Carvana Image Masking Challenge

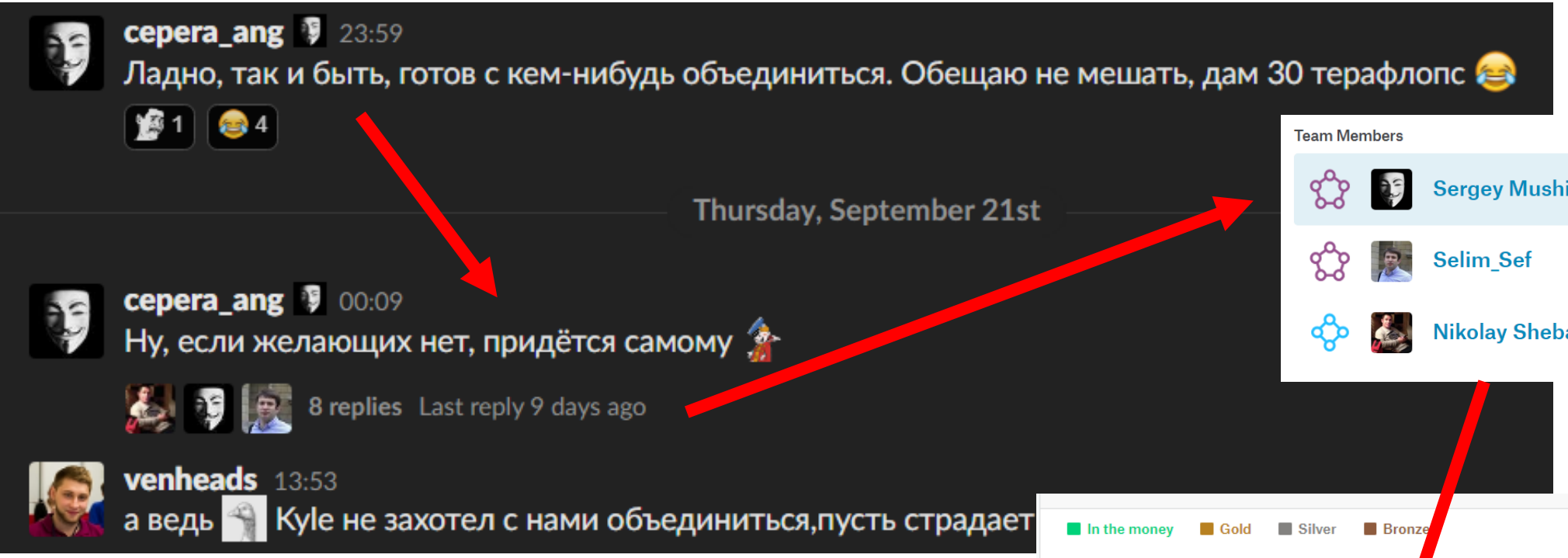
4th place solution

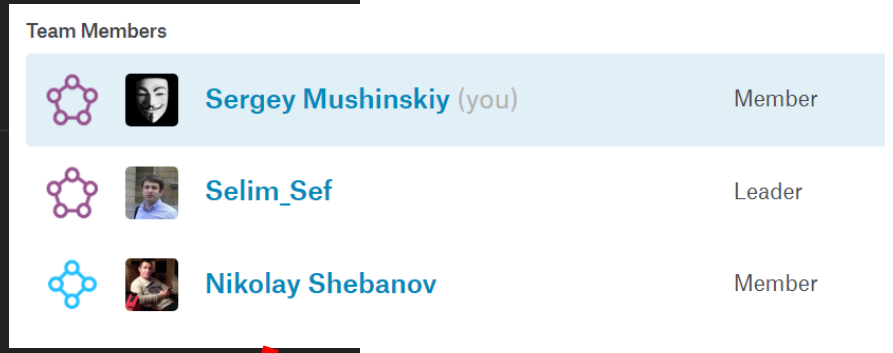
Team 80TFlops

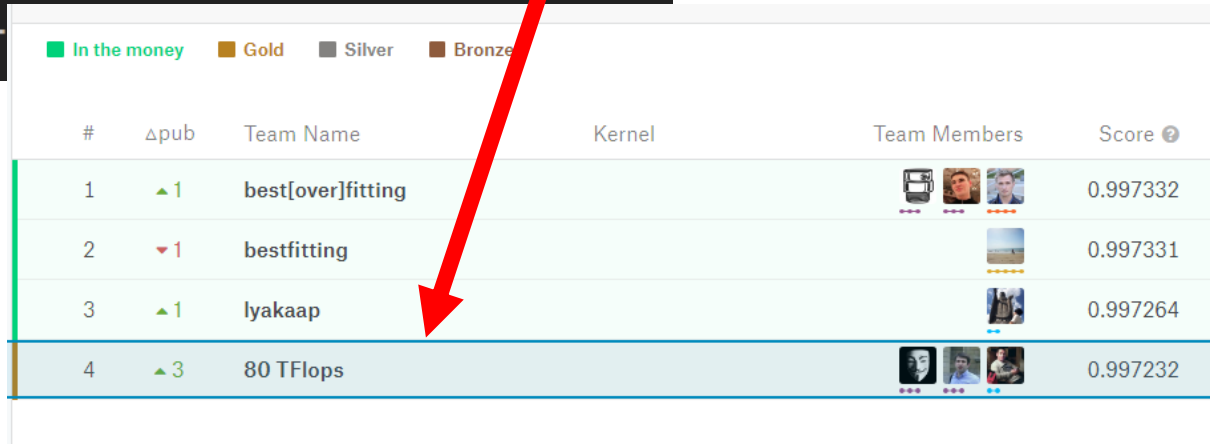
Agenda

- Team
- Challenge
- Solution
- Results
- Drama
- More solutions

Team 80 TFlops







Competition

- Extract mask of a car from photo
- 16 angles, 316 cars, 5056 train photos
- 100064 test photos (only 5k real, so 1200 in Public, 3600 in Private)
- Metric: Dice coefficient
- 2 month, 877 competitors
- Data is pretty clear (instant 99.6% solution. Top: 99.72)

Challenges

- Pretty high resolution
- Painfully slow on train:
  - Single image per second on 1080ti
  - ~1.5hours per epoch
- Awfully slow on test
  - 5 images per second on three cards (2*1080 + 1080)
  - 6 hours full test
  - Times 5 folds, times two TTA: 180 card*hours per network
  - With ~3 trillion pixels even simple averaging takes hours (no time to optimize anything)
- H/W utilization&teamwork

Our solution


UNET
UNET EVERYWHERE

- Baseline: vanilla Unet
- More Unets with pretrained weights (VGG, Resnet-50, etc)
- Inception-Resnet v2, MobileNet
- Tricks:
  - 5 folds
  - TTA: horizontal flips (for best models)
  - Pseudo-labeling: pretrain networks on full test predicted before
- Ensemble everything: the more the better
  - not always, weak networks add only if used in small amount
  - Simple mean between all masks
- Second layer Unet: refine masks with another neural network
- But…


SO MANY IDEAS
NOT ENOUGH GPUS
risova

Teamwork

- It is crucial to fully utilize limited resources
  - Teammates time (everyone working full time)
  - CPU and GPU time
  - Shared storage for results
- Usual routine:
  - GitHub repo with ready to run code
  - Trello board with full pipeline:
    - what everybody going/what hardware doing/when results expected/backlog of ideas
  - Google Drive with full results:
    - Trained models, predictions, etc.
  - Slack channel

One crucial trick

- There was single van in train
- Almost every network failed
- Filling hole with the  trick gave +0.0002

Some fun moments

cepera_ang 21:38

uploaded this image: **image.png**

3

cepera_ang 21:38

Гуси формируют тевтонскую свинью, готовится прорыв линии тренда

12

n01z3 01:30
In the money!

4 3 4 1

nizhib 01:30
... добрый вечер

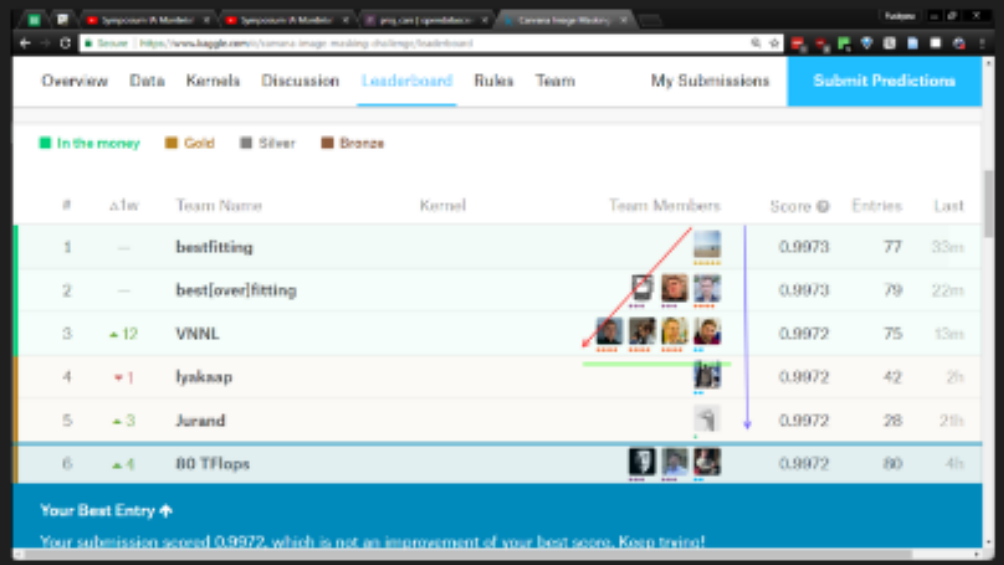
cepera_ang 01:30

Драма невер стопс

shvetsiya 01:32
@cepera_ang ваш ход 😉

cepera_ang 01:33 ☆
uploaded this image: **Как по учебнику: идеальный обрат**

😂



submission-2017-09-28_01-03-56.csv... a minute ago 61 seconds 0 seconds NULL

Behind 15 others

We are going to need more mechanical turks...

Thursday, September

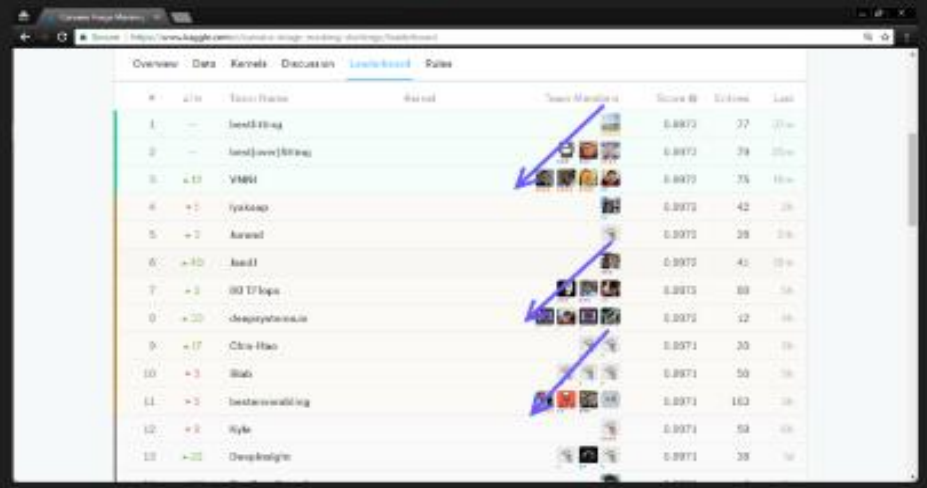
Всем спасибо, все красавцы

18

cepera_ang 03:04
Вы тоже навели шороху в последний момент!

Смотрим на чарты!

cepera_ang 03:04 ☆
mentioned this image: **Признак большого шейкапа на прайвате** 😂



😂 6

cepera_ang 03:05

8

Other solutions


UNET
UNET EVERYWHERE

In the money | Gold | Silver | Bronze

#	△pub	Team Name	Kernel	Team Members	Score	Entries
1	▲1	best[over]fitting			0.997332	80
2	▼1	bestfitting			0.997331	78
3	▲1	lyakaap			0.997264	43
4	▲3	80 TFlops			0.997232	82
5	▲7	Kyle			0.997209	59
6	▼3	JbestDeepGooseFlops			0.997190	76
7	▲1	deepsystems.io			0.997151	12
8	▲17				138	16
9	▲13	lizy			126	25
10	▲20	David			0.997123	65

Basically, just be a part of ODS

7th place: deepsystems.io

- Credit to Renat Bashirov
- Unet-1:
  - Find bbox
  - Pad 150px
  - Goto 2
- Unet-2:
  - unet with resnet head
  - Crops 512x512
  - Full res inference
  - TTA: flips
- No ensembling, no folds

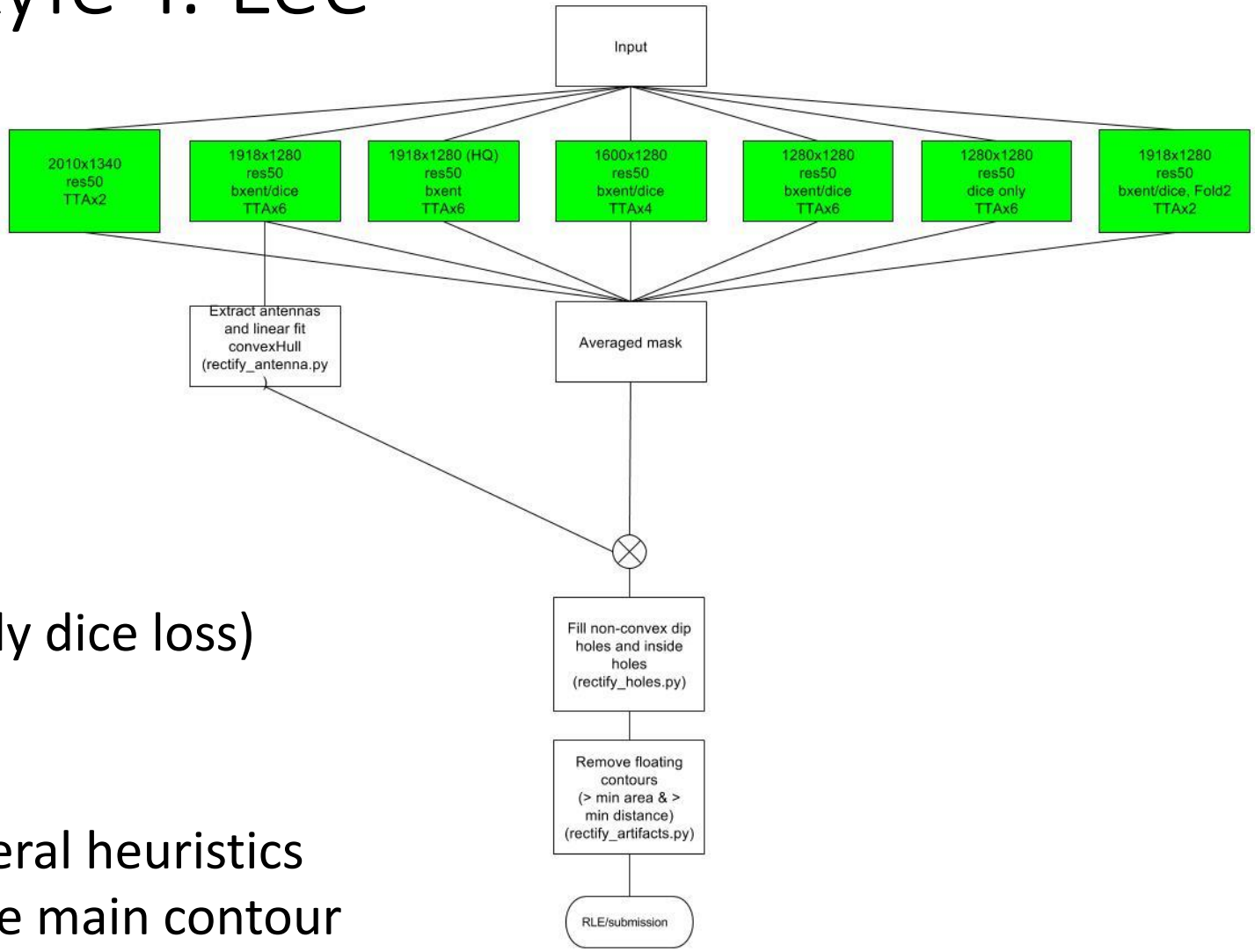
6th place: JbestDeepGooseFlops

- Masters of crazy ensembling and insane architectures
- Like "incnet" — inception-v3 + linknet decoders and "dinknet" — densenet + the same decoders.

		TTA		OOF	mean	0	1	2	3	4	std
venheads	unet_mix_opt_loss	???	CV		0.99679	0.99680	0.99667	0.99689	0.99683	0.99677	0.00008
			LB	**0.9970**	0.99656	0.9966	0.9963	0.9966	0.9965	0.9968	0.00018
n01z3	linknet18_mxnet	1crop	CV		0.99667	0.99668	0.99660	0.99677	0.99679	0.99653	0.00011
			LB								
	linknet34_mxnet	1crop	CV		0.99640	0.99678	0.99633	0.99614	0.99655	0.99622	0.00026
			LB								
	linknet18_pytorch	1crop	CV		0.99681	0.99683	0.99669	0.99691	0.99686	0.99677	0.00008
nizhib	linknet_18_new	1crop	CV	0.99696	0.99687	0.99692	0.99681	0.99691	0.99687	0.99683	0.00005
			LB								
	linknet_34_new	1crop	CV	0.99711	0.99703	0.99705	0.99699	0.99709	0.99704	0.99696	0.00005
			LB								
	incnet_full	1crop	CV	0.99711	0.99705	0.99706	0.99703	0.99709	0.99708	0.99697	0.00005
			LB								
	dinknet_full	1crop	CV	0.99713	0.99707	0.99711	0.99705	0.99712	0.99709	0.99699	0.00005
			LB								
roman	pspnet_18_full	1crop	CV			0.99690	0.99663	0.99663	0.99664	0.99663	
			LB								
	pspnet_resnet34	1crop	CV			0.99678	0.99660	0.99667	0.99684	0.99670	

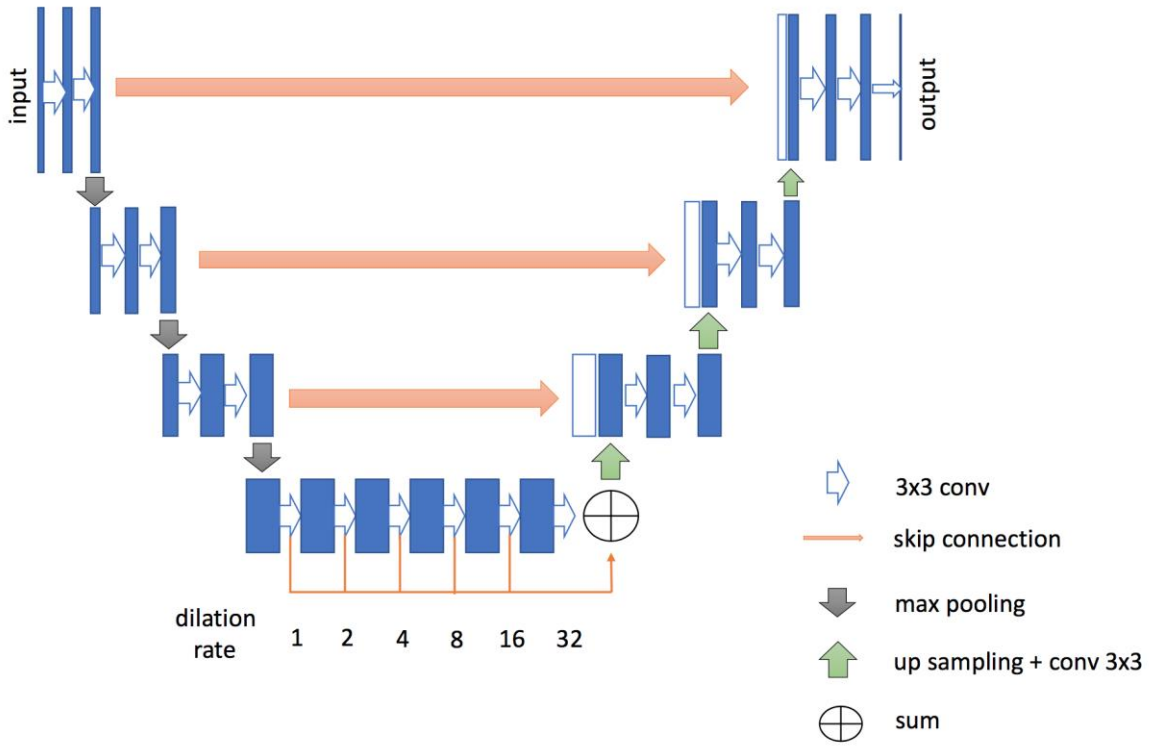
5th place: α  Kyle Y. Lee

- Resnet-50 based FCN
  - 1918x1280(HQ, TTA6x)
  - 1918x1280(TTA6x)
  - 1918x1280(HQ, TTA2x)
  - 1600x1280(TTA4x)
  - 2010x1340(TTA2x)
  - 1280x1280(TTA6x)
  - 1280x1280(TTA6x, using only dice loss)
- Extreme post processing
  - Find&connect antennas
  - Close holes in cars with several heuristics
  - Remove anything far outside main contour



3rd place: lyakaap (simplicity is the king)

- Used 1536x1024 & 1920x1280 resolution.
- Modified U-Net:
  - It has several dilated convolution layers in bottleneck block. (i.e. where the resolution of feature maps are lowest)
- only horizontal flip.
  - Scaling, Shifting, and Shifting HSV were results in overfitting.
- Pseudo Labeling
- Loss function: bce + dice loss
- Ensemble: 5 fold ensemble @1536x1024 + 6 fold ensemble @1920x1280, weighted average by LB ranking



1st place: best[over]fitting

- Alexander Buslaev:
  - Linknet + resnet34
  - ImageNet pretrain
  - Adam -> RMSProp
  - CLAHE
  - Hard negative mining
  - HSV augmentation
  - TTA: hflips
- Artem Sanakoyeu:
  - Unet from scratch
  - Unet from VGG11
  - Unet from VGG11 + transposed conf
- Vladimi Iglovikov:
  - Unet from VGG
  - HUE augmentation + flips
  - Cyclic LR
- General:
  - Cyclic LR
  - Pseudo Labels
  - Smart ensembling

~2.5 pixels per image difference with the 2nd place

Honorable mentions

- 10th place:
  - 1st model: Background detection.
  - 2nd model: Classic Unet architecture
  - 3rd model: Same Unet on images in half
  - Post processing: clear antennas
- 11th place:
  - Stage 1: Predict rough outline of mask with 1/4 downscaled image
  - Stage 2: Train/Predict mask only using images around edge areas
- 15th place:
  - Predict a coarse mask (CO). For this we used a regular UNET @ full resolution.
  - Refine the mask using patches around the mask contour
- 23rd place: bestensembling (fell 12 places :( )
  - Second layer Unet
  - Unlucky van?

Thanks for watching