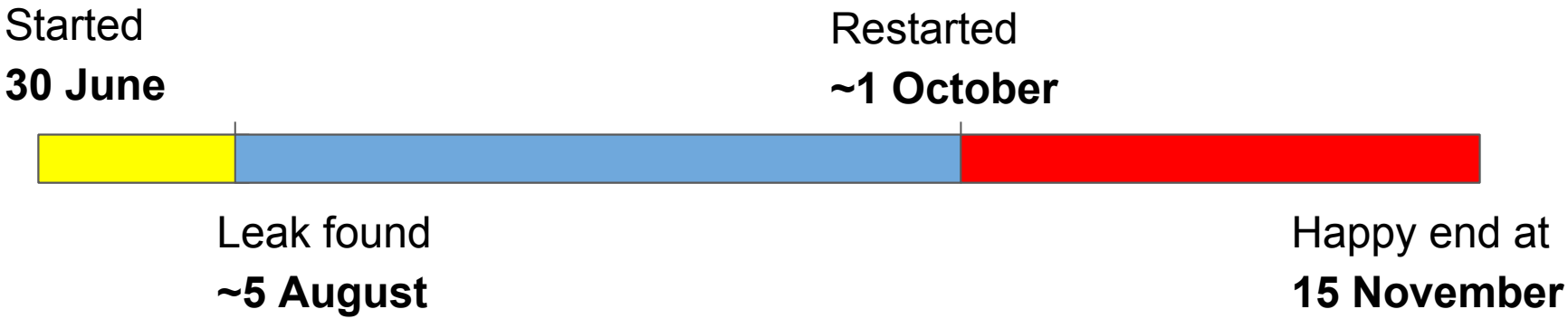
kaggle: Airbus Ship Detection Challenge

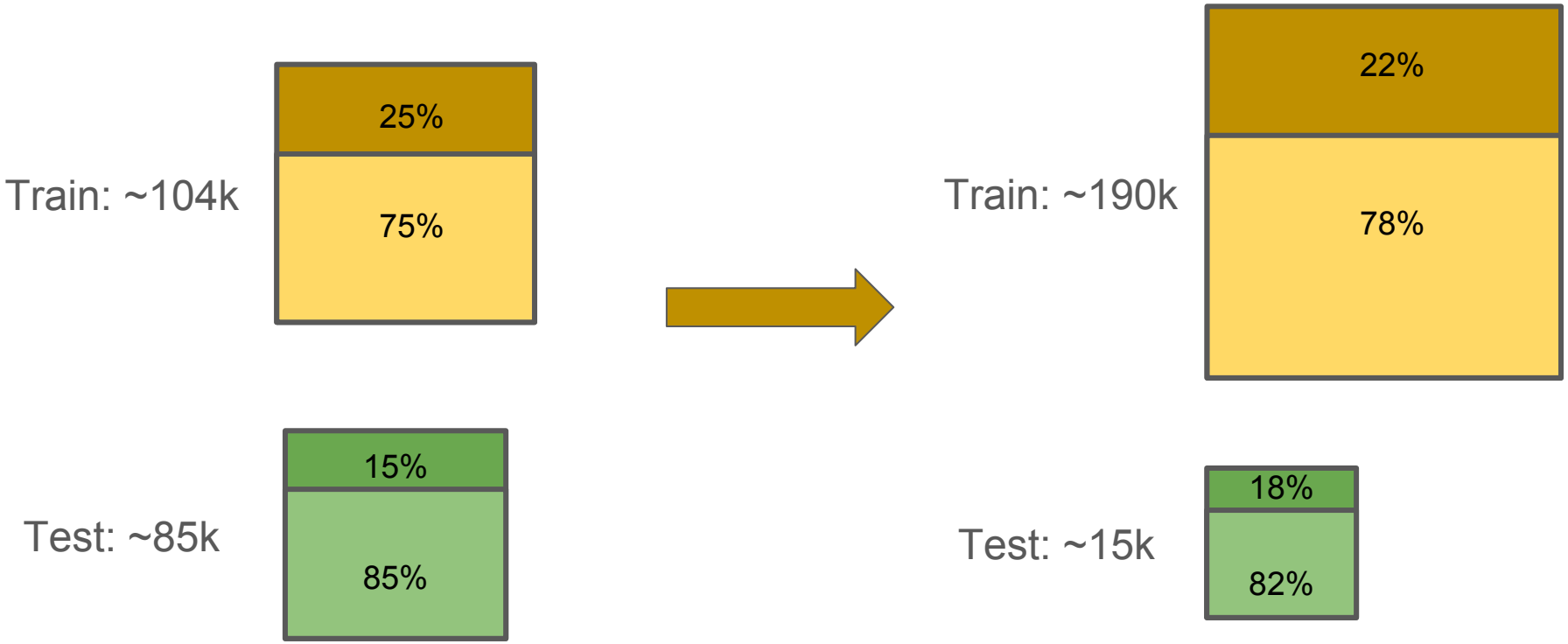
Konstantin Gavrilchik
Evgeny Kononenko

Problem

Competition timeline



Data



Evaluation

F2 metric over thresholds: (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95)

$$F_\beta(t) = \frac{(1+\beta^2)\cdot TP(t)}{(1+\beta^2)\cdot TP(t) + \beta^2 \cdot FN(t) + FP(t)}.$$

$$\frac{1}{|thresholds|} \sum_t F_2(t).$$

Hardware

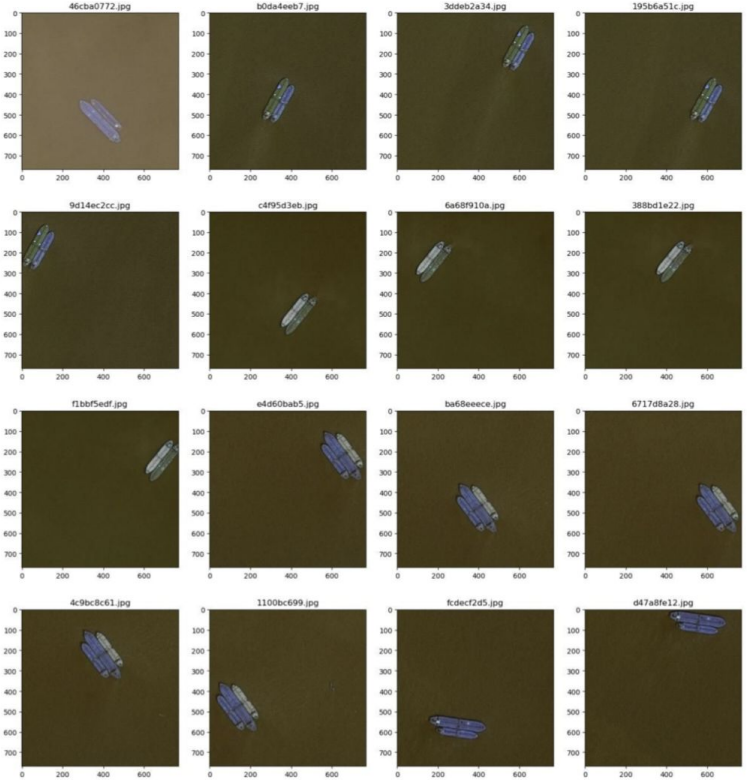
Before last week:

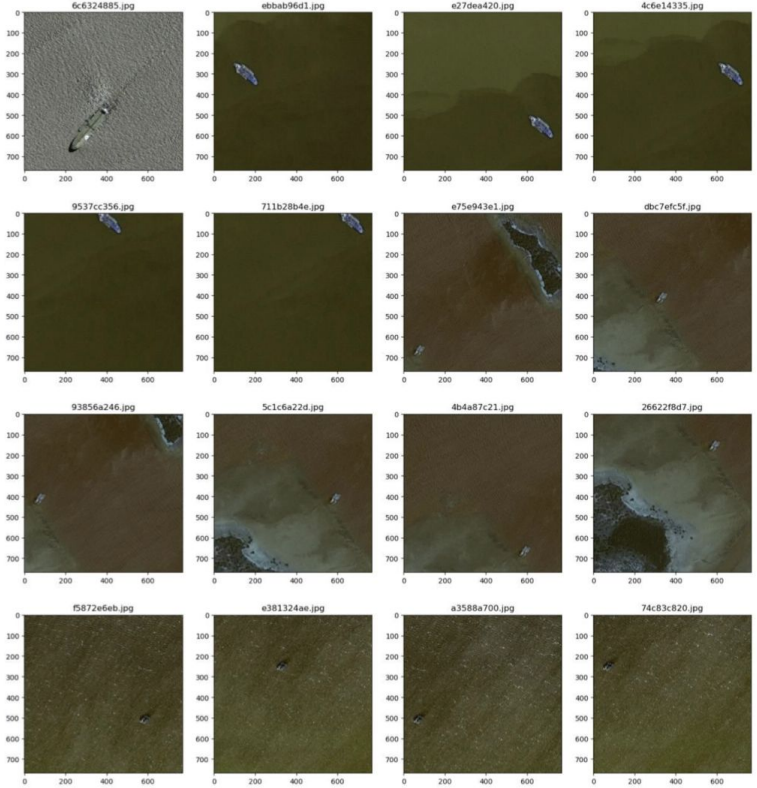
1) Threadripper 1950x (16 cores, 32 threads) + 2x 1080Ti
2) Intel Core i5 (4 cores) + 1x 1080Ti

Last week (big thanks to Artur Kuzin aka n01z3):

+ Intel Core i7 (8 cores) + 2x 1080Ti

Leak





https://youtu.be/MIbetMAnC04

Puzzle & validation strategy

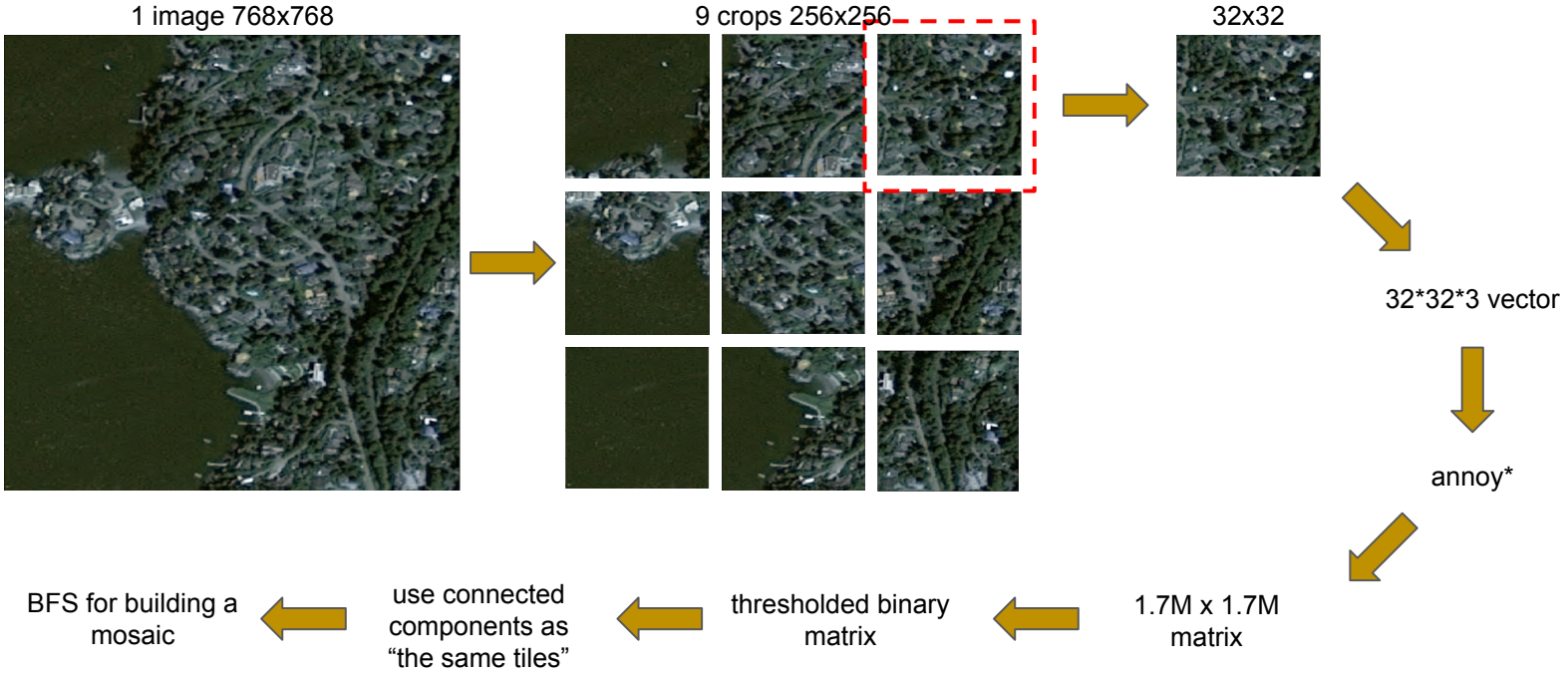
Puzzles:

- 30k mosaics in total (60k in public kernels)
- some of them at the same place (different moments of time)

Validation strategy:

- sort by length
- create hold-out (1/15 part of all dataset)
- use final metric for evaluation

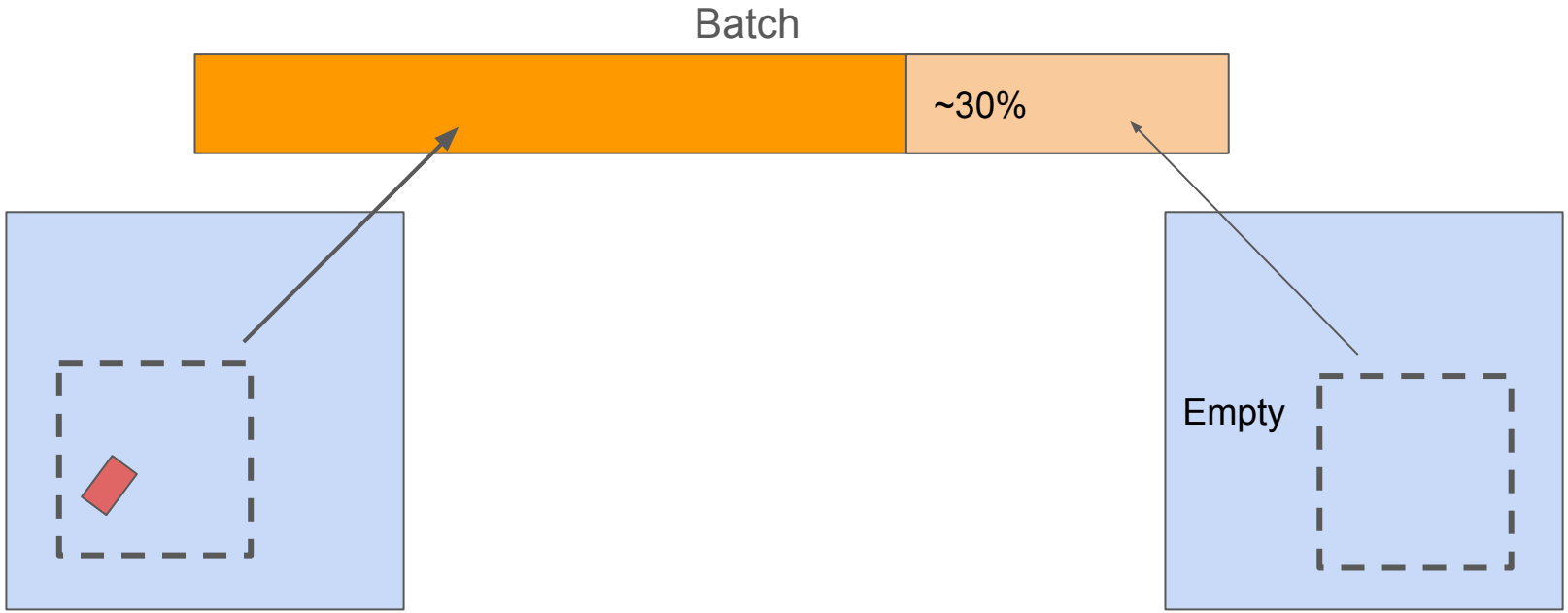
How to solve the puzzle?



* https://github.com/spotify/annoy

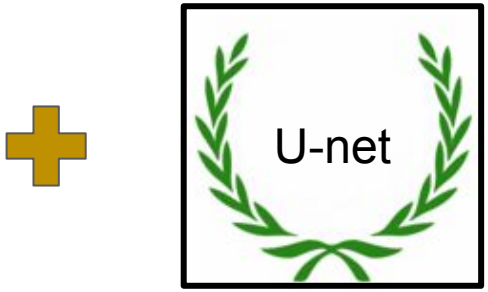
How to prepare data for the models?

- Random 256x256 crop **with ship**
- Sample 30% of empty pictures (randomly 256x256 crop from random pictures)



UNet is all you need...

ResNext50
ResNext101
Densenet121
DPN92



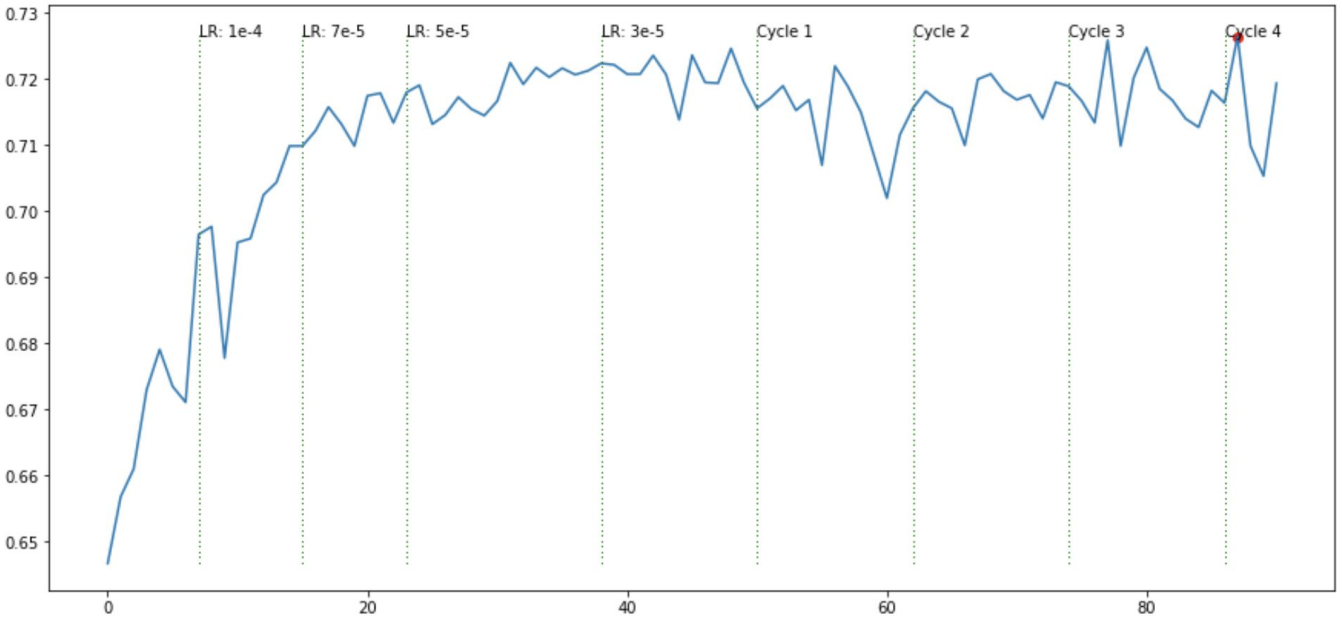
Training process

Loss = BCELoss + DiceLoss

LRScheduler:
- optimizer: Adam
- LRs [1e-4, 7e-5, 5e-5]
- milestones = [7, 15, 23]

ReduceLROnPlateu:
- optimizer: Adam
- lr = 3e-5
- factor = 0.7
- patience = 3
- epochs = 15

CosineAnnealing / CycleLR:
- optimizer: RMSprop
- lr = 3e-5
- min_lr = 8e-6
- T = 6
- epochs = 50

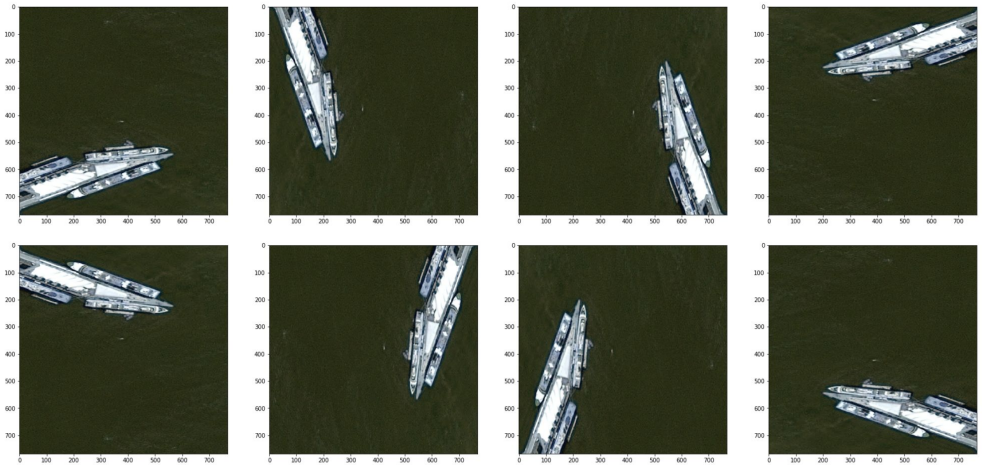


Augmentations

Konstantin	Evgeny
- Flip - RandomRotate90 - Transpose	- RandomRotate90 - Flip - Transpose - ShiftScaleRotate - RandomBrightness(limit=0.05) - RandomContrast(limit=0.05) - GaussNoize(limit=(0, 10)) - Blur(limit=4)

TTA

- HorizontalFlip + VerticalFlip
- Rotate90



Mean of 8 predictions

Model: first attempt

Model	CV	Public leaderboard	Private leaderboard
Unet - DenseNet121	0.726	0.723 (0.740 with TTA)	0.846
Unet - ResNeXt50	0.734	0.707 (0.744 with TTA)	0.841
Blend	???	??? (0.748 with TTA)	0.848

Model: pseudo labeling

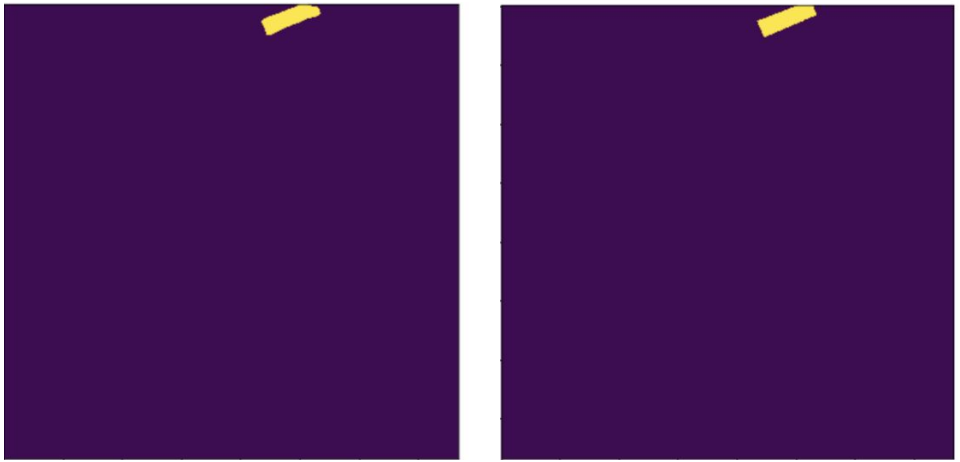
Algorithm:

- Select only the most confident predictions (top 1500)
- Add them to the training set and sample them in each epoch
- Don’t use pretrained weights from previous models

Hint:

All masks are rectangular and we optimize rectangles if:

- it don’t change the IoU more than 5%
- only masks with area >300

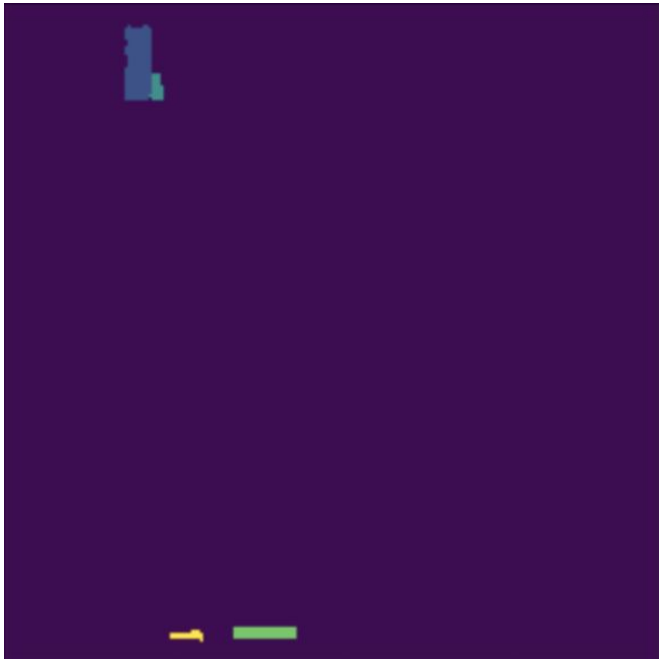


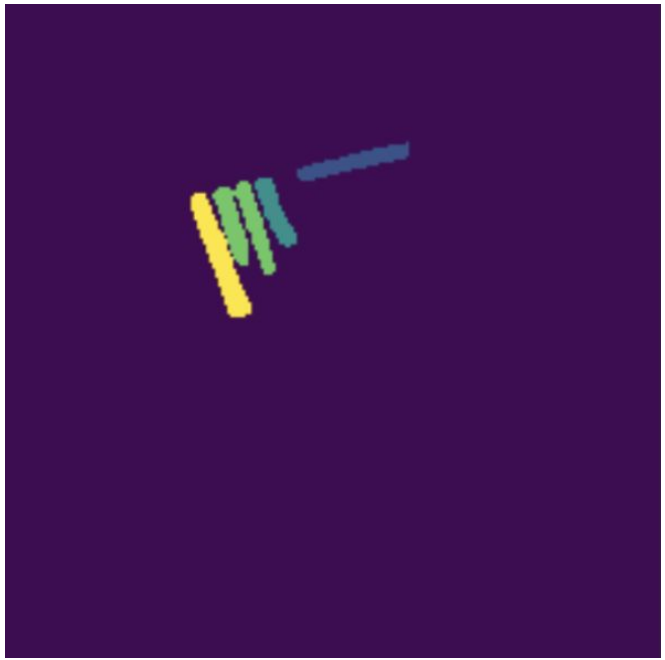
Model: pseudo labeling

Model	CV	Public leaderboard	Private leaderboard
Unet - DenseNet121	0.728	0.749 with TTA	0.851
Unet - ResNeXt50	0.738	0.748 with TTA	0.848
Blend	???	0.751 with TTA	0.852

Splitting by instances

Watershed + KNN





Score: 0.758 Public / 0.8549 Private

Final model

Blend of 5 models:

- DenseNet121 (no pseudo)
- DenseNet121 (pseudo)
- ResNeXt50 (no pseudo)
- ResNeXt50 (pseudo)
- ResNeXt101 (no pseudo)

ResNet50 classifier

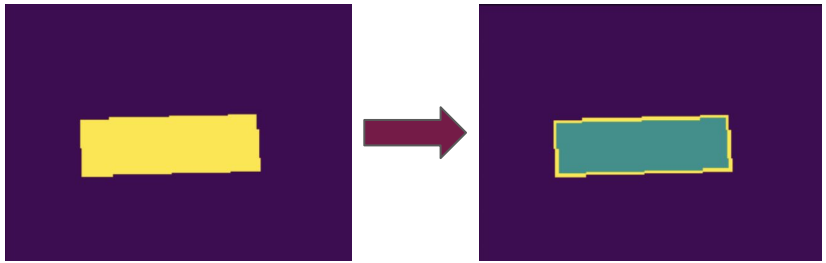
Score: 0.864 Public / 0.8544 Private

Shake up

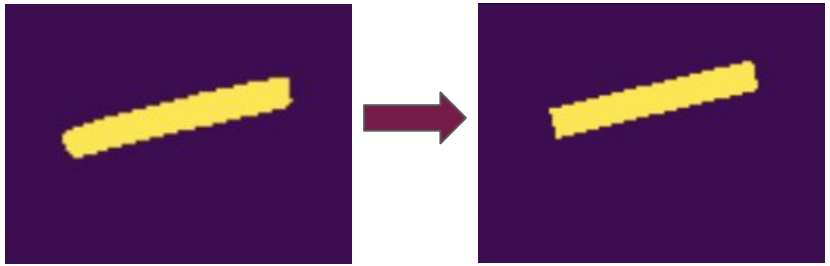
1	▲ 1	[ods.ai] Rectangle is all you n...	0.76444	75	9d
2	▲ 76	[ods.ai] n01z3	0.75740	50	9d
3	—	lafoss & trian2018 & phun	0.75698	156	9d
4	▲ 356	valilenk	0.75514	17	9d
5	▲ 1	ChinaAndGermany	0.75421	191	9d
6	▼ 2	Kensho Mosaic	0.75377	108	9d
7	new	[ods.ai] ValeriyBabushkin	0.75375	16	9d
8	▲ 134	[ods.ai] Ilya Kibardin	0.75372	21	9d
9	▼ 8	x0x0w1	0.75260	105	9d
10	▼ 5	bestfitting	0.75248	146	9d
11	▲ 1	See&Edu	0.75236	76	9d
12	▲ 40	TreeNewBee	0.75097	94	9d
13	▼ 3	Soonhwan Kwon	0.75090	90	9d
14	▼ 6	Kohei & tito	0.75054	105	9d
15	▼ 8	Igor Praznik	0.74946	71	9d

1	—	[ods.ai] Rectangle is all you n...	0.85448	75	9d
2	▲ 15	[ods.ai] topcoders	0.85433	54	9d
3	▲ 7	bestfitting	0.85428	146	9d
4	▲ 22	[attention heads]	0.85411	55	9d
5	▲ 13	dhammack	0.85350	14	10d
6	▲ 35	[ods.ai] BZS	0.85252	81	9d
7	▲ 12	[Eversec] Apocalypse Lab 天...	0.85225	49	9d
8	▼ 3	ChinaAndGermany	0.85199	191	9d
9	▼ 3	Kensho Mosaic	0.85184	108	9d
10	▲ 30	tkuanlun350	0.85178	66	10d
11	▲ 4	Igor Praznik	0.85156	71	9d
12	▲ 22	[ods.ai] Scizzzo	0.85066	76	9d
13	▲ 24	[ods.ai] Pavel Pleskov	0.85039	10	10d
14	▲ 10	robga & kerem & radek	0.85009	171	9d
15	▲ 29	YaG320	0.84998	136	9d

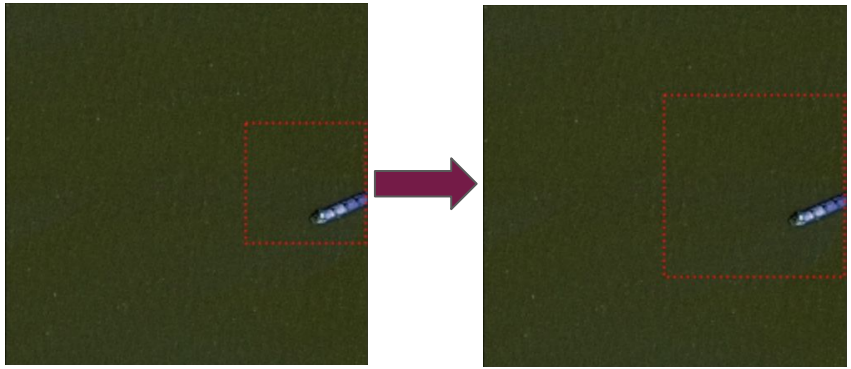
What we had tried, but did not work for us



Boundaries as 2nd channel



More rectangular predictions

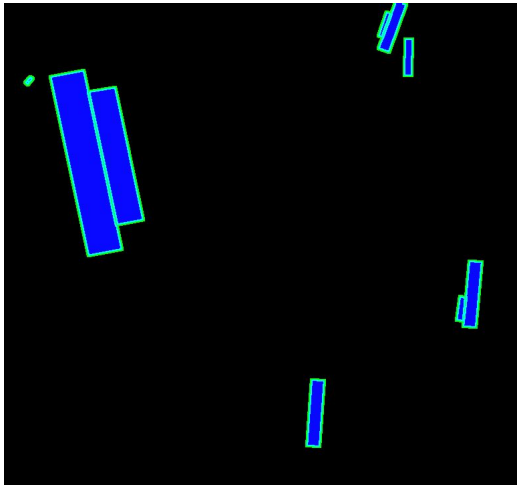


Bigger resolution

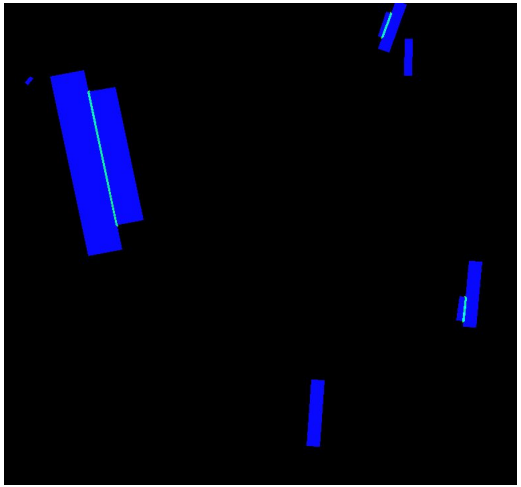
Other solutions

- **Extra channels in mask**

Second channel



Third channel



- **Mask R-CNN**

Thank you for your attention!