Find the nuclei in divergent images to advance medical discovery

First place solution overview by

Alexander Buslaev, Selim Seferbekov, Victor Durnov

public lb:

#	△1w	Team Name	Kernel	Team Members	Score	Entries	Last
1	–	Allen Goodman (not prize elig...			0.634	14	2mo
2	–	[ods.ai] topcoders			0.598	156	2h
Your Best Entry ↑							
Your submission scored 0.594, which is not an improvement of your best score. Keep trying!							
3	▲1	Nuclear Vision			0.575	123	5h
4	▼1	Daydreamers			0.570	177	3h
5	–	JianPeiAI			0.568	300	11h
6	–	ploider			0.564	292	3h
7	▲9	BIOMAGic		+10	0.561	107	2h
8	▲1	Nuclear Fission			0.559	222	10h
9	▲10	OsciiArt			0.557	250	35m
10	▼3	MLT			0.554	164	19h
11	▲12	[ods.ai] SAAZ			0.553	318	3h
12	▲22	MSL_SKKU			0.553	158	6h
13	▲62	[ods.ai] amirassov			0.553	123	1h
14	▼1	TonyShard			0.553	151	12h
15	▼7	[ods.ai] Vladimir Larin			0.550	224	37m
16	▼6	Tariq S.			0.548	20	2mo
17	▼6	[qgraph.io]QGraph			0.547	282	9h
18	▼6	jeandebleau			0.545	122	3h
19	▲2	zeus			0.542	202	9h
20	–	elituv			0.542	175	15h
21	▲4	Pokfulam			0.540	142	17h

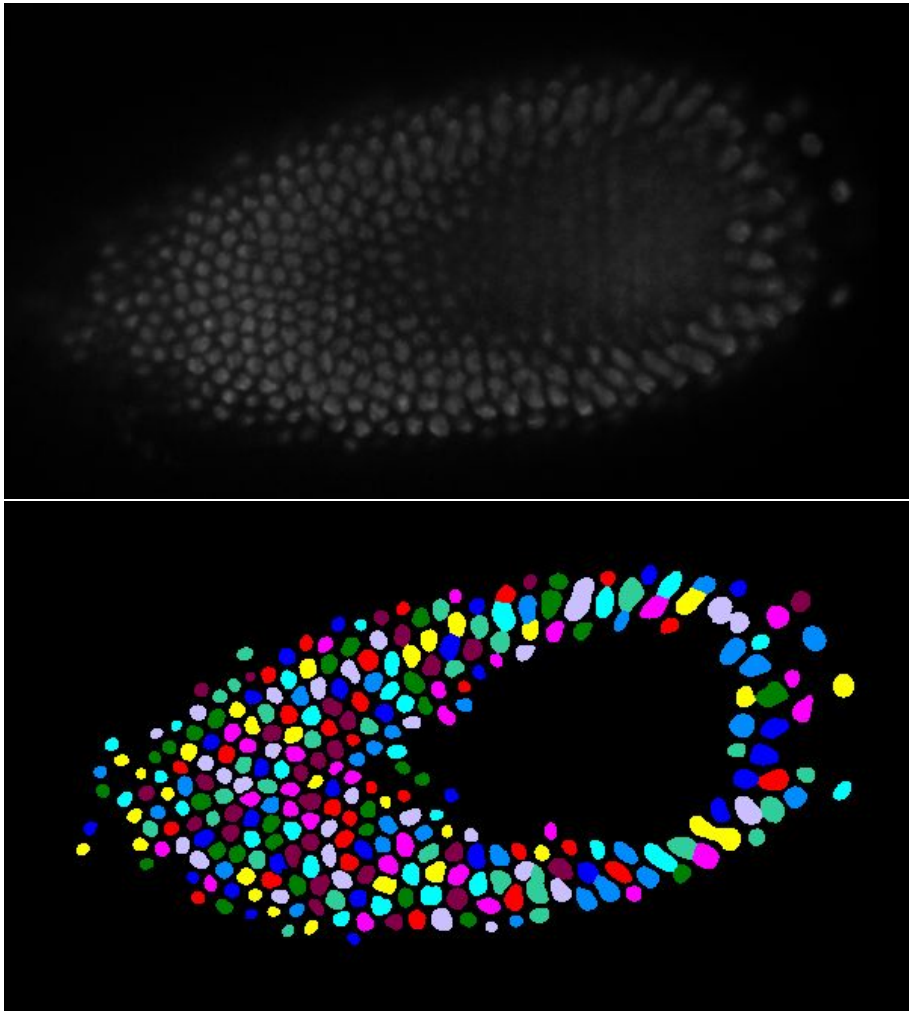
private lb:

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲187	[ods.ai] topcoders			0.631	2	8d
2	▲331	jacobkie			0.614	2	5d
3	▲233	Deep Retina			0.614	2	7d
4	▲193	Nuclear Vision			0.610	2	6d
5	▲75	Inom Mirzaev			0.609	4	6d
6	▲172	Gangadhar Payyavula			0.591	5	6d
7	▲137	[ods.ai] Gold Diggers		+3	0.584	6	5d
8	▲140	MPWARE Team			0.578	9	4d
9	▲110	emergent complexity			0.576	3	5d
10	▲219	Daydreamers			0.574	2	6d
11	▲259	oversam			0.574	2	5d
12	▲264	Adel Valiullin			0.574	2	5d
13	▲277	Comp540_Rice_yl148_ds60			0.574	4	5d
14	▲109	Two Masketeers			0.571	4	5d
15	▼12	BIOMAGic		+10	0.570	5	5d
16	▲84	[ods.ai] Ilya Dobrynin			0.569	2	5d
17	▲69	Andrzej			0.569	6	8d
18	▲39	SeniorBusDriver			0.568	5	5d
19	▲5	Oscar Ji			0.564	4	5d
20	▲122	[ods.ai] Vladimir Larin			0.562	2	8d
21	▲12	[ods.ai] Konstantin Maksimov			0.560	2	9d

Problem statement

- Medical imaging
- Instance segmentation
- 665 images in train
- 65 images for stage 1
- 3019 images in stage 2 (but unknown number of images actually used for scoring)
- metric:

$$\frac{1}{|thresholds|}\sum_{t}\frac{TP(t)}{TP(t)+FP(t)+FN(t)}.$$



Our team

True story:
Topcoder is yet another platform for programming contests and data science competitions with strict rules and nice prizes. We compete to each other all the time but for this competition we decided to conclude a truce for a big goal - to **win** ds bowl 2018..

spacenet road detection:

1	albu
2	pfr
3	cannab
4	selim_sef
5	fbastani
6	ipraznik
7	tcghanareddy
8	hasan.asyari
9	aveysov

urban mapper 3d:

	albu
	selim_sef
	cannab
4	alina.marcu
5	kylelee
	ZFTurbo
7	v.stupnitsky
8	peARrr
9	tcghanareddy

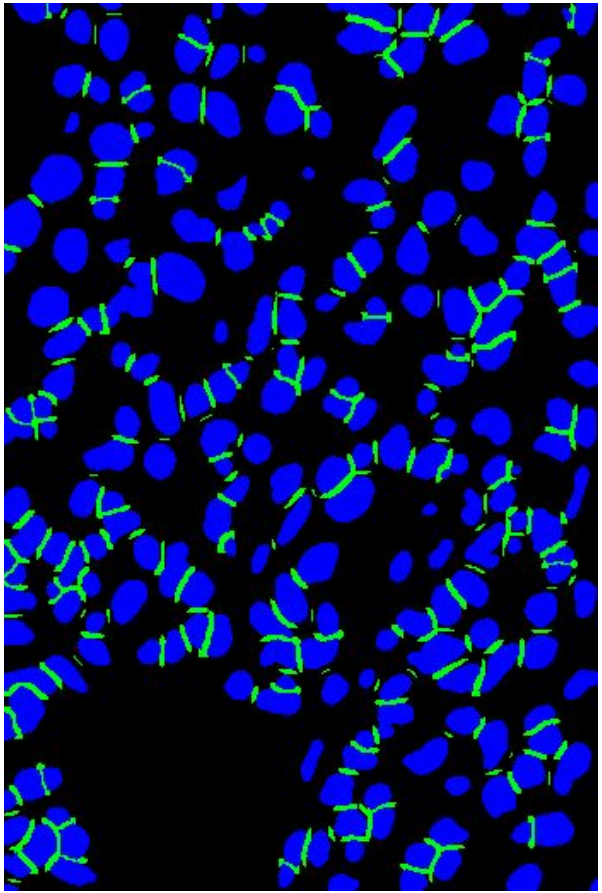
Hint: cannab = @Victor

- 3 individual submissions to decide who will be a leader
- Super-simple baseline: encoder-decoder on masks separated by watershed line gave me ~470 using resnet34-unet trained in 30mins without any postprocessing

Data preparation

Mask evolution:

1. Only nuclei body. Separations are background and nuclei body does not contain watershed single pixel line.
2. Nuclei body + full contour with adaptive contour line width
3. Nuclei body + separations
4. Nuclei body + separations + full nuclei body

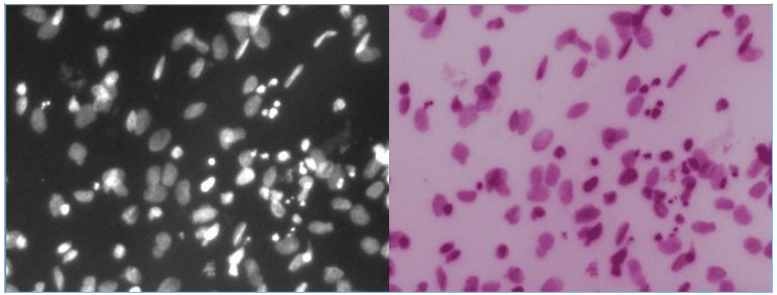


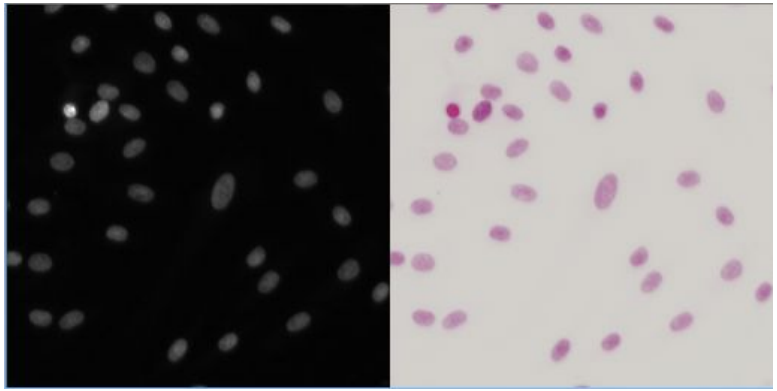
Generalization?

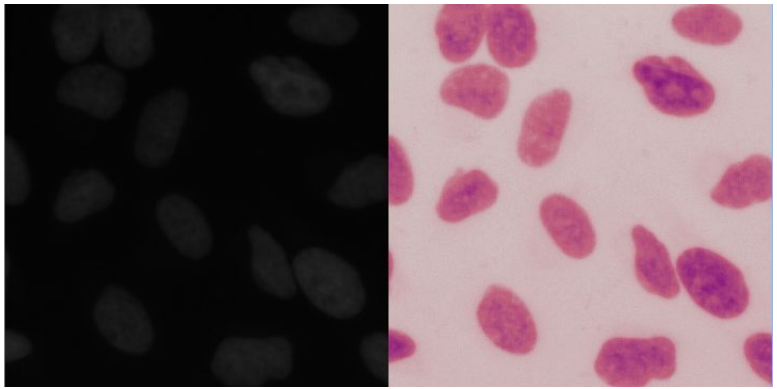
- We used lots of augmentations for better generalization (maybe much more than we actually needed)
- Most controversial - image inversion and channel shuffle. They actually break a lot of relations and may harm learned filters.
- Also Victor used “nuclei copy” augmentation - copying object to another place

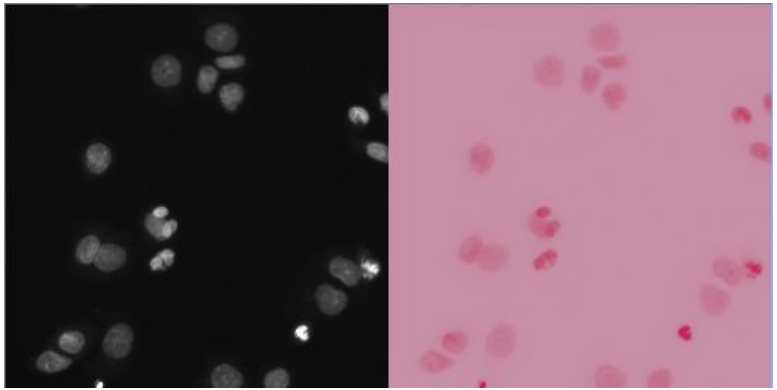
```
def aug_mega_hardcore(prob=0.9):
    return Compose([
        OneOf([
            CLAHE(clipLimit=2, prob=.5),
            IAASharpen(prob=.25),
            IAAEmboss(prob=.25)
        ], prob=.35),
        OneOf([
            IAAAdditiveGaussianNoise(prob=.3),
            GaussNoise(prob=.7),
            ], prob=.5),
        ToGray(prob=.25),
        InvertImg(prob=.2),
        Remap(prob=.4),
        RandomRotate90(),
        Flip(),
        Transpose(),
        OneOf([
            MotionBlur(prob=.2),
            MedianBlur(blur_limit=3, prob=.3),
            Blur(blur_limit=3, prob=.5),
        ], prob=.4),
        OneOf([
            RandomContrast(prob=.5),
            RandomBrightness(prob=.5),
        ], prob=.4),
        ShiftScaleRotate(shift_limit=.0625, scale_limit=0.4, rotate_limit=45, prob=.7),
        OneOf([
            Distort1(prob=.2),
            Distort2(prob=.2),
            ElasticTransform(prob=.2),
            IAAPerspective(prob=.2),
            IAAPiecewiseAffine(prob=.2),
        ], prob=.6),
        HueSaturationValue(prob=.5),
        ChannelShuffle(prob=.2),
    ], prob=prob)
```

Gray -> color

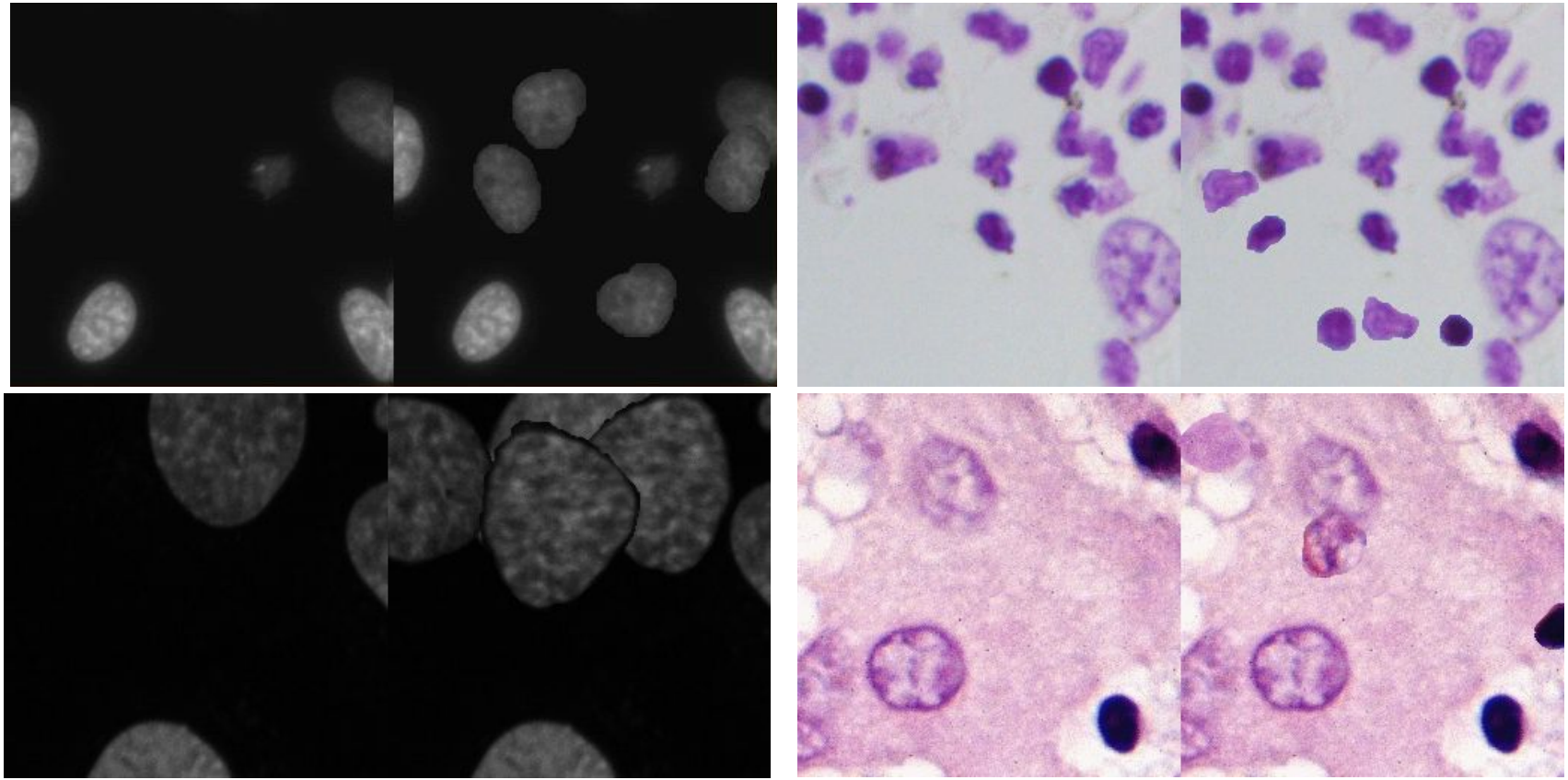




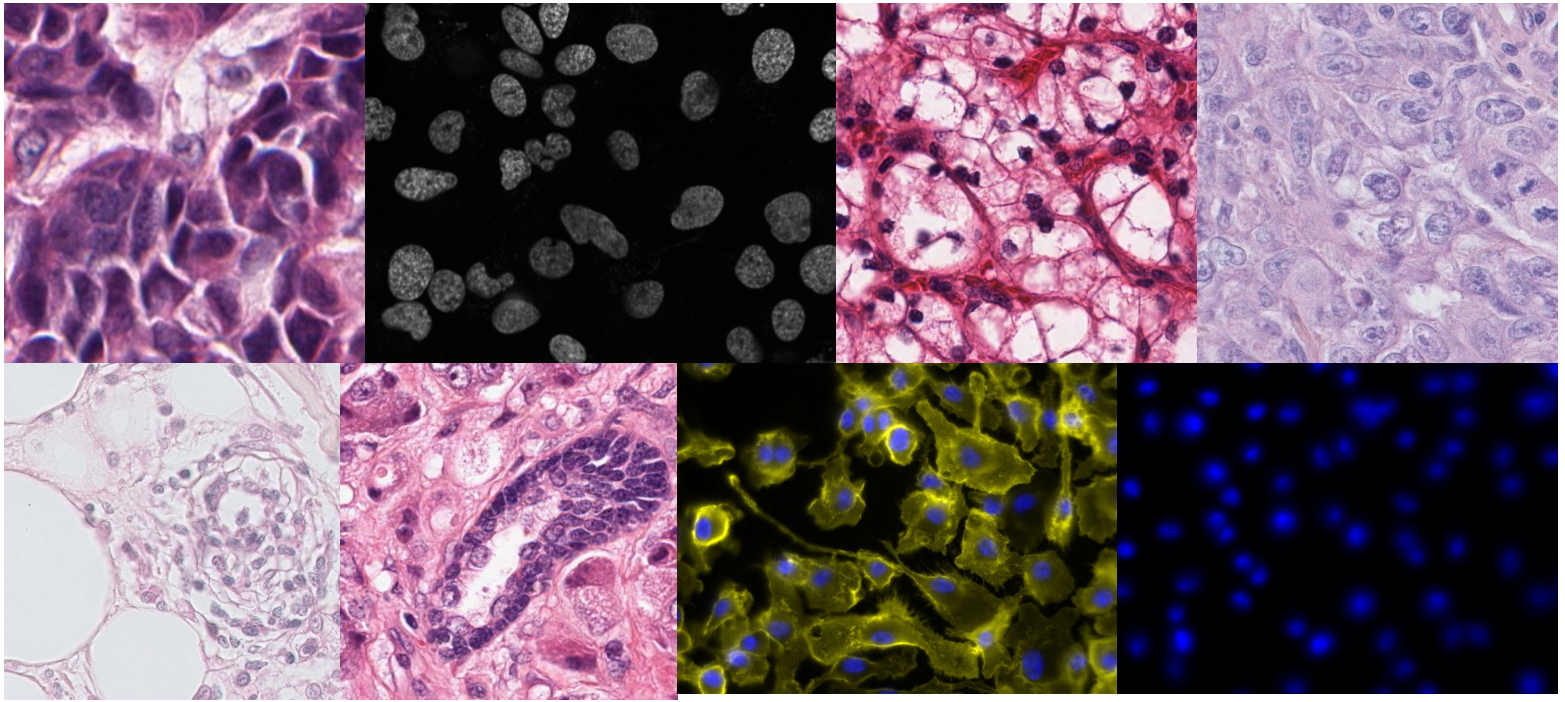




Nuclei copy

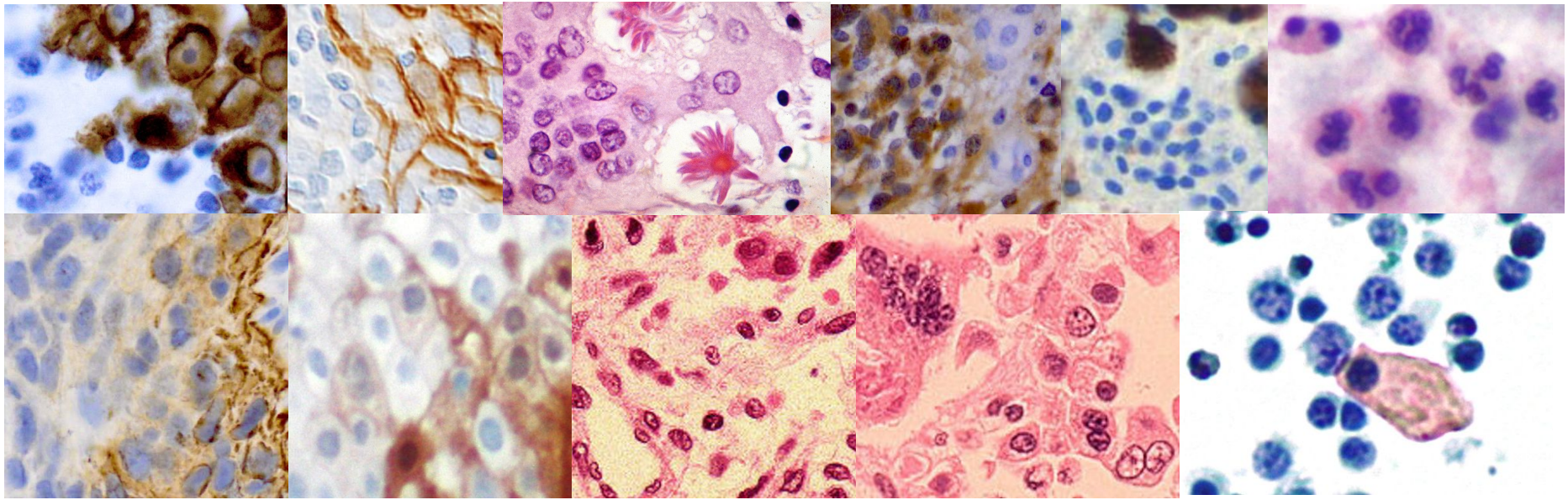


External data



Kumar, TNBC, BBBC020, isbi2009, Janowczyk

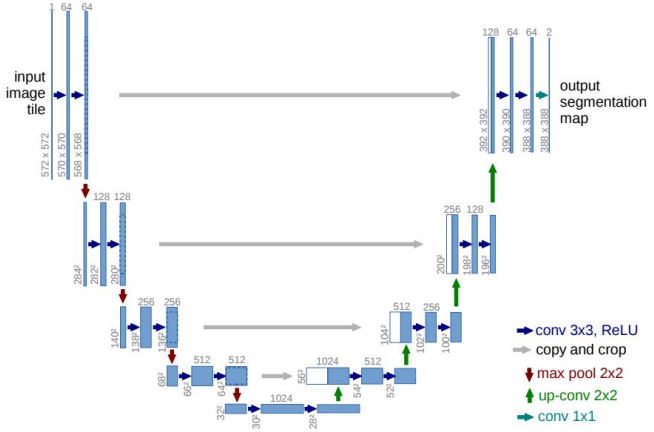
External data (wikimedia)

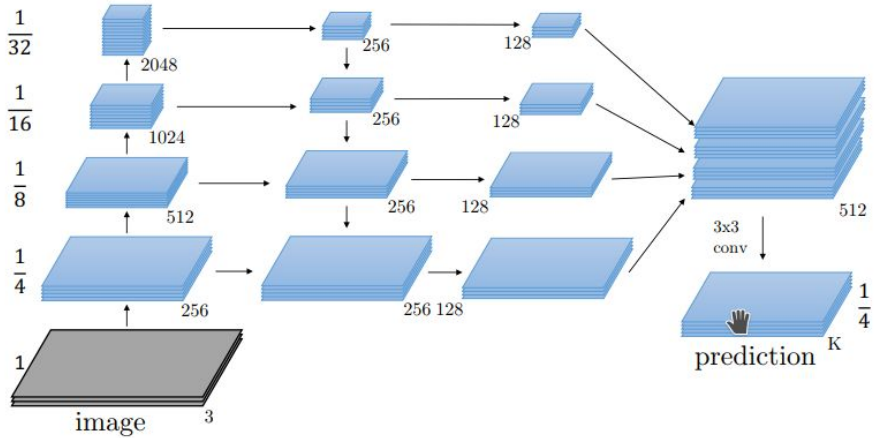


11 images from wikimedia were hand-labeled in gimp and added to each fold

Segmentation networks

1. Standard encoder-decoder architectures with skip connections, similar to U-Net .We used very deep pretrained encoders, with SAME padding everywhere.
2. Encoder-decoder with FPN based decoder, similar to the winning solution for MS COCO Stuff "A Unified Architecture for Instance and Semantic Segmentation".





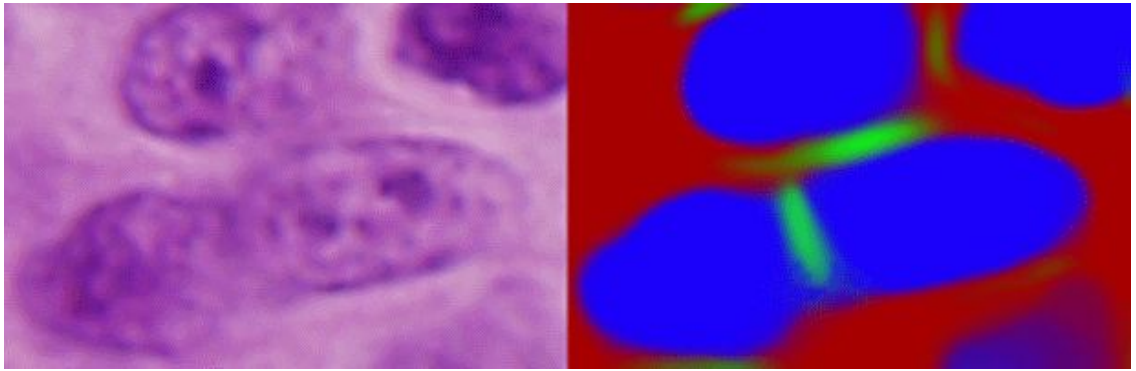
Training

- 4 folds + oversampling by clusters
- Random Crops 256x256
- Batch 16 images
- Adam optimizer
- Small learning rates <= 1e-4
- 20k+ iterations
- takes 4-6 hours for 1 fold on 1080 Ti
- Loss for sigmoid: `bce + soft_dice_loss(mask) + soft_dice_loss(border)`
- Loss for softmax: `0.6*cce + 0.2*soft_dice_loss(mask) + 0.2*soft_dice_loss(border)`
- Hardware: 2 devboxes with 2x1080 Ti, 1 devbox with 2xTitan XP

Validation strategy

1. 4 folds using mosaics as groups
2. best checkpoints by validation loss
3. OOF scores with external data 0.62+ from start (0.47 on the leaderboard)
4. often improvements on OOF scores led to lower scores on the LB
5. OOF postprocessing tuning led to very high thresholds => manual threshold search with visual validation on the stage 1 test images
6. A checklist consisting of hard images from public leaderboard where models usually fail to generalize or split nuclei helped to choose better encoders
7. UNet with VGG16 encoder could not properly predict images from the checklist even after 30 hours of training

Trees and mask expansion (second level model)

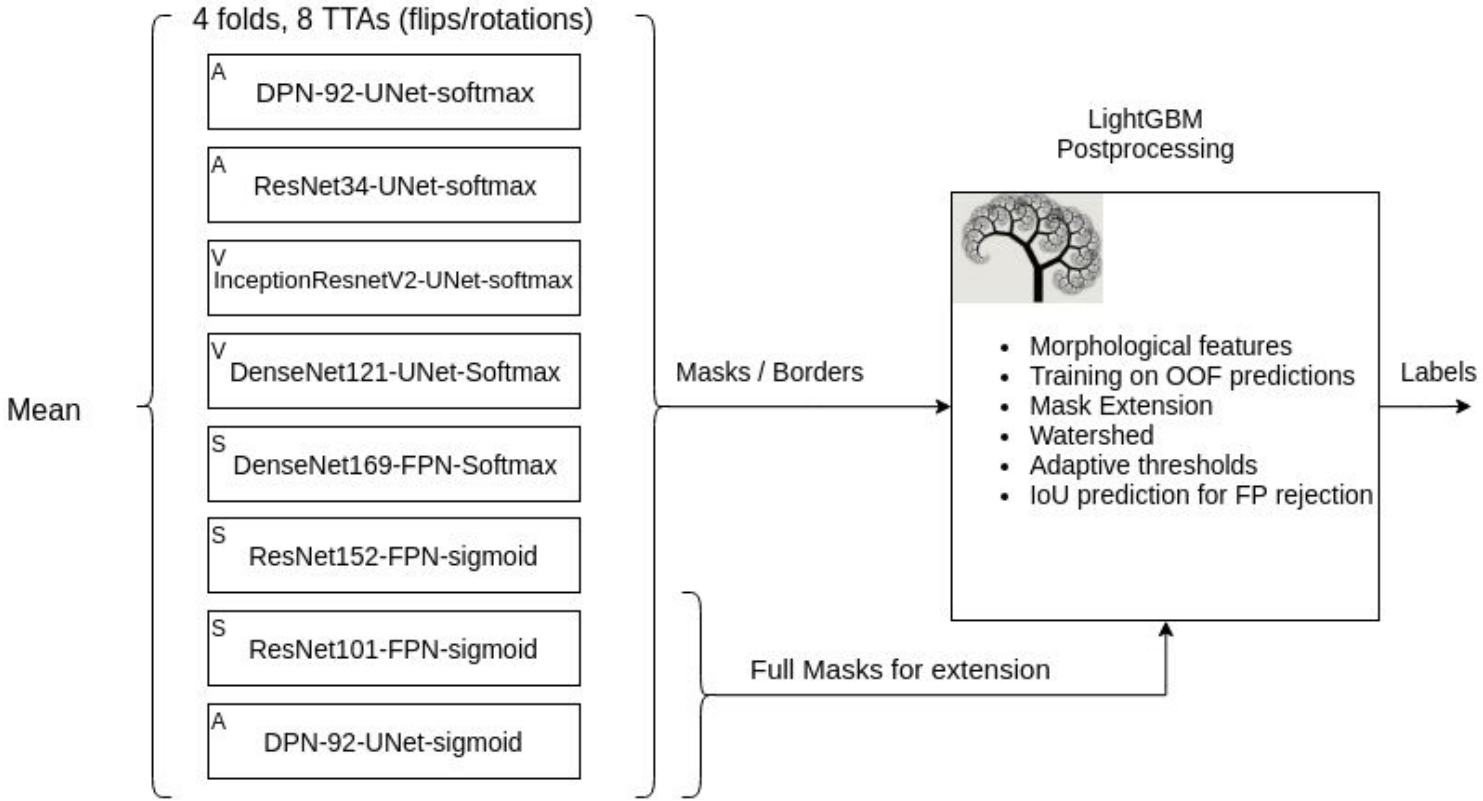


- Different thresholds for candidates
- Expand masks
- Extract features: solidity, circularity, convexity, area, neighbors median area, count, etc.
- LightGBM regression to predict IoU
- Select threshold with highest predicted IoU for each candidate separately
- Remove candidates with low IoU

Base threshold:

High threshold+erosion:

Ensembling/postprocessing combined

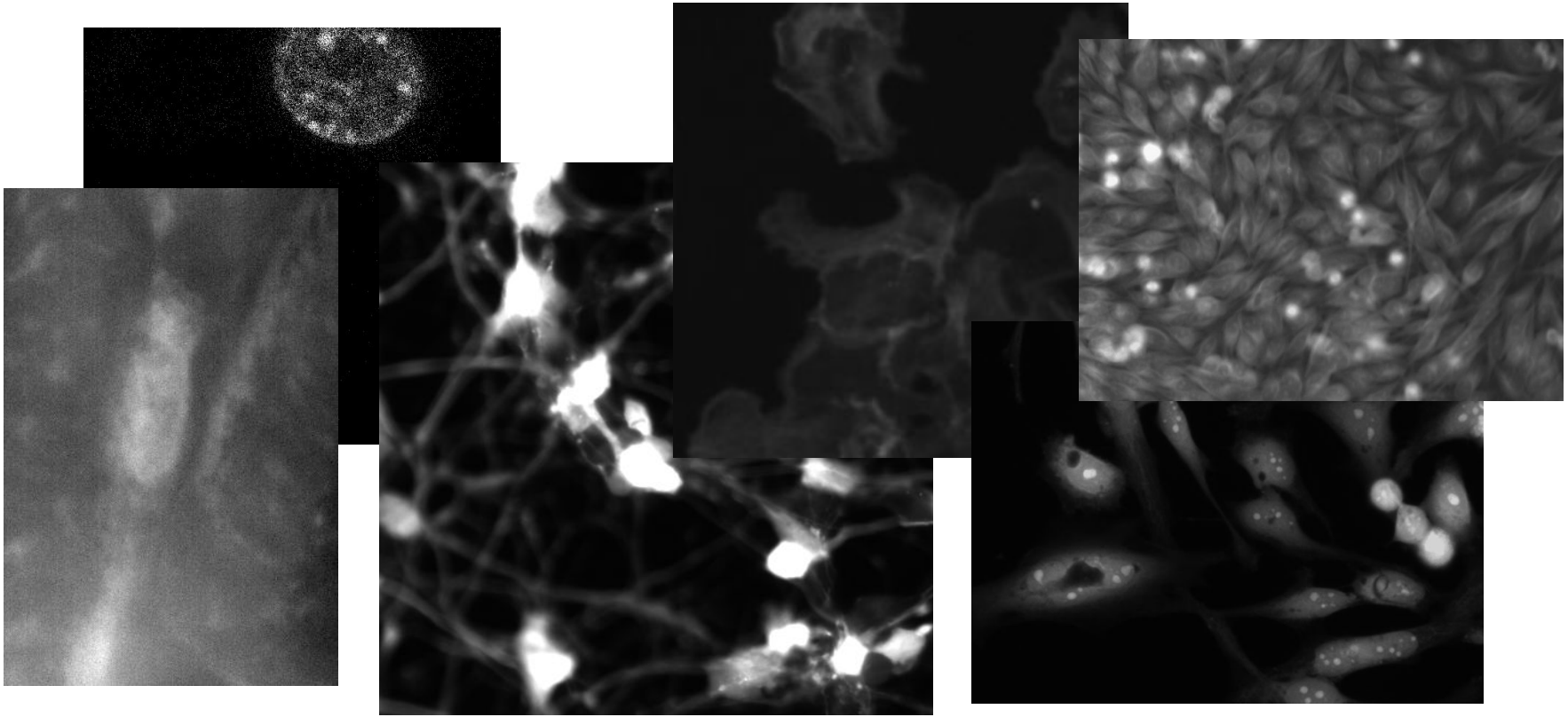


Stage 2 code submission

Stage 2 permitted to tune networks on stage 1 labels so we prepared our code and ran consistency test (few thousands of iterations on final stage 1 prediction, i.e. pseudo labeling). So this is how we jumped from 578 to 598 score in the last day.

Without tuning on stage1 labels our models were still on the first place with a little lower scores. Even 65 additional images (10% from stage1 data) can improve score!

Stage 2 data



Fear and loathing on stage 2

Heng CherKeng

Options

The test2 images are indeed different ...

posted in 2018 Data Science Bowl 10 days ago

31

Looking at the test2 data, I think my results will be terrible .

I note that there are some images without nuclei? Are there some nerve cells as well?

cepera_ang 7:34 PM

Надо было вообще без трейна делать, типа находите какие хотите картинки и попробуйте сделать на неизвестных до второй стадии предикт

cepera_ang 7:59 PM

Дисквалифицируют всех , кто на этой картинке правильно предсказал

💯 7

selim_sef 11:51 AM

DataScienceБоль2018

20 13 18 4

Heng CherKeng

Lesson learned and moving forward

posted in 2018 Data Science Bowl 9 days ago

66

This is an usual kaggle challenge for me. It is the first time i took part in 2 stage challenge. Basically i do commercial algorithm development and i love the idea of "total black-box" testing.

The usual one stage kaggle competition is "gray-box" testing, you see that data, but not the labels. But do note that, whether "black or grey box", this will affect everyone and it is still a fair playing ground.

[things that I did wrong]

1. Spend too much time, trying to find the "perfect" mask rcnn. I though that everything can be learned (including post processing). However, end-to-end solution doesn't exist here. I think here performance is limited by data quality, so trying different architecture, learning parameters etc doesn't seem to work.

cepera_ang 7:34 AM

Да, это тоже кайф, собирали данные откуда смогли)

7:35 AM Кажется этот конкурс по тяге превзойдет рекорд спутников с запасом

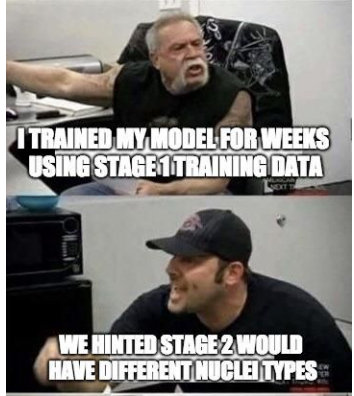
🚀 7 7

Victor 7:36 AM

тяга будет хорошая

cepera_ang 7:38 AM

Можно уже готовить мемасики



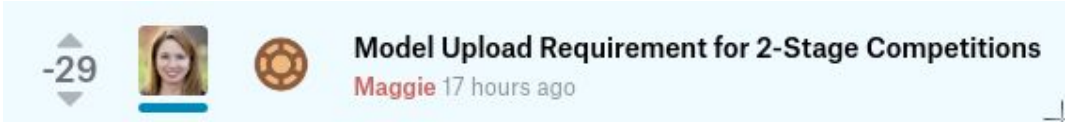
dmytro 7:05 AM

I'm looking forward to your forthcoming Nature paper, "Muffin Adversaries for Biological Imaging!"

стоило фото с яичницей добросить себе в external data, и с собаками

After stage 2: Show must go on!

Kaggle updated the rules for two-stage competitions:



- the old rule was that model upload is required only for prize winners
- now it is required to upload code/model even for just medals/ranking
- the new rule was applied retrospectively…
- more than 50% of the teams are kicked out from stage 2 leaderboard
- top pixel (zero score) benchmark gets bronze!