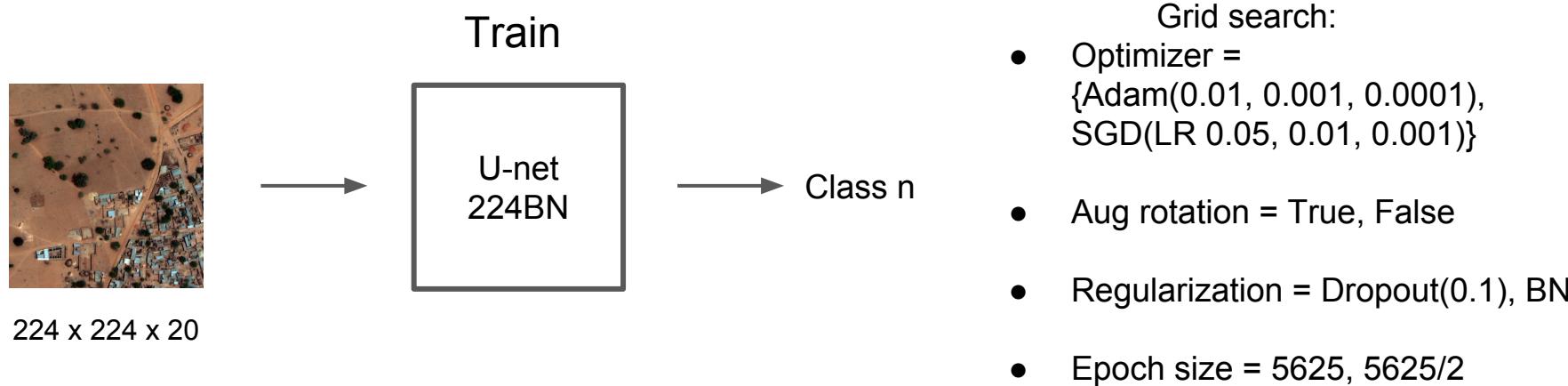
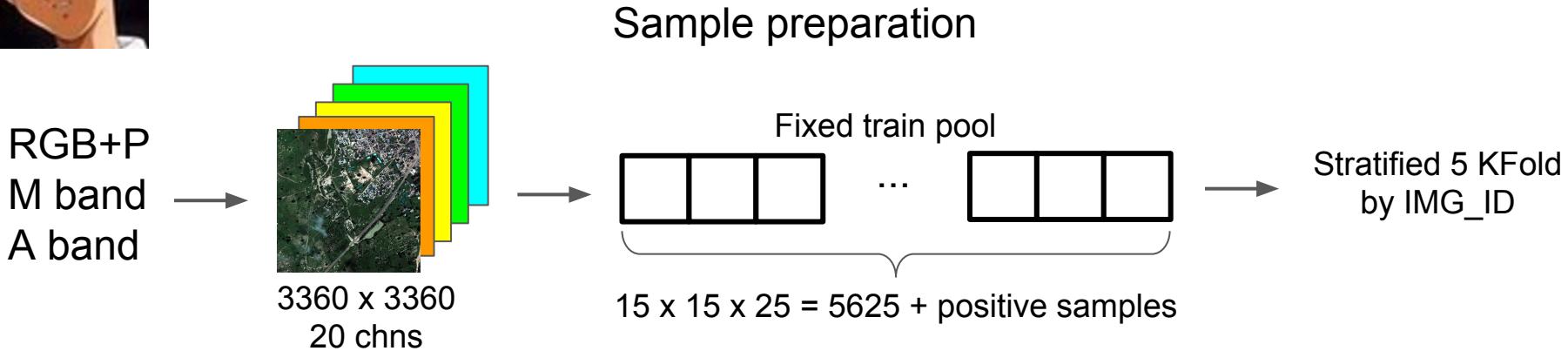


Test check

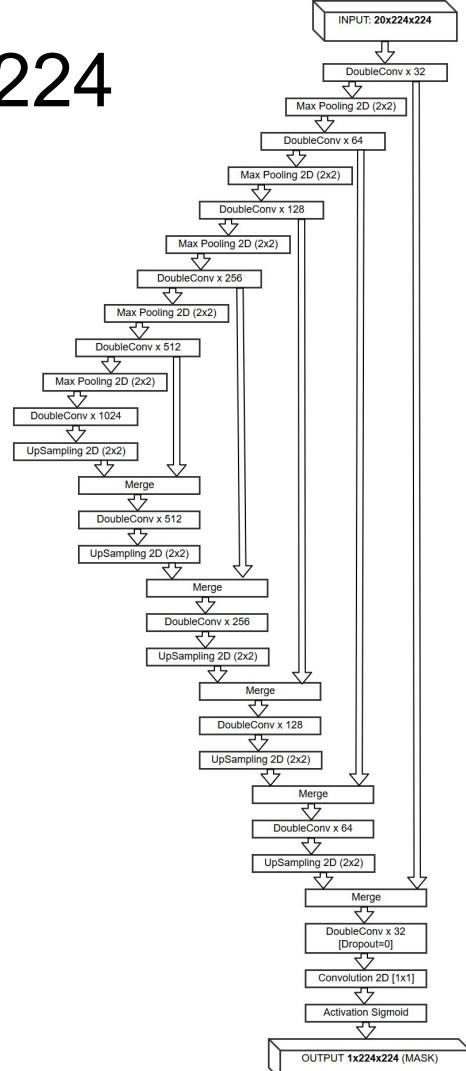
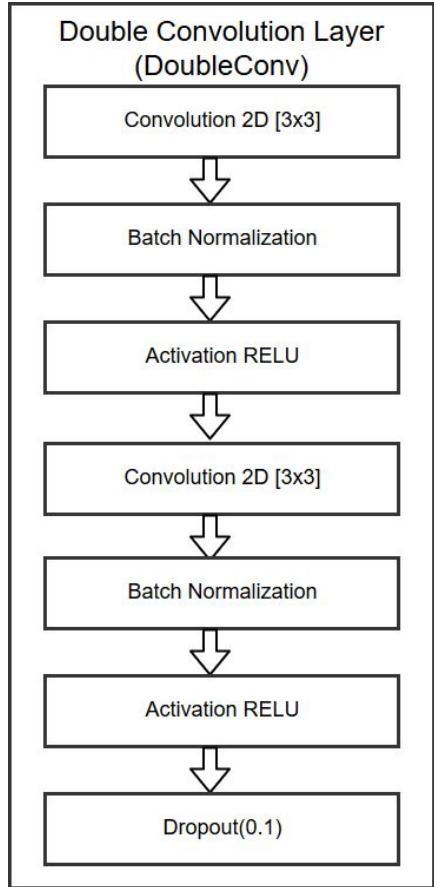
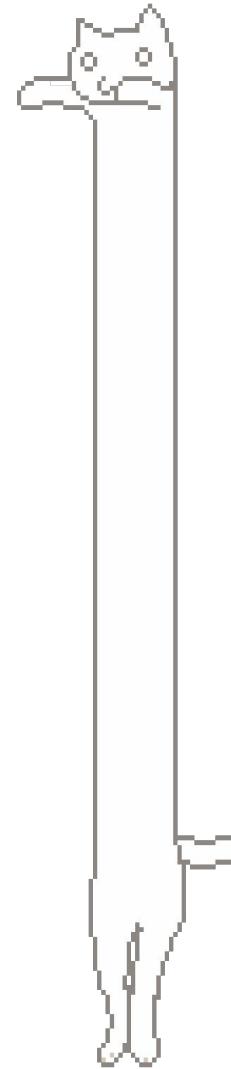




ZFTurbo solution



Структура U-net BN224

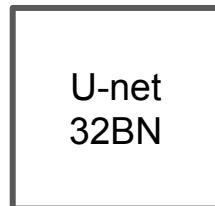


U-net BN32 для машин

RGB+P+M



32 x 32 x 12

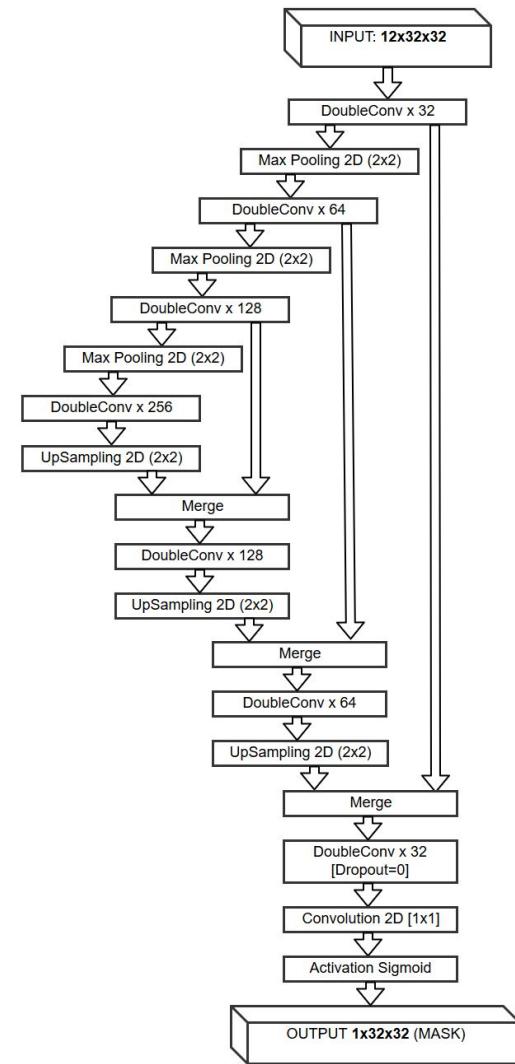


Class n

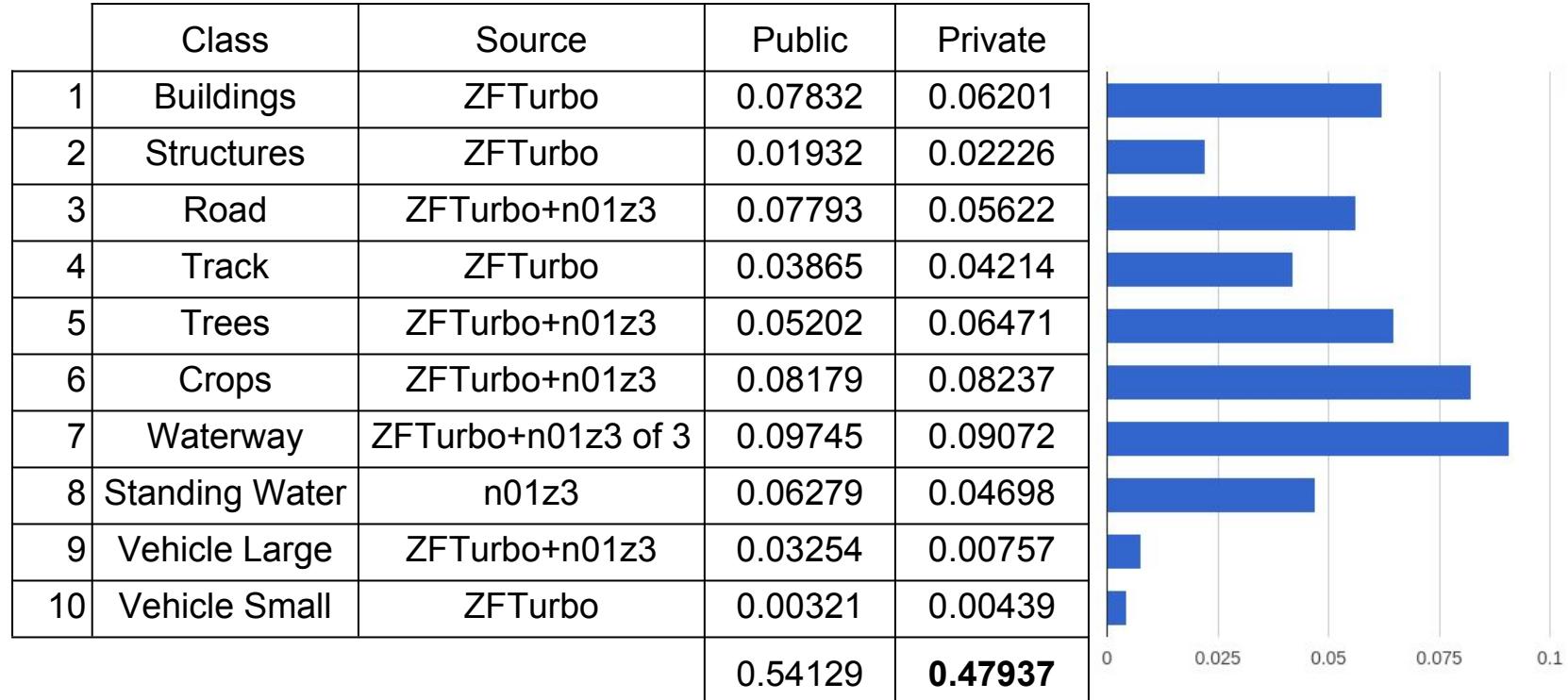
Augs: Rotation, Mirror, Flip

Коэффициент Тверски loss

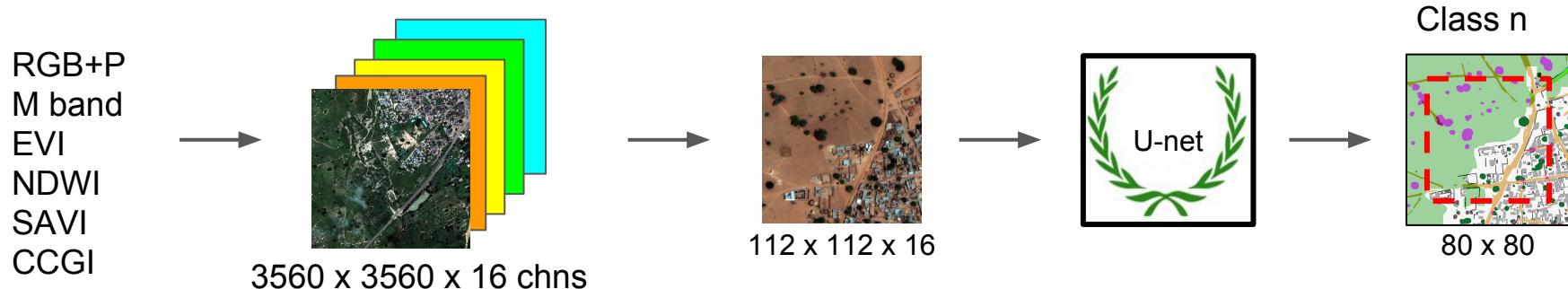
$$S(X, Y) = \frac{|X \cap Y|}{|X \cap Y| + \alpha|X - Y| + \beta|Y - X|}$$



Final merge (2nd place)



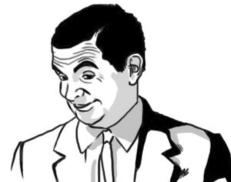
Vladimir Iglovikov, Ph.D. & Sergey Mushinskiy (3rd place)



- Conv + ReLu → Conv + BN + Elu
- Batch 128
- Train aug: Mirror, Flip, 45 Flip (transpose) [D4 group]
- Nadam (LR 0.001, 0.0001)
- Test time aug: original + flips + transpose



P.S. Sergey is open for *new opportunities* in Data Science
linkedin.com/in/sergeymushinskiy



alno & kostia & deepsystems

Many different approaches:

- UNet with smaller patches
- Modified DenseNet (Tiramisu) github.com/SimJeg/FC-DenseNet
- SegNet (conv/deconv without skip connections)
- Using pre-trained VGG (smart pre-processing by deepsystems)
- Some post-processing for water
- Voting/union of several models

2nd place on Public LB → 5th place on Private LB



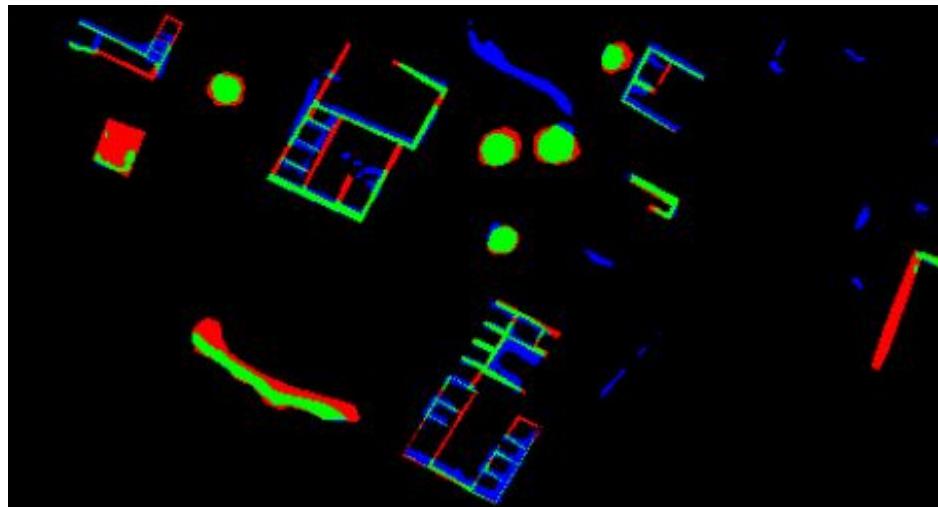
Standing water 0.072 → 0.018, other classes were good.

Small vehicle augmentation (didn't finish in time)



Misc. Manmade structures

- Very thin, lot of loss on polygon → mask → polygon:
 - pixel jaccard 0.27 → polygon jaccard 0.23 :(
- Solution: before making **train** polygons, do `.buffer(0.5)`:
 - pixel jaccard 0.29 → polygon jaccard 0.29 on validation, 0.027 on Private LB



54



[LB 0.42]Ultimate full solution (run on your HW)

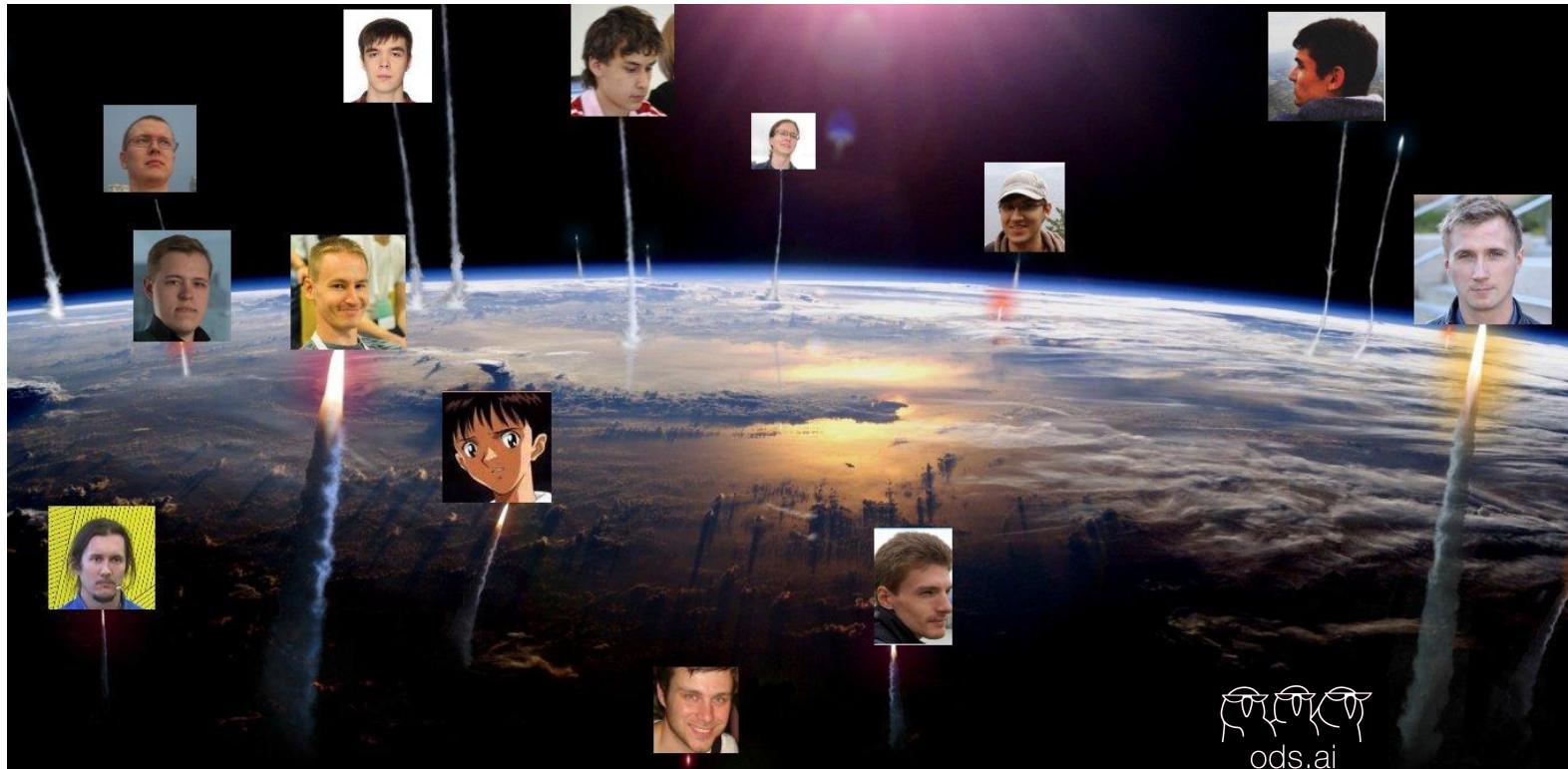
run 19 days ago by Sergey Mushinskiy



25	11	Alexander Movchan	0.42424	19	 0.41907
26	18	Kamenshchikov Mikhail	0.42156	27	 0.41907
27	136	neongen	0.41907	37	 0.41907
28	new	Dmitry Kozlov	0.41907	3	 0.41907
29	16	Rakpong Kittinaradorn	0.41907	45	 0.41907
The Machine Learning Society					 0.41907
<ul style="list-style-type: none"> • Tristen Tyler Blake • BrandonBosse • Cliff Green • asif • Rohit Patnaik • Joexjmmvhm • OP • Jesus Zaragoza • Naveen • Jason Liu • Adarsh Golkeri • MattCamack • JasonClose • jrodf555 • MigueAragon • Lalithanath Behera • Andres Zacarias • janewang • Simatal • AbhijeetSG • ErvTong • gavrand • Lehmliller • Matthew Bae • Peter Lee • Sia A • atet • AjaniGyasi • Frederick Cabasa • cplatt • Marc-AntoineArgenton • Tony Reina • Ashley Jiang • KmShannon 					
30	10		0.41907	17	 0.41846
31	191	clustifier	0.41897	20	 0.41846
32	18	Denis Sokolov	0.41846	22	0.41846
33	new	Ondrej Svitek	0.41817	1	0.41846

3:00 AM, Moscow time

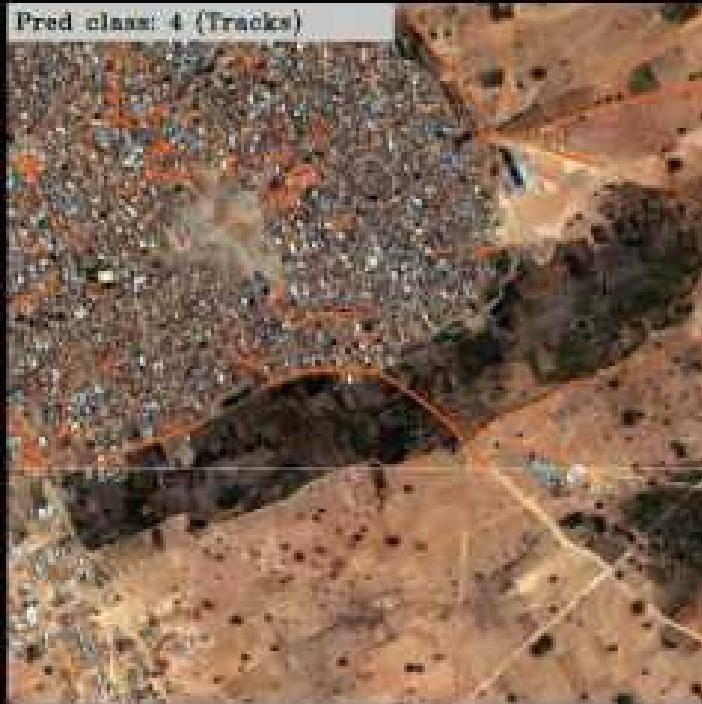
⚠ This competition has completed. The results will become available after they're verified by the competition organizers.



Real class: 4 (Tracks)



Pred class: 4 (Tracks)



Hardware & frameworks



Ubuntu 16.04.2 LTS
Python 2.7
Keras 1.1.2
Theano-0.9.0rc2.dev

Dev	Learn 24/7
Core i7	Core i5
RAM 48 Gb	RAM 32 Gb
GTX 1080	Titan Black

Learn 24/7
Core i5
RAM 64 Gb
2x Titan X 12 Gb (Maxwell)



Windows 10
Python 3.4
Keras 1.0.8
Theano-0.8.2

Dev/Learn
Core i7
RAM 32 Gb
980 Ti 6 GB

Full scheme

