Approaches to Solving Semantic Segmentation

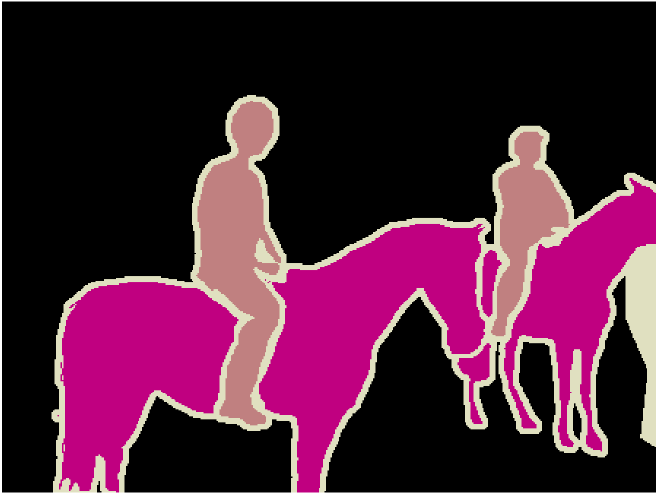
Sergey Rusakov

Lehigh University

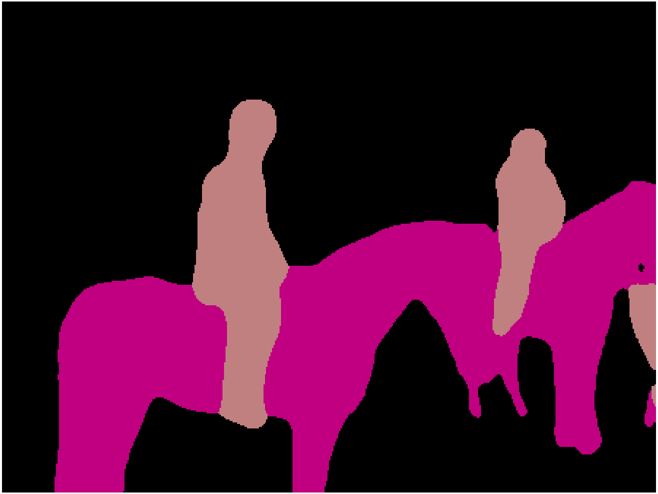
September 25, 2019

Problem description

Problem description

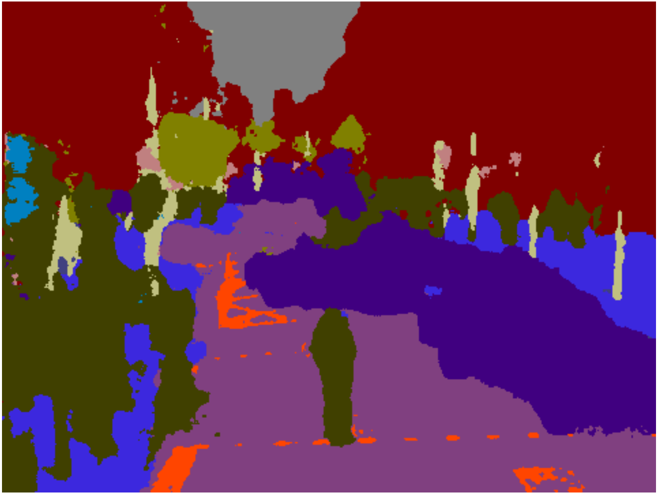


Problem description



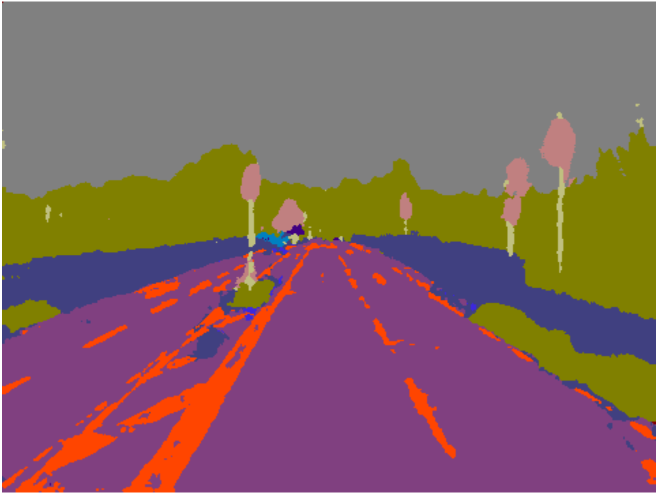
Problem description

Problem description



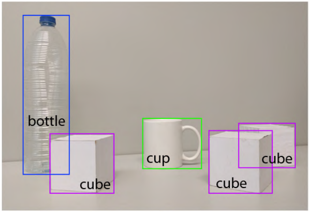
Problem description

Problem description

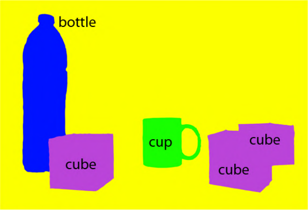


Problem description

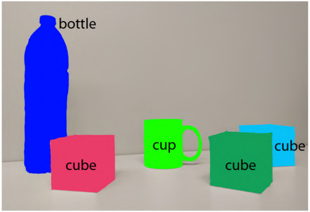
(a) Image classification



(b) Object localization



(c) Semantic segmentation



(d) Instance segmentation

Optimisation opportunities

$$\mathcal{X} \xrightarrow{f} \mathcal{Y}$$

- $\mathcal{X}$ is a set of three dimensional matrices, $\mathcal{X} \subseteq [0, 1]^{n \times m \times 3}$
- $\mathcal{Y}$ is a set of three dimensional matrices, $\mathcal{Y} \subseteq \mathcal{C}^{n \times m}$, where $\mathcal{C}$ is the set, that can be tailored to the specific needs, but generally somehow represents a collection of classes
  - $\mathcal{C} = \{1, 2, ..., C\}$, where C is the number of classes
  - $\mathcal{C} = \{e_1, e_2, ..., e_C\}$, where C is the number of classes and e_i is a one-hot vector with 1 at position i
  - $\mathcal{C} = \{e_1, e_2, ..., e_C\} \cup [0, 1]^3$

Problem description

Optimisation usually consists of variants of minimizing sum of pixel-wise cross entropy with your favorite first order method

$$\frac{1}{N}\min_{w}\sum_{i=1}^{N}\sum_{p} CE_p\left(\hat{f}_w(x_i), y_i\right)$$

where p goes through all pixels

Obvious optimisation opportunities

Instead of blindly playing with architectures, one can play with regularization

- Border smoothness
- Number of connected regions
- Any possible intuition about the desired result

Research horizons

Data from 2017 survey [2]

Name and Reference	Purpose	Year	Classes	Data	Resolution	Sequence	Synthetic/Real	Samples (training)	Samples (validation)	Samples (test)
PASCAL VOC 2012 Segmentation [27]	Generic	2012	21	2D	Variable	✗	R	1464	1449	Private
PASCAL-Context [28]	Generic	2014	540 (59)	2D	Variable	✗	R	10103	N/A	9637
PASCAL-Part [29]	Generic-Part	2014	20	2D	Variable	✗	R	10103	N/A	9637
SBD [30]	Generic	2011	21	2D	Variable	✗	R	8498	2857	N/A
Microsoft COCO [31]	Generic	2014	+80	2D	Variable	✗	R	82783	40504	81434
SYNTHIA [32]	Urban (Driving)	2016	11	2D	960×720	✗	S	13407	N/A	N/A
Cityscapes (fine) [33]	Urban	2015	30 (8)	2D	2048×1024	✓	R	2975	500	1525
Cityscapes (coarse) [33]	Urban	2015	30 (8)	2D	2048×1024	✓	R	22973	500	N/A
CamVid [34]	Urban (Driving)	2009	32	2D	960×720	✓	R	701	N/A	N/A
CamVid-Sturgess [35]	Urban (Driving)	2009	11	2D	960×720	✓	R	367	100	233
KITTI-Layout [36] [37]	Urban/Driving	2012	3	2D	Variable	✗	R	323	N/A	N/A
KITTI-Ros [38]	Urban/Driving	2015	11	2D	Variable	✗	R	170	N/A	46
KITTI-Zhang [39]	Urban/Driving	2015	10	2D/3D	1226×370	✗	R	140	N/A	112
Stanford background [40]	Outdoor	2009	8	2D	320×240	✗	R	725	N/A	N/A
SiftFlow [41]	Outdoor	2011	33	2D	256×256	✗	R	2688	N/A	N/A
Youtube-Objects-Jain [42]	Objects	2014	10	2D	480×360	✓	R	10167	N/A	N/A
Adobe's Portrait Segmentation [26]	Portrait	2016	2	2D	600×800	✗	R	1500	300	N/A
MINC [43]	Materials	2015	23	2D	Variable	✗	R	7061	2500	5000
DAVIS [44] [45]	Generic	2016	4	2D	480p	✓	R	4219	2023	2180
NYUDv2 [46]	Indoor	2012	40	2.5D	480×640	✗	R	795	654	N/A
SUN3D [47]	Indoor	2013	–	2.5D	640×480	✓	R	19640	N/A	N/A
SUNRGBD [48]	Indoor	2015	37	2.5D	Variable	✗	R	2666	2619	5050
RGB-D Object Dataset [49]	Household objects	2011	51	2.5D	640×480	✓	R	207920	N/A	N/A
ShapeNet Part [50]	Object/Part	2016	16/50	3D	N/A	✗	S	31, 963	N/A	N/A
Stanford 2D-3D-S [51]	Indoor	2017	13	2D/2.5D/3D	1080×1080	✓	R	70469	N/A	N/A
3D Mesh [52]	Object/Part	2009	19	3D	N/A	✗	S	380	N/A	N/A
Sydney Urban Objects Dataset [53]	Urban (Objects)	2013	26	3D	N/A	✗	R	41	N/A	N/A
Large-Scale Point Cloud Classification Benchmark [54]	Urban/Nature	2016	8	3D	N/A	✗	R	15	N/A	15

Research horizons

		Targets								
Name and Reference	**Architecture**	**Accuracy**	**Efficiency**	**Training**	**Instance**	**Sequences**	**Multi-modal**	**3D**	**Source Code**	**Contribution(s)**
Fully Convolutional Network [65]	VGG-16(FCN)	★	★	★	✗	✗	✗	✗	✓	Forerunner
SegNet [66]	VGG-16 + Decoder	★★★	★★	★	✗	✗	✗	✗	✓	Encoder-decoder
Bayesian SegNet [67]	SegNet	★★★	★	★	✗	✗	✗	✗	✓	Uncertainty modeling
DeepLab [68] [69]	VGG-16/ResNet-101	★★★	★	★	✗	✗	✗	✗	✓	Standalone CRF, atrous convolutions
MINC-CNN [43]	GoogLeNet(FCN)	★	★	★	✗	✗	✗	✗	✓	Patchwise CNN, Standalone CRF
CRFasRNN [70]	FCN-8s	★	★★	★★★	✗	✗	✗	✗	✓	CRF reformulated as RNN
Dilation [71]	VGG-16	★★★	★	★	✗	✗	✗	✗	✓	Dilated convolutions
ENet [72]	ENet bottleneck	★★	★★★	★	✗	✗	✗	✗	✓	Bottleneck module for efficiency
Multi-scale-CNN-Raj [73]	VGG-16(FCN)	★★★	★	★	✗	✗	✗	✗	✗	Multi-scale architecture
Multi-scale-CNN-Eigen [74]	Custom	★★★	★	★	✗	✗	✗	✗	✓	Multi-scale sequential refinement
Multi-scale-CNN-Roy [75]	Multi-scale-CNN-Eigen	★★★	★	★	✗	✗	★★	✗	✗	Multi-scale coarse-to-fine refinement
Multi-scale-CNN-Bian [76]	FCN	★★	★	★★	✗	✗	✗	✗	✗	Independently trained multi-scale FCNs
ParseNet [77]	VGG-16	★★★	★	★	✗	✗	✗	✗	✓	Global context feature fusion
ReSeg [78]	VGG-16 + ReNet	★★	★	★	✗	✗	✗	✗	✓	Extension of ReNet to semantic segmentation
LSTM-CF [79]	Fast R-CNN + DeepMask	★★★	★	★	✗	✗	✗	✗	✓	Fusion of contextual information from multiple sources
2D-LSTM [80]	MDRNN	★★	★★	★	✗	✗	✗	✗	✗	Image context modelling
rCNN [81]	MDRNN	★★★	★★	★	✗	✗	✗	✗	✓	Different input sizes, image context
DAG-RNN [82]	Elman network	★★★	★	★	✗	✗	✗	✗	✓	Graph image structure for context modelling
SDS [10]	R-CNN + Box CNN	★★★	★	★	★★	✗	✗	✗	✓	Simultaneous detection and segmentation
DeepMask [83]	VGG-A	★★★	★	★	★★	✗	✗	✗	✓	Proposals generation for segmentation
SharpMask [84]	DeepMask	★★★	★	★	★★★	✗	✗	✗	✓	Top-down refinement module
MultiPathNet [85]	Fast R-CNN + DeepMask	★★★	★	★	★★★	✗	✗	✗	✓	Multi path information flow through network
Huang-3DCNN [86]	Own 3DCNN	★	★	★	✗	✗	✗	★★★	✗	3DCNN for voxelized point clouds
PointNet [87]	Own MLP-based	★★	★	★	✗	✗	✗	★★★	✓	Segmentation of unordered point sets
Clockwork Convnet [88]	FCN	★★	★★	★	✗	★★★	✗	✗	✓	Clockwork scheduling for sequences
3DCNN-Zhang	Own 3DCNN	★★	★	★	✗	★★★	✗	✗	✓	3D convolutions and graph cut for sequences
End2End Vox2Vox [89]	C3D	★★	★	★	✗	★★★	✗	✗	✗	3D convolutions/deconvolutions for sequences

Research horizons

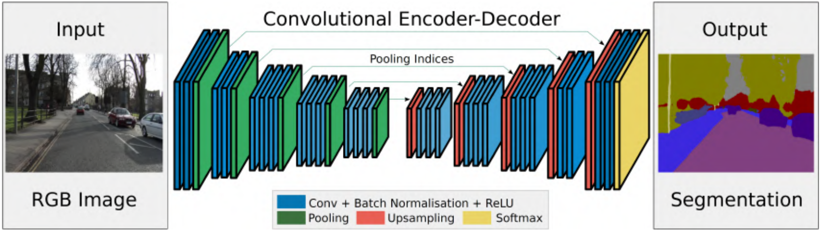
Notable detail - architectures are often modular, significant parts are just borrowed from classification context

- Is it worth to consider architecture search over vast blocks instead of individual weights/layers?
- Focus computational resources on connecting structures
- Even simple automatisation of exhaustive search over large architecture blocks can be beneficial, considering the plethora of existing results (more of a commercial opportunity; TensorFlow might already include the functionality)

SegNet [1], 2016

- Specifically for pixel-wise segmentation, initial intended use for road/roadside segmentation
- Novel encoder-decoder architecture (at the time)
- Very simple method overall

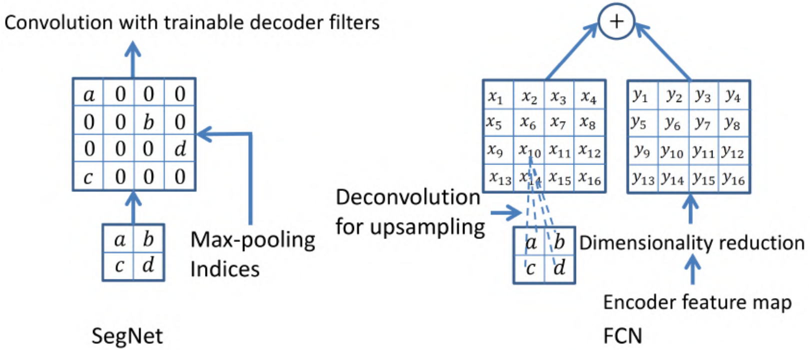
SegNet, 2016


Input
RGB Image
Convolutional Encoder-Decoder
Pooling Indices
Conv + Batch Normalisation + ReLU
Pooling
Upsampling
Softmax
Output
Segmentation

SegNet, 2016

- Encoder is VGG16 without dense layers
- Decoder uses saved indices from max-pooling with subsequent convolutions to restore the original size
- Removal of dense layers allows to literally use the CE pixel sum objective without resorting to training on separate regions
- Modular structure of the network allows for a more detailed analysis of decoder structure

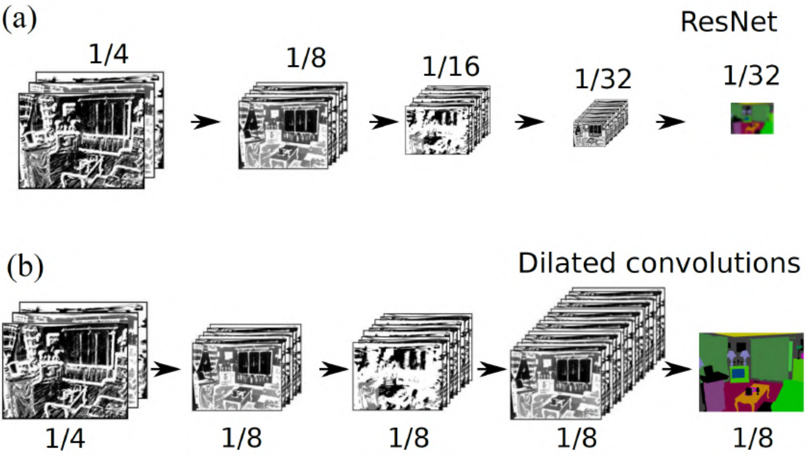
SegNet, 2016


Convolution with trainable decoder filters
a 0 0 0
0 0 b 0
0 0 0 d
c 0 0 0
a b
c d
Max-pooling Indices
SegNet
Deconvolution for upsampling
x1 x2 x3 x4
x5 x6 x7 x8
x9 x10 x11 x12
x13 x14 x15 x16
y1 y2 y3 y4
y5 y6 y7 y8
y9 y10 y11 y12
y13 y14 y15 y16
+
a b
c d
Dimensionality reduction
Encoder feature map
FCN

RefineNet [3], 2017

- SegNet, although attempting to restore image information during decoding, still loses some of it
- As mentioned, one way to try and improve the quality of the segmentation regions is maybe an addition of informed regularisation
- Another way is to make an informed choice of the architecture, trying to save low/mid/high level feature information along all levels of processing

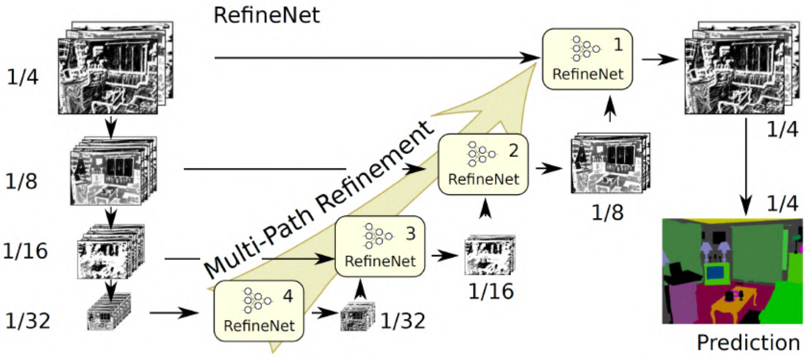
RefineNet, 2017



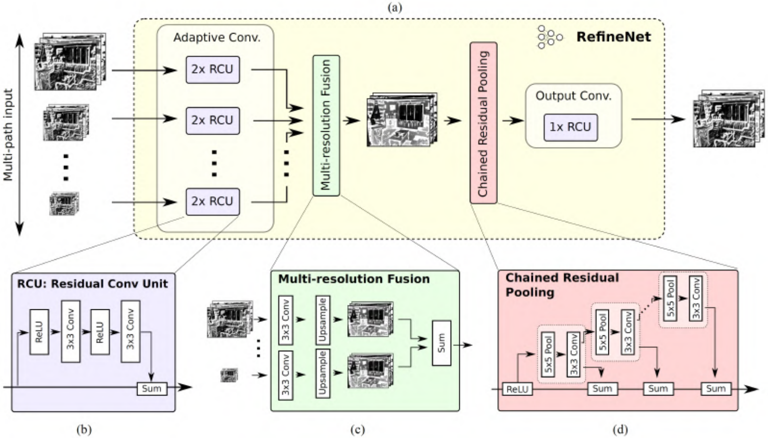
RefineNet, 2017

- ResNet convolutional architecture is motivated by lowering computational resources requirements
- Dilated convolution saves the resolution of an image, but still requires storage of large amounts of filter application results
- We want the benefit of both without disadvatages of any

RefineNet, 2017



RefineNet, 2017



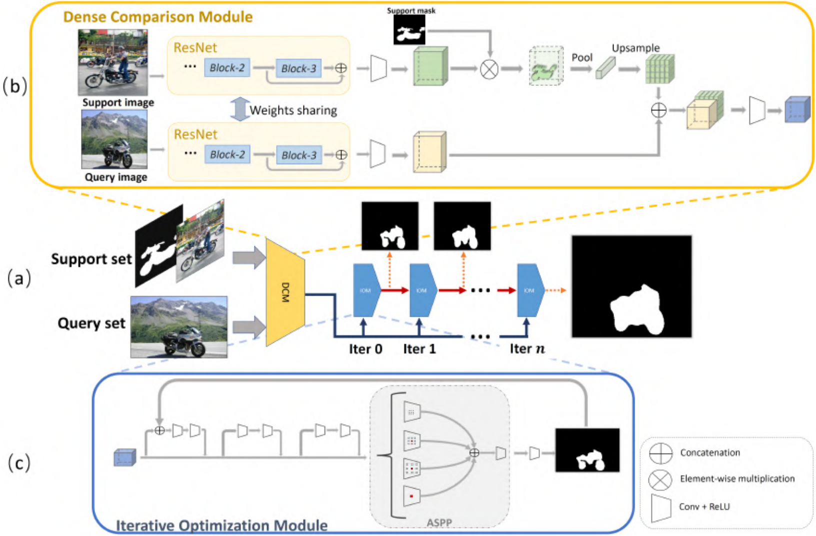
CANet [4], 2019

- Having architecture, that is empirically proven to be good is still not enough
- Sometimes the problem comes from class under-representation in data
- This can be argued to be an even more serious problem, than searching for architectures

CANet, 2018

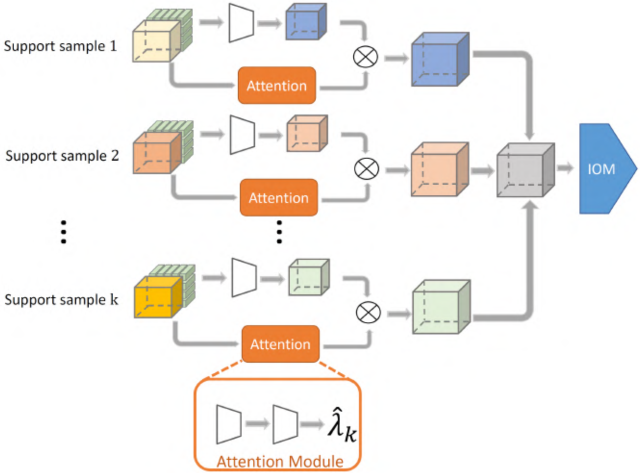
- We will modify the initial formulation through change in $\mathcal{X}$
- Now $\mathcal{X}$ is a set of **triplets** of three dimensional matrices, (x_T, x_S, B_S), where we want to obtain a segmentation of image x_T and x_S serves as "support" image, with B_S being its binary segmentation mask
- Value of the support image comes from the fact, that it can be taken from underrepresented class and, combined with input, still achieve good segmentation result on barely seen classes
- For $\mathcal{Y}$, we are only concerned with segmentation of a single object, so just a binary mask

CANet, 2018

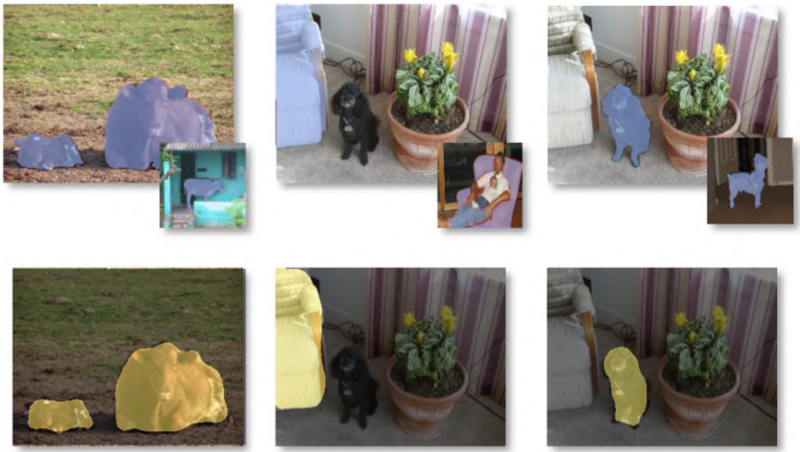


(b)
Dense Comparison Module
Support mask
Support image
ResNet
Block-2
Block-3
Weights sharing
Query image
Pool
Upsample
(a)
Support set
Query set
DCM
IOM
Iter 0
Iter 1
Iter n
(c)
ASPP
Iterative Optimization Module
Concatenation
Element-wise multiplication
Conv + ReLU

CANet, 2018


Support sample 1
Support sample 2
Support sample k
Attention
Attention
Attention
IOM
$\hat{\lambda}_k$
Attention Module

CANet, 2018



CANet, 2018

- Aside from the change in the problem formulation, significant parts of the architecture are borrowed
- ASPP is a module from DeepLabV3, which serves the same purpose as structure of RefineNet
- Only further supports the idea about "large scale" architecture search

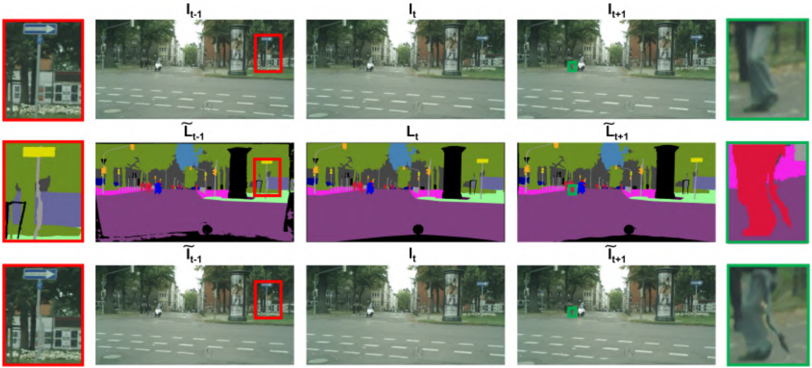
Automatic Data Augmentation [5], 2019

- Paper, again, deals with the issue of insufficient data for training
- Considered case is video with sparsely annotated frames
- Proposed solution is to use video prediction tools to simultaneously predict frames and labels

Automatic Data Augmentation, 2019

- Obvious idea - use existing frame prediction methods to predict future frames and apply the result on labels
- Particular implementation predicts (u, v) translation of the pixel in the frame and then applies this translation to corresponding label pixel
- Since we have access to all frames, we then pair known frames and label prediction
- Approach encounters some problems

Automatic Data Augmentation, 2019



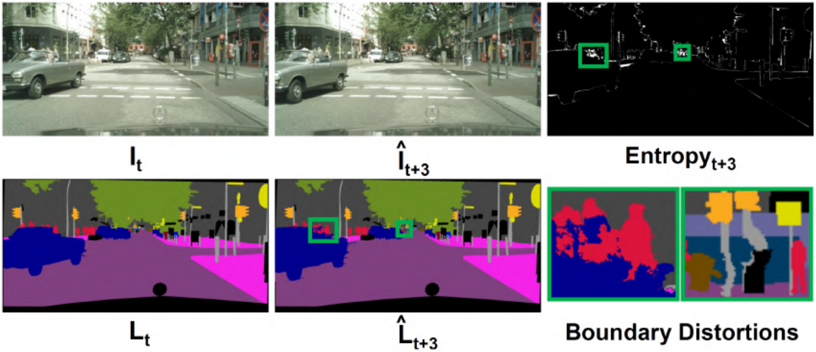
Automatic Data Augmentation, 2019

- Solution - pair **predicted** labels with **predicted** frames
- Predicted frames might be incorrect, but labeling will be more in line with them, which is our goal, when augmenting a data set
- We can even *condition our predictive model on future frames*, since the only information, that we don't have is label assignment; turns prediction into reconstruction

Automatic Data Augmentation, 2019

- Still, if want to construct labels even for several frames into the future, we need to deal with severe artifacts of the prediction model
- Proposed solution - instead of maximising a probability of one class for pixels, which are placed on the border between objects, we will maximise the joint probability of labels, corresponding to these classes
- Surprisingly, paper shows, that this helps, which allows authors to use up to 5 frames into past and future, effectively multiplying the size of the data set by 10

Automatic Data Augmentation, 2019



References I

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla.
Segnet: A deep convolutional encoder-decoder architecture for image segmentation.
CoRR, abs/1511.00561, 2015.

Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez.
A review on deep learning techniques applied to semantic segmentation.
CoRR, abs/1704.06857, 2017.

Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid.
Refinenet: Multi-path refinement networks for high-resolution semantic segmentation.
CoRR, abs/1611.06612, 2016.

References II

Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen.
Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning.
CoRR, abs/1903.02351, 2019.

Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn D. Newsam, Andrew Tao, and Bryan Catanzaro.
Improving semantic segmentation via video propagation and label relaxation.
CoRR, abs/1812.01593, 2018.