Semantic Segmentation

Convolutional Neural Networks

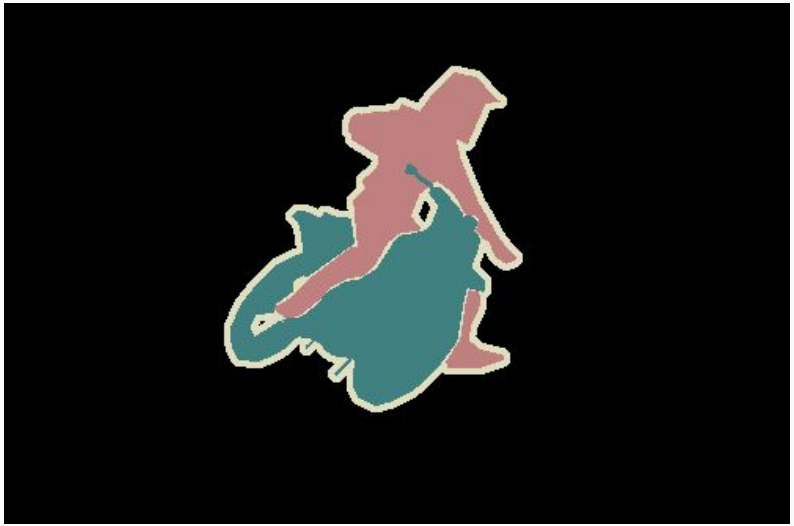
MSU/MIPT Fall 2020

Semantic Segmentation

The goal is to get pixel-level predictions

Pascal VOC was introduced in 2007

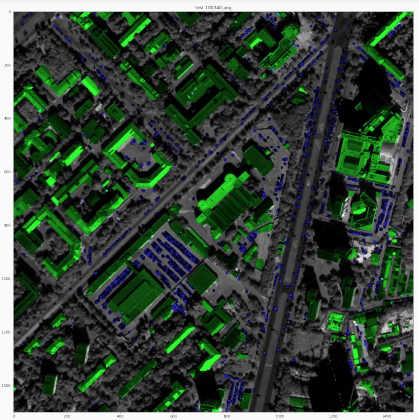
First CNN-based approaches — 2014

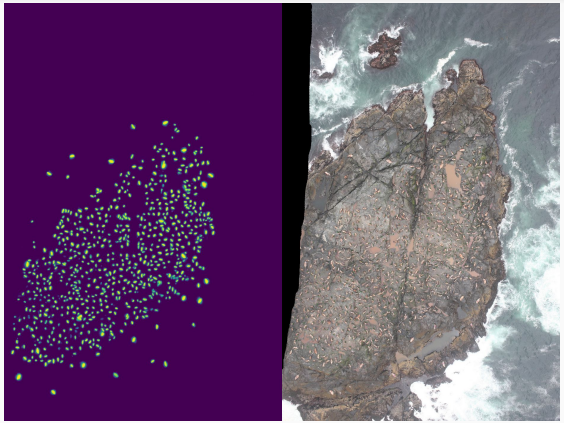


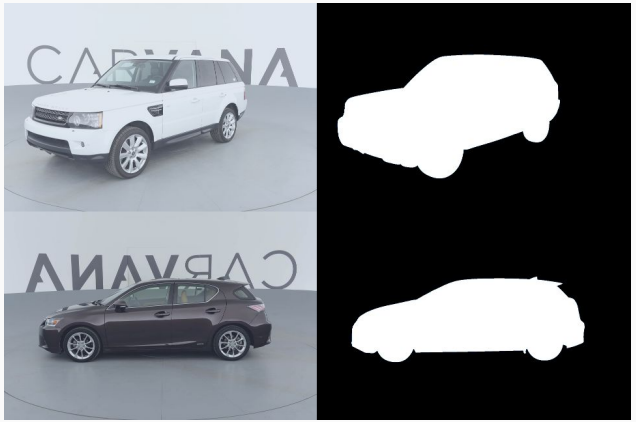
Dataset	Training	Testing	#Classes
CamVid	468	233	11
PascalVOC 2012	9963	1447	20
NYUDv2	795	645	40
Cityscapes	2975	500	19
Sun-RGBD	10355	2860	37
MS COCO '15	80000	40000	80
ADE20K	20210	2000	150

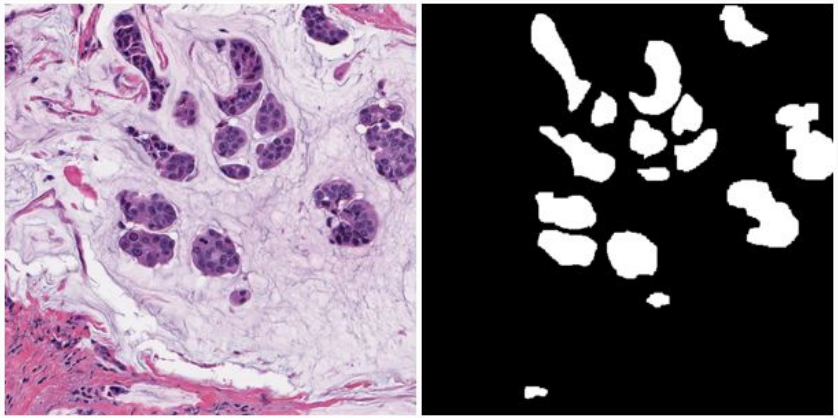
+ Mapillary Vistas, Apollo, Lyft contest, ...
+ SpaceNet, xView[2], ...

Relevance









Relevance

As of Oct 2020:

- [signate] The 4th Tellus Satellite Challenge
- [drivendata] TissueNet: Detect Lesions in Cervical Biopsies
- [topcoder] Spacenet 7: Multi-Temporal Urban Development Challenge
- [kaggle] Lyft Motion Prediction for Autonomous Vehicles

Relevance

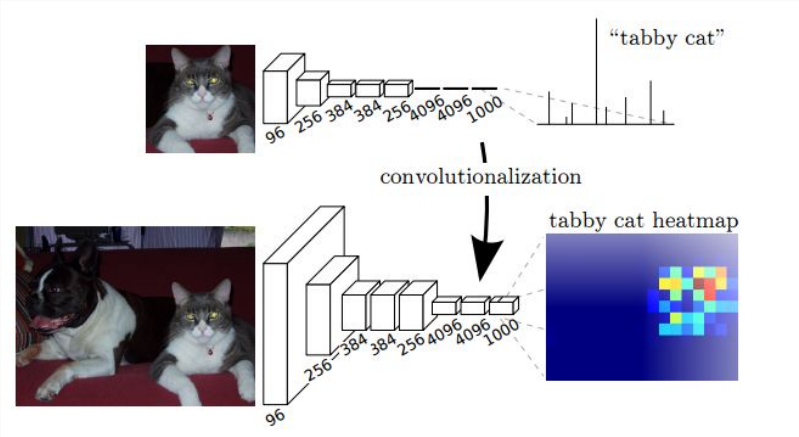
Fully Convolutional Neural Networks for Semantic Segmentation

The final prediction for 224x224 image is a 1x1 semantic map

Make the network output larger semantic map for larger images

Effective resolution downscale can be x1/32, not x1/224 (+Avg 7x7 for resnets)

Convolutionalization — making network fully-convolutional (FC layers -> 1x1 convs)

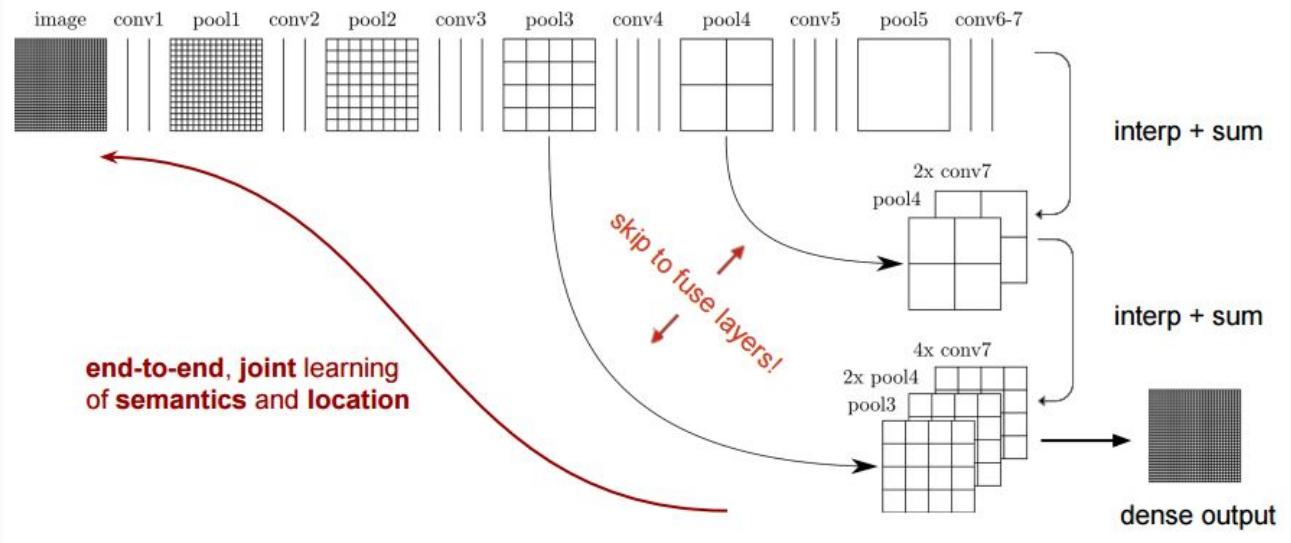


x32 upscale is still needed to get full-resolution semantic maps though

Fully Convolutional Neural Networks for Semantic Segmentation

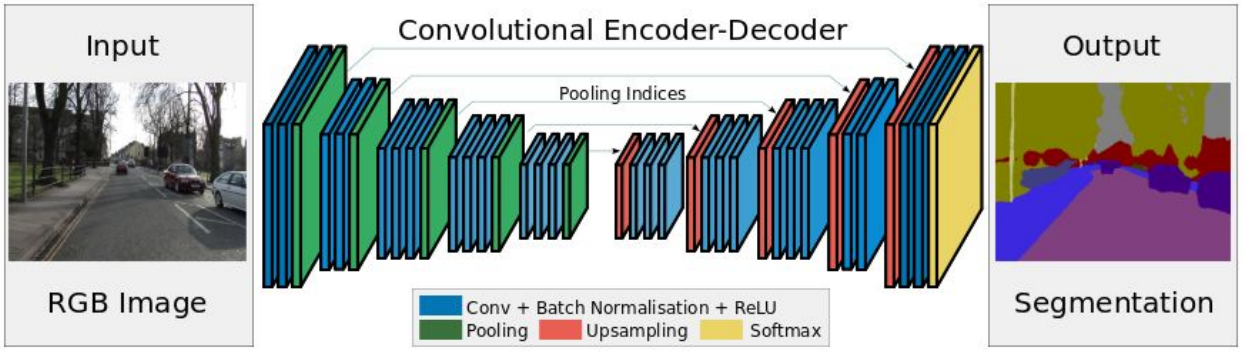
Resolution downgrading after pooling ops results into bigger conv layers effective area, thus lowering final predicitions positioning accuracy

It was proposed to use preliminary layers outputs to increase the accuracy



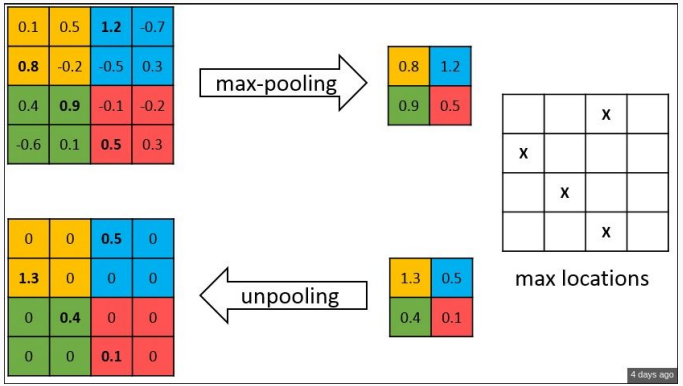
SegNet: A Deep Convolutional Encoder-Decoder for Image Segmentation

One of the first “Encoder-Decoder”-based approaches



A special **unpooling** layer is used to restore the full resolution semantic map

The network is light-weight as no FC-layers are used



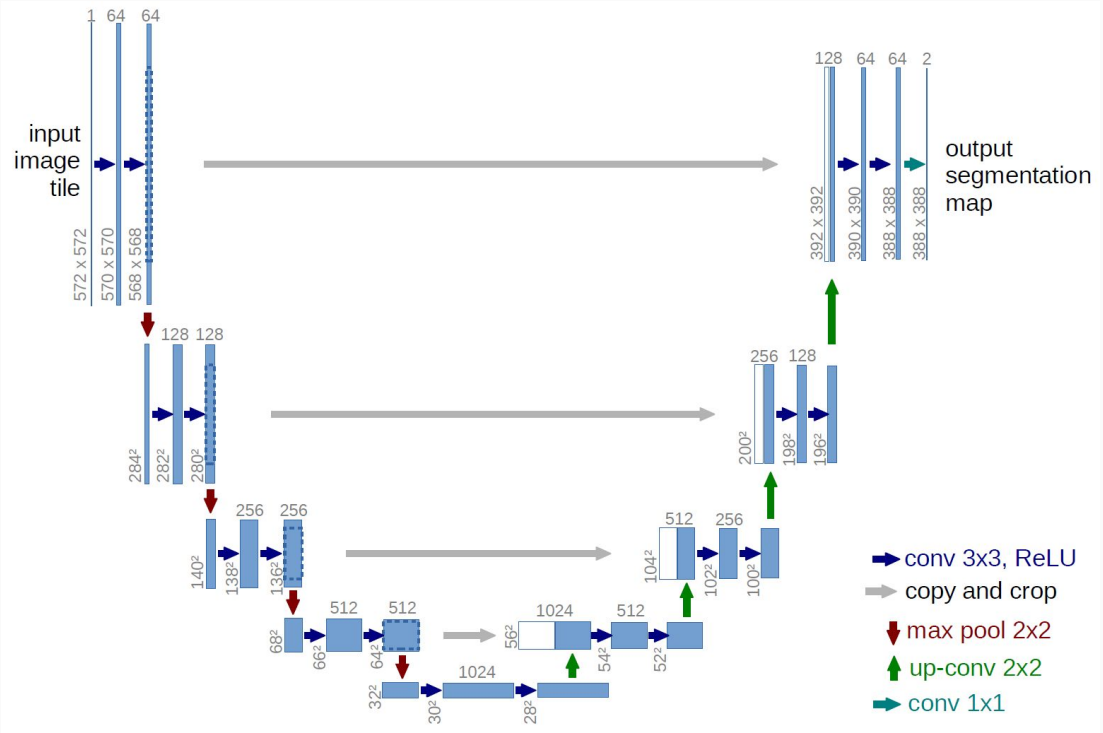
U-Net: Convolutional Networks for Biomedical Image Segmentation

Initially introduced in a medical journal, used for segmenting medical images

Encoder preliminary outputs are concatenated with decoder outputs, then upsampling is used

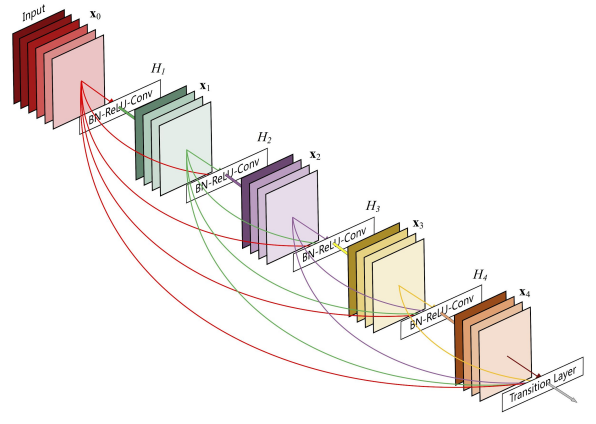
Wasn't scored on popular datasets

Became acknowledged as a part of many top-placed contests solutions from Ultrasound Nerve Segmentation (Aug 2016) to Kaggle DSB (2018), TGS Salt (2018) and Airbus Ship Detection Chellenge (2018).



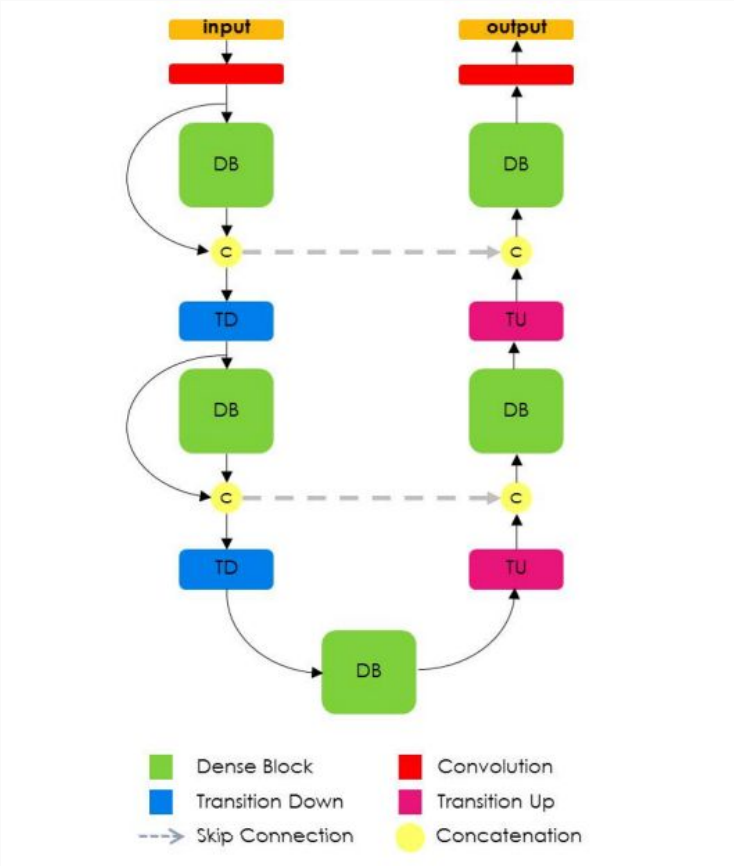
The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for SemSeg

Applies DenseNet ideas to Enc-Dec architectures (which in turn inherits ResNet ones)



Both encoder and decoder use Dense Blocks

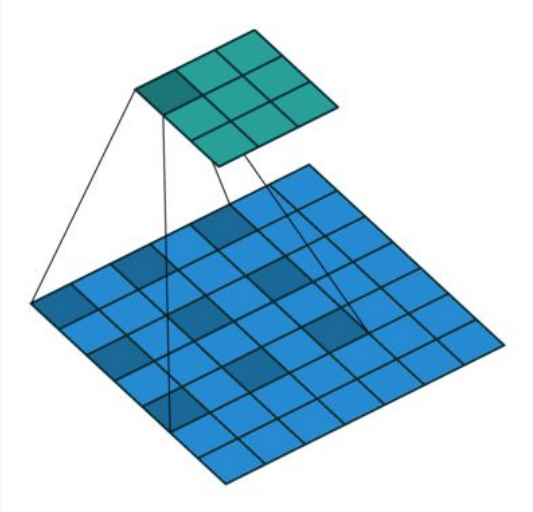
Famous for being used in several competitions with no success stories records found up to date



Multi-Scale Context Aggregation by Dilated Convolutions

Besides existing approaches of initial resolution restoration some developed approaches to avoid resolution reduction in the first place

Dilated/Atrous convolutions — have bigger effective resolution with no need in pooling ops

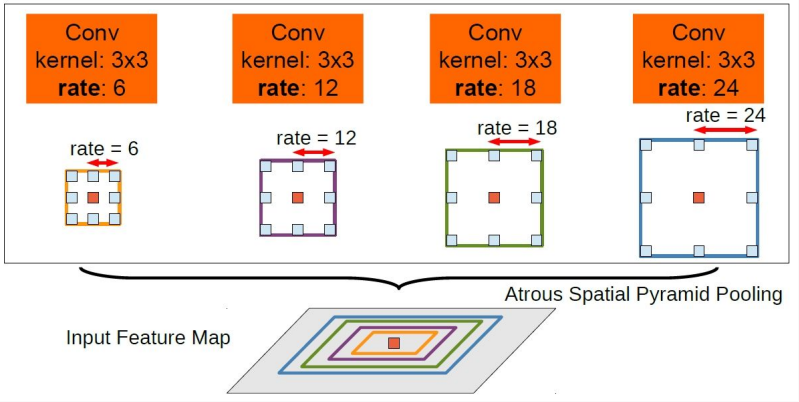


It raises theoretical complexity, but practical one suffers even more than that

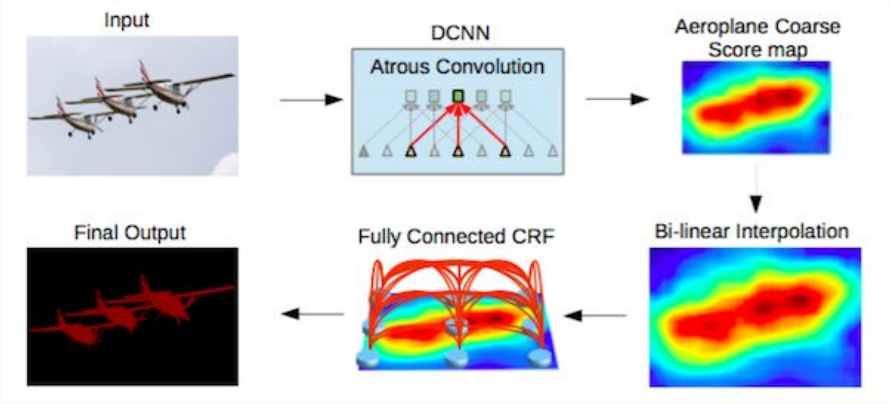
DeepLab (v1 & v2 & v3 & …?)

Replaced “common” convolutions with dilated ones in the Encoder.

ASPP - aggregation of different dilated convs



CRF is used for post-processing

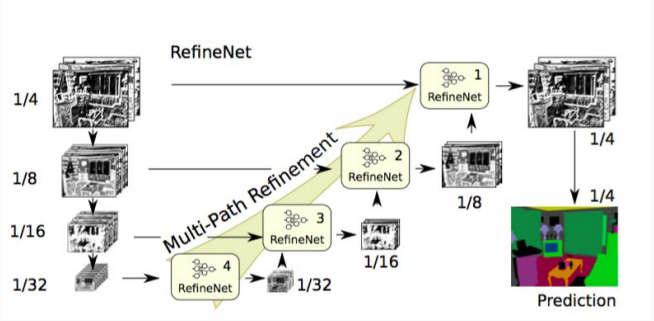


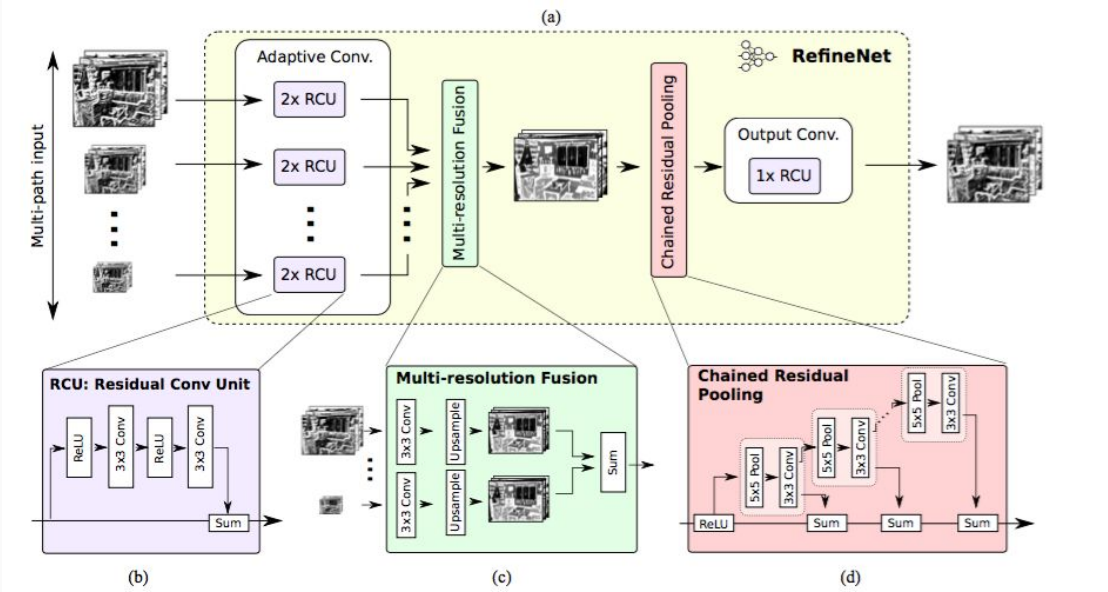
MS COCO-pretrained models were evaluated on VOC 2012

Best: ResNet-101 + atrous Convolutions + ASPP + CRF (very heavy)

RefineNet: Multi-Path Refinement Networks for High-Resolution SemSeg

An example of complex “enhancers” on decoding stage





“Enhancers” include

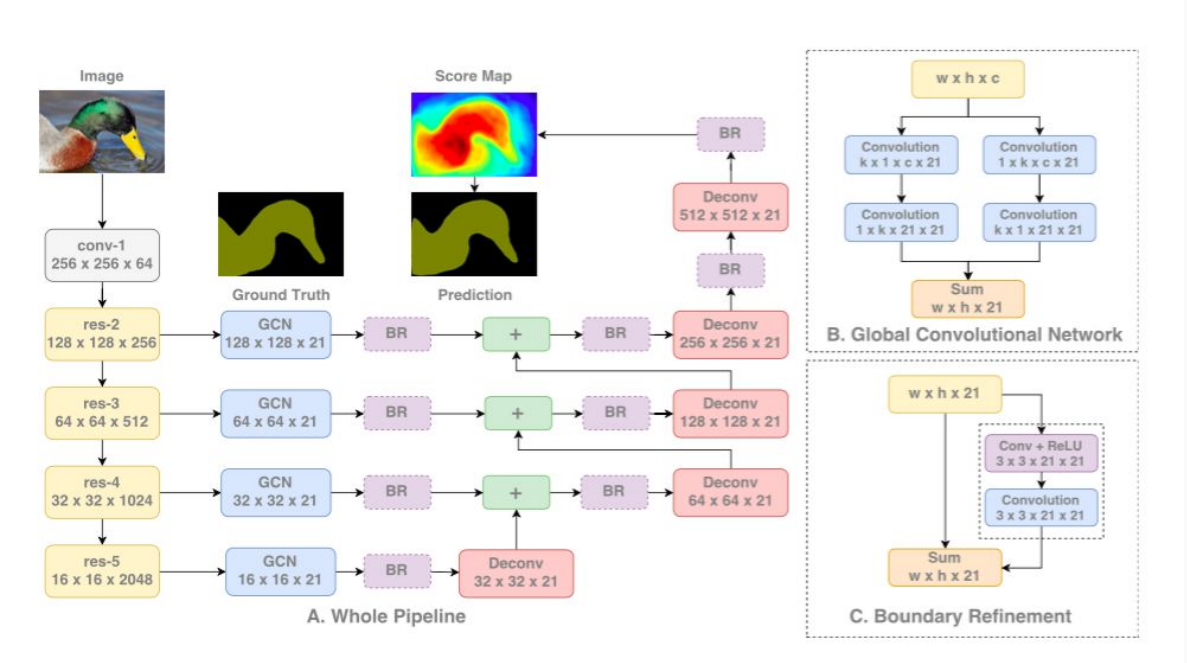
- ResNet Residual blocks
- Mixing blocks of different image scales
- Context aggregators

Each of them but the lowest one concatenates inputs from 2 different scales

Large Kernel Matters — Improve SemSeg by Global Convolutional Network

Another example of the architecture with lots of “enhancers”

- Global Convolitonal Network aggregates the context with large k x k-sized convolutions decomposed into 1 x k + k x 1
- Boundary Refinement is a residual-enhancer, similar to GCP (RefineNet)



LinkNet: Exploiting Encoder Representations for Efficient SemSeg

Another Encoder-Decoder architecture example

Encoder and decoder outputs are summed up in contrast with U-Net concatenations

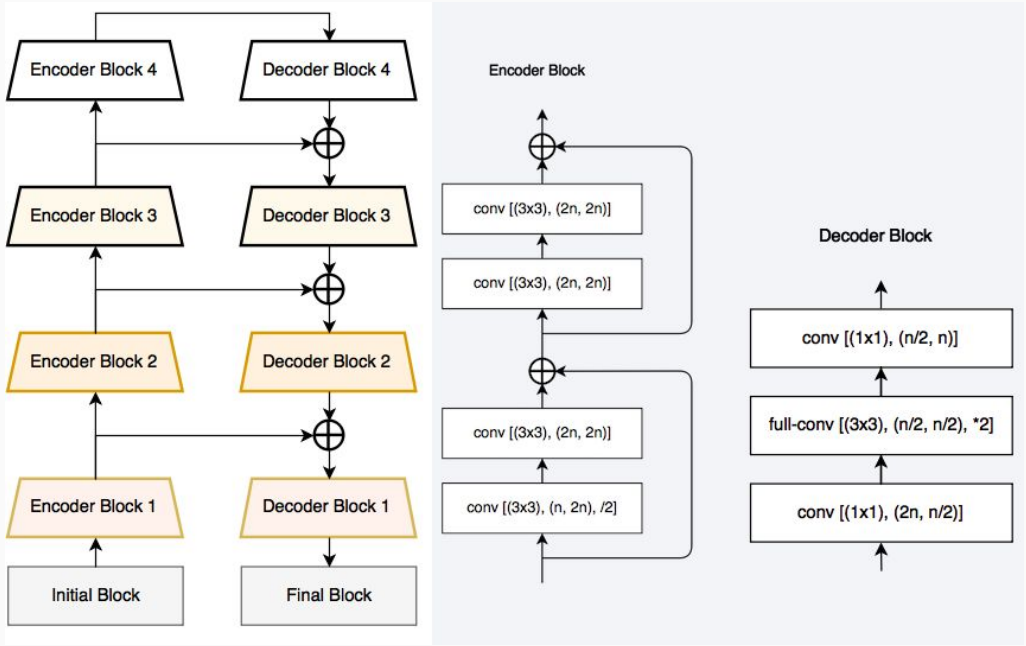
Decoder is kept being light-weight:

- 1x1 convolutions are used for x1/4 resolution reduction
- Transposed convolutions are used for x2 upsampling instead of simple upscaling
- Another convoluional layer is used to restore feature dims

The accruacy is similar to PSPNet

ResNet 18->LinkNet is x2 in infer. time

Successfully used in several competitions



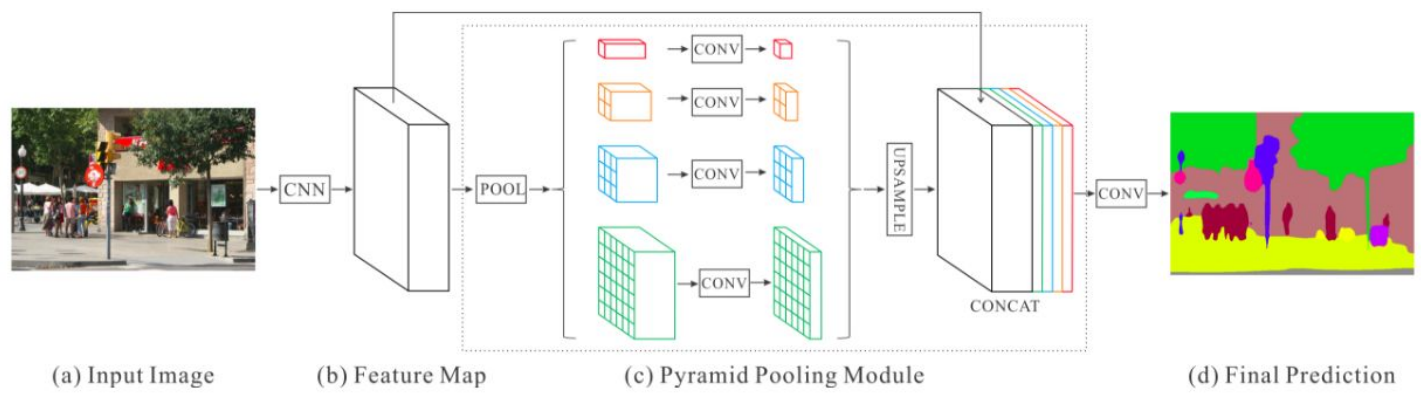
PSPNet: Pyramid Scene Parsing Network

Recent state-of-the-art / on par with deeplab v3 from “heavy decoders” family

Making all the conv layers dilated after sevaral initial ones in ResNet (x ¼)

Additional loss functions applied to ResNet blocks outputs

Spatial Pyramid Pooling module is applied to ResNet output, gathering global context info



SPP helps us to avoid large semantic errors but lacks pixel-level details (x4 upscaling in the end)

Resuming: semantic segmentation

Different approaches in different situations

Encoder/Decoder vs “heavy-weight enhancers and restorators”

(restoring the resolution or avoiding its reduction in the first place)

Details enhancement vs global context engaging

Devil in the Decoder vs Devil in the Encoder

Carvana Image Masking Challenge

In some cases model architectures and lots of rigs is the key to success

In contest we needed to segment vehicles from the shop background

The leaderboard metric is dice score (= #union / #intersection)

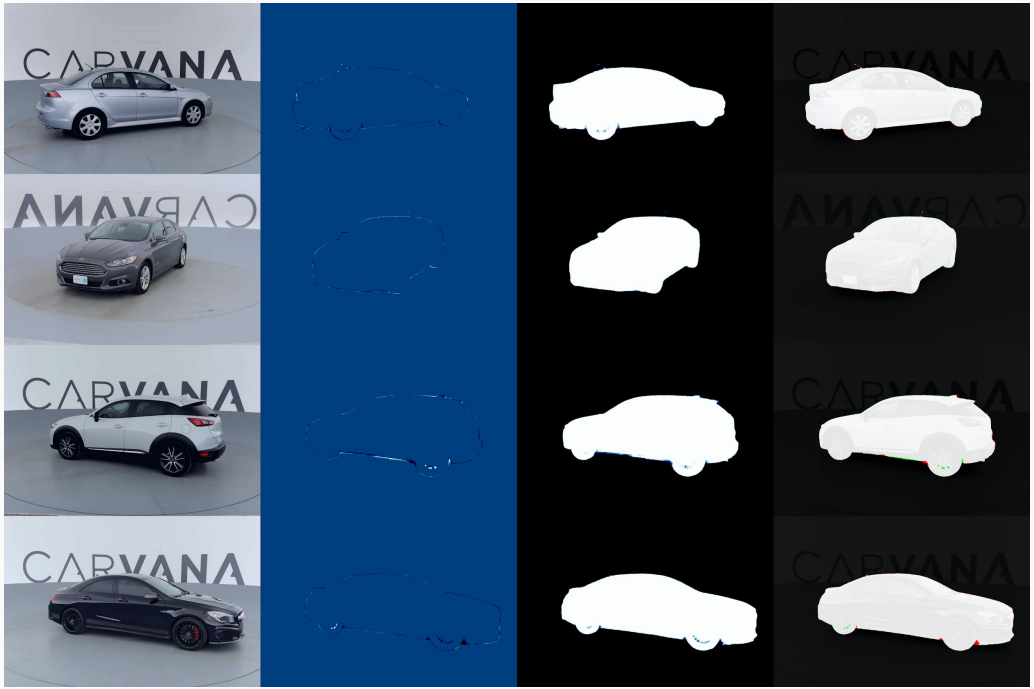
Train dataset: ~5000 images with 16 different viewing angles

Test dataset: ~100 000 images with 5000 that really counted (200 IQ organizers)

Carvana Image Masking Challenge

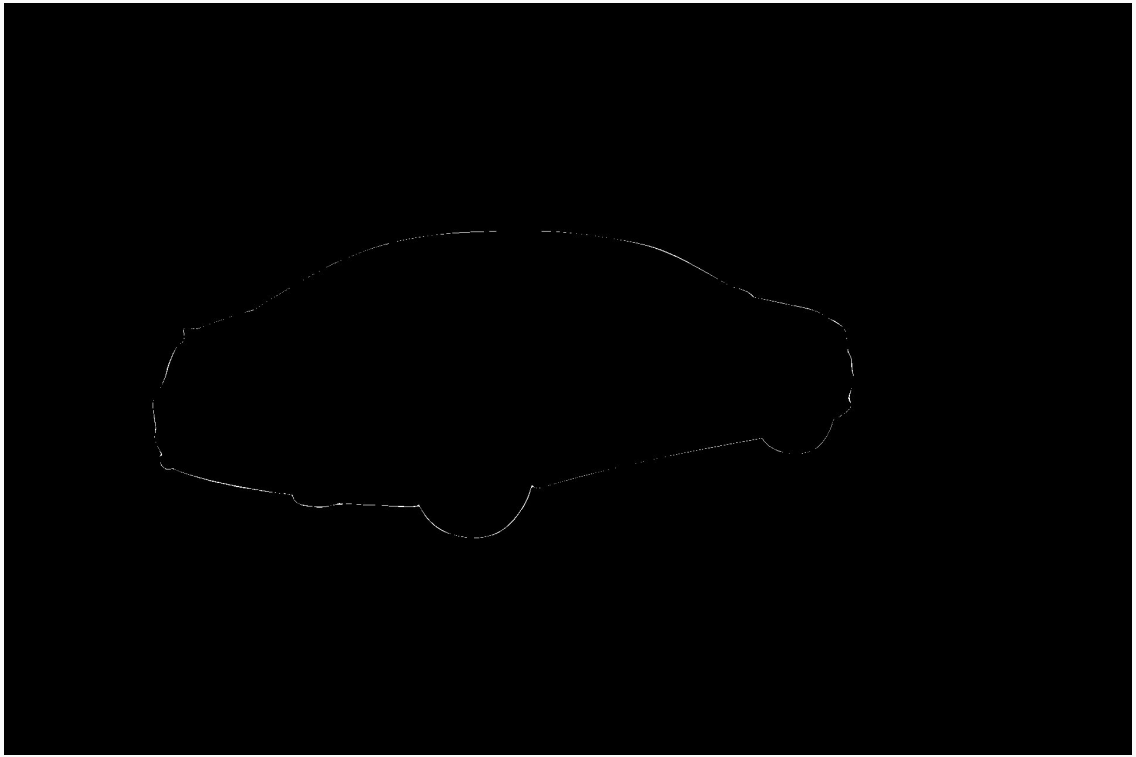
It was easy to achieve dice > 0.996.

The real fight was in the 0.9770-0.9772 range



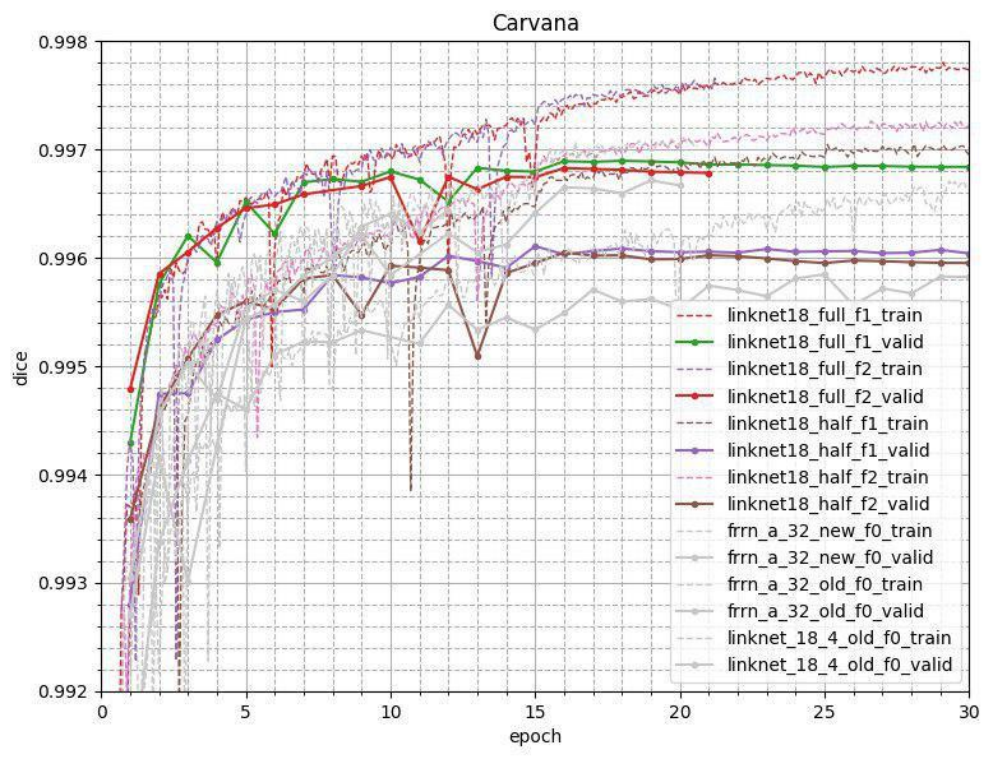
Carvana Image Masking Challenge

In “easy” images not only model accuracies counted but the initial mturk guys accuracy



Carvana Image Masking Challenge

LinkNet had easily achived good scores from the begining. The team decided to keep improving it



Carvana Image Masking Challenge

Links in LinkNet come from natural ResNet resolution reduction points

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

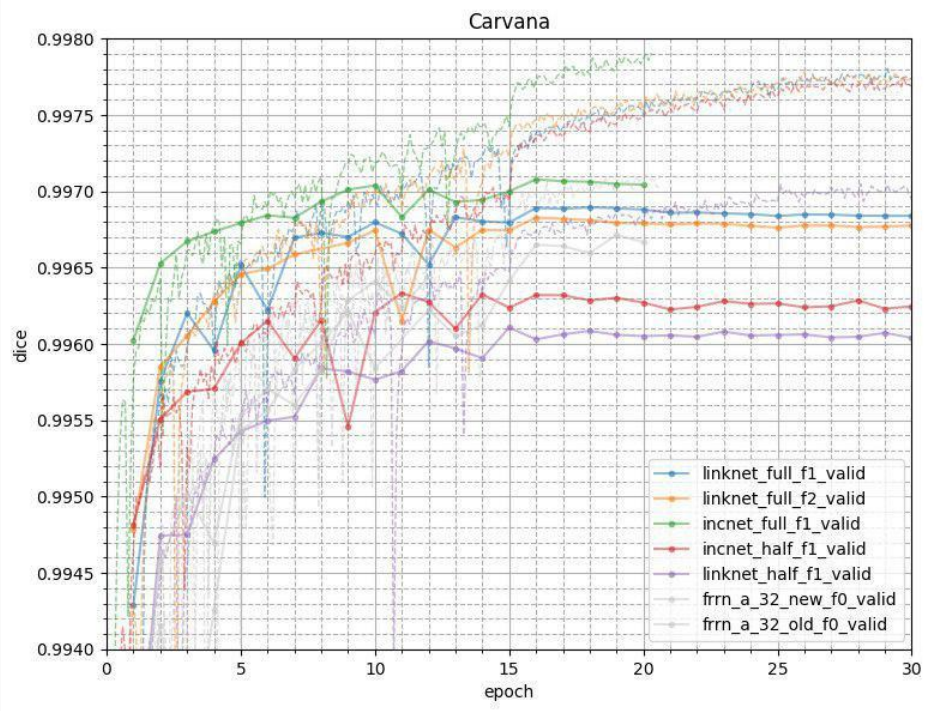
Carvana Image Masking Challenge

One can find similar reduction points in different architectures (Inception-V3 is below)

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Carvana Image Masking Challenge

Several frankenstein LinkNet-based approaches were tried
LinkNet + Inception (a.k.a. “IncNet”) on its own could be used to stay in gold (top-10 LB)



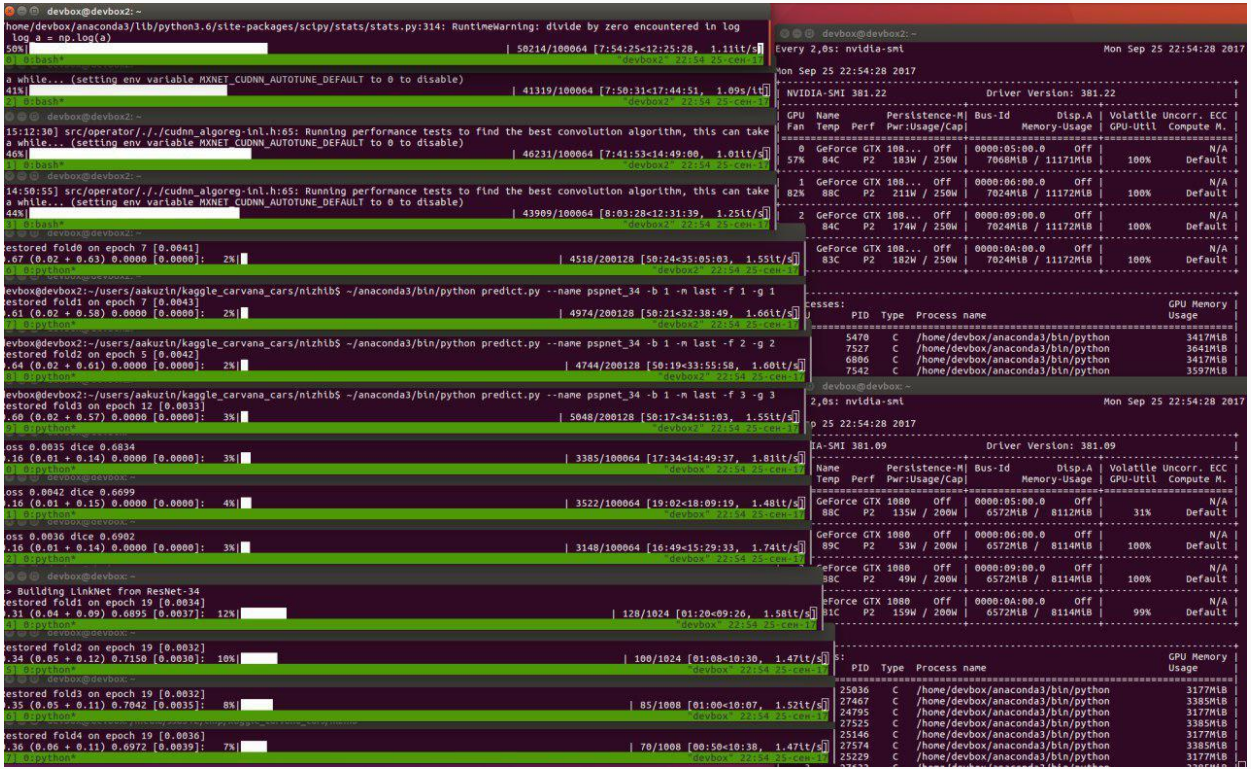
Carvana Image Masking Challenge

The final approach was built upon LinkNet with “heavy” encoders being the best:

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

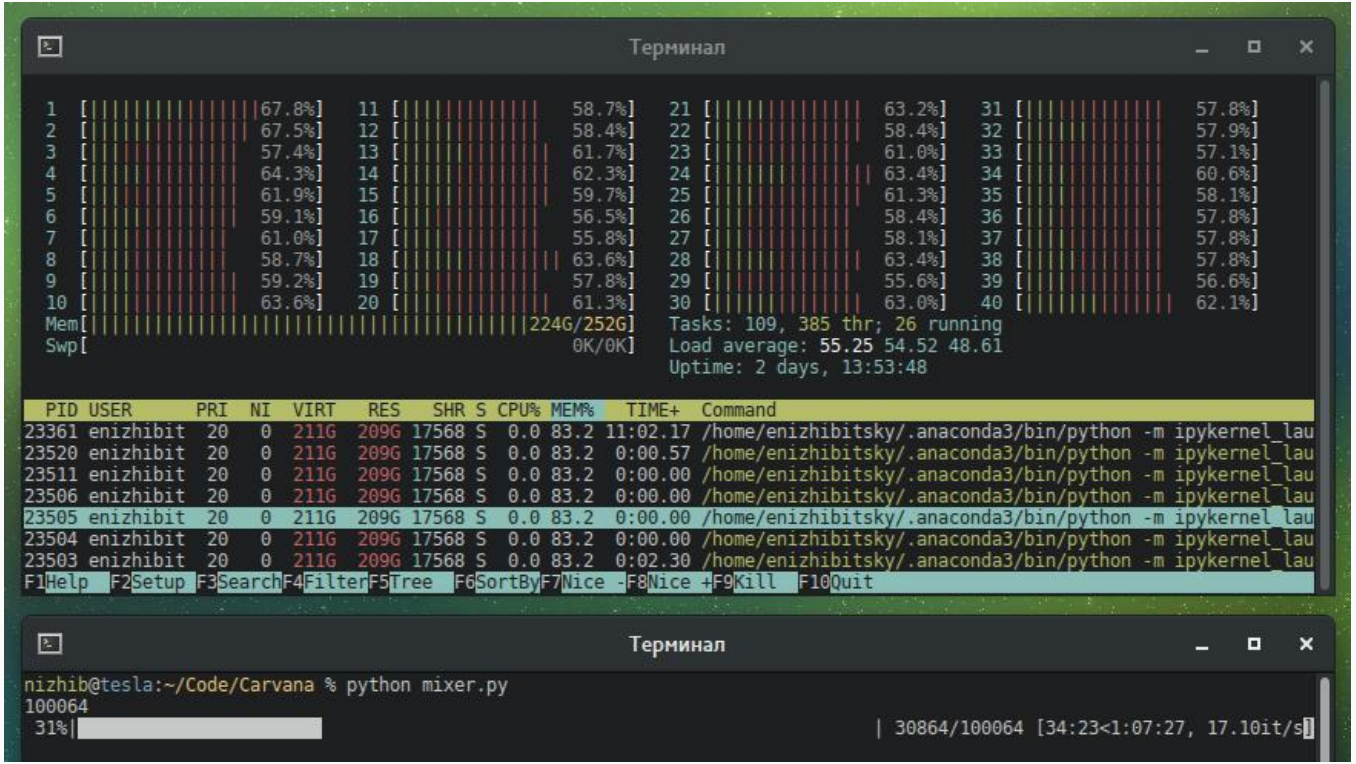
Carvana Image Masking Challenge

“One more thing” left — making 100k full-hd semantic maps with 5x10 models on dozens of GPUs …



Carvana Image Masking Challenge

… and mixing all the resulting semantic maps to create an ensembled prediction



Carvana Image Masking Challenge

We've failed to stay in the money though :(

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	jizs			0.997138	16
9	▲ 13	lizy			0.997126	25
10	▲ 20	David			0.997123	65

* the ones highlithed are those from opendatascience (ods.ai) community

Konica-Minolta: Detecting Abnormality for Automated Quality Assurance

Sometimes the key to success is enlarging the dataset

One needed to decide 1 of 10 camera models

Initial dataset contained several GB of photos, 1000 per each of the camera models

Contrary to common practice of banning external dataset usage or making it explicit no such rule had been added to this contest

#	△pub	Team Name	Kernel	Team Members	Score
1	▲ 1	[ods.ai] STAMP			0.989642
2	▲ 6	[ods.ai] GPU_muscles_SPcup_e...			0.987976
3	▼ 2	FIIGO_SPcup_eligible			0.987857
4	▲ 5	Guanshuo Xu			0.987023
5	▲ 2	[ods.ai] 10011000			0.986547
6	▼ 3	[ods.ai] Evgeny Nizhibitsky			0.986190
7	▲ 4	blzr_SPcup_eligible			0.985595
8	▲ 4	Master			0.985595
9	▼ 5	[ods.ai] SVM punks			0.985357
10	▼ 4	Make Ensemble Great Again!			0.984761
11	▼ 1	[ods.ai] Nokia3310			0.984404

Top-scorers involved flickr, wikimedia, yandex.foto parsers with complex exif-based post-filtering

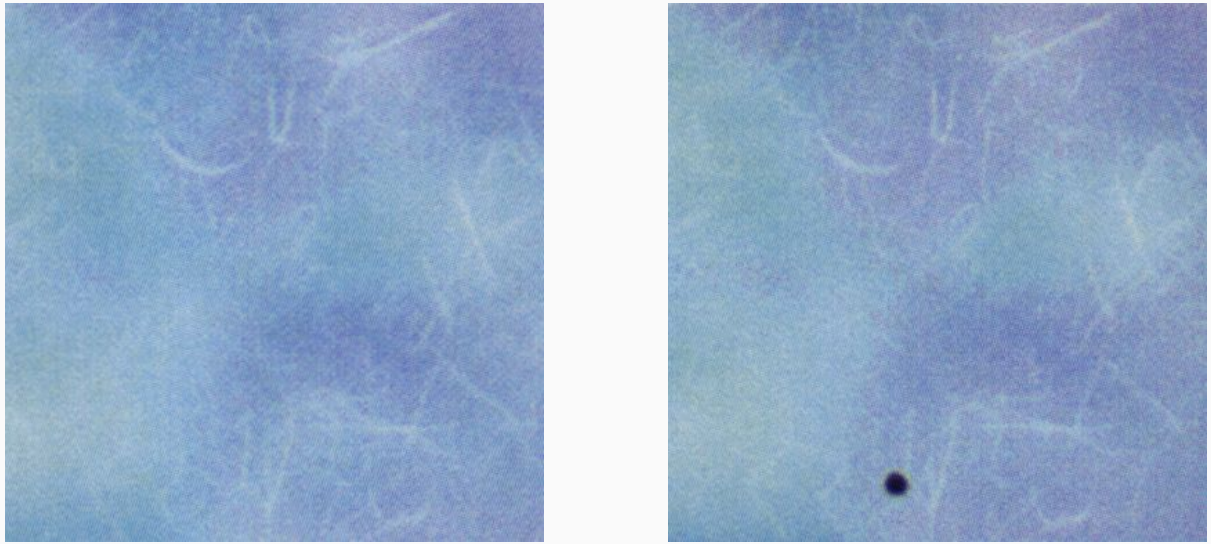
Final datasets contained up to 100-120k photos with total size of 500-600 GB

Konica-Minolta: Detecting Abnormality for Automated Quality Assurance

Sometimes the key is in mindfull data exploration

The task of segmenting dirt (or nuclear rockets?) patterns appeared between 2 screen photo shots

As natural photos are used, there are lots of possible natural differences between them



some easy example from the dataset

Konica-Minolta: Detecting Abnormality for Automated Quality Assurance

Let’s diminish some natural differences with

```
aligned = cv2.findTransformECC(img, ref, np.eye(2, 3), cv2.MOTION_EUCLIDEAN, ...)
```

as in Prokudin-Gorsky RGB photos restoration example:

Konica-Minolta: Detecting Abnormality for Automated Quality Assurance

Look closely at the train and find the difference between actual dirt masks and the masks

While dirt patterns can be several px-sized, the difference can be up to 2-3 pixels on the borders

Let's add magic transformers to fix that

```
abstract.TripleCompose([
    abstract.ImgAndRef(augment.ScaleAndCrop( scale=0.9858, padding=3)),
    abstract.ImgAndRef(augment.Pad101(padding=3)),
    augment.TripleCVToPIL(),
    abstract.Tripled(transforms.ToTensor()),
    abstract.ImgAndRef(normalize)
])
```

And train one simple **vanilla U-Net** (VGG-based, not used much not actually, **not pretrained**)

That results into 0.804 dice. Training another one results into 0.807 dice.

Konica-Minolta: Detecting Abnormality for Automated Quality Assurance

Final leaderboard

Rank	Handle	Provisional Score	Final Score
1	nizhib	807,010.58	785,837.62
2	n01z3	801,152.65	777,559.13
3	albu	801,070.87	775,202.37
4	codecrux	797,608.39	764,023.51
5	selim_sef	790,481.22	756,822.47
6	cannab	785,037.06	755,475.96
7	wleite	791,016.02	751,224.69
8	ipraznik	780,653.56	747,202.84
9	nofto	760,183.48	717,308.10
10	Mloody2000	740,908.33	714,981.75

Cinema Headcount and Socdem Classification

We want to evaluate ads coverage
and classify the audience in cinemas

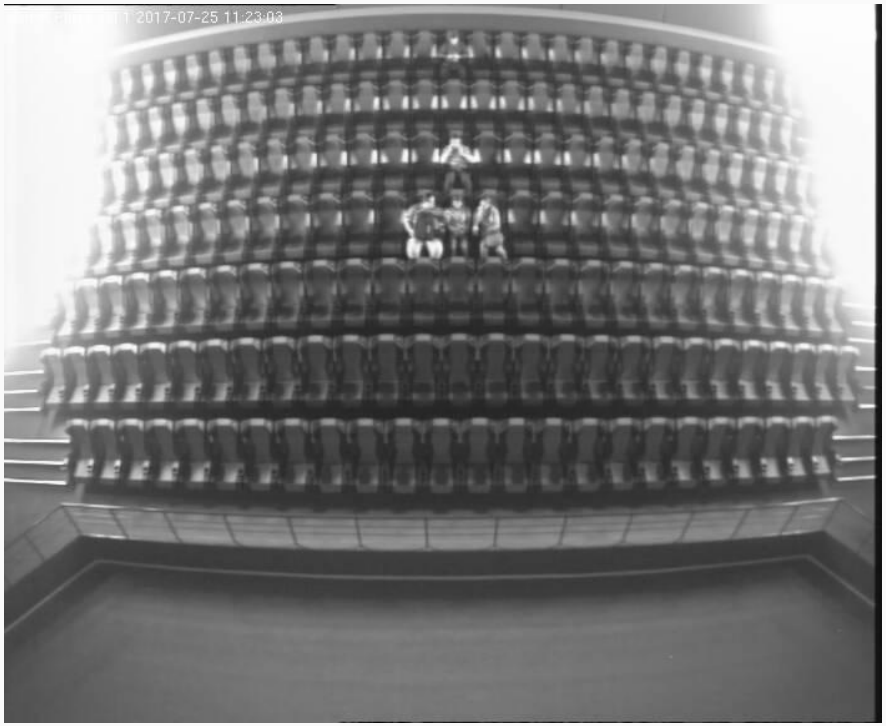
Ticket counting isn’t accurate enough

SocDem of those buying them isn’t either

But we have CCTV cameras!

Cinema Headcount and Socdem Classification

Data quality and diversity



Cinema Headcount and Socdem Classification

Data quality and diversity

Cinema Headcount and Socdem Classification

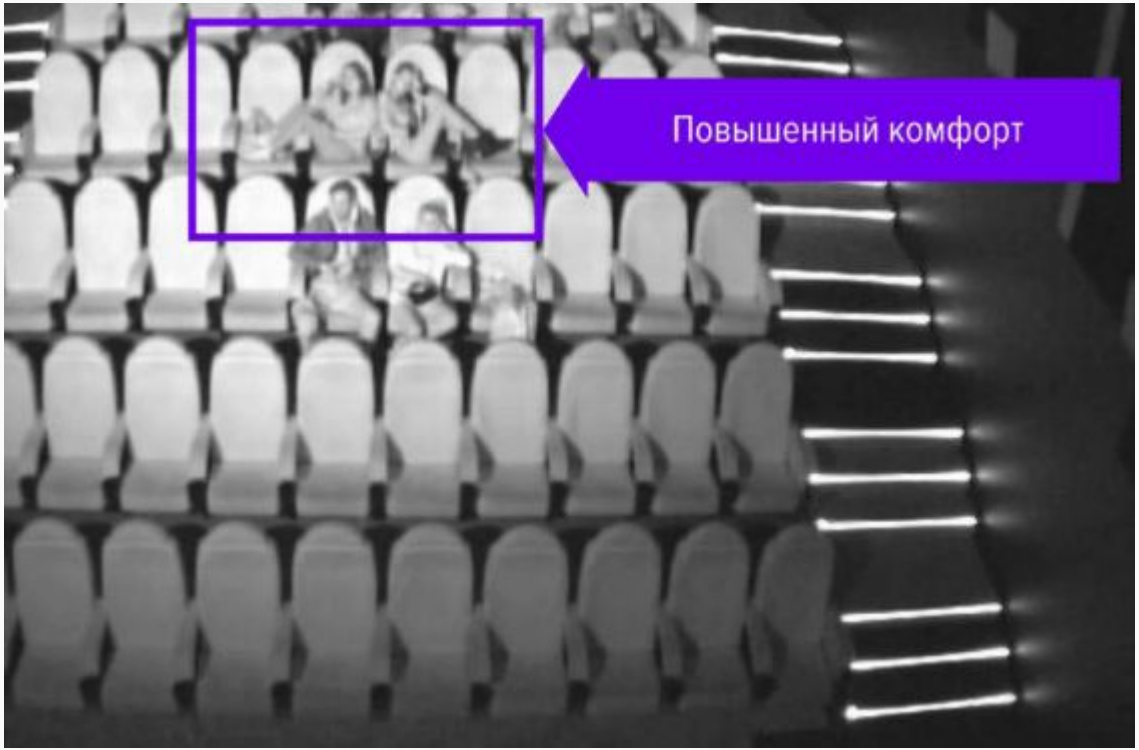
Data quality and diversity

Cinema Headcount and Socdem Classification

Data quality and diversity

Cinema Headcount and Socdem Classification

Data features

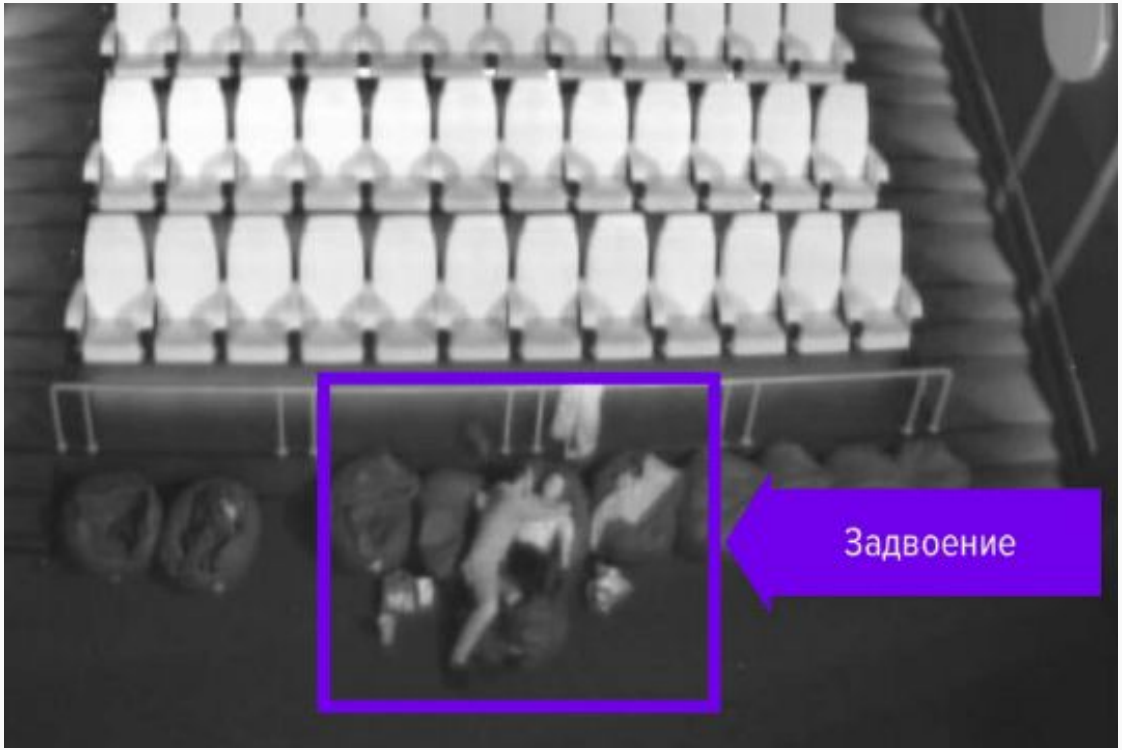


Cinema Headcount and Socdem Classification

Data features

Cinema Headcount and Socdem Classification

Data features

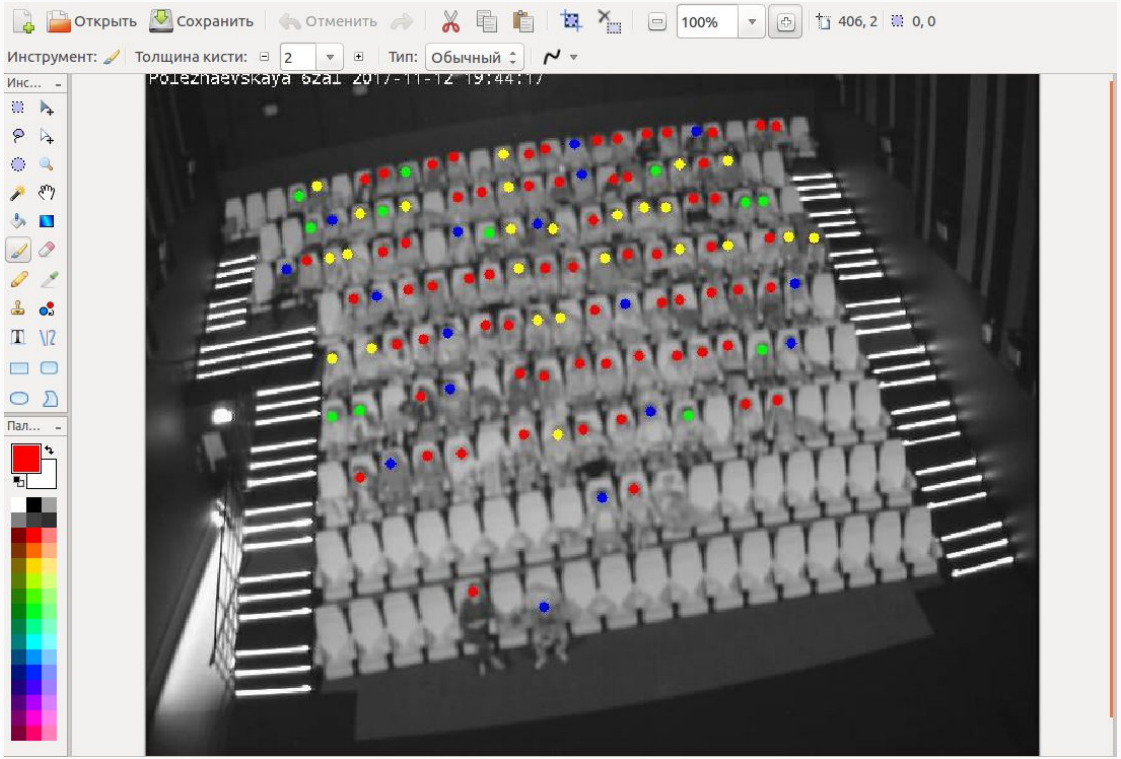


Cinema Headcount and Socdem Classification

The one approach is to find heads being unfortunately non-static

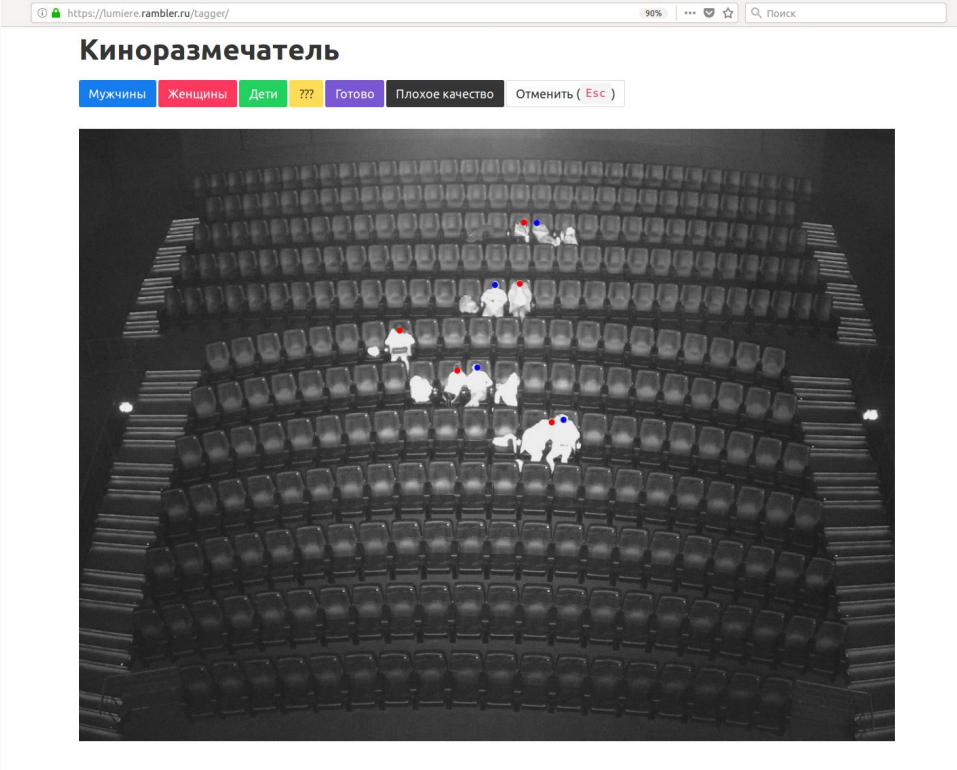
Cinema Headcount and Socdem Classification

The truth hurts — images don’t label themselves :(



Cinema Headcount and Socdem Classification

The truth hurts — images don’t label themselves :(

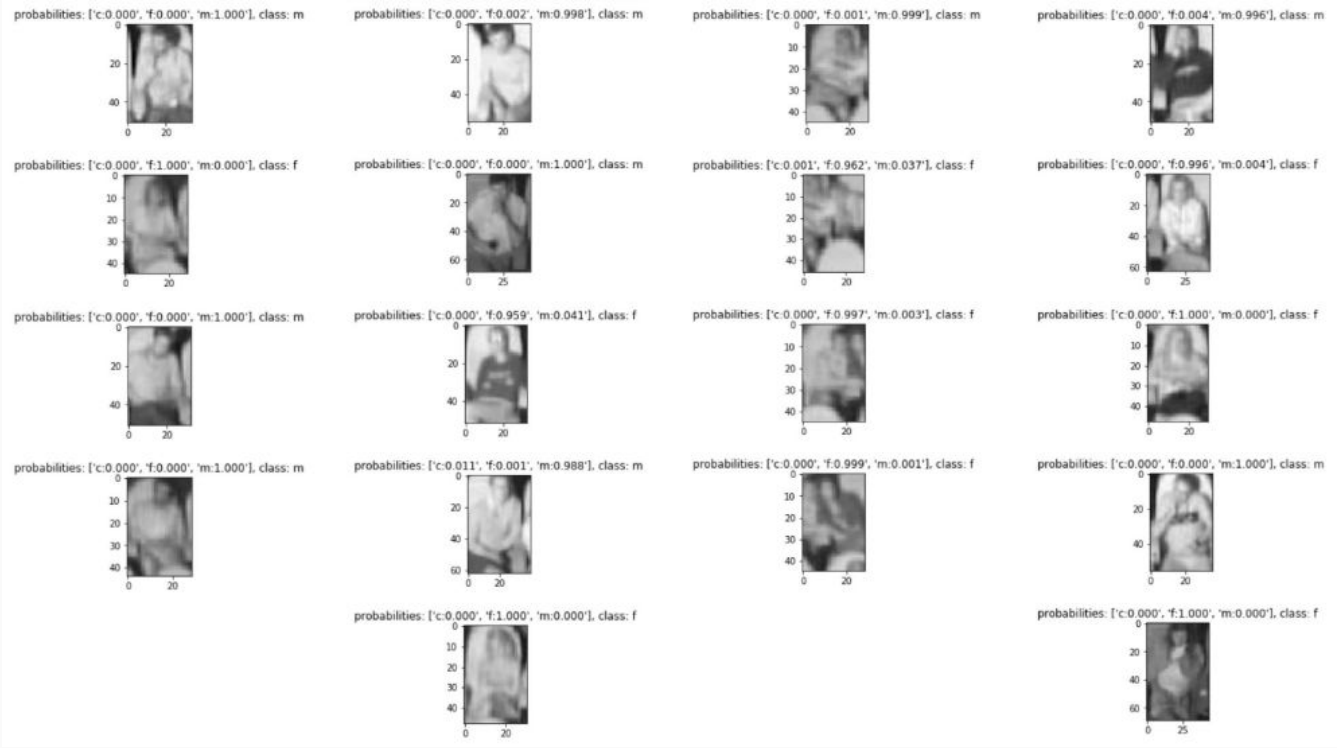


Cinema Headcount and Socdem Classification

Final segmentation examples

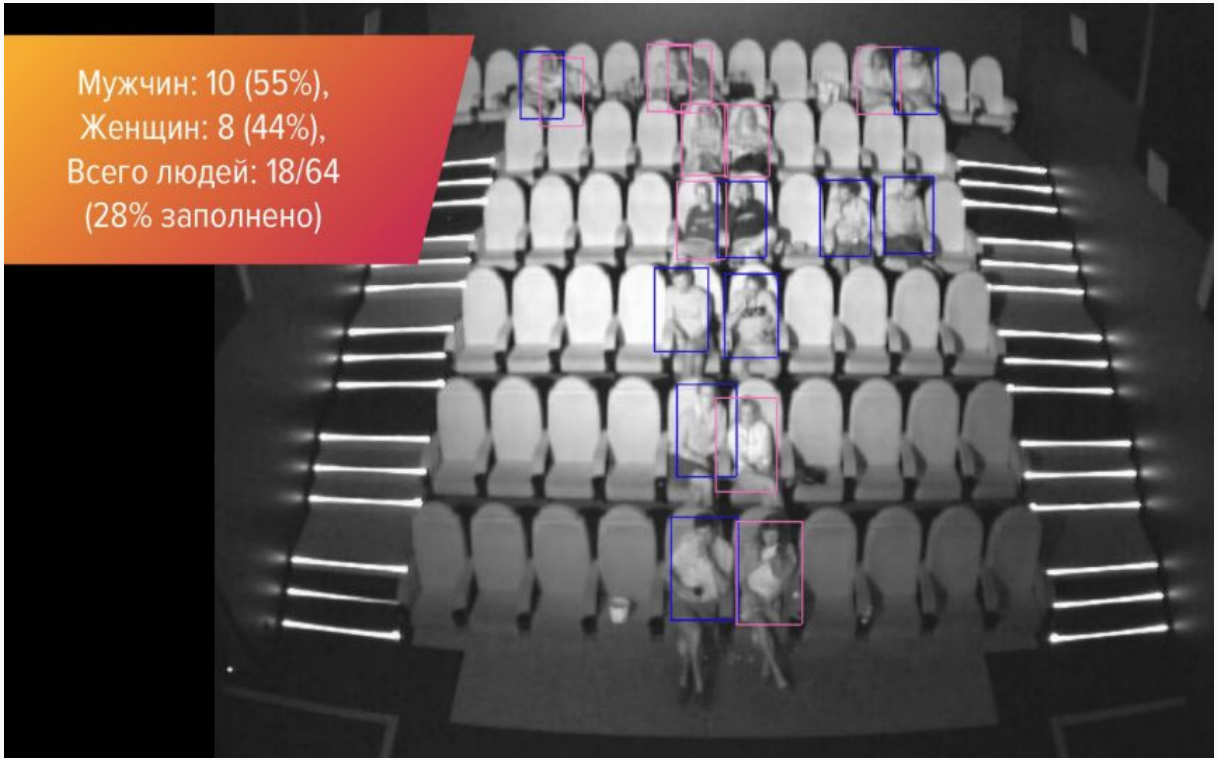
Cinema Headcount and Socdem Classification

Some general classifier to predict socdem attributes based on bodies found



Cinema Headcount and Socdem Classification

Final results



Cinema Headcount and Socdem Classification

Final “leaderboard”